# KFM4GH6Q4M
# KFN8GH6Q4M
# KFKAGH6Q4M

## 4Gb Flex-MuxOneNAND M-die

# Revision History

## Document Title

**Flex-MuxOneNAND**

## Revision History

| Revision No. | History | Draft Date | Remark |
|---|---|---|---|
| 0.0 | 1. Initial issue. | Nov. 28, 2006 | Advanced |
| 0.1 | 1. Corrected errata.<br>2. Chapter 1.3 Product Features revised.<br>3. Chapter 2.8.16 Start Address8 Register F107 revised.<br>4. Chapter 2.8.18 Command Register F220h revised.<br>5. Chapter 2.8.19 System Configuration 1 Register F221h corrected errata.<br>6. Chapter 2.8.21 Controller Status Register F240h revised.<br>7. Chapter 2.8.22 Interrupt Status Register F241h revised.<br>8. Chapter 3.1.2 Load Data Into Buffer Command.<br>9. Chapter 3.3 Reset Mode Operation revised.<br>10. Chapter 3.3.1 Cold Reset Mode Operation revised.<br>11. Chapter 3.4.3 NAND Array Write Protection States corrected errata.<br>12. Chapter 3.4.3.1 Unlocked NAND Array Write Protection State.<br>13. Chapter 3.4.4 Data Protection Operation Flow Diagram revised.<br>14. Chapter 3.4.4 All Block Unlock Flow Diagram revised.<br>15. Chapter 3.6 Load Operation Flow Chart Diagram revised.<br>16. Chapter 3.6.1 Superload Operation revised.<br>17. Chapter 3.6.2 LSB Page Recovery Read updated.<br>18. Chapter 3.8 Synchronous Write revised.<br>19. Chapter 3.9 Program Operation Flow Diagram revised.<br>Program Interleave Flow Chart updated.<br>20. Chapter 3.9.1 Cache Program Operation Flow diagram revised.<br>21. Chapter 3.9.2 Interleave Cache Program Operation revised.<br>22. Chapter 3.10 Copy-Back Program Operation with Random Data Input Flow Chart revised.<br>23. Chapter 3.11.1 Block Erase Operation Flow Chart revised.<br>Erase Interleave Flow Chart updated.<br>24. Chapter 3.12 Partition Information Block corrected errata.<br>25. Chapter 3.12.1 PI Block Load Operation revised.<br>26. Chapter 3.12.2 PI Block Boundary Information setting updated.<br>27. OTP Operation revised.<br>28. Chapter 3.13.2 OTP Block Program Operation revised.<br>29. Chapter 3.13.3 OTP Block Lock Operation and Flow Chart revised.<br>30. Chapter 3.13.4 1st Block OTP Lock Operation revised.<br>31. Chapter 3.13.5 OTP and 1st Block OTP Lock Operation revised.<br>32. Chapter 3.16 Invalid Block Operation revised.<br>33. Chapter 4.1 Absolute Maximum Ratings revised.<br>34. Chapter 4.2 Operating Conditions revised.<br>35. Chapter 4.3 DC Characteristics revised.<br>36. Chapter 5.1 AC Test Conditions revised.<br>37. Chapter 5.3 Valid Block Characteristics revised.<br>38. Chapter 5.4 AC Characteristics for Synchronous Burst Read revised.<br>39. Chapter 5.8 AC Characteristics for Burst Write Operation revised.<br>40. Chapter 5.9 AC Characteristics for Load/Program/Erase Performance revised.<br>41. Chapter 6.15 Cold Reset Timing revised.<br>42. Chapter 7.3 Partition of Flex-MuxOneNAND corrected errata. | Aug. 13, 2007 | Preliminary |

# Revision History

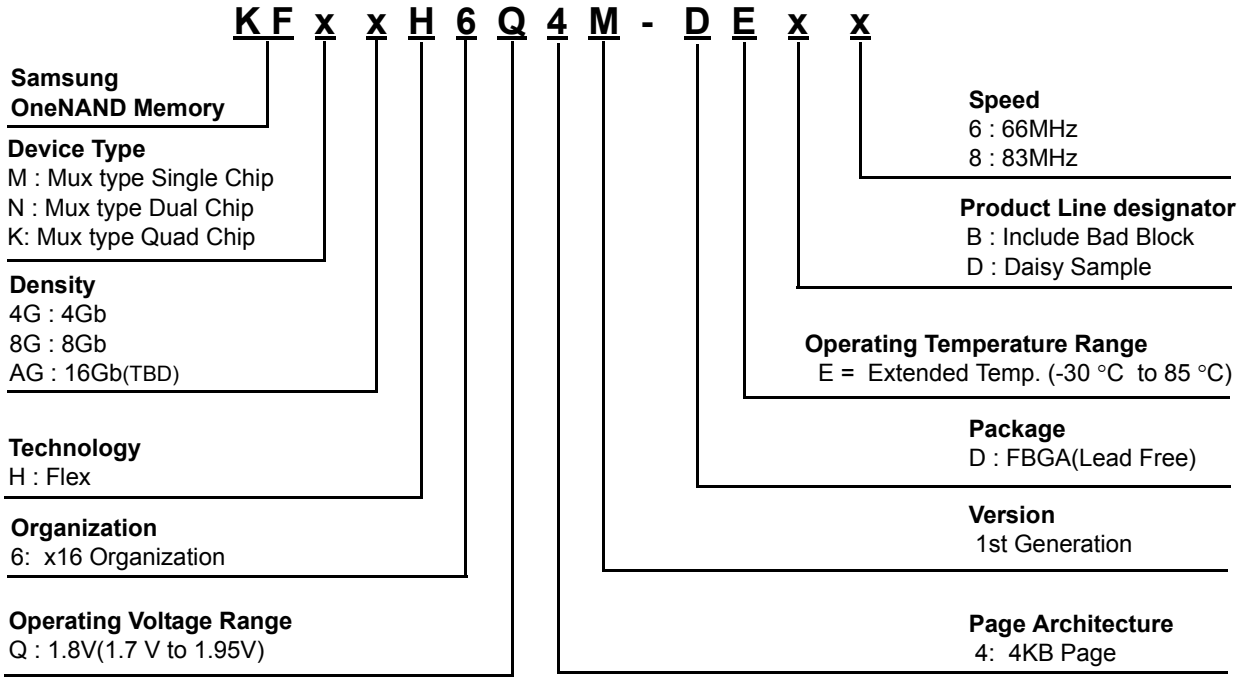
| Revision No. | History | Draft Date | Remark |
|---|---|---|---|
| 0.2 | 1. Corrected errata.<br>2. Chapter 2.1 Detailed Product Description revised.<br>3. Chapter 2.2 Definitions revised.<br>4. Chapter 2.8.3 Device ID Register F001h(R) revised.<br>5. Chapter 2.8.8 Technology Register F006h(R) revised.<br>6. Chapter 2.8.10 Start Address2 Register F101h(R/W) revised.<br>7. Chapter 2.8.16 Start Address8 Register F107h(R/W) revised.<br>8. Chapter 2.8.18 Command Register F220h(R/W) revised.<br>9. Chapter 2.8.22 Interrupt Status Register F241h(R/W) revised.<br>10. Chapter 3.1 Command Based Operation revised.<br>11. Chapter 3.3 Reset Mode Operation revised.<br>12. Chapter 3.4.3 NAND Array Write Protection States revised.<br>13. Chapter 3.4.3.1 Unlocked NAND Array Write Protection State revised.<br>14. Chapter 3.4.3.3 Locked-tight NAND Array Write Protection State revised.<br>15. Chapter 3.4.4 NAND Flash Array Write Protection State Diagram revised.<br>16. Chapter 3.6.2 LSB Page Recovery Read revised.<br>17. Chapter 3.7.2 Synchronous Read Mode Operation revised.<br>18. Chapter 3.7.2.1 Continuous Linear Burst Read Operation revised.<br>19. Chapter 3.9 Program Operation revised.<br>20. Chapter 3.9.1 Cache Program Operation revised.<br>21. Chapter 3.9.2 Interleave Cache Program Operation revised.<br>22. Chapter 3.11.1 Block Erase Operation revised.<br>23. Chapter 3.11.2 Erase Suspend / Erase Resume Operation revised.<br>24. Chapter 3.12 Partition Information (PI) Block(SLC Only) revised.<br>25. Chapter 3.12.1 PI Block Boundary Information setting revised.<br>26. Chapter 3.12.1.1 PI Block Access mode entry revised.<br>27. Chapter 3.12.1.2 PI Block Erase revised.<br>28. Chapter 3.12.1.3 PI Block Program Operation revised.<br>29. Chapter 3.12.1.4 PI Update revised.<br>30. Chapter 3.13 OTP Operation (SLC only) revised.<br>31. Chapter 3.13.1 OTP Block Load Operation revised.<br>32. Chapter 3.16.2 Invalid Block Replacement Operation revised.<br>33. Chapter 5.5 AC Characteristics for Asynchronous Read revised.<br>34. Chapter 6.3 Asynchronous Read(VA Transition Before AVD Low) tOEH removed.<br>35. Chapter 6.4 Asynchronous Read(VA Transition After AVD Low) tOEH removed.<br>36. Chapter 7.4 DDP and QDP Description inserted. | Oct. 30, 2007 | Preliminary |
| 1.0 | 1. New Format(font size, color etc.)<br>2. Corrected errata.<br>3. Added a comment(Chapter 3.11.1 & 3.12.1.2 & 3.12.1.3 & 3.12.1.4)<br>4. Chapter 2.8.17 Start Buffer Register F200h (R/W) revised.<br>5. Chapter 3.1.2 Load Data Into Buffer Command revised.<br>6. Chapter 3.12.2 PI Block Load Operation revised.<br>7. Chapter 4.3 DC Characteristics revised. | Feb. 04, 2008 | Final |
| 1.1 | 1. Chapter 3.6.2 LSB Page Recovery read flow chart revised.<br>2. Chapter 3.9.1 Cache Program Operation revised.<br>3. Chapter 3.13.1 OTP Block Read Operation Flow Chart revised.<br>4. Chapter 3.13.2 OTP Block Program Operation Flow Chart revised.<br>5. Chapter 3.13.3 OTP Block Lock Operation Flow Chart revised.<br>6. Chapter 3.13.4 1st Block OTP Lock Operation revised.<br>7. Chapter 3.13.5 OTP and 1st Block OTP Lock Operation Flow Chart revised. | Aug. 07, 2008 | Final |

# 1.0 INTRODUCTION

This specification contains information about the Samsung Electronics Company Flex-MuxOneNAND™, Flash memory product family. Section 1.0 includes a general overview, revision history, and product ordering information.
Section 2.0 describes the Flex-MuxOneNAND device. Section 3.0 provides information about device operation. Electrical specifications and timing waveforms are in Sections 4.0 through 6.0. Section 7.0 provides additional application and technical notes pertaining to use of the Flex-MuxOneNAND. Package dimensions are found in Section 8.0

| Density | Part No. | Vcc(core & IO) | Temperature | PKG |
|---|---|---|---|---|
| 4Gb | KFM4GH6Q4M-DEBx | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |
| 8Gb | KFN8GH6Q4M-DEBX | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |
| 16Gb(TBD) | KFKAGH6Q4M-DEBX | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |

## 1.1 Ordering Information



## 1.2 General Overview

Flex-MuxOneNAND™ is a monolithic integrated circuit with a NAND Flash array using a NOR Flash interface.

The chip integrates system features including:

- A BootRAM(1KB) and bootloader
- 4KB DataRAM buffers
- A High-Speed x16 Host Interface
- On-chip Error Correction
- On-chip NOR interface controller

This on-chip integration enables system designers to reduce external system logic and use high-density NAND Flash in applications that would otherwise have to use more NOR components.

Flex-MuxOneNAND takes advantage of the higher performance NAND program time, low power, and high density and combines it with the synchronous read performance of NOR. The NOR Flash host interface makes Flex-MuxOneNAND an ideal solution for mobile applications that have large, advanced multimedia applications and operating systems and need high performance.

When integrated into a Samsung Multi-Chip-Package with Samsung Mobile DDR SDRAM, designers can complete a high-performance, small footprint solution.

The device operates up to a maximum host-driven clock frequency of 66MHz / 83MHz for synchronous reads at Vcc(or Vccq. Refer to chapter 4.2) with 4~7-clock latency. Appropriate wait cycles are determined by programmable read latency.

Flex-MuxOneNAND provides for multiple sector read operations by assigning the number of sectors to be read in the sector counter register. The device includes one block-sized OTP (One Time Programmable) area and user-controlled 1st block OTP(Block 0) that can be used to increase system security or to provide identification capabilities.

## 1.3 Product Features

**Device Architecture**

- Design Technology: M die
- Supply Voltage: 1.8V (1.7V ~ 1.95V)
- Host Interface: 16 bit
- 5KB Internal BufferRAM: 1KB BootRAM, 4KB DataRAM
- NAND Array: SLC : (4K+128)B Page Size
  (256K+8K)B Block Size (64pages)
  MLC : (4K+128)B Page Size
  (512K+16K)B Block Size (128pages)

**Device Performance**

- Host Interface Type:
  Synchronous Burst Read
  - Up to 66MHz / 83MHz clock frequency
  - Linear Burst 4-, 8-, 16-, 32-words with wrap around
  - Continuous 1K words Sequential Burst

  Synchronous Write
  - Up to 66MHz / 83MHz clock frequency
  - Linear Burst 4-, 8-, 16-, 32-, 1K-words with wrap around
  - Continuous 1K words Sequential Burst

  Asynchronous Random Read
  - 76ns access time

  Asynchronous Random Write
- Programmable Burst Read Latency: Latency 3,4(Default),5,6 and 7
  1~40MHz : Latency 3 available
  1~66MHz : Latency 4,5,6 and 7 available
  Over 66MHz : Latency 6,7 available
- Multiple Reset Modes: Cold/Warm/Hot/NAND Flash Core Reset
- Low Power Dissipation: Typical Power,
  - Standby current : 10uA (Single)
  - Synchronous Burst Read current(66MHz/83MHz, single) : 20/25mA
  - Synchronous Burst Write current(66MHz/83MHz, single) : 20/25mA
  - Load current : 50mA
  - Program current : 35mA
  - Erase current : 40mA
- Reliable CMOS Floating-Gate Technology
  - Endurance : 50K Program/Erase Cycles (SLC)
    10K Program/Erase Cycles (MLC)
  - Data Retention : 10 Years(SLC) /10 Years(MLC)

**System Hardware**

- Voltage detector generating internal reset signal from Vcc
- Hardware reset input ($\overline{RP}$)
- Data Protection Modes
  - Write Protection for BootRAM
  - Write Protection for NAND Flash Array
  - Write Protection during power-up
  - Write Protection during power-down
- User-controlled One Time Programmable(OTP) area
- Internal 4bit ECC
- Internal Bootloader supports Booting Solution in system
- Handshaking Feature
  - INT pin indicates Ready / Busy
  - Polling the interrupt register status bit
- Detailed chip information
  - by ID register

**Package size**

- 4G products: 63ball, 10mm x 13mm x max 1.0mmt , 0.8mm ball pitch FBGA
- 8G products: 63ball, 10mm x 13mm x max 1.2mmt , 0.8mm ball pitch FBGA
- 16G products(TBD): 63ball, 10mm x 13mm x max 1.4mmt , 0.8mm ball pitch FBGA (TBD)

# 2.0 DEVICE DESCRIPTION

## 2.1 Detailed Product Description

The Flex-MuxOneNAND is an advanced generation, high-performance MLC NAND-based Flash memory(Which can be programmed as both SLC and MLC).

It integrates on-chip a convertible(SLC and MLC) NAND Flash Array memory with two independent data buffers, boot RAM buffer, a page buffer for the Flash array, and a one-time-programmable block.

The combination of these memory areas enable high-speed pipelining of reads from host, BufferRAM, Page Buffer, and NAND Flash Array.

Clock speeds up to 66MHz / 83MHz with a x16 wide I/O yields a 83MByte/second in SLC and 71MByte/second in MLC read bandwidth

The Flex-MuxOneNAND also includes a Boot RAM and boot loader. This enables the device to efficiently load boot code at device startup from the NAND Array without the need for off-chip boot device.

One block of the NAND Array is set aside as an OTP memory area, and 1st Block (Block 0) can be used as OTP area. This area, available to the user, can be configured and locked with secured user information.

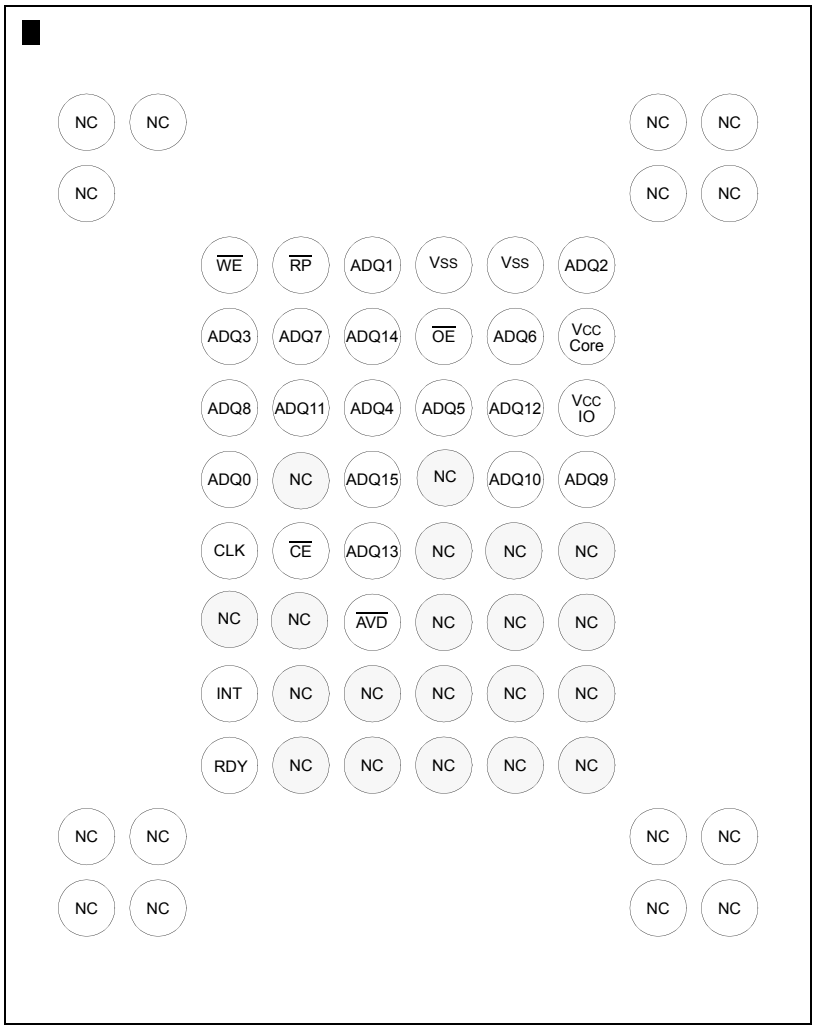
On-chip controller interfaces enable the device to operate in systems without NAND Host controllers.

## 2.2 Definitions

| | |
|---|---|
| B (capital letter) | Byte, 8bits |
| W (capital letter) | Word, 16bits |
| b (lower-case letter) | Bit |
| ECC | Error Correction Code |
| Calculated ECC | ECC that has been calculated during a load or program access |
| Written ECC | ECC that has been stored as data in the NAND Flash array or in the BufferRAM |
| BufferRAM | On-chip internal buffer consisting of BootRAM and DataRAM |
| BootRAM | A 1KB portion of the BufferRAM reserved for Boot Code buffering |
| DataRAM | A 4KB portion of the BufferRAM reserved for Data buffering (2KB x2) |
| Sector | Part of a Page of which 512B is the main data area and 16B is the spare data area. |
| Data unit | Possible data unit to be read from memory to BufferRAM or to be programmed to memory.<br>- 4224B of which 4096 is in main area and 128B in spare area |
| DDP | Dual Die Package |
| QDP | Quad Die Package |
| OTP | One Time Programmable |

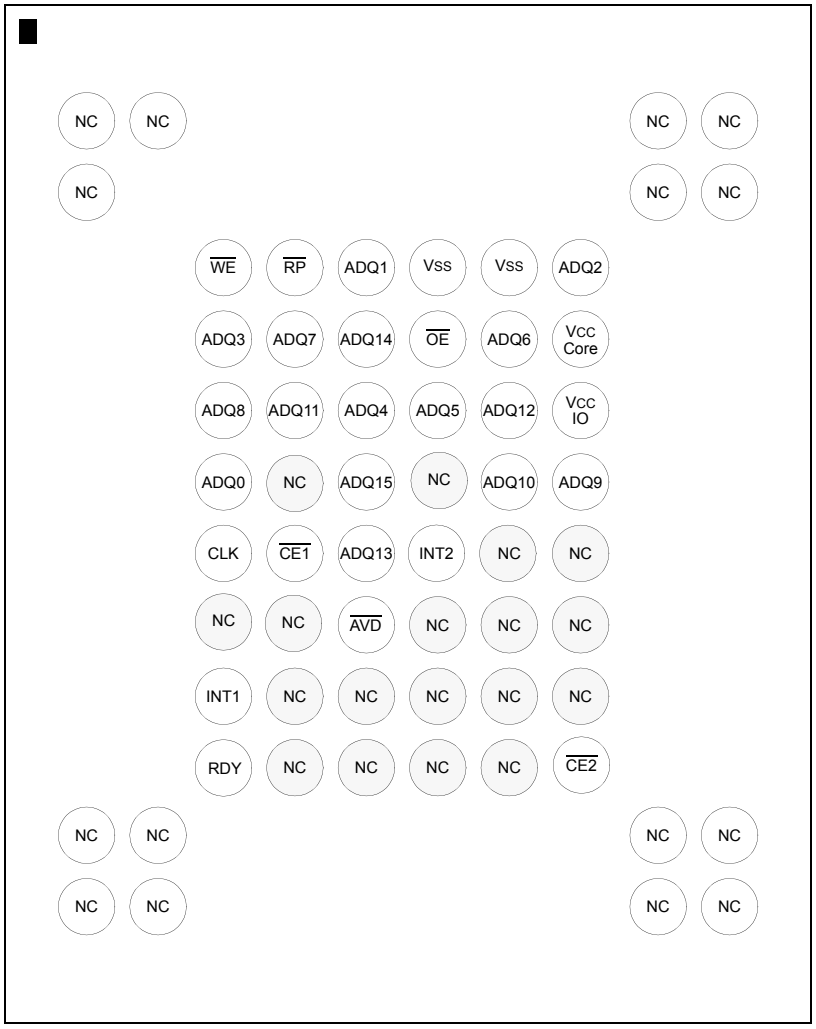
## 2.3 Pin Configuration

### 2.3.1 4Gb (KFM4GH6Q4M) / 8Gb (KFN8GH6Q4M)




**(TOP VIEW, Balls Facing Down)**
**63ball FBGA Flex-MuxOneNAND Chip**
63ball, 10mm x 13mm x max 1.0mmt , 0.8mm ball pitch FBGA(4Gb)
63ball, 10mm x 13mm x max 1.2mmt , 0.8mm ball pitch FBGA (8Gb)

### 2.3.2 16Gb Product (KFKAGH6Q4M) (TBD)



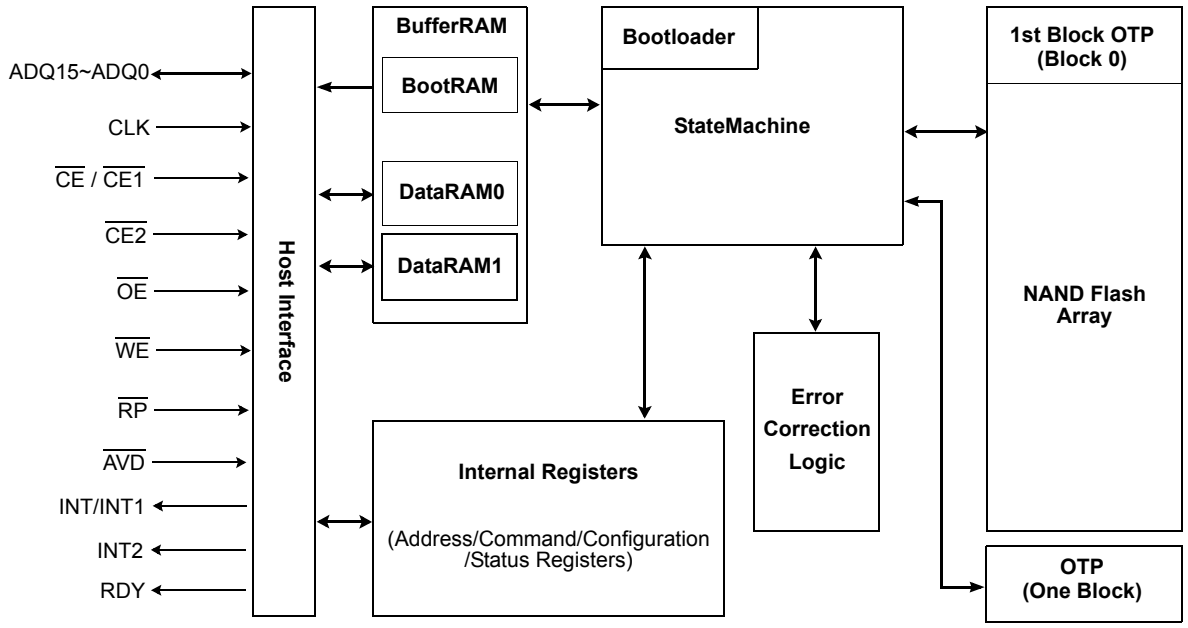

**(TOP VIEW, Balls Facing Down)**
**63ball FBGA OneNAND Chip**
63ball, 10mm x 13mm x max 1.4mmt , 0.8mm ball pitch FBGA

## 2.4 Pin Description

| Pin Name | Type | Nameand Description |
|---|---|---|
| ***Host Interface*** | | |
| ADQ15~ADQ0 | I/O | **Multiplexed Address/Data bus**<br>- Inputs for addresses during read operation, which are for addressing BufferRAM & Register.<br>- Inputs data during program and commands for all operations, outputs data during memory array/ register read cycles.<br>Data pins float to high-impedance when the chip is deselected or outputs are disabled. |
| INT / INT1 | O | **Interrupt**<br>Notifies the Host when a command is completed. After power-up, it is at hi-z condition. Once IOBE is set to 1, it does not float to hi-z condition even when $\overline{CE}$ is disabled or $\overline{OE}$ is disabled. Especially, only when reset(Cold, Warm, Hot, NAND Flash Core) command in DDP are issued, it operates as open drain output with internal resistor (~50Kohm). The INT is the interrupt for Single or DDP device. The INT1 is the interrupt for the first DDP device(KFN8GH6Q4M) in QDP(KFKAGH6Q4M) |
| INT2 | O | **Interrupt**<br>The INT2 is the interrupt for the second DDP device(KFN8GH6Q4M) in QDP(KFKAGH6Q4M) |
| RDY | O | **Ready**<br>Indicates data valid in synchronous read modes and is activated while $\overline{CE}$ is low |
| CLK | I | **Clock**<br>CLK synchronizes the device to the system bus frequency in synchronous read mode.<br>The first rising edge of CLK in conjunction with $\overline{AVD}$ low latches address input. |
| $\overline{WE}$ | I | **Write Enable**<br>$\overline{WE}$ controls writes to the bufferRAM and registers. Datas are latched on the $\overline{WE}$ pulse's rising edge |
| $\overline{AVD}$ | I | **Address Valid Detect**<br>Indicates valid address presence on address inputs. During asynchronous read operation, all addresses are valid while $\overline{AVD}$ is low, and during synchronous read operation, all addresses are latched on CLK's rising edge while $\overline{AVD}$ is held low for one clock cycle.<br>> Low : for asynchronous mode, indicates valid address; for burst mode, causes starting address to be latched on rising edge on CLK<br>> High : device ignores address inputs |
| $\overline{RP}$ | I | **Reset Pin**<br>When low, $\overline{RP}$ resets internal operation of Flex-MuxOneNAND. $\overline{RP}$ status is do not care during power-up and bootloading. When high, $\overline{RP}$ level must be equivalent to Vcc-IO / Vccq level. |
| $\overline{CE}$ / $\overline{CE1}$ | I | **Chip Enable**<br>$\overline{CE}$-low activates internal control logic, and $\overline{CE}$-high deselects the device, places it in standby state, and places DQ in Hi-Z.<br>The $\overline{CE}$ input enables device for Single or DDP .<br>The $\overline{CE1}$ input enables the first DDP device(KFN8GH6Q4M) in QDP(KFKAGH6Q4M) |
| $\overline{CE2}$ | I | **Chip Enable**<br>The $\overline{CE2}$ input enables the second DDP device(KFN8GH6Q4M) in QDP(KFKAGH6Q4M) |
| $\overline{OE}$ | I | **Output Enable**<br>$\overline{OE}$-low enables the device's output data buffers during a read cycle. |
| ***Power Supply*** | | |
| VCC-Core / Vcc | | **Power for Flex-MuxOneNAND Core**<br>This is the power supply for Flex-MuxOneNAND Core. |
| VCC-IO / Vccq | | **Power for Flex-MuxOneNAND I/O**<br>This is the power supply for Flex-MuxOneNAND I/O<br>Vcc-IO / Vccq is internally separated from Vcc-Core / Vcc. |
| VSS | | **Ground for Flex-MuxOneNAND** |
| ***etc.*** | | |
| DNU | | **Do Not Use**<br>Leave it disconnected. These pins are used for testing. |
| NC | | **No Connection**<br>Lead is not internally connected. |

**NOTE :**
Do not leave power supply(Vcc-Core/Vcc-IO, VSS) disconnected.

## 2.5 Block Diagram




## 2.6 Memory Array Organization

The Flex-MuxOneNAND architecture integrates several memory areas on a single chip.

### 2.6.1 Internal (NAND Array) Memory Organization

The on-chip internal memory is a convertible(SLC and MLC) NAND array used for data storage and code. The internal memory is divided into a main area and a spare area.

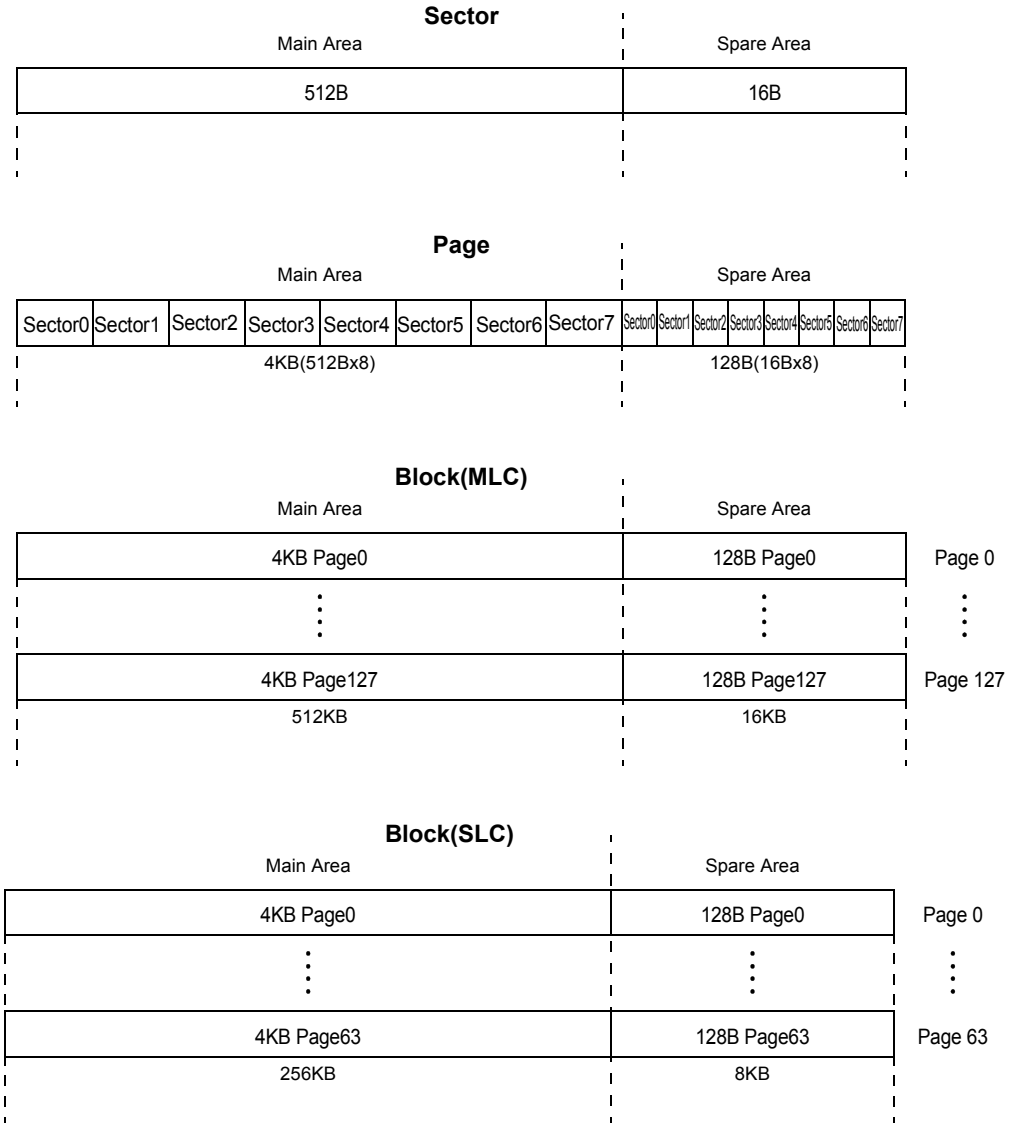
**Main Area**
The main area is the primary memory array. A block incorporates 64pages(SLC) or 128pages(MLC). A main page size is 4KB and a main page is comprised of 8 sectors each size of which is 512Byte.

**Spare Area**
The spare area is used for invalid block information and ECC storage. Spare area internal memory is associated with corresponding main area memory. A spare page size is 128B and a spare page is comprised of 8 sectors each size of which is 16Byte.

**Internal Memory Array Information**

| Area | Block | Page | Sector |
|---|---|---|---|
| Main(SLC) | 256KB | 4KB | 512B |
| Main(MLC) | 512KB | | |
| Spare(SLC) | 8KB | 128B | 16B |
| Spare(MLC) | 16KB | | |

**Internal Memory Array Organization**

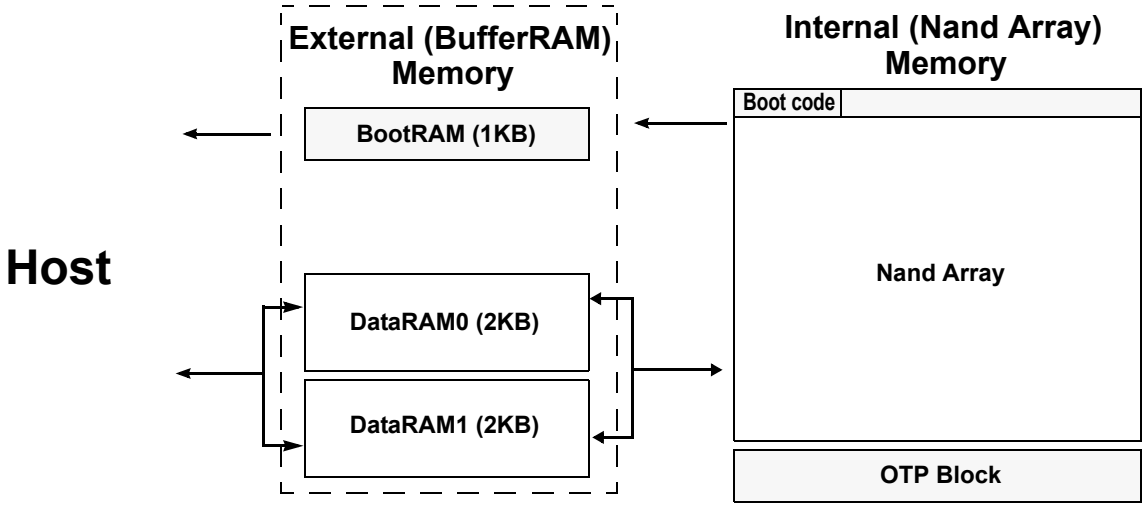


### 2.6.2 External (BufferRAM) Memory Organization

The on-chip external memory is comprised of 3 buffers used for Boot Code storage and data buffering.

The BootRAM is a buffer that receives Boot Code from the internal memory and makes it available to the host at start up.

There are 4KB bi-directional data buffers(2KB x2), DataRAM0 and DataRAM1. During Boot Up, the BootRam is used by the host to initialize the main memory, and deliver boot code from NAND Flash core to host.

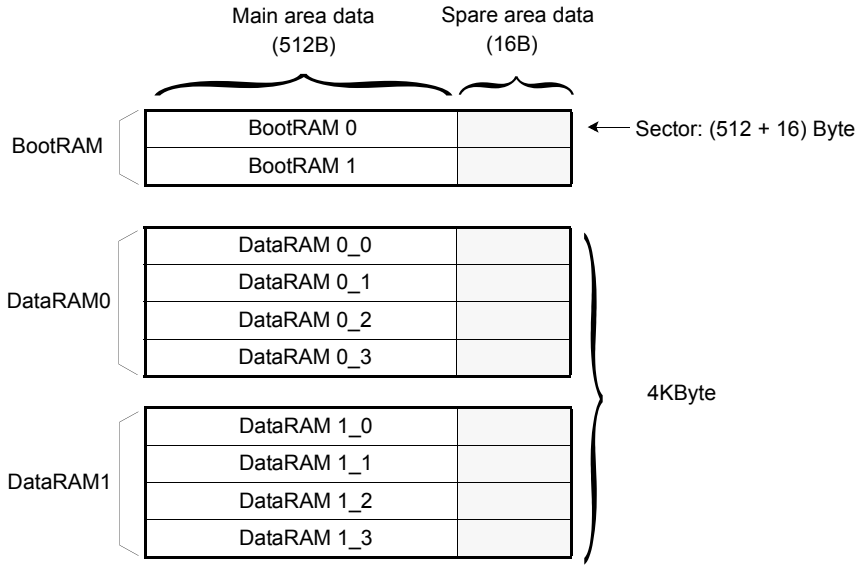



The external memory is divided into a main area and a spare area. Each buffer is the equivalent size of a Sector.
The main area data is 512B. The spare area data is 16B.

**External Memory Array Information**

| Area | | BootRAM | DataRAM0 | DataRAM1 |
|---|---|---|---|---|
| Total Size | | 1KB+32B | 2KB+64B | 2KB+64B |
| Number of Sectors | | 2 | 4 | 4 |
| Sector | Main | 512B | 512B | 512B |
| | Spare | 16B | 16B | 16B |

**External Memory Array Organization**



## 2.7 Memory Map

The following tables are the memory maps for the Flex-MuxOneNAND.

### 2.7.1 Internal (NAND Array) Memory Organization

The following tables show the Internal Memory address map in word order.

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| **Block0** | **0000h** | **0000h~00FCh** | **256KB** | Block32 | 0020h | | |
| Block1 | 0001h | | | Block33 | 0021h | | |
| Block2 | 0002h | | | Block34 | 0022h | | |
| Block3 | 0003h | | | Block35 | 0023h | | |
| Block4 | 0004h | | | Block36 | 0024h | | |
| Block5 | 0005h | | | Block37 | 0025h | | |
| Block6 | 0006h | | | Block38 | 0026h | | |
| Block7 | 0007h | | | Block39 | 0027h | | |
| Block8 | 0008h | | | Block40 | 0028h | | |
| Block9 | 0009h | | | Block41 | 0029h | | |
| Block10 | 000Ah | | | Block42 | 002Ah | | |
| Block11 | 000Bh | | | Block43 | 002Bh | | |
| Block12 | 000Ch | | | Block44 | 002Ch | | |
| Block13 | 000Dh | | | Block45 | 002Dh | | |
| Block14 | 000Eh | SLC: 0000h~00FCh*, MLC: 0000h~01FCh* | SLC: 256KB, MLC: 512KB | Block46 | 002Eh | SLC: 0000h~00FCh*, MLC: 0000h~01FCh* | SLC: 256KB, MLC: 512KB |
| Block15 | 000Fh | | | Block47 | 002Fh | | |
| Block16 | 0010h | | | Block48 | 0030h | | |
| Block17 | 0011h | | | Block49 | 0031h | | |
| Block18 | 0012h | | | Block50 | 0032h | | |
| Block19 | 0013h | | | Block51 | 0033h | | |
| Block20 | 0014h | | | Block52 | 0034h | | |
| Block21 | 0015h | | | Block53 | 0035h | | |
| Block22 | 0016h | | | Block54 | 0036h | | |
| Block23 | 0017h | | | Block55 | 0037h | | |
| Block24 | 0018h | | | Block56 | 0038h | | |
| Block25 | 0019h | | | Block57 | 0039h | | |
| Block26 | 001Ah | | | Block58 | 003Ah | | |
| Block27 | 001Bh | | | Block59 | 003Bh | | |
| Block28 | 001Ch | | | Block60 | 003Ch | | |
| Block29 | 001Dh | | | Block61 | 003Dh | | |
| Block30 | 001Eh | | | Block62 | 003Eh | | |
| Block31 | 001Fh | | | Block63 | 003Fh | | |

* Only four sectors are addressable, see Start Address Register .

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block64 | 0040h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block96 | 0060h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block65 | 0041h | | | Block97 | 0061h | | |
| Block66 | 0042h | | | Block98 | 0062h | | |
| Block67 | 0043h | | | Block99 | 0063h | | |
| Block68 | 0044h | | | Block100 | 0064h | | |
| Block69 | 0045h | | | Block101 | 0065h | | |
| Block70 | 0046h | | | Block102 | 0066h | | |
| Block71 | 0047h | | | Block103 | 0067h | | |
| Block72 | 0048h | | | Block104 | 0068h | | |
| Block73 | 0049h | | | Block105 | 0069h | | |
| Block74 | 004Ah | | | Block106 | 006Ah | | |
| Block75 | 004Bh | | | Block107 | 006Bh | | |
| Block76 | 004Ch | | | Block108 | 006Ch | | |
| Block77 | 004Dh | | | Block109 | 006Dh | | |
| Block78 | 004Eh | | | Block110 | 006Eh | | |
| Block79 | 004Fh | | | Block111 | 006Fh | | |
| Block80 | 0050h | | | Block112 | 0070h | | |
| Block81 | 0051h | | | Block113 | 0071h | | |
| Block82 | 0052h | | | Block114 | 0072h | | |
| Block83 | 0053h | | | Block115 | 0073h | | |
| Block84 | 0054h | | | Block116 | 0074h | | |
| Block85 | 0055h | | | Block117 | 0075h | | |
| Block86 | 0056h | | | Block118 | 0076h | | |
| Block87 | 0057h | | | Block119 | 0077h | | |
| Block88 | 0058h | | | Block120 | 0078h | | |
| Block89 | 0059h | | | Block121 | 0079h | | |
| Block90 | 005Ah | | | Block122 | 007Ah | | |
| Block91 | 005Bh | | | Block123 | 007Bh | | |
| Block92 | 005Ch | | | Block124 | 007Ch | | |
| Block93 | 005Dh | | | Block125 | 007Dh | | |
| Block94 | 005Eh | | | Block126 | 007Eh | | |
| Block95 | 005Fh | | | Block127 | 007Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block128 | 0080h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block160 | 00A0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block129 | 0081h | | | Block161 | 00A1h | | |
| Block130 | 0082h | | | Block162 | 00A2h | | |
| Block131 | 0083h | | | Block163 | 00A3h | | |
| Block132 | 0084h | | | Block164 | 00A4h | | |
| Block133 | 0085h | | | Block165 | 00A5h | | |
| Block134 | 0086h | | | Block166 | 00A6h | | |
| Block135 | 0087h | | | Block167 | 00A7h | | |
| Block136 | 0088h | | | Block168 | 00A8h | | |
| Block137 | 0089h | | | Block169 | 00A9h | | |
| Block138 | 008Ah | | | Block170 | 00AAh | | |
| Block139 | 008Bh | | | Block171 | 00ABh | | |
| Block140 | 008Ch | | | Block172 | 00ACh | | |
| Block141 | 008Dh | | | Block173 | 00ADh | | |
| Block142 | 008Eh | | | Block174 | 00AEh | | |
| Block143 | 008Fh | | | Block175 | 00AFh | | |
| Block144 | 0090h | | | Block176 | 00B0h | | |
| Block145 | 0091h | | | Block177 | 00B1h | | |
| Block146 | 0092h | | | Block178 | 00B2h | | |
| Block147 | 0093h | | | Block179 | 00B3h | | |
| Block148 | 0094h | | | Block180 | 00B4h | | |
| Block149 | 0095h | | | Block181 | 00B5h | | |
| Block150 | 0096h | | | Block182 | 00B6h | | |
| Block151 | 0097h | | | Block183 | 00B7h | | |
| Block152 | 0098h | | | Block184 | 00B8h | | |
| Block153 | 0099h | | | Block185 | 00B9h | | |
| Block154 | 009Ah | | | Block186 | 00BAh | | |
| Block155 | 009Bh | | | Block187 | 00BBh | | |
| Block156 | 009Ch | | | Block188 | 00BCh | | |
| Block157 | 009Dh | | | Block189 | 00BDh | | |
| Block158 | 009Eh | | | Block190 | 00BEh | | |
| Block159 | 009Fh | | | Block191 | 00BFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block192 | 00C0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block224 | 00E0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block193 | 00C1h | | | Block225 | 00E1h | | |
| Block194 | 00C2h | | | Block226 | 00E2h | | |
| Block195 | 00C3h | | | Block227 | 00E3h | | |
| Block196 | 00C4h | | | Block228 | 00E4h | | |
| Block197 | 00C5h | | | Block229 | 00E5h | | |
| Block198 | 00C6h | | | Block230 | 00E6h | | |
| Block199 | 00C7h | | | Block231 | 00E7h | | |
| Block200 | 00C8h | | | Block232 | 00E8h | | |
| Block201 | 00C9h | | | Block233 | 00E9h | | |
| Block202 | 00CAh | | | Block234 | 00EAh | | |
| Block203 | 00CBh | | | Block235 | 00EBh | | |
| Block204 | 00CCh | | | Block236 | 00ECh | | |
| Block205 | 00CDh | | | Block237 | 00EDh | | |
| Block206 | 00CEh | | | Block238 | 00EEh | | |
| Block207 | 00CFh | | | Block239 | 00EFh | | |
| Block208 | 00D0h | | | Block240 | 00F0h | | |
| Block209 | 00D1h | | | Block241 | 00F1h | | |
| Block210 | 00D2h | | | Block242 | 00F2h | | |
| Block211 | 00D3h | | | Block243 | 00F3h | | |
| Block212 | 00D4h | | | Block244 | 00F4h | | |
| Block213 | 00D5h | | | Block245 | 00F5h | | |
| Block214 | 00D6h | | | Block246 | 00F6h | | |
| Block215 | 00D7h | | | Block247 | 00F7h | | |
| Block216 | 00D8h | | | Block248 | 00F8h | | |
| Block217 | 00D9h | | | Block249 | 00F9h | | |
| Block218 | 00DAh | | | Block250 | 00FAh | | |
| Block219 | 00DBh | | | Block251 | 00FBh | | |
| Block220 | 00DCh | | | Block252 | 00FCh | | |
| Block221 | 00DDh | | | Block253 | 00FDh | | |
| Block222 | 00DEh | | | Block254 | 00FEh | | |
| Block223 | 00DFh | | | Block255 | 00FFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block256 | 0100h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block288 | 0120h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block257 | 0101h | | | Block289 | 0121h | | |
| Block258 | 0102h | | | Block290 | 0122h | | |
| Block259 | 0103h | | | Block291 | 0123h | | |
| Block260 | 0104h | | | Block292 | 0124h | | |
| Block261 | 0105h | | | Block293 | 0125h | | |
| Block262 | 0106h | | | Block294 | 0126h | | |
| Block263 | 0107h | | | Block295 | 0127h | | |
| Block264 | 0108h | | | Block296 | 0128h | | |
| Block265 | 0109h | | | Block297 | 0129h | | |
| Block266 | 010Ah | | | Block298 | 012Ah | | |
| Block267 | 010Bh | | | Block299 | 012Bh | | |
| Block268 | 010Ch | | | Block300 | 012Ch | | |
| Block269 | 010Dh | | | Block301 | 012Dh | | |
| Block270 | 010Eh | | | Block302 | 012Eh | | |
| Block271 | 010Fh | | | Block303 | 012Fh | | |
| Block272 | 0110h | | | Block304 | 0130h | | |
| Block273 | 0111h | | | Block305 | 0131h | | |
| Block274 | 0112h | | | Block306 | 0132h | | |
| Block275 | 0113h | | | Block307 | 0133h | | |
| Block276 | 0114h | | | Block308 | 0134h | | |
| Block277 | 0115h | | | Block309 | 0135h | | |
| Block278 | 0116h | | | Block310 | 0136h | | |
| Block279 | 0117h | | | Block311 | 0137h | | |
| Block280 | 0118h | | | Block312 | 0138h | | |
| Block281 | 0119h | | | Block313 | 0139h | | |
| Block282 | 011Ah | | | Block314 | 013Ah | | |
| Block283 | 011Bh | | | Block315 | 013Bh | | |
| Block284 | 011Ch | | | Block316 | 013Ch | | |
| Block285 | 011Dh | | | Block317 | 013Dh | | |
| Block286 | 011Eh | | | Block318 | 013Eh | | |
| Block287 | 011Fh | | | Block319 | 013Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block320 | 0140h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block352 | 0160h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block321 | 0141h | | | Block353 | 0161h | | |
| Block322 | 0142h | | | Block354 | 0162h | | |
| Block323 | 0143h | | | Block355 | 0163h | | |
| Block324 | 0144h | | | Block356 | 0164h | | |
| Block325 | 0145h | | | Block357 | 0165h | | |
| Block326 | 0146h | | | Block358 | 0166h | | |
| Block327 | 0147h | | | Block359 | 0167h | | |
| Block328 | 0148h | | | Block360 | 0168h | | |
| Block329 | 0149h | | | Block361 | 0169h | | |
| Block330 | 014Ah | | | Block362 | 016Ah | | |
| Block331 | 014Bh | | | Block363 | 016Bh | | |
| Block332 | 014Ch | | | Block364 | 016Ch | | |
| Block333 | 014Dh | | | Block365 | 016Dh | | |
| Block334 | 014Eh | | | Block366 | 016Eh | | |
| Block335 | 014Fh | | | Block367 | 016Fh | | |
| Block336 | 0150h | | | Block368 | 0170h | | |
| Block337 | 0151h | | | Block369 | 0171h | | |
| Block338 | 0152h | | | Block370 | 0172h | | |
| Block339 | 0153h | | | Block371 | 0173h | | |
| Block340 | 0154h | | | Block372 | 0174h | | |
| Block341 | 0155h | | | Block373 | 0175h | | |
| Block342 | 0156h | | | Block374 | 0176h | | |
| Block343 | 0157h | | | Block375 | 0177h | | |
| Block344 | 0158h | | | Block376 | 0178h | | |
| Block345 | 0159h | | | Block377 | 0179h | | |
| Block346 | 015Ah | | | Block378 | 017Ah | | |
| Block347 | 015Bh | | | Block379 | 017Bh | | |
| Block348 | 015Ch | | | Block380 | 017Ch | | |
| Block349 | 015Dh | | | Block381 | 017Dh | | |
| Block350 | 015Eh | | | Block382 | 017Eh | | |
| Block351 | 015Fh | | | Block383 | 017Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block384 | 0180h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block416 | 01A0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block385 | 0181h | | | Block417 | 01A1h | | |
| Block386 | 0182h | | | Block418 | 01A2h | | |
| Block387 | 0183h | | | Block419 | 01A3h | | |
| Block388 | 0184h | | | Block420 | 01A4h | | |
| Block389 | 0185h | | | Block421 | 01A5h | | |
| Block390 | 0186h | | | Block422 | 01A6h | | |
| Block391 | 0187h | | | Block423 | 01A7h | | |
| Block392 | 0188h | | | Block424 | 01A8h | | |
| Block393 | 0189h | | | Block425 | 01A9h | | |
| Block394 | 018Ah | | | Block426 | 01AAh | | |
| Block395 | 018Bh | | | Block427 | 01ABh | | |
| Block396 | 018Ch | | | Block428 | 01ACh | | |
| Block397 | 018Dh | | | Block429 | 01ADh | | |
| Block398 | 018Eh | | | Block430 | 01AEh | | |
| Block399 | 018Fh | | | Block431 | 01AFh | | |
| Block400 | 0190h | | | Block432 | 01B0h | | |
| Block401 | 0191h | | | Block433 | 01B1h | | |
| Block402 | 0192h | | | Block434 | 01B2h | | |
| Block403 | 0193h | | | Block435 | 01B3h | | |
| Block404 | 0194h | | | Block436 | 01B4h | | |
| Block405 | 0195h | | | Block437 | 01B5h | | |
| Block406 | 0196h | | | Block438 | 01B6h | | |
| Block407 | 0197h | | | Block439 | 01B7h | | |
| Block408 | 0198h | | | Block440 | 01B8h | | |
| Block409 | 0199h | | | Block441 | 01B9h | | |
| Block410 | 019Ah | | | Block442 | 01BAh | | |
| Block411 | 019Bh | | | Block443 | 01BBh | | |
| Block412 | 019Ch | | | Block444 | 01BCh | | |
| Block413 | 019Dh | | | Block445 | 01BDh | | |
| Block414 | 019Eh | | | Block446 | 01BEh | | |
| Block415 | 019Fh | | | Block447 | 01BFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block448 | 01C0h | | | Block480 | 01E0h | | |
| Block449 | 01C1h | | | Block481 | 01E1h | | |
| Block450 | 01C2h | | | Block482 | 01E2h | | |
| Block451 | 01C3h | | | Block483 | 01E3h | | |
| Block452 | 01C4h | | | Block484 | 01E4h | | |
| Block453 | 01C5h | | | Block485 | 01E5h | | |
| Block454 | 01C6h | | | Block486 | 01E6h | | |
| Block455 | 01C7h | | | Block487 | 01E7h | | |
| Block456 | 01C8h | | | Block488 | 01E8h | | |
| Block457 | 01C9h | | | Block489 | 01E9h | | |
| Block458 | 01CAh | | | Block490 | 01EAh | | |
| Block459 | 01CBh | | | Block491 | 01EBh | | |
| Block460 | 01CCh | | | Block492 | 01ECh | | |
| Block461 | 01CDh | | | Block493 | 01EDh | | |
| Block462 | 01CEh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block494 | 01EEh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block463 | 01CFh | | | Block495 | 01EFh | | |
| Block464 | 01D0h | | | Block496 | 01F0h | | |
| Block465 | 01D1h | | | Block497 | 01F1h | | |
| Block466 | 01D2h | | | Block498 | 01F2h | | |
| Block467 | 01D3h | | | Block499 | 01F3h | | |
| Block468 | 01D4h | | | Block500 | 01F4h | | |
| Block469 | 01D5h | | | Block501 | 01F5h | | |
| Block470 | 01D6h | | | Block502 | 01F6h | | |
| Block471 | 01D7h | | | Block503 | 01F7h | | |
| Block472 | 01D8h | | | Block504 | 01F8h | | |
| Block473 | 01D9h | | | Block505 | 01F9h | | |
| Block474 | 01DAh | | | Block506 | 01FAh | | |
| Block475 | 01DBh | | | Block507 | 01FBh | | |
| Block476 | 01DCh | | | Block508 | 01FCh | | |
| Block477 | 01DDh | | | Block509 | 01FDh | | |
| Block478 | 01DEh | | | Block510 | 01FEh | | |
| Block479 | 01DFh | | | Block511 | 01FFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block512 | 0200h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block544 | 0220h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block513 | 0201h | | | Block545 | 0221h | | |
| Block514 | 0202h | | | Block546 | 0222h | | |
| Block515 | 0203h | | | Block547 | 0223h | | |
| Block516 | 0204h | | | Block548 | 0224h | | |
| Block517 | 0205h | | | Block549 | 0225h | | |
| Block518 | 0206h | | | Block550 | 0226h | | |
| Block519 | 0207h | | | Block551 | 0227h | | |
| Block520 | 0208h | | | Block552 | 0228h | | |
| Block521 | 0209h | | | Block553 | 0229h | | |
| Block522 | 020Ah | | | Block554 | 022Ah | | |
| Block523 | 020Bh | | | Block555 | 022Bh | | |
| Block524 | 020Ch | | | Block556 | 022Ch | | |
| Block525 | 020Dh | | | Block557 | 022Dh | | |
| Block526 | 020Eh | | | Block558 | 022Eh | | |
| Block527 | 020Fh | | | Block559 | 022Fh | | |
| Block528 | 0210h | | | Block560 | 0230h | | |
| Block529 | 0211h | | | Block561 | 0231h | | |
| Block530 | 0212h | | | Block562 | 0232h | | |
| Block531 | 0213h | | | Block563 | 0233h | | |
| Block532 | 0214h | | | Block564 | 0234h | | |
| Block533 | 0215h | | | Block565 | 0235h | | |
| Block534 | 0216h | | | Block566 | 0236h | | |
| Block535 | 0217h | | | Block567 | 0237h | | |
| Block536 | 0218h | | | Block568 | 0238h | | |
| Block537 | 0219h | | | Block569 | 0239h | | |
| Block538 | 021Ah | | | Block570 | 023Ah | | |
| Block539 | 021Bh | | | Block571 | 023Bh | | |
| Block540 | 021Ch | | | Block572 | 023Ch | | |
| Block541 | 021Dh | | | Block573 | 023Dh | | |
| Block542 | 021Eh | | | Block574 | 023Eh | | |
| Block543 | 021Fh | | | Block575 | 023Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block576 | 0240h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block608 | 0260h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block577 | 0241h | | | Block609 | 0261h | | |
| Block578 | 0242h | | | Block610 | 0262h | | |
| Block579 | 0243h | | | Block611 | 0263h | | |
| Block580 | 0244h | | | Block612 | 0264h | | |
| Block581 | 0245h | | | Block613 | 0265h | | |
| Block582 | 0246h | | | Block614 | 0266h | | |
| Block583 | 0247h | | | Block615 | 0267h | | |
| Block584 | 0248h | | | Block616 | 0268h | | |
| Block585 | 0249h | | | Block617 | 0269h | | |
| Block586 | 024Ah | | | Block618 | 026Ah | | |
| Block587 | 024Bh | | | Block619 | 026Bh | | |
| Block588 | 024Ch | | | Block620 | 026Ch | | |
| Block589 | 024Dh | | | Block621 | 026Dh | | |
| Block590 | 024Eh | | | Block622 | 026Eh | | |
| Block591 | 024Fh | | | Block623 | 026Fh | | |
| Block592 | 0250h | | | Block624 | 0270h | | |
| Block593 | 0251h | | | Block625 | 0271h | | |
| Block594 | 0252h | | | Block626 | 0272h | | |
| Block595 | 0253h | | | Block627 | 0273h | | |
| Block596 | 0254h | | | Block628 | 0274h | | |
| Block597 | 0255h | | | Block629 | 0275h | | |
| Block598 | 0256h | | | Block630 | 0276h | | |
| Block599 | 0257h | | | Block631 | 0277h | | |
| Block600 | 0258h | | | Block632 | 0278h | | |
| Block601 | 0259h | | | Block633 | 0279h | | |
| Block602 | 025Ah | | | Block634 | 027Ah | | |
| Block603 | 025Bh | | | Block635 | 027Bh | | |
| Block604 | 025Ch | | | Block636 | 027Ch | | |
| Block605 | 025Dh | | | Block637 | 027Dh | | |
| Block606 | 025Eh | | | Block638 | 027Eh | | |
| Block607 | 025Fh | | | Block639 | 027Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block640 | 0280h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block672 | 02A0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block641 | 0281h | | | Block673 | 02A1h | | |
| Block642 | 0282h | | | Block674 | 02A2h | | |
| Block643 | 0283h | | | Block675 | 02A3h | | |
| Block644 | 0284h | | | Block676 | 02A4h | | |
| Block645 | 0285h | | | Block677 | 02A5h | | |
| Block646 | 0286h | | | Block678 | 02A6h | | |
| Block647 | 0287h | | | Block679 | 02A7h | | |
| Block648 | 0288h | | | Block680 | 02A8h | | |
| Block649 | 0289h | | | Block681 | 02A9h | | |
| Block650 | 028Ah | | | Block682 | 02AAh | | |
| Block651 | 028Bh | | | Block683 | 02ABh | | |
| Block652 | 028Ch | | | Block684 | 02ACh | | |
| Block653 | 028Dh | | | Block685 | 02ADh | | |
| Block654 | 028Eh | | | Block686 | 02AEh | | |
| Block655 | 028Fh | | | Block687 | 02AFh | | |
| Block656 | 0290h | | | Block688 | 02B0h | | |
| Block657 | 0291h | | | Block689 | 02B1h | | |
| Block658 | 0292h | | | Block690 | 02B2h | | |
| Block659 | 0293h | | | Block691 | 02B3h | | |
| Block660 | 0294h | | | Block692 | 02B4h | | |
| Block661 | 0295h | | | Block693 | 02B5h | | |
| Block662 | 0296h | | | Block694 | 02B6h | | |
| Block663 | 0297h | | | Block695 | 02B7h | | |
| Block664 | 0298h | | | Block696 | 02B8h | | |
| Block665 | 0299h | | | Block697 | 02B9h | | |
| Block666 | 029Ah | | | Block698 | 02BAh | | |
| Block667 | 029Bh | | | Block699 | 02BBh | | |
| Block668 | 029Ch | | | Block700 | 02BCh | | |
| Block669 | 029Dh | | | Block701 | 02BDh | | |
| Block670 | 029Eh | | | Block702 | 02BEh | | |
| Block671 | 029Fh | | | Block703 | 02BFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block704 | 02C0h | | | Block736 | 02E0h | | |
| Block705 | 02C1h | | | Block737 | 02E1h | | |
| Block706 | 02C2h | | | Block738 | 02E2h | | |
| Block707 | 02C3h | | | Block739 | 02E3h | | |
| Block708 | 02C4h | | | Block740 | 02E4h | | |
| Block709 | 02C5h | | | Block741 | 02E5h | | |
| Block710 | 02C6h | | | Block742 | 02E6h | | |
| Block711 | 02C7h | | | Block743 | 02E7h | | |
| Block712 | 02C8h | | | Block744 | 02E8h | | |
| Block713 | 02C9h | | | Block745 | 02E9h | | |
| Block714 | 02CAh | | | Block746 | 02EAh | | |
| Block715 | 02CBh | | | Block747 | 02EBh | | |
| Block716 | 02CCh | | | Block748 | 02ECh | | |
| Block717 | 02CDh | | | Block749 | 02EDh | | |
| Block718 | 02CEh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block750 | 02EEh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block719 | 02CFh | | | Block751 | 02EFh | | |
| Block720 | 02D0h | | | Block752 | 02F0h | | |
| Block721 | 02D1h | | | Block753 | 02F1h | | |
| Block722 | 02D2h | | | Block754 | 02F2h | | |
| Block723 | 02D3h | | | Block755 | 02F3h | | |
| Block724 | 02D4h | | | Block756 | 02F4h | | |
| Block725 | 02D5h | | | Block757 | 02F5h | | |
| Block726 | 02D6h | | | Block758 | 02F6h | | |
| Block727 | 02D7h | | | Block759 | 02F7h | | |
| Block728 | 02D8h | | | Block760 | 02F8h | | |
| Block729 | 02D9h | | | Block761 | 02F9h | | |
| Block730 | 02DAh | | | Block762 | 02FAh | | |
| Block731 | 02DBh | | | Block763 | 02FBh | | |
| Block732 | 02DCh | | | Block764 | 02FCh | | |
| Block733 | 02DDh | | | Block765 | 02FDh | | |
| Block734 | 02DEh | | | Block766 | 02FEh | | |
| Block735 | 02DFh | | | Block767 | 02FFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block768 | 0300h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block800 | 0320h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block769 | 0301h | | | Block801 | 0321h | | |
| Block770 | 0302h | | | Block802 | 0322h | | |
| Block771 | 0303h | | | Block803 | 0323h | | |
| Block772 | 0304h | | | Block804 | 0324h | | |
| Block773 | 0305h | | | Block805 | 0325h | | |
| Block774 | 0306h | | | Block806 | 0326h | | |
| Block775 | 0307h | | | Block807 | 0327h | | |
| Block776 | 0308h | | | Block808 | 0328h | | |
| Block777 | 0309h | | | Block809 | 0329h | | |
| Block778 | 030Ah | | | Block810 | 032Ah | | |
| Block779 | 030Bh | | | Block811 | 032Bh | | |
| Block780 | 030Ch | | | Block812 | 032Ch | | |
| Block781 | 030Dh | | | Block813 | 032Dh | | |
| Block782 | 030Eh | | | Block814 | 032Eh | | |
| Block783 | 030Fh | | | Block815 | 032Fh | | |
| Block784 | 0310h | | | Block816 | 0330h | | |
| Block785 | 0311h | | | Block817 | 0331h | | |
| Block786 | 0312h | | | Block818 | 0332h | | |
| Block787 | 0313h | | | Block819 | 0333h | | |
| Block788 | 0314h | | | Block820 | 0334h | | |
| Block789 | 0315h | | | Block821 | 0335h | | |
| Block790 | 0316h | | | Block822 | 0336h | | |
| Block791 | 0317h | | | Block823 | 0337h | | |
| Block792 | 0318h | | | Block824 | 0338h | | |
| Block793 | 0319h | | | Block825 | 0339h | | |
| Block794 | 031Ah | | | Block826 | 033Ah | | |
| Block795 | 031Bh | | | Block827 | 033Bh | | |
| Block796 | 031Ch | | | Block828 | 033Ch | | |
| Block797 | 031Dh | | | Block829 | 033Dh | | |
| Block798 | 031Eh | | | Block830 | 033Eh | | |
| Block799 | 031Fh | | | Block831 | 033Fh | | |

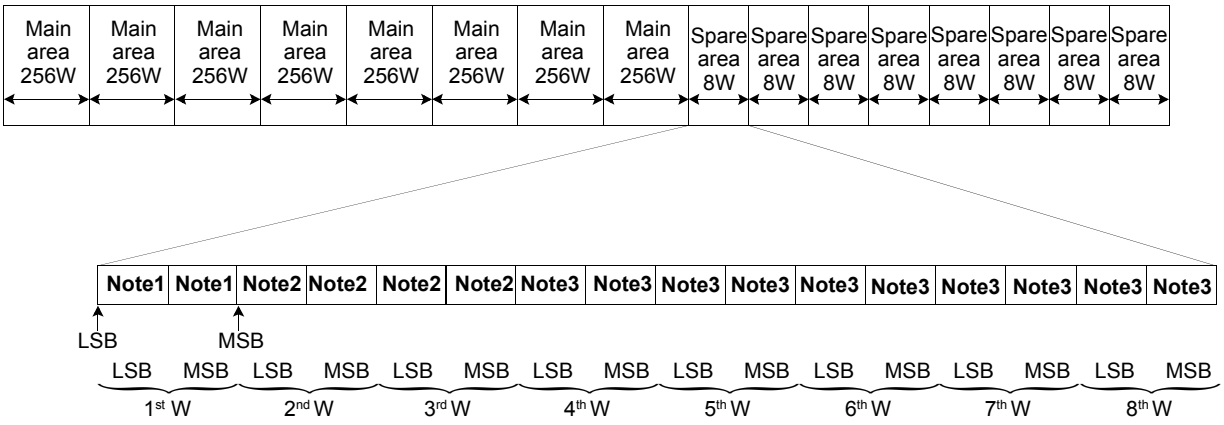
| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block832 | 0340h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block864 | 0360h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block833 | 0341h | | | Block865 | 0361h | | |
| Block834 | 0342h | | | Block866 | 0362h | | |
| Block835 | 0343h | | | Block867 | 0363h | | |
| Block836 | 0344h | | | Block868 | 0364h | | |
| Block837 | 0345h | | | Block869 | 0365h | | |
| Block838 | 0346h | | | Block870 | 0366h | | |
| Block839 | 0347h | | | Block871 | 0367h | | |
| Block840 | 0348h | | | Block872 | 0368h | | |
| Block841 | 0349h | | | Block873 | 0369h | | |
| Block842 | 034Ah | | | Block874 | 036Ah | | |
| Block843 | 034Bh | | | Block875 | 036Bh | | |
| Block844 | 034Ch | | | Block876 | 036Ch | | |
| Block845 | 034Dh | | | Block877 | 036Dh | | |
| Block846 | 034Eh | | | Block878 | 036Eh | | |
| Block847 | 034Fh | | | Block879 | 036Fh | | |
| Block848 | 0350h | | | Block880 | 0370h | | |
| Block849 | 0351h | | | Block881 | 0371h | | |
| Block850 | 0352h | | | Block882 | 0372h | | |
| Block851 | 0353h | | | Block883 | 0373h | | |
| Block852 | 0354h | | | Block884 | 0374h | | |
| Block853 | 0355h | | | Block885 | 0375h | | |
| Block854 | 0356h | | | Block886 | 0376h | | |
| Block855 | 0357h | | | Block887 | 0377h | | |
| Block856 | 0358h | | | Block888 | 0378h | | |
| Block857 | 0359h | | | Block889 | 0379h | | |
| Block858 | 035Ah | | | Block890 | 037Ah | | |
| Block859 | 035Bh | | | Block891 | 037Bh | | |
| Block860 | 035Ch | | | Block892 | 037Ch | | |
| Block861 | 035Dh | | | Block893 | 037Dh | | |
| Block862 | 035Eh | | | Block894 | 037Eh | | |
| Block863 | 035Fh | | | Block895 | 037Fh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block896 | 0380h | | | Block928 | 03A0h | | |
| Block897 | 0381h | | | Block929 | 03A1h | | |
| Block898 | 0382h | | | Block930 | 03A2h | | |
| Block899 | 0383h | | | Block931 | 03A3h | | |
| Block900 | 0384h | | | Block932 | 03A4h | | |
| Block901 | 0385h | | | Block933 | 03A5h | | |
| Block902 | 0386h | | | Block934 | 03A6h | | |
| Block903 | 0387h | | | Block935 | 03A7h | | |
| Block904 | 0388h | | | Block936 | 03A8h | | |
| Block905 | 0389h | | | Block937 | 03A9h | | |
| Block906 | 038Ah | | | Block938 | 03AAh | | |
| Block907 | 038Bh | | | Block939 | 03ABh | | |
| Block908 | 038Ch | | | Block940 | 03ACh | | |
| Block909 | 038Dh | | | Block941 | 03ADh | | |
| Block910 | 038Eh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block942 | 03AEh | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block911 | 038Fh | | | Block943 | 03AFh | | |
| Block912 | 0390h | | | Block944 | 03B0h | | |
| Block913 | 0391h | | | Block945 | 03B1h | | |
| Block914 | 0392h | | | Block946 | 03B2h | | |
| Block915 | 0393h | | | Block947 | 03B3h | | |
| Block916 | 0394h | | | Block948 | 03B4h | | |
| Block917 | 0395h | | | Block949 | 03B5h | | |
| Block918 | 0396h | | | Block950 | 03B6h | | |
| Block919 | 0397h | | | Block951 | 03B7h | | |
| Block920 | 0398h | | | Block952 | 03B8h | | |
| Block921 | 0399h | | | Block953 | 03B9h | | |
| Block922 | 039Ah | | | Block954 | 03BAh | | |
| Block923 | 039Bh | | | Block955 | 03BBh | | |
| Block924 | 039Ch | | | Block956 | 03BCh | | |
| Block925 | 039Dh | | | Block957 | 03BDh | | |
| Block926 | 039Eh | | | Block958 | 03BEh | | |
| Block927 | 039Fh | | | Block959 | 03BFh | | |

| Block | Block Address [F100h] | Page Address [F107h] | Size | Block | Block Address [F100h] | Page Address [F107h] | Size |
|---|---|---|---|---|---|---|---|
| Block960 | 03C0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB | Block992 | 03E0h | SLC: 0000h~00FCh, MLC: 0000h~01FCh | SLC: 256KB, MLC: 512KB |
| Block961 | 03C1h | | | Block993 | 03E1h | | |
| Block962 | 03C2h | | | Block994 | 03E2h | | |
| Block963 | 03C3h | | | Block995 | 03E3h | | |
| Block964 | 03C4h | | | Block996 | 03E4h | | |
| Block965 | 03C5h | | | Block997 | 03E5h | | |
| Block966 | 03C6h | | | Block998 | 03E6h | | |
| Block967 | 03C7h | | | Block999 | 03E7h | | |
| Block968 | 03C8h | | | Block1000 | 03E8h | | |
| Block969 | 03C9h | | | Block1001 | 03E9h | | |
| Block970 | 03CAh | | | Block1002 | 03EAh | | |
| Block971 | 03CBh | | | Block1003 | 03EBh | | |
| Block972 | 03CCh | | | Block1004 | 03ECh | | |
| Block973 | 03CDh | | | Block1005 | 03EDh | | |
| Block974 | 03CEh | | | Block1006 | 03EEh | | |
| Block975 | 03CFh | | | Block1007 | 03EFh | | |
| Block976 | 03D0h | | | Block1008 | 03F0h | | |
| Block977 | 03D1h | | | Block1009 | 03F1h | | |
| Block978 | 03D2h | | | Block1010 | 03F2h | | |
| Block979 | 03D3h | | | Block1011 | 03F3h | | |
| Block980 | 03D4h | | | Block1012 | 03F4h | | |
| Block981 | 03D5h | | | Block1013 | 03F5h | | |
| Block982 | 03D6h | | | Block1014 | 03F6h | | |
| Block983 | 03D7h | | | Block1015 | 03F7h | | |
| Block984 | 03D8h | | | Block1016 | 03F8h | | |
| Block985 | 03D9h | | | Block1017 | 03F9h | | |
| Block986 | 03DAh | | | Block1018 | 03FAh | | |
| Block987 | 03DBh | | | Block1019 | 03FBh | | |
| Block988 | 03DCh | | | Block1020 | 03FCh | | |
| Block989 | 03DDh | | | Block1021 | 03FDh | | |
| Block990 | 03DEh | | | Block1022 | 03FEh | | |
| Block991 | 03DFh | | | Block1023 | 03FFh | | |

### 2.7.2 Internal Memory Spare Area Assignment

The figure below shows the assignment of the spare area in the Internal Memory NAND Array.




**Spare Area Assignment in the Internal Memory NAND Array Information**

| Word | Byte | Note | Description |
|---|---|---|---|
| 1 | LSB | 1 | Invalid Block information in 1st and 2nd page of an invalid block |
| | MSB | | |
| 2 | LSB | 2 | Managed by internal ECC logic for Logical Sector Number area |
| | MSB | | |
| 3 | LSB | | |
| | MSB | | |
| 4 | LSB | 3 | 4bit ECC parity values |
| | MSB | | |
| 5 | LSB | | |
| | MSB | | |
| 6 | LSB | | |
| | MSB | | |
| 7 | LSB | | |
| | MSB | | |
| 8 | LSB | | |
| | MSB | | |

### 2.7.3 External Memory (BufferRAM) Address Map

The following table shows the External Memory address map in Word and Byte Order.
Note that the data output is unknown while host reads a register bit of reserved area and dual buffering is not applicable.

| Division | Address (word order) | Address (byte order) | Size (total 128KB) | | | Usage | Description |
|---|---|---|---|---|---|---|---|
| Main area (64KB) | 0000h~00FFh | 00000h~001FEh | 512B | 1KB | R | BootM 0 | BootRAM, Main, block0/page0/sector0 |
| | 0100h~01FFh | 00200h~003FEh | 512B | | | BootM 1 | BootRAM, Main, block0/page0/sector1 |
| | 0200h~02FFh | 00400h~005FEh | 512B | 4KB | R/W | DataM 0_0 | DataRAM, Main, nth page/sector0 |
| | 0300h~03FFh | 00600h~007FEh | 512B | | | DataM 0_1 | DataRAM, Main, nth page/sector1 |
| | 0400h~04FFh | 00800h~009FEh | 512B | | | DataM 0_2 | DataRAM, Main, nth page/sector2 |
| | 0500h~05FFh | 00A00h~00BFEh | 512B | | | DataM 0_3 | DataRAM, Main, nth page/sector3 |
| | 0600h~06FFh | 00C00h~00DFEh | 512B | | | DataM 1_0 | DataRAM, Main, nth page/sector4 |
| | 0700h~07FFh | 00E00h~00FFEh | 512B | | | DataM 1_1 | DataRAM, Main, nth page/sector5 |
| | 0800h~08FFh | 01000h~011FEh | 512B | | | DataM 1_2 | DataRAM, Main, nth page/sector6 |
| | 0900h~09FFh | 01200h~013FEh | 512B | | | DataM 1_3 | DataRAM, Main, nth page/sector7 |
| | 0A00h~7FFFh | 01400h~0FFFEh | 59K | 59K | - | Reserved | Reserved |
| Spare area (8KB) | 8000h~8007h | 10000h~1000Eh | 16B | 32B | R | BootS 0 | BootRAM, Spare, block0/page0/sector0 |
| | 8008h~800Fh | 10010h~1001Eh | 16B | | | BootS 1 | BootRAM, Spare, block0/page0/sector1 |
| | 8010h~8017h | 10020h~1002Eh | 16B | 128B | R/W | DataS 0_0 | DataRAM, Spare, nth page/sector0 |
| | 8018h~801Fh | 10030h~1003Eh | 16B | | | DataS 0_1 | DataRAM, Spare, nth page/sector1 |
| | 8020h~8027h | 10040h~1004Eh | 16B | | | DataS 0_2 | DataRAM, Spare, nth page/sector2 |
| | 8028h~802Fh | 10050h~1005Eh | 16B | | | DataS 0_3 | DataRAM, Spare, nth page/sector3 |
| | 8030h~8037h | 10060h~1006Eh | 16B | | | DataS 1_0 | DataRAM, Spare, nth page/sector4 |
| | 8038h~803Fh | 10070h~1007Eh | 16B | | | DataS 1_1 | DataRAM, Spare, nth page/sector5 |
| | 8040h~8047h | 10080h~1008Eh | 16B | | | DataS 1_2 | DataRAM, Spare, nth page/sector6 |
| | 8048h~804Fh | 10090h~1009Eh | 16B | | | DataS 1_3 | DataRAM, Spare, nth page/sector7 |
| | 8050h~8FFFh | 100A0h~11FFEh | 8032B | 8032B | - | Reserved | Reserved |
| Reserved (24KB) | 9000h~BFFFh | 12000h~17FFEh | 24KB | 24KB | - | Reserved | Reserved |
| Reserved (8KB) | C000h~CFFFh | 18000h~19FFEh | 8KB | 8KB | - | Reserved | Reserved |
| Reserved (16KB) | D000h~EFFFh | 1A000h~1DFFEh | 16KB | 16KB | - | Reserved | Reserved |
| Registers (8KB) | F000h~FFFFh | 1E000h~1FFFEh | 8KB | 8KB | R or R/W | Registers | Registers |

### 2.7.4 External Memory Map Detail Information

The tables below show Word Order Address Map information for the BootRAM and DataRAM main and spare areas.

• **BootRAM(Main area)**

-0000h~01FFh: 2(sector) x 512byte(NAND main area) = 1KB

| | |
|---|---|
| 0000h~00FFh(512B)<br>BootM 0<br>(sector 0 of page 0/block 0) | 0100h~01FFh(512B)<br>BootM 1<br>(sector 1 of page 0/block 0) |

• **DataRAM(Main area)**

-0200h~09FFh: 8(sector) x 512byte(NAND main area) = 4KB

| | | | |
|---|---|---|---|
| 0200h~02FFh(512B)<br>DataM 0_0<br>(sector 0 of nth page) | 0300h~03FFh(512B)<br>DataM 0_1<br>(sector 1 of nth page) | 0400h~04FFh(512B)<br>DataM 0_2<br>(sector 2 of nth page) | 0500h~05FFh(512B)<br>DataM 0_3<br>(sector 3 of nth page) |
| 0600h~06FFh(512B)<br>DataM 1_0<br>(sector 4 of nth page) | 0700h~07FFh(512B)<br>DataM 1_1<br>(sector 5 of nth page) | 0800h~08FFh(512B)<br>DataM 1_2<br>(sector 6 of nth page) | 0900h~09FFh(512B)<br>DataM 1_3<br>(sector 7 of nth page) |

• **BootRAM(Spare area)**

-8000h~800Fh: 2(sector) x 16byte(NAND spare area) = 32B

| | |
|---|---|
| 8000h~8007h(16B)<br>BootS 0<br>(sector 0 of page 0/block 0) | 8008h~800Fh(16B)<br>BootS 1<br>(sector 1 of page 0/block 0) |

• **DataRAM(Spare area)**

-8010h~804Fh: 8(sector) x 16byte(NAND spare area) = 128B

| | | | |
|---|---|---|---|
| 8010h~8017h(16B)<br>DataS 0_0<br>(sector 0 of nth page) | 8018h~801Fh(16B)<br>DataS 0_1<br>(sector 1 of nth page) | 8020h~8027h(16B)<br>DataS 0_2<br>(sector 2 of nth page) | 8028h~802Fh(16B)<br>DataS 0_3<br>(sector 3 of nth page) |
| 8030h~8037h(16B)<br>DataS 1_0<br>(sector 4 of nth page) | 8038h~803Fh(16B)<br>DataS 1_1<br>(sector 5 of nth page) | 8040h~8047h(16B)<br>DataS 1_2<br>(sector 6 of nth page) | 8048h~804Fh(16B)<br>DataS 1_3<br>(sector 7 of nth page) |

*NAND Flash array consists of 4KB page size and 256KB(SLC)/512KB(MLC) block size.

### 2.7.5 External Memory Spare Area Assignment

Equivalent to 1word of NAND Flash

| Buf. | Word Address | Byte Address | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BootS 0 | 8000h | 10000h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8001h | 10002h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 8002h | 10004h | | | | | | | | | | | | | | | | |
| | 8003h | 10006h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 8004h | 10008h | | | | | | | | | | | | | | | | |
| | 8005h | 1000Ah | | | | | | | | | | | | | | | | |
| | 8006h | 1000Ch | | | | | | | | | | | | | | | | |
| | 8007h | 1000Eh | | | | | | | | | | | | | | | | |
| BootS 1 | 8008h | 10010h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8009h | 10012h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 800Ah | 10014h | | | | | | | | | | | | | | | | |
| | 800Bh | 10016h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 800Ch | 10018h | | | | | | | | | | | | | | | | |
| | 800Dh | 1001Ah | | | | | | | | | | | | | | | | |
| | 800Eh | 1001Ch | | | | | | | | | | | | | | | | |
| | 800Fh | 1001Eh | | | | | | | | | | | | | | | | |
| DataS 0_0 | 8010h | 10020h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8011h | 10022h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 8012h | 10024h | | | | | | | | | | | | | | | | |
| | 8013h | 10026h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 8014h | 10028h | | | | | | | | | | | | | | | | |
| | 8015h | 1002Ah | | | | | | | | | | | | | | | | |
| | 8016h | 1002Ch | | | | | | | | | | | | | | | | |
| | 8017h | 1002Eh | | | | | | | | | | | | | | | | |
| DataS 0_1 | 8018h | 10030h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8019h | 10032h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 801Ah | 10034h | | | | | | | | | | | | | | | | |
| | 801Bh | 10036h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 801Ch | 10038h | | | | | | | | | | | | | | | | |
| | 801Dh | 1003Ah | | | | | | | | | | | | | | | | |
| | 801Eh | 1003Ch | | | | | | | | | | | | | | | | |
| | 801Fh | 1003Eh | | | | | | | | | | | | | | | | |

| Buf. | Word Address | Byte Address | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DataS 0_2 | 8020h | 10040h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8021h | 10042h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 8022h | 10044h | | | | | | | | | | | | | | | | |
| | 8023h | 10046h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 8024h | 10048h | | | | | | | | | | | | | | | | |
| | 8025h | 1004Ah | | | | | | | | | | | | | | | | |
| | 8026h | 1004Ch | | | | | | | | | | | | | | | | |
| | 8027h | 1004Eh | | | | | | | | | | | | | | | | |
| DataS 0_3 | 8028h | 10050h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8029h | 10052h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 802Ah | 10054h | | | | | | | | | | | | | | | | |
| | 802Bh | 10056h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 802Ch | 10058h | | | | | | | | | | | | | | | | |
| | 802Dh | 1005Ah | | | | | | | | | | | | | | | | |
| | 802Eh | 1005Ch | | | | | | | | | | | | | | | | |
| | 802Fh | 1005Eh | | | | | | | | | | | | | | | | |
| DataS 1_0 | 8030h | 10060h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8031h | 10062h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 8032h | 10064h | | | | | | | | | | | | | | | | |
| | 8033h | 10066h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 8034h | 10068h | | | | | | | | | | | | | | | | |
| | 8035h | 1006Ah | | | | | | | | | | | | | | | | |
| | 8036h | 1006Ch | | | | | | | | | | | | | | | | |
| | 8037h | 1006Eh | | | | | | | | | | | | | | | | |
| DataS 1_1 | 8038h | 10070h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8039h | 10072h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 803Ah | 10074h | | | | | | | | | | | | | | | | |
| | 803Bh | 10076h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 803Ch | 10078h | | | | | | | | | | | | | | | | |
| | 803Dh | 1007Ah | | | | | | | | | | | | | | | | |
| | 803Eh | 1007Ch | | | | | | | | | | | | | | | | |
| | 803Fh | 1007Eh | | | | | | | | | | | | | | | | |
| DataS 1_2 | 8040h | 10080h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8041h | 10082h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 8042h | 10084h | | | | | | | | | | | | | | | | |
| | 8043h | 10086h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 8044h | 10088h | | | | | | | | | | | | | | | | |
| | 8045h | 1008Ah | | | | | | | | | | | | | | | | |
| | 8046h | 1008Ch | | | | | | | | | | | | | | | | |
| | 8047h | 1008Eh | | | | | | | | | | | | | | | | |

Equivalent to 1word of NAND Flash

| Buf. | Word Address | Byte Address | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DataS 1_3 | 8048h | 10090h | BI(Bad block Information ) | | | | | | | | | | | | | | | |
| | 8049h | 10092h | Managed by internal ECC logic | | | | | | | | | | | | | | | |
| | 804Ah | 10094h | | | | | | | | | | | | | | | | |
| | 804Bh | 10096h | 4bit ECC parity values | | | | | | | | | | | | | | | |
| | 804Ch | 10098h | | | | | | | | | | | | | | | | |
| | 804Dh | 1009Ah | | | | | | | | | | | | | | | | |
| | 804Eh | 1009Ch | | | | | | | | | | | | | | | | |
| | 804Fh | 1009Eh | | | | | | | | | | | | | | | | |

**NOTE :**
In case of 'with ECC' mode, Flex-MuxOneNAND automatically generates ECC code for both main and spare data of memory during program operation, but does not update ECC code to spare bufferRAM during load operation.

## 2.8 Registers

Section 2.8 of this specification provides information about the Flex-MuxOneNAND4G registers.

### 2.8.1 Register Address Map

This map describes the register addresses, register name, register description, and host accessibility.

| Address (word order) | Address (byte order) | Name | Host Access | Description |
|---|---|---|---|---|
| F000h | 1E000h | Manufacturer ID | R | Manufacturer identification |
| F001h | 1E002h | Device ID | R | Device identification |
| F002h | 1E004h | Version ID | R | N/A |
| F003h | 1E006h | Data Buffer size | R | Data buffer size |
| F004h | 1E008h | Boot Buffer size | R | Boot buffer size |
| F005h | 1E00Ah | Amount of buffers | R | Amount of data/boot buffers |
| F006h | 1E00Ch | Technology | R | Info about technology |
| F007h~F0FFh | 1E00Eh~1E1FEh | Reserved | - | Reserved for user |
| F100h | 1E200h | Start address 1 | R/W | Chip address for selection of NAND Core in DDP & Block address |
| F101h | 1E202h | Start address 2 | R/W | Chip address for selection of BufferRAM in DDP |
| F102h~F106h | 1E204h~1E20Ch | Reserved | - | Reserved for user |
| F107h | 1E20Eh | Start address 8 | R/W | NAND Flash Page & Sector Address |
| F108h~F1FFh | 1E210h~1E3FEh | Reserved | - | Reserved for user |
| F200h | 1E400h | Sector Count | R/W | Sector Number for the page data transfer from the memory and the BufferRAM |
| F201h~F21Fh | 1E402h~1E43Eh | Reserved | - | Reserved for vendor specific purposes |
| F220h | 1E440h | Command | R/W | Host control and memory operation commands |
| F221h | 1E442h | System Configuration 1 | R, R/W | memory and Host Interface Configuration |
| F222h~F22Fh | 1E444h~1E45Eh | Reserved | - | Reserved for user |
| F230h~F23Fh | 1E460h~1E47Eh | Reserved | - | Reserved for vendor specific purposes |
| F240h | 1E480h | Controller Status | R | Controller Status and result of memory operation |
| F241h | 1E482h | Interrupt | R/W | Memory Command Completion Interrupt Status |
| F242h~F24Bh | 1E484h~1E496h | Reserved | - | Reserved for user |
| F24Ch | 1E498h | Start Block Address | R/W | Start memory block address in Write Protection mode |
| F24Dh | 1E49Ah | Reserved | - | Reserved for user |
| F24Eh | 1E49Ch | Write Protection Status | R | Current memory Write Protection status (unlocked/locked/tight-locked) |
| F24Fh~FEFFh | 1E49Eh~1FDFEh | Reserved | - | Reserved for user |
| FF00h | 1FE00h | ECC Status Register 1 | R | ECC status of sector0 and sector 1 |
| FF01h | 1FE02h | ECC Status Register 2 | R | ECC status of sector2 and sector 3 |
| FF02h | 1FE04h | ECC Status Register 3 | R | ECC status of sector4 and sector 5 |
| FF03h | 1FE06h | ECC Status Register 4 | R | ECC status of sector6 and sector 7 |
| FF04h~FFFFh | 1FE08h~1FFFEh | Reserved | - | Reserved for vendor specific purposes |

### 2.8.2 Manufacturer ID Register F000h (R)

This Read register describes the manufacturer's identification.
Samsung Electronics Company manufacturer's ID is 00ECh.

F000h, default = 00ECh

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ManufID | | | | | | | | | | | | | | | |

### 2.8.3 Device ID Register F001h (R)

This Read register describes the device.

F001h, see table for default.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DeviceID | | | | | | | | | | | | | | | |

**Device Identification**

| Device Identification | Description |
|---|---|
| DeviceID [1:0] Vcc | 00 = 1.8V, 01 = 3.3V, 10/11 = reserved |
| DeviceID [2] Muxed/Demuxed | 0 = Muxed, 1 = Demuxed |
| DeviceID [3] Single/DDP | 0 = Single, 1 = DDP/QDP |
| DeviceID [7:4] Density | 0000 = 128Mb, 0001 = 256Mb, 0010 = 512Mb,<br>0011 = 1Gb, 0100 = 2Gb, 0101=4Gb, 0110=8Gb, 0111=16Gb |
| DeviceID [9:8] Separation | 10=Flex[SLC&MLC], 01=MLC, 00=SLC, 11=reserved |

**Device ID Default**

| Device | DeviceID[15:0] |
|---|---|
| KFM4GH6Q4M | 0250h |
| KFN8GH6Q4M | 0268h |
| KFKAGH6Q4M | 0268h[1)] |

**NOTE :**
1) The base density of all the three device is 4Gb, DDP and QDP use 2 and 4 multiplexed chips respectively, hence DDP and QDP device ID is same.

### 2.8.4 Version ID Register F002h

This register is reserved for future use.

### 2.8.5 Data Buffer Size Register F003h (R)

This Read register describes the size of the Data Buffer.

F003h, default = 0800h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DataBufSize | | | | | | | | | | | | | | | |

### 2.8.6 Boot Buffer Size Register F004h (R)

This Read register describes the size of the Boot Buffer.

F004h, default = 0200h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BootBufSize | | | | | | | | | | | | | | | |

| Register Information | Description |
|---|---|
| BootBufSize | Total boot buffer size in Words equal to 1 buffer of 512 Words ($1 \times 512 = 2^9$) in the memory interface |

### 2.8.7 Amount of Buffers Register F005h (R)

This Read register describes the number of each Buffer.

F005h, default = 0201h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DataBufAmount | | | | | | | | BootBufAmount | | | | | | | |

**Number of Buffers Information**

| Register Information | Description |
|---|---|
| DataBufAmount | The number of data buffers = 2 ($2^N$, N=1) |
| BootBufAmount | The number of boot buffers = 1 ($2^N$, N=0) |

### 2.8.8 Technology Register F006h (R)

This Read register describes the internal NAND array technology.

F006h, default = 0001h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tech | | | | | | | | | | | | | | | |

**Technology Information**

| Technology | Register Setting |
|---|---|
| NAND SLC | 0000h |
| NAND MLC | 0001h |
| Reserved | 0002h ~ FFFFh |

**NOTE :**
Flex-OneNAND has underlying MLC technology.

### 2.8.9 Start Address1 Register F100h (R/W)

This Read/Write register describes the NAND Flash block address which will be loaded, programmed, or erased.

F100h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DFS | Reserved(00000) | | | | | FBA | | | | | | | | | |

| Device | Number of Block | FBA |
|---|---|---|
| 4Gb | 1024 | FBA[9:0] |
| 8Gb DDP | 2048 | DFS[15] & FBA[9:0] |

**NOTE :**
For QDP, See Section 7.4

**Start Address1 Information**

| Register Information | Description |
|---|---|
| FBA | NAND Flash Block Address |
| DFS | Flash Core of DDP (Device Flash Core Select) |

### 2.8.10 Start Address2 Register F101h (R/W)

This Read/Write register describes the method to select the BufferRAM of DDP (Device BufferRAM Select)

F101h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DBS | Reserved(000000000000000) | | | | | | | | | | | | | | |

**Start Address2 Information**

| Register Information | Description |
|---|---|
| DBS | BufferRAM and Register of DDP (Device BufferRAM Select) |

>DBS should be set to 1 when accessing the BufferRAM of the second chip(MSB chip) in a DDP.
>Since DDP chip has 2 BufferRAMs multiplexed, the BufferRAM which corresponds to the Flash core that is intended to be accessed must be selected using DBS.
>Data in BufferRAM of one chip is not accessible to the Flash Core of the other chip in a DDP See Section 7.4.

### 2.8.11~15 Start Address3~7 Register F102h~F106h

This Register is reserved for future use.

### 2.8.16 Start Address8 Register F107h (R/W)

This Read/Write register describes the NAND Flash start page address in a block for a page load, program operation and the NAND Flash start sector address in a page for a load, or program operation.

F107h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved (0000000) | | | | | | | FPA | | | | | | | FSA | |

**Start Address8 Information**

| Item | Description | Default Value | Range |
|---|---|---|---|
| FPA | NAND Flash Page Address | 0000000 | 0000000 ~ 1111111,<br>7 bits for 128 pages[1)] |
| FSA | NAND Flash Sector Address | 00 | 00 (sector0), 01 (sector1),<br>10 (Sector2) , 11 (sector3) |

**NOTE :**
1) Only 6bits must be used for 64pages in SLC area. (SLC:64pages, MLC:128pages)
2) Sectors 4-7 in a page are not directly addessable using FSA. However, they can be accessed using BSA and BSC (See Below).FSA must be 00 in program operation.

### 2.8.17 Start Buffer Register F200h (R/W)

This Buffer Sector Count(BSC) specifies the number of sectors to be loaded.

F200h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved(0000) | | | | BSA[1)] | | | | Reserved(00000) | | | | | BSC | | |

The BufferRAM Sector Address (BSA) is the sector 0~3 address in the internal BootRAM and DataRAM where data is placed.

**NOTE :**
1) In case of 'Program' and 'Load', Internally BSA fix first sector of DataRAM0(BSA=1000).

| Item | Description | BSC Value | Number of Sectors |
|---|---|---|---|
| BSC<br>(CASE1 : FSA=00) | Buffer Sector Count | 000 (Default) | 8 sectors |
| | | 001 | 1 sectors |
| | | 010 | 2 sectors |
| | | 011 | 3 sectors |
| | | 100 | 4 sectors |
| | | 101 | 5 sectors |
| | | 110 | 6 sectors |
| | | 111 | 7 sectors |
| BSC<br>(CASE2 : FSA=01) | Buffer Sector Count | 001 | 1 sectors |
| | | 010 | 2 sectors |
| | | 011 | 3 sectors |
| BSC<br>(CASE3 : FSA=10) | Buffer Sector Count | 001 | 1 sectors |
| | | 010 | 2 sectors |
| BSC<br>(CASE4 : FSA=11) | Buffer Sector Count | 001 | 1 sectors |

**NOTE :**
1) BSC is used only on load operation.
2) Operation not guaranteed for cases not defined in above table(CASE1, CASE2, CASE3,CASE4).

**Sector allocation according to BSC(CASE1 : FSA=00)**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **BSC = 000** | Sector0 | Sector1 | Sector2 | Sector3 | Sector4 | Sector5 | Sector6 | Sector7 |
| **BSC = 001** | Sector0 | | | | | | | |
| **BSC = 010** | Sector0 | Sector1 | | | | | | |
| **BSC = 011** | Sector0 | Sector1 | Sector2 | | | | | |
| **BSC = 100** | Sector0 | Sector1 | Sector2 | Sector3 | | | | |
| **BSC = 101** | Sector0 | Sector1 | Sector2 | Sector3 | Sector4 | | | |
| **BSC = 110** | Sector0 | Sector1 | Sector2 | Sector3 | Sector4 | Sector5 | | |
| **BSC =111** | Sector0 | Sector1 | Sector2 | Sector3 | Sector4 | Sector5 | Sector6 | |

**Sector allocation according to BSC(CASE2 : FSA=01)**

| | | | |
|---|---|---|---|
| **BSC = 001** | Sector1 | | |
| **BSC = 010** | Sector1 | Sector2 | |
| **BSC = 011** | Sector1 | Sector2 | Sector3 |

**Sector allocation according to BSC(CASE3 : FSA=10)**

| | | |
|---|---|---|
| **BSC = 001** | Sector2 | |
| **BSC = 010** | Sector2 | Sector3 |

**Sector allocation according to BSC(CASE4 : FSA=11)**

| | |
|---|---|
| **BSC = 001** | Sector3 |

*** The first sector from Flash(The first sector is determined by FSA. In case of FSA=01[CASE2], the first sector is Sector1.) is transferred to the 1st sector(sector0) of DataRAM0, and the other sectors are transferred sequentially.**

### 2.8.18 Command Register F220h (R/W)

Command can be issued by two following methods, and user may select one way or the other to issue appropriate command;

1. Write command into Command Register when INT is at ready state. INT will automatically turn to busy state as command is issued. Once the desired operation is completed, INT will go back ready state.
2. Write 0000h to INT bit of Interrupt Status Register, and then write command into Command Register. Once the desired operation is completed, INT will go back to ready state.
(00F0h and 00F3h may be accepted during busy state of some operations. Refer to the right most column of the command register table below.)

F220h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Command | | | | | | | | | | | | | | | |

| CMD | Operation | Acceptable command during busy |
|---|---|---|
| 0000h | Load a page unit into buffer | 00F0h, 00F3h |
| 0003h | Superload a page unit from buffer | 00F0h, 00F3h |
| 0005h | LSB page recovery Read [1)]<br>PI update [2)] | 00F0h, 00F3h |
| 0080h | Program a page unit from buffer &<br>Finish Program operation at Cache Program operation | 00F0h, 00F3h |
| 007Fh | Cache Program operation | 00F0h, 00F3h |
| 0023h | Unlock NAND array a block | 00F0h, 00F3h |
| 002Ah | Lock NAND array a block | 00F0h, 00F3h |
| 002Ch | Lock-tight NAND array a block | 00F0h, 00F3h |
| 0027h | All Block Unlock | 00F0h, 00F3h |
| 0094h | Block Erase | 00F0h, 00F3h |
| 00B0h | Erase Suspend | 00F0h, 00F3h |
| 0030h | Erase Resume | 00F0h, 00F3h |
| 00F0h | Reset NAND Flash Core | - |
| 00F3h | Reset Flex-MuxOneNAND [3)] | - |
| 0065h | OTP Access | 00F0h, 00F3h |
| 0066h | Access to Partition Information(PI) Block | 00F0h, 00F3h |

**NOTE :**
1) LSB page recovery Read command can always be issued but not in the PI Block access mode.
2) In PI Block Access mode, PI update can be issued.
3) 'Reset Flex-MuxOneNAND'(=Hot reset) command makes the registers and NAND Flash core into default state.

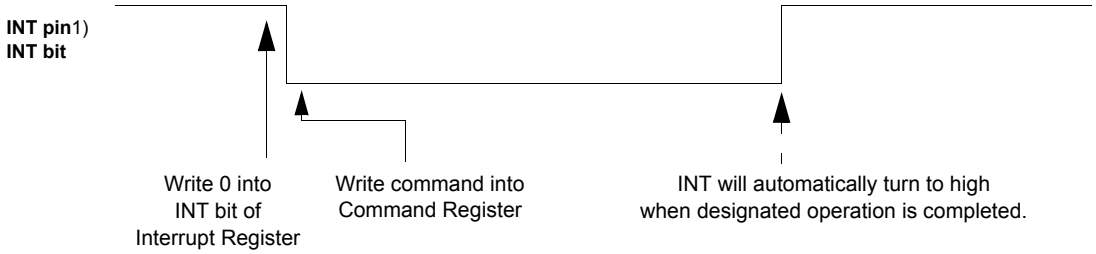
### 2.8.18.1 Two Methods to Clear Interrupt Register in Command Input

To clear Interrupt Register in command input, user may select one from either following methods.
First method is to turn INT to low by manually writing 0000h to INT bit of Interrupt Register. 1)
Second method is to input command while INT is high, and the device will automatically turn INT to low.1)
(Second method is equivalent with method used in general NAND Flash)

User may choose the desirable method to clear Interrupt Register.

**Method 1: Manually set INT=0 before writing command into Command Register: Manual INT Mode**

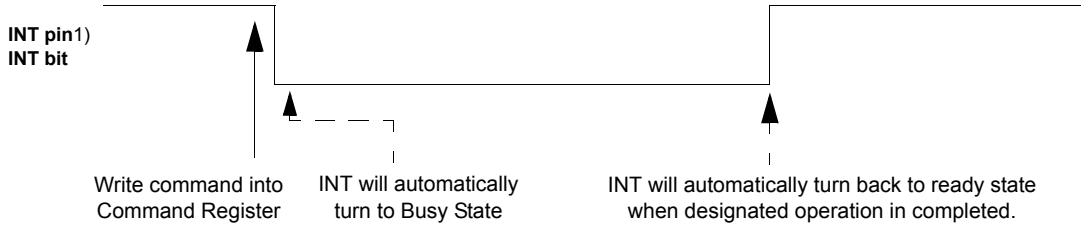
(1) Clear Interrupt Register (F241h) by writing 0000h into INT bit of Interrupt Register. This operation will make INT pin turn low. 1)
(2) Write command into Command Register. This will make the device to perform the designated operation.
(3) INT pin will turn back to high once the operation is completed. 1)




**NOTE : 1) INT pin polarity is based on 'IOBE=1 and INT pol=1 (default)' setting**

**Method 2: Write command into Command Register at INT ready state: Auto INT Mode**

(1) Write command into Command Register. This will automatically turn INT from high to low. 1)
(2) INT pin will turn back to high once the operation is completed. 1)




**NOTE : 1) INT pin polarity is based on 'IOBE=1 and INT pol=1 (default)' setting**

### 2.8.19 System Configuration 1 Register F221h (R, R/W)

This Read/Write register describes the system configuration.

F221h, default =40C0h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R/W | R/W | | | R/W | | | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R |
| RM | BRWL | | | BL | | | ECC | RDY pol | INT pol | IOBE | RDY Conf | Reserved | HF | WM | BWPS |

**Read Mode (RM)**

| RM | Read Mode |
|---|---|
| 0 | Asynchronous read(default) |
| 1 | Synchronous read |

**Read Mode Information[15]**

| Item | Definition | Description |
|---|---|---|
| RM | Read Mode | Selects between asynchronous read mode and synchronous read mode |

**Burst Read Write Latency (BRWL)**

| BRWL | Latency Cycles (Read/Write) | | |
|---|---|---|---|
| | under 40MHz (HF=0) | 40MHz~66MHz (HF=0) | over 66MHz (HF=1) |
| 000~010 | Reserved | | |
| 011 | 3(up to 40MHz. min) | 3(N/A) | 3(N/A) |
| 100 (default) | 4 | 4(min.) | 4(N/A) |
| 101 | 5 | 5 | 5(N/A) |
| 110 | 6 | 6 | 6(min.) |
| 111 | 7 | 7 | 7 |

* Default value of BRWL and HF value is BRWL=4, HF=0.
For host frequency over 66MHz, BRWL should be 6 or 7 while HF is 1.
For host frequency range of 40MHz~66MHz, BRWL should be set to 4~7 while HF is 0.
For host frequency under 40MHz, BRWL should be set to 3~7 while HF is 0.

**Burst Read Write Latency (BRWL) Information[14:12]**

| Item | Definition | Description |
|---|---|---|
| BRWL | Burst Read Latency / Burst Write Latency | Specifies the access latency in the burst read / write transfer for the initial access |

**Burst Length (BL)**

Host must follow burst length set by BL when reading data in synchronous burst read.

| BL | Burst Length(Main) | Burst Length(Spare) |
|---|---|---|
| 000 | Continuous(Default)[1)] | |
| 001 | 4 words | |
| 010 | 8 words | |
| 011 | 16 words | |
| 100 | 32 words | N/A |
| 101~111 | Reserved | |

**NOTE :**
1) In case of BootRAM : Main=512word, Spare=16word
In case of DataRAM : Main=1Kword, Spare=32word

**Burst Length (BL) Information[11:9]**

| Item | Definition | Description |
|---|---|---|
| BL | Burst Length | Specifies the size of the burst length during a synchronous linear burst read and wrap around. And also burst length during a synchronous linear burst write |

**Error Correction Code (ECC) Information[8]**

| Item | Definition | Description |
|---|---|---|
| ECC | Error Correction Code Operation | 0 = with correction (default)<br>1 = without correction (bypassed) |

**RDY Polarity (RDYpol) Information[7]**

| Item | Definition | Description |
|---|---|---|
| RDYpol | RDY signal polarity | 1 = high for ready (default)<br>0 = low for ready |

**INT Polarity (INTpol) Information[6]**

| INTpol | INT bit of Interrupt Status Register | INT Pin output |
|---|---|---|
| 0 | 0 (busy) | High |
| | 1 (ready) | Low |
| 1 (default) | 0 (busy) | Low |
| | 1 (ready) | High |

**I/O Buffer Enable (IOBE)**

IOBE is the I/O Buffer Enable for the INT and RDY signals. At startup, INT and RDY outputs are High-Z. Bits 6 and 7 become valid after IOBE is set to "1". IOBE can be reset by a Cold Reset or by writing "0" to bit 5 of System Configuration1 Register.

**I/O Buffer Enable Information[5]**

| Item | Definition | Description |
|---|---|---|
| IOBE | I/O Buffer Enable for INT and RDY signals | 0 = disable (default)<br>1 = enable |

**RDY Configuration (RDY conf)**

**RDY Configuration Information[4]**

| Item | Definition | Description |
|---|---|---|
| RDY conf | RDY configuration | 0=active with valid data (default)<br>1=active one clock before valid data |

**HF Enable (HF)**

| HF | Description |
|---|---|
| 0 | HF Disable (default, under 66MHz) |
| 1 | HF Enable (over 66MHz) |

**HF Information[2]**

| Item | Definition | Description |
|---|---|---|
| HF | High Frequency | Selects between HF Disable and HF Enable |

**Write Mode (WM)**

| WM | Write Mode |
|---|---|
| 0 | Asynchronous Write(default) |
| 1 | Synchronous Write |

**Write Mode Information[1]**

| Item | Definition | Description |
|---|---|---|
| WM | Write Mode | Selects between asynchronous Write Mode and synchronous Write Mode |

**MRS(Mode Register Setting) Description**

| RM | WM | Mode Description |
|---|---|---|
| 0 | 0 | Asynch Read & Asynch Write (Default) |
| 1 | 0 | Sync Read & Asynch Write |
| 1 | 1 | Sync Read & Synch Write |
| Other Cases | | Reserved[1)] |

**NOTE :**
1) Operation not guaranteed for cases not defined in above table.

**Boot Buffer Write Protect Status(BWPS)**

**Boot Buffer Write Protect Status Information[0]**

| Item | Definition | Description |
|---|---|---|
| BWPS | Boot Buffer Write Protect Status | 0=locked(fixed) |

### 2.8.20 System Configuration 2 Register F222h

This register is reserved for future use.

### 2.8.21 Controller Status Register F240h (R)

This Read register shows the overall internal status of the Flex-MuxOneNAND and the controller.

F240h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OnGo | Reserved(0000) | | | | Error | Reserved(0) | $PI_L$ | Reserved(0) | $OTP_L$ | $OTP_{BL}$ | Reserved(00) | | Previous | Current | TO (0) |

**OnGo**

This bit shows the overall internal status of the Flex-MuxOneNAND device.
In Cache Program Operation, OnGo bit shows the overall status of Cache Program process.

**OnGo Information[15]**

| Item | Definition | Description |
|---|---|---|
| OnGo | Internal Device Status | 0 = ready<br>1 = busy |

**Error**

This bit shows the overall Error status.
In case of Cache Program, Error bit will show the accumulative error status of Cache Program operation, so that if an error occurs during Cache Program, this bit will stay as Fail status, until the end of Cache Program.

**Error Information[10]**

| Error | Load Program, Cache Program, and Erase Result |
|---|---|
| 0 | Pass |
| 1 | Fail |

**PI Lock Status ($PI_L$)**

This bit shows whether the PI block is locked or unlocked. Locking the PI has the effect of a 'Program/Erase protect' to guard against accidental re-programming of data stored in the PI block.
The PI status bit is automatically updated at power-on and PI update operation by PI Update command.

**OTP Lock Status ($OTP_L$)**

This bit shows whether the OTP block is locked or unlocked. Locking the OTP has the effect of a 'write-protect' to guard against accidental re-programming of data stored in the OTP block.
The $OTP_L$ status bit is automatically updated at power-on.

**OTP Lock Information[6]**

| $OTP_L$ | OTP Locked/Unlocked Status |
|---|---|
| 0 | OTP Block Unlock Status(Default) |
| 1 | OTP Block Lock Status(Disable OTP Program/Erase) |

**1st Block OTP Lock Status ($OTP_{BL}$)**

This bit shows whether the 1st Block OTP is locked or unlocked.
Locking the 1st Block OTP has the effect of a 'Program/Erase protect' to guard against accidental re-programming of data stored in the 1st block.
The $OTP_{BL}$ status bit is automatically updated at power-on.

**OTP Lock Information[5]**

| $OTP_{BL}$ | 1st Block OTP Locked/Unlocked Status |
|---|---|
| 0 | 1st Block OTP Unlock Status(Default) |
| 1 | 1st Block OTPLock Status(Disable 1st Block OTP Program/Erase) |

**Previous Cache Program status (Previous)**

This bit shows the previous program status of Cache Program. This value is invalid only at the first ‘Read Controller Status Register’ step of Cache Program operation. (Refer to 6.12 and 6.13)

**Previous [2]**

| Previous | Status of previous program |
|---|---|
| 0 | Pass |
| 1 | Fail |

**Current Cache Program Status (Current)**

This bit shows the current program status only at Final Cache Program.

**Current Information[1]**

| Current | Status of current program |
|---|---|
| 0 | Pass |
| 1 | Fail |

**Time Out (TO)**

This bit determines if there is a time out for load, program, and erase operations. It is fixed at 'no time out'.

**TO Information[0]**

| Item | Definition | Description |
|---|---|---|
| TO | Time Out | 0 = no time out |

**Controller Status Register Output Modes**

| Mode | Controller Status Register [15:0] | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | [15] | [14] | [13] | [12] | [11] | [10] | [9] | [8] | [7] | [6] | [5] | [4] | [3] | [2] | [1] | [0] |
| | OnGo | Reserved | | | | Error | Reserved | $PI_L$ 1) | Reserved | $OTP_L$ 2) | $OTPB_L$ 3) | Reserved | | Previous | Current | TO |
| Operation Ongoing | 1 | 0000 | | | | 0 | 0 | 0/1 | 0 | 0/1 | 0/1 | 00 | | 0 | 0 | 0 |
| Operation OK | 0 | | | | | 0 | | 0/1 | | 0/1 | 0/1 | | | 0 | 0 | 0 |
| Operation Fail | 0 | | | | | 1 | | 0/1 | | 0/1 | 0/1 | | | 0 | 0 | 0 |
| Program fail on Cache Program | 0 | | | | | 1 | | 0/1 | | 0/1 | 0/1 | | | 0 | 1 | 0 |
| Previous program fail during Cache Program | 0 | | | | | 1 | | 0/1 | | 0/1 | 0/1 | | | 1 | 0 | 0 |
| Program fail after Finish Cache Program | 0 | | | | | 1 | | 0/1 | | 0/1 | 0/1 | | | (Note 4) | (Note 4) | 0 |
| Reset during Program/Erase/Load | 0 | | | | | 0 | | 0/1 | | 0/1 | 0/1 | | | 0 | 0 | 0 |
| Program/Erase to the locked block, Load to the BootRAM | 0 | | | | | 1 | | 0/1 | | 0/1 | 0/1 | | | 0 | 0 | 0 |
| OTP Program Fail(Lock) | 0 | | | | | 1 | | 0/1 | | 1 | 1 | | | 0 | 0 | 0 |
| OTP Program Fail | 0 | | | | | 1 | | 0/1 | | 0 | 0 | | | 0 | 0 | 0 |

**NOTE :**
1) "1" for $PI_L$ Block Lock, "0" for $PI_L$ Block Unlock.
2) "1" for 1st Block OTP Lock, "0" for 1st Block OTP Unlock.
3) "1" for OTP Block Lock, "0" for OTP Block Unlock.
4) After Finish Cache Program operation, pass/fail status of Current Cache Program and Previous Cache Program will be updated.

## 2.8.22 Interrupt Status Register F241h (R/W)

This Read/Write register shows status of the Flex-MuxOneNAND interrupts.

F241h, defaults = 8080h after Cold Reset; 8010h after Warm/Hot Reset

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INT | Reserved(0000000) | | | | | | | RI | WI | EI | RSTI | Reserved(0000) | | | |

**Interrupt (INT)**

This is the master interrupt bit. The INT bit is wired directly to the INT pin on the chip. Upon writing '0' to the INT bit, the INT pin goes low if INTpol is high and goes high if INTpol is low.

**INT Interrupt [15]**

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|---|---|---|---|---|---|
| | | Cold | Warm/hot | | |
| | | 1 | 1 | 0 | off |
| sets itself to '1' | Commands in the command table in page43 (Refer to Chapter 2.8.18) are completed. | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, Cold/Warm/Hot reset is being performed, or command is written to Command Register in INT auto mode | | | 1→0 | off |

**Read Interrupt (RI)**

This is the Read interrupt bit.

**RI Interrupt [7]**

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|---|---|---|---|---|---|
| | | Cold | Warm/hot | | |
| | | 1 | 0 | 0 | off |
| sets itself to '1' | At the completion of a Load, Superload or LSB Page Recovery Read Operation. (0000h, 0003h or 0005h) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, Cold/Warm/Hot reset is being performed, or command is written to Command Register in INT auto mode | | | 1→0 | off |

**Write Interrupt (WI)**

This is the Write interrupt bit.

**WI Interrupt [6]**

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|---|---|---|---|---|---|
| | | Cold | Warm/hot | | |
| | | 0 | 0 | 0 | off |
| sets itself to '1' | At the completion of an Program Operation (0080h and 007Fh) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, Cold/Warm/Hot reset is being performed, or command is written to Command Register in INT auto mode | | | 1→0 | off |

**Erase Interrupt (EI)**

This is the Erase interrupt bit.

**EI Interrupt [5]**

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|---|---|---|---|---|---|
| | | Cold | Warm/hot | | |
| | | 0 | 0 | 0 | off |
| sets itself to '1' | At the completion of an Erase Operation (0094h and 0030h) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, Cold/Warm/Hot reset is being performed, or command is written to Command Register in INT auto mode | | | 1→0 | off |

**Reset Interrupt (RSTI)**

This is the Reset interrupt bit.

**RSTI Interrupt [4]**

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|---|---|---|---|---|---|
| | | Cold | Warm/hot | | |
| | | 0 | 1 | 0 | off |
| sets itself to '1' | At the completion of an Reset Operation (00B0h, 00F0h, 00F3h or warm reset is released) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, or command is written to Command Register in INT auto mode | | | 1→0 | off |

## 2.8.23 Start Block Address Register F24Ch (R/W)

This Read/Write register shows the NAND Flash block address in the Write Protection mode. Setting this register precedes a 'Lock Block' command, 'Unlock Block' command, or 'Lock-Tight' Command.

F24Ch, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved(000000) | | | | | | SBA | | | | | | | | | |

| Device | Number of Block | SBA |
|---|---|---|
| 4Gb | 1024 | [9:0] |

## 2.8.24 Start Block Address Register F24Dh (R/W)

This register is reserved for future use.

## 2.8.25 NAND Flash Write Protection Status Register F24Eh (R)

This Read register shows the Write Protection Status of the NAND Flash memory array.
To read the write protection status, FBA(DFS and DBS also in case of DDP) has to be set before reading the register.

F24Eh, default = 0002h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved(0000000000000) | | | | | | | | | | | | | US | LS | LTS |

**Write Protection Status Information[2:0]**

| Item | Bit | Definition | Description |
|---|---|---|---|
| US | 2 | Unlocked Status | 1 = current NAND Flash block is unlocked |
| LS | 1 | Locked Status | 1 = current NAND Flash block is locked<br>Or First Block of NAND Flash Array is Locked to be OTP |
| LTS | 0 | Locked-Tight Status | 1 = current NAND Flash block is locked-tight |

### 2.8.26 ECC Status Register 1 FF00h (R)

This Read register shows the Error Correction Status. The Flex-MuxOneNAND can correct up to 4-bit errors.
ECC can be performed on the NAND Flash main and spare memory areas. The ECC status register can also show the number of errors in a sector as a result of an ECC check in during a load operation. ECC status bits are also updated during a boot loading operation.

FF00h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | ER1 | | | | | Reserved | | | ER0 | | | | |

### 2.8.27 ECC Status Register 2 FF01h (R)

FF01h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | ER3 | | | | | Reserved | | | ER2 | | | | |

### 2.8.28 ECC Status Register 3 FF02h (R)

FF02h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | ER5 | | | | | Reserved | | | ER4 | | | | |

### 2.8.29 ECC Status Register 4 FF03h (R)

FF03h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | ER7 | | | | | Reserved | | | ER6 | | | | |

**Error Status**

| ER | ECC Status |
|---|---|
| 00000 | No Error |
| 00001 | 1bit error(correctable) |
| 00010 | 2bit error(correctable) |
| 00100 | 3bit error(correctable) |
| 01000 | 4bit error(correctable) |
| 10000 | Uncorrectable |

**ECC location Information**

| Item | Definition |
|---|---|
| ER0 | Error status of 1st selected sector (Main and Spare area) |
| ER1 | Error status of 2nd selected sector (Main and Spare area) |
| ER2 | Error status of 3rd selected sector (Main and Spare area) |
| ER3 | Error status of 4th selected sector (Main and Spare area) |
| ER4 | Error status of 5th selected sector (Main and Spare area) |
| ER5 | Error status of 6th selected sector (Main and Spare area) |
| ER6 | Error status of 7th selected sector (Main and Spare area) |
| ER7 | Error status of 8th selected sector (Main and Spare area) |

# 3.0 DEVICE OPERATION

This section of the data sheet discusses the operation of the Flex-MuxOneNAND device. It is followed by AC/DC Characteristics and Timing Diagrams which may be consulted for further information.

The Flex-MuxOneNAND supports a limited command-based interface in addition to a register-based interface for performing operations on the device.

## 3.1 Command Based Operation

Flex-OneNAND supports a limited command based interface. The address range of BootRAM ([0000h - 01FFh, 8000h - 800Fh]), called the Boot Partition is actually a read only area. This is because it contains bootloader code which must not be overwritten.
Therefore any attempt of data write to the Boot Partition is interpreted by Flex-OneNAND as a "Command based operation".
Commands can only be written with a Boot Partition address. Thus, the command-based interface is active only in the boot partition.
The remaining address range, except for the boot area, (address range [0200h - FFFFh]) can be used as a read/write data buffer.(with a few exceptions like ID registers). Writes outside the boot partition are treated as normal writes to the buffers or registers.
The command consists of one or more cycles depending on the command. After completion of the command the device starts its execution. Writing incorrect information including address and data to the boot partition or writing an improper command will terminate the previous command sequence and make the device enter the ready status.
The defined valid command sequences are stated in Command Sequences Table. Command based operations are mainly used when Flex-MuxOneNAND is used as Booting device, and all command based operations only supports asynchronous reads and writes. With DDP, command based operation except reset is applicable only on chip1.

**Command Sequences**

| Command Definition | | Cycles | 1st cycle | 2nd cycle |
|---|---|---|---|---|
| Reset Flex-MuxOneNAND | Add | 1 | BP[1] | |
| | Data | | 00F0h | |
| Load Data into Buffer[2] | Add | 2 | BP | BP |
| | Data | | 00E0h | 0000h[3] |
| Read Identification Data [5] | Add | 2 | BP | XXXXh[4] |
| | Data | | 0090h | Data |

**NOTE :**
1) BP(Boot Partition) : BootRAM Area [0000h ~ 01FFh, 8000h ~ 800Fh].
2) Load Data into Buffer operation is available within a block(128KB(SLC), 256KB(MLC)) (Chip1 only in case of DDP)
3) Load 4KB unit into DataRAM0, DataRAM1. Current Start address(FPA) is automatically incresed by 4KB unit after the load.
4) 0000h -> Data is Manufacturer ID (Chip1 only in case of DDP)
0001h -> Data is Device ID (Chip1 only in case of DDP)
0002h -> Current Block Write Protection Status (Chip1 only in case of DDP)
5) $\overline{WE}$ toggling can terminate 'Read Identification Data' operation.

### 3.1.1 Reset Flex-MuxOneNAND Command

The Reset command is given by writing 00F0h to the boot partition address. Reset will return all default values into the device.

### 3.1.2 Load Data Into Buffer Command

Load Data into Buffer command is a two-cycle command. Two sequential designated command activates this operation. Sequentially writing 00E0h and 0000h to the boot partition [0000h~01FFh, 8000h~800Fh] will load one page to DataRAM0 and DataRAM1. This operation refers to FBA. FSA must be 00 and BSA must be 1000.
At the end of this operation, FPA will be automatically increased by 1. So continuous issue of this command will sequentially load data in next page to DataRAM0, DataRAM1. This page address increment is restricted within a block.
The default value of FBA and FPA is 0. Therefore, initial issue of this command after power on will load the first page of memory, which is usually boot code.

### 3.1.3 Read Identification Data Command

The Read Identification Data command consists of two cycles. It gives out the devices identification data according to the given address. The first cycle is 0090h to the boot partition address and second cycle is read from the addresses specified in Identification Data Description Table.

**Identification Data Description**

| Address | Data Out |
|---|---|
| 0000h | Manufacturer ID (00ECh) |
| 0001h | Device ID[1)] |
| 0002h | Current Block Write Protection Status[2)] |

**NOTE :**
1) Refer to Device ID Register (Chapter 2.8.3)
2) To read the write protection status, FBA has to be set before issuing this command.

## 3.2 Device Bus Operation

The device bus operations are shown in the table below.

| Operation | $\overline{CE}$ | $\overline{OE}$ | $\overline{WE}$ | ADQ0~15 | $\overline{RP}$ | CLK | $\overline{AVD}$ |
|---|---|---|---|---|---|---|---|
| Standby | H | X | X | High-Z | H | X | X |
| Warm Reset | X | X | X | High-Z | L | X | X |
| Asynchronous Write | L | H | L | Add. In /Data In | H | L |  |
| Asynchronous Read | L | L | H | Add. In /Data Out | H | L |  |
| Start Initial Burst Read | L | H | H | Add. In | H |  |  |
| Burst Read | L | L | H | Burst Data Out | H |  | H |
| Terminate Burst Read Cycle | H | X | H | High-Z | H | X | X |
| Terminate Burst Read Cycle via $\overline{RP}$ | X | X | X | High-Z | L | X | X |
| Terminate Current Burst Read Cycle and Start New Burst Read Cycle | L | H | H | Add In | H |  |  |
| Start Initial Burst Write | L | H | L | Add In | H |  |  |
| Burst Write | L | H | X | Data In | H |  | H |
| Terminate Burst Write Cycle | H | H | X | High-Z | H | X | X |
| Terminate Burst Write Cycle via $\overline{RP}$ | X | X | X | High-Z | L | X | X |
| Terminate Current Burst Write Cycle and Start New Burst Write Cycle |  | H | L | Add In | H |  |  |

**NOTE :**
1) L=VIL (Low), H=VIH (High), X=Don't Care.

## 3.3 Reset Mode Operation

The Flex-MuxOneNAND has 4 reset modes: Cold/Warm/Hot Reset, and NAND Flash Array Reset. Section 3.3 discusses the operation of these reset modes.

The Register Reset Table shows the which registers are affected by the various types of Reset operations.

**Internal Register Reset Table**

| | Internal Registers | Cold Reset (Default) | Warm Reset (RP) | Hot Reset (00F3h) | Hot Reset (BP-F0h) | NAND Flash Core Reset (00F0h) |
|---|---|---|---|---|---|---|
| F000h | Manufacturer ID Register (R) | 00ECh | 00EC | 00ECh | | 00ECh |
| F001h | Device ID Register (R): Flex-MuxOneNAND | (Note 3) | N/A | N/A | | N/A |
| F002h | Version ID Register (R) | N/A | N/A | N/A | | N/A |
| F003h | Data Buffer size Register (R) | 0800h | N/A | N/A | | N/A |
| F004h | Boot Buffer size Register (R) | 0200h | N/A | N/A | | N/A |
| F005h | Amount of Buffers Register (R) | 0201h | N/A | N/A | | N/A |
| F006h | Technology Register (R) | 0001h | N/A | N/A | | N/A |
| F100h | Start Address1 Register (R/W): DFS, FBA | 0000h | 0000h | 0000h | | N/A |
| F101h | Start Address2 Register (R/W): DBS | 0000h | 0000h | 0000h | | N/A |
| F107h | Start Address8 Register (R/W): FPA | 0000h | 0000h | 0000h | | N/A |
| F200h | Start Buffer Register (R/W): BSC, BSA | 0000h | 0000h | 0000h | | N/A |
| F220h | Command Register (R/W) | 0000h | 0000h | 0000h | | N/A |
| F221h | System Configuration 1 Register (R/W) | 40C0h | (Note 1a) | (Note 1a) | | N/A |
| F240h | Controller Status Register (R) (Note 1b) (Note 4) | 0000h | 0000h | 0000h | | N/A |
| F241h | Interrupt Status Register (R/W) | 8080h | 8010h | 8010h | | N/A |
| F24Ch | Start Block Address (R/W) | 0000h | 0000h | N/A | | N/A |
| F24Eh | NAND Flash Write Protection Status (R) (Note 5) | 0002h | 0002h | N/A | | N/A |
| FF00h | ECC Status Register 1 (R) (Note 2) | 0000h | 0000h | 0000h | | N/A |
| FF01h | ECC Status Register 2 (R) (Note 2) | 0000h | 0000h | 0000h | | N/A |
| FF02h | ECC Status Register 3 (R) (Note 2) | 0000h | 0000h | 0000h | | N/A |
| FF03h | ECC Status Register 4 (R) (Note 2) | 0000h | 0000h | 0000h | | N/A |

**NOTE :**
1a) RDYpol, RDYconf, INTpol, IOBE are reset by Cold reset. The other bits are reset by cold/warm/hot reset
1b) The other bits except OTPL and OTPBL are reset by cold/warm/hot reset.
2) ECC Status Register 1~4 are reset when any command is issued.
3) Refer to Device ID Register F001h.
4) Resetting during IDLE state, this is valid. But resetting during BUSY state, refer to Chapter 2.8.21.
5) To read NAND Flash Write Protection status, Block Address register must be written before.

### 3.3.1 Cold Reset Mode Operation

**See Timing Diagram 6.15**

At system power-up, the voltage detector in the device detects the rising edge of Vcc and releases an internal power-up reset signal. This triggers boot code loading. Bootcode loading means that the boot loader in the device copies designated sized data (1KB) from the beginning of memory into the BootRAM. This sequence is the Cold Reset of Flex-MuxOneNAND.

The POR(Power On Reset) triggering level is typically 1.5V. Boot code copy operation activates 400us after POR.
Therefore, the system power should reach 1.7V within 400us from the POR triggering level for bootcode data to be valid.

It takes approximately 250us to copy 1KB of boot code. Upon completion of loading into the BootRAM, it is available to be read by the host. The INT pin is not available until after IOBE = 1 and IOBE bit can be changed by host.

### 3.3.2 Warm Reset Mode Operation

**See Timing Diagrams 6.16**

A Warm Reset means that the host resets the device by using the $\overline{RP}$ pin. When the a $\overline{RP}$ low is issued, the device logic stops all current operations and executes internal reset operation and resets current NAND Flash core operation synchronized with the falling edge of $\overline{RP}$.

During an Internal Reset Operation, the device initializes internal registers and makes output signals go to default status.
The BufferRAM data is kept unchanged after Warm/Hot reset operations.

The device guarantees the logic reset operation in case $\overline{RP}$ pulse is longer than tRP min(200ns).
The device may reset if tRP < tRP min(200ns), but this is not guaranteed.

Warm reset will abort the current NAND Flash core operation. During a warm reset, the content of memory cells being altered is no longer valid as the data will be partially programmed or erased.

Warm reset has no effect on contents of BootRAM and DataRAM.

### 3.3.3 Hot Reset Mode Operation

**See Timing Diagrams 6.17**

A Hot Reset means that the host resets the device by Reset command. The reset command can be either Command based or Register Based. Upon receiving the Reset command, the device logic stops all current operation and executes an internal reset operation and resets the current NAND Flash core operation.

During an Internal Reset Operation, the device initializes internal registers and makes output signals go to default status. The BufferRAM data is kept unchanged after Warm/Hot reset operations.

Hot reset has no effect on contents of BootRAM and DataRAM.

### 3.3.4 NAND Flash Core Reset Mode Operation

**See Timing Diagrams 6.18**

The Host can reset the NAND Flash Core operation by issuing a NAND Flash Core reset command. NAND Flash core reset will abort the current NAND Flash core operation. During a NAND Flash core reset, the content of memory cells being altered is no longer valid as the data will be partially programmed or erased.

NAND Flash Core Reset has an effect on neither contents of BootRAM and DataRAM nor register values.

## 3.4 Write Protection Operation

The Flex-MuxOneNAND can be write-protected to prevent re-programming or erasure of data.
The areas of write-protection are the BootRAM, and the NAND Flash Array.

### 3.4.1 BootRAM Write Protection Operation

At system power-up, voltage detector in the device detects the rising edge of Vcc and releases the internal power-up reset signal which triggers boot code loading. And the designated size data(1KB) is copied from the first page of the first block in the NAND flash array to the BootRAM.
After the bootcode loading is completed, the BootRAM is always locked to protect the boot code from the accidental write.

### 3.4.2 NAND Flash Array Write Protection Operation

The device has both hardware and software write protection of the NAND Flash array.

**Hardware Write Protection Operation**
The hardware write protection operation is implemented by executing a Cold or Warm Reset. On power up, the NAND Flash Array is in its default, locked state. The entire NAND Flash array goes to a locked state after a Cold or Warm Reset.

**Software Write Protection Operation**
The software write protection operation is implemented by writing a Lock command (002Ah) or a Lock-tight command (002Ch) to command register (F220h).

Lock (002Ah) and Lock-tight (002Ch) commands write protects the block defined in the Start Block Address Register F24Ch.

### 3.4.3 NAND Array Write Protection States

There are three lock states in the NAND Array: unlocked, locked, and locked-tight. On power up, all blocks in the NAND array go to Locked state. The lock status is maintained for each block in the NAND array. Any changes made to lock status of blocks are lost when Cold/warm reset occurs.
Flex-MuxOneNAND supports 4 commands for changing Write Protection states of the blocks: lock/unlock/lock-tight by one block, and All Block Unlock at once.
All Block Unlock command fails if there are lock-tight blocks in flash.

**Write Protection Status**
The current block Write Protection status can be read in NAND Flash Write Protection Status Register(F24Eh). There are three bits - US, LS, LTS -, which are not cleared by hot reset and NAND Flash Core Reset. These Write Protection status registers are updated when FBA is set, and when Write Protection command is entered.
The followings summarize locking status.

By default, [2:0] values are 010. For example:

-> If host executes unlock block operation, then [2:0] values turn to 100.

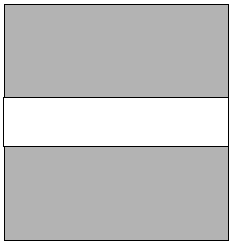
-> If host executes lock-tight block operation, then [2:0] values turn to 001.

### 3.4.3.1 Unlocked NAND Array Write Protection State

An Unlocked block can be programmed or erased. The status of an unlocked block can be changed to locked or locked-tight using the appropriate software command(Locked-tight state can be achieved in 2 steps. First, the block should be locked via the lock command. Then, Lock tight command must be issued.).

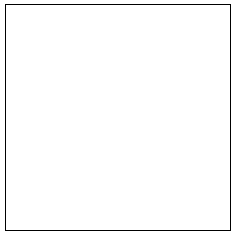
Only one block can be released from lock state to unlock state with Unlock command and addresses. The unlocked block can be changed with new lock command. Therefore, each block has its own lock/unlock/lock-tight state.

**Unlocked**



**Unlock Command Sequence:**
Start block address+Unlock block command (0023h)

**Unlocked**



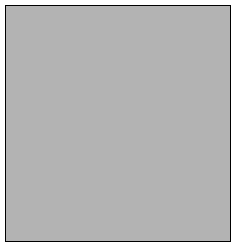
**All Block Unlock Command Sequence:**
Start block address(000h)+All Block Unlock command (0027h)

**NOTE :**
Even though SBA is fixed to 000h, Unlock will be done for all block. All block unlock is not valid if there is a lock-tight block. With DDP, all block unlock command must be issued on each chip.

### 3.4.3.2 Locked NAND Array Write Protection State

A Locked block cannot be programmed or erased. All blocks default to a locked state following a Cold or Warm Reset. Unlocked blocks can be changed to locked using the Lock block command. The status of a locked block can be changed to unlocked or locked-tight using the appropriate software command.

**Locked**

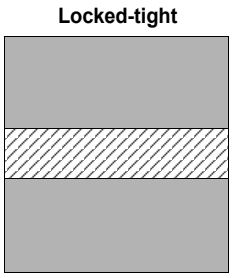


**Lock Command Sequence:**
Start block address+Lock block command (002Ah)

### 3.4.3.3 Locked-tight NAND Array Write Protection State

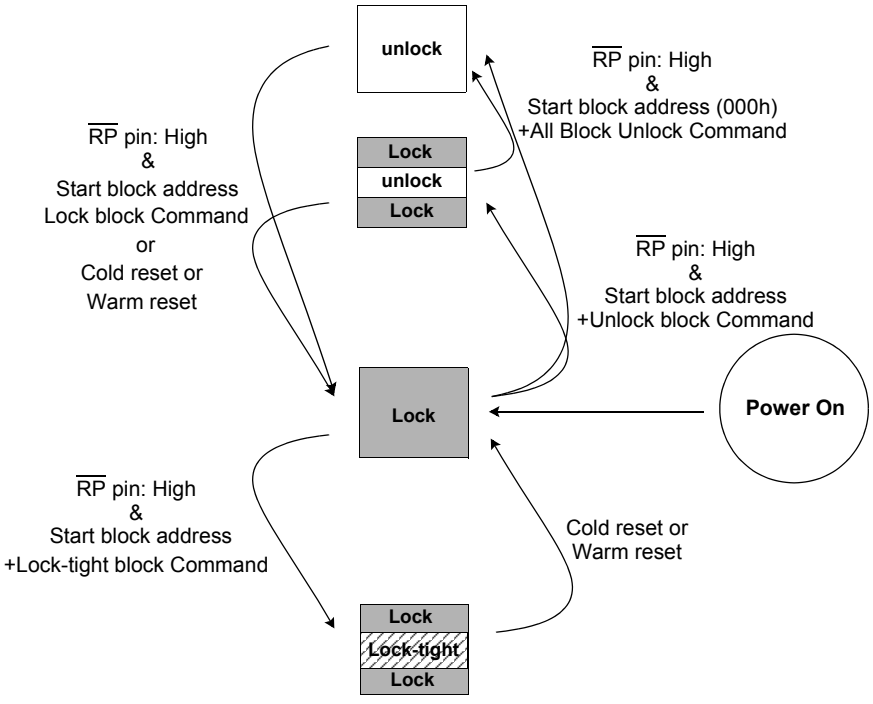
A block that is in a locked-tight state can only be changed to locked state after a Cold or Warm Reset. Unlock and Lock command sequences will not affect its state. This is an added level of write protection security.

A block must first be set to a locked state before it can be changed to locked-tight using the Lock-tight command. locked-tight blocks will revert to a locked state following a Cold or Warm Reset.
When there are Lock-tight blocks in the flash array, All Block Unlock Command will fail and there will be no change in the lock status of the blocks of the Flash array.
Thus, All Block Unlock command succeeds only when there are no tightly-locked blocks in Flash.



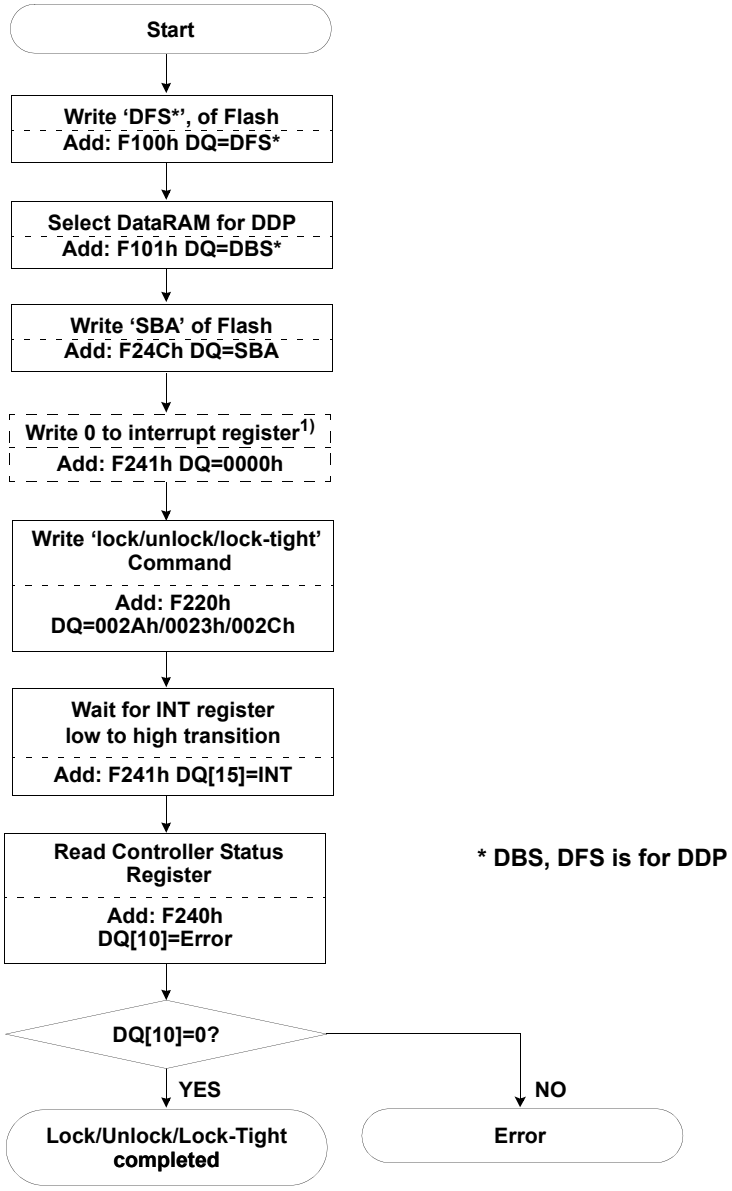

**Lock-Tight Command Sequence:**
Start block address+Lock-tight block command (002Ch)

### 3.4.4 NAND Flash Array Write Protection State Diagram




***NOTE : If the 1st Block is set to be OTP, Block 0 will always be Lock Status**

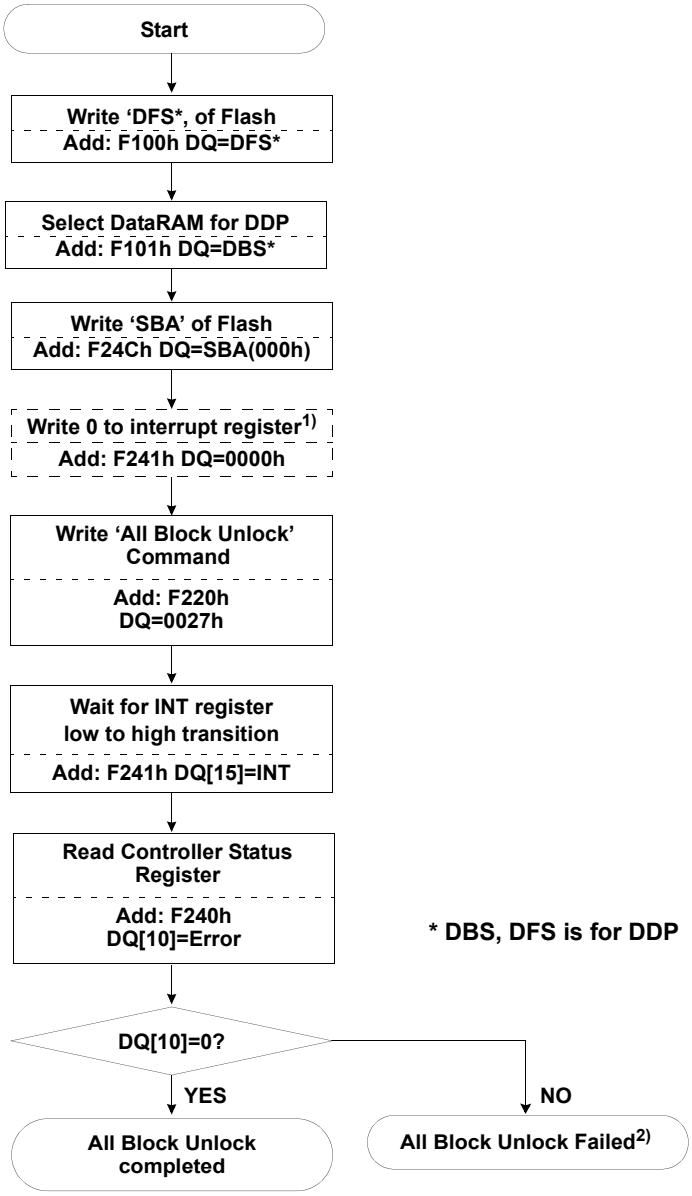
**Data Protection Operation Flow Diagram**




*** Samsung strongly recommends to follow the above flow chart**

**NOTE :**
1) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

**All Block Unlock Flow Diagram**




***Samsung strongly recommends to follow the above flow chart**

**NOTE :**
1) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
2) All Block Unlock command fails if there are lock-tight blocks in flash.

## 3.5 Data Protection During Power Down Operation

**See Timing Diagrams 6.19**

The device is designed to offer protection from any involuntary program/erase during power-transitions.
$\overline{RP}$ pin which provides hardware protection is recommended to be kept at VIL before Vcc drops to 1.5V.

## 3.6 Load Operation

**See Timing Diagrams 6.9**

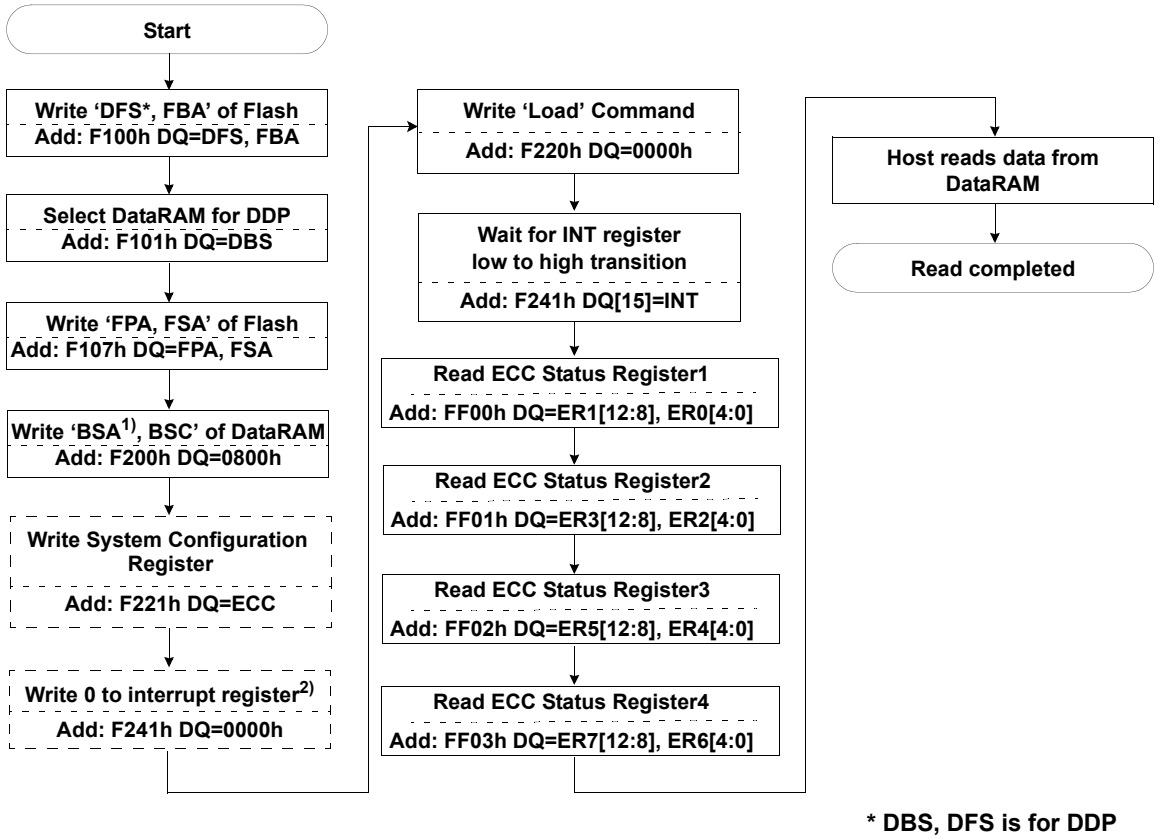
The Load operation is initiated by setting up the start address from which the data is to be loaded. The Load command is issued in order to initiate the load.

During a Load operation, the device:

-Transfers the data from NAND Flash array into the BufferRAM
-ECC is checked and any detected and corrected error is reported in the status response as well as any unrecoverable error.

Once the BufferRAM has been filled, an interrupt is issued to the host so that the contents of the BufferRAM can be read. The read from the BufferRAM can be an asynchronous read mode or synchronous read mode. The status information related to load operation can be checked by the host if required.

**Load Operation Flow Chart Diagram**




*** DBS, DFS is for DDP**

**NOTE :**
1) BSA must be 1000.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

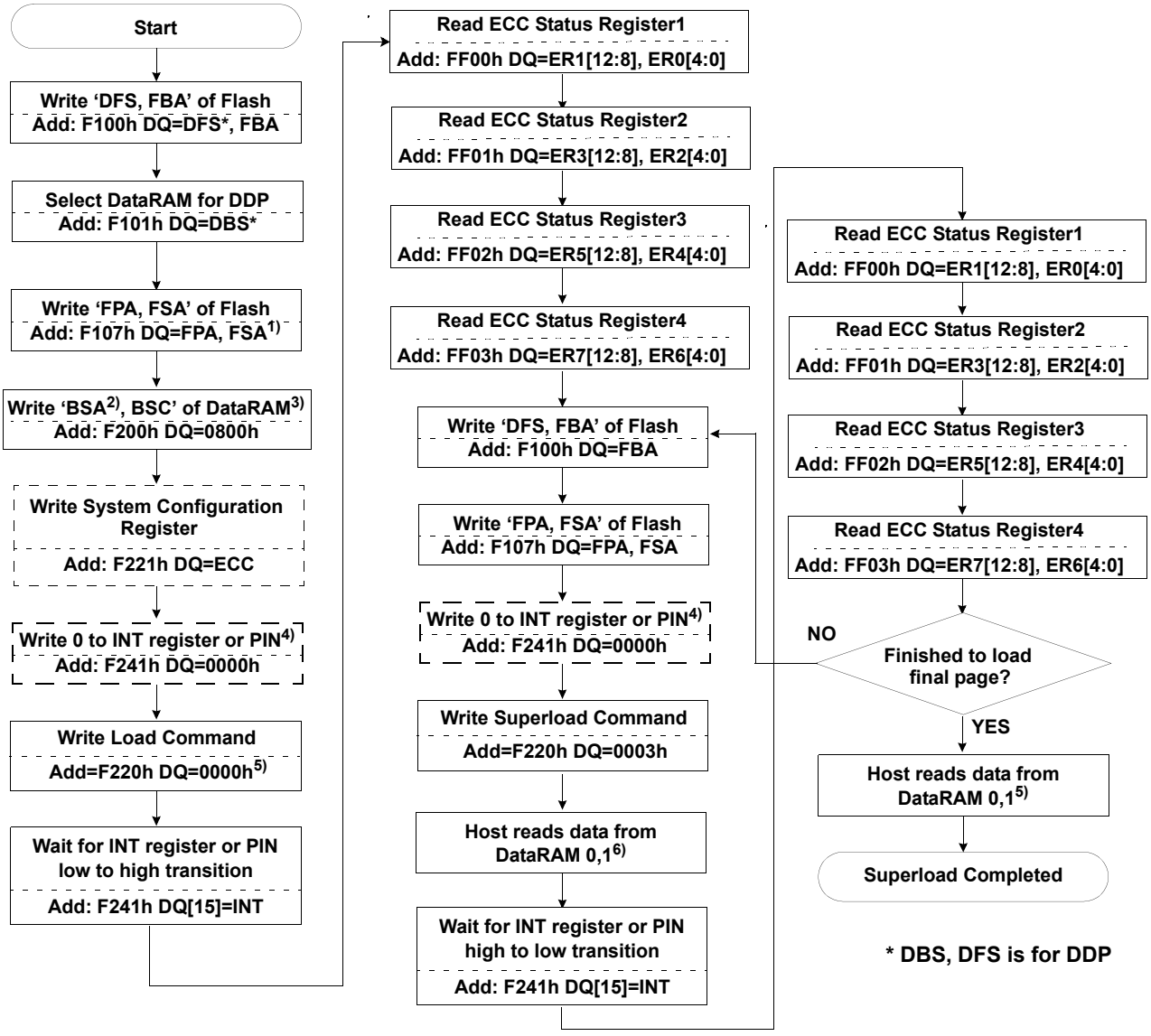
### 3.6.1 Superload Operation

**See Timing Diagrams 6.10**

The Superload operation is used to read multiple pages. During Superload operation, up to 4bit errors are corrected.
Once the first data is loaded, an interrupt status returns to ready. The data in DataRAM should be read after next Superload command is issued. Data is being loaded from NAND to page buffer until **whole data in DataRAM** is read. The read from the DataRAM can be only synchronous read mode. The status information related to load operation can be checked by the host if required. When host accesses DataRAM, the address of DataRAM must be a multiple of 4.

**Superload operation must be utilized within a same area partitioned as SLC or MLC.**

**Superload Operation Flow Chart Diagram**

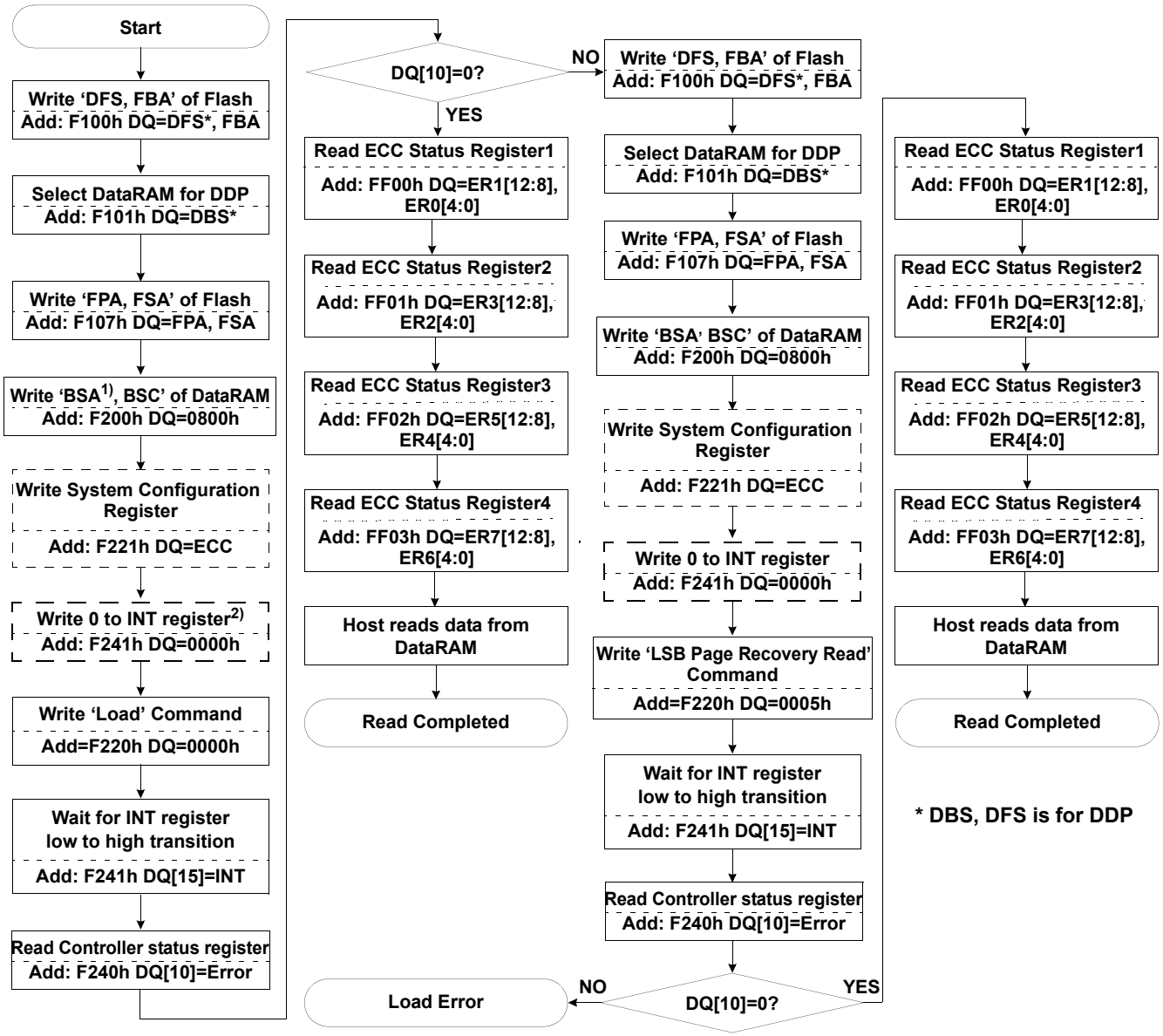



**NOTE :**
1) FSA must be 00 and BSC must be 000 always for Superload operation.
2) BSA must be 1000.
3) In case of Superload operation, the number of sectors to be loaded is 8 sectors.
4) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18
5) For the first load, hosts must issue 'Load(0000h)' command.
6) In case of Superload operation, only synchronous read mode is valid.
Host should read data out until end of DataRAM(804FH).
After Reading out the last data(Add:804F), Additional clock should not be asserted.

### 3.6.2 LSB Page Recovery Read

MLC NAND Flash cell has paired pages - LSB page and MSB page. LSB page has lower page address and MSB page has higher page address in paired pages. If power off occurs during MSB page program, the paired LSB page data can become corrupt. LSB page recovery read is a way to read LSB page though page data are corrupted. When uncorrectable error occurrs as a result of LSB page read after power up, issue LSB page recovery read. Its command is '0005h'. Flow chart below shows LSB page read sequence.

**LSB Page Recovery read flow chart**




**NOTE :**
1) BSA must be 1000.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

## 3.7 Read Operation

**See Timing Diagrams 6.1,6.2, 6.3 and 6.4.**

The device has two read modes; Asynchronous Read and Synchronous Burst Read.

The initial state machine automatically sets the device into the Asynchronous Read Mode (RM=0) to prevent the spurious altering of memory content upon device power up or after a Hardware reset. No commands are required to retrieve data in Asynchronous Read Mode.

The Synchronous Read Mode is enabled by setting RM bit of System Configuration1 Register (F221h) to Synchronous Read Mode (RM=1). See section 2.8.19 for more information about System Configuration1 Register.

### 3.7.1 Asynchronous Read Mode Operation (RM=0, WM=0)

**See Timing Diagrams 6.3 and 6.4**

In an Asynchronous Read Mode, data is output with respect to a logic input, $\overline{AVD}$.
Output data will appear on DQ15-DQ0 when a valid address is asserted on A15-A0 while driving $\overline{AVD}$ and $\overline{CE}$ to VIL. $\overline{WE}$ is held at VIH. The function of the $\overline{AVD}$ signal is to latch the valid address.

Address access time from $\overline{AVD}$ low (tAA) is equal to the delay from valid addresses to valid output data.

The Chip Enable access time (tCE) is equal to the delay from the falling edge of $\overline{CE}$ to valid data at the outputs.

The Output Enable access time (tOE) is the delay from the falling edge of $\overline{OE}$ to valid data at the output.

### 3.7.2 Synchronous Read Mode Operation (RM=1, WM=X)

**See Timing Diagrams 6.1and 6.2**

In a Synchronous Read Mode, data is output with respect to a clock input.

The device is capable of a continuous linear burst operation and a fixed-length linear burst operation of a preset length. Burst address sequences for continuous and fixed-length burst operations are shown in the table below.

**Burst Address Sequences**

| | Start Addr. | Burst Address Sequence(Decimal) | | | | |
|---|---|---|---|---|---|---|
| | | Continuous Burst | 4-word Burst | 8-word Burst | 16-word Burst | 32-word Burst |
| Wrap around | 0 | 0-1-2-3-4-5-6... | 0-1-2-3-0... | 0-1-2-3-4-5-6-7-0... | 0-1-2-3-4-....-13-14-15-0... | 0-1-2-3-4-....-29-30-31-0... |
| | 1 | 1-2-3-4-5-6-7... | 1-2-3-0-1... | 1-2-3-4-5-6-7-0-1... | 1-2-3-4-5-....-14-15-0-1... | 1-2-3-4-5-....-30-31-0-1... |
| | 2 | 2-3-4-5-6-7-8... | 2-3-0-1-2... | 2-3-4-5-6-7-0-1-2... | 2-3-4-5-6-....-15-0-1-2... | 2-3-4-5-6-....-31-0-1-2... |
| | . . | . . | . . | . . | . . | . . |

In the burst mode, the initial word will be output asynchronously, regardless of BRWL. While the following words will be determined by BRWL value.

The latency is determined by the host based on the BRWL bit setting in the System Configuration 1 Register. The default BRWL is 4 latency cycles. At clock frequencies of 40MHz or lower, latency cycles can be reduced to 3. BRWL can be set up to 7 latency cycles.

The BRWL registers in System Configuration 1 Register can be read during a burst read mode by using the $\overline{AVD}$ signal with the address F221h.

### 3.7.2.1 Continuous Linear Burst Read Operation

**See Timing Diagram 6.2**

**First Clock Cycle**

The initial word is output at tIAA after the rising edge of the first CLK cycle. The RDY output indicates the initial word is ready to the system by pulsing high. If the device is accessed synchronously while it is set to Asynchronous Read Mode, the first data can still be read out.

**Subsequent Clock Cycles**

Subsequent words are output (Burst Access Time from Valid Clock to Output) tBA after the rising edge of each successive clock cycle, which automatically increments the internal address counter.

**Terminating Burst Read**

The device will continue to output sequential burst data until the system asserts $\overline{CE}$ high, or $\overline{RP}$ low, wrapping around until it reaches the designated address (see section 2.7.3 for address map information). Alternately, a Cold/Warm/Hot Reset, or a $\overline{WE}$ low pulse will terminate the burst read operation.

**Synchronous Read Boundary**

| Division | Add.map(word order) | |
|---|---|---|
| BootRAM Main(0.5KW) | 0000h~01FFh | Not Supported |
| BufferRAM0 Main(1KW) | 0200h~05FFh | Not Supported |
| BufferRAM1 Main(1KW) | 0600h~09FFh | |
| Reserved Main* | 0A00h~7FFFh | |
| BootRAM Spare(16W) | 8000H~800Fh | Not Supported |
| BufferRAM0 Spare(32W) | 8010h~802Fh | Not Supported |
| BufferRAM1 Spare(32W) | 8030h~804Fh | Not Supported |
| Reserved Spare* | 8050h~8FFFh | |
| Reserved Register* | 9000h~EFFFh | |
| Register(4KW) | F000h~FFFFh | |

*** Reserved area is not available on Synchronous read**

**NOTE :**
Continuous burst read should be done, with in the address range of the selected buffer RAM, dataRAM0 or DataRAM1.

### 3.7.2.2 4-, 8-, 16-, 32-Word Linear Burst Read Operation

**See Timing Diagram 6.1**

An alternate Burst Read Mode enables a fixed number of words to be read from consecutive address.

The device supports a burst read from consecutive addresses of 4-, 8-, 16-, and 32-words with a linear-wrap around. When the last word in the burst has been reached, assert $\overline{CE}$ and $\overline{OE}$ high to terminate the operation.

In this mode, the start address for the burst read can be any address of the address map with one exception. The device does not support a 32-word linear burst read on the spare area of the BufferRAM.

### 3.7.2.3 Programmable Burst Read Latency Operation

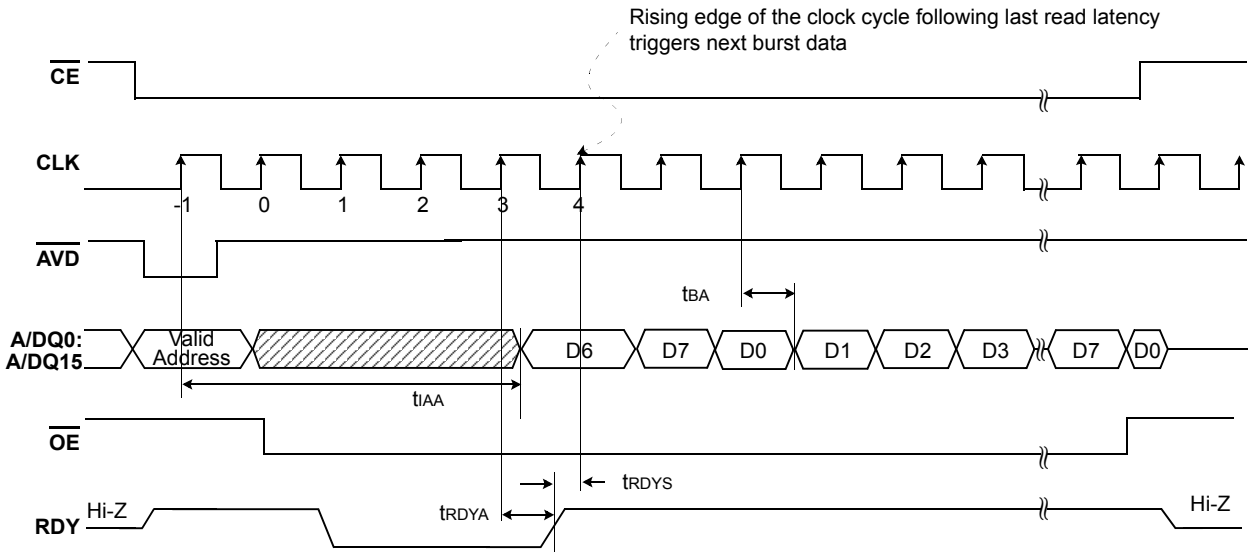
**See Timing Diagrams 6.1 and 6.2**

Upon power up, the number of initial clock cycles from Valid Address ($\overline{AVD}$) to initial data defaults to four clocks.

The number of clock cycles (n) which are inserted after the clock which is latching the address. The host can read the first data with the (n+1)th rising edge.

The number of total initial access cycles is programmable from three to seven cycles. After the number of programmed burst clock cycles is reached, the rising edge of the next clock cycle triggers the next burst data.

**Four Clock Burst Read Latency (BRWL=4 case)**




***NOTE :**
BRWL=4, HF=0 is recommended for 40MHz~66MHz. For frequency over 66MHz, BRWL should be 6 or 7 while HF=1.
Also, for frequency under 40MHz, BRWL can be reduced to 3, and HF=0.

### 3.7.3 Handshaking Operation

The handshaking feature allows the host system to simply monitor the RDY signal from the device to determine when the initial word of burst data is ready to be read.

To set the number of initial cycles for optimal burst mode, the host should use the programmable burst read latency configuration (see section 2.8.19, "System Configuration1 Register").

The rising edge of RDY which is derived at one cycle prior of data fetch clock indicates the initial word of valid burst data.

### 3.7.4 Output Disable Mode Operation

When the $\overline{CE}$ or $\overline{OE}$ input is at $V_{IH}$, output from the device is disabled.
The outputs are placed in the high impedance state.

## 3.8 Synchronous Write(RM=1, WM=1)

**See Timing Diagram 6.6, 6.7 and 6.8**

Burst mode operations enable high-speed synchronous read and write operations. Burst operations consist of a multi-clock sequence that must be performed in an ordered fashion. After $\overline{CE}$ goes low, the address to access is latched on the next rising edge of clk that $\overline{ADV}$ is low. During this first clock rising edge, $\overline{WE}$ indicates whether the operation is going to be a read ($\overline{WE}$ = high) or write ($\overline{WE}$ = low). The size of a burst can be specified in the BL as either a fixed length or continuous. Fixed-length bursts consist of 4, 8, 16, and 32 words. Continuous burst write has the ability to start at a specified address and burst within the designated DataRAM. The latency count stored in the BRWL defines the number of clock cycles that elapse before the initial data value is transferred between the processor and Flex-MuxOneNAND device. The RDY output will be asserted as soon as a burst is initiated, and will be de-asserted to indicate when data is to be transferred into (or out of) the memory. The processor can access other devices without incurring the timing penalty of the initial latency for a new burst by suspending burst mode. Bursts are suspended by stopping clk. clk can be stopped high or low. Note that the RDY output will continue to be active, and as a result no other devices should directly share the RDY connection to the controller.
To continue the burst sequence, clk is restarted after valid data is available on the bus.

Same as the normal burst mode, the latency is determined by the host based on the BRWL bit setting in the System Configuration 1 Register. The default BRWL is 4 latency cycles. At clock frequencies of 40MHz or lower, latency cycles can be reduced to 3, at frequency range from 40MHz to 66MHz, latency cycle should be over 4. Over clock frequency of 66MHz, latency cycle should be over 6.

For BufferRAMs, both ‘Start Initial Burst Write’ and ‘Burst Write’ is supported. (Refer to Chapter 3.2) However, for Register Access, only ‘Start Initial Burst Write’ is supported. Therefore, Synchronous Burst Write on Register is prohibited.(Refer to Chapter 3.2 and 6.8)

## 3.9 Program Operation

**See Timing Diagrams 6.11**

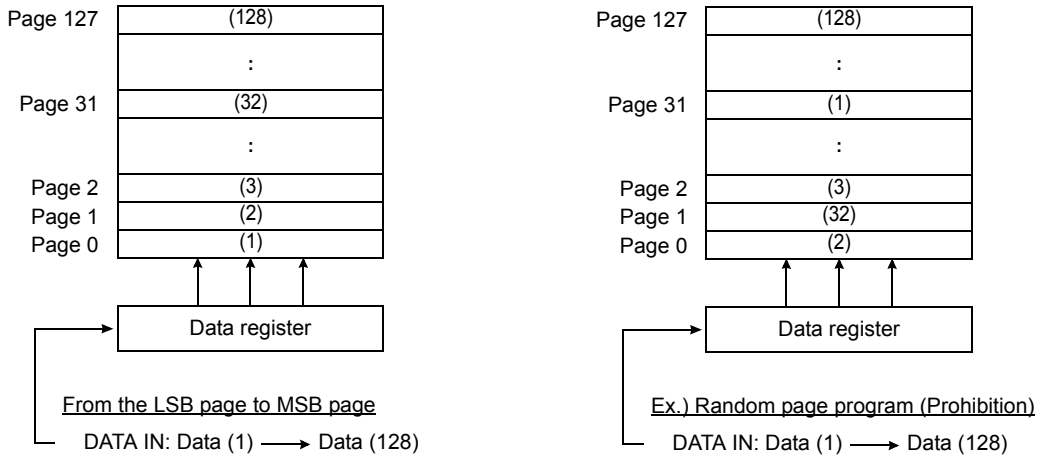
The Program operation is used to program data from the on-chip BufferRAMs into the NAND FLASH memory array.

The device has two 2KB data buffers, 1 Page (4KB + 128B) in size. A page has 8 sectors of 512B each main area and 16B spare area. The device can be programmed in units of 8 sectors at once.

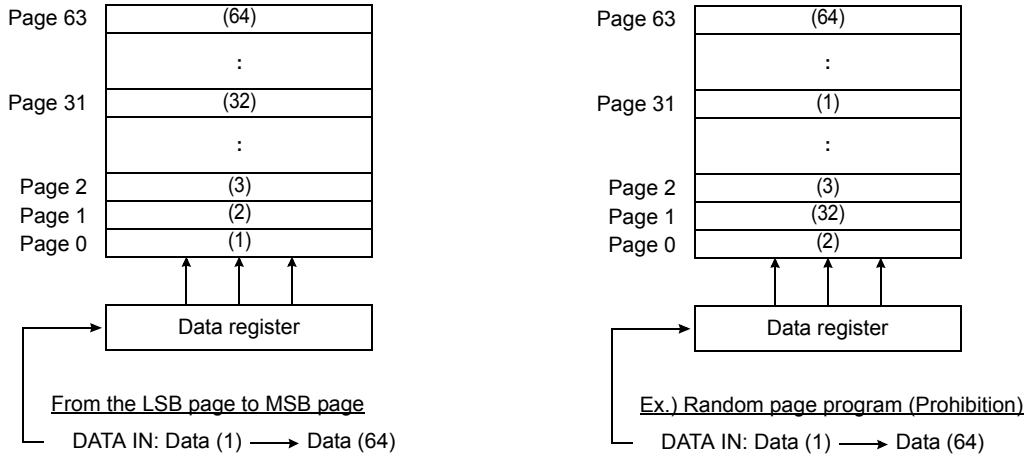
**Addressing for program operation**

Within a block, the pages must be programmed consecutively from the LSB (least significant bit) page of the block to MSB (most significant bit) pages of the block. Random page address programming is prohibited. Once users start to write data on a certain page, the page is a LSB page, therefore LSB page does not have to always be a page 0.

**MLC Block**




**SLC Block**



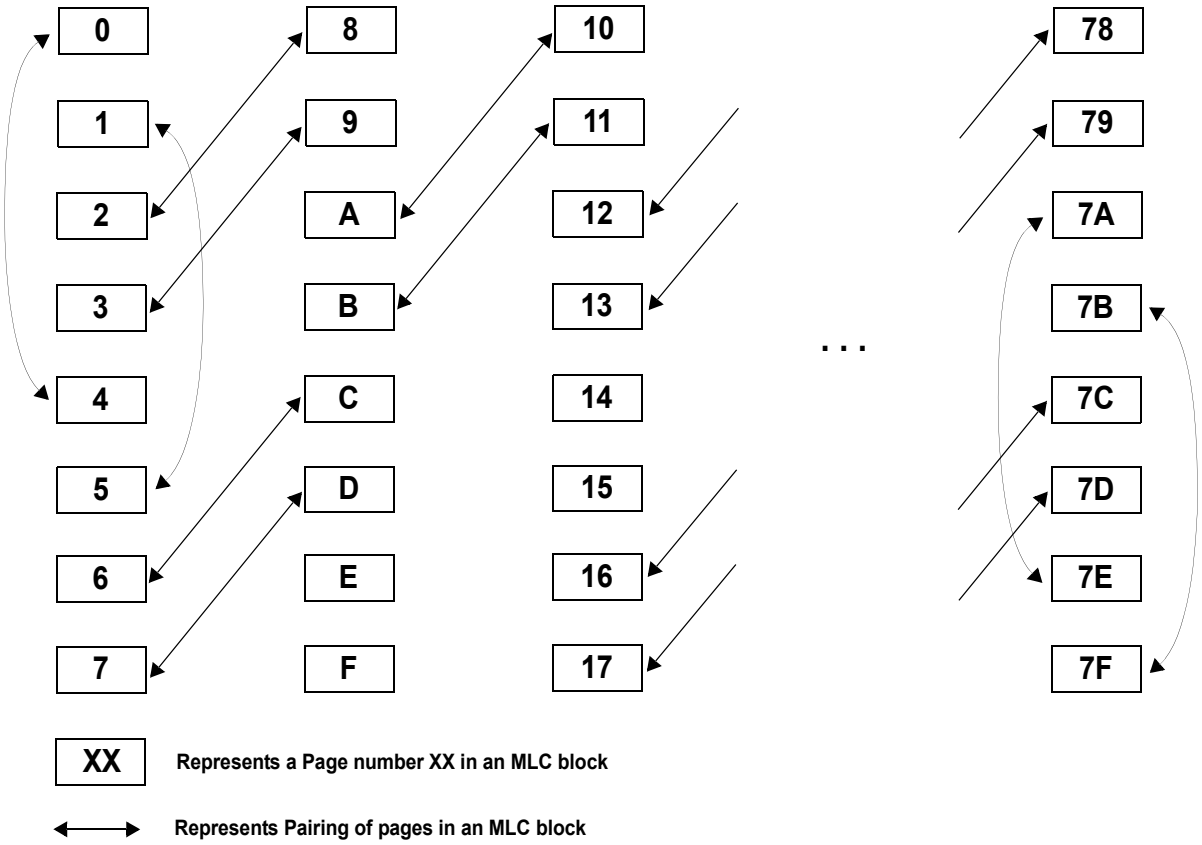

**NOTE :**
The figure explains the order of page programming in a block. (x) indicates that the corresponding page is the Xth page to be written in the block.

**Paired Page Address Information**

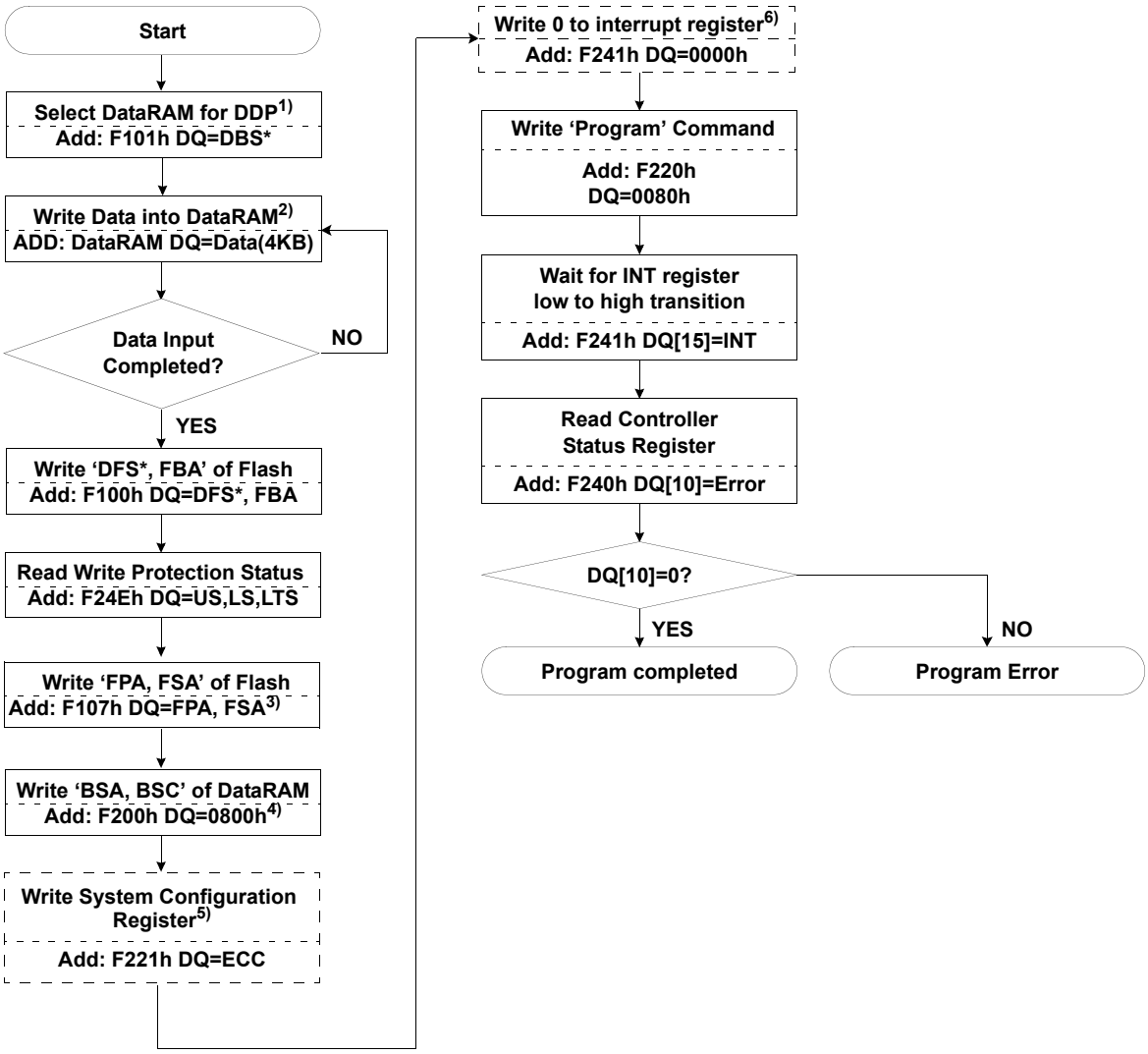
In case of MLC partition, when Program, Cache Program, Interleave cache program, Copy-back with random data in operations are abnormally aborted(eg. power-down), not only page data under program but also paired page data may be damaged.

| Paired Page Address | | Paired Page Address | |
|---|---|---|---|
| 00h | 04h | 01h | 05h |
| 02h | 08h | 03h | 09h |
| 06h | 0Ch | 07h | 0Dh |
| 0Ah | 10h | 0Bh | 11h |
| 0Eh | 14h | 0Fh | 15h |
| 12h | 18h | 13h | 19h |
| 16h | 1Ch | 17h. | 1Dh |
| 1Ah | 20h | 1Bh | 21h |
| 1Eh | 24h | 1Fh | 25h |
| 22h | 28h | 23h | 29h |
| 26h | 2Ch | 27h | 2Dh |
| 2Ah | 30h | 2Bh | 31h |
| 2Eh | 34h | 2Fh | 35h |
| 32h | 38h | 33h | 39h |
| 36h | 3Ch | 37h | 3Dh |
| 3Ah | 40h | 3Bh | 41h |
| 3Eh | 44h | 3Fh | 45h |
| 42h | 48h | 43h | 49h |
| 46h | 4Ch | 47h | 4Dh |
| 4Ah | 50h | 4Bh | 51h |
| 4Eh | 54h | 4Fh | 55h |
| 52h | 58h | 53h | 59h |
| 56h | 5Ch | 57h | 5Dh |
| 5Ah | 60h | 5Bh | 61h |
| 5Eh | 64h | 5Fh | 65h |
| 62h | 68h | 63h | 69h |
| 66h | 6Ch | 67h | 6Dh |
| 6Ah | 70h | 6Bh | 71h |
| 6Eh | 74h | 6Fh | 75h |
| 72h | 78h | 73h | 79h |
| 76h | 7Ch | 77h | 7Dh |
| 7Ah | 7Eh | 7Bh | 7Fh |

**Pairing of pages in MLC block:**



**Program Operation Flow Diagram**




*** DBS, DFS is for DDP**

**(*) : If program operation results in an error, map out the block including the page in error and copy the target data to another block.**

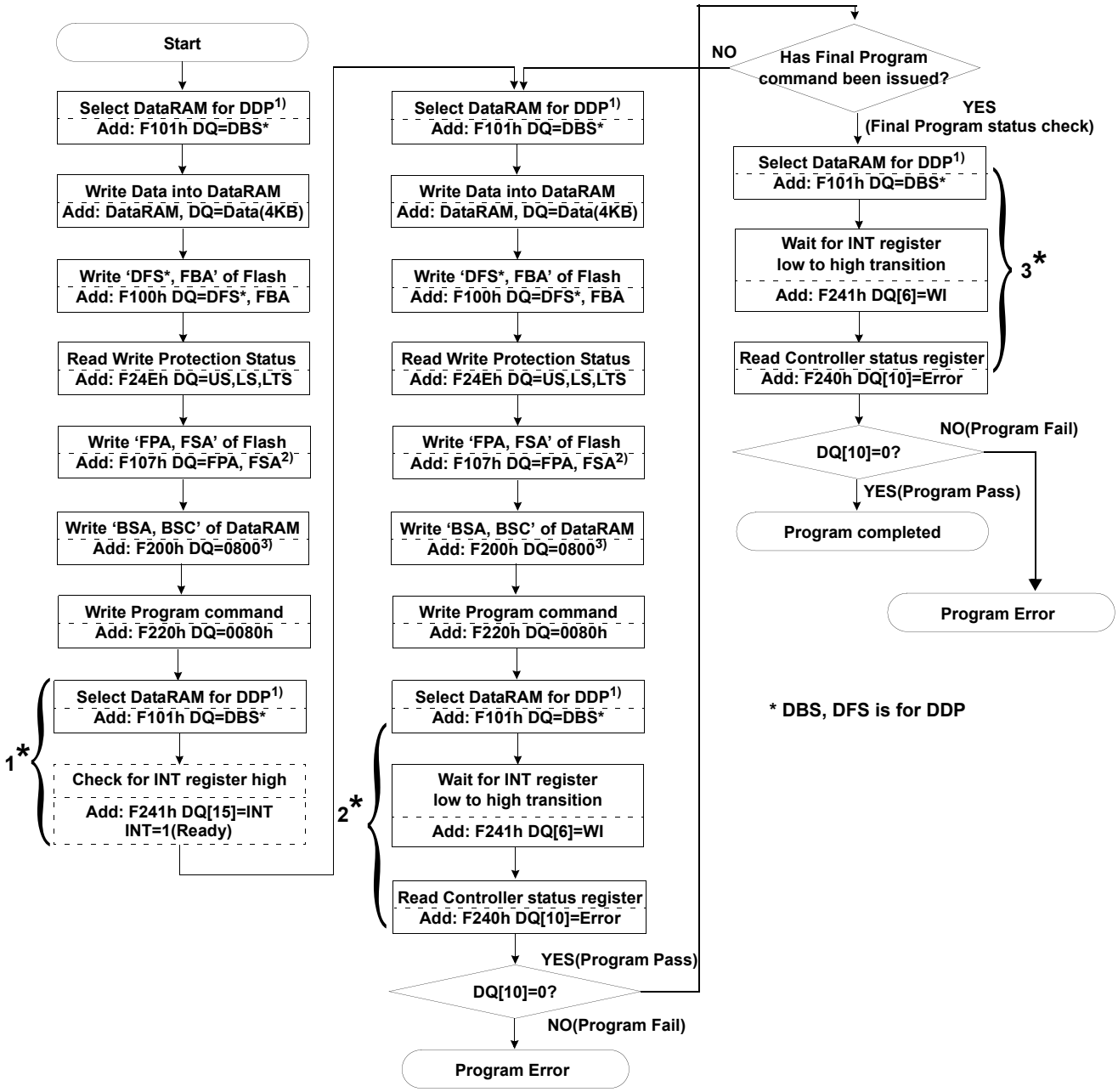
**NOTE :**
1) DBS must be set before data input.
2) Data input could be done anywhere between "Start" and "Write Program Command".
3) FSA must be 00 within program operation.
4) BSA must be 1000 and BSC must be 000.
5) Writing System Configuration Register is optional.
6) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

During the execution of the Internal Program Routine, the host is not required to provide any further controls or timings. Furthermore, all commands, except a Reset command, will be ignored. A reset during a program operation will cause data corruption at the corresponding location.

If a program error is detected at the completion of the Internal Program Routine, map out the block, including the page in error, and copy the target data to another block. An error is signaled if DQ10 = "1" of Controller Status Register(F240h).

If Power off occurs during a Program operation, the page that is being programmed might be corrupted. Data from paired pages may be affected.

**Program Interleave(@DDP) Flow Chart**




1* Check the chip status before command issues.

2* Previous Program Status Check
DBS must be changed to indicate chip. Program has been issued prior to current program ongoing

3* Final Program Status Check

(*) Program Interleave can work in Auto INT Mode.
Interrupt register must not be written.

**NOTE :**
1) DBS must be set before data input.
2) FSA must be 00 and BSC must be 000 within program operation
3) BSA must be 1000 and BSC must be 000.

### 3.9.1 Cache Program Operation

**See Timing Diagram 6.12**

The Cache Program is to enhance the performance of Program Operation. Employing Cache Program operation, transfer time from Host to DataRAM can be shadowed, therefore write performance will increase.

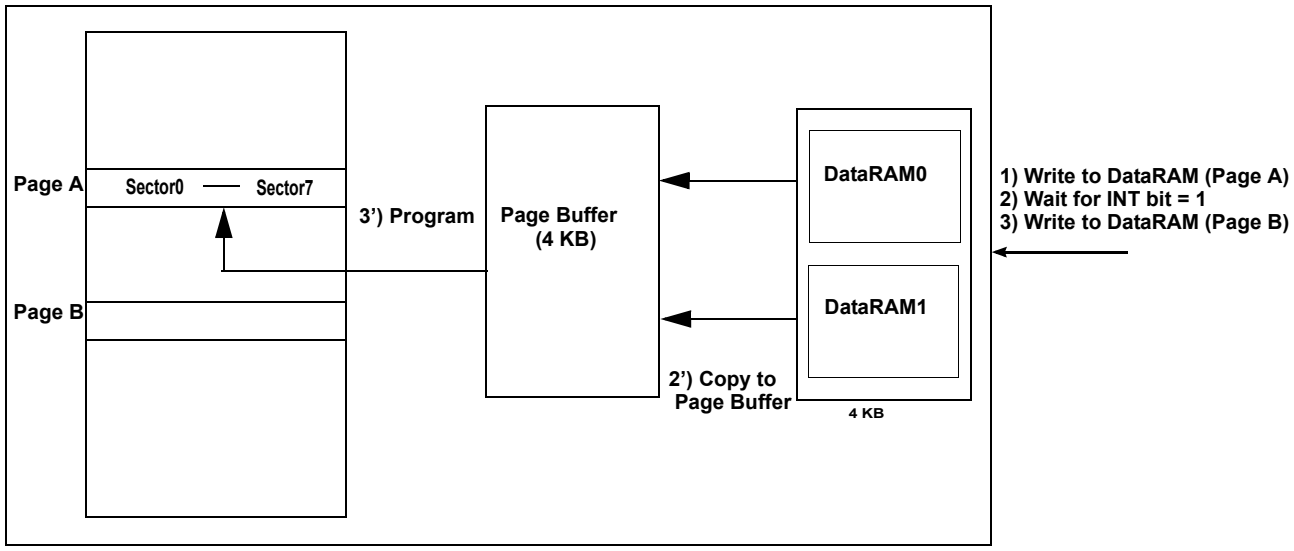
In Cache Program, since 4KB data is to be programmed into NAND Flash Array in another advanced way.

1. 4KB Data write from host to DataRAMs.
2. Cache Program command issue. This will turn INT pin to busy state[1)], OnGo bit sets to '1'.
   (Note that before issuing 'Cache Program Command', host should make sure that the target blocks are unlocked.)
3. 4KB data will be sequentially transferred to a page buffer in NAND Flash Array.
4. When this transfer operation is complete, programming into NAND Flash Array will automatically start, and at the same time, INT bit will turn to '1' to indicate that DataRAMs are now ready to be written with next 4KB data.
5. When second 4KB is written to two DataRAMs, another Cache Program command is issued and INT bit will go to '0'[1)].

If host wants to program data less than 8 sectors, unwanted area to be programmed must be written to all '1's.
When INT bit goes to '1' after second data transfer from DataRAMs to Page Buffers are complete, user may check the Status Register to check the Cache program status. During Cache Program, Error bit shows the status of previous program operation.

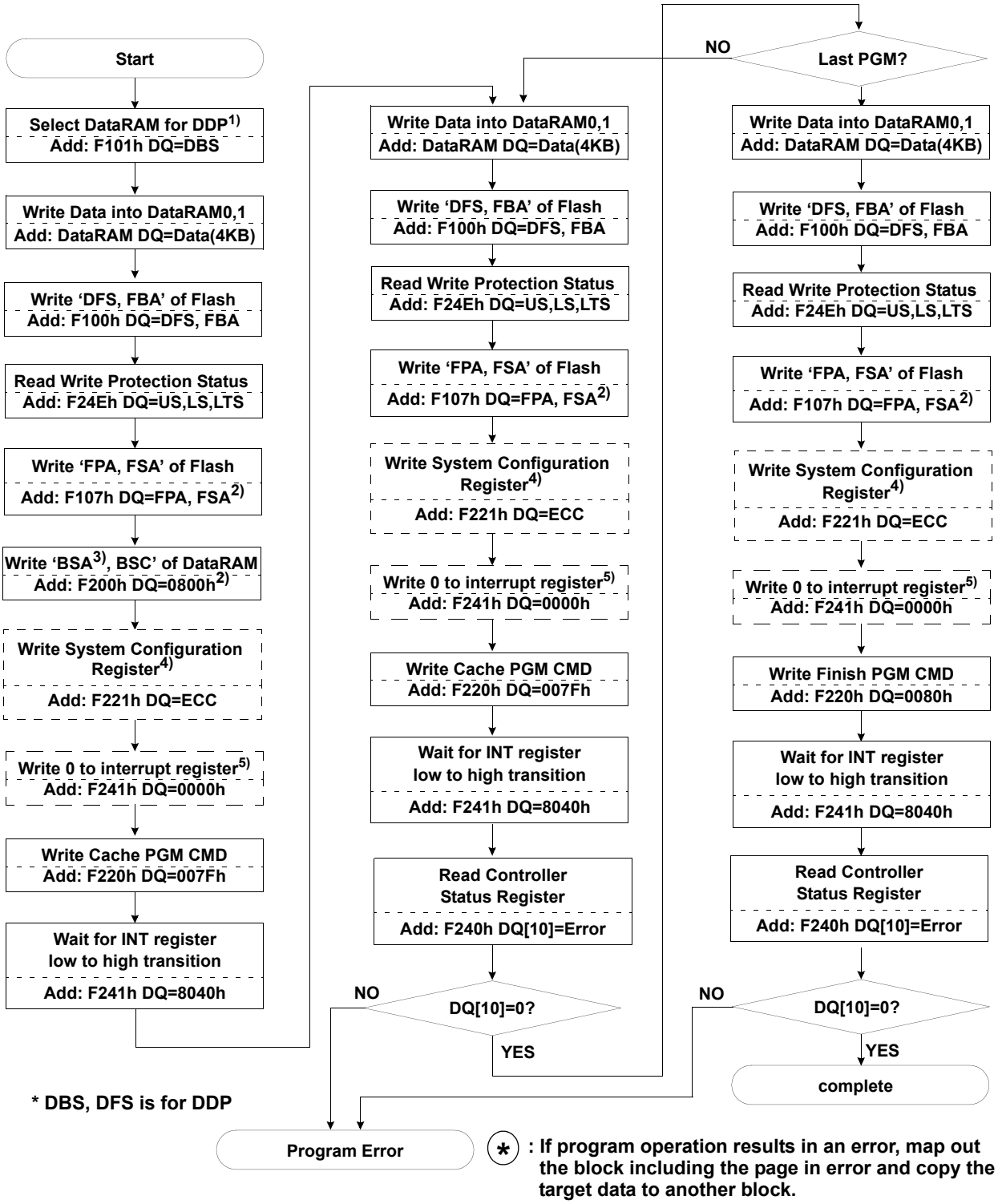
For the final 4KB program of Cache Program scheme, host should issue Program Command(0080h). And when the final page is programmed, INT bit will turn to '1' and OnGo status bit - which indicates the overall Cache Program ongoing status - will go to '0'. At the completion of Cache Program operation, Error bit will show the pass/fail status overall status of program, and previous ~ current bit will show where the error occurred accordingly (Refer to the below diagram.)

Note that Cache Program command cannot be performed on OTP block and 1st block OTP.
Cache Program operation must be utilized within a same area partitioned as SLC or MLC.




**NOTE :**
2 and 2' are concurrent; 3 and 3' are concurrent

**Cache Program Operation Flow Diagram**




**NOTE :**
1) DBS must be set before data input.
2) FSA must be 00 and BSC must be 000 within program operation.
3) BSA must be 1000.
4) Writing System Configuration Register is optional.
5) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1.

### 3.9.2 Interleave Cache Program Operation

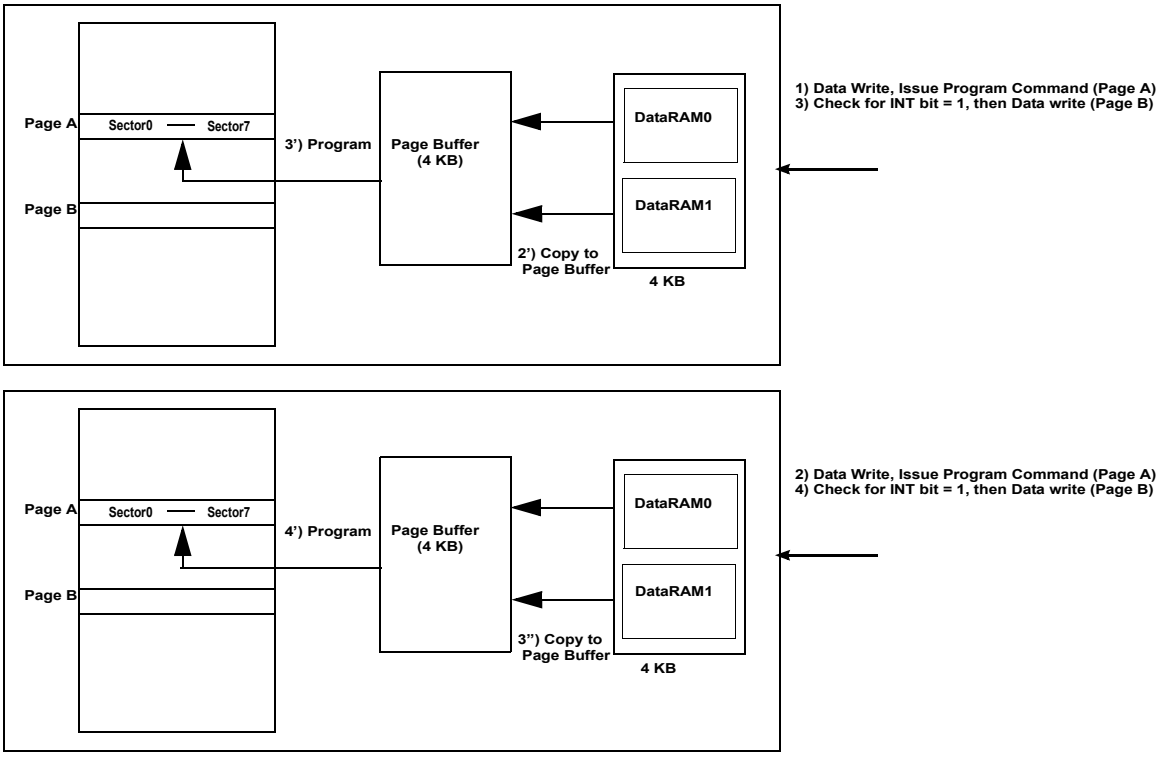
The Interleave Cache Program is available only on DDP. Host can write data on a chip while programming another chip with this operation.

Interleave Cache Program is executing as following:
1. 4KB Data are written from host to DataRAMs in Chip1.
2. Cache Program command issue. This will turn INT bit to busy state[1)], OnGo bit sets to '1'.
   (Note that before issuing 'Interleave Cache Program Command', host should make sure that the target blocks are unlocked.)
3. 4KB data will be sequentially transferred to each page buffer in NAND Flash Array.
4. While these data are transferring, Host can write another 4KB Data to DataRAM in Chip2.
5. When the transfer operation is completed, programming into NAND Flash Array will automatically start, and at the same time, INT bit will turn to '1' to indicate that DataRAMs are now ready to be written with next 4KB data.
6. Second 4KB is writable on Chip1 when INT1 goes to '1'.
7. When second 4KB is written to two DataRAMs of Chip1, another Cache Program command is issued and INT1 bit will go to '0'[1)] again.

When INT bit goes to '1' after second data transfer from DataRAMs to Page Buffers are complete, user may check the Status Register to check the Cache program status. During Cache Program, previous bit shows the status of previous program operation.

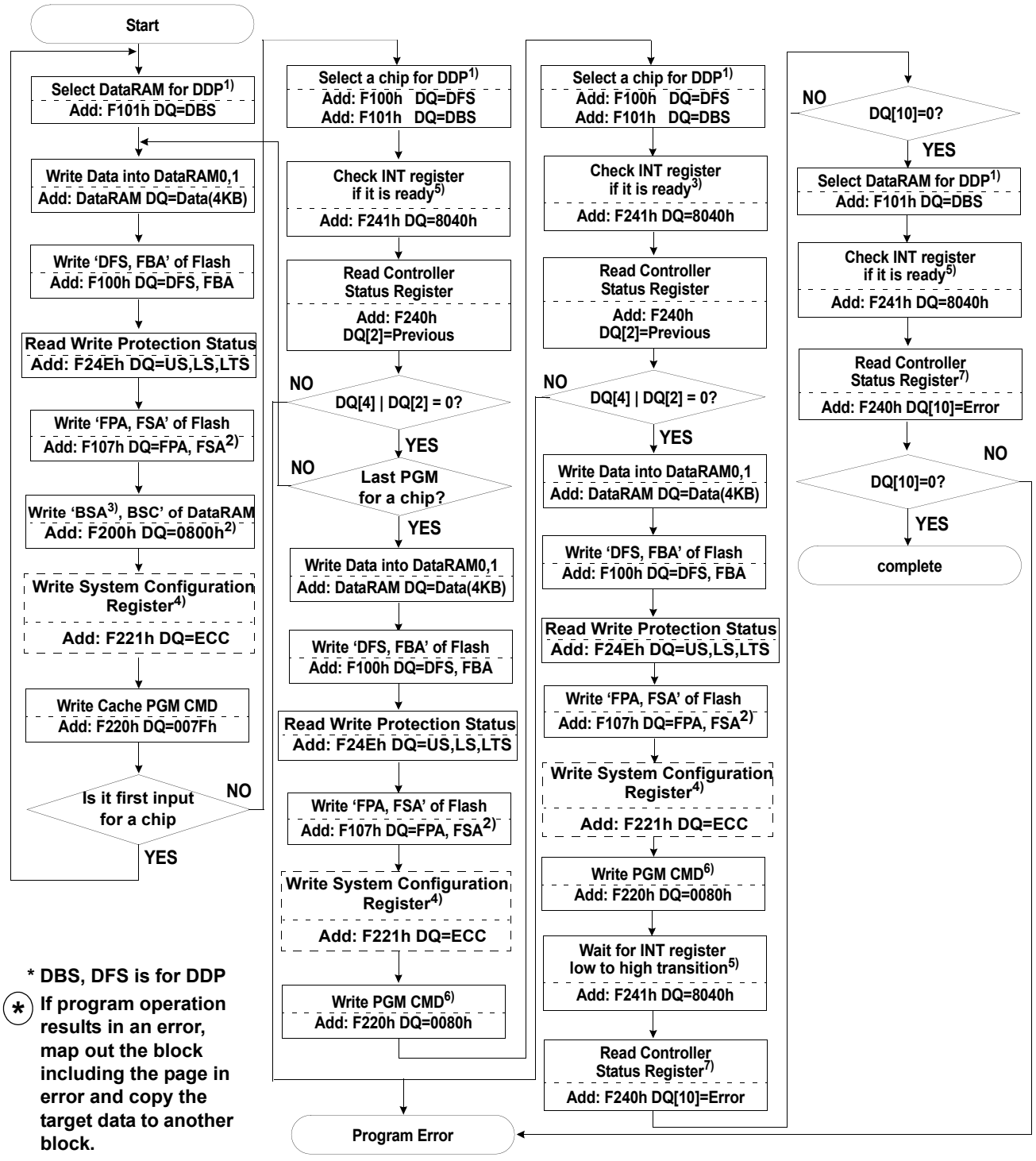
For the final 4KB program of Interleave Cache Program scheme, host should issue Program Command(0080h) on each chip. If host issues 0080h on only a chip, another chip will be on operation as it is not finished. Ongo status bit will show the ongoing status of each chip. Its operation is same as Cache Program operation on each chip. Error bit will show the pass/fail status of each chip of Interleave Cache program, and previous ~ current bit will show where the error occurred accordingly .

Note that OTP block and 1st block OTP cannot be Interleave Cache Programmed.
**Interleave Cache Program operation must be utilized within a same area partitioned as SLC or MLC.**




**NOTE :**
2 and 2' are concurrent; 3, 3' and 3'' are concurrent; 4 and 4' are concurrent.

**Interleave Cache Program Operation Flow Diagram**



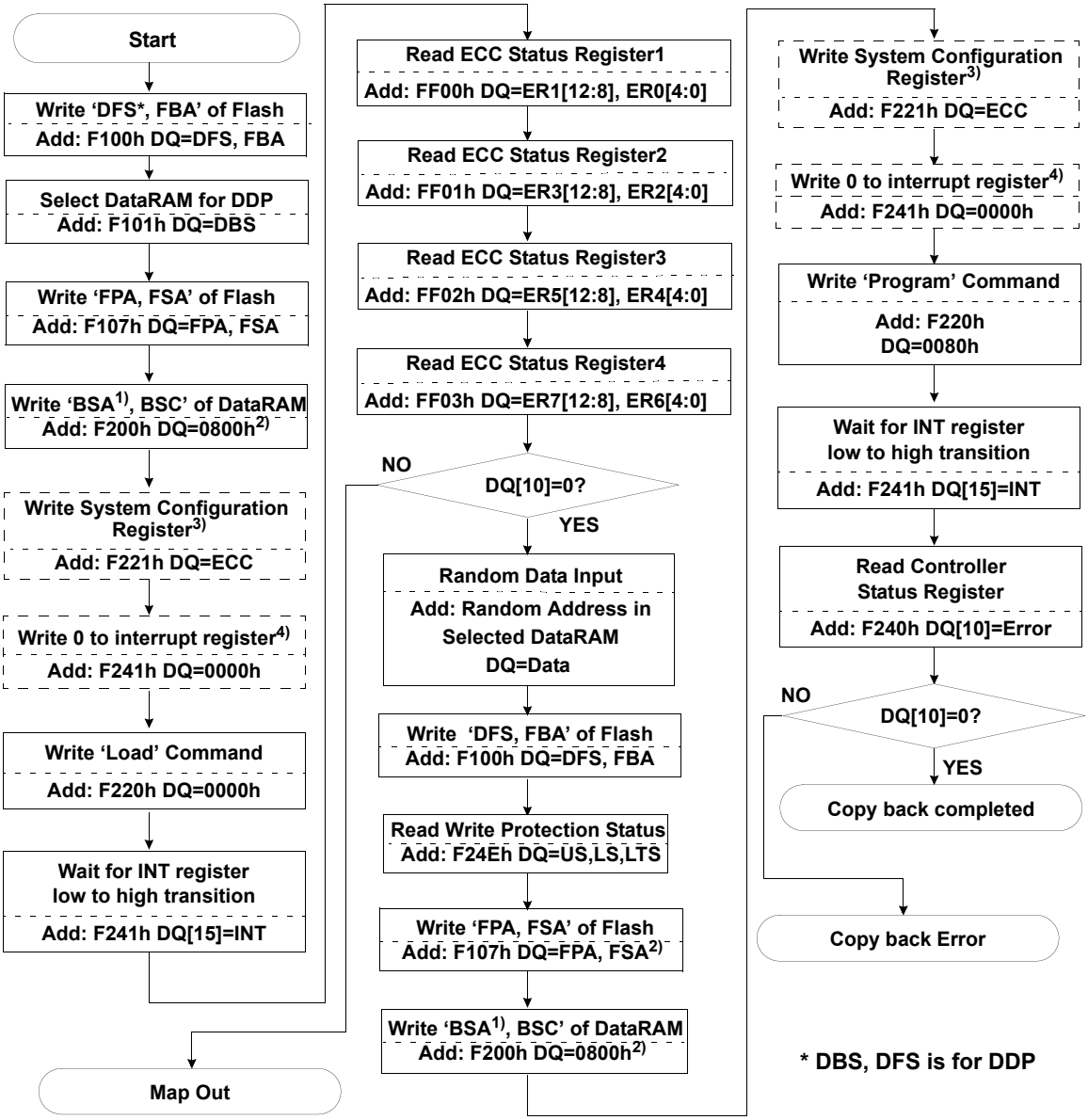

**NOTE :**

1) DBS must be set before data input.
2) FSA must be 00 and BSC must be 000 within program operation.
3) BSA must be 1000.
4) Writing System Configuration Register is optional.
5) Host is strongly recommended to see the INT register(F241h) of each chip.
6) Once 'PGM command' is issued onto a chip, the same command(PGM) must be issued onto another chip. If not, Samsung cannot gurantee the following operation.
7) If error bit is set at this step, DQ[1]~[4] shoulde be checked in order to find where the error occurred.

## 3.10 Copy-Back Program Operation with Random Data Input

The Copy-Back Program Operation with Random Data Input in Flex-MuxOneNAND consists of 3 phases, Load data into DataRAM, Modify data and program into designated page. Data from the source page is saved in one of the on-chip DataRAM buffers and modified by the host, then programmed into the destination page.
As shown in the flow chart, data modification is possible upon completion of load operation. ECC is also available at the end of load operation. Therefore, using hardware ECC of Flex-MuxOneNAND, accumulation of 4 bit error can be avoided.
Copy-Back Program Operation with Random Data Input will be effectively utilized at modifying certain bit, byte, word, or sector of source page to destination page while it is being copied.

**Copy-Back Program Operation with Random Data Input Flow Chart**




**NOTE :**
1) BSA must be 1000.
2) FSA must be 00 and BSC must be 000 within program operation.
3) Writing System Configuration Register is optional.
4) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

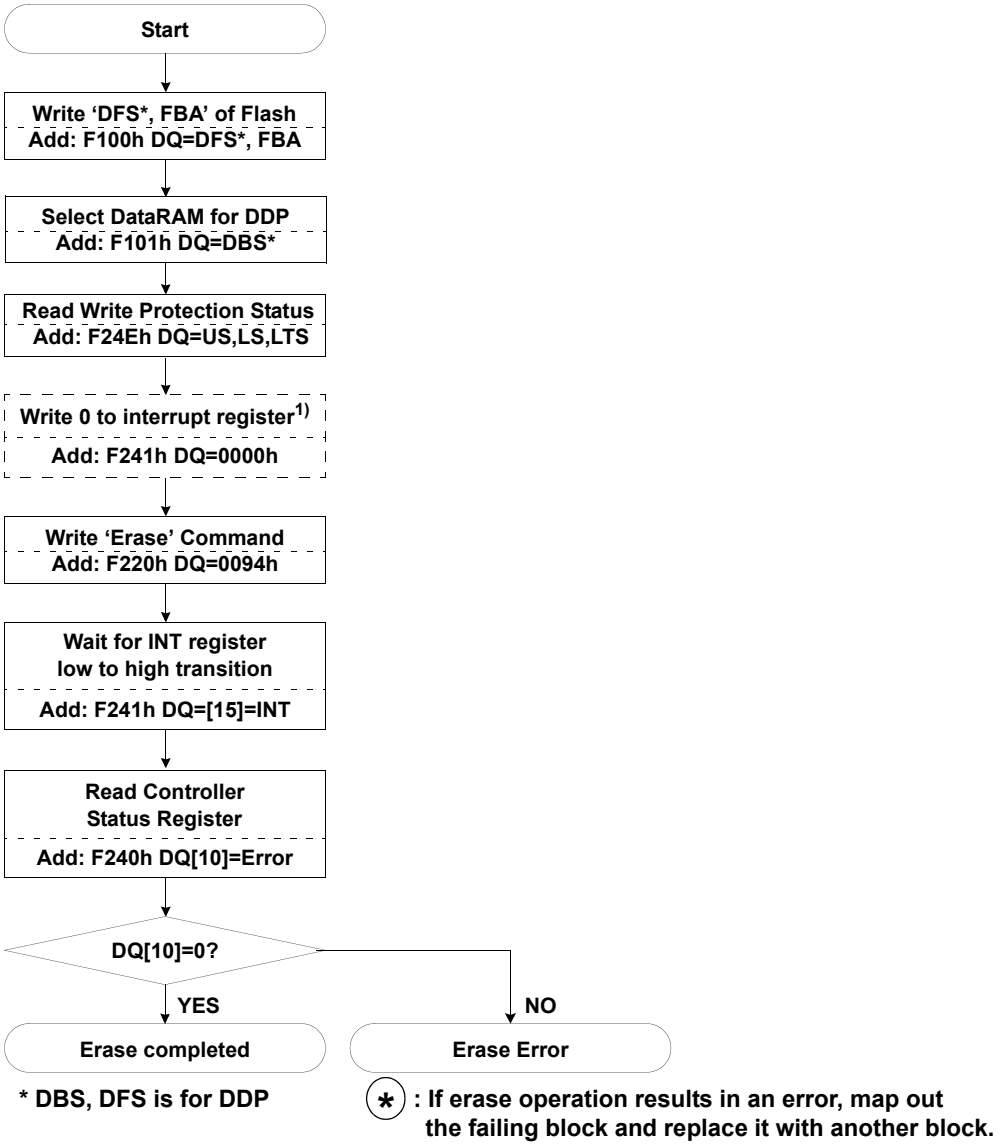
## 3.11 Erase Operation

### 3.11.1 Block Erase Operation

**See Timing Diagram 6.14**

The device can be erased one block at a time. To erase a block is to write all 1's into the desired memory block by executing the Internal Erase Routine. All previous data is lost.

**Block Erase Operation Flow Chart**



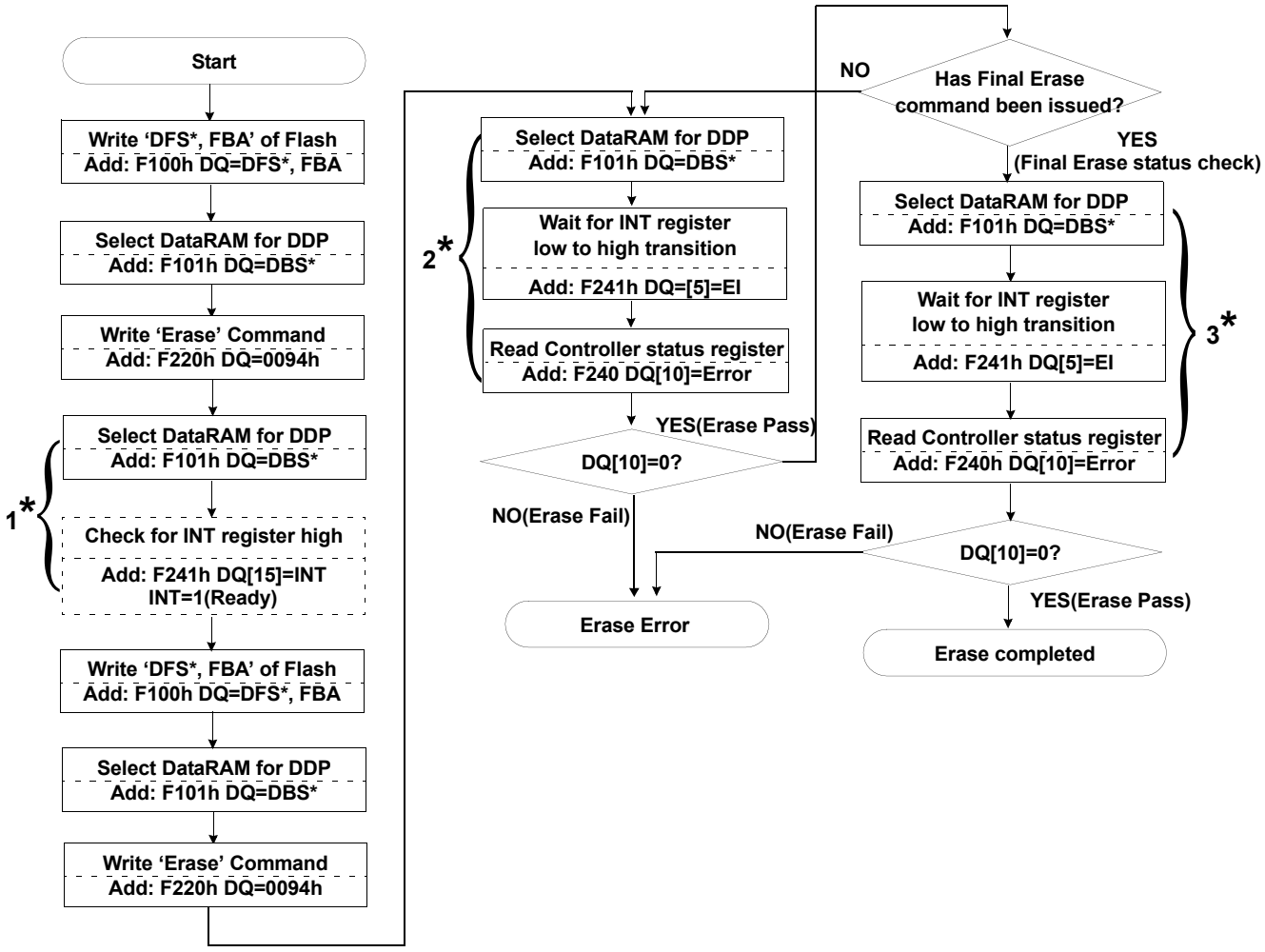

**NOTE :**
1) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

In order to perform the Internal Erase Routine, the following command sequence is necessary.

- The Host selects Flash Core of DDP chip.
- The Host sets the block address of the memory location.
- The Erase Command initiates the Internal Erase Routine. During the execution of the Routine, the host is not required to provide further controls or timings. During the Internal erase routine, all commands, except the Reset command and Erase Suspend Command, written to the device will be ignored.

A reset or power off during an erase operation will cause data corruption at the corresponding location Block.

**Erase Interleave[1] (@DDP) Flow Chart**




(*) **Erase Interleave can work in Auto INT Mode.**
**Interrupt register must not be written.**

**1* Check the chip status before command issues.**

**2* Previous Erase Status Check**
**DBS must be changed to indicate chip. Erase has been issued prior to current erase ongoing**

**3* Final Erase Status Check**

**NOTE :**
1) Erase Suspend and Erase Resume Operations are not supported in Erase Interleave(@DDP).

### 3.11.2 Erase Suspend / Erase Resume Operation

The Erase Suspend/Erase Resume Commands interrupt and restart a Block Erase operation so that user may perform another urgent operation on the block that is not being designated by Erase Operation.

**Erase Suspend During a Block Erase Operation**
When Erase Suspend command is written during a Block Erase operation, the device requires a maximum of 500us to suspend erase operation. Erase Suspend Command issue during Block Address latch sequence is prohibited.

After the erase operation has been suspended, the device is ready for the next operation including a load, program, Lock, Unlock, Lock-tight, Hot Reset, NAND Flash Core Reset, Command Based Reset, or OTP Access.

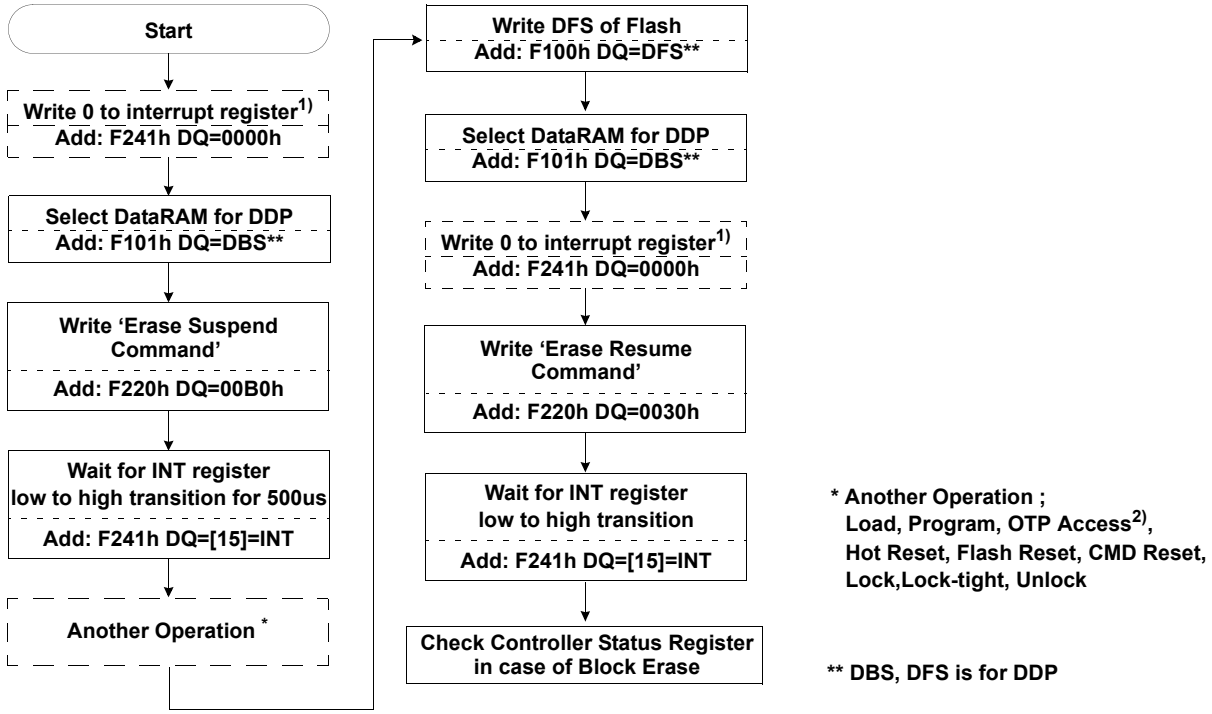
The subsequent operation can be to any block that was NOT being erased.

A special case arises in Erase Suspend operation pertaining to the OTP. A Reset command is used to exit from the OTP Access mode. If the Reset-triggered exit from the OTP Access Mode happens after an Erase Suspend Operation, the erase routine could fail. Therefore to exit from the OTP Access Mode without causing the erase suspend/resume operation to fail, a 'NAND Flash Core Reset' command should be issued.

For the duration of the Erase Suspend period the following commands are not accepted:

- Block Erase/Erase Suspend

**Erase Suspend and Erase Resume Operation Flow Chart**




**NOTE :**
1) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
2) If OTP access mode exit happens with Reset operation during Erase Suspend mode, Reset operatin could hurt the erase operation. So if a user wants to exit from OTP access mode without the erase operation stop, Reset NAND Flash Core command should be used.

**Erase Resume**
When the Erase Resume command is executed, the Block Erase will restart. The Erase Resume operation does not actually resume the erase, but starts it again from the beginning.

When an Erase Suspend or Erase Resume command is executed, the addresses are in Don't Care state.

## 3.12 Partition Information (PI) Block (SLC Only)

One Block of the SLC NAND Flash Array memory is reserved for Partition Information (PI) Block.

The block can be read, programmed and erased using the same operations as any other NAND Flash Array memory block.
Only Load, Erase and Program can be performed. PI Block is not able to cover with internal ECC Engine in OneNAND, so it has to be accessed under ECC off mode.

PI block is guaranteed to be a valid block up to 1K program/erase cycles.

**Entering the PI Block**
The PI block is separately accessible from the rest of the NAND Flash Array by using the PI Access command instead of the Flash Block Address (FBA).

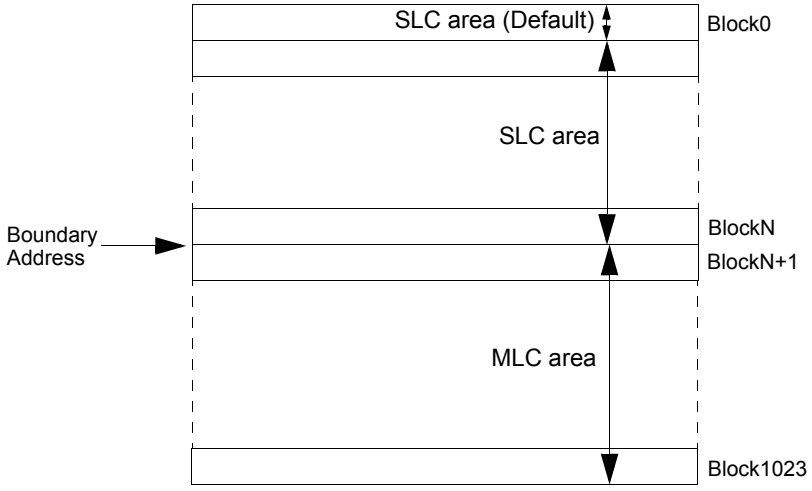
**Exiting the PI Block**
To exit the PI Access Mode, a Cold-, Warm-, Hot-, or NAND Flash Core Reset operation is performed.

**Exiting the PI Block during an Erase Operation**
If the Reset-triggered exit from the PI Access Mode happens after during an Erase Suspend Operation, the erase routine could fail. Therefore to exit from the PI Access Mode without causing the erase operation to fail, a 'NAND Flash Core Reset' command should be issued.

**PI Block Page Allocation Information**
This is located at the 1st word of sector0 of page0 of main area in the Block.
The allocated word in 1st page is programed with data FC00h initially after shipment, whole block is set as MLC except Block 0.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lock | | Reserved(1111) | | | | Boundary Address (end of SLC area) | | | | | | | | | |

**PI Lock bits and Boundary Address.**
PI Block can be locked only by programming lock bits into [15:14] of the 1st word of sector0, page0 of the main memory area of PI.
The first block is a SLC block. The MLC block will be defined from the next block of the block designated by boundary address programmed into[9:0] of the 1st word of sector0, page0 of the main memory area of PI.




[NAND Flash Array]

### 3.12.1 PI Block Boundary Information setting

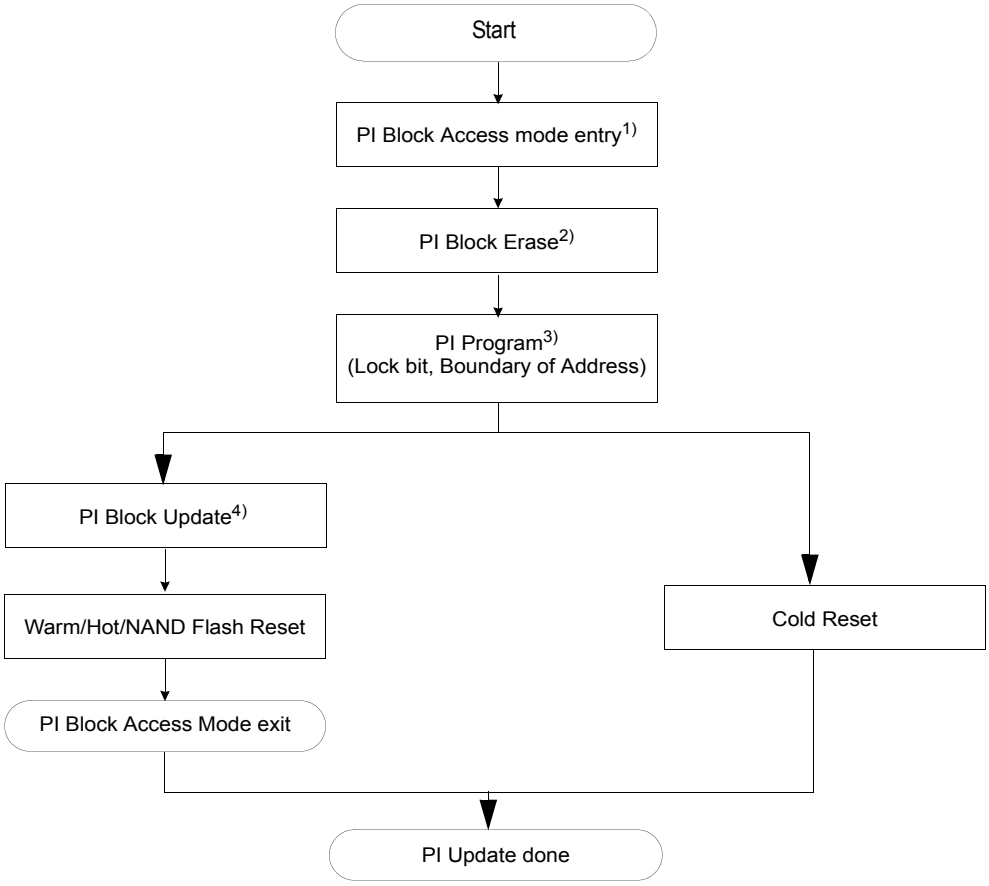
It is 1st word of sector0 of page0 of main area of PI Block. The Lock bits for PI Block and Boundary address of SLC and MLC are stored. After shipment, it is initially programmed as data FC00h(Lock bit[15:14]: 11b(binary), Boundary address[9:0]: 000h).
To change PI Block contents (i.e, lock bits and boundary address), Erase/Program sequence should be followed as below.

**PI Block Boundary Information setting steps**

- Enter PI Block access mode(Refer to Chapter 3.12.1.1).
- Issue PI Block erase(Refer to Chapter 3.12.1.2).
- Issue PI program(Refer to Chapter 3.12.1.3).
- Exit PI Block Access mode & Update new Partition Information.

PI block Access mode exit can be done through a Warm/Cold/Hot/NAND Flash Reset.
However, PI Update can only be done by two methods: PI Update Command and Cold Reset.
The following flow chart shows two methods for updating the PI and exiting PI access mode.

**PI Block Boundary Information setting Flow Chart**



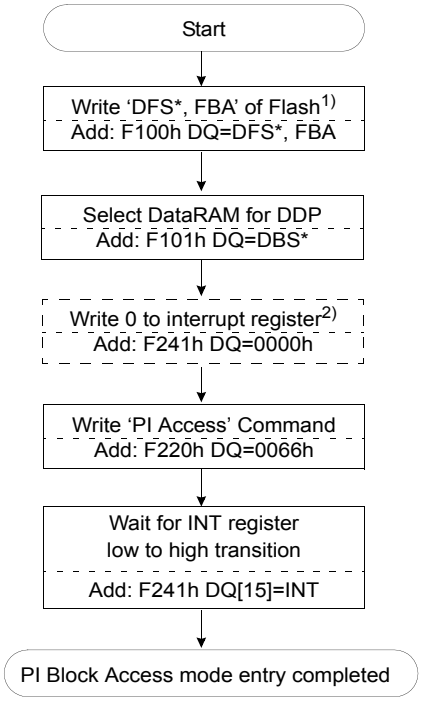

**NOTE :**
1) Refer to Chapter 3.12.1.1
2) Refer to Chapter 3.12.1.2
3) Refer to Chapter 3.12.1.3
4) Refer to Chapter 3.12.1.4

### 3.12.1.1 PI Block Access mode entry

The PI area is a separate part of the NAND Flash Array memory. It is accessed by issuing PI Access command(66h) instead of writing a Flash Block Address(FBA) in the StartAddress1 register.

After being accessed through the PI Access Command, the contents of PI memory area can be programmed, erased or loaded using the same operations as a normal program, erase or load operation to the NAND Flash Array memory.

**PI Block Access mode entry Flow Chart**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

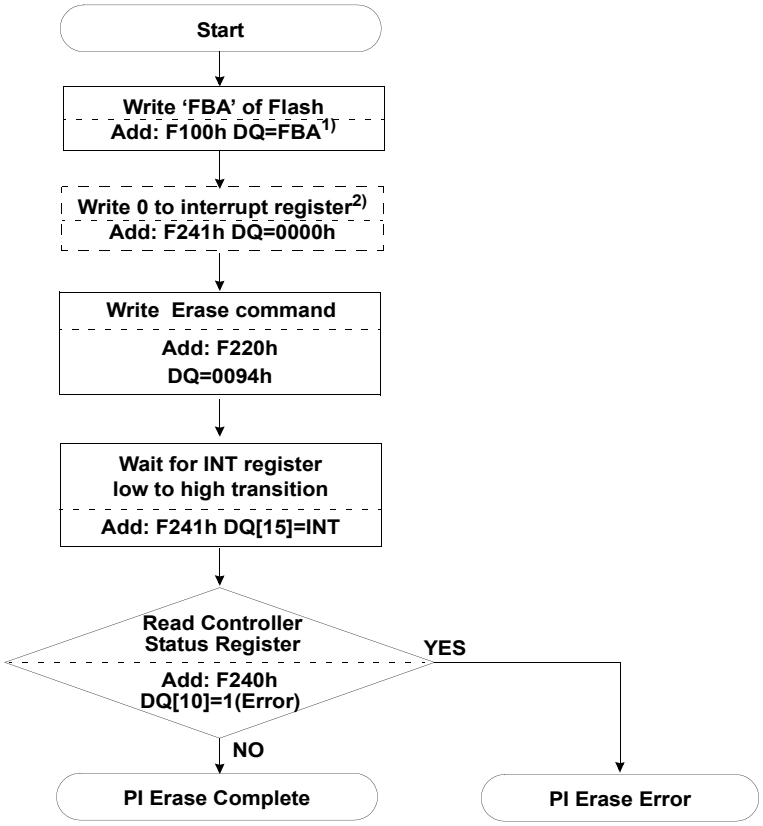
### 3.12.1.2 PI Block Erase

The PI Block Erase Operation erases the entire PI block including Partition Information. PI Block Access mode entry must be done before issuing Erase operation for PI Block.

**Erasing the PI Area**

- Issue the PI Access Command(Refer to Chapter 3.12.1.1).
- Issue an Erase command to erase the PI area.

**PI Block Erase Operation Flow Chart (In PI Block Access Mode)**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) must be 0000h.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

### 3.12.1.3 PI Block Program Operation

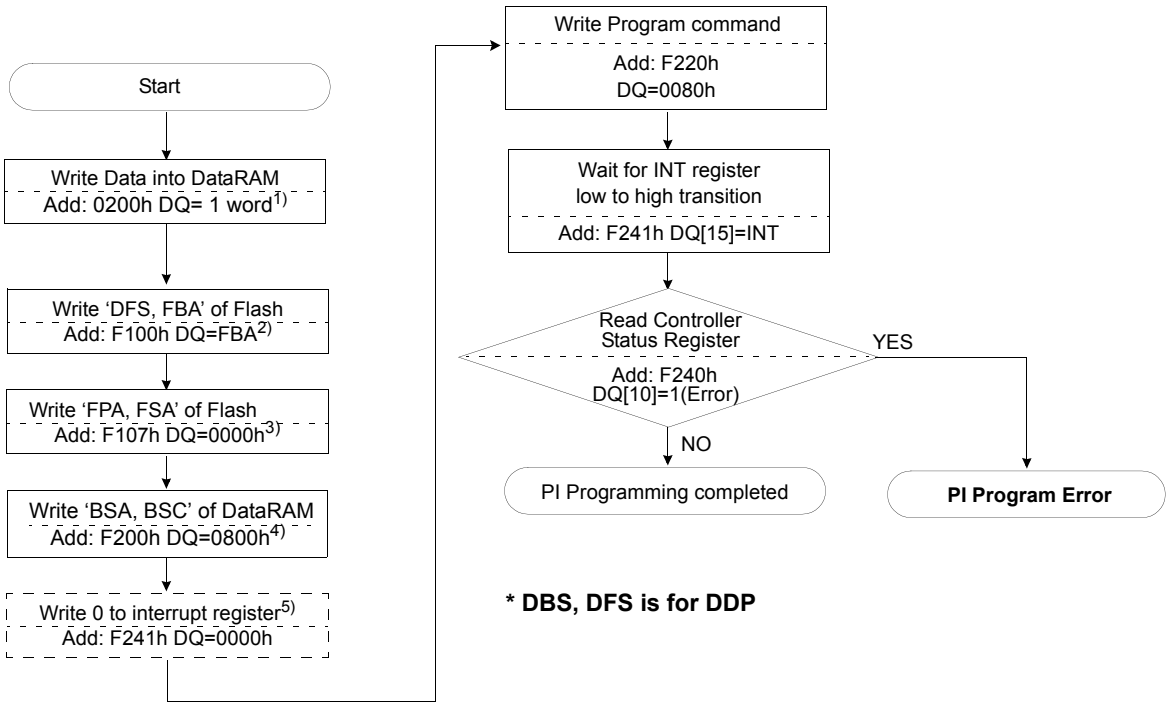
The PI Block Program Operation accesses the PI area and programs content from the DataRAM on-chip buffer to the designated page(s) of the PI.

**A memory location in the PI area can be program.**
The PI area is programmed using the same sequence as normal program operation after being accessed by the PI Block Access mode entry command (see section 3.8 for more information).

**Programming the PI Area**

- Issue the PI Access Command(Refer to Chapter 3.12.1.1).
- Write data into the DataRAM
  - In case of PI Lock(Add: 0200h, DQ=3XXXh, The lower 10 bits[9:0] are boundary address) .
  - In case of PI Unlock(Add: 0200h, DQ=FXXXh, The lower 10 bits[9:0] are boundary address).
- Write 0000h into Flash Block Address (FBA), that is address of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the PI.

**PI Block Program Operation Flow Chart (In PI Block Access Mode)**




**Locking the PI**
Programming to PI block can be prevented by locking the PI area. Locking the PI area is accomplished by programming 3XXXh to 1st word of sector0 of main of the page0 memory area in the PI block(XXXh out of 3XXXh is a boundary block address that ends SLC area).
Once Lock bits are programmed as lock status, PI block will be protected from program and erase. Boundary address is alterable before PI block is locked, but it is not recommended.
At device power-up and PI Update operation, this word is updated internally. If 3XXXh is found(i.e. the status of PI is locked), Program/Erase operations to PI block result in an error and the device updates the Error Bit of the Controller Status Register as "1"(fail).

**NOTE :**
1) Only the 1st word of 1st page of PI block (PI block Boundary Information) can be programmed in PI Block. The rest of the block cannot be programmed.
2) FBA(NAND Flash Block Address) must be 0000h.
3) FPA must be 00h and FSA must be 00.
4) BSA must be 1000 and BSC must be 000.
5) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1

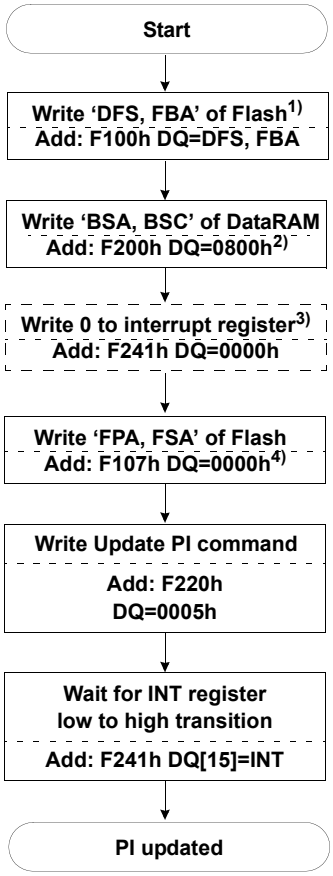
### 3.12.1.4 PI Update

Once new partition information is programmed into the PI block, an internal register that is invisible to users must be updated for the changes in PI to be applied. This internal register which stores partition information(i.e. the last address of SLC area and lock bits) will be automatically updated through cold reset. However, the internal register can also be updated by issuing Partition Information Update command(05h) after PI Access mode entry.

**Update the PI Area**

- Issue the PI Access mode(Refer to Chapter 3.12.1.1).
- Issue the update PI command.

**PI Block Update (In PI Block Access Mode)**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) must be 0000h.
2) BSA must be 1000 and BSC must be 000.
3) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
4) FPA must be 00h and FSA must be 00.

### 3.12.2 PI Block Load Operation

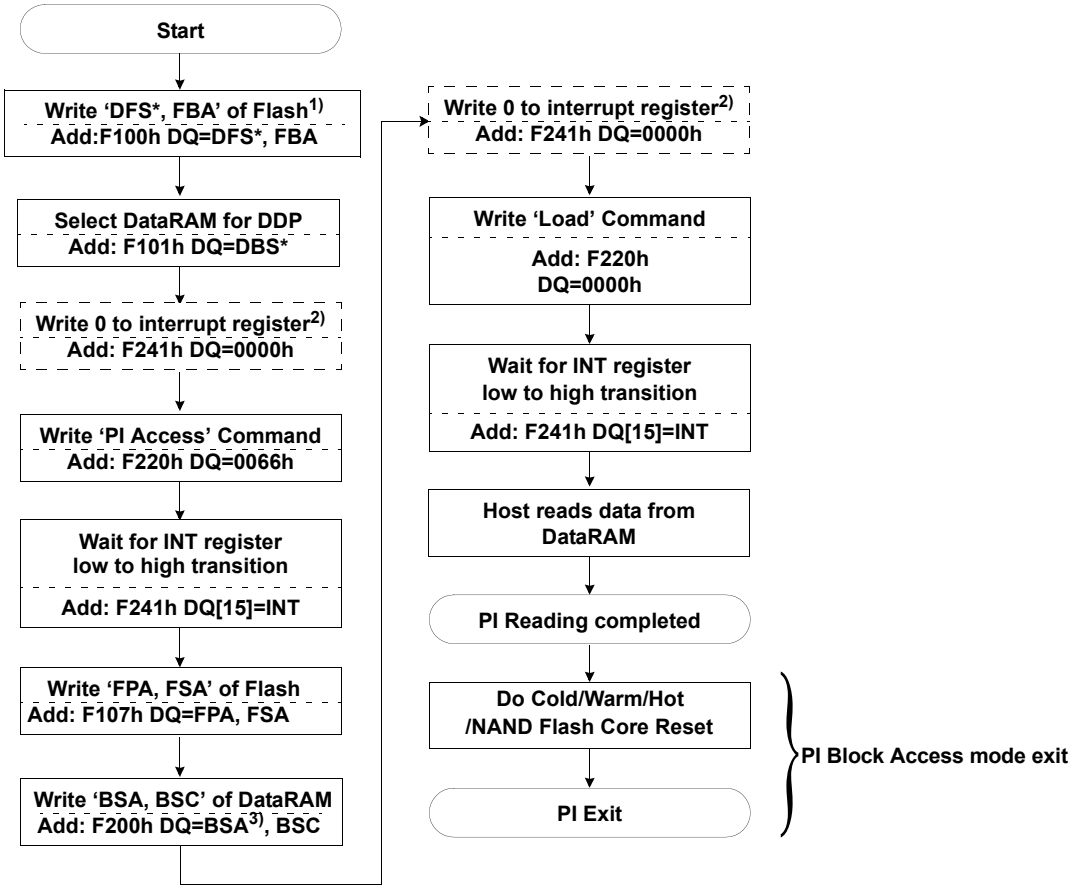
A PI Block Load Operation accesses the PI area and transfers identified content from the PI to the DataRAM on-chip buffer, thus making the PI contents available to the Host.

The PI area is a separate part of the NAND Flash Array memory. It is accessed by issuing PI Access command(66h).

After being accessed with the PI Access Command, the contents of PI memory area are loaded using the same operations as a normal load operation to the NAND Flash Array memory (see section 3.6 for more information).

To exit the PI access mode after an PI Block Load Operation, a Cold-, Warm-, Hot-, or NAND Flash Core Reset operation is performed.

**PI Block Read Operation Flow Chart (In PI Block Access Mode)**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) BSA must be 1000.

## 3.13 OTP Operation (SLC only)

One Block of the NAND Flash Array memory is reserved as a One-Time Programmable Block memory area.
Also, 1st Block of NAND Flash Array can be used as OTP.
OTP area and 1st block OTP area must be utilized as a SLC block.

The OTP block can be read, programmed and locked using the same operations as any other NAND Flash Array memory block.
OTP block cannot be erased. Note that Cache program and Finish Cache program cannot be performed on OTP and 1st Block OTP area.

OTP block is fully-guaranteed to be a valid block by an internal ECC engine.

**Entering the OTP Block**
The OTP block is separately accessible from the rest of the NAND Flash Array by using the OTP Access command instead of the Flash Block Address (FBA).

**Exiting the OTP Block**
To exit the OTP Access Mode, a Cold-, Warm-, Hot-, or NAND Flash Core Reset operation is performed.

**Exiting the OTP Block during an Erase Operation**
If the Reset-triggered exit from the OTP Access Mode happens during an Erase Suspend Operation, the erase routine could fail. Therefore to exit from the OTP Access Mode without suspending the erase operation, a 'NAND Flash Core Reset' command should be issued.

**The OTP Block Page Assignment**
OTP area is one block size (128KB+4KB, 64 Pages) and is divided into two areas. The 50-page User Area is available as an OTP storage area. The 14-page Manufacturer Area is programmed by the manufacturer prior to shipping the device to the user.

**OTP Block Page Allocation Information**

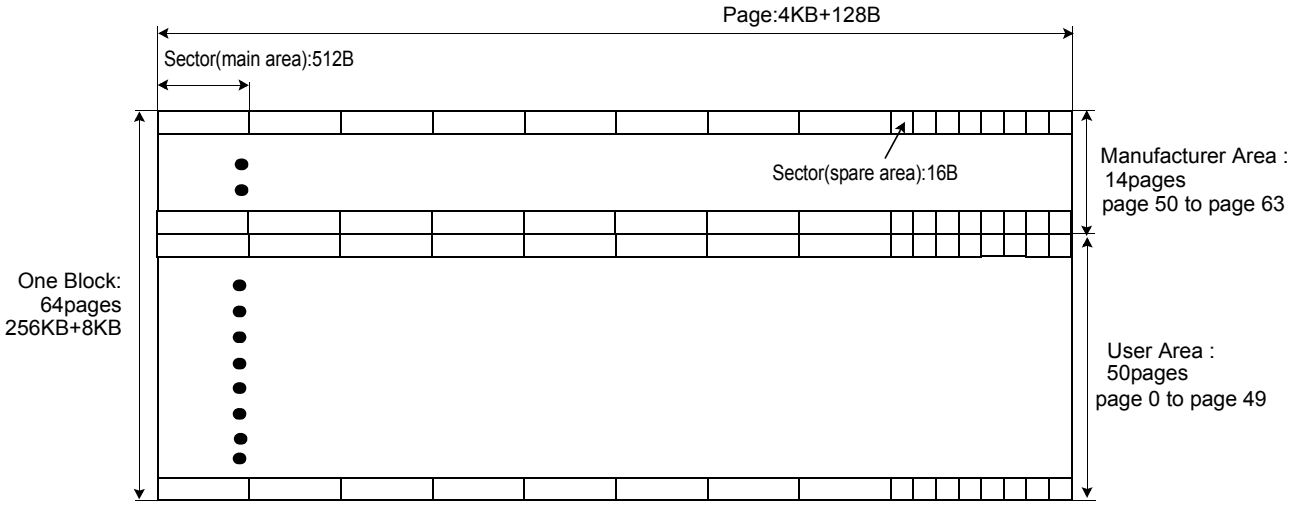
| Area | Page | Use |
|---|---|---|
| User | 0 ~ 49 (50 pages) | Designated as user area |
| Manufacturer | 50 ~ 63 (14 pages) | Used by the device manufacturer |

**Three Possible OTP Lock Sequence (Refer to Chapter 3.13.3~3.13.5 for more information)**
Since OTP Block and 1st Block OTP can be locked only by programming into 1st word of sector4, page49 of the main memory area of OTP, OTP Block and 1st Block OTP lock sequence is restricted into three following cases.

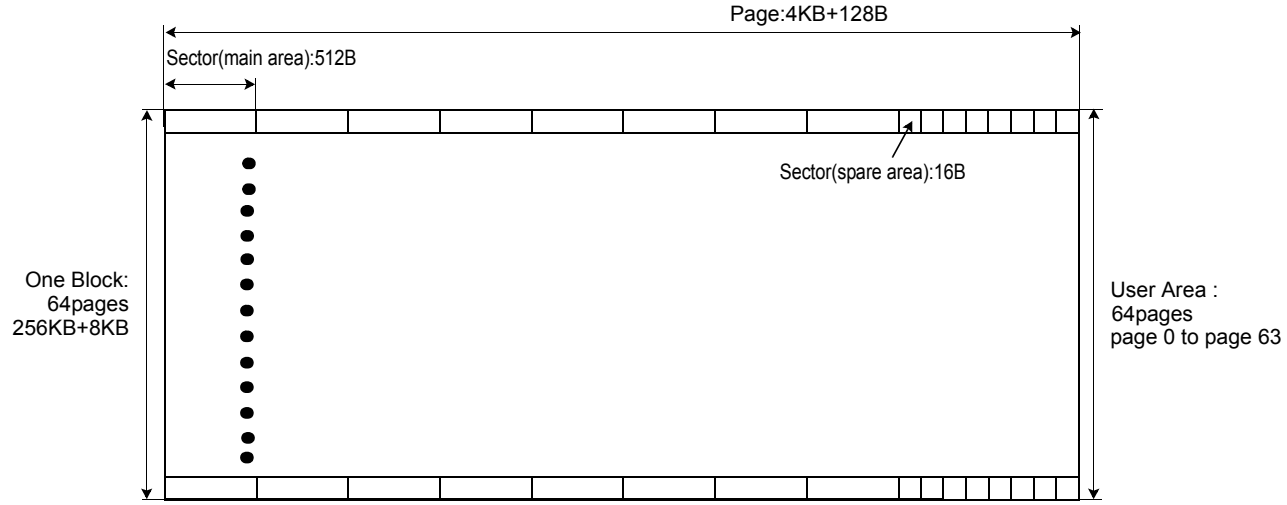
Note that user should be careful, because locking OTP Block before locking 1st Block OTP will disable locking 1st Block OTP.

1. OTP Block Lock Only :
   Once the OTP Block is locked, 1st Block OTP Lock is impossible.
2. 1st Block OTP Lock Only:
   Locking 1st Block OTP does not lock the OTP block, but the OTP Block Lock cannot be performed thereafter.
3. OTP Block Lock and 1st Block OTP Lock simultaneously:
   This simultaneous operation can be done by programming into 1st word of sector4, page49 of the main memory area of OTP.

**OTP Block Area Structure**




**1st Block OTP Area Structure**



### 3.13.1 OTP Block Load Operation

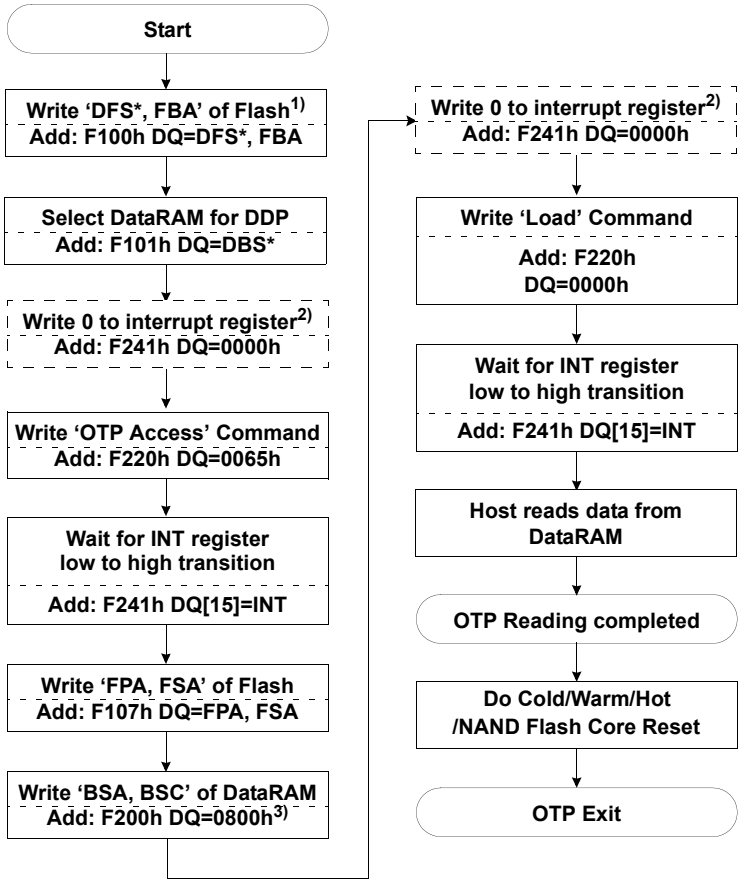
An OTP Block Load Operation accesses the OTP area and transfers identified content from the OTP to the DataRAM on-chip buffer, thus making the OTP contents available to the Host.

The OTP area is a separate part of the NAND Flash Array memory. It is accessed by issuing OTP Access command(65h) instead of a Flash Block Address (FBA) value in Start Address1 Register.
.

After being accessed with the OTP Access Command, the contents of OTP memory area are loaded using the same operations as a normal load operation to the NAND Flash Array memory (see section 3.6 for more information).

To exit the OTP access mode after an OTP Block Load Operation, a Cold-, Warm-, Hot-, or NAND Flash Core Reset operation is performed.

**OTP Block Read Operation Flow Chart**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) BSA must be 1000 and BSC must be 000.

### 3.13.2 OTP Block Program Operation

An OTP Block Program Operation accesses the OTP area and programs content from the DataRAM on-chip buffer to the designated page(s) of the OTP.

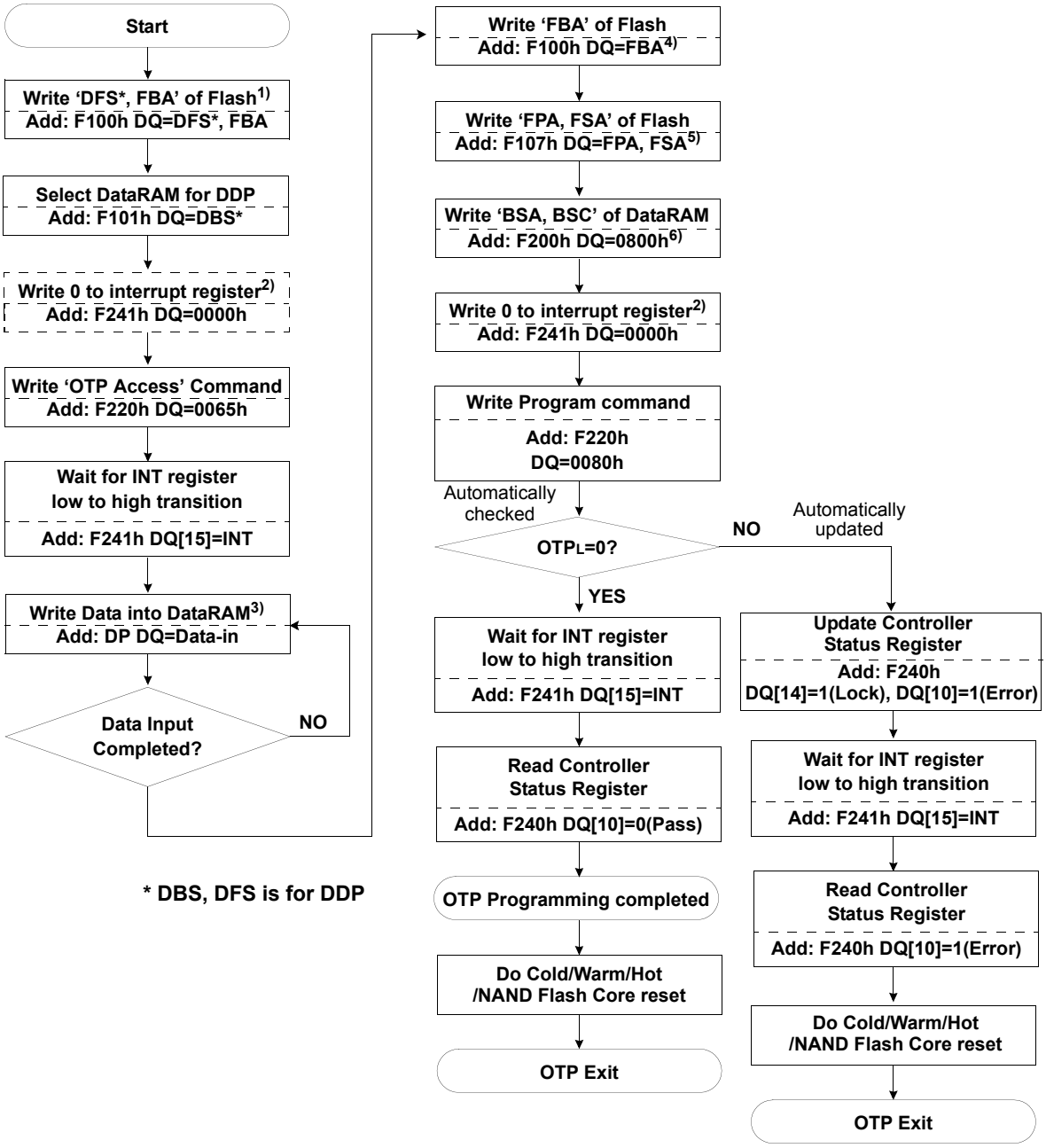
**A memory location in the OTP area can be programmed only one time (no erase operation permitted).**

The OTP area is programmed using the same sequence as normal program operation after being accessed by the command (see section 3.9 for more information).

**Programming the OTP Area**

- Issue the OTP Access Command
- Write data into the DataRAM (data can be input at anytime between the "Start" and "Write Program commands".
- Issue a Flash Block Address (FBA) which is 0000h of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the OTP
- When the OTP Block programming is complete,
  do a Cold-, Warm-, Hot-, NAND Flash Core Reset to exit the OTP Access mode.

**OTP Block Program Operation Flow Chart**




**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) Data input could be done anywhere between "Start" and "Write Program Command".
4) FBA must be 0000.
5) FSA must be 00 within program operation.
6) BSA must be 1000 and BSC must be 000.

### 3.13.3 OTP Block Lock Operation

Even though the OTP area can only be programmed once without erase capability, it can be locked when the device starts up to prevent any changes from being made.

Unlike the main area of the NAND Flash Array memory, **once the OTP block is locked, it cannot be unlocked, for locking bit for both blocks lies in the same word of OTP area.**
Therefore, if OTP Block is locked prior to 1st Block OTP lock, 1st Block OTP cannot be locked.

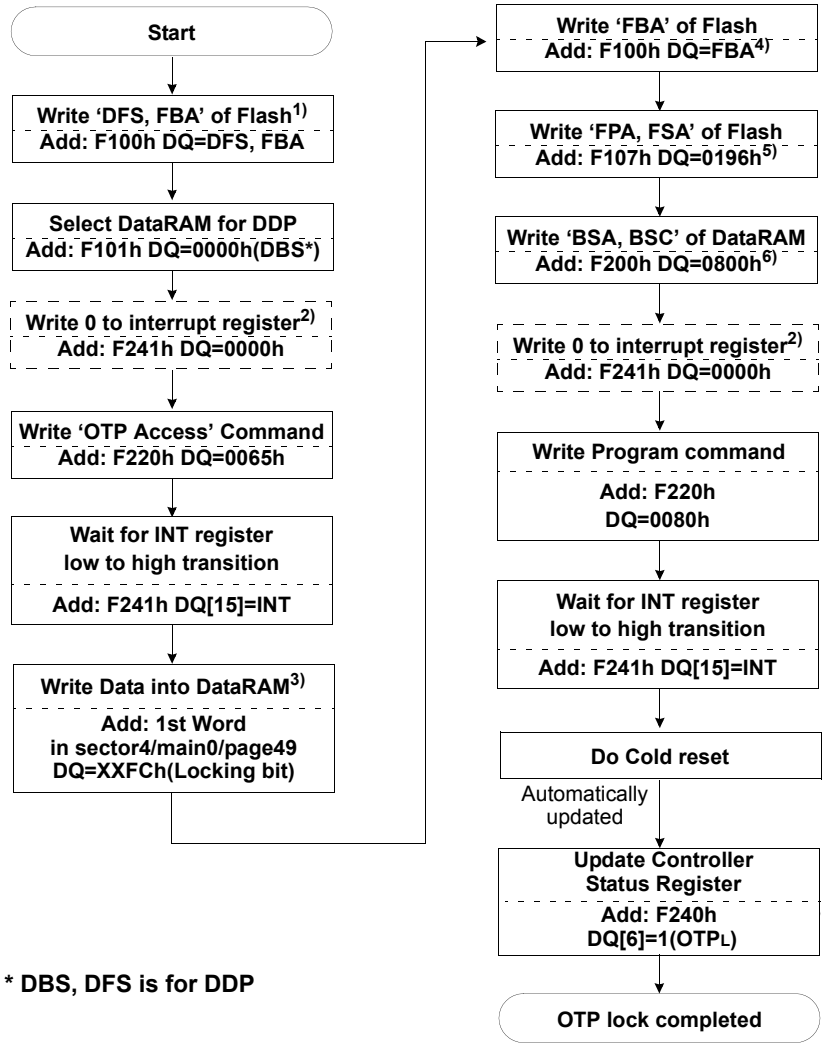
**Locking the OTP**
Programming to the OTP area can be prevented by locking the OTP area. Locking the OTP area is accomplished by programming **XXFCh** to the 1st word of sector4 of main of the page49 memory area in the OTP block.

At device power-up, this word location is checked and if **XXFCh** is found, the $OTP_L$ bit of the Controller Status Register is set to "1", indicating the OTP is locked. When the Program Operation finds that the status of the OTP is locked, the device updates the Error Bit of the Controller Status Register as "1" (fail).

**OTP Lock Operation Steps**

- Issue the OTP Access Command
- Fill data to be programmed into DataRAM (data can be input at anytime between the "Start" and "Write Program" commands)
- Write '**XXFCh**' data into the 1st word of sector4 of main of the page49 memory area of the DataRAM.
- Issue a Flash Block Address (FBA) which is 0000h of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the OTP
- When the OTP lock is complete, do a Cold Reset to exit the OTP Access mode and update OTP lock bit[6].
- OTP lock bit[6] of the Controller Status Register will be set to "1" and the OTP will be locked.

**OTP Block Lock Operation Flow Chart**




**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) Data input could be done anywhere between "Start" and "Write Program Command".
4) FBA must be 0000.
5) FSA must be 00 within program operation. The 0196h is the page49 of NAND Flash Array address map.
6) BSA must be 1000 and BSC must be 000.

### 3.13.4 1st Block OTP Lock Operation

1st Block can be used as OTP, for secured booting operation.
1st Block OTP can be accessed just as any other NAND Flash Array Blocks before it is locked, however, once 1st Block is locked to be OTP, 1st Block OTP cannot be erased or programmed.
Note that once OTP Block is locked, 1st Block OTP lock is impossible also OTP Block cannot be locked freely after locking 1st Block OTP. OTP Block and 1st Block OTP should be locked at the same time.

**Locking the 1st Block OTP**
Programming to the 1st Block OTP area can be prevented by locking the OTP area. Locking the OTP area is accomplished by programming **XXF3h** to the 1st word of sector4 of main of the page49 memory area in the OTP block.

At device power-up, this word location is checked and if **XXF3h** is found, the $OTP_{BL}$ bit of the Controller Status Register is set to "1", indicating the 1st Block is locked. When the Program Operation finds that the status of the 1st Block is locked, the device updates the Error Bit of the Controller Status Register as "1" (fail).

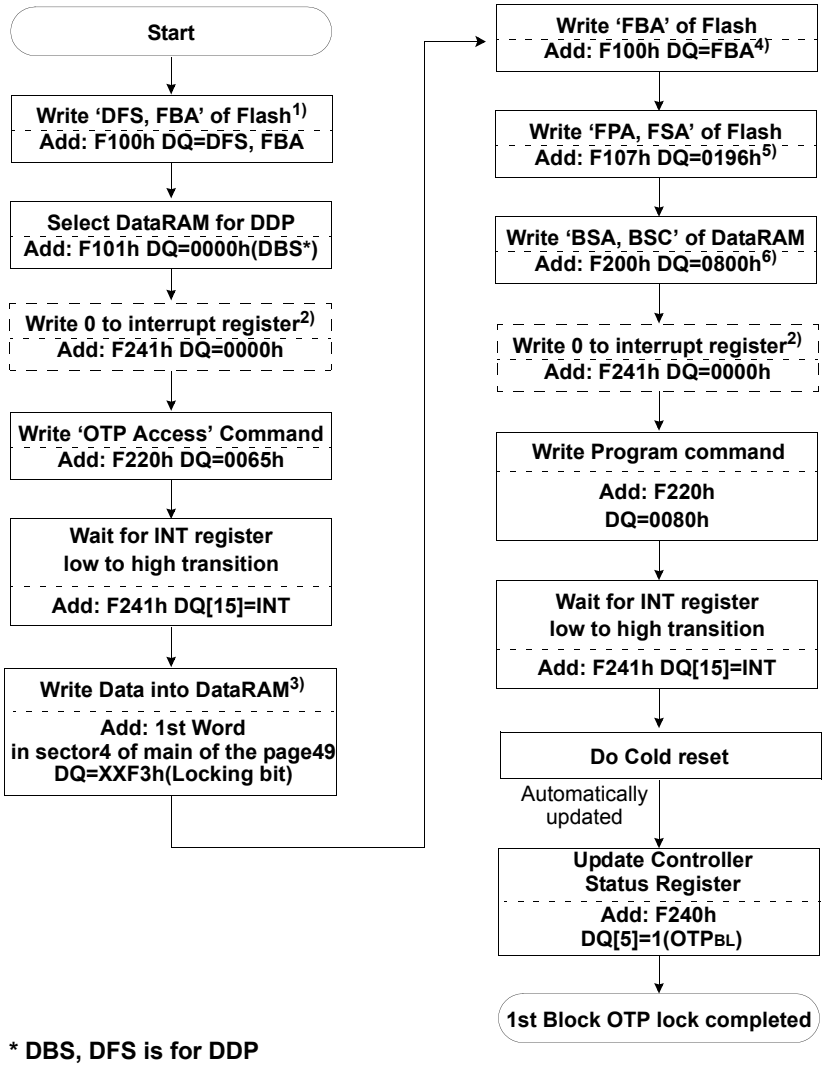
**1st Block OTP Lock Operation Steps**

- Issue the OTP Access Command
- Fill data to be programmed into DataRAM (data can be input at anytime between the "Start" and "Write Program" commands)
- Write '**XXF3h**' data into the 1st word of sector4 of main of the page49 memory area of the DataRAM.
- Issue a Flash Block Address (FBA) which is 0000h of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the OTP
- When the 1st Block OTP lock is complete, do a Cold Reset to exit the OTP Access mode and update 1st Block OTP lock bit[5].
- 1st Block OTP lock bit[5] of the Controller Status Register will be set to "1" and the 1st Block will be locked.

Even though the OTP area can only be programmed once without erase capability, it can be locked when the device starts up to prevent any changes from being made.

Unlike other remaining main area of the NAND Flash Array memory, **once the 1st block OTP is locked, it cannot be unlocked.**
Once 1st block is set as OTP, NAND Flash Write Protection status register(F24Eh) indicates only 'Lock' state although 'Lock tight' or 'Unlock' command is issued.

**1st Block OTP Lock Operation Flow Chart**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) Data input could be done anywhere between "Start" and "Write Program Command".
4) FBA must be 0000.
5) FSA must be 00 within program operation. The 0196h is the page49 of NAND Flash Array address map.
6) BSA must be 1000 and BSC must be 000.

### 3.13.5 OTP and 1st Block OTP Lock Operation

OTP and 1st Block can be locked simultaneously, for locking bit lies in the same word of OTP area.
1st Block OTP can be accessed just as any other NAND Flash Array Blocks before it is locked, however, once 1st Block is locked to be OTP, 1st Block OTP cannot be erased or programmed. Also, OTP area can only be programmed once without erase capability, it can be locked when the device starts up to prevent any changes from being made.

**Locking the OTP and 1st Block OTP**
Programming to the OTP area and 1st Block OTP area can be prevented by locking the OTP area. Locking the OTP area is accomplished by programming **XXF0h** to the 1st word of sector4 of main of the page49 memory area in the OTP block.

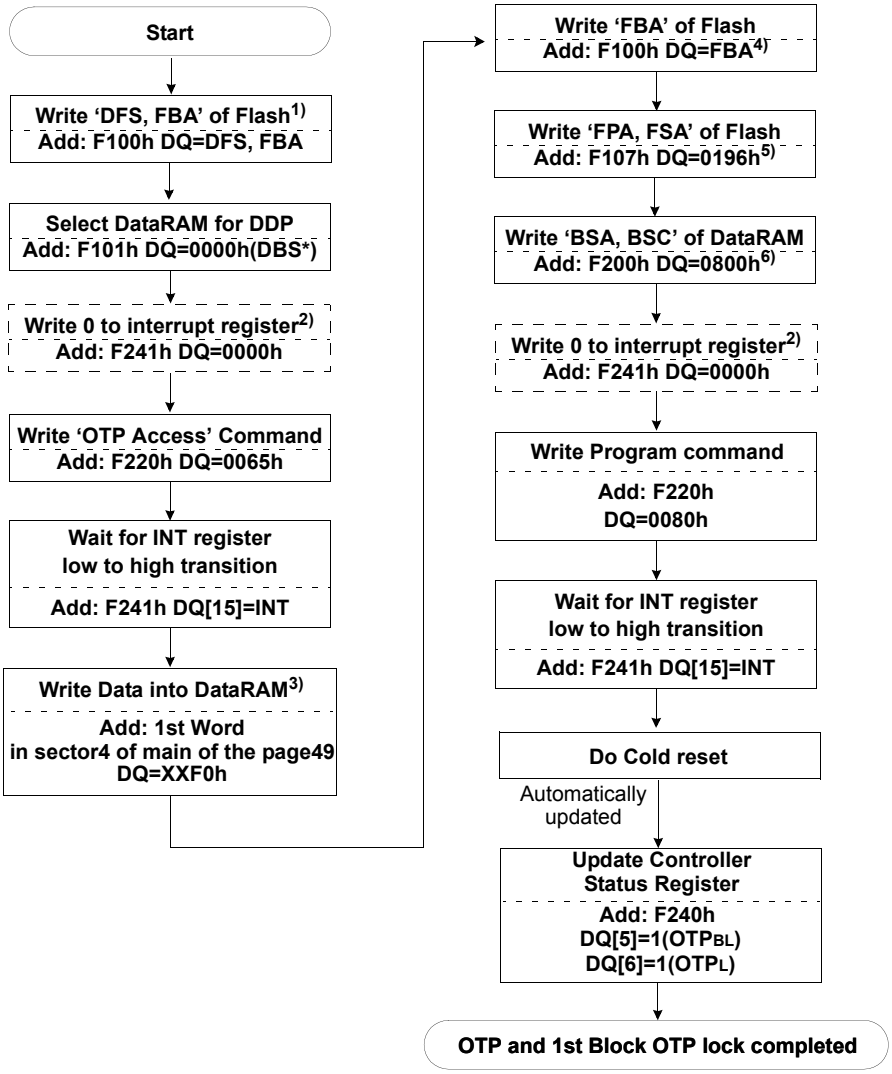
At device power-up, this word location is checked and if **XXF0h** is found, the $OTP_L$ and $OTP_{BL}$ bit of the Controller Status Register is set to "1", indicating the OTP and 1st Block is locked. When the Program Operation finds that the status of the OTP and 1st Block is locked, the device updates the Error Bit of the Controller Status Register as "1" (fail).

**OTP and 1st Block OTP simultaneous Lock Operation Steps**

- Issue the OTP Access Command
- Fill data to be programmed into DataRAM (data can be input at anytime between the "Start" and "Write Program" commands)
- Write '**XXF0h**' data into the 1st word of sector4 of main of the page49 memory area of the DataRAM.
- Issue a Flash Block Address (FBA) which is 0000h of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the OTP
- When the 1st Block OTP lock is complete, do a Cold Reset to exit the OTP Access mode and update 1st Block OTP lock bit[5] and OTP lock bit[6].
- 1st Block OTP lock bit[5] and OTP lock bit[6] of the Controller Status Register will be set to "1" and the OTP and 1st Block will be locked.

Unlike other remaining main area of the NAND Flash Array memory, **once the OTP block and the 1st block OTP are locked, it cannot be unlocked.**

**OTP and 1st Block OTP Lock Operation Flow Chart**




*** DBS, DFS is for DDP**

**NOTE :**
1) FBA(NAND Flash Block Address) could be omitted or any address.
2) 'Write 0 to interrupt register' step may be ignored when using INT auto mode. Refer to chapter 2.8.18.1
3) Data input could be done anywhere between "Start" and "Write Program Command".
4) FBA must be 0000.
5) FSA msut be 00 within program operation. The 0196h is the page49 of NAND Flash Array address map.
6) BSA msut be 1000 and BSC must be 000.

## 3.14 DQ6 Toggle Bit

The Flex-MuxOneNAND device has DQ6 Toggle bit. Toggle bit is another option to detect whether an internal load operation is in progress or completed. Once the BufferRAM(BootRAM, DataRAM0, DataRAM1) is at a busy state during internal load operation, DQ6 will toggle. Toggling DQ6 will stop after the device completes its internal load operation. The Flex-MuxOneNAND device's DQ6 Toggle will be valid only when host reads BufferRAM which will be loaded by internal load operation. DQ6 toggle can be used 350ns after load command(0000h of Command based Operation) issue, until data sensing from the NAND Flash Array memory into Page Buffer and transferring from the Page Buffer to the DataRAM are finished. By reading the same address more than twice utilizing asynchronous read (Figure 6.20, 6.21), the host will read toggled value of DQ6 and the rest of DQ's are not guaranteed to be fixed value. DQ6 toggle is only for reading status of BufferRAM which is being loaded by internal operation, that is, BufferRAM.

DQ6 toggle bit can be useful at Cold Reset to determine the ready/busy state of Flex-MuxOneNAND. Since INT pin is initially at High-Z state, when host needs to check the completion of boot code copy operation, the host cannot judge the ready/busy status of Flex-MuxOneNAND by INT pin. Therefore, by checking DQ6 toggle of BootRAM, the host should detect the completion of boot code copy.

| | Status | DQ15~DQ7 | DQ6 | DQ5~DQ0 |
|---|---|---|---|---|
| In Progress | Data Loading | X (Don't Care) | Toggle | X (Don't Care) |

## 3.15 ECC Operation

The Flex-MuxOneNAND device has on-chip ECC with the capability of correcting up to 4-bit errors in the NAND Flash Array memory main and spare areas (512+16)B.

As the device transfers data from a BufferRAM to the NAND Flash Array memory Page Buffer for Program Operation, the device initiates a background operation which generates an Error Correction Code (ECC).

During a Load operation from the NAND Flash Array memory Page, the on-chip ECC engine generates a new ECC. The 'Load ECC result' is compared to the originally programmed ECC' thus detecting the number of errors. Up to 4-bit errors are corrected.

ECC is updated by the device automatically. After a Load Operation, the Host can determine whether there was error by reading the 'ECC Status Register' (Refer to section 2.8.26~2.8.29).

Error types are divided into 'no error', 'correctable error(1bit~4bit)', and 'uncorrectable error(more than 4bit)'.

When the device reads the NAND Flash Array memory main and spare area data with an ECC operation, the device doesn't place the newly generated ECC for main and spare area into the buffer. Instead it places the ECC which was generated and written during the program operation into the buffer.

An ECC operation is also done during the Boot Loading operation.

### 3.15.1 ECC Bypass Operation

In an ECC bypass operation, the device does not generate ECC as a background operation.

In a Program Operation the ECC code to NAND Flash Array memory spare area is not updated.
During a Load operation, the on-chip ECC engine does not generate a new ECC internally. Also the ECC Status Registers are invalid. The error is not corrected and detected by itself, so that ECC bypass operation is not recommended for host.

ECC bypass operation is set by the 9bit of System Configuration 1 Register (see section 2.8.19)

**ECC Code and ECC Result by ECC Operation**

| Operation | Program operation | Load operation | | |
|---|---|---|---|---|
| | ECC Code Update to NAND Flash Array Spare Area | ECC Code at BufferRAM Spare Area | ECC Status & Result Update to Registers | 1bit~4bit Error |
| ECC operation | Update | Pre-written ECC code[1)] loaded | Update | Correct |
| ECC bypass | Not update | Pre-written code[1)] loaded | Invalid | Not correct |

**NOTE :**
1) Pre-written ECC code : ECC code which is previously written to NAND Flash Spare Area in program operation.

## 3.16 Invalid Block Operation

Invalid blocks are defined as blocks in the device's NAND Flash Array memory that contain five or more invalid bits which cause status failure during Program and Erase operation.

The information regarding the invalid block(s) is called the Invalid Block Information. Devices with invalid block(s) have the same quality level as devices with all valid blocks and have the same AC and DC characteristics.

An invalid block(s) does not affect the performance of valid block(s) because it is isolated from the bit line and the common source line by a select transistor.

The system that adopts Flash memory must be able to mask out the invalid block(s) by software. The 1st block is always fully guaranteed to be a valid block by an internal ECC engine.

Due to invalid marking, during load operation for indentifying invalid block, a load error may occur.

### 3.16.1 Invalid Block Identification Table Operation

A system must be able to recognize invalid block(s) based on the original invalid block information and create an invalid block table.

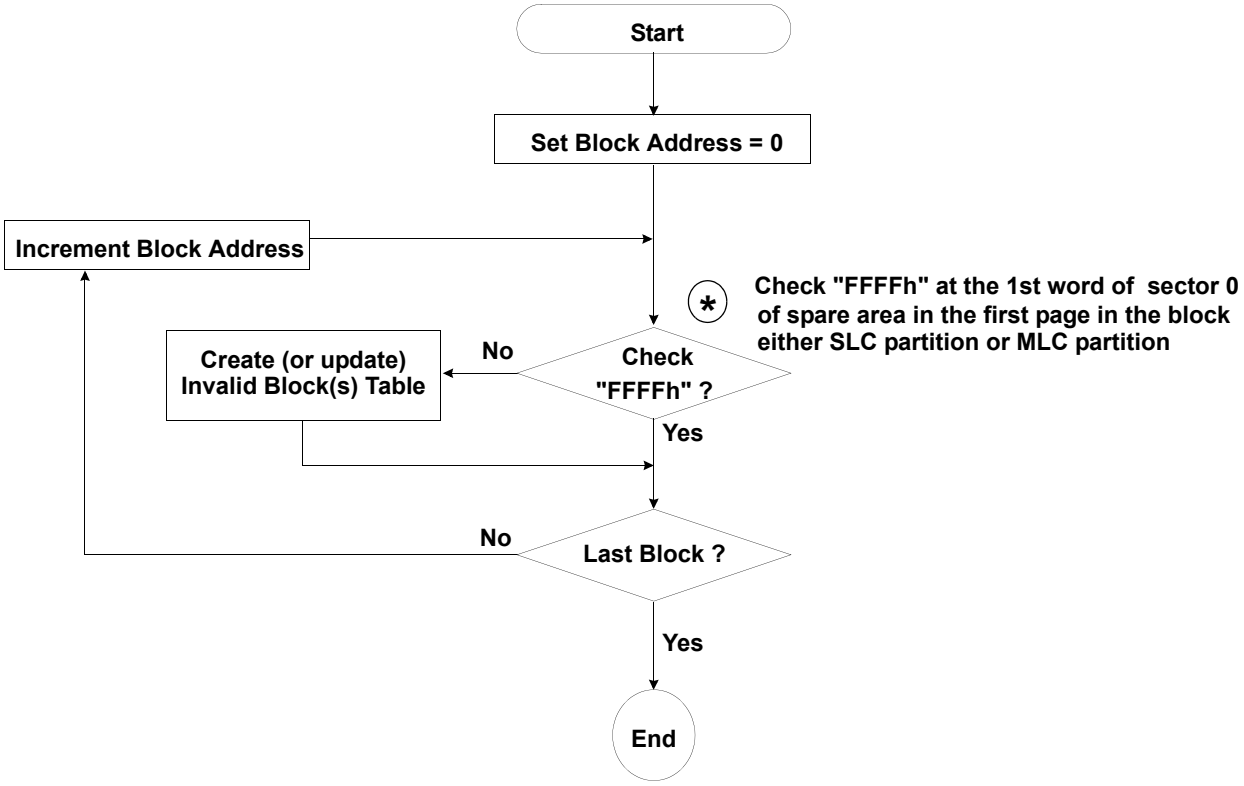
Invalid blocks are identified by erasing all address locations in the NAND Flash Array memory except locations where the invalid block(s) information is written prior to shipping.

An invalid block(s) status is defined by the 1st word in the spare area. Samsung makes sure that the first page in the block either SLC partition or MLC partition of every invalid block has non-FFFFh data at the 1st word of sector0 of pages 0 or 1 in the spare area.

Since the invalid block information is also erasable in most cases, it is impossible to recover the information once it has been erased. Any intentional erase of the original invalid block information is prohibited.

The following suggested flow chart can be used to create an Invalid Block Table.

**Invalid Block Table Creation Flow Chart**




### 3.16.2 Invalid Block Replacement Operation

Within its life time, additional invalid blocks may develop with NAND Flash Array memory. Refer to the device's qualification report for the actual data.

The following possible failure modes should be considered to implement a highly reliable system.

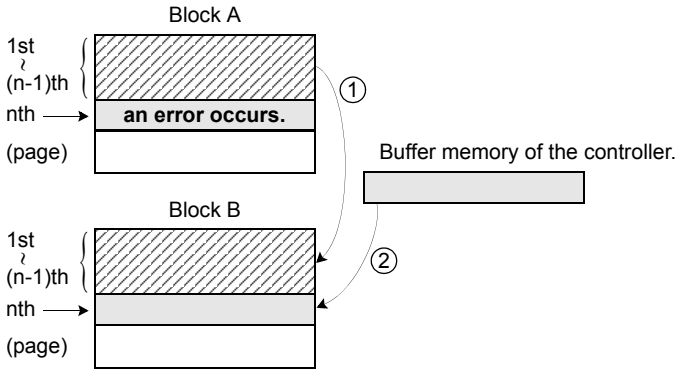
In the case of a status read failure after erase or program, a block replacement should be done. Program status failure during a page program does not affect the data of the other pages in the same block within a SLC partition, while Progrm status failure could contaminate the data of the paired page within a MLC partition. So, users must make sure how software handle the program failure occurrs.

**Block Failure Modes and Countermeasures**

| Failure Mode | Detection and Countermeasure sequence |
|---|---|
| Erase Failure | Status Read after Erase --> Block Replacement |
| Program Failure | Status Read after Program --> Block Replacement |
| Four Bit Failure in Load Operation | Error Correction by ECC mode of the device |

Referring to the diagram for further illustration, when an error happens in the nth page of block 'A' during program operation, copy the data in the 1st ~ (n-1)th page to the same location of block 'B' via DataRAM.
Then re-program the nth page to the nth page of block 'B' or any free block.
Do not further erase or program block 'A' but instead complete the operation by creating an 'Invalid Block Table' or other appropriate scheme.

**Block Replacement Operation Sequence**



# 4.0 DC CHARACTERISTICS

## 4.1 Absolute Maximum Ratings

| Parameter | | Symbol | Rating | Unit |
|---|---|---|---|---|
| Voltage on any pin relative to $V_{SS}$ | Vcc | Vcc | -0.5 to + 2.45 | V |
| | All Pins | $V_{IN}$ | -0.5 to + 2.45 | |
| Temperature Under Bias | Extended | $T_{bias}$ | -30 to +125 | °C |
| | Industrial | | -40 to +125 | |
| Storage Temperature | | $T_{stg}$ | -65 to +150 | °C |
| Short Circuit Output Current | | $I_{OS}$ | 5 | mA |
| Recommended Operating Temperature | | $T_A$ (Extended Temp.) | -30 to +85 | °C |
| | | $T_A$ (Industrial Temp.) | -40 to +85 | |

**NOTE :**
1) Minimum DC voltage is -0.5V on Input/ Output pins. During transitions, this level should not fall to POR level(typ. 1.5V@1.8V device). Maximum DC voltage may overshoot to Vcc+2.0V for periods <20ns.
2) Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to the conditions detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

## 4.2 Operating Conditions

Voltage reference to GND

| Parameter | Symbol | KFM4GH6Q4M | | | Unit |
|---|---|---|---|---|---|
| | | Min | Typ. | Max | |
| Supply Voltage | $V_{CC}$-core / Vcc | 1.7 | 1.8 | 1.95 | V |
| | $V_{CC}$- IO / Vccq | | | | |
| | $V_{SS}$ | 0 | 0 | 0 | V |

**NOTE :**
1) Vcc-Core (or Vcc) should reach the operating voltage level prior to or at the same time as Vcc-IO (or Vccq).

## 4.3 DC Characteristics

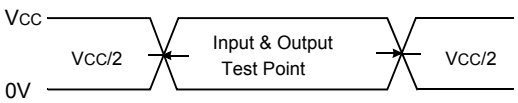
| Parameter | Symbol | Test Conditions | | RMS Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min | Typ | Max | |
| Input Leakage Current | $I_{LI}$ | $V_{IN}$=$V_{SS}$ to $V_{CC}$, $V_{CC}$=$V_{CCmax}$ | Single | - 1.0 | - | + 1.0 | µA |
| | | | DDP | - 2.0 | - | + 2.0 | |
| Output Leakage Current | $I_{LO}$ | $V_{OUT}$=$V_{SS}$ to $V_{CC}$, $V_{CC}$=$V_{CCmax}$, $\overline{CE}$ or $\overline{OE}$=$V_{IH}$(Note 1) | Single | - 1.0 | - | + 1.0 | µA |
| | | | DDP | - 2.0 | | + 2.0 | |
| Active Asynchronous Read Current (Note 2) | $I_{CC1}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$ | | - | 8 | 15 | mA |
| Active Burst Read Current (Note 2) | $I_{CC2R}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$, $\overline{WE}$=$V_{IH}$ | 66MHz | - | 20 | 30 | mA |
| | | | 83MHz | - | 25 | 35 | mA |
| | | | 1MHz | - | 3 | 4 | mA |
| | | | 66MHz (DDP) | - | 22 | 35 | mA |
| | | | 83MHz (DDP) | - | 26 | 40 | mA |
| | | | 1MHz (DDP) | - | 3 | 4 | mA |
| Active Burst Write Current (Note 2) | $I_{CC2W}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$, $\overline{WE}$=$V_{IL}$ | 66MHz | - | 20 | 30 | mA |
| | | | 83MHz | - | 25 | 35 | mA |
| | | | 1MHz | - | 3 | 4 | mA |
| | | | 66MHz (DDP) | - | 22 | 35 | mA |
| | | | 83MHz (DDP) | - | 26 | 40 | mA |
| | | | 1MHz (DDP) | - | 3 | 4 | mA |
| Active Asynchronous Write Current (Note 2) | $I_{CC3}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$ | Single | - | 8 | 15 | mA |
| | | | DDP | - | 17 | 25 | mA |
| Active Load Current (Note 3) | $I_{CC4}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$, $\overline{WE}$=$V_{IH}$ | | - | 50 | 65 | mA |
| Active Program Current (Note 3) | $I_{CC5}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$, $\overline{WE}$=$V_{IH}$ | | - | 35 | 45 | mA |
| Active Erase Current (Note 3) | $I_{CC6}$ | $\overline{CE}$=$V_{IL}$, $\overline{OE}$=$V_{IH}$, $\overline{WE}$=$V_{IH}$ | | - | 40 | 50 | mA |
| Standby Current | $I_{SB}$ | $\overline{CE}$= $\overline{RP}$=$V_{CC}$ ± 0.2V | Single | - | 10 | 50 | µA |
| | | | DDP | - | 20 | 100 | |
| Input Low Voltage | $V_{IL}$ | - | | -0.5 | - | 0.4 | V |
| Input High Voltage (Note 4) | $V_{IH}$ | - | | $V_{CCq}$-0.4 | - | $V_{CCq}$+0.4 | V |
| Output Low Voltage | $V_{OL}$ | $I_{OL}$ = 100 µA ,$V_{CC}$=$V_{CCmin}$ , $V_{CCq}$=$V_{CCqmin}$ | | - | - | 0.2 | V |
| Output High Voltage | $V_{OH}$ | $I_{OH}$ = -100 µA , $V_{CC}$=$V_{CCmin}$ , $V_{CCq}$=$V_{CCqmin}$ | | $V_{CCq}$-0.1 | - | - | V |

**NOTE :**
1) $\overline{CE}$ should be VIH for RDY. IOBE should be '0' for INT.
2) $I_{CC}$ active for Host access
3) $I_{CC}$ active for Internal operation. (without host access)
4) Vccq is equivalent to Vcc-IO

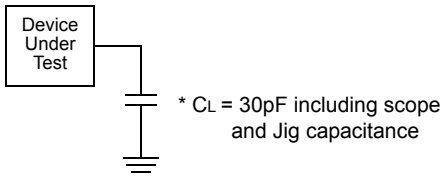
# 5.0 AC CHARACTERISTICS

## 5.1 AC Test Conditions

| Parameter | | Value (66MHz) | Value (83MHz) |
|---|---|---|---|
| Input Pulse Levels | | 0V to $V_{CC}$ | 0V to $V_{CC}$ |
| Input Rise and Fall Times | CLK | 3ns | 2ns |
| | other inputs | 5ns | 2ns |
| Input and Output Timing Levels | | $V_{CC}/2$ | $V_{CC}/2$ |
| Output Load | | $C_L$ = 30pF | $C_L$ = 30pF |




**Input Pulse and Test Point**




**Output Load**

## 5.2 Device Capacitance

**CAPACITANCE**($T_A$ = 25 °C, $V_{CC}$ = 1.8V, f = 1.0MHz)

| Item | Symbol | Test Condition | Single | | DDP | | QDP | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| Input Capacitance | $C_{IN1}$ | $V_{IN}$=0V | - | 10 | - | 20 | - | 40 | pF |
| Control Pin Capacitance | $C_{IN2}$ | $V_{IN}$=0V | - | 10 | - | 20 | - | 40 | |
| Output Capacitance | $C_{OUT}$ | $V_{OUT}$=0V | - | 10 | - | 20 | - | 40 | |
| INT Capacitance | $C_{INT}$ | $V_{OUT}$=0V | - | 10 | - | 20 | - | 40 | |

**NOTE :**
1) Capacitance is periodically sampled and not 100% tested.

## 5.3 Valid Block Characteristics

| Parameter | | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| Valid Block Number | Single | $N_{VB}$ | 998 | - | 1024 | Blocks |
| | DDP | | 1996 | - | 2048 | |
| | QDP | | 3993 | - | 4096 | |

**NOTE :**
1) The device may include invalid blocks when first shipped. Additional invalid blocks may develop while being used. The number of valid blocks is presented with both cases of invalid blocks considered. Invalid blocks are defined as blocks that contain five or more bad bits which cause status failure during Program and Erase operation. Do not erase or program factory-marked bad blocks.
2) The 1st block, which is placed on 00h block address, is guaranteed to be a valid block up to 1K program/erase cycles with 4bit/528Byte ECC.

## 5.4 AC Characteristics for Synchronous Burst Read

**See Timing Diagrams 6.1 and 6.2**

| Parameter | Symbol | 66MHz | | 83MHz | | Unit |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| Clock | CLK | 1 | 66 | 1 | 83 | MHz |
| Clock Cycle | $t_{CLK}$ | 15 | - | 12 | - | ns |
| Initial Access Time | $t_{IAA}$ | - | 70 | - | 70 | ns |
| Burst Access Time Valid Clock to Output Delay | $t_{BA}$ | - | 11 | - | 9 | ns |
| $\overline{AVD}$ Setup Time to CLK | $t_{AVDS}$ | 5 | - | 4 | - | ns |
| $\overline{AVD}$ Hold Time from CLK | $t_{AVDH}$ | 2 | - | 2 | - | ns |
| $\overline{AVD}$ High to $\overline{OE}$ Low | $t_{AVDO}$ | 0 | - | 0 | - | ns |
| Address Setup Time to CLK | $t_{ACS}$ | 5 | - | 4 | - | ns |
| Address Hold Time from CLK | $t_{ACH}$ | 6 | - | 6 | - | ns |
| Data Hold Time from Next Clock Cycle | $t_{BDH}$ | 3 | - | 2 | - | ns |
| Output Enable to Data | $t_{OE}$ | - | 20 | - | 20 | ns |
| $\overline{CE}$ Disable to Output & RDY High Z | $t_{CEZ}$ 1) | - | 20 | - | 20 | ns |
| $\overline{OE}$ Disable to Output High Z | $t_{OEZ}$ 1) | - | 15 | - | 15 | ns |
| $\overline{CE}$ Setup Time to CLK | $t_{CES}$ | 6 | - | 4.5 | - | ns |
| CLK High or Low Time | $t_{CLKH/L}$ | $t_{CLK}$/3 | - | 5 | - | ns |
| CLK 2) to RDY valid | $t_{RDYO}$ | - | 11 | - | 9 | ns |
| CLK to RDY Setup Time | $t_{RDYA}$ | - | 11 | - | 9 | ns |
| RDY Setup Time to CLK | $t_{RDYS}$ | 4 | - | 3 | - | ns |
| $\overline{CE}$ low to RDY valid | $t_{CER}$ | - | 15 | - | 15 | ns |

**NOTE :**
1) If $\overline{OE}$ is disabled at the same time or before $\overline{CE}$ is disabled, the output will go to high-z by $t_{OEZ}$.
If $\overline{CE}$ is disabled at the same time or before $\overline{OE}$ is disabled, the output will go to high-z by $t_{CEZ}$.
If $\overline{CE}$ and $\overline{OE}$ are disabled at the same time, the output will go to high-z by $t_{OEZ}$.
2) It is the following clock of address fetch clock.

## 5.5 AC Characteristics for Asynchronous Read

**See Timing Diagrams 6.3 and 6.4.**

| Parameter | Symbol | KFM4GH6Q4M/ KFN8GH6Q4M/ KFKAGH6Q4M(TBD) | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Access Time from $\overline{CE}$ Low | $t_{CE}$ | - | 76 | ns |
| Asynchronous Access Time from $\overline{AVD}$ Low | $t_{AA}$ | - | 76 | ns |
| Asynchronous Access Time from address valid | $t_{ACC}$ | - | 76 | ns |
| Read Cycle Time | $t_{RC}$ | 76 | - | ns |
| $\overline{AVD}$ Low Time | $t_{AVDP}$ | 12 | - | ns |
| Address Setup to rising edge of $\overline{AVD}$ | $t_{AAVDS}$ | 5 | - | ns |
| Address Hold from rising edge of $\overline{AVD}$ | $t_{AAVDH}$ | 6 | - | ns |
| Output Enable to Output Valid | $t_{OE}$ | - | 20 | ns |
| $\overline{CE}$ Setup to $\overline{AVD}$ falling edge | $t_{CA}$ | 0 | - | ns |
| $\overline{CE}$ Disable to Output & RDY High Z[1)] | $t_{CEZ}$ | - | 20 | ns |
| $\overline{OE}$ Disable to Output High Z[1)] | $t_{OEZ}$ | - | 15 | ns |
| $\overline{AVD}$ High to $\overline{OE}$ Low | $t_{AVDO}$ | 0 | - | ns |
| $\overline{CE}$ Low to RDY Valid | $t_{CER}$ | - | 15 | ns |
| $\overline{WE}$ Disable to $\overline{AVD}$ Enable | $t_{WEA}$ | 15 | - | ns |
| Address to $\overline{OE}$ low | $t_{ASO}$[2)] | 10 | - | ns |

**NOTE :**
1) If $\overline{OE}$ is disabled at the same time or before $\overline{CE}$ is disabled, the output will go to high-z by tOEZ.
If $\overline{CE}$ is disabled at the same time or before $\overline{OE}$ is disabled, the output will go to high-z by tCEZ.
If $\overline{CE}$ and $\overline{OE}$ are disabled at the same time, the output will go to high-z by tOEZ.
These parameters are not 100% tested.
2) This Parameter is valid at toggle bit timing in asynchronous read only. (timing diagram 6.20 and 6.21)

## 5.6 AC Characteristics for Warm Reset ($\overline{RP}$), Hot Reset and NAND Flash Core Reset

**See Timing Diagrams 6.16, 6.17 and 6.18.**

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| $\overline{RP}$ & Reset Command Latch to BootRAM Access | tReady1 (BufferRAM) | - | 5 | µs |
| $\overline{RP}$ & Reset Command Latch(During Load Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 10 | µs |
| $\overline{RP}$ & Reset Command Latch(During Program Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 20 | µs |
| $\overline{RP}$ & Reset Command Latch(During Erase Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 150 | µs |
| $\overline{RP}$ & Reset Command Latch(NOT During Internal Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 10 | µs |
| $\overline{RP}$ Pulse Width (Note2) | tRP | 200 | - | ns |

**NOTE :**
1) These parameters are tested based on INT bit of interrupt register. Because the time on INT pin is related to the pull-up and pull-down resistor value.
2) The device may reset if tRP < tRP min(200ns), but this is not guaranteed.

## 5.7 AC Characteristics for Asynchronous Write

**See Timing Diagrams 6.5**

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| $\overline{WE}$ Cycle Time | $t_{WC}$ | 70 | - | ns |
| $\overline{AVD}$ low pulse width | $t_{AVDP}$ | 12 | - | ns |
| Address Setup Time | $t_{AAVDS}$ | 5 | - | ns |
| Address Hold Time | $t_{AAVDH}$ | 6 | - | ns |
| Data Setup Time | $t_{DS}$ | 30 | - | ns |
| Data Hold Time | $t_{DH}$ | 0 | - | ns |
| $\overline{CE}$ Setup Time | $t_{CS}$ | 0 | - | ns |
| $\overline{CE}$ Hold Time | $t_{CH}$ | 0 | - | ns |
| $\overline{WE}$ Pulse Width | $t_{WPL}$ | 40 | - | ns |
| $\overline{WE}$ Pulse Width High | $t_{WPH}$ | 30 | - | ns |
| $\overline{WE}$ Disable to $\overline{AVD}$ Enable | $t_{WEA}$ | 15 | - | ns |
| $\overline{CE}$ Low to RDY Valid | $t_{CER}$ | - | 15 | ns |
| $\overline{CE}$ Disable to Output & RDY High Z | $t_{CEZ}$ | - | 20 | ns |

## 5.8 AC Characteristics for Burst Write Operation

**See Timing Diagrams 6.6, 6.7 and 6.8.**

| Parameter | Symbol | 66MHz | | 83MHz | | Unit |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| Clock | CLK[1)] | 1 | 66 | 1 | 83 | MHz |
| Clock Cycle | $t_{CLK}$ | 15 | - | 12 | - | ns |
| $\overline{AVD}$ Setup to CLK | $t_{AVDS}$ | 5 | - | 4 | - | ns |
| $\overline{AVD}$ Hold Time from CLK | $t_{AVDH}$ | 2 | - | 2 | - | ns |
| Address Setup Time to CLK | $t_{ACS}$ | 5 | - | 4 | - | ns |
| Address Hold Time from CLK | $t_{ACH}$ | 6 | - | 6 | - | ns |
| Data Setup Time to CLK | $t_{WDS}$ | 5 | - | 4 | - | ns |
| Data Hold Time from CLK | $t_{WDH}$ | 2 | - | 2 | - | ns |
| $\overline{WE}$ Setup Time to CLK | $t_{WES}$ | 5 | - | 4 | - | ns |
| $\overline{WE}$ Hold Time from CLK | $t_{WEH}$ | 6 | - | 6 | - | ns |
| CLK High or Low Time | $t_{CLKH/L}$ | $t_{CLK}$/3 | - | 5 | - | ns |
| $\overline{CE}$ high pulse width | $t_{CEHP}$ | 10 | - | 10 | - | ns |
| CLK to RDY Valid | $t_{RDYO}$ | - | 11 | - | 9 | ns |
| CLK to RDY Setup Time | $t_{RDYA}$ | - | 11 | - | 9 | ns |
| RDY Setup Time to CLK | $t_{RDYS}$ | 4 | - | 3 | - | ns |
| $\overline{CE}$ low to RDY valid | $t_{CER}$ | - | 15 | - | 15 | ns |
| Clock to $\overline{CE}$ disable | $t_{CEH}$ | 2 | $t_{CLK}$-4.5 | 2 | $t_{CLK}$-4.5 | ns |
| $\overline{CE}$ Setup Time to CLK | $t_{CES}$ | 6 | - | 4.5 | - | ns |
| $\overline{CE}$ Disable to Output & RDY High Z | $t_{CEZ}$ | - | 20 | - | 20 | ns |

**NOTE :**
1)Target Clock frequency is 83Mhz

## 5.9 AC Characteristics for Load/Program/Erase Performance

**See Timing Diagrams 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14**

| Parameter | Symbol | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Sector Load time(Note 1) | $t_{RD1}$ | SLC | - | 25 | 75 | µs |
| | | MLC | | 30 | 100 | µs |
| Page Load time(Note 1) | $t_{RD2}$ | SLC | - | 45 | 400 | µs |
| | | MLC | - | 50 | 420 | µs |
| Page Program time(Note 1) | $t_{PGM2}$ | SLC | - | 240 | 770 | µs |
| | | MLC | | 1000 | 5000 | µs |
| OTP Access Time(Note 1) | $t_{OTP}$ | | - | 500 | 700 | ns |
| Lock/Unlock/Lock-tight(Note 1) | $t_{LOCK}$ | | - | 500 | 700 | ns |
| All Block Unlock Time(Note 1) | $t_{ABU}$ | | - | 2 | 3 | µs |
| Erase Suspend Time(Note 1) | $t_{ESP}$ | | - | 400 | 500 | µs |
| Erase Resume Time(Note 1) | $t_{ERS1}$ | | - | 0.5 | 11 | ms |
| Number of Partial Program Cycles in the page (Including main and spare area) | NOP | SLC | - | - | 1 | cycles |
| | | MLC | - | - | 1 | cycles |
| Block Erase time (Note 1) | $t_{BERS1}$ | | - | 0.5 | 11 | ms |

**NOTE :**
1) These parameters are tested based on INT bit of interrupt register. Because the time on INT pin is related to the pull-up and pull-down resistor value.

## 5.10 AC Characteristics for INT Auto Mode

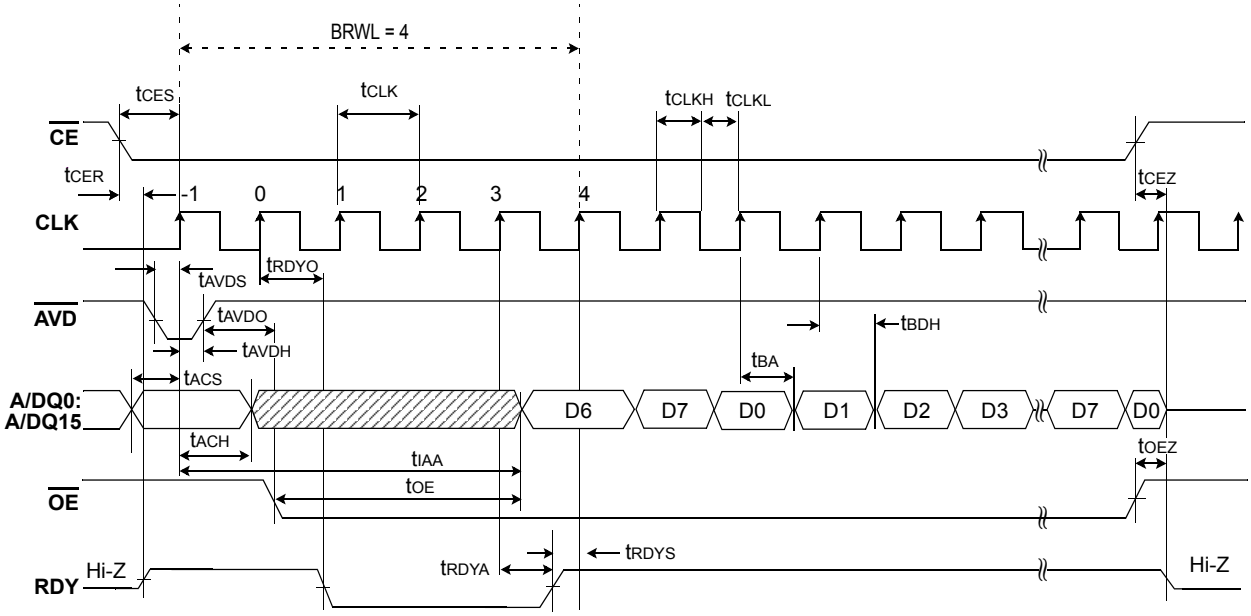
**See Timing Diagram 6.22**

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Command Input to INT Low | $t_{WB}$ | - | 200 | ns |

# 6.0 TIMING DIAGRAMS

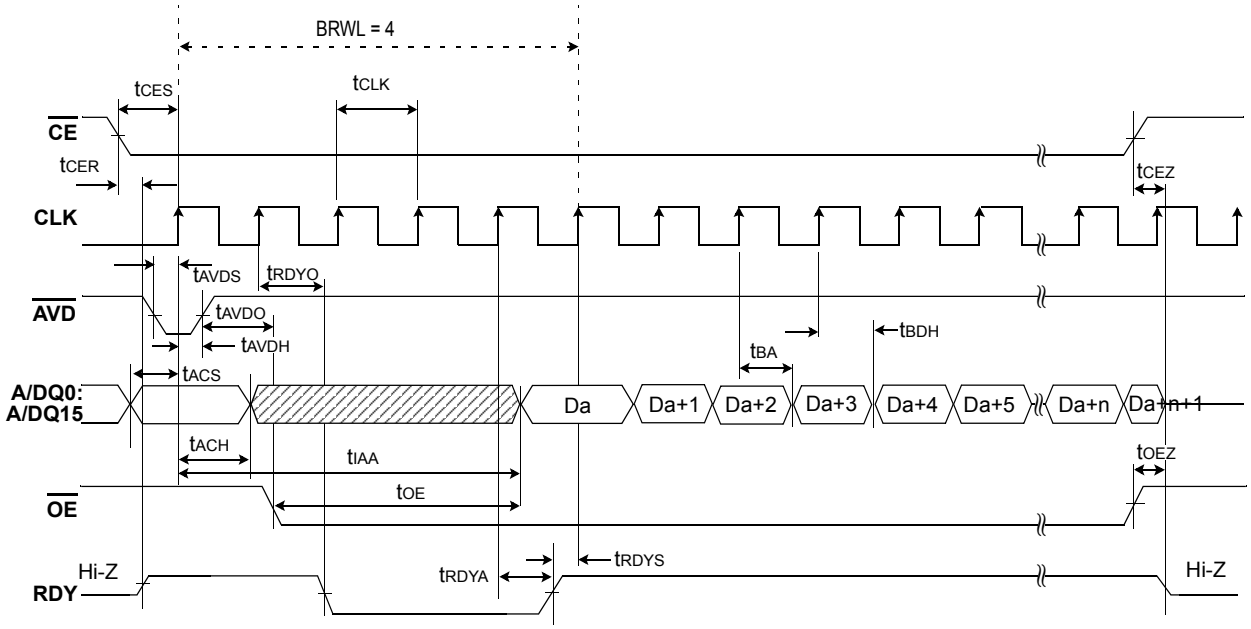
## 6.1 8-Word Linear Burst Read Mode with Wrap Around

**See AC Characteristics Table 5.4**



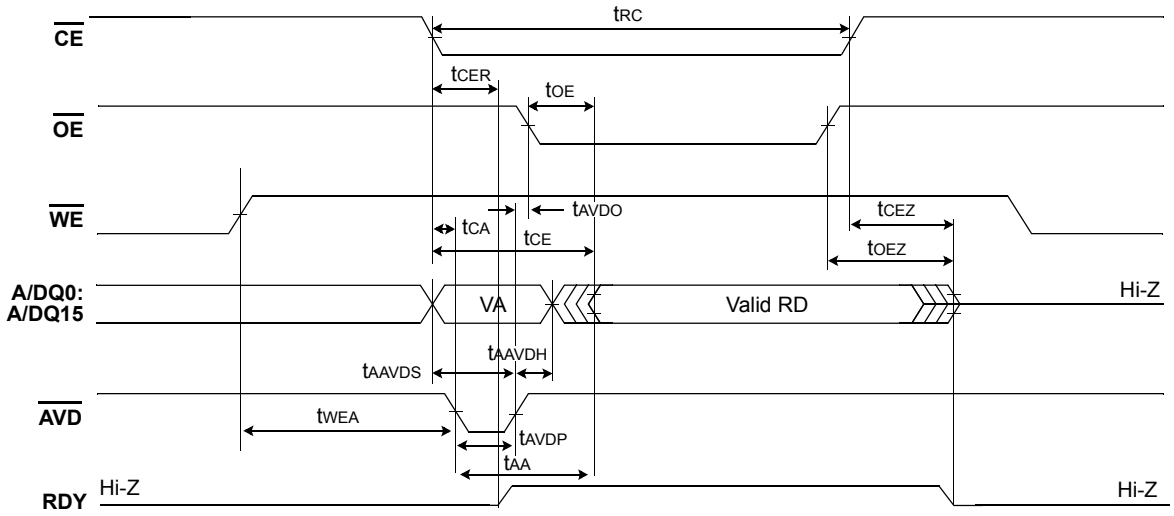

## 6.2 Continuous Linear Burst Read Mode with Wrap Around

**See AC Characteristics Table 5.4**



## 6.3 Asynchronous Read (VA Transition Before $\overline{AVD}$ Low)

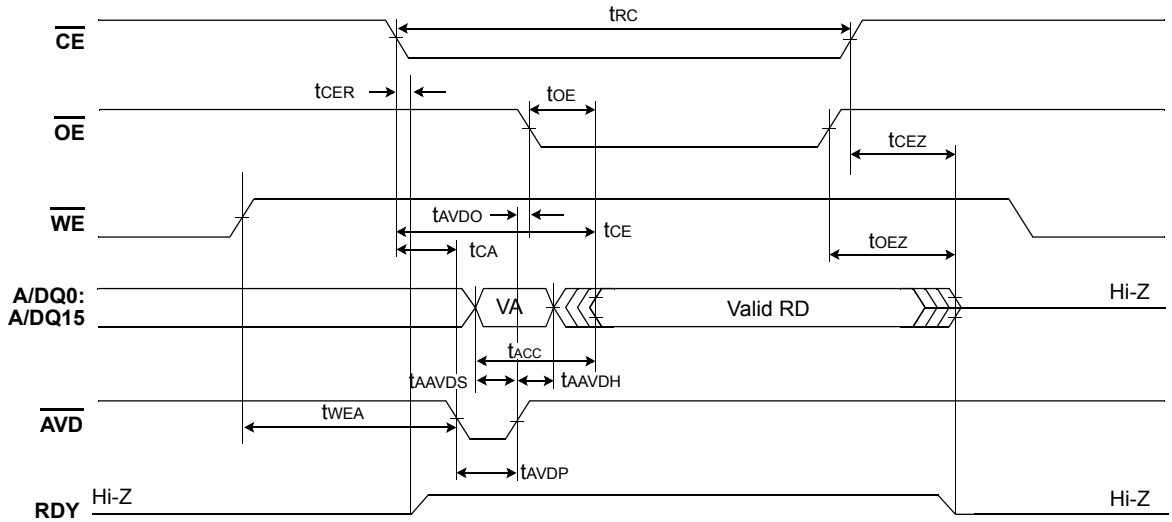
**See AC Characteristics Table 5.5**




**NOTE :**
VA=Valid Read Address, RD=Read Data.
See timing diagram 6.20, 6.21 for tASO

## 6.4 Asynchronous Read (VA Transition After $\overline{AVD}$ Low)

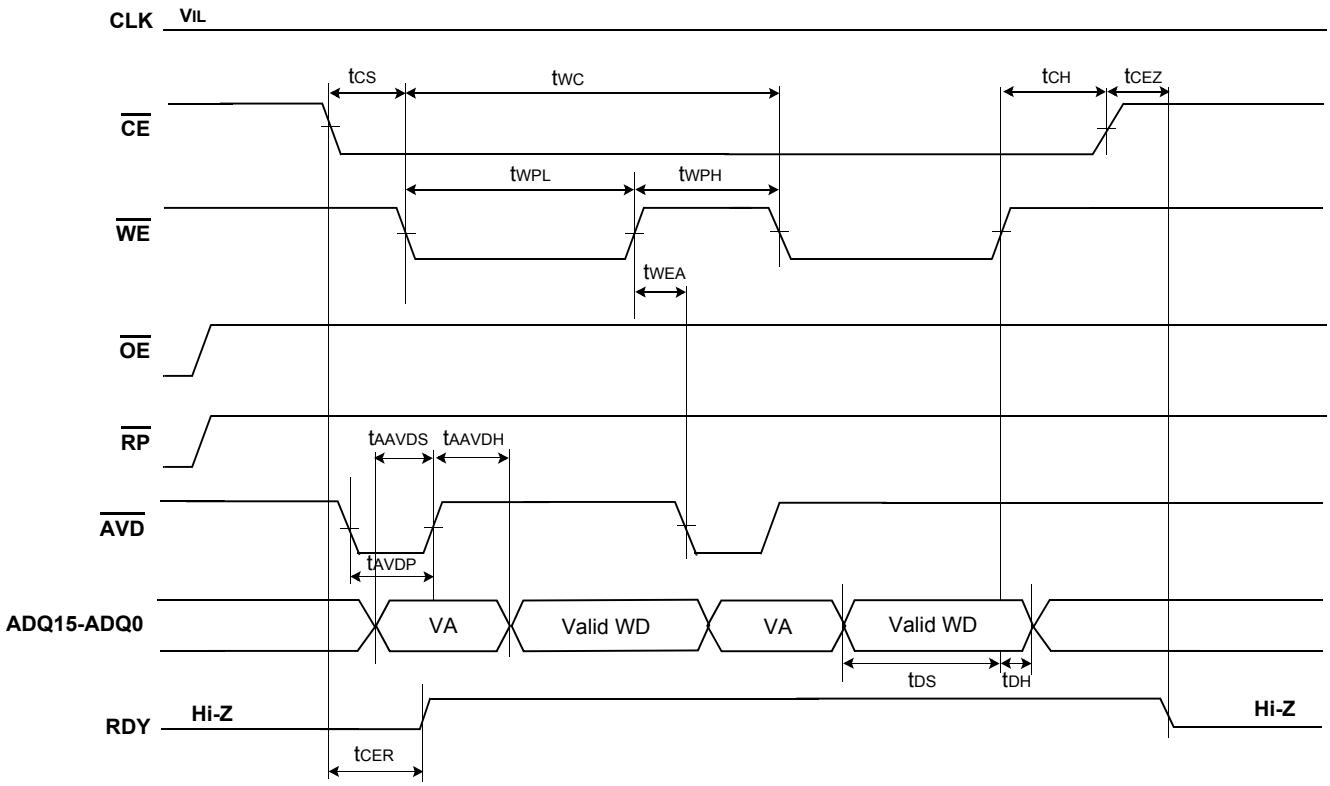
**See AC Characteristics Table 5.5**




**NOTE :**
VA=Valid Read Address, RD=Read Data.
See timing diagram 6.21, 6.22 for tASO

## 6.5 Asynchronous Write

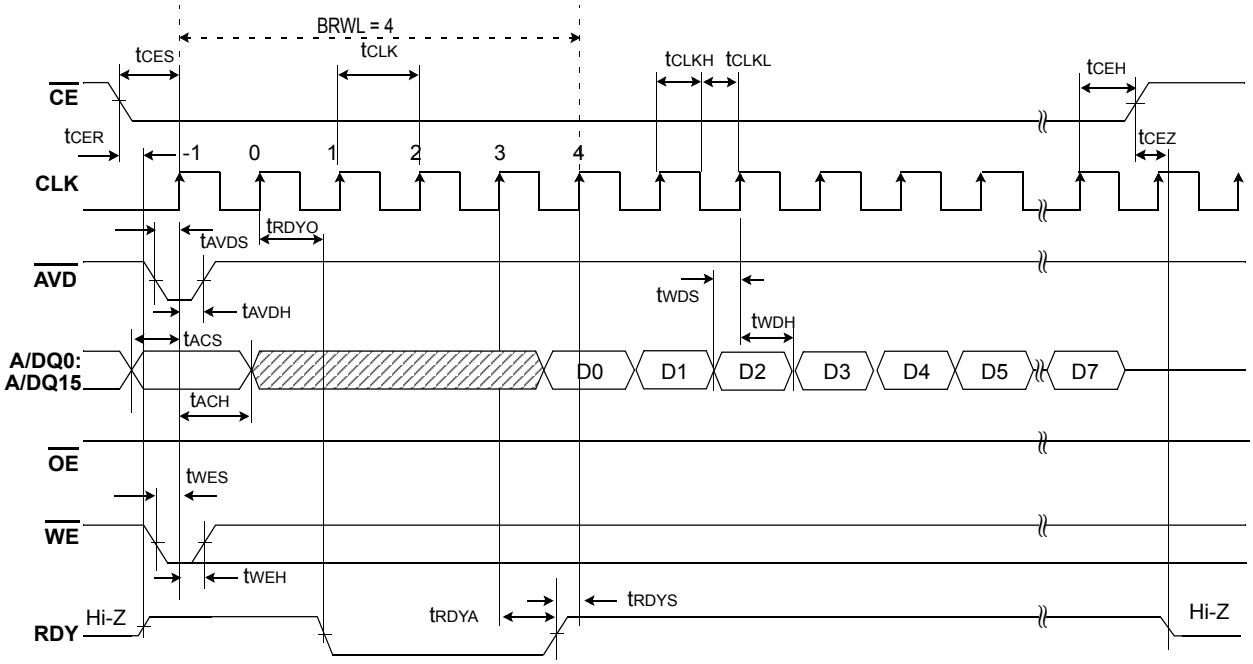
**See AC Characteristics Table 5.7**




**NOTE :**
VA=Valid Read Address, WD=Write Data.

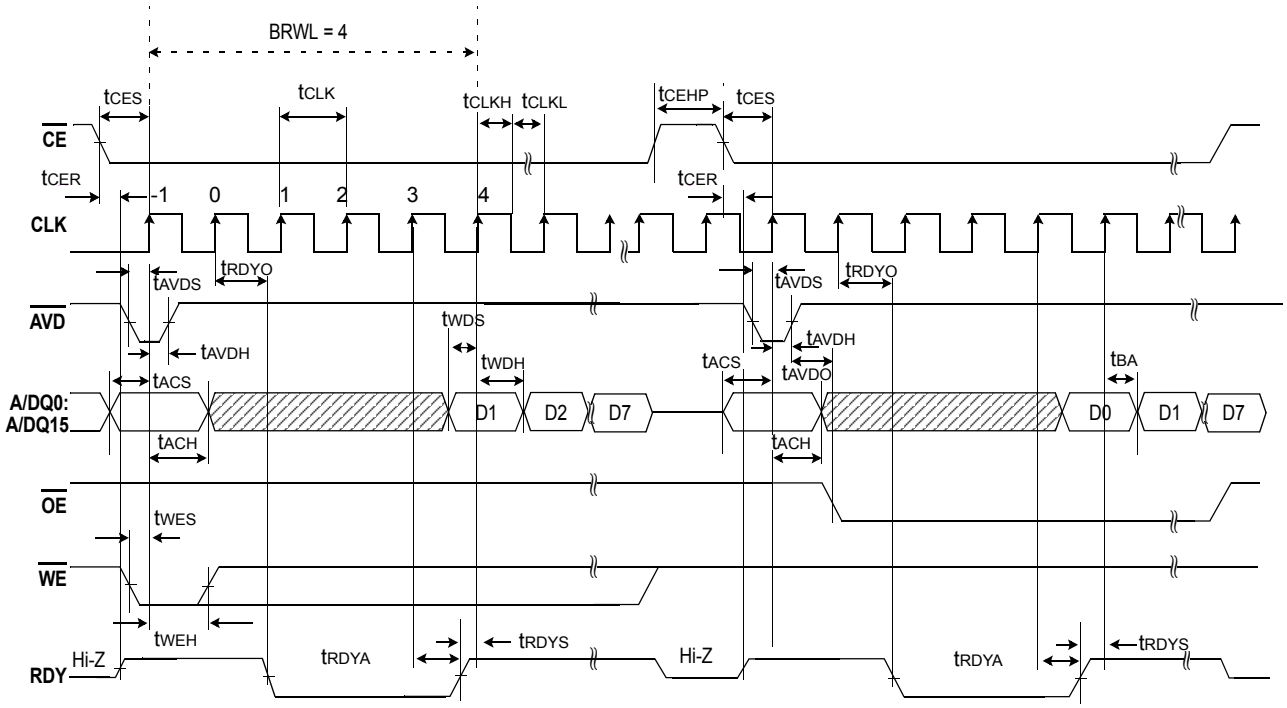
## 6.6 8-Word Linear Burst Write Mode

**See AC Characteristics Table 5.8**



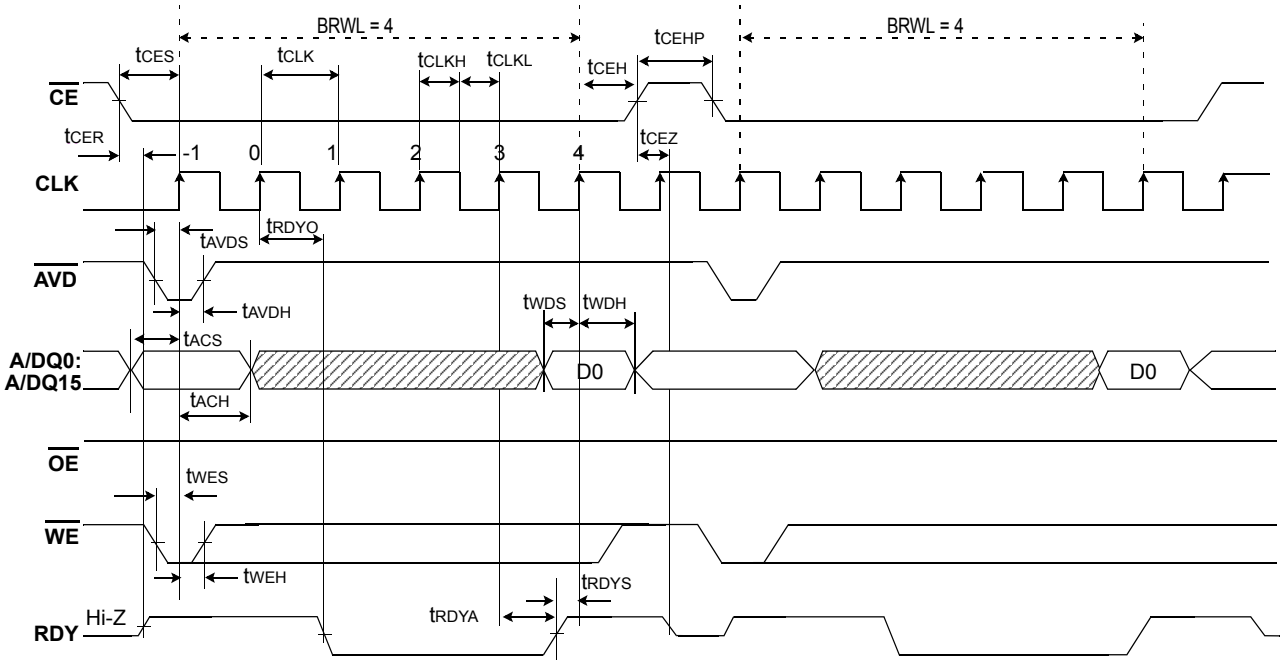

## 6.7 Burst Write Operation followed by Burst Read

**See AC Characteristics Table 5.8**



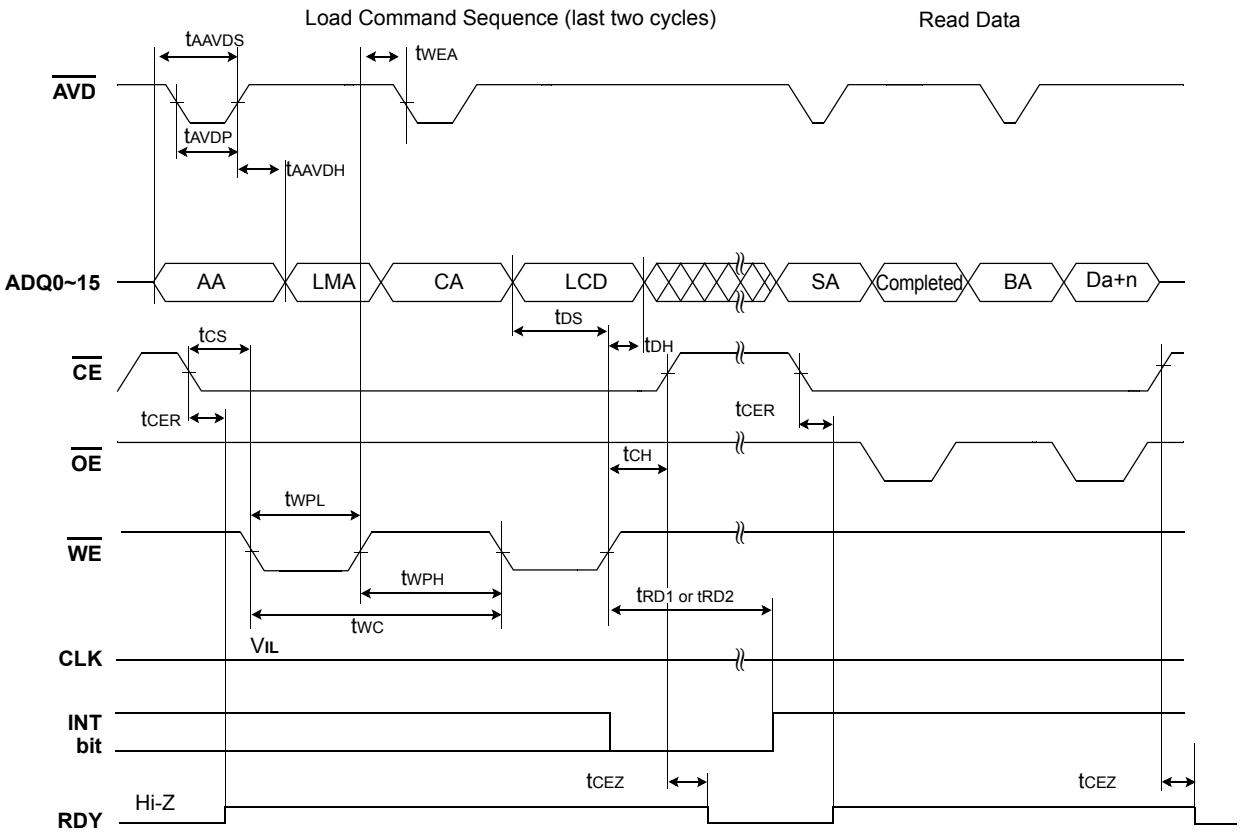
## 6.8 Start Initial Burst Write Operation

**See AC Characteristics Table 5.8**



## 6.9 Load Operation Timing

**See AC Characteristics Table 5.7 and Table 5.9**

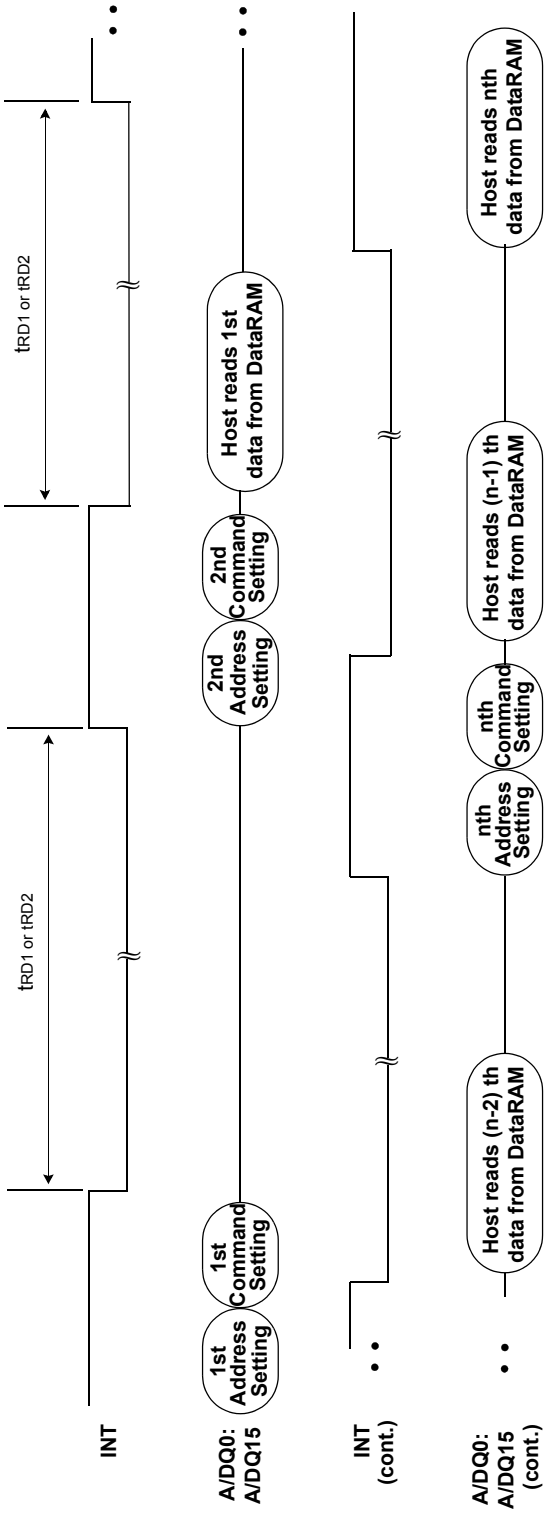



**NOTE :**
1) AA = Address of address register
CA = Address of command register
LCD = Load Command
LMA = Address of memory to be loaded
BA = Address of BufferRAM to load the data
SA = Address of status register
2) “In progress” and “complete” refer to status register
3) Status reads in this figure is asynchronous read, but status read in synchronous mode is also supported.

## 6.10 Superload Operation Timing

**See AC Characteristics Table 5.7 and Table 5.9**

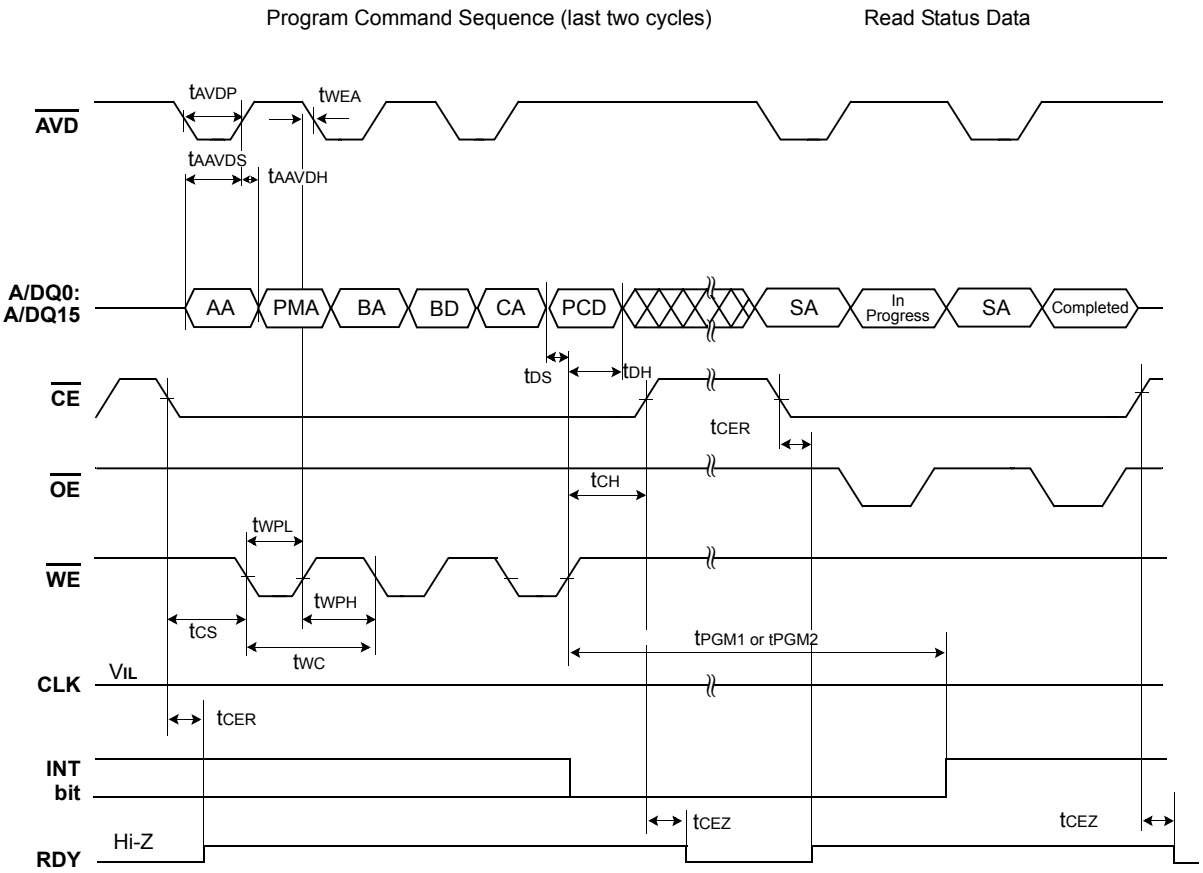
**Superload Operation Timing Diagram**




- 1st~nth Address Setting : Address Setting Operation for first page load(DFS, FBA, DBS and FPA).
- 1st Command Setting : 0000h, 2nd~nth Command Setting : 0003h
  (In INT auto mode, writing 0 to Interrupt register may be ignored)
- In case of page transfer, sensing time is tRD2, in case of sector transfer, sensing time is tRD1.
- Host read 1st~nth data from DataRAM : Host can read data from DataRAM only synchronus read mode.

## 6.11 Program Operation Timing

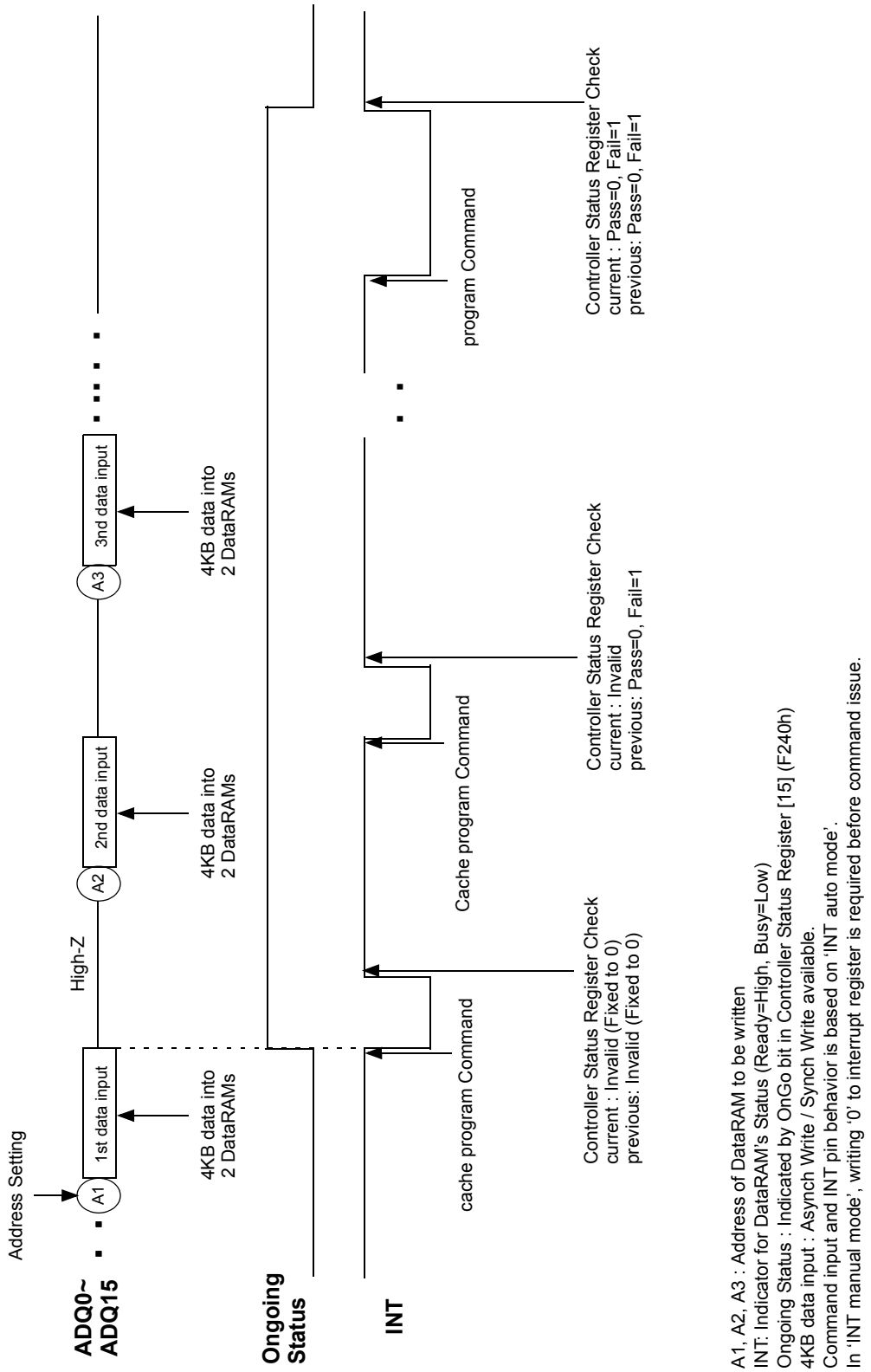
**See AC Characteristics Table 5.7 and Table 5.9**




**NOTE :**
1) AA = Address of address register
CA = Address of command register
PCD = Program Command
PMA = Address of memory to be programmed
BA = Address of BufferRAM to write the data
BD = Program Data
SA = Address of status register
2) “In progress” and “complete” refer to status register
3) Status reads in this figure is asynchronous read, but status read in synchronous mode is also supported.

## 6.12 Cache Program Operation Timing

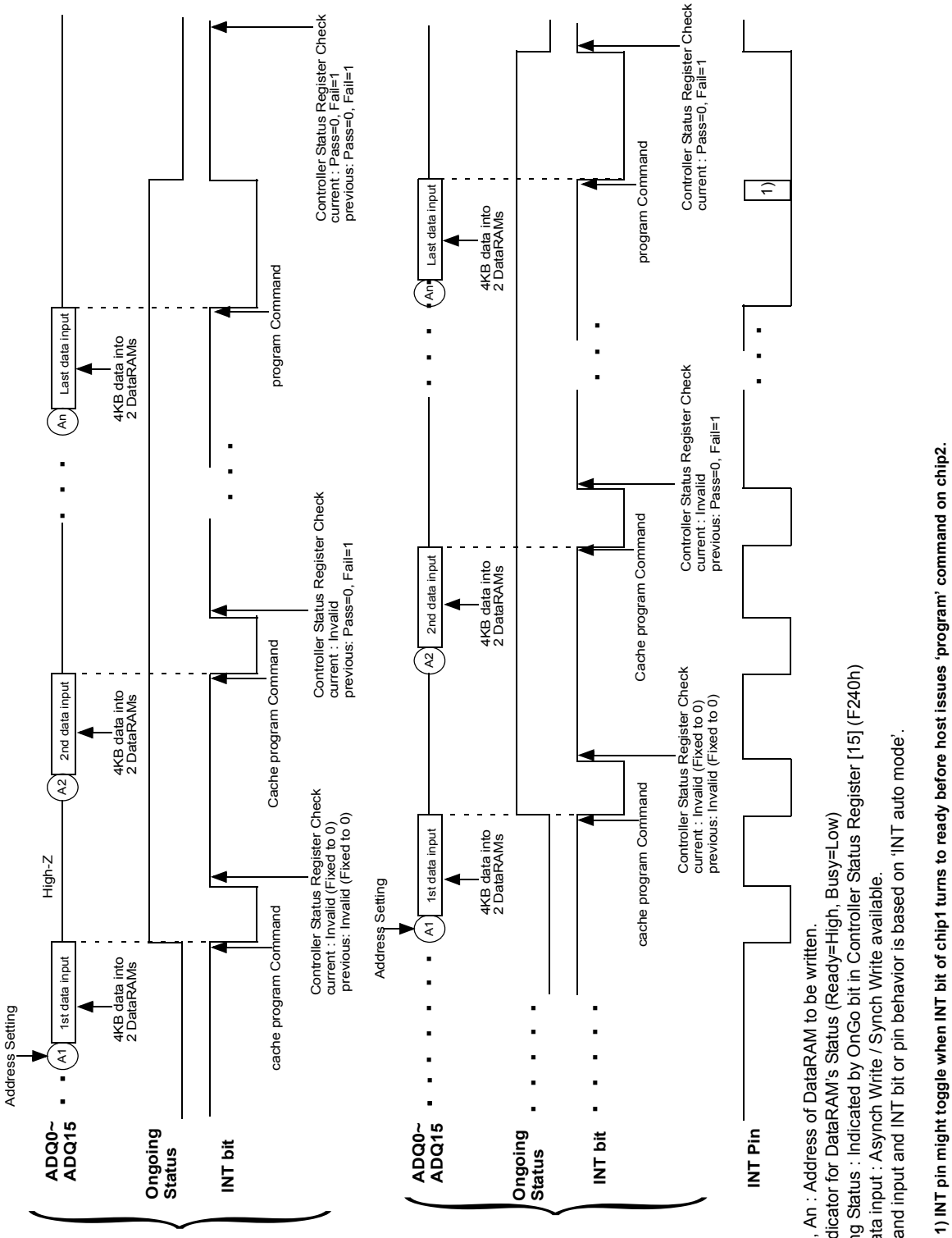
**See AC Characteristics Table 5.7 and Table 5.9**




A1, A2, A3 : Address of DataRAM to be written
INT: Indicator for DataRAM's Status (Ready=High, Busy=Low)
Ongoing Status : Indicated by OnGo bit in Controller Status Register [15] (F240h)
4KB data input : Asynch Write / Synch Write available.
Command input and INT pin behavior is based on 'INT auto mode'.
In 'INT manual mode', writing '0' to interrupt register is required before command issue.

## 6.13 Interleave Cache Program Operation Timing

**See AC Characteristics Table 5.7 and Table 5.9**

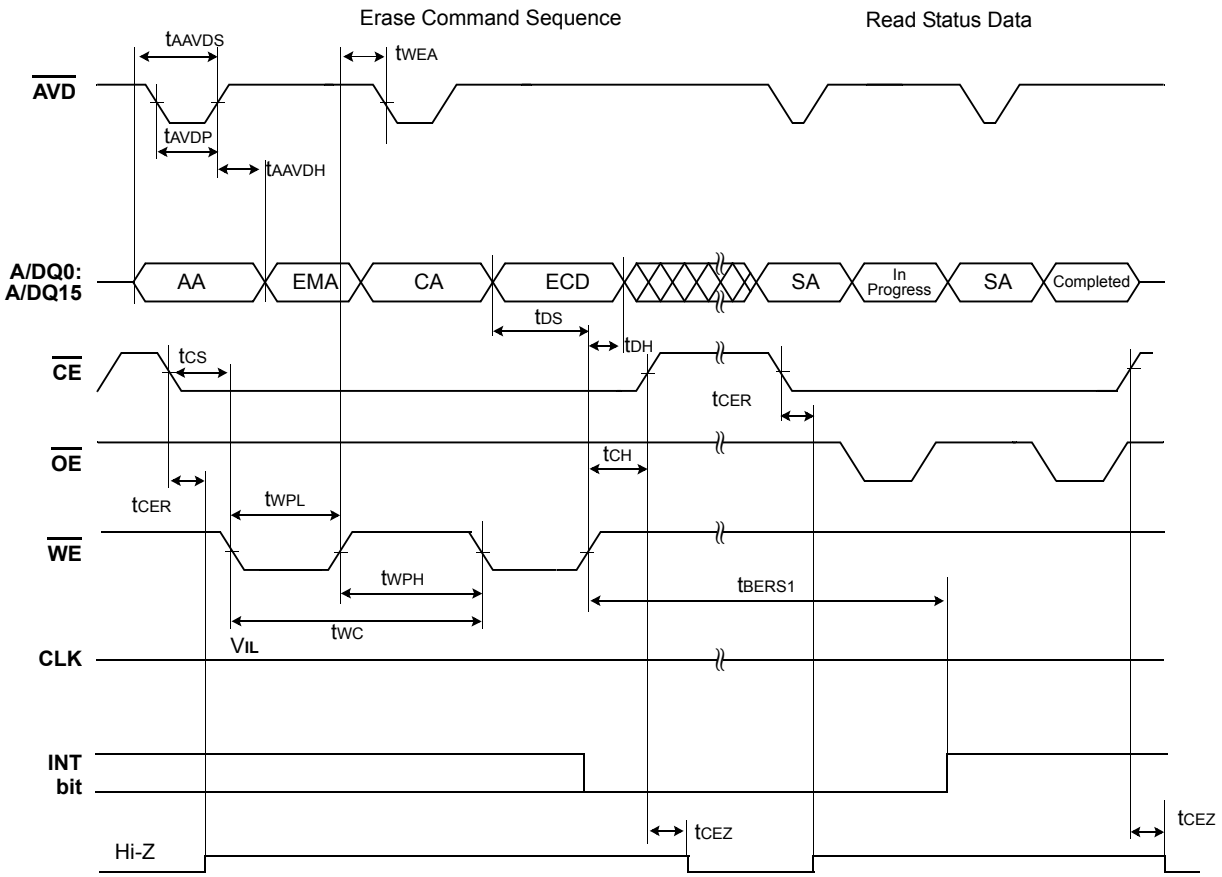



A1, A2, An : Address of DataRAM to be written.
INT: Indicator for DataRAM's Status (Ready=High, Busy=Low)
Ongoing Status : Indicated by OnGo bit in Controller Status Register [15] (F240h)
4KB data input : Asynch Write / Synch Write available.
Command input and INT bit or pin behavior is based on 'INT auto mode'.

**NOTE : 1) INT pin might toggle when INT bit of chip1 turns to ready before host issues 'program' command on chip2.**

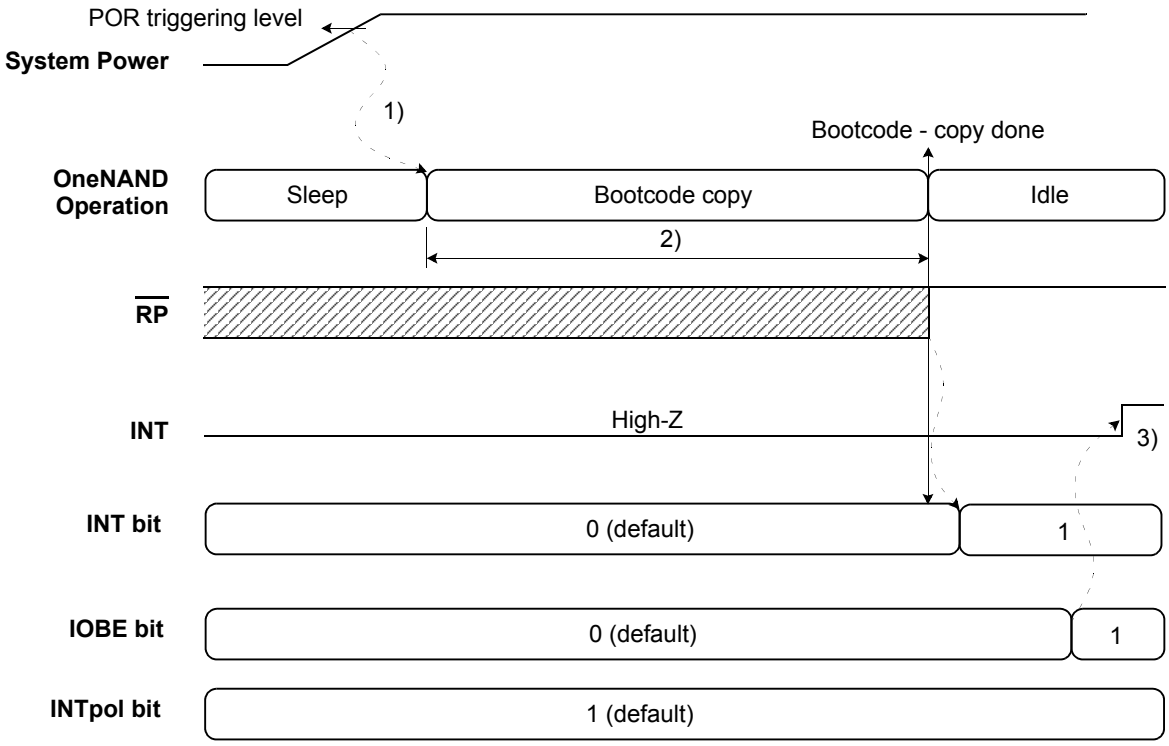
## 6.14 Block Erase Operation Timing

**See AC Characteristics Table 5.7 and Table 5.9**




**NOTE :**
1) AA = Address of address register
CA = Address of command register
ECD = Erase Command
EMA = Address of memory to be erased
SA = Address of status register
2) For “In progress” and “complete” status, refer to status register.
3) Status reads in this figure is asynchronous read, but status read in synchronous mode is also supported.

## 6.15 Cold Reset Timing

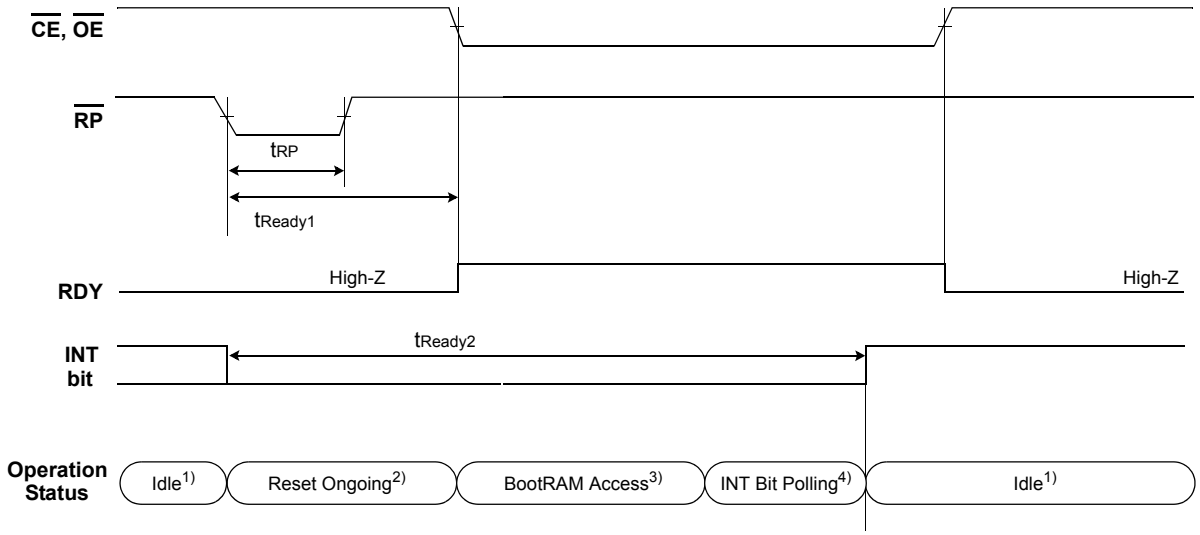



**NOTE :**

1) Bootcode copy operation starts 400us later than POR activation.
   The system power should reach Vcc after POR triggering level(typ. 1.5V) within 400us for valid boot code data.
2) 1KB Bootcode copy and internal update operation take 250us(estimated) from sector0 and 1/page0/block0 of NAND Flash array to BootRAM.
   Host can read Bootcode in BootRAM(1K bytes) after Bootcode copy completion.
3) INT register goes 'Low' to 'High' on the condition of 'Bootcode-copy done' and $\overline{RP}$ rising edge.
   If $\overline{RP}$ goes 'Low' to 'High' before 'Bootcode-copy done', INT register goes to 'Low' to 'High' as soon as 'Bootcode-copy done'

## 6.16 Warm Reset Timing

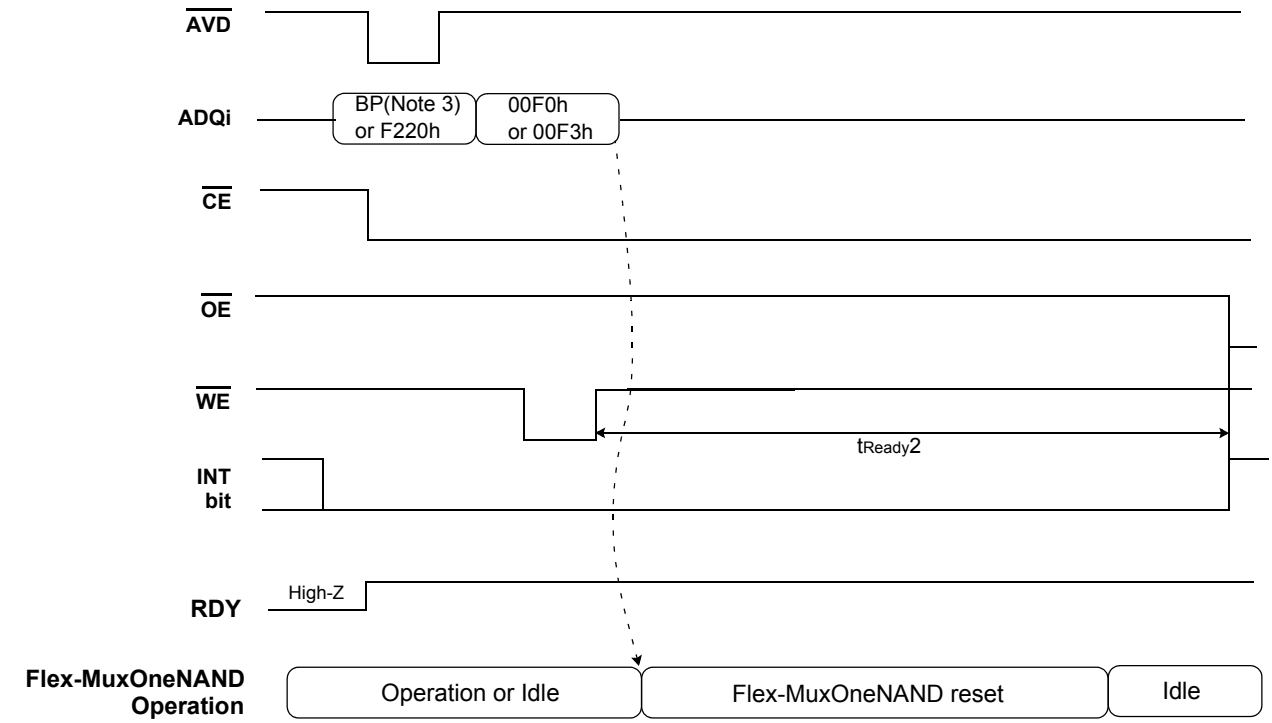
**See AC Characteristics Table 5.6**




**NOTE :**
1) The status which can accept any register based operation(Load, Program, Erase command, etc).
2) The status where reset is ongoing.
3) The status allows only BootRAM(BL1) read operation for Boot Sequence.(Refer to 7.2.2 Boot Sequence)
4) To read BL2 of Boot Sequence, Host should wait INT until becomes ready. and then, Host can issue load command.
(Refer to 7.2.2 Boot Sequence, 7.1 Methods of Determining Interrupt status)

## 6.17 Hot Reset Timing

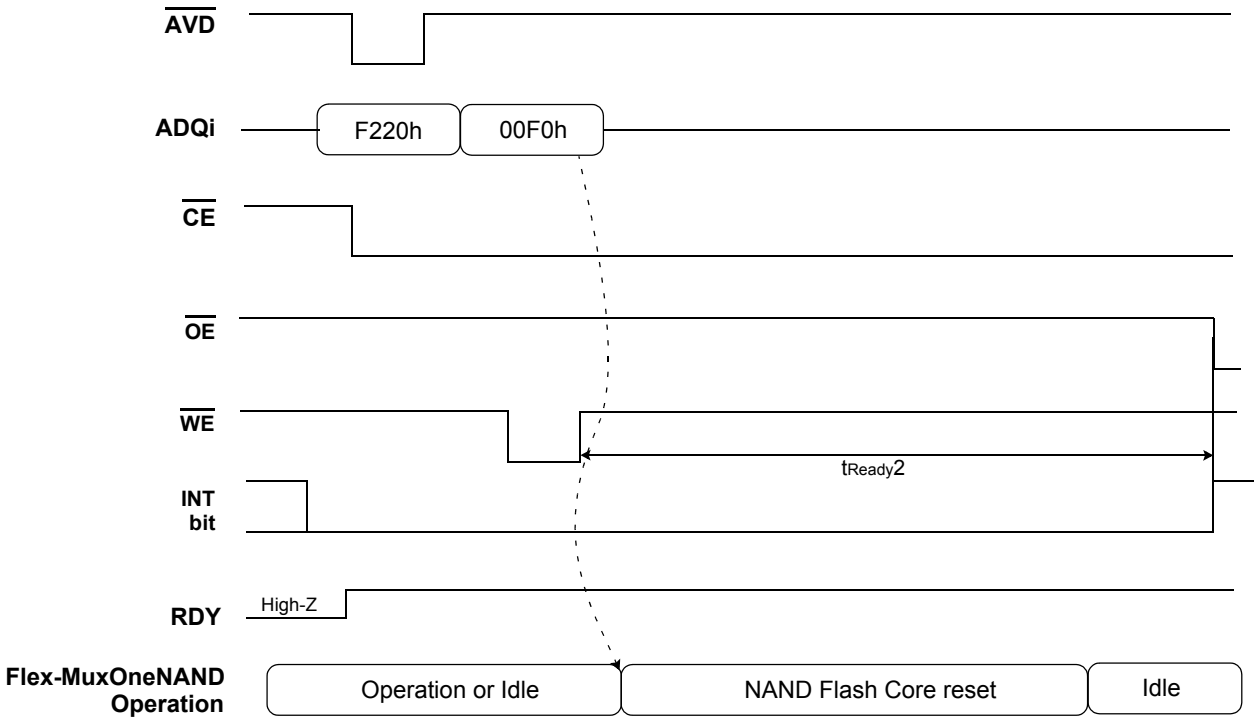
**See AC Characteristics Table 5.6**




**NOTE :**
1) Internal reset operation means that the device initializes internal registers and makes output signals go to default status and bufferRAM data are kept unchanged after Warm/Hot reset operations.
2) Reset command : Command based reset or Register based reset.
3) BP(Boot Partition): BootRAM area [0000h~01FFh, 8000h~800Fh]
4) 00F0h for BP, and 00F3h for F220h

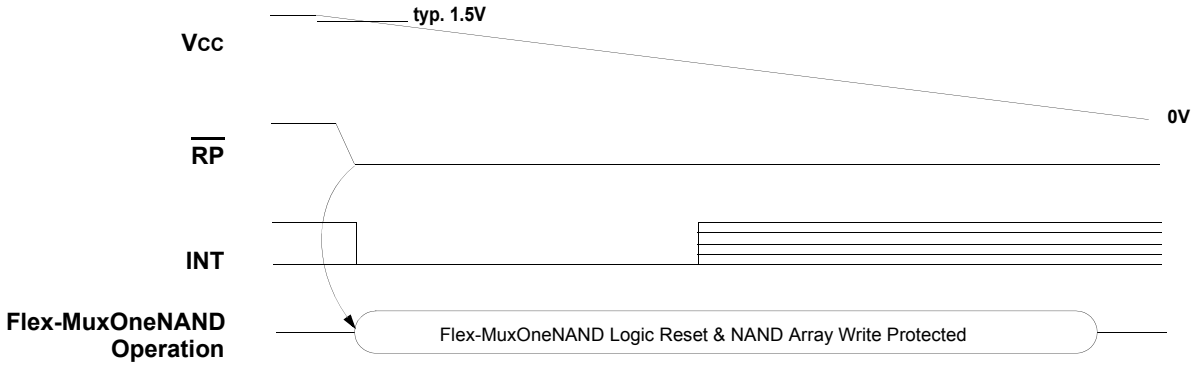
## 6.18 NAND Flash Core Reset Timing

**See AC Characteristics Table 5.6**



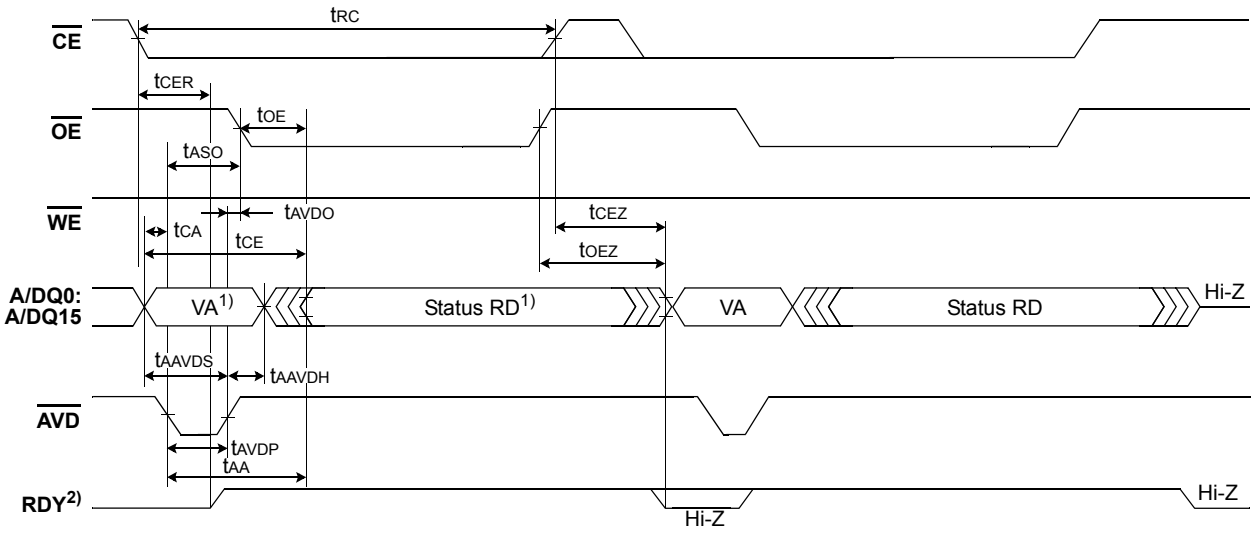

## 6.19 Data Protection Timing During Power Down

The device is designed to offer protection from any involuntary program/erase during power-transitions. $\overline{RP}$ pin provides hardware protection and is recommended to be kept at $V_{IL}$ before Vcc drops to 1.5V



## 6.20 Toggle Bit Timing in Asynchronous Read (VA Transition Before $\overline{AVD}$ Low)

**See AC Characteristics Table 5.5**

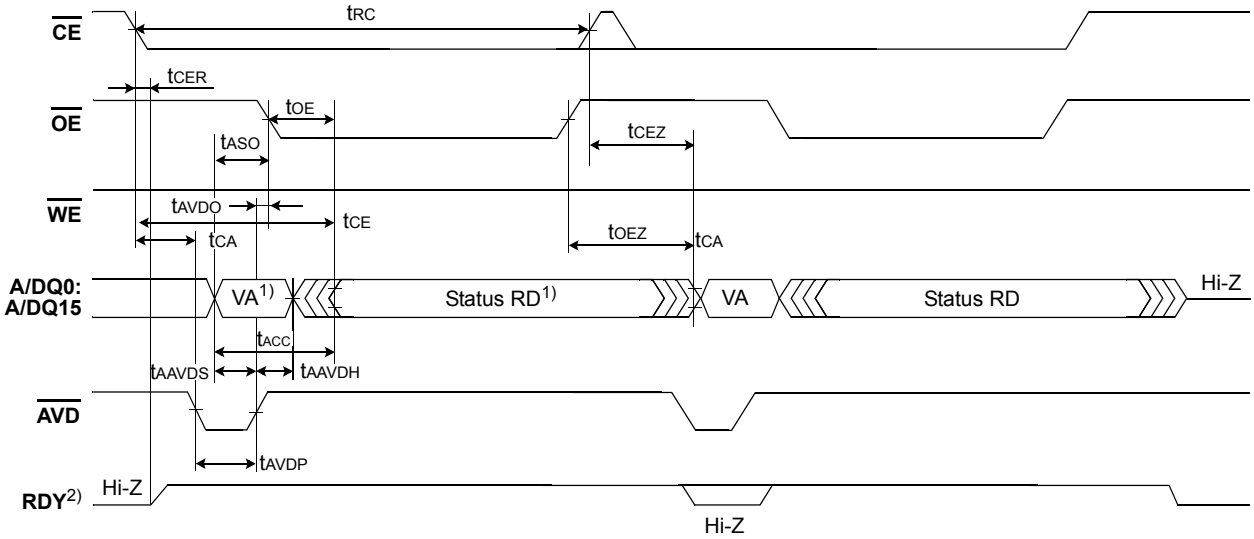



Note :

**NOTE :**
1) VA=Valid Read Address, RD=Read Data.
2) Before IOBE is set to 1, RDY and INT pin are High-Z state.
3) Refer to chapter 5.5 for tASO description and value.

## 6.21 Toggle Bit Timing in Asynchronous Read (VA Transition After $\overline{AVD}$ Low)

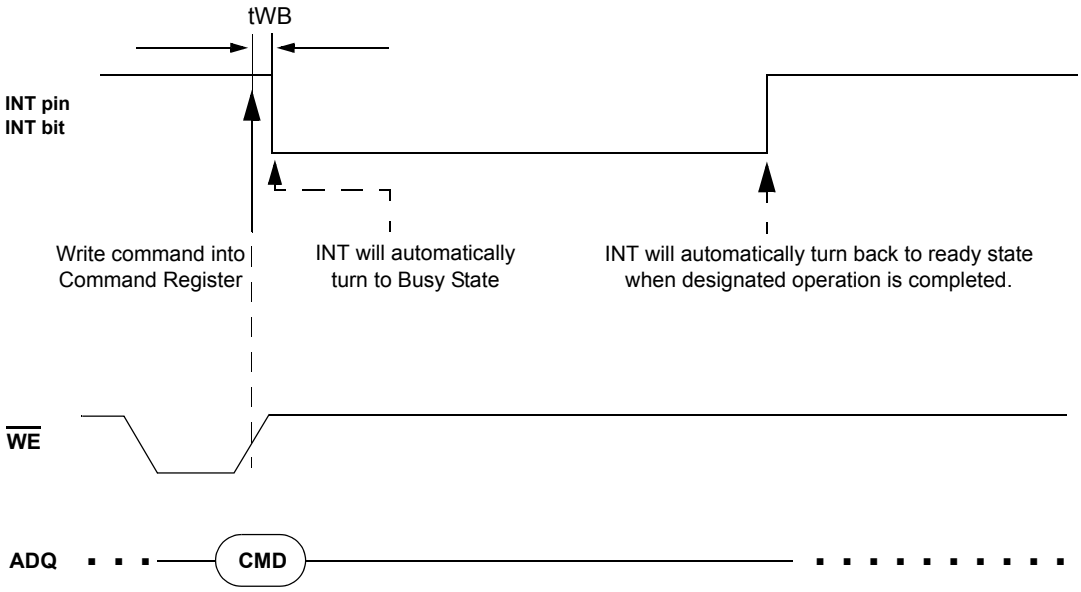
**See AC Characteristics Table 5.5**




**NOTE :**
1) VA=Valid Read Address, RD=Read Data.
2) Before IOBE is set to 1, RDY and INT pin are High-Z state.
3) Refer to chapter 5.5 for tASO description and value.

## 6.22 INT auto mode

**See AC Characteristics Table 5.10.**




**NOTE :**
1) INT pin polarity is based on 'IOBE=1 and INT pol=1 (default)' setting

# 7.0 TECHNICAL AND APPLICATION NOTES

From time-to-time supplemental technical information and application notes pertaining to the design and operation of the device in a system are included in this section. Contact your Samsung Representative to determine if additional notes are available.

## 7.1 Methods of Determining Interrupt Status

There are two methods of determining Interrupt Status on the Flex-MuxOneNAND. Using the INT pin or monitoring the Interrupt Status Register Bit.

The Flex-MuxOneNAND INT pin is an output pin function used to notify the Host when a command has been completed. This provides a hardware method of signaling the completion of a program, erase, or load operation.

In its normal state, the INT pin is high if the INT polarity bit is default. In case of normal INT mode, before a command is written to the command register, the INT bit must be written to '0' for the INT pin transitions to a low state indicating start of the operation. In case of 'INT auto mode', INT bit is written to '0' automatically right after command issued. Upon completion of the command operation by the Flex-MuxOneNAND's internal controller, INT returns to a high state.

INT pin is a DQ-type output except 'Reset' and 'Interleave Cache program' in DDP allowing two INT outputs to be Or-tied together. In case of 'Reset' and 'Interleave Cache Program' in DDP, INT pin operates as an open drain with 50K ohm. INT is an INT does not float to a hi-Z condition when $\overline{CE}$ is disabled or $\overline{OE}$ is disabled. Refer to section 2.8 for additional information about INT.

INT can be implemented by tying INT to a host GPIO or by continuous polling of the Interrupt status register.

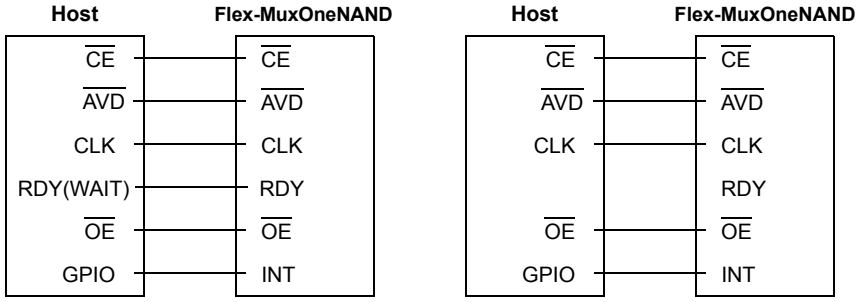
| | INT Type (Mono) | INT Type (DDP) |
|---|---|---|
| General Operation | DQ type | DQ type |
| Reset Operation (Cold,Warm,Hot and Flash Core Reset) and final command of Interleave cache program | DQ type | Open drain (with 50K ohm) |

### 7.1.1 The INT Pin to a Host General Purpose I/O

INT can be tied to a Host GPIO to detect the rising edge of INT, signaling the end of a command operation.


This can be configured to operate either synchronously or asynchronously as shown in the diagrams below.

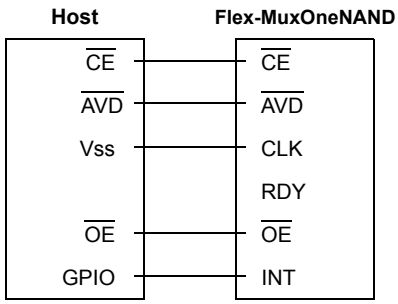
**Synchronous Mode Using the INT Pin**
When operating synchronously, INT is tied directly to a Host GPIO. RDY could be connected as one of following guides.




Handshaking Mode

Non-Handshaking Mode

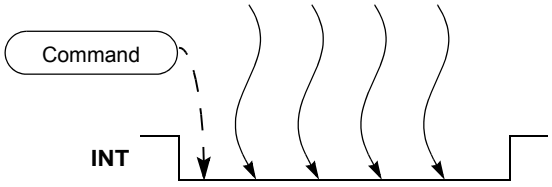
**Asynchronous Mode Using the INT Pin**
When configured to operate in an asynchronous mode, $\overline{CE}$, $\overline{AVD}$ and $\overline{OE}$ of the Flex-MuxOneNAND are tied to corresponding pins of the Host. CLK is tied to the Host Vss (Ground). RDY is tied to a no-connect. $\overline{OE}$ of the Flex-MuxOneNAND and Host are tied together and INT is tied to a GPIO.



### 7.1.2 Polling the Interrupt Register Status Bit

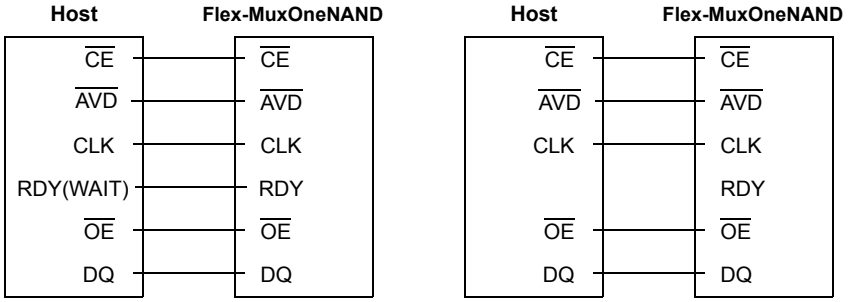
An alternate method of determining the end of an operation is to continuously monitor the Interrupt Status Register Bit instead of using the INT pin.
When using interrupt register instead of INT pin, INT must be unconnected

.

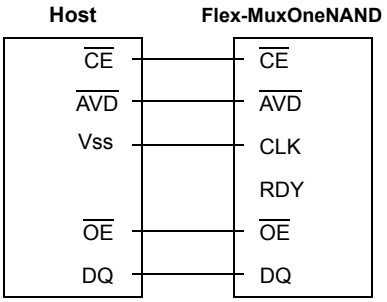



This can be configured in either a synchronous mode or an asynchronous mode.

**Synchronous Mode Using Interrupt Status Register Bit Polling**
When operating synchronously, $\overline{CE}$, $\overline{AVD}$, CLK, RDY, $\overline{OE}$, and DQ pins on the host and Flex-MuxOneNAND are tied together.
RDY could be connected as one of following guides.



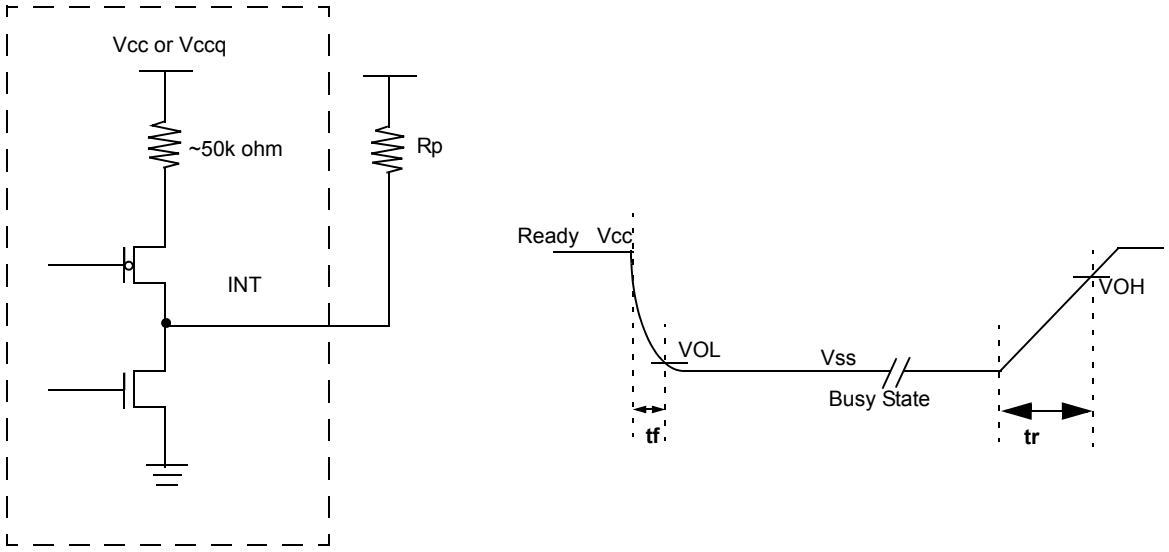

**Asynchronous Mode Using Interrupt Status Register Bit Polling**
When configured to operate in an asynchronous mode, $\overline{CE}$, $\overline{AVD}$, $\overline{OE}$ and DQ of the Flex-MuxOneNAND are tied to corresponding pins of the Host. CLK is tied to the Host Vss (Ground). RDY is NOT connected.

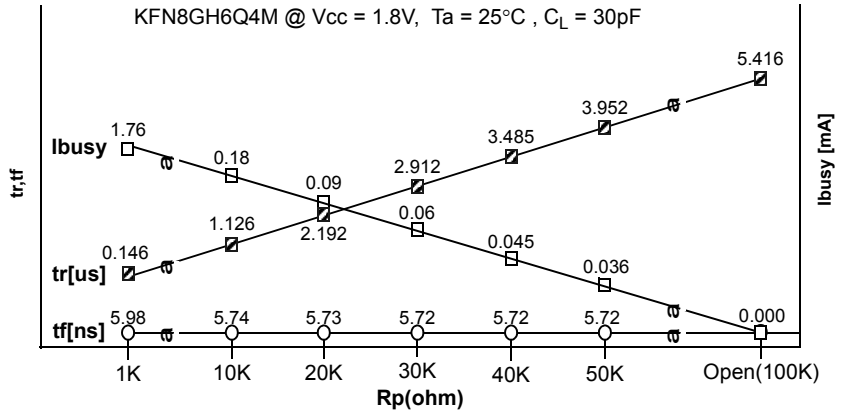


### 7.1.3 Determining Rp Value (DDP, QDP Only)

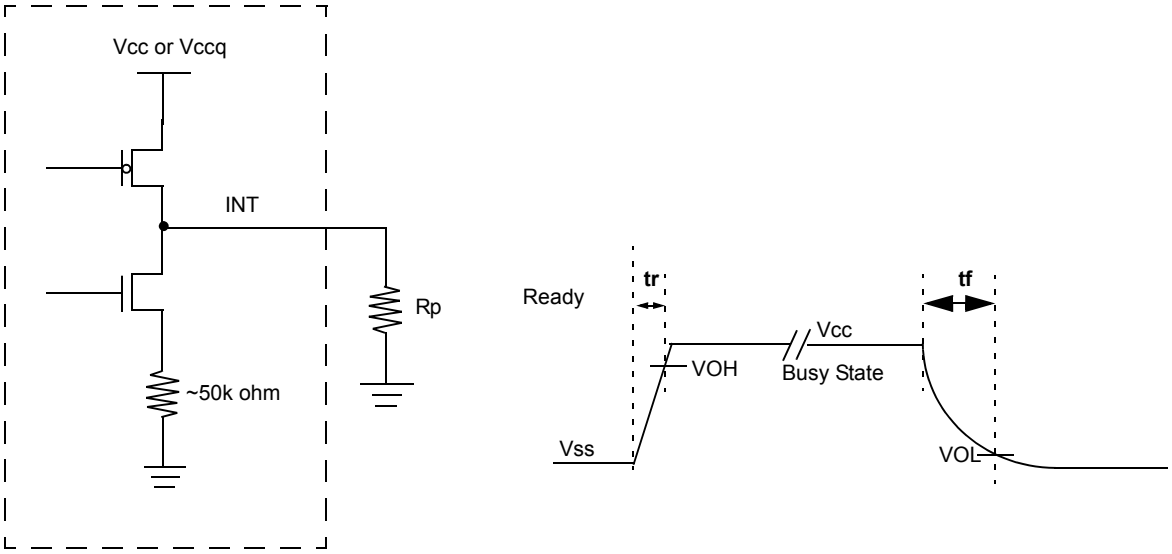
For general operation, INT operates as normal output pin, so that tF is equivalent to tR (below 10ns). But since INT operates as open drain with 50K ohm for Reset (Cold/Hot/Warm/NAND Flash Core) operations and 'Cache program operation' case at DDP option, the pull-up resistor value is related to tr(INT). And appropriate value can be obtained with the following reference charts.

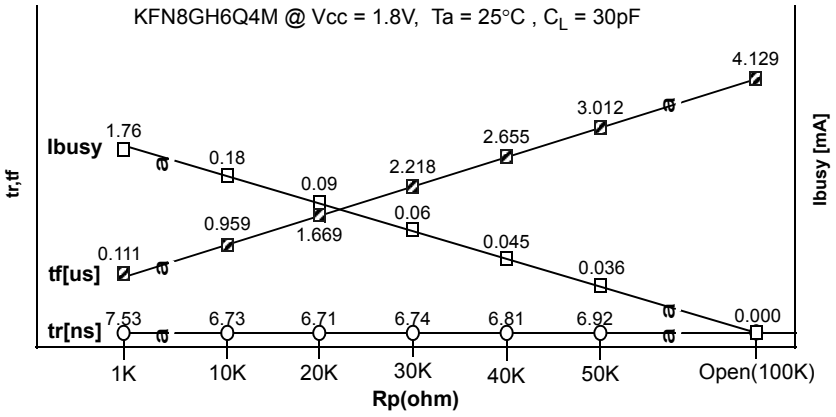
**INT pol = 'High' (Default)**



## INT pol = 'Low'


Vcc or Vccq
INT
Rp
~50k ohm
Ready
tr
tf
Vcc
VOH
Busy State
Vss
VOL


KFN8GH6Q4M @ Vcc = 1.8V, Ta = 25°C , C_L = 30pF
tr,tf
Ibusy [mA]
Ibusy
tf[us]
tr[ns]
1.76
0.18
0.09
2.218
2.655
3.012
4.129
0.111
0.959
1.669
0.06
0.045
0.036
0.000
7.53
6.73
6.71
6.74
6.81
6.92
1K
10K
20K
30K
40K
50K
Open(100K)
Rp(ohm)

## 7.2 Boot Sequence

One of the best features Flex-MuxOneNAND has is that it can be a booting device itself since it contains an internally built-in boot loader despite the fact that its core architecture is based on NAND Flash. Thus, Flex-MuxOneNAND does not make any additional booting device necessary for a system, which imposes extra cost or area overhead on the overall system.

As the system power is turned on, the boot code originally stored in the first block which is SLC area is moved to BootRAM automatically and then fetched by CPU through the same interface as SRAM's or NOR Flash's if the size of the boot code is less than 1KB. If its size is larger than 1KB and less than or equal to 3KB, only 1KB of it can be moved to BootRAM automatically and fetched by CPU, and the rest of it can be loaded into one of the DataRAMs whose size is 2KB by Load Command and CPU can take it from the DataRAM after finishing the code-fetching job for BootRAM. If its size is larger than 3KB, the 1KB portion of it can be moved to BootRAM automatically and fetched by CPU, and its remaining part can be moved to DRAM through two DataRAMs and taken by CPU to reduce CPU fetch time.

A typical boot scheme usually used to boot the system with Flex-MuxOneNAND is explained at Partition of NAND Flash Array and Flex-MuxOneNAND Boot Sequence. In this boot scheme, boot code is comprised of BL1, where BL stands for Boot Loader, BL2, and BL3. Moreover, the size of the boot code is larger than 3KB (the 3rd case above). BL1 is called primary boot loader in other words. Here is the table of detailed explanations about the function of each boot loader in this specific boot scheme.

### 7.2.1 Boot Loaders in Flex-MuxOneNAND

**Boot Loaders in Flex-MuxOneNAND**

| Boot Loader | Description |
|---|---|
| BL1 | Moves BL2 from NAND Flash Array to DRAM through two DataRAMs |
| BL2 | Moves OS image (or BL3 optionally) from NAND Flash Array to DRAM through two DataRams |
| BL3 (Optional) | Moves or writes the image through USB interface |

NAND Flash Array of Flex-MuxOneNAND is divided into the partitions as described at Partition of NAND Flash Array to show where each component of code is located and how much portion of the overall NAND Flash Array each one occupies. In addition, the boot sequence is listed below and depicted at Boot Sequence.

### 7.2.2 Boot Sequence

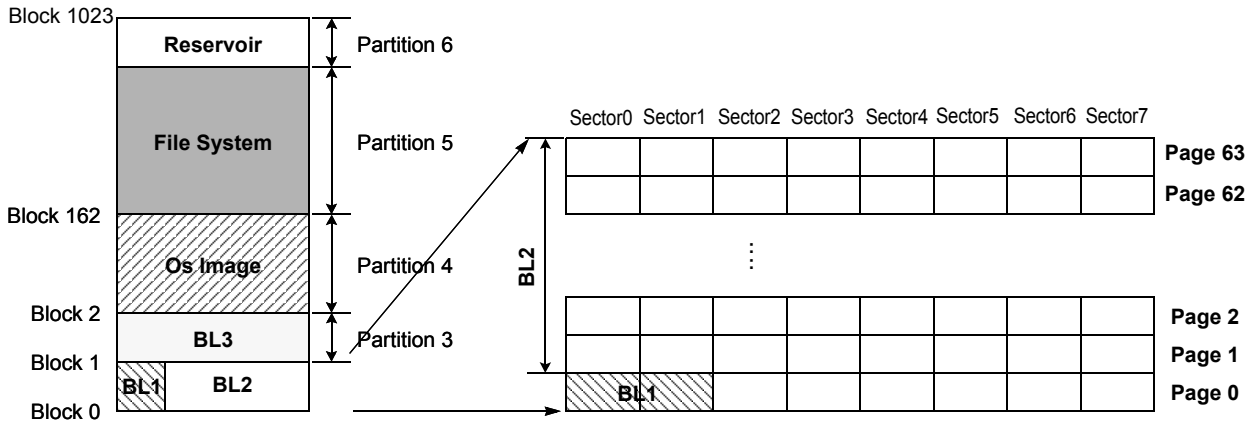
Boot Sequence :

1. Power is on
   BL1 is loaded into BootRAM

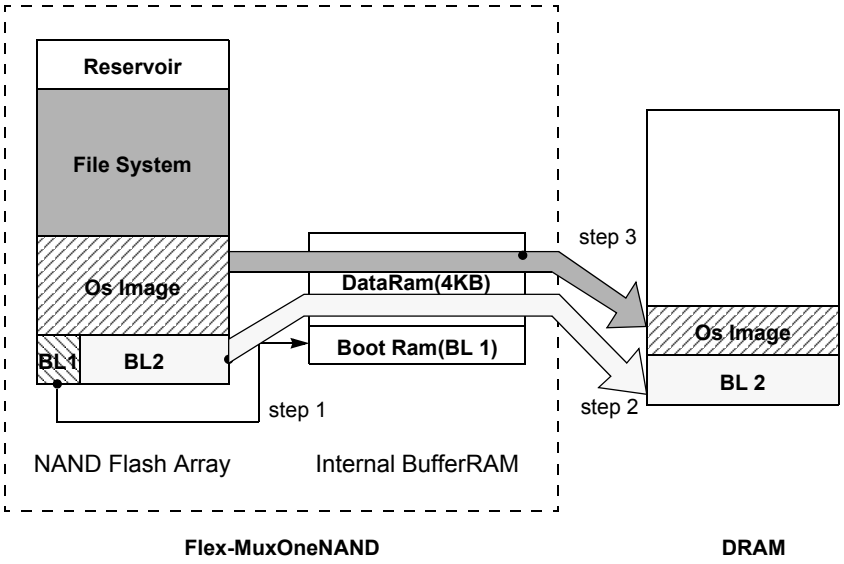
2. BL1 is executed in BootRAM
   BL2 is loaded into DRAM through two DataRams by BL1

3. BL2 is executed in DRAM
   OS image is loaded into DRAM through two DataRams by BL2

4. OS is running




**Partition of NAND Flash array**



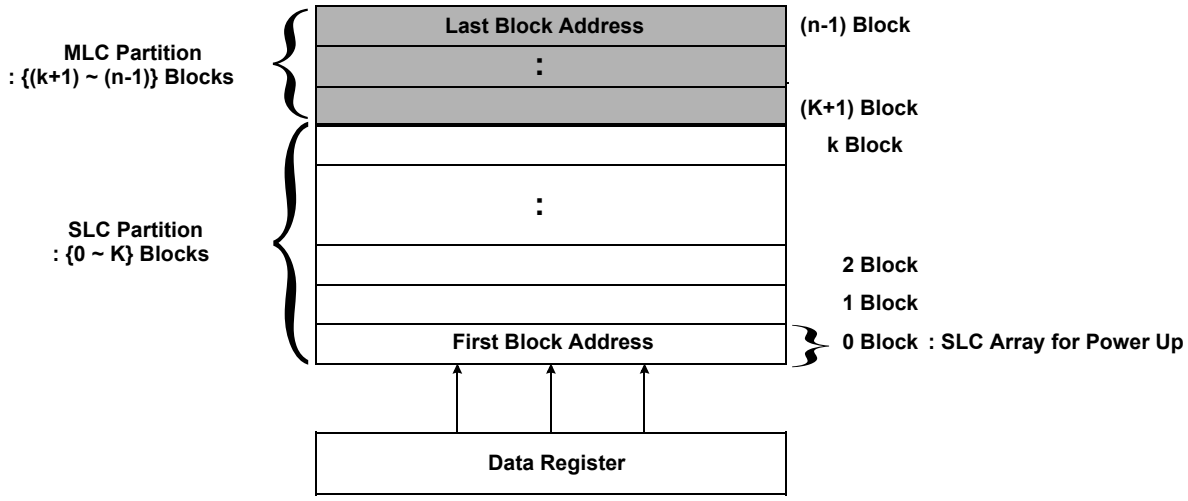

**Flex-MuxOneNAND Boot Sequence**

**NOTE : 1) Step 2 and Step 3 can be copied into DRAM through two DataRAMs.**

## 7.3 Partition of Flex-MuxOneNAND

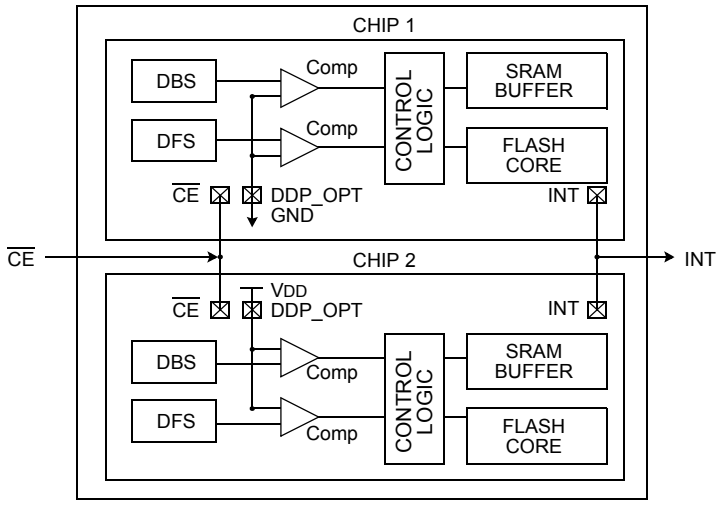
Flex-MuxOneNAND is the combo device which has SLC and MLC partition in one-chip. Generally, write intensive data require SLC-reliability but read and density oriented data such as music and movie satisfy with MLC-reliability. Therefore, some of the mobile phone is using both SLC-reliability memory and MLC-reliability memory to meet both requirements of reliability oriented and density oriented data using separate chip. but in case of Flex-MuxOneNAND, only one-chip of Flex-MuxOneNAND can meet the both requirement.

And also, Flex-MuxOneNAND has internal Error Correction Circuit and common NOR interface. User can take full advantage of advances in NAND Flash memory capacity.




**NOTE :**
1) K is the boundary address(the end of SLC). Samsung will decide the value of K before final specification open (K=TBD).
2) For the partitionning method, samsung will support application note and guidance code.

## 7.4 DDP and QDP Description




**DDP(Dual Die Package):**

8Gb DDP Flex-OneNAND contains two chips of 4Gb which are multiplexed such that they provide a single address range interface, with double the storage capacity.

Since the address range is single, the BootRAM, the bufferRAM and the register set are multiplexed.

BootRAM: The bootRAM of chip1 is selected always, and the contents of the block 0 of chip1 are copied to it at startup.

DataRAM: DBS setting in Start Address2 Register(See Section 2.8.10) decides which DataRAM is selected.

Register Set: In the case of write, both registers in chip1 and chip2 will be written(Regardless of DBS). Reading out from register of chip1/chip2 follows the DBS setting(See Section 2.8.10).

**QDP(Quad Die Package):**

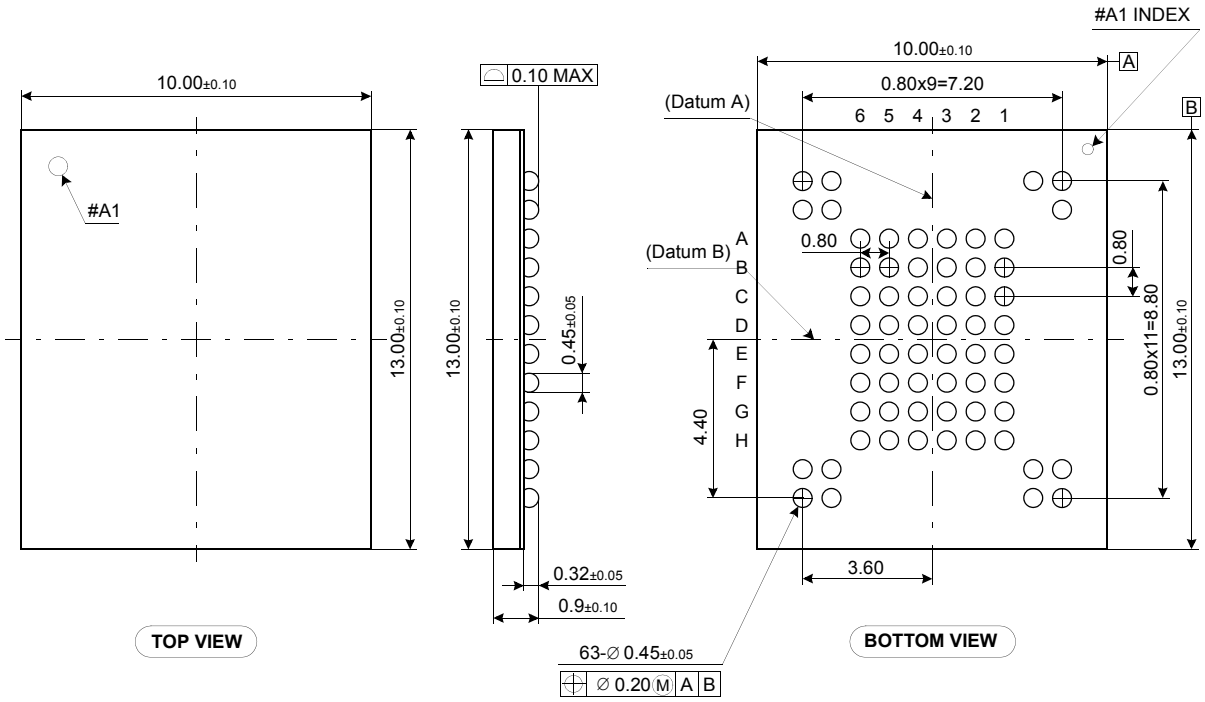
A QDP is made up of 2 DDP chips and is effectively 2 separate Flex-OneNAND devices in the same die.

There are 2 chip select pins (CE1 and CE2) on a QDP device, using which one of the two devices can be selected.

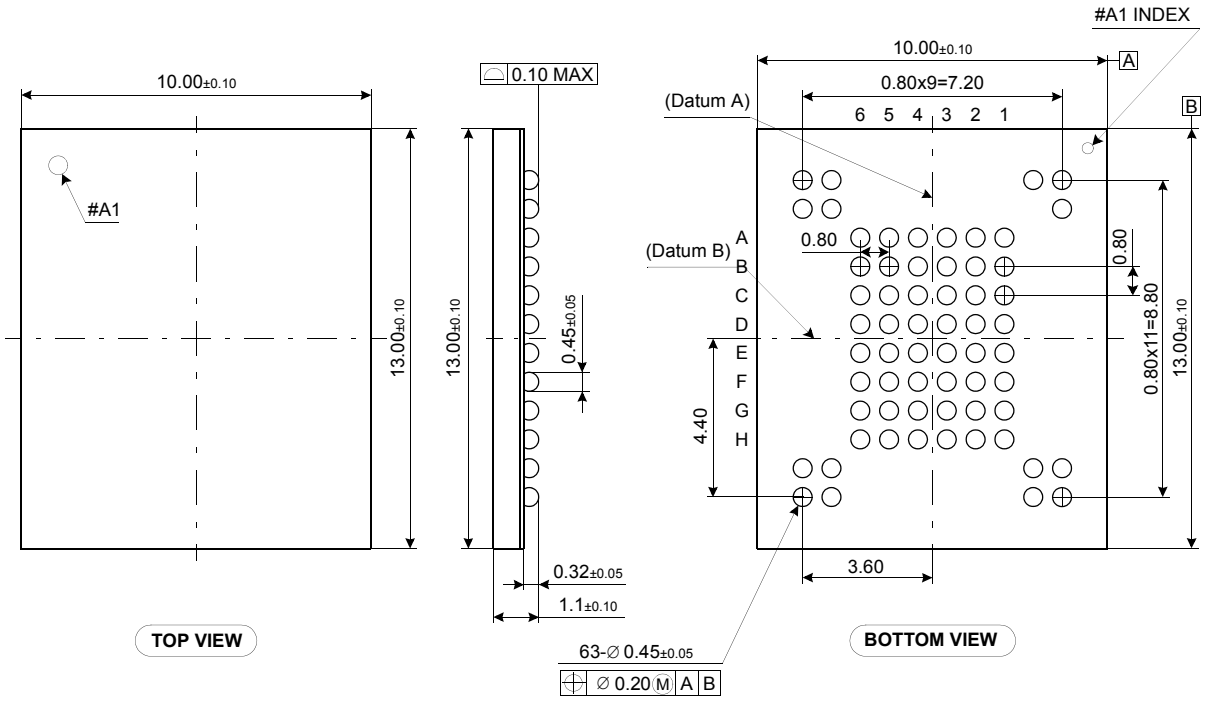
Since there are separate chip-selects for the two devices, they have different address ranges and register sets which can be directly accessed by the processor.

Thus each of the registers/BufferRAMs can be selected by using the CE pin, and then using the same settings that apply to a DDP chip.

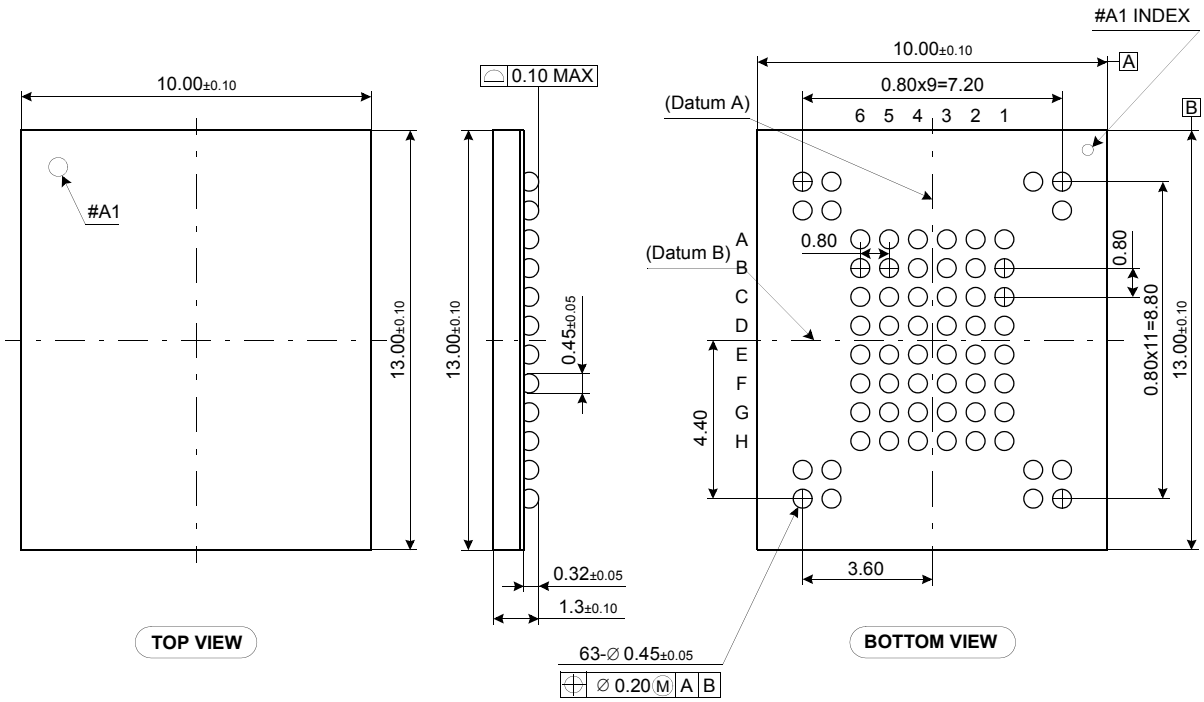
# 8.0 PACKAGE DIMENSIONS (TBD)




**4G product (KFM4GH6Q4M)**




**8G product (KFN8GH6Q4M)**




**16G product (KFKAGH6Q4M)**