

**OneNAND SPECIFICATION**

| Density | Part No.       | Vcc(core & IO)   | Temperature | PKG        |
|---------|----------------|------------------|-------------|------------|
| 512Mb   | KFG1216Q2M-DEB | 1.8V(1.7V~1.95V) | Extended    | 63FBGA(LF) |
|         | KFG1216D2M-DEB | 2.65V(2.4V~2.9V) | Extended    | 63FBGA(LF) |
|         | KFG1216U2M-DIB | 3.3V(2.7V~3.6V)  | Industrial  | 63FBGA(LF) |
| 1Gb     | KFH1G16Q2M-DEB | 1.8V(1.7V~1.95V) | Extended    | N/A        |

**Version: Ver. 1.4**

**Date: June 15th, 2005**

**INFORMATION IN THIS DOCUMENT IS PROVIDED IN RELATION TO SAMSUNG PRODUCTS, AND IS SUBJECT TO CHANGE WITHOUT NOTICE. NOTHING IN THIS DOCUMENT SHALL BE CONSTRUED AS GRANTING ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IN SAMSUNG PRODUCTS OR TECHNOLOGY. ALL INFORMATION IN THIS DOCUMENT IS PROVIDED ON AS "AS IS" BASIS WITHOUT GUARANTEE OR WARRANTY OF ANY KIND.**

1. For updates or additional information about Samsung products, contact your nearest Samsung office.
2. Samsung products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where Product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.

OneNAND™, is a trademark of Samsung Electronics Company, Ltd. Other names and brands may be claimed as the property of their rightful owners.

Copyright © 2005, Samsung Electronics Company, Ltd

Document Title

OneNAND

Revision History

| <u>Revision No.</u> | <u>History</u>   | <u>Draft Date</u> | <u>Remark</u> |
|---------------------|--|-------------------|---------------|
| 0.0                 | Initial issue.   | Jan. 07, 2004     | Preliminary   |
| 0.0.1               | 1. Add the "Invalid block management" and "Error management in read and write operation"<br>2. Add the restriction in addressing for program operation.<br>3. Add the asynchronous write and latched asynchronous write mode timing diagram.<br>4. Define new parameters in asynchronous write mode.<br>-tCH1 : 10ns, tCH2 : 0ns   | Jan. 29, 2004     | Preliminary   |
| 0.0.2               | 1. Add the dual operation diagram.<br>2. Add the block replacement diagram   | Jan. 30, 2004     | Preliminary   |
| 0.0.3               | 1. Edit the block replacement diagram<br>2. Add the 3.3V product.  | Feb. 03, 2004     | Preliminary   |
| 0.1                 | 1. Excluded Cache Program Operation<br>2. Added the descriptions for below operations<br>- Reset<br>- Write Protection<br>- Burst Read Latency<br>- Dual Operation<br>- Invalid block definition and Identification method<br>- Error in write or read operation<br>- ECC<br>3. Revised program sequence<br>4. Some AC parameters are changed.<br>tACH : 9ns-->7ns, tCES : 7ns-->9ns, tAAVDS : 5ns-->7ns<br>tDS : 30ns-->10ns, tDH : 0ns-->4ns<br>5. Define new AC parameter.<br>tAWES(Address hold time in AVD low case of asynchronous write mode)<br>Min. 0ns | Feb.11, 2004      | Preliminary   |
| 0.1.1               | 1. Correct an errata<br>Ball pitch of package is corrected.<br>0.5mm --> 0.8mm<br><br>2. Edit the timing diagram of burst read wrap around.(Figure 23,24)  | Mar.9, 2004       | Preliminary   |
| 0.2                 | 1. The specification of 2.7V device is added.  | Mar. 22, 2004     | Preliminary   |
| 0.3                 | 1. The specification of 3.3V device is deleted.<br>2. Correct some typos.  | Mar. 31, 2004     | Preliminary   |

The attached datasheets are prepared and approved by SAMSUNG Electronics. SAMSUNG Electronics CO., LTD. reserve the right to change the specifications. SAMSUNG Electronics will evaluate and reply to your requests and questions about device. If you have any questions, please contact the SAMSUNG branch office near you.

Document Title

OneNAND

Revision History

| <u>Revision No.</u> | <u>History</u>  | <u>Draft Date</u> | <u>Remark</u> |
|---------------------|---|-------------------|---------------|
| 0.4                 | <ol style="list-style-type: none"> <li>1. Corrected the errata</li> <li>2. Added spare assignment information in detail</li> <li>3. Added NAND array memory map</li> <li>4. Added manufacturer ID for CS as 00ECh</li> <li>5. Added stepping ID for CS in version ID register</li> <li>6. Divided default status of interrupt status register by Warm, Hot reset and Cold reset</li> <li>7. Revised Load operation flow chart</li> <li>8. Revised Program operation flow chart</li> <li>9. Deleted DBS setting step in Copy-back operation</li> <li>10. Added OTP description</li> <li>11. Revised OTP Load and Program flow chart</li> <li>12. Added INT guidance</li> <li>13. ECC description is revised</li> <li>14. Added Data Protection Scheme during Power-down</li> <li>15. Added DC/AC parameters</li> </ol>   | June 22, 2004     | Preliminary   |
| 1.0                 | <ol style="list-style-type: none"> <li>1. Deleted 2.7V product</li> <li>2. Added 2.65V product</li> <li>3. Added 3.3V product and industrial temperature in 3.3V product</li> <li>4. Deleted Unlock/Lock BootRAM command</li> <li>5. Added DBS setting step in Copy-back operation</li> <li>6. Added 2.65V/3.3V DC parameters</li> <li>7. Revised tCES from 9ns to 7ns</li> <li>8. Deleted tOEH in asynchronous read operation</li> <li>9. Revised NOP from 4 times per each main and spare in a page to 2 times per sector</li> <li>10. Revised Write Protection status description</li> <li>11. Added DDP selection and operation guidance</li> <li>12. Added 1Gb DDP device ID</li> <li>13. Added INT bit status in Cold Reset operation</li> <li>14. Moved Interrupt register setting before inputting command in all flow charts</li> <li>15. Revised Dual operation diagrams</li> <li>16. Added and revised the asynchronous read operation timing diagram</li> <li>17. Revised the asynchronous write operation timing diagram</li> <li>18. Added the tREADY parameter in Hot Reset operation</li> </ol> | August 5, 2004    | Final         |
| 1.1                 | <ol style="list-style-type: none"> <li>1. Revised standby current for DDP</li> </ol>  | August 26, 2004   | Final         |
| 1.2                 | <ol style="list-style-type: none"> <li>1. Corrected DDP device ID</li> <li>2. Excluded Commercial Temperature range</li> <li>3. Revised Cold Reset timing diagram</li> <li>4. Added CE and RDY in Warm Reset diagram</li> <li>5. Excluded Write while Load and Read while Program operation</li> <li>6. Revised Extended Temperature minimum value from -25 to -30</li> <li>7. Revised typical tOTP, tLOCK from 300us to 600us</li> <li>8. Revised max tOTP, tLOCK from 600us to 1000us</li> <li>9. Revised Icc4, Icc5 test condition</li> <li>10. Added Endurance and Data Retention</li> </ol>  | October 26, 2004  |               |

The attached datasheets are prepared and approved by SAMSUNG Electronics. SAMSUNG Electronics CO., LTD. reserve the right to change the specifications. SAMSUNG Electronics will evaluate and reply to your requests and questions about device. If you have any questions, please contact the SAMSUNG branch office near you.

Document Title

OneNAND

Revision History

| <u>Revision No.</u> | <u>History</u>  | <u>Draft Date</u> | <u>Remark</u> |
|---------------------|---|-------------------|---------------|
| 1.3                 | <ol style="list-style-type: none"> <li>1. Deleted Manufacturer ID for ES</li> <li>2. Excluded bit error case in Load operation</li> <li>3. Revised tWEA value from max to min</li> <li>4. Revised tRD1 typical value from 35us to 40us</li> <li>5. Revised tRD2 typical value from 75us to 85us</li> <li>6. Added technical note for OneNAND boot sequence</li> <li>7. Revised Asynchronous Read timing diagram for <math>\overline{CE}</math> don't care mode</li> <li>8. Revised Asynchronous Write timing diagram for <math>\overline{CE}</math> don't care mode</li> <li>9. Revised Load operation timing diagram for <math>\overline{CE}</math> don't care mode</li> </ol>   | Dec. 16, 2004     | Final         |
| 1.4                 | <ol style="list-style-type: none"> <li>1. Added Copyright Notice in the beginning</li> <li>2. Corrected Errata</li> <li>3. Updated Icc2, Icc4, Icc5, Icc6 and I<sub>SB</sub></li> <li>4. Revised INT pin description</li> <li>5. Changed default of Manufacturer ID Register with 00ECh</li> <li>6. Removed "or erase case, refer to the table 3" from descriptions of WB, EB</li> <li>7. Added OTP erase case NOTE</li> <li>8. Revised case definitions of Interrupt Status Register</li> <li>9. Added a NOTE to Command register</li> <li>10. Added ECClogSector Information table</li> <li>11. Removed 'data unit based data handling' from description of Device Operation</li> <li>12. Revised description on Warm/Hot/NAND Flash Core Reset</li> <li>13. Revised Warm Reset Timing</li> <li>14. Revised description for 4-, 8-, 16-, 32-Word Linear Burst Mode</li> <li>15. Revised OTP operation description</li> <li>16. Restored earlier text for OTP Programming</li> <li>17. Added supplemental explanation for ECC Operation</li> <li>18. Replaced "read" with "load" in ECC bypass</li> <li>19. Removed redundant sentence from ECC Bypass Operation</li> <li>20. Added technical note for INT pin connection guide</li> </ol> | Jun. 15, 2005     |               |

The attached datasheets are prepared and approved by SAMSUNG Electronics. SAMSUNG Electronics CO., LTD. reserve the right to change the specifications. SAMSUNG Electronics will evaluate and reply to your requests and questions about device. If you have any questions, please contact the SAMSUNG branch office near you.

## 1. FEATURES

### ◆ Architecture

- Design Technology: 0.12um
- Voltage Supply
  - 1.8V device(KFG1216Q2M) : 1.7V~1.95V
  - 2.65V device(KFG1216D2M) : 2.4V~2.9V
  - 3.3V device(KFG1216U2M) : 2.7V~3.6V
- Organization
  - Host Interface:16bit
- Internal BufferRAM(5K Bytes)
  - 1KB for BootRAM, 4KB for DataRAM
- NAND Array
  - Page Size : (2K+64)bytes
  - Block Size : (128K+4K)bytes

### ◆ Performance

- Host Interface type
  - Synchronous Burst Read
    - : Clock Frequency: up to 54MHz
    - : Linear Burst - 4, 8, 16, 32 words with wrap-around
    - : Continuous Sequential Burst(1K words)
  - Asynchronous Random Read
    - : Access time of 76ns
  - Asynchronous Random Write
- Programmable Read latency
- Multiple Sector Read
  - Read multiple sectors by Sector Count Register(up to 4 sectors)
- Reset Mode
  - Cold Reset / Warm Reset / Hot Reset / NAND Flash Reset
- Power dissipation (typical values)
  - Standby current : 10uA@1.8V, 20uA@2.65V/3.3V for single, 20uA@1.8V, 40uA@2.65V/3.3V for DDP
  - Synchronous Burst Read current(54MHz) : 12mA@1.8V device, 20mA@2.65V/3.3V device
  - Load current : 20mA@1.8V device, 25mA@2.65V/3.3V device
  - Program current: 20mA@1.8V device, 25mA@2.65V/3.3V device
  - Erase current: 15mA@1.8V device, 20mA@2.65V/3.3V device
- Reliable CMOS Floating-Gate Technology
  - Endurance : 100K Program/Erase Cycles
  - Data Retention : 10 Years

### ◆ Hardware Features

- Voltage detector generating internal reset signal from Vcc
- Hardware reset input ( $\overline{RP}$ )
- Data Protection
  - Write Protection for BootRAM
  - Write Protection mode for NAND Flash Array
  - Write protection during power-up
  - Write protection during power-down
- User-controlled One Time Programmable(OTP) area
- Internal 2bit EDC / 1bit ECC
- Internal Bootloader supports Booting Solution in system

### ◆ Software Features

- Handshaking Feature
  - INT pin: Indicates Ready / Busy of OneNAND
  - Polling method: Provides a software method of detecting the Ready / Busy status of OneNAND
- Detailed chip information by ID register

### ◆ Packaging

- Package
  - 63ball, 9.5mm x 12mm x max 1.0mm t , 0.8mm ball pitch FBGA

## 2. GENERAL DESCRIPTION

OneNAND is a single-die chip with standard NOR Flash interface using NAND Flash Array. This device is comprised of logic and NAND Flash Array and 5KB internal BufferRAM. 1KB BootRAM is used for reserving bootcode, and 4KB DataRAM is used for buffering data. The operating clock frequency is up to 54MHz. This device is X16 interface with Host, and has the speed of ~76ns random access time. Actually, it is accessible with minimum 4clock latency(host-driven clock for synchronous read), but this device adopts the appropriate wait cycles by programmable read latency. OneNAND provides the multiple sector read operation by assigning the number of sectors to be read in the sector counter register. The device includes one block sized OTP(One Time Programmable), which can be used to increase system security or to provide identification capabilities.

## 3. PIN DESCRIPTION

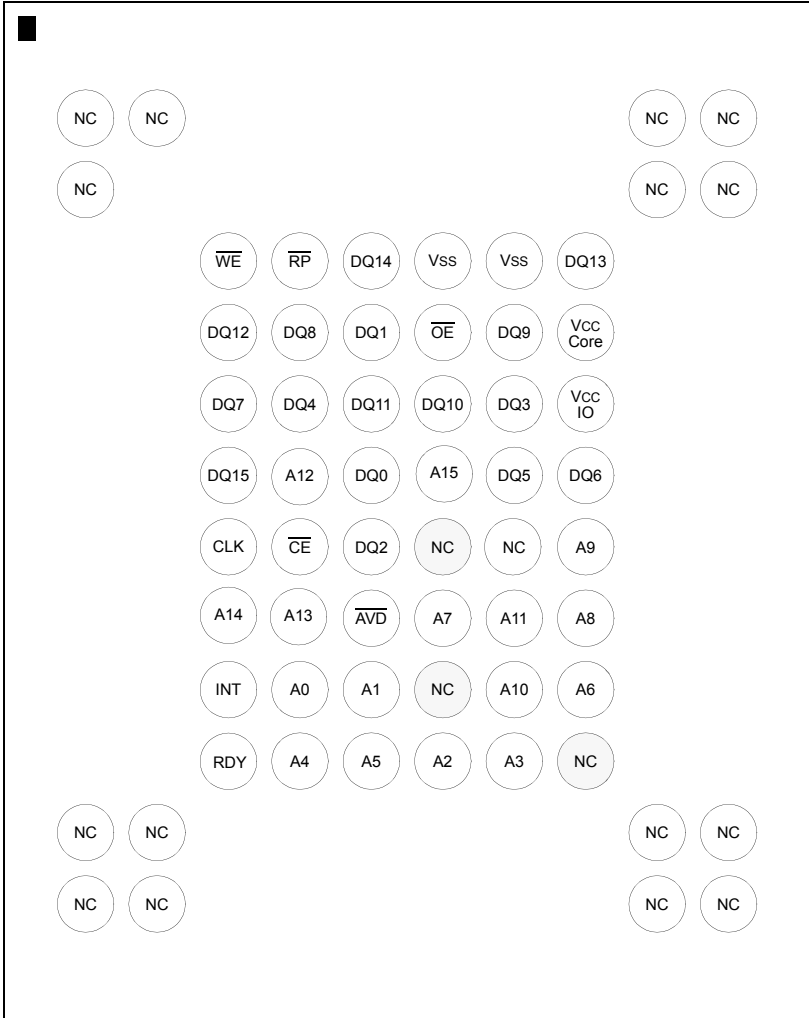
| Pin Name              | Type | Name and Description  |
|-----------------------|------|---|
| <b>Host Interface</b> |      |   |
| A15~A0                | I    | <b>Address Inputs</b><br>- Inputs for addresses during read operation, which are for addressing BufferRAM & Register.   |
| DQ15~DQ0              | I/O  | <b>Data Inputs/Outputs</b><br>- Inputs data during program and commands during 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                   | O    | <b>Interrupt</b><br>Notifying Host when a command has completed. It is open drain output with internal resistor (~50kohms). 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 the chip is deselected or when outputs are disabled.  |
| RDY                   | O    | <b>Ready</b><br>Indicates data valid in synchronous read modes and is activated while $\overline{CE}$ is low  |
| CLK                   | I    | <b>Clock</b><br>CLK synchronizes the device to the system bus frequency in synchronous read mode. The first rising edge of CLK in conjunction with $\overline{AVD}$ low latches address input.  |
| $\overline{WE}$       | I    | <b>Write Enable</b><br>$\overline{WE}$ controls writes to the bufferRAM and registers. Datas are latched on the $\overline{WE}$ pulse's rising edge   |
| $\overline{AVD}$      | I    | <b>Address Valid Detect</b><br>Indicates valid address presence on address inputs. During asynchronous read operation, all addresses are latched on $\overline{AVD}$ 's rising edge, 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    | <b>Reset Pin</b><br>When low, $\overline{RP}$ resets internal operation of OneNAND. $\overline{RP}$ status is don't care during power-up and bootloading.   |
| $\overline{CE}$       | I    | <b>Chip Enable</b><br>$\overline{CE}$ -low activates internal control logic, and $\overline{CE}$ -high deselected the device, places it in standby state, and places A/DQ in Hi-Z   |
| $\overline{OE}$       | I    | <b>Output Enable</b><br>$\overline{OE}$ -low enables the device's output data buffers during a read cycle.  |
| <b>Power Supply</b>   |      |   |
| Vcc-Core/Vcc          |      | <b>Power for OneNAND Core</b><br>This is the power supply for OneNAND Core.   |
| Vcc-IO/Vccq           |      | <b>Power for OneNAND I/O</b><br>This is the power supply for OneNAND I/O<br>Vcc-IO is internally connected to Vcc-Core, thus should be connected to the same power supply.  |
| Vss                   |      | <b>Ground for OneNAND</b>   |
| <b>etc.</b>           |      |   |
| DNU                   |      | <b>Do Not Use</b><br>Leave it disconnected. These pins are used for testing.  |
| NC                    |      | <b>No Connection</b><br>Lead is not internally connected.   |

**NOTE:**

Do not leave power supply(VCC, VSS) disconnected.



4. PIN CONFIGURATION

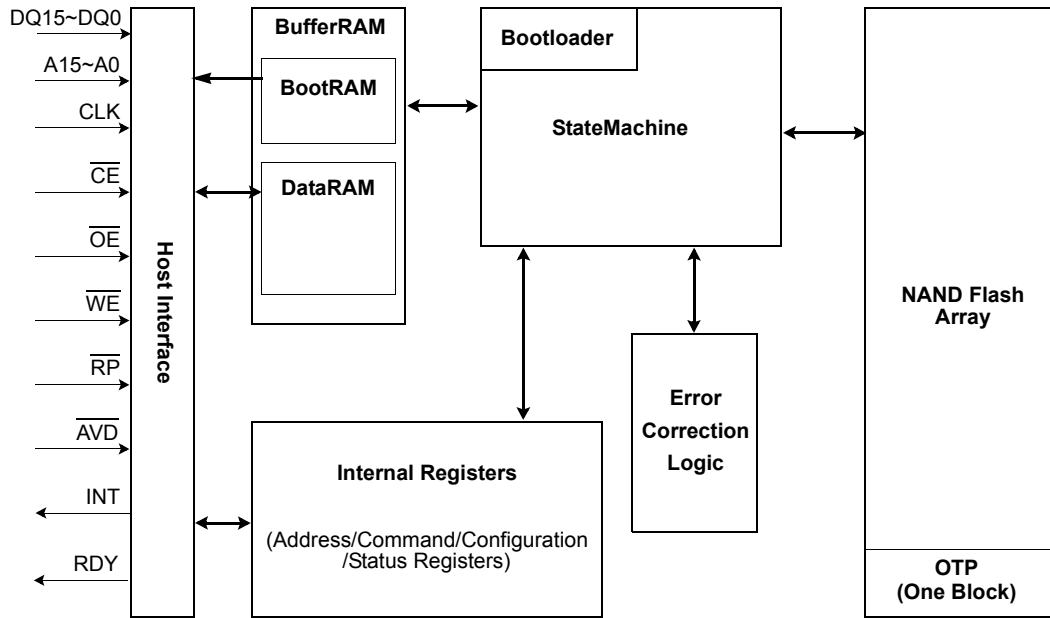


(TOP VIEW, Balls Facing Down)  
63ball FBGA OneNAND Chip  
63ball, 9.5mm x 12mm x max 1.0mmt , 0.8mm ball pitch FBGA

## DEFINITIONS

|                       |  |
|-----------------------|--|
| B (capital letter)    | Byte, 8bits  |
| W (capital letter)    | Word, 16bits   |
| b (lower-case letter) | Bit  |
| ECC                   | Error Correction Code  |
| Calculated ECC        | ECC which has been calculated during load or program access  |
| Written ECC           | ECC which 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 Bootcode buffering   |
| DataRAM               | A 4KB portion of the BufferRAM reserved for Data buffering   |
| Memory                | NAND Flash array which is embedded on OneNAND  |
| Sector                | Partial unit of page, of which size is 512B for main area and 16B for spare area data. It is the minimum Load/Program/Copy-Back program unit while one~four sector operation is available  |
| Data unit             | Possible data unit to be read from memory to BufferRAM or to be programmed to memory. <ul style="list-style-type: none"> <li>- 528B of which 512B is in main area and 16B in spare area</li> <li>- 1056B of which 1024B is in main area and 32B in spare area</li> <li>- 1584B of which 1536B is in main area and 48B in spare area</li> <li>- 2112B of which 2048B is in main area and 64B in spare area</li> </ul> |

5. BLOCK DIAGRAM



- Host Interface
- BufferRAM(BootRAM, DataRAM)
- Command and status registers
- State Machine (Bootloader is included)
- Error Correction Logic
- Memory(NAND Flash Array, OTP)

**NOTE:**

1) At cold reset, bootloader copies boot code(1K byte size) from NAND Flash Array to BootRAM.

Figure 1. Internal Block Diagram

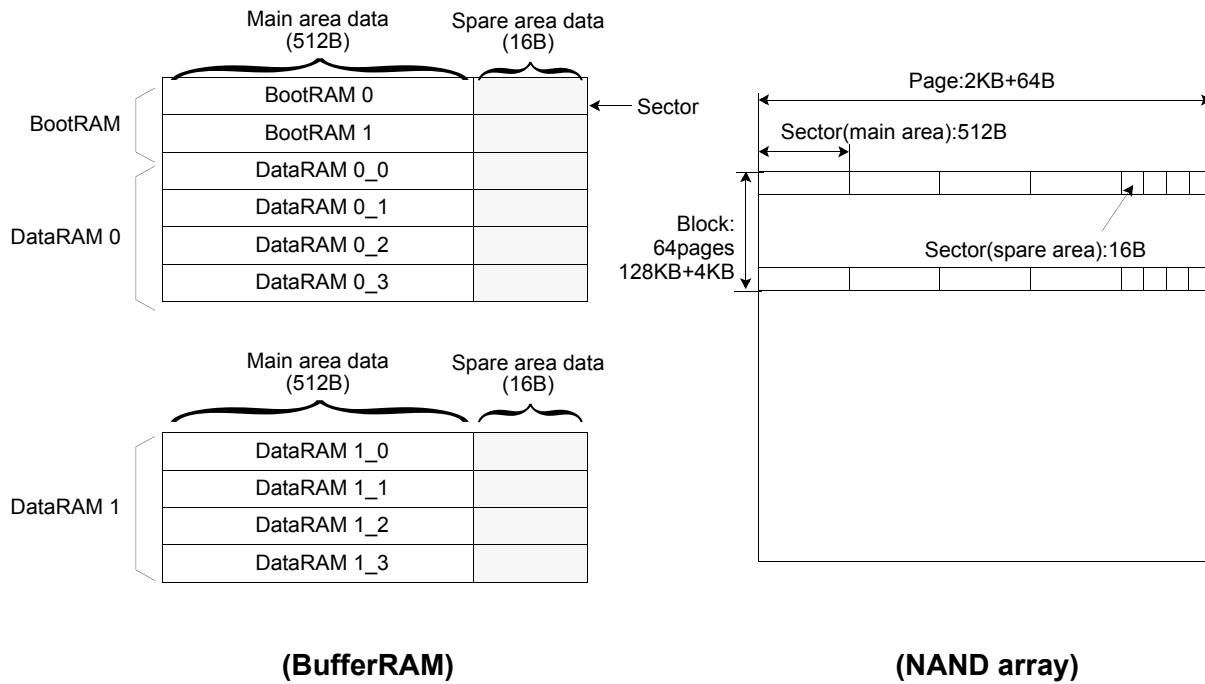
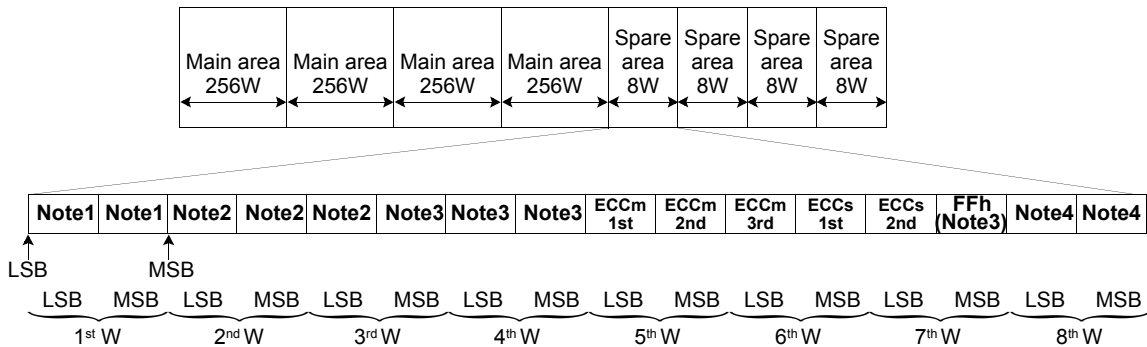


Figure 2. BufferRAM and NAND array structure



**NOTE:**

- 1) The 1st word of spare area in 1st and 2nd page of every invalid block is reserved for the invalid block information by manufacturer. Please refer to page 64 about the details.
- 2) These words are managed by internal ECC logic. So it is recommended that the important data like LSN(Logical Sector Number) are written.
- 3) These words are reserved for the future purpose by manufacturer. These words will be dedicated to internal logic.
- 4) These words are for free usage.
- 5) The 5th, 6th and 7th words are dedicated to internal ECC logic. So these words are only readable. The other words are programmable by command.
- 6) ECCm 1st, ECCm 2nd, ECCm 3rd: ECC code for Main area data
- 7) ECCs 1st, ECCs 2nd: ECC code for 2nd and 3rd word of spare area.

Figure 3. Spare area of NAND array assignment

6.1 ADDRESS MAP For OneNAND

| Division         | Address (word order) | Address (byte order) | Size (total 128KB) |          | Usage     | Description                 |
|------------------|----------------------|----------------------|--------------------|----------|-----------|-----------------------------|
| Main area (64KB) | 0000h~00FFh          | 00000h~001FEh        | 512B               | 1KB      | BootM 0   | BootRAM Main sector0        |
|                  | 0100h~01FFh          | 00200h~003FEh        | 512B               |          | BootM 1   | BootRAM Main sector1        |
|                  | 0200h~02FFh          | 00400h~005FEh        | 512B               | 4KB      | DataM 0_0 | DataRAM Main page0/sector0  |
|                  | 0300h~03FFh          | 00600h~007FEh        | 512B               |          | DataM 0_1 | DataRAM Main page0/sector1  |
|                  | 0400h~04FFh          | 00800h~009FEh        | 512B               |          | DataM 0_2 | DataRAM Main page0/sector2  |
|                  | 0500h~05FFh          | 00A00h~00BFEh        | 512B               |          | DataM 0_3 | DataRAM Main page0/sector3  |
|                  | 0600h~06FFh          | 00C00h~00DFEh        | 512B               |          | DataM 1_0 | DataRAM Main page1/sector0  |
|                  | 0700h~07FFh          | 00E00h~00FFEh        | 512B               |          | DataM 1_1 | DataRAM Main page1/sector1  |
|                  | 0800h~08FFh          | 01000h~011FEh        | 512B               |          | DataM 1_2 | DataRAM Main page1/sector2  |
|                  | 0900h~09FFh          | 01200h~013FEh        | 512B               |          | DataM 1_3 | DataRAM Main page1/sector3  |
|                  | 0A00h~7FFFh          | 01400h~0FFFEh        | 59KB               | 59KB     | Reserved  | Reserved                    |
| Spare area (8KB) | 8000h~8007h          | 10000h~1000Eh        | 16B                | 32B      | BootS 0   | BootRAM Spare sector0       |
|                  | 8008h~800Fh          | 10010h~1001Eh        | 16B                |          | BootS 1   | BootRAM Spare sector1       |
|                  | 8010h~8017h          | 10020h~1002Eh        | 16B                | 128B     | DataS 0_0 | DataRAM Spare page0/sector0 |
|                  | 8018h~801Fh          | 10030h~1003Eh        | 16B                |          | DataS 0_1 | DataRAM Spare page0/sector1 |
|                  | 8020h~8027h          | 10040h~1004Eh        | 16B                |          | DataS 0_2 | DataRAM Spare page0/sector2 |
|                  | 8028h~802Fh          | 10050h~1005Eh        | 16B                |          | DataS 0_3 | DataRAM Spare page0/sector3 |
|                  | 8030h~8037h          | 10060h~1006Eh        | 16B                |          | DataS 1_0 | DataRAM Spare page1/sector0 |
|                  | 8038h~803Fh          | 10070h~1007Eh        | 16B                |          | DataS 1_1 | DataRAM Spare page1/sector1 |
|                  | 8040h~8047h          | 10080h~1008Eh        | 16B                |          | DataS 1_2 | DataRAM Spare page1/sector2 |
|                  | 8048h~804Fh          | 10090h~1009Eh        | 16B                |          | DataS 1_3 | DataRAM Spare page1/sector3 |
| 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      | Registers | Registers                   |

NOTE 1) Data output is unknown while host reads a register bit of reserved area

## 6.2 ADDRESS MAP For OneNAND NAND Array (word order)

| Block   | Block Address | Page and Sector Address | Size  | Block   | Block Address | Page and Sector Address | Size  |
|---------|---------------|-------------------------|-------|---------|---------------|-------------------------|-------|
| Block0  | 0000h         | 0000h~00FFh             | 128KB | Block32 | 0020h         | 0000h~00FFh             | 128KB |
| Block1  | 0001h         | 0000h~00FFh             | 128KB | Block33 | 0021h         | 0000h~00FFh             | 128KB |
| Block2  | 0002h         | 0000h~00FFh             | 128KB | Block34 | 0022h         | 0000h~00FFh             | 128KB |
| Block3  | 0003h         | 0000h~00FFh             | 128KB | Block35 | 0023h         | 0000h~00FFh             | 128KB |
| Block4  | 0004h         | 0000h~00FFh             | 128KB | Block36 | 0024h         | 0000h~00FFh             | 128KB |
| Block5  | 0005h         | 0000h~00FFh             | 128KB | Block37 | 0025h         | 0000h~00FFh             | 128KB |
| Block6  | 0006h         | 0000h~00FFh             | 128KB | Block38 | 0026h         | 0000h~00FFh             | 128KB |
| Block7  | 0007h         | 0000h~00FFh             | 128KB | Block39 | 0027h         | 0000h~00FFh             | 128KB |
| Block8  | 0008h         | 0000h~00FFh             | 128KB | Block40 | 0028h         | 0000h~00FFh             | 128KB |
| Block9  | 0009h         | 0000h~00FFh             | 128KB | Block41 | 0029h         | 0000h~00FFh             | 128KB |
| Block10 | 000Ah         | 0000h~00FFh             | 128KB | Block42 | 002Ah         | 0000h~00FFh             | 128KB |
| Block11 | 000Bh         | 0000h~00FFh             | 128KB | Block43 | 002Bh         | 0000h~00FFh             | 128KB |
| Block12 | 000Ch         | 0000h~00FFh             | 128KB | Block44 | 002Ch         | 0000h~00FFh             | 128KB |
| Block13 | 000Dh         | 0000h~00FFh             | 128KB | Block45 | 002Dh         | 0000h~00FFh             | 128KB |
| Block14 | 000Eh         | 0000h~00FFh             | 128KB | Block46 | 002Eh         | 0000h~00FFh             | 128KB |
| Block15 | 000Fh         | 0000h~00FFh             | 128KB | Block47 | 002Fh         | 0000h~00FFh             | 128KB |
| Block16 | 0010h         | 0000h~00FFh             | 128KB | Block48 | 0030h         | 0000h~00FFh             | 128KB |
| Block17 | 0011h         | 0000h~00FFh             | 128KB | Block49 | 0031h         | 0000h~00FFh             | 128KB |
| Block18 | 0012h         | 0000h~00FFh             | 128KB | Block50 | 0032h         | 0000h~00FFh             | 128KB |
| Block19 | 0013h         | 0000h~00FFh             | 128KB | Block51 | 0033h         | 0000h~00FFh             | 128KB |
| Block20 | 0014h         | 0000h~00FFh             | 128KB | Block52 | 0034h         | 0000h~00FFh             | 128KB |
| Block21 | 0015h         | 0000h~00FFh             | 128KB | Block53 | 0035h         | 0000h~00FFh             | 128KB |
| Block22 | 0016h         | 0000h~00FFh             | 128KB | Block54 | 0036h         | 0000h~00FFh             | 128KB |
| Block23 | 0017h         | 0000h~00FFh             | 128KB | Block55 | 0037h         | 0000h~00FFh             | 128KB |
| Block24 | 0018h         | 0000h~00FFh             | 128KB | Block56 | 0038h         | 0000h~00FFh             | 128KB |
| Block25 | 0019h         | 0000h~00FFh             | 128KB | Block57 | 0039h         | 0000h~00FFh             | 128KB |
| Block26 | 001Ah         | 0000h~00FFh             | 128KB | Block58 | 003Ah         | 0000h~00FFh             | 128KB |
| Block27 | 001Bh         | 0000h~00FFh             | 128KB | Block59 | 003Bh         | 0000h~00FFh             | 128KB |
| Block28 | 001Ch         | 0000h~00FFh             | 128KB | Block60 | 003Ch         | 0000h~00FFh             | 128KB |
| Block29 | 001Dh         | 0000h~00FFh             | 128KB | Block61 | 003Dh         | 0000h~00FFh             | 128KB |
| Block30 | 001Eh         | 0000h~00FFh             | 128KB | Block62 | 003Eh         | 0000h~00FFh             | 128KB |
| Block31 | 001Fh         | 0000h~00FFh             | 128KB | Block63 | 003Fh         | 0000h~00FFh             | 128KB |

| Block   | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|---------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block64 | 0040h         | 0000h~00FFh             | 128KB | Block96  | 0060h         | 0000h~00FFh             | 128KB |
| Block65 | 0041h         | 0000h~00FFh             | 128KB | Block97  | 0061h         | 0000h~00FFh             | 128KB |
| Block66 | 0042h         | 0000h~00FFh             | 128KB | Block98  | 0062h         | 0000h~00FFh             | 128KB |
| Block67 | 0043h         | 0000h~00FFh             | 128KB | Block99  | 0063h         | 0000h~00FFh             | 128KB |
| Block68 | 0044h         | 0000h~00FFh             | 128KB | Block100 | 0064h         | 0000h~00FFh             | 128KB |
| Block69 | 0045h         | 0000h~00FFh             | 128KB | Block101 | 0065h         | 0000h~00FFh             | 128KB |
| Block70 | 0046h         | 0000h~00FFh             | 128KB | Block102 | 0066h         | 0000h~00FFh             | 128KB |
| Block71 | 0047h         | 0000h~00FFh             | 128KB | Block103 | 0067h         | 0000h~00FFh             | 128KB |
| Block72 | 0048h         | 0000h~00FFh             | 128KB | Block104 | 0068h         | 0000h~00FFh             | 128KB |
| Block73 | 0049h         | 0000h~00FFh             | 128KB | Block105 | 0069h         | 0000h~00FFh             | 128KB |
| Block74 | 004Ah         | 0000h~00FFh             | 128KB | Block106 | 006Ah         | 0000h~00FFh             | 128KB |
| Block75 | 004Bh         | 0000h~00FFh             | 128KB | Block107 | 006Bh         | 0000h~00FFh             | 128KB |
| Block76 | 004Ch         | 0000h~00FFh             | 128KB | Block108 | 006Ch         | 0000h~00FFh             | 128KB |
| Block77 | 004Dh         | 0000h~00FFh             | 128KB | Block109 | 006Dh         | 0000h~00FFh             | 128KB |
| Block78 | 004Eh         | 0000h~00FFh             | 128KB | Block110 | 006Eh         | 0000h~00FFh             | 128KB |
| Block79 | 004Fh         | 0000h~00FFh             | 128KB | Block111 | 006Fh         | 0000h~00FFh             | 128KB |
| Block80 | 0050h         | 0000h~00FFh             | 128KB | Block112 | 0070h         | 0000h~00FFh             | 128KB |
| Block81 | 0051h         | 0000h~00FFh             | 128KB | Block113 | 0071h         | 0000h~00FFh             | 128KB |
| Block82 | 0052h         | 0000h~00FFh             | 128KB | Block114 | 0072h         | 0000h~00FFh             | 128KB |
| Block83 | 0053h         | 0000h~00FFh             | 128KB | Block115 | 0073h         | 0000h~00FFh             | 128KB |
| Block84 | 0054h         | 0000h~00FFh             | 128KB | Block116 | 0074h         | 0000h~00FFh             | 128KB |
| Block85 | 0055h         | 0000h~00FFh             | 128KB | Block117 | 0075h         | 0000h~00FFh             | 128KB |
| Block86 | 0056h         | 0000h~00FFh             | 128KB | Block118 | 0076h         | 0000h~00FFh             | 128KB |
| Block87 | 0057h         | 0000h~00FFh             | 128KB | Block119 | 0077h         | 0000h~00FFh             | 128KB |
| Block88 | 0058h         | 0000h~00FFh             | 128KB | Block120 | 0078h         | 0000h~00FFh             | 128KB |
| Block89 | 0059h         | 0000h~00FFh             | 128KB | Block121 | 0079h         | 0000h~00FFh             | 128KB |
| Block90 | 005Ah         | 0000h~00FFh             | 128KB | Block122 | 007Ah         | 0000h~00FFh             | 128KB |
| Block91 | 005Bh         | 0000h~00FFh             | 128KB | Block123 | 007Bh         | 0000h~00FFh             | 128KB |
| Block92 | 005Ch         | 0000h~00FFh             | 128KB | Block124 | 007Ch         | 0000h~00FFh             | 128KB |
| Block93 | 005Dh         | 0000h~00FFh             | 128KB | Block125 | 007Dh         | 0000h~00FFh             | 128KB |
| Block94 | 005Eh         | 0000h~00FFh             | 128KB | Block126 | 007Eh         | 0000h~00FFh             | 128KB |
| Block95 | 005Fh         | 0000h~00FFh             | 128KB | Block127 | 007Fh         | 0000h~00FFh             | 128KB |

| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block128 | 0080h         | 0000h~00FFh             | 128KB | Block160 | 00A0h         | 0000h~00FFh             | 128KB |
| Block129 | 0081h         | 0000h~00FFh             | 128KB | Block161 | 00A1h         | 0000h~00FFh             | 128KB |
| Block130 | 0082h         | 0000h~00FFh             | 128KB | Block162 | 00A2h         | 0000h~00FFh             | 128KB |
| Block131 | 0083h         | 0000h~00FFh             | 128KB | Block163 | 00A3h         | 0000h~00FFh             | 128KB |
| Block132 | 0084h         | 0000h~00FFh             | 128KB | Block164 | 00A4h         | 0000h~00FFh             | 128KB |
| Block133 | 0085h         | 0000h~00FFh             | 128KB | Block165 | 00A5h         | 0000h~00FFh             | 128KB |
| Block134 | 0086h         | 0000h~00FFh             | 128KB | Block166 | 00A6h         | 0000h~00FFh             | 128KB |
| Block135 | 0087h         | 0000h~00FFh             | 128KB | Block167 | 00A7h         | 0000h~00FFh             | 128KB |
| Block136 | 0088h         | 0000h~00FFh             | 128KB | Block168 | 00A8h         | 0000h~00FFh             | 128KB |
| Block137 | 0089h         | 0000h~00FFh             | 128KB | Block169 | 00A9h         | 0000h~00FFh             | 128KB |
| Block138 | 008Ah         | 0000h~00FFh             | 128KB | Block170 | 00AAh         | 0000h~00FFh             | 128KB |
| Block139 | 008Bh         | 0000h~00FFh             | 128KB | Block171 | 00ABh         | 0000h~00FFh             | 128KB |
| Block140 | 008Ch         | 0000h~00FFh             | 128KB | Block172 | 00ACh         | 0000h~00FFh             | 128KB |
| Block141 | 008Dh         | 0000h~00FFh             | 128KB | Block173 | 00ADh         | 0000h~00FFh             | 128KB |
| Block142 | 008Eh         | 0000h~00FFh             | 128KB | Block174 | 00AEh         | 0000h~00FFh             | 128KB |
| Block143 | 008Fh         | 0000h~00FFh             | 128KB | Block175 | 00AFh         | 0000h~00FFh             | 128KB |
| Block144 | 0090h         | 0000h~00FFh             | 128KB | Block176 | 00B0h         | 0000h~00FFh             | 128KB |
| Block145 | 0091h         | 0000h~00FFh             | 128KB | Block177 | 00B1h         | 0000h~00FFh             | 128KB |
| Block146 | 0092h         | 0000h~00FFh             | 128KB | Block178 | 00B2h         | 0000h~00FFh             | 128KB |
| Block147 | 0093h         | 0000h~00FFh             | 128KB | Block179 | 00B3h         | 0000h~00FFh             | 128KB |
| Block148 | 0094h         | 0000h~00FFh             | 128KB | Block180 | 00B4h         | 0000h~00FFh             | 128KB |
| Block149 | 0095h         | 0000h~00FFh             | 128KB | Block181 | 00B5h         | 0000h~00FFh             | 128KB |
| Block150 | 0096h         | 0000h~00FFh             | 128KB | Block182 | 00B6h         | 0000h~00FFh             | 128KB |
| Block151 | 0097h         | 0000h~00FFh             | 128KB | Block183 | 00B7h         | 0000h~00FFh             | 128KB |
| Block152 | 0098h         | 0000h~00FFh             | 128KB | Block184 | 00B8h         | 0000h~00FFh             | 128KB |
| Block153 | 0099h         | 0000h~00FFh             | 128KB | Block185 | 00B9h         | 0000h~00FFh             | 128KB |
| Block154 | 009Ah         | 0000h~00FFh             | 128KB | Block186 | 00BAh         | 0000h~00FFh             | 128KB |
| Block155 | 009Bh         | 0000h~00FFh             | 128KB | Block187 | 00BBh         | 0000h~00FFh             | 128KB |
| Block156 | 009Ch         | 0000h~00FFh             | 128KB | Block188 | 00BCh         | 0000h~00FFh             | 128KB |
| Block157 | 009Dh         | 0000h~00FFh             | 128KB | Block189 | 00BDh         | 0000h~00FFh             | 128KB |
| Block158 | 009Eh         | 0000h~00FFh             | 128KB | Block190 | 00BEh         | 0000h~00FFh             | 128KB |
| Block159 | 009Fh         | 0000h~00FFh             | 128KB | Block191 | 00BFh         | 0000h~00FFh             | 128KB |



| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block192 | 00C0h         | 0000h~00FFh             | 128KB | Block224 | 00E0h         | 0000h~00FFh             | 128KB |
| Block193 | 00C1h         | 0000h~00FFh             | 128KB | Block225 | 00E1h         | 0000h~00FFh             | 128KB |
| Block194 | 00C2h         | 0000h~00FFh             | 128KB | Block226 | 00E2h         | 0000h~00FFh             | 128KB |
| Block195 | 00C3h         | 0000h~00FFh             | 128KB | Block227 | 00E3h         | 0000h~00FFh             | 128KB |
| Block196 | 00C4h         | 0000h~00FFh             | 128KB | Block228 | 00E4h         | 0000h~00FFh             | 128KB |
| Block197 | 00C5h         | 0000h~00FFh             | 128KB | Block229 | 00E5h         | 0000h~00FFh             | 128KB |
| Block198 | 00C6h         | 0000h~00FFh             | 128KB | Block230 | 00E6h         | 0000h~00FFh             | 128KB |
| Block199 | 00C7h         | 0000h~00FFh             | 128KB | Block231 | 00E7h         | 0000h~00FFh             | 128KB |
| Block200 | 00C8h         | 0000h~00FFh             | 128KB | Block232 | 00E8h         | 0000h~00FFh             | 128KB |
| Block201 | 00C9h         | 0000h~00FFh             | 128KB | Block233 | 00E9h         | 0000h~00FFh             | 128KB |
| Block202 | 00CAh         | 0000h~00FFh             | 128KB | Block234 | 00EAh         | 0000h~00FFh             | 128KB |
| Block203 | 00CBh         | 0000h~00FFh             | 128KB | Block235 | 00EBh         | 0000h~00FFh             | 128KB |
| Block204 | 00CCh         | 0000h~00FFh             | 128KB | Block236 | 00ECh         | 0000h~00FFh             | 128KB |
| Block205 | 00CDh         | 0000h~00FFh             | 128KB | Block237 | 00EDh         | 0000h~00FFh             | 128KB |
| Block206 | 00CEh         | 0000h~00FFh             | 128KB | Block238 | 00EEh         | 0000h~00FFh             | 128KB |
| Block207 | 00CFh         | 0000h~00FFh             | 128KB | Block239 | 00EFh         | 0000h~00FFh             | 128KB |
| Block208 | 00D0h         | 0000h~00FFh             | 128KB | Block240 | 00F0h         | 0000h~00FFh             | 128KB |
| Block209 | 00D1h         | 0000h~00FFh             | 128KB | Block241 | 00F1h         | 0000h~00FFh             | 128KB |
| Block210 | 00D2h         | 0000h~00FFh             | 128KB | Block242 | 00F2h         | 0000h~00FFh             | 128KB |
| Block211 | 00D3h         | 0000h~00FFh             | 128KB | Block243 | 00F3h         | 0000h~00FFh             | 128KB |
| Block212 | 00D4h         | 0000h~00FFh             | 128KB | Block244 | 00F4h         | 0000h~00FFh             | 128KB |
| Block213 | 00D5h         | 0000h~00FFh             | 128KB | Block245 | 00F5h         | 0000h~00FFh             | 128KB |
| Block214 | 00D6h         | 0000h~00FFh             | 128KB | Block246 | 00F6h         | 0000h~00FFh             | 128KB |
| Block215 | 00D7h         | 0000h~00FFh             | 128KB | Block247 | 00F7h         | 0000h~00FFh             | 128KB |
| Block216 | 00D8h         | 0000h~00FFh             | 128KB | Block248 | 00F8h         | 0000h~00FFh             | 128KB |
| Block217 | 00D9h         | 0000h~00FFh             | 128KB | Block249 | 00F9h         | 0000h~00FFh             | 128KB |
| Block218 | 00DAh         | 0000h~00FFh             | 128KB | Block250 | 00FAh         | 0000h~00FFh             | 128KB |
| Block219 | 00DBh         | 0000h~00FFh             | 128KB | Block251 | 00FBh         | 0000h~00FFh             | 128KB |
| Block220 | 00DCh         | 0000h~00FFh             | 128KB | Block252 | 00FCh         | 0000h~00FFh             | 128KB |
| Block221 | 00DDh         | 0000h~00FFh             | 128KB | Block253 | 00FDh         | 0000h~00FFh             | 128KB |
| Block222 | 00DEh         | 0000h~00FFh             | 128KB | Block254 | 00FEh         | 0000h~00FFh             | 128KB |
| Block223 | 00DFh         | 0000h~00FFh             | 128KB | Block255 | 00FFh         | 0000h~00FFh             | 128KB |

| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block256 | 0100h         | 0000h~00FFh             | 128KB | Block288 | 0120h         | 0000h~00FFh             | 128KB |
| Block257 | 0101h         | 0000h~00FFh             | 128KB | Block289 | 0121h         | 0000h~00FFh             | 128KB |
| Block258 | 0102h         | 0000h~00FFh             | 128KB | Block290 | 0122h         | 0000h~00FFh             | 128KB |
| Block259 | 0103h         | 0000h~00FFh             | 128KB | Block291 | 0123h         | 0000h~00FFh             | 128KB |
| Block260 | 0104h         | 0000h~00FFh             | 128KB | Block292 | 0124h         | 0000h~00FFh             | 128KB |
| Block261 | 0105h         | 0000h~00FFh             | 128KB | Block293 | 0125h         | 0000h~00FFh             | 128KB |
| Block262 | 0106h         | 0000h~00FFh             | 128KB | Block294 | 0126h         | 0000h~00FFh             | 128KB |
| Block263 | 0107h         | 0000h~00FFh             | 128KB | Block295 | 0127h         | 0000h~00FFh             | 128KB |
| Block264 | 0108h         | 0000h~00FFh             | 128KB | Block296 | 0128h         | 0000h~00FFh             | 128KB |
| Block265 | 0109h         | 0000h~00FFh             | 128KB | Block297 | 0129h         | 0000h~00FFh             | 128KB |
| Block266 | 010Ah         | 0000h~00FFh             | 128KB | Block298 | 012Ah         | 0000h~00FFh             | 128KB |
| Block267 | 010Bh         | 0000h~00FFh             | 128KB | Block299 | 012Bh         | 0000h~00FFh             | 128KB |
| Block268 | 010Ch         | 0000h~00FFh             | 128KB | Block300 | 012Ch         | 0000h~00FFh             | 128KB |
| Block269 | 010Dh         | 0000h~00FFh             | 128KB | Block301 | 012Dh         | 0000h~00FFh             | 128KB |
| Block270 | 010Eh         | 0000h~00FFh             | 128KB | Block302 | 012Eh         | 0000h~00FFh             | 128KB |
| Block271 | 010Fh         | 0000h~00FFh             | 128KB | Block303 | 012Fh         | 0000h~00FFh             | 128KB |
| Block272 | 0110h         | 0000h~00FFh             | 128KB | Block304 | 0130h         | 0000h~00FFh             | 128KB |
| Block273 | 0111h         | 0000h~00FFh             | 128KB | Block305 | 0131h         | 0000h~00FFh             | 128KB |
| Block274 | 0112h         | 0000h~00FFh             | 128KB | Block306 | 0132h         | 0000h~00FFh             | 128KB |
| Block275 | 0113h         | 0000h~00FFh             | 128KB | Block307 | 0133h         | 0000h~00FFh             | 128KB |
| Block276 | 0114h         | 0000h~00FFh             | 128KB | Block308 | 0134h         | 0000h~00FFh             | 128KB |
| Block277 | 0115h         | 0000h~00FFh             | 128KB | Block309 | 0135h         | 0000h~00FFh             | 128KB |
| Block278 | 0116h         | 0000h~00FFh             | 128KB | Block310 | 0136h         | 0000h~00FFh             | 128KB |
| Block279 | 0117h         | 0000h~00FFh             | 128KB | Block311 | 0137h         | 0000h~00FFh             | 128KB |
| Block280 | 0118h         | 0000h~00FFh             | 128KB | Block312 | 0138h         | 0000h~00FFh             | 128KB |
| Block281 | 0119h         | 0000h~00FFh             | 128KB | Block313 | 0139h         | 0000h~00FFh             | 128KB |
| Block282 | 011Ah         | 0000h~00FFh             | 128KB | Block314 | 013Ah         | 0000h~00FFh             | 128KB |
| Block283 | 011Bh         | 0000h~00FFh             | 128KB | Block315 | 013Bh         | 0000h~00FFh             | 128KB |
| Block284 | 011Ch         | 0000h~00FFh             | 128KB | Block316 | 013Ch         | 0000h~00FFh             | 128KB |
| Block285 | 011Dh         | 0000h~00FFh             | 128KB | Block317 | 013Dh         | 0000h~00FFh             | 128KB |
| Block286 | 011Eh         | 0000h~00FFh             | 128KB | Block318 | 013Eh         | 0000h~00FFh             | 128KB |
| Block287 | 011Fh         | 0000h~00FFh             | 128KB | Block319 | 013Fh         | 0000h~00FFh             | 128KB |

| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block320 | 0140h         | 0000h~00FFh             | 128KB | Block352 | 0160h         | 0000h~00FFh             | 128KB |
| Block321 | 0141h         | 0000h~00FFh             | 128KB | Block353 | 0161h         | 0000h~00FFh             | 128KB |
| Block322 | 0142h         | 0000h~00FFh             | 128KB | Block354 | 0162h         | 0000h~00FFh             | 128KB |
| Block323 | 0143h         | 0000h~00FFh             | 128KB | Block355 | 0163h         | 0000h~00FFh             | 128KB |
| Block324 | 0144h         | 0000h~00FFh             | 128KB | Block356 | 0164h         | 0000h~00FFh             | 128KB |
| Block325 | 0145h         | 0000h~00FFh             | 128KB | Block357 | 0165h         | 0000h~00FFh             | 128KB |
| Block326 | 0146h         | 0000h~00FFh             | 128KB | Block358 | 0166h         | 0000h~00FFh             | 128KB |
| Block327 | 0147h         | 0000h~00FFh             | 128KB | Block359 | 0167h         | 0000h~00FFh             | 128KB |
| Block328 | 0148h         | 0000h~00FFh             | 128KB | Block360 | 0168h         | 0000h~00FFh             | 128KB |
| Block329 | 0149h         | 0000h~00FFh             | 128KB | Block361 | 0169h         | 0000h~00FFh             | 128KB |
| Block330 | 014Ah         | 0000h~00FFh             | 128KB | Block362 | 016Ah         | 0000h~00FFh             | 128KB |
| Block331 | 014Bh         | 0000h~00FFh             | 128KB | Block363 | 016Bh         | 0000h~00FFh             | 128KB |
| Block332 | 014Ch         | 0000h~00FFh             | 128KB | Block364 | 016Ch         | 0000h~00FFh             | 128KB |
| Block333 | 014Dh         | 0000h~00FFh             | 128KB | Block365 | 016Dh         | 0000h~00FFh             | 128KB |
| Block334 | 014Eh         | 0000h~00FFh             | 128KB | Block366 | 016Eh         | 0000h~00FFh             | 128KB |
| Block335 | 014Fh         | 0000h~00FFh             | 128KB | Block367 | 016Fh         | 0000h~00FFh             | 128KB |
| Block336 | 0150h         | 0000h~00FFh             | 128KB | Block368 | 0170h         | 0000h~00FFh             | 128KB |
| Block337 | 0151h         | 0000h~00FFh             | 128KB | Block369 | 0171h         | 0000h~00FFh             | 128KB |
| Block338 | 0152h         | 0000h~00FFh             | 128KB | Block370 | 0172h         | 0000h~00FFh             | 128KB |
| Block339 | 0153h         | 0000h~00FFh             | 128KB | Block371 | 0173h         | 0000h~00FFh             | 128KB |
| Block340 | 0154h         | 0000h~00FFh             | 128KB | Block372 | 0174h         | 0000h~00FFh             | 128KB |
| Block341 | 0155h         | 0000h~00FFh             | 128KB | Block373 | 0175h         | 0000h~00FFh             | 128KB |
| Block342 | 0156h         | 0000h~00FFh             | 128KB | Block374 | 0176h         | 0000h~00FFh             | 128KB |
| Block343 | 0157h         | 0000h~00FFh             | 128KB | Block375 | 0177h         | 0000h~00FFh             | 128KB |
| Block344 | 0158h         | 0000h~00FFh             | 128KB | Block376 | 0178h         | 0000h~00FFh             | 128KB |
| Block345 | 0159h         | 0000h~00FFh             | 128KB | Block377 | 0179h         | 0000h~00FFh             | 128KB |
| Block346 | 015Ah         | 0000h~00FFh             | 128KB | Block378 | 017Ah         | 0000h~00FFh             | 128KB |
| Block347 | 015Bh         | 0000h~00FFh             | 128KB | Block379 | 017Bh         | 0000h~00FFh             | 128KB |
| Block348 | 015Ch         | 0000h~00FFh             | 128KB | Block380 | 017Ch         | 0000h~00FFh             | 128KB |
| Block349 | 015Dh         | 0000h~00FFh             | 128KB | Block381 | 017Dh         | 0000h~00FFh             | 128KB |
| Block350 | 015Eh         | 0000h~00FFh             | 128KB | Block382 | 017Eh         | 0000h~00FFh             | 128KB |
| Block351 | 015Fh         | 0000h~00FFh             | 128KB | Block383 | 017Fh         | 0000h~00FFh             | 128KB |

| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block384 | 0180h         | 0000h~00FFh             | 128KB | Block416 | 01A0h         | 0000h~00FFh             | 128KB |
| Block385 | 0181h         | 0000h~00FFh             | 128KB | Block417 | 01A1h         | 0000h~00FFh             | 128KB |
| Block386 | 0182h         | 0000h~00FFh             | 128KB | Block418 | 01A2h         | 0000h~00FFh             | 128KB |
| Block387 | 0183h         | 0000h~00FFh             | 128KB | Block419 | 01A3h         | 0000h~00FFh             | 128KB |
| Block388 | 0184h         | 0000h~00FFh             | 128KB | Block420 | 01A4h         | 0000h~00FFh             | 128KB |
| Block389 | 0185h         | 0000h~00FFh             | 128KB | Block421 | 01A5h         | 0000h~00FFh             | 128KB |
| Block390 | 0186h         | 0000h~00FFh             | 128KB | Block422 | 01A6h         | 0000h~00FFh             | 128KB |
| Block391 | 0187h         | 0000h~00FFh             | 128KB | Block423 | 01A7h         | 0000h~00FFh             | 128KB |
| Block392 | 0188h         | 0000h~00FFh             | 128KB | Block424 | 01A8h         | 0000h~00FFh             | 128KB |
| Block393 | 0189h         | 0000h~00FFh             | 128KB | Block425 | 01A9h         | 0000h~00FFh             | 128KB |
| Block394 | 018Ah         | 0000h~00FFh             | 128KB | Block426 | 01AAh         | 0000h~00FFh             | 128KB |
| Block395 | 018Bh         | 0000h~00FFh             | 128KB | Block427 | 01ABh         | 0000h~00FFh             | 128KB |
| Block396 | 018Ch         | 0000h~00FFh             | 128KB | Block428 | 01ACh         | 0000h~00FFh             | 128KB |
| Block397 | 018Dh         | 0000h~00FFh             | 128KB | Block429 | 01ADh         | 0000h~00FFh             | 128KB |
| Block398 | 018Eh         | 0000h~00FFh             | 128KB | Block430 | 01AEh         | 0000h~00FFh             | 128KB |
| Block399 | 018Fh         | 0000h~00FFh             | 128KB | Block431 | 01AFh         | 0000h~00FFh             | 128KB |
| Block400 | 0190h         | 0000h~00FFh             | 128KB | Block432 | 01B0h         | 0000h~00FFh             | 128KB |
| Block401 | 0191h         | 0000h~00FFh             | 128KB | Block433 | 01B1h         | 0000h~00FFh             | 128KB |
| Block402 | 0192h         | 0000h~00FFh             | 128KB | Block434 | 01B2h         | 0000h~00FFh             | 128KB |
| Block403 | 0193h         | 0000h~00FFh             | 128KB | Block435 | 01B3h         | 0000h~00FFh             | 128KB |
| Block404 | 0194h         | 0000h~00FFh             | 128KB | Block436 | 01B4h         | 0000h~00FFh             | 128KB |
| Block405 | 0195h         | 0000h~00FFh             | 128KB | Block437 | 01B5h         | 0000h~00FFh             | 128KB |
| Block406 | 0196h         | 0000h~00FFh             | 128KB | Block438 | 01B6h         | 0000h~00FFh             | 128KB |
| Block407 | 0197h         | 0000h~00FFh             | 128KB | Block439 | 01B7h         | 0000h~00FFh             | 128KB |
| Block408 | 0198h         | 0000h~00FFh             | 128KB | Block440 | 01B8h         | 0000h~00FFh             | 128KB |
| Block409 | 0199h         | 0000h~00FFh             | 128KB | Block441 | 01B9h         | 0000h~00FFh             | 128KB |
| Block410 | 019Ah         | 0000h~00FFh             | 128KB | Block442 | 01BAh         | 0000h~00FFh             | 128KB |
| Block411 | 019Bh         | 0000h~00FFh             | 128KB | Block443 | 01BBh         | 0000h~00FFh             | 128KB |
| Block412 | 019Ch         | 0000h~00FFh             | 128KB | Block444 | 01BCh         | 0000h~00FFh             | 128KB |
| Block413 | 019Dh         | 0000h~00FFh             | 128KB | Block445 | 01BDh         | 0000h~00FFh             | 128KB |
| Block414 | 019Eh         | 0000h~00FFh             | 128KB | Block446 | 01BEh         | 0000h~00FFh             | 128KB |
| Block415 | 019Fh         | 0000h~00FFh             | 128KB | Block447 | 01BFh         | 0000h~00FFh             | 128KB |

| Block    | Block Address | Page and Sector Address | Size  | Block    | Block Address | Page and Sector Address | Size  |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block448 | 01C0h         | 0000h~00FFh             | 128KB | Block480 | 01E0h         | 0000h~00FFh             | 128KB |
| Block449 | 01C1h         | 0000h~00FFh             | 128KB | Block481 | 01E1h         | 0000h~00FFh             | 128KB |
| Block450 | 01C2h         | 0000h~00FFh             | 128KB | Block482 | 01E2h         | 0000h~00FFh             | 128KB |
| Block451 | 01C3h         | 0000h~00FFh             | 128KB | Block483 | 01E3h         | 0000h~00FFh             | 128KB |
| Block452 | 01C4h         | 0000h~00FFh             | 128KB | Block484 | 01E4h         | 0000h~00FFh             | 128KB |
| Block453 | 01C5h         | 0000h~00FFh             | 128KB | Block485 | 01E5h         | 0000h~00FFh             | 128KB |
| Block454 | 01C6h         | 0000h~00FFh             | 128KB | Block486 | 01E6h         | 0000h~00FFh             | 128KB |
| Block455 | 01C7h         | 0000h~00FFh             | 128KB | Block487 | 01E7h         | 0000h~00FFh             | 128KB |
| Block456 | 01C8h         | 0000h~00FFh             | 128KB | Block488 | 01E8h         | 0000h~00FFh             | 128KB |
| Block457 | 01C9h         | 0000h~00FFh             | 128KB | Block489 | 01E9h         | 0000h~00FFh             | 128KB |
| Block458 | 01CAh         | 0000h~00FFh             | 128KB | Block490 | 01EAh         | 0000h~00FFh             | 128KB |
| Block459 | 01CBh         | 0000h~00FFh             | 128KB | Block491 | 01EBh         | 0000h~00FFh             | 128KB |
| Block460 | 01CCh         | 0000h~00FFh             | 128KB | Block492 | 01ECh         | 0000h~00FFh             | 128KB |
| Block461 | 01CDh         | 0000h~00FFh             | 128KB | Block493 | 01EDh         | 0000h~00FFh             | 128KB |
| Block462 | 01CEh         | 0000h~00FFh             | 128KB | Block494 | 01EEh         | 0000h~00FFh             | 128KB |
| Block463 | 01CFh         | 0000h~00FFh             | 128KB | Block495 | 01EFh         | 0000h~00FFh             | 128KB |
| Block464 | 01D0h         | 0000h~00FFh             | 128KB | Block496 | 01F0h         | 0000h~00FFh             | 128KB |
| Block465 | 01D1h         | 0000h~00FFh             | 128KB | Block497 | 01F1h         | 0000h~00FFh             | 128KB |
| Block466 | 01D2h         | 0000h~00FFh             | 128KB | Block498 | 01F2h         | 0000h~00FFh             | 128KB |
| Block467 | 01D3h         | 0000h~00FFh             | 128KB | Block499 | 01F3h         | 0000h~00FFh             | 128KB |
| Block468 | 01D4h         | 0000h~00FFh             | 128KB | Block500 | 01F4h         | 0000h~00FFh             | 128KB |
| Block469 | 01D5h         | 0000h~00FFh             | 128KB | Block501 | 01F5h         | 0000h~00FFh             | 128KB |
| Block470 | 01D6h         | 0000h~00FFh             | 128KB | Block502 | 01F6h         | 0000h~00FFh             | 128KB |
| Block471 | 01D7h         | 0000h~00FFh             | 128KB | Block503 | 01F7h         | 0000h~00FFh             | 128KB |
| Block472 | 01D8h         | 0000h~00FFh             | 128KB | Block504 | 01F8h         | 0000h~00FFh             | 128KB |
| Block473 | 01D9h         | 0000h~00FFh             | 128KB | Block505 | 01F9h         | 0000h~00FFh             | 128KB |
| Block474 | 01DAh         | 0000h~00FFh             | 128KB | Block506 | 01FAh         | 0000h~00FFh             | 128KB |
| Block475 | 01DBh         | 0000h~00FFh             | 128KB | Block507 | 01FBh         | 0000h~00FFh             | 128KB |
| Block476 | 01DCh         | 0000h~00FFh             | 128KB | Block508 | 01FCh         | 0000h~00FFh             | 128KB |
| Block477 | 01DDh         | 0000h~00FFh             | 128KB | Block509 | 01FDh         | 0000h~00FFh             | 128KB |
| Block478 | 01DEh         | 0000h~00FFh             | 128KB | Block510 | 01FEh         | 0000h~00FFh             | 128KB |
| Block479 | 01DFh         | 0000h~00FFh             | 128KB | Block511 | 01FFh         | 0000h~00FFh             | 128KB |

## Detailed information of Address Map (word order)

### • 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) | 0100h~01FFh(512B)<br>BootM 1<br>(sector 1 of page 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 page 0) | 0300h~03FFh(512B)<br>DataM 0_1<br>(sector 1 of page 0) | 0400h~04FFh(512B)<br>DataM 0_2<br>(sector 2 of page 0) | 0500h~05FFh(512B)<br>DataM 0_3<br>(sector 3 of page 0) |
| 0600h~06FFh(512B)<br>DataM 1_0<br>(sector 0 of page 1) | 0700h~07FFh(512B)<br>DataM 1_1<br>(sector 1 of page 1) | 0800h~08FFh(512B)<br>DataM 1_2<br>(sector 2 of page 1) | 0900h~09FFh(512B)<br>DataM 1_3<br>(sector 3 of page 1) |

### • 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) | 8008h~800Fh(16B)<br>BootS 1<br>(sector 1 of page 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 page 0) | 8018h~801Fh(16B)<br>DataS 0_1<br>(sector 1 of page 0) | 8020h~8027h(16B)<br>DataS 0_2<br>(sector 2 of page 0) | 8028h~802Fh(16B)<br>DataS 0_3<br>(sector 3 of page 0) |
| 8030h~8037h(16B)<br>DataS 1_0<br>(sector 0 of page 1) | 8038h~803Fh(16B)<br>DataS 1_1<br>(sector 1 of page 1) | 8040h~8047h(16B)<br>DataS 1_2<br>(sector 2 of page 1) | 8048h~804Fh(16B)<br>DataS 1_3<br>(sector 3 of page 1) |

\*NAND Flash array consists of 2KB page size and 128KB block size.

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  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8001h        | 10002h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8002h        | 10004h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 8003h        | 10006h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8004h        | 10008h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 8005h        | 1000Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 8006h        | 1000Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 8007h        | 1000Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| BootS 1   | 8008h        | 10010h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8009h        | 10012h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 800Ah        | 10014h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 800Bh        | 10016h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 800Ch        | 10018h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 800Dh        | 1001Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 800Eh        | 1001Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 800Fh        | 1001Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 0_0 | 8010h        | 10020h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8011h        | 10022h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8012h        | 10024h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 8013h        | 10026h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8014h        | 10028h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 8015h        | 1002Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 8016h        | 1002Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 8017h        | 1002Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 0_1 | 8018h        | 10030h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8019h        | 10032h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 801Ah        | 10034h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 801Bh        | 10036h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 801Ch        | 10038h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 801Dh        | 1003Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 801Eh        | 1003Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 801Fh        | 1003Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

|           |              |              | ← 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 0_2 | 8020h        | 10040h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8021h        | 10042h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8022h        | 10044h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 8023h        | 10046h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8024h        | 10048h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 8025h        | 1004Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 8026h        | 1004Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 8027h        | 1004Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 0_3 | 8028h        | 10050h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8029h        | 10052h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 802Ah        | 10054h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 802Bh        | 10056h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 802Ch        | 10058h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 802Dh        | 1005Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 802Eh        | 1005Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 802Fh        | 1005Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 1_0 | 8030h        | 10060h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8031h        | 10062h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8032h        | 10064h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 8033h        | 10066h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8034h        | 10068h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 8035h        | 1006Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 8036h        | 1006Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 8037h        | 1006Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 1_1 | 8038h        | 10070h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8039h        | 10072h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 803Ah        | 10074h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 803Bh        | 10076h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 803Ch        | 10078h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 803Dh        | 1007Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 803Eh        | 1007Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 803Fh        | 1007Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| DataS 1_2 | 8040h        | 10080h       | BI  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8041h        | 10082h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8042h        | 10084h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 8043h        | 10086h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8044h        | 10088h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 8045h        | 1008Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 8046h        | 1008Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 8047h        | 1008Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |



← 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  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 8049h        | 10092h       | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 804Ah        | 10094h       | Reserved for the future use                     |   |   |   |   |   |   |   | Managed by Internal ECC logic                   |   |   |   |   |   |   |   |
|           | 804Bh        | 10096h       | Reserved for the current and future use         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|           | 804Ch        | 10098h       | ECC Code for Main area data (2 <sup>nd</sup> )  |   |   |   |   |   |   |   | ECC Code for Main area data (1 <sup>st</sup> )  |   |   |   |   |   |   |   |
|           | 804Dh        | 1009Ah       | ECC Code for Spare area data (1 <sup>st</sup> ) |   |   |   |   |   |   |   | ECC Code for Main area data (3 <sup>rd</sup> )  |   |   |   |   |   |   |   |
|           | 804Eh        | 1009Ch       | FFh(Reserved for the future use)                |   |   |   |   |   |   |   | ECC Code for Spare area data (2 <sup>nd</sup> ) |   |   |   |   |   |   |   |
|           | 804Fh        | 1009Eh       | Free Usage                                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

**NOTE:**

- BI: Bad block Information

>Host can use complete spare area except BI and ECC code area. For example,

Host can write data to Spare area buffer except for the area controlled by ECC logic at program operation.

>In case of 'with ECC' mode, OneNAND 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.

>When loading/programming spare area, spare area BufferRAM address(BSA) and BufferRAM sector count(BSC) is chosen via Start buffer register as it is.

## 7. Detailed address map for registers

| Address<br>(word order) | Address<br>(byte order) | Name                       | Host<br>Access | Description  |
|-------------------------|-------------------------|----------------------------|----------------|--|
| F000h                   | 1E000h                  | Manufacturer ID            | R              | Manufacturer identification  |
| F001h                   | 1E002h                  | Device ID                  | R              | Device identification  |
| F002h                   | 1E004h                  | Version ID                 | R              | Version identification   |
| F003h                   | 1E006h                  | Data Buffer size           | R              | Data buffer size   |
| F004h                   | 1E008h                  | Boot Buffer size           | R              | Boot buffer size   |
| F005h                   | 1E00Ah                  | Amount of<br>buffers       | R              | Amount of data/boot buffers  |
| F006h                   | 1E00Ch                  | Technology                 | R              | Info about technology used for OneNAND   |
| F007h~F0FFh             | 1E00Eh~1E1FEh           | Reserved                   | -              | Reserved for user  |
| F100h                   | 1E200h                  | Start address 1            | R/W            | Chip address for selection of NAND<br>Core in DDP & Block address  |
| F101h                   | 1E202h                  | Start address 2            | R/W            | Chip address for selection of BufferRAM in DDP   |
| F102h                   | 1E204h                  | Start address 3            | R/W            | Destination Block address for Copy back program  |
| F103h                   | 1E206h                  | Start address 4            | R/W            | Destination Page & Sector address for Copy<br>back program   |
| F104h                   | 1E208h                  | Start address 5            | -              | N/A  |
| F105h                   | 1E20Ah                  | Start address 6            | -              | N/A  |
| F106h                   | 1E20Ch                  | Start address 7            | -              | N/A  |
| F107h                   | 1E20Eh                  | Start address 8            | R/W            | NAND Flash Page & Sector address   |
| F108h~F1FFh             | 1E210h~1E3FEh           | Reserved                   | -              | Reserved for user  |
| F200h                   | 1E400h                  | Start Buffer               | R/W            | Buffer Number for the page data transfer to/from the<br>OneNAND and the start Buffer Address<br>The meaning is with which buffer to start and how many<br>buffers to use for the data transfer |
| F201h~F207h             | 1E402h~1E40Eh           | Reserved                   | -              | Reserved for user  |
| F208h~F21Fh             | 1E410h~1E43Eh           | Reserved                   | -              | Reserved for vendor specific purposes  |
| F220h                   | 1E440h                  | Command                    | R/W            | Host control and OneNAND operation commands  |
| F221h                   | 1E442h                  | System<br>Configuration 1  | R, R/W         | OneNAND and Host Interface Configuration   |
| F222h                   | 1E444h                  | System<br>Configuration 2  | -              | N/A  |
| F223h~F22Fh             | 1E446h~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 OneNAND operation  |
| F241h                   | 1E482h                  | Interrupt                  | R/W            | OneNAND Command Completion Interrupt Status  |
| F242h~F24Bh             | 1E484h~1E496h           | Reserved                   | -              | Reserved for user  |
| F24Ch                   | 1E498h                  | Start<br>Block Address     | R/W            | Start OneNAND block address to unlock in Write<br>Protection mode  |
| F24Dh                   | 1E49Ah                  | End<br>Block Address       | R/W            | End OneNAND block address to unlock in Write<br>Protection mode  |
| F24Eh                   | 1E49Ch                  | Write Protection<br>Status | R              | Current OneNAND Write Protection status<br>(unlocked/locked/tight-locked)  |
| F24Fh~FEFFh             | 1E49Eh~1FDFEh           | Reserved                   | -              | Reserved for user  |

| Address<br>(word order) | Address<br>(byte order) | Name                          | Host<br>Access | Description  |
|-------------------------|-------------------------|-------------------------------|----------------|--|
| FF00h                   | 1FE00h                  | ECC Status Register           | R              | ECC status of sector   |
| FF01h                   | 1FE02h                  | ECC Result of main area data  | R              | ECC error position of Main area data error for first selected Sector   |
| FF02h                   | 1FE04h                  | ECC Result of spare area data | R              | ECC error position of Spare area data error for first selected Sector  |
| FF03h                   | 1FE06h                  | ECC Result of main area data  | R              | ECC error position of Main area data error for second selected Sector  |
| FF04h                   | 1FE08h                  | ECC Result of spare area data | R              | ECC error position of Spare area data error for second selected Sector |
| FF05h                   | 1FE0Ah                  | ECC Result of main area data  | R              | ECC error position of Main area data error for third selected Sector   |
| FF06h                   | 1FE0Ch                  | ECC Result of spare area data | R              | ECC error position of Spare area data error for third selected Sector  |
| FF07h                   | 1FE0Eh                  | ECC Result of main area data  | R              | ECC error position of Main area data error for fourth selected Sector  |
| FF08h                   | 1FE10h                  | ECC Result of spare area data | R              | ECC error position of Spare area data error for fourth selected Sector |
| FF09h~FFFFh             | 1FE12h~1FFFEh           | Reserved                      | -              | Reserved for vendor specific purposes                                  |

## 7.1 Manufacturer ID Register (R): F000h, default=00ECh

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

**ManufID** (Manufacturer ID): manufacturer identification, 00ECh for Samsung Electronics Corp.

## 7.2 Device ID Register (R): F001h, default=refer to Table1

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

**DeviceID** (Device ID): Device Identification,

**Table 1.**

| Device     | DeviceID[15:0] |
|------------|----------------|
| KFG1216Q2M | 0024h          |
| KFG1216D2M | 0025h          |
| KFG1216U2M | 0025h          |
| KFH1G16Q2M | 0024h          |
| KFH1G16D2M | 0025h          |
| KFH1G16U2M | 0025h          |

## 7.3 Version ID Register (R): F002h

: N/A

**7.4 Data Buffer size Register(R): F003h, default=0800h**

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

**DataBufSize:** total data buffer size in words in the memory interface  
 Equals two buffers of 1024 words each( $2 \times 1024 = 2^N$ , N=11)

**7.5 Boot Buffer size Register (R): F004h, default=0200h**

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

**BootBufSize:** total boot buffer size in words in the memory interface  
 (512 words= $2^9$ , N=9)

**7.6 Amount of Buffers Register (R): F005h, default=0201h**

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

**DataBufAmount:** the amount of data buffer= $2(2^N)$ , N=1  
**BootBufAmount:** the amount of boot buffer= $1(2^N)$ , N=0

**7.7 Technology Register (R): F006h, default=0000h**

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

**Tech:** technology information, what technology is used for the memory

| Tech        | Technology |
|-------------|------------|
| 0000h       | NAND SLC   |
| 0001h       | NAND MLC   |
| 0002h-FFFFh | Reserved   |

7.8 Start Address1 Register (R/W): F100h, default=0000h

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

**DFS** (Device Flash Core Select): it selects Flash Core in two Flash Core of DDP  
**FBA** (NAND Flash Block Address): NAND Flash block address which will be loaded or programmed or erased.

| Device  | Number of Block | FBA                |
|---------|-----------------|--------------------|
| 1Gb DDP | 1024            | DFS[15] & FBA[8:0] |
| 512Mb   | 512             | FBA[8:0]           |

7.9 Start Address2 Register (R/W): F101h, default=0000h

|     |    |                            |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|----|----------------------------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15  | 14 | 13                         | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBS |    | Reserved(0000000000000000) |    |    |    |   |   |   |   |   |   |   |   |   |   |

**DBS** (Device BufferRAM Select): it selects BufferRAM in two BufferRAM of DDP

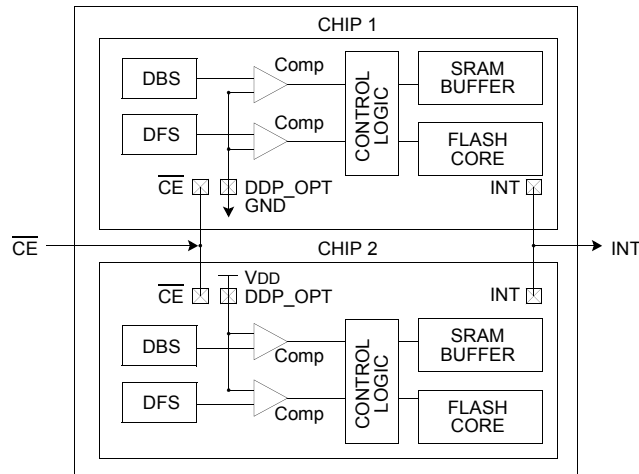


Figure 4. Chip selection method in DDP

7.10 Start Address3 Register (R/W): F102h, default=0000h

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

**FCBA** (NAND Flash Copy Back Block Address): NAND Flash destination block address which will be copy back programmed.

7.11 Start Address4 Register (R/W): F103h, default=0000h

|                    |    |    |    |    |    |   |      |   |   |   |   |      |   |   |   |
|--------------------|----|----|----|----|----|---|------|---|---|---|---|------|---|---|---|
| 15                 | 14 | 13 | 12 | 11 | 10 | 9 | 8    | 7 | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reserved(00000000) |    |    |    |    |    |   | FCPA |   |   |   |   | FCSA |   |   |   |

**FCPA** (NAND Flash Copy Back Page Address): NAND Flash destination page address in a block for copy back program operation.

FCPA(default value) = 000000

FCPA range : 000000~111111, 6bits for 64 pages

**FCSA** (NAND Flash Copy Back Sector Address): NAND Flash destination sector address in a page for copy back program operation.

FCSA(default value) = 00

FCSA range : 00~11, 2bits for 4 sectors

7.12 Start Address5 Register: F104h

: N/A

7.13 Start Address6 Register: F105h

: N/A

7.14 Start Address7 Register: F106h

: N/A

7.15 Start Address8 Register (R/W): F107h, default=0000h

|                     |    |    |    |    |    |   |   |     |   |   |   |   |   |     |   |
|---------------------|----|----|----|----|----|---|---|-----|---|---|---|---|---|-----|---|
| 15                  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1   | 0 |
| Reserved (00000000) |    |    |    |    |    |   |   | FPA |   |   |   |   |   | FSA |   |

**FPA** (NAND Flash Page Address): NAND Flash start page address in a block for page load or copy back program or program operation.

FPA(default value)=000000

FPA range: 000000~111111 , 6bits for 64 pages

**FSA** (Flash Sector Address): NAND Flash start sector address in a page for read or copy back program or program operation.

FSA(default value) = 00

FSA range : 00~11, 2bits for 4 sectors

7.16 Start Buffer Register (R/W): F200h, default=0000h

|                |    |    |    |     |    |   |   |                  |   |   |   |   |   |     |   |
|----------------|----|----|----|-----|----|---|---|------------------|---|---|---|---|---|-----|---|
| 15             | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7                | 6 | 5 | 4 | 3 | 2 | 1   | 0 |
| Reserved(0000) |    |    |    | BSA |    |   |   | Reserved(000000) |   |   |   |   |   | BSC |   |

**BSC** (BufferRAM Sector Count): this field specifies the number of sectors to be read or programmed or copy back programmed.

Its maximum count is 4 sectors at 00(default value)value. For a single sector access, it should be programmed as value 01.

However internal RAM buffer reached to 11vaule(max value), it count up to 00 value to satisfy BSC value.

For example1) If BSA=1010, BSC=11, then selected BufferRAM are '1010 ->1011 ->1000'

There is restriction in BootRAM case.

For example2) If BSA=0000, BSC should be 01 or 10.

If BSA=0001, BSC should be 01.

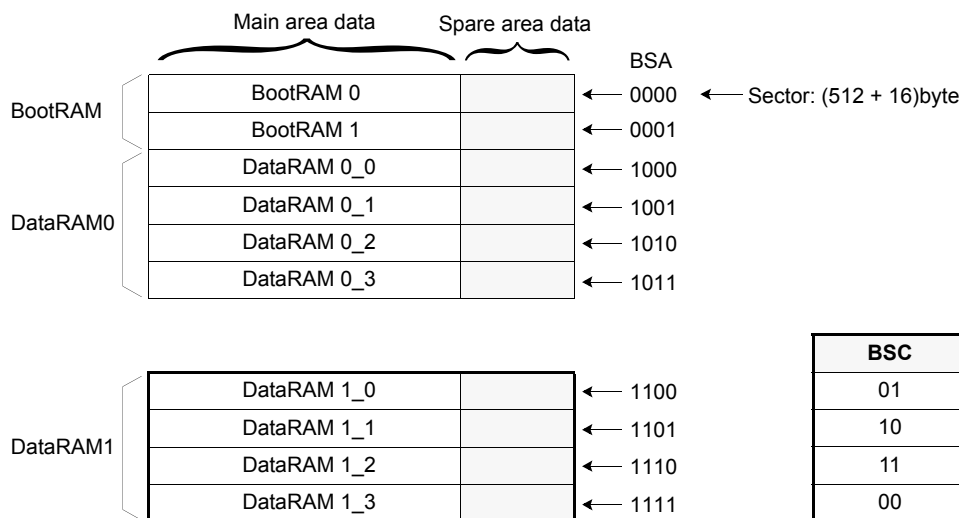
**BSA** (BufferRAM Sector Address): It is the place where data is placed and specifies the sector 0~3 in the internal BootRAM and DataRAM

**BSA[3]** is the selection bit between **BootRAM** and **DataRAM**

**BSA[2]** is the selection bit between **DataRAM0** and **DataRAM1**

**BSA[1:0]** are the selection bits for sectors in a **BufferRAM**

While one of **BootRAM** or **DataRAM0** interfaces with memory, the other **RAM** is inaccessible.



## 7.17 Command Register (R/W): F220h, default=0000h

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

**Command:** operation of the memory interface

| CMD           | Operation  | Acceptable command during busy |
|---------------|--|--------------------------------|
| 0000h         | Load single/multiple sector data unit into buffer                        | 00F0h, 00F3h                   |
| 0013h         | Load single/multiple spare sector into buffer                            | 00F0h, 00F3h                   |
| 0080h         | Program single/multiple sector data unit from buffer                     | 00F0h, 00F3h                   |
| 001Ah         | Program single/multiple spare area sector from buffer                    | 00F0h, 00F3h                   |
| 001Bh         | Copy back program  | 00F0h, 00F3h                   |
| 0023h         | Unlock NAND array block(s) from start block address to end block address | -                              |
| 002Ah         | Lock all NAND array block(s)   | -                              |
| 002Ch         | Lock-tight all locked block(s)   | -                              |
| 0094h         | Block Erase  | 00F0h, 00F3h                   |
| 00F0h         | Reset NAND Flash Core  | -                              |
| 00F3h         | Reset OneNAND <sup>1)</sup>  | -                              |
| 0075h - 0065h | OTP Access   | 00F0h, 00F3h                   |

**NOTE:**

1)'Reset OneNAND' (=Hot reset) command makes the registers(except RDYpol, INTpol, IOBE bits) and NAND Flash core into default state as the warm reset(=reset by RP pin).

This R/W register describes the operation of the OneNAND interface.

Note that all commands should be issued right after INT is turned from ready state to busy state. (i.e. right after 0 is written to INT register.) After any command is issued and the corresponding operation is completed, INT goes back to ready state. (00F0h and 00F3h may be accepted during busy state of some operations. Refer to the rightmost column of the command register table above.)



## 7.18 System Configuration 1 Register (R, R/W): 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              |   |   |   | R     |   |
| RM  | BRL |    | BL  |    |    | ECC | RDY pol | INT pol | IOBE | Reserved(0000) |   |   |   | BW PS |   |

**RM** (Read Mode): this field specifies the selection between asynchronous read mode and synchronous read mode

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

**BRL** (Burst Read Latency): this field specifies the initial access latency in the burst read transfer.

| BRL | Latency Cycles   |
|-----|------------------|
| 000 | 8(N/A)           |
| 001 | 9(N/A)           |
| 010 | 10(N/A)          |
| 011 | 3(N/A)           |
| 100 | 4(default, min.) |
| 101 | 5                |
| 110 | 6                |
| 111 | 7                |

**BL** (Burst Length): this field specifies the size of burst length during Sync. burst read. Wrap around and linear burst.

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

**ECC**: Error Correction Operation,  
0=with correction(default), 1=without correction(by-passed)

**RDYpol**: RDY signal polarity  
0=low for ready, 1=high for ready((default)

**INTpol**: INT pin polarity  
0=low for Interrupt pending , 1=high for Interrupt pending (default)

| INTpol | INT bit of Interrupt Status Register | INT Pin output |
|--------|--------------------------------------|----------------|
| 0      | 0                                    | 1              |
| 1      | 0                                    | 0              |

**IOBE**: I/O buffer enable for INT and RDY signals, INT and RDY outputs are HighZ at power-up, bit 7 and 6 become valid after IOBE is set to 1.  
IOBE can be reset only by Cold reset or by writing 0 to bit 5 of System Configuration 1 register.  
0=disable(default), 1=enable

**BWPS**: boot buffer write protect status,  
0=locked(default)

**7.19 System Configuration 2 Register : F222h**

: N/A

**7.22 Controller Status Register (R): F240h, default=0000h**

|    |    |    |    |    |     |                 |                 |      |                  |   |   |   |   |   |           |
|----|----|----|----|----|-----|-----------------|-----------------|------|------------------|---|---|---|---|---|-----------|
| 15 | 14 | 13 | 12 | 11 | 10  | 9               | 8               | 7    | 6                | 5 | 4 | 3 | 2 | 1 | 0         |
| CB | FC | RB | WB | EB | WRc | Reserv<br>ed(0) | Reserv<br>ed(0) | RSTB | Reserved(000000) |   |   |   |   |   | TO<br>(0) |

**CB**(Controller Busy) : this bit shows the overall internal status of OneNAND  
0=ready, 1=busy

**FC** (Fault Check): this bit shows whether host loads data from NAND Flash array into locked BootRAM or programs/erases locked block of NAND Flash array or put invalid command into the device.

| FC | Fault Check Result |
|----|--------------------|
| 0  | No fault           |
| 1  | Fault              |

**WRc** (Current Sector/Page Write Result): this bit shows current sector/page Program/Copy Back Program/Erase result of flash memory.

| WRc | Current Sector/Page Program/CopyBack.<br>Program/Erase Result |
|-----|---|
| 0   | Pass  |
| 1   | Fail  |

**TO** (Time Out): time out for load/program/copy back program/erase  
0=no time out(fixed)

**RB**(Read Busy) : this bit shows the Load operation status  
0=ready(default), 1=busy

**WB**(Write Busy) : this bit shows the Program operation status  
0=ready(default), 1=busy

**EB**(Erase Busy) : this bit shows the Erase operation status  
0=ready(default), 1=busy

**RSTB**(Reset Busy) : this bit shows the Reset operation status  
0=ready(default), 1=busy

table 2. Controller Status Register output for modes.

| Mode                     | Controller Status Register [15:0] |    |    |    |    |     |             |     |      |             |    |
|--------------------------|-----------------------------------|----|----|----|----|-----|-------------|-----|------|-------------|----|
|                          | CB                                | FC | RB | WB | EB | WRc | Reserved(0) | PRp | RSTB | Reserved(0) | TO |
| Load Ongoing             | 1                                 | 0  | 1  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Program Ongoing          | 1                                 | 0  | 0  | 1  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Erase Ongoing            | 1                                 | 0  | 0  | 0  | 1  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Reset Ongoing            | 1                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 1    | 000000      | 0  |
| Load OK                  | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Program OK               | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Erase OK                 | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Load Fail <sup>1)</sup>  | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Program Fail             | 0                                 | 0  | 0  | 0  | 0  | 1   | 0           | 0   | 0    | 000000      | 0  |
| Erase Fail               | 0                                 | 0  | 0  | 0  | 0  | 1   | 0           | 0   | 0    | 000000      | 0  |
| Load Reset <sup>2)</sup> | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Program Reset            | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Erase Reset              | 0                                 | 0  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Program Lock             | 0                                 | 1  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Erase Lock               | 0                                 | 1  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| Load Lock(Buffer Lock)   | 0                                 | 1  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| OTP Program Fail(Lock)   | 0                                 | 1  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |
| OTP Program Fail         | 0                                 | 0  | 0  | 0  | 0  | 1   | 0           | 0   | 0    | 000000      | 0  |
| Invalid Command          | 0                                 | 1  | 0  | 0  | 0  | 0   | 0           | 0   | 0    | 000000      | 0  |

**NOTE:** 1. ERm and/or ERs bits in ECC status register at Load Fail case is 10. (2bits error - uncorrectable)

2. ERm and ERs bits in ECC status register at Load Reset case are 00. (No error)

3. OTP Erase does not update the register and the previous value is kept.

**7.23 Interrupt Status Register (R/W): F241h, default=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) |   |   |   |  |

| Bit Address | Bit Name   | Default State |          | Valid States | Function          |
|-------------|--|---------------|----------|--------------|-------------------|
|             |  | Cold          | Warm/Hot |              |                   |
| 15          | INT(interrupt): the master interrupt bit   | 1             | 1        | 0            | Interrupt Off     |
|             | - Set to '1' of itself when one or more of RI, WI, EI and RSTI is set to '1', or Unlock(0023h), Lock(002Ah), Lock-tight(002Ch) or OTP access(0075h - 0065h) operation is completed.<br>- Cleared to '0' when by writing '0' to this bit or by reset(Cold/Warm/Hot reset).<br>'0' in this bit means that INT pin is low status.<br>(This INT bit is directly wired to the INT pin on the chip. INT pin goes low upon writing '0' to this bit when INTpol is high and goes high upon writing '0' to this bit when INTpol is low. ) |               |          | 0->1         | Interrupt Pending |
| 7           | RI(Read Interrupt):  | 1             | 0        | 0            | Interrupt Off     |
|             | - Set to '1' of itself at the completion of Load Operation (0000h, 0013h, or boot is done.)<br>- Cleared to '0' when by writing '0' to this bit or by reset (Cold/Warm/Hot reset).   |               |          | 0->1         | Interrupt Pending |
| 6           | WI(Write Interrupt):   | 0             | 0        | 0            | Interrupt Off     |
|             | - Set to '1' of itself at the completion of Program Operation (0080h, 001Ah, or 001Bh)<br>- Cleared to '0' when by writing '0' to this bit or by reset (Cold/Warm/Hot reset).  |               |          | 0->1         | Interrupt Pending |
| 5           | EI(Erase Interrupt):   | 0             | 0        | 0            | Interrupt Off     |
|             | - Set to '1' of itself at the completion of Erase Operation (0094h)<br>- Cleared to '0' when by writing '0' to this bit or by reset (Cold/Warm/Hot reset).   |               |          | 0->1         | Interrupt Pending |
| 4           | RSTI(Reset Interrupt):   | 0             | 1        | 0            | Interrupt Off     |
|             | - Set to '1' of itself at the completion of Reset Operation (00F0h, 00F3h, or warm reset is released.)<br>- Cleared to '0' when by writing '0' to this bit.  |               |          | 0->1         | Interrupt Pending |

**7.24 Start Block Address (R/W): F24Ch, default=0000h**

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

**SBA** (Start Block Address): Start NAND Flash block address to unlock in Write Protection mode, which precedes 'Unlock block command'.

**7.25 End Block Address (R/W): F24Dh, default=0000h**

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

**EBA** (End Block Address): End NAND Flash block address to unlock in Write Protection mode, which precedes 'Unlock block command'. EBA should be equal to or larger than SBA.

| Device | Number of Block | SBA/EBA |
|--------|-----------------|---------|
| 512Mb  | 512             | [8:0]   |

**7.26 NAND Flash Write Protection Status (R): F24Eh, default=0002h**

|                          |    |    |    |    |    |   |   |   |   |   |   |   |    |    |     |
|--------------------------|----|----|----|----|----|---|---|---|---|---|---|---|----|----|-----|
| 15                       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2  | 1  | 0   |
| Reserved(00000000000000) |    |    |    |    |    |   |   |   |   |   |   |   | US | LS | LTS |

US (Unlocked Status): '1' value of this bit specifies that there is unlocked block in NAND Flash.

LS (Locked Status): '1' value of this bit specifies that there is locked block in NAND Flash.

LTS (Lock-tighten Status): '1' value of this bit specifies that 'Locked block(s)' is lock-tighten.

**7.27 ECC Status Register(R): FF00h, default=0000h**

|      |    |    |      |    |    |      |   |   |      |   |   |      |   |   |      |  |  |      |  |  |      |  |  |
|------|----|----|------|----|----|------|---|---|------|---|---|------|---|---|------|--|--|------|--|--|------|--|--|
| 15   | 14 | 13 | 12   | 11 | 10 | 9    | 8 | 7 | 6    | 5 | 4 | 3    | 2 | 1 | 0    |  |  |      |  |  |      |  |  |
| ERm3 |    |    | ERs3 |    |    | ERm2 |   |   | ERs2 |   |   | ERm1 |   |   | ERs1 |  |  | ERm0 |  |  | ERs0 |  |  |

ERm (ECC Error for Main area data) & ERs (ECC Error for Spare area data)

ERm0/1/2/3 is for first/second/third/fourth selected sector main of BufferRAM, ERs0/1/2/3 is for first/second/third/fourth selected sector spare of BufferRAM.

ERm and ERs show the number of error in a sector as a result of ECC check at the load operation.

ERm and ERs bits are updated in boot loading operation, too.

| ERm, ERs | ECC Status                               |
|----------|--|
| 00       | No Error                                 |
| 01       | 1-bit error(correctable)                 |
| 10       | 2-bit error(uncorrectable) <sup>1)</sup> |
| 11       | Reserved                                 |

**NOTE:**

1. 3bits or more error detection is not supported.

**7.28 ECC Result of first selected Sector Main area data Register (R): FF01h, default=0000h**

|                |    |    |    |             |    |   |   |   |   |   |   |           |   |   |   |
|----------------|----|----|----|-------------|----|---|---|---|---|---|---|-----------|---|---|---|
| 15             | 14 | 13 | 12 | 11          | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reserved(0000) |    |    |    | ECCposWord0 |    |   |   |   |   |   |   | ECCposIO0 |   |   |   |

**7.29 ECC Result of first selected Sector Spare area data Register (R): FF02h, default=0000h**

|                      |    |    |    |    |    |   |   |   |   |               |   |   |           |   |   |
|----------------------|----|----|----|----|----|---|---|---|---|---------------|---|---|-----------|---|---|
| 15                   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5             | 4 | 3 | 2         | 1 | 0 |
| Reserved(0000000000) |    |    |    |    |    |   |   |   |   | ECClogSector0 |   |   | ECCposIO0 |   |   |

**7.30 ECC Result of second selected Sector Main area data Register (R): FF03h, default=0000h**

|                |    |    |    |             |    |   |   |   |   |   |   |           |   |   |   |
|----------------|----|----|----|-------------|----|---|---|---|---|---|---|-----------|---|---|---|
| 15             | 14 | 13 | 12 | 11          | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reserved(0000) |    |    |    | ECCposWord1 |    |   |   |   |   |   |   | ECCposIO1 |   |   |   |

**7.31 ECC Result of second selected Sector Spare area data Register (R): FF04h, default=0000h**

|                      |    |    |    |    |    |   |   |   |   |               |   |   |           |   |   |
|----------------------|----|----|----|----|----|---|---|---|---|---------------|---|---|-----------|---|---|
| 15                   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5             | 4 | 3 | 2         | 1 | 0 |
| Reserved(0000000000) |    |    |    |    |    |   |   |   |   | ECClogSector1 |   |   | ECCposIO1 |   |   |

**7.32 ECC Result of third selected Sector Main area data Register (R): FF05h, default=0000h**

|                |    |    |    |             |    |   |   |   |   |   |   |           |   |   |   |
|----------------|----|----|----|-------------|----|---|---|---|---|---|---|-----------|---|---|---|
| 15             | 14 | 13 | 12 | 11          | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reserved(0000) |    |    |    | ECCposWord2 |    |   |   |   |   |   |   | ECCposIO2 |   |   |   |

**7.33 ECC Result of third selected Sector Spare area data Register (R): FF06h, default=0000h**

|                      |    |    |    |    |    |   |   |   |   |               |   |   |           |   |   |
|----------------------|----|----|----|----|----|---|---|---|---|---------------|---|---|-----------|---|---|
| 15                   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5             | 4 | 3 | 2         | 1 | 0 |
| Reserved(0000000000) |    |    |    |    |    |   |   |   |   | ECClogSector2 |   |   | ECCposIO2 |   |   |

**7.34 ECC Result of fourth selected Sector Main area data Register (R): FF07h, default=0000h**

|                |    |    |    |             |    |   |   |   |   |   |   |           |   |   |   |
|----------------|----|----|----|-------------|----|---|---|---|---|---|---|-----------|---|---|---|
| 15             | 14 | 13 | 12 | 11          | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reserved(0000) |    |    |    | ECCposWord3 |    |   |   |   |   |   |   | ECCposIO3 |   |   |   |

**7.35 ECC Result of fourth selected Sector Spare area data Register (R): FF08h, default=0000h**

|                      |    |    |    |    |    |   |   |   |   |               |   |   |           |   |   |
|----------------------|----|----|----|----|----|---|---|---|---|---------------|---|---|-----------|---|---|
| 15                   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5             | 4 | 3 | 2         | 1 | 0 |
| Reserved(0000000000) |    |    |    |    |    |   |   |   |   | ECClogSector3 |   |   | ECCposIO3 |   |   |

**NOTE:**

1. ECCposWord: ECC error position address that selects one of Main area data(256words)
2. ECCposIO: ECC error position address which selects one of sixteen DQs (DQ 0~DQ 15).
3. ECClogSector: ECC error position address that selects one of the 2nd word and LSB of the 3rd word of spare area. Refer to the below table.

**ECClogSector Information [5:4]**

| ECClogSector | Error Position |
|--------------|----------------|
| 00           | 2nd word       |
| 01           | 3rd word       |
| 10, 11       | Reserved       |

## 8 Device Operation

The device supports both a limited command based and a register based interface for performing operations on the device, reading device ID, writing data to buffer etc. The command based interface is active in the boot partition, i.e. commands can only be written with a boot area address. Boot area data is only returned if no command has been issued prior to the read.

### 8.1 Command based operation

The entire address range, except for the boot area, can be used for the data buffer. All commands are written to the boot partition. 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 which include address and data or writing an improper command will terminate the previous command sequence and make the device go to the ready status. The defined valid command sequences are stated in Table3.

**Table 3. Command Sequences**

| Command Definition                     |      | Cycles | 1st cycle        | 2nd cycle           |
|--|------|--------|------------------|---------------------|
| Read Data from Buffer                  | Add  | 1      | DP <sup>1)</sup> |                     |
|  | Data |        | Data             |                     |
| Write Data to Buffer                   | Add  | 1      | DP               |                     |
|  | Data |        | Data             |                     |
| Reset OneNAND                          | Add  | 1      | BP <sup>2)</sup> |                     |
|  | Data |        | 00F0h            |                     |
| Load Data into Buffer <sup>3)</sup>    | Add  | 2      | BP               | BP                  |
|  | Data |        | 00E0h            | 0000h <sup>4)</sup> |
| Read Identification Data <sup>6)</sup> | Add  | 2      | BP               | XXXXh <sup>5)</sup> |
|  | Data |        | 0090h            | Data                |

**NOTE:**

- 1) DP(Data Partition) : DataRAM Area
- 2) BP(Boot Partition) : BootRAM Area [0000h ~ 01FFh, 8000h ~ 800Fh]. It is locked after power-up.
- 3) Load Data into Buffer operation is available within a block(128KB)
- 4) Load 2KB unit into DataRAM0. Current Start address(FPA) is automatically incremented by 2KB unit after the load.
- 5) 0000h -> Data is Manufacturer ID  
0001h -> Data is Device ID
- 6) WE toggling can terminate 'Read Identification Data' operation.

#### 8.1.1 Read Data from Buffer

Buffer can be read by addressing a read to a wanted buffer area

#### 8.1.2 Write Data to Buffer

Buffer can be written by addressing a write to a wanted buffer area

#### 8.1.3 Reset OneNAND

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

#### 8.1.4 Load Data into Buffer

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. This operation refers to FBA and FPA. FSA, BSA, and BSC are not considered. 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. 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.

#### 8.1.5 Read Identification Data

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 Table5.

Table 4. Identification data description

| Address | Data Out        |                  |
|---------|-----------------|------------------|
| 0000h   | Manufacturer ID | 00ECh            |
| 0001h   | Device ID       | refer to table 1 |

Device Bus Operations

| Operation   | $\overline{CE}$ | $\overline{OE}$ | $\overline{WE}$ | ADD0~15 | DQ0~15         | $\overline{RP}$ | CLK | $\overline{AVD}$ |
|---|-----------------|-----------------|-----------------|---------|----------------|-----------------|-----|------------------|
| Standby   | H               | X               | X               | X       | High-Z         | H               | X   | X                |
| Warm Reset  | X               | 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   |                  |
| Load Initial Burst Address  | L               | H               | H               | Add. In | X              | H               |     |                  |
| Burst Read  | L               | L               | H               | X       | Burst Data Out | H               |     | X                |
| Terminate Burst Read Cycle  | H               | X               | H               | X       | High-Z         | H               | X   | X                |
| Terminate Burst Read Cycle via $\overline{RP}$                    | X               | X               | X               | X       | High-Z         | L               | X   | X                |
| Terminate Current Burst Read Cycle and Start New Burst Read Cycle |                 | H               | H               | Add. In | High-Z         | H               |     |                  |

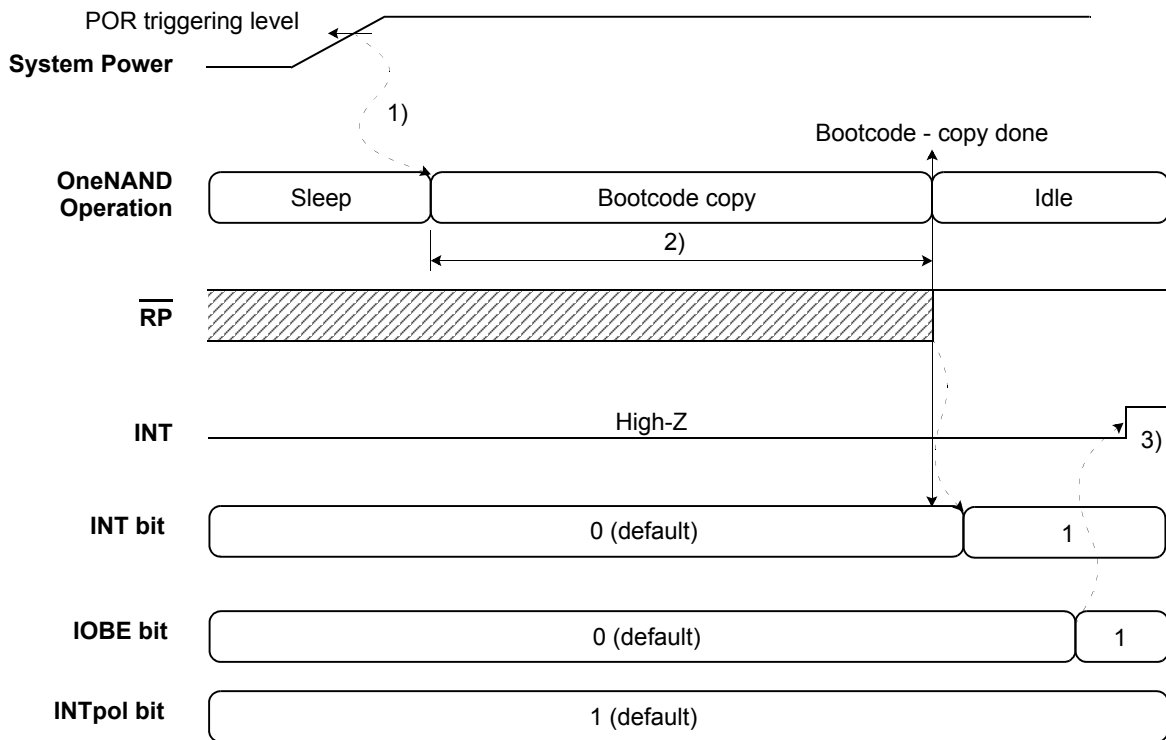
Note : L=V<sub>IL</sub> (Low), H=V<sub>IH</sub> (High), X=Don't Care.



**Reset Mode**

**Cold Reset**

At system power-up, the voltage detector in the device detects the rising edge of Vcc and releases internal power-up reset signal which triggers bootcode loading. Bootcode loading means that the boot loader in the device copies designated sized data(1KB) from the beginning of memory to the BootRAM.

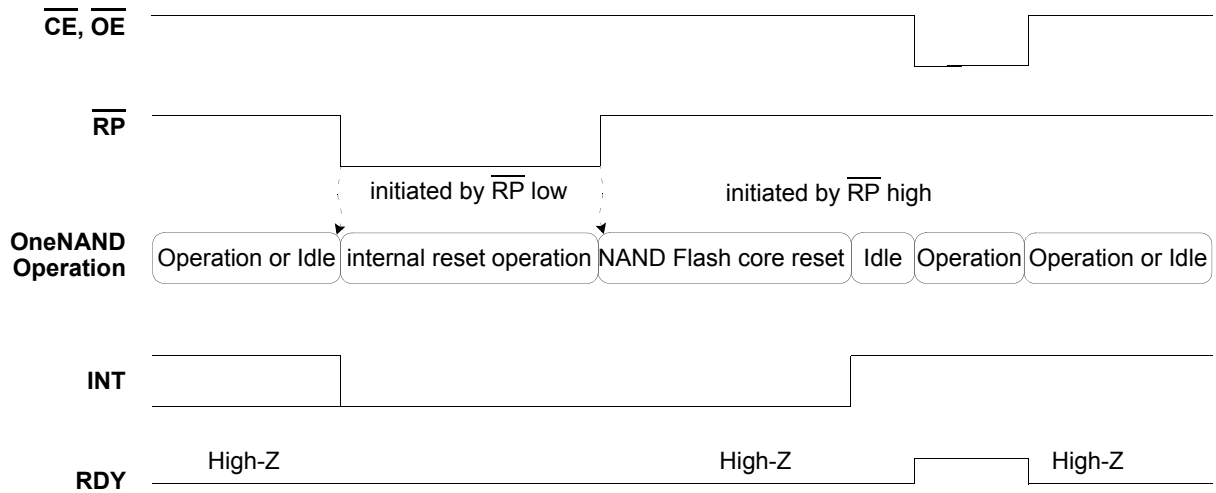


- Note: 1) Bootcode copy operation starts 400us later than POR activation.  
 The system power should reach Vcc after POR triggering level(typ. 1.5V) within 400us for valid boot code data.  
 2) 1K bytes Bootcode copy takes 70us(estimated) from sector0 and sector1/page0/block0 of NAND Flash array to 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'

**Figure 5. Cold Reset Timings**

**Warm Reset**

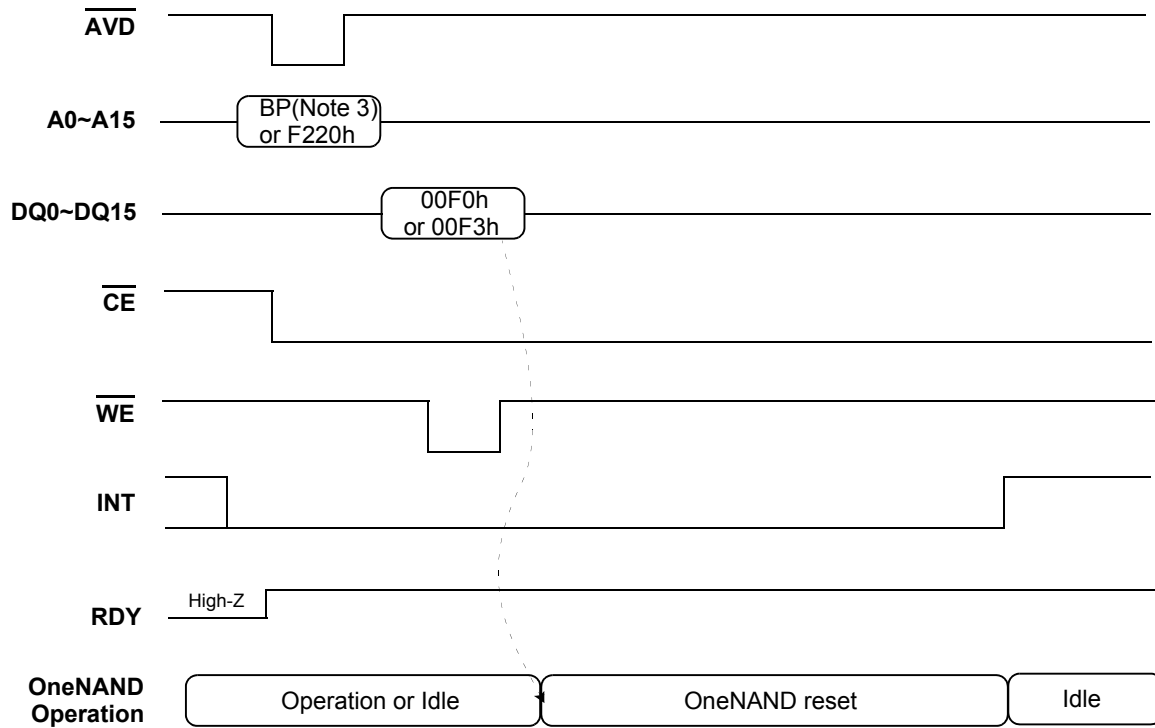
Warm reset means that the host resets the device by  $\overline{RP}$  pin, and then the device logic stops all current operation and executes internal reset operation (Note 1) synchronized with the falling edge of  $\overline{RP}$  and resets current NAND Flash core operation synchronized with the rising edge of  $\overline{RP}$ . The device logic will not be reset in case  $\overline{RP}$  pulses shorter than 200ns, but the device guarantees the logic reset operation in case  $\overline{RP}$  pulse is longer than 200ns. NAND Flash core reset will abort current NAND Flash Core operation. The contents of memory cells being altered are no longer valid as the data will be partially programmed or erased. Warm reset has no effect on contents of BootRAM and DataRAM.



**Figure 6. Warm Reset Timings**

**Hot Reset**

Hot reset means that the host resets the device by reset command(Note 2), and then the device logic stops all current operation and executes internal reset operation(Note 1), and resets current NAND Flash core operation. Hot reset has no effect on contents of BootRAM and DataRAM.



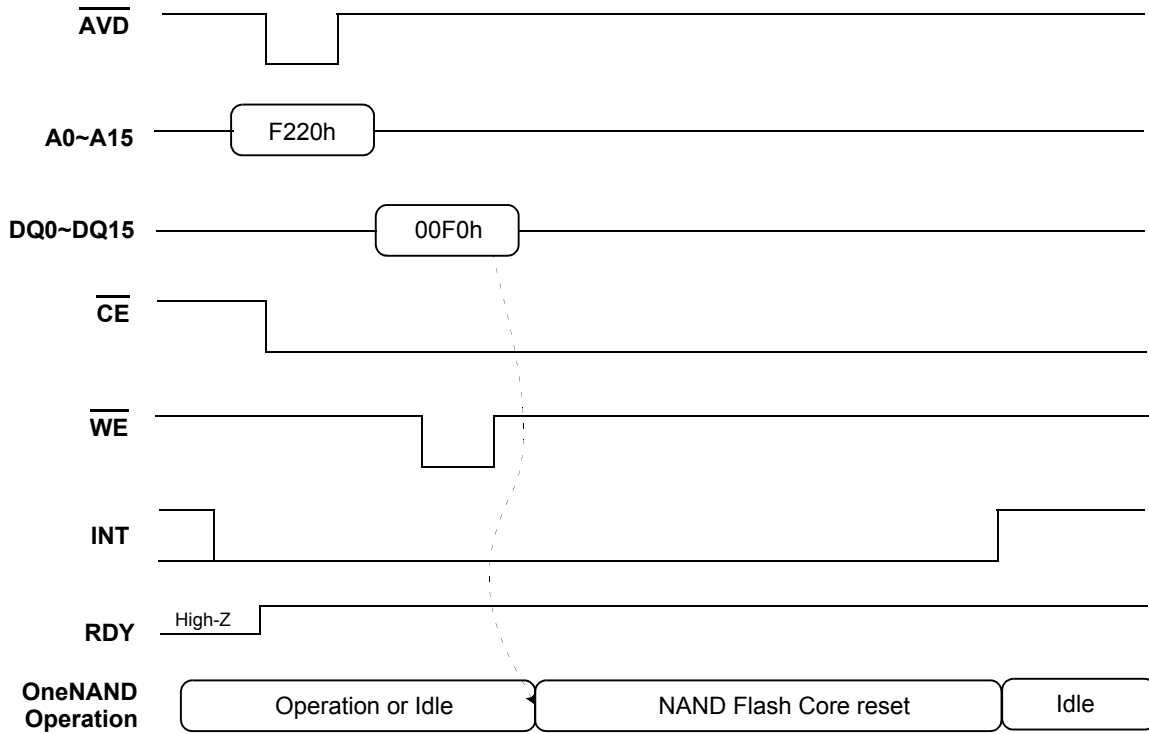
**Figure 7. Hot Reset Timings**

**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]

**NAND Flash Core Reset**

Host can reset NAND Flash Core operation by 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.



**Figure 8. NAND Flash Core Reset Timings**

Table 7. Internal Register reset

|       | Internal Registers                                  | Default | Cold Reset | Warn Reset (RP) | Hot Reset (00F3h) | Hot Reset (BP-F0) | NAND Flash Reset(00F0h) |
|-------|---|---------|------------|-----------------|-------------------|-------------------|-------------------------|
| F000h | Manufacturer ID Register (R)                        | 00ECh   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F001h | Device ID Register (R)                              | Note3   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F002h | Version ID Register (rR): 54MHz                     | 001Eh   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F003h | Data Buffer size Register (R)                       | 0800h   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F004h | Boot Buffer size Register (R)                       | 0200h   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F005h | Amount of Buffers Register (R)                      | 0201h   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F006h | Technology Register (R)                             | 0000h   | N/A        | N/A             | N/A               | N/A               | N/A                     |
| F100h | Start Address1 Register (R/W): DFS, FBA             | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F101h | Start Address2 Register (R/W): DBS                  | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F102h | Start Address3 Register (R/W): FCBA                 | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F103h | Start Address4 Register (R/W): FCPA, FCSA           | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F107h | Start Address5 Register (R/W): FPA, FSA             | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F200h | Start Buffer Register (R/W): BSA, BSC               | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F220h | Command Register (R/W)                              | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F221h | System Configuration 1 Register (R/W)               | 40C0h   | 40C0h      | O (Note1)       | O (Note1)         |                   | N/A                     |
| F240h | Controller Status Register (R)                      | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| F241h | Interrupt Status Register (R/W)                     | -       | 8080h      | 8010h           | 8010h             | 8010h             | N/A                     |
| F24Ch | Start Block Address (R/W)                           | 0000h   | 0000h      | 0000h           | N/A               | N/A               | N/A                     |
| F24Dh | End Block Address (R/W)                             | 0000h   | 0000h      | 0000h           | N/A               | N/A               | N/A                     |
| F24Eh | NAND Flash Write Protection Status (R)              | 0002h   | 0002h      | 0002h           | N/A               | N/A               | N/A                     |
| FF00h | ECC Status Register (R) (Note2)                     | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF01h | ECC Result of Sector 0 Main area data Register(R)   | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF02h | ECC Result of Sector 0 Spare area data Register (R) | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF03h | ECC Result of Sector 1 Main area data Register(R)   | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF04h | ECC Result of Sector 1 Spare area data Register (R) | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF05h | ECC Result of Sector 2 Main area data Register(R)   | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF06h | ECC Result of Sector 2 Spare area data Register (R) | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF07h | ECC Result of Sector 3 Main area data Register(R)   | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |
| FF08h | ECC Result of Sector 3 Spare area data Register (R) | 0000h   | 0000h      | 0000h           | 0000h             | 0000h             | N/A                     |

**NOTE:** 1) RDYpol, INTpol, and IOBE are reset by Cold reset. The other bits are reset by Cold/Warm/Hot reset.

2) ECC Status Register & ECC Result Registers are reset when any command is issued.

3) Refer to table 1

## Write Protection

### Write Protection for BootRAM

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

### Write Protection for NAND Flash array

#### Write Protection Modes

The device offers both hardware and software write protection features for NAND Flash array. The software write protection feature is used by writing Lock command or Lock-tight command to command register;The 002Ah or 002Ch command is written into F220h register. And the hardware write protection feature is used by executing cold or warm reset. The default state is locked, and all NAND Flash array goes to locked state after cold or warm reset.

#### Write Protection Commands

The instant secured block protects code and data by allowing blocks to be locked or lock-tighten. The write protection scheme offers two levels of protection. The first allows software-only control of write protection(useful for frequently changed data blocks), while the second requires hardware interaction before locking can be changed(protects infrequently changed code blocks).

The followings summarize the locking functionality

- > All blocks power-up in a locked state. Unlock commands can unlock these blocks.
- >The lock-tight command makes locked block(s) lock-tighten block(s). And lock-tight state can be returned to lock state only when cold or warm reset is asserted.
- > Lock-tighten blocks offer the user an additional level of write protection beyond that of a regular locked block. Lock-tighten block can't have it's state changed by software, it can be changed by warm reset or cold reset.
- > Unlock start and end block address are reflected immediately to the device only when the unlock command is issued, and NAND Flash write protection status register is also updated at that time.
- > Unlocked blocks can be programmed or erased.
- > Only one consecutive area can be released to unlock state from lock state, i.e unlocking multi area is not available.
- > Partial block lock (a range) is not available, i.e lock operation is only available for all blocks.

#### Write Protection Status

The device current 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. These Write Protection status registers are updated when Write Protection command is entered.

The followings summarize locking status.

example1)

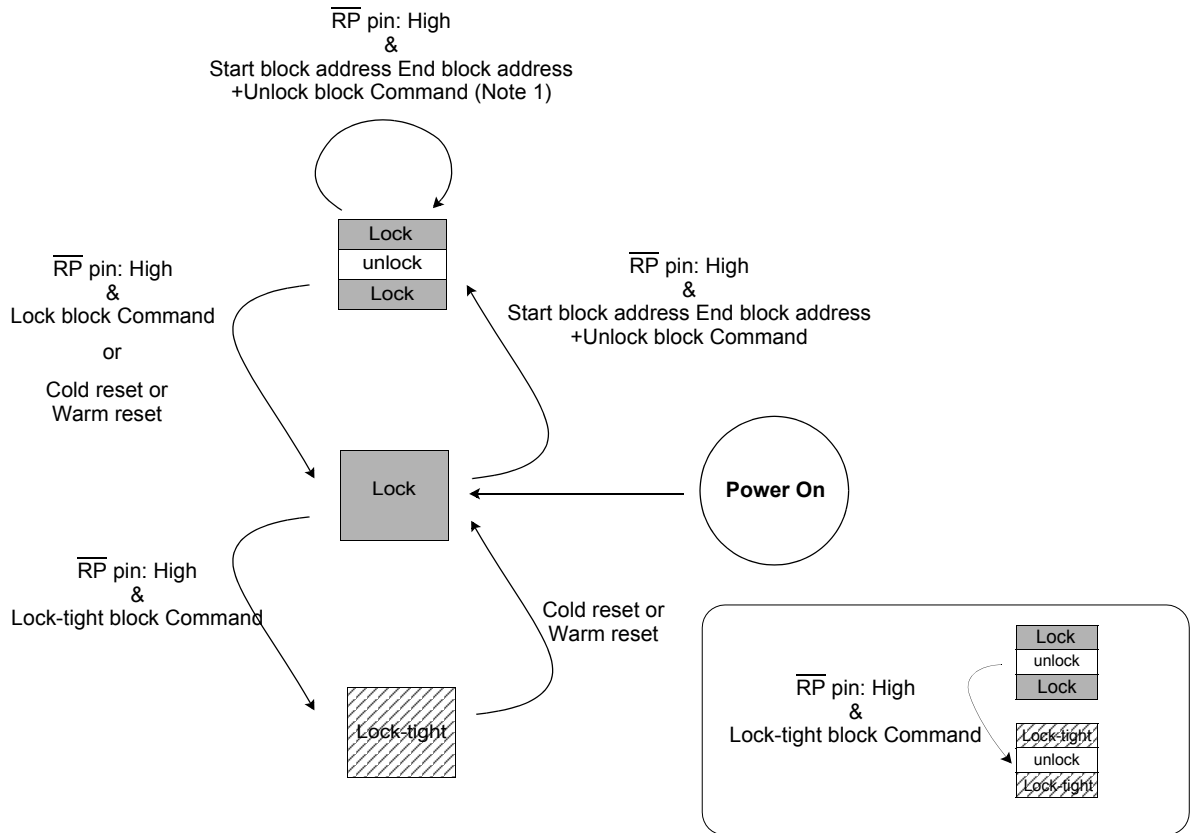
In default, [2:0] values are 010.

- > If host executes unlock block operation, then [2:0] values turn to 110.
- > If host executes lock-tight block operation, then [2:0] values turn to 101.

example2)

If host executes lock block operation, then [2:0] values turn to 010.

- > If host executes lock-tight block operation, then [2:0] values turn to 001.
- > If cold or warm reset is entered, then [2:0] values turn to 010.



**NOTE:**

1. Unlock range(from Start block address to End block address) can be modified by unlock command sequence(Start block address+End block address).

**Figure 9. State diagram of NAND Flash Write Protection**

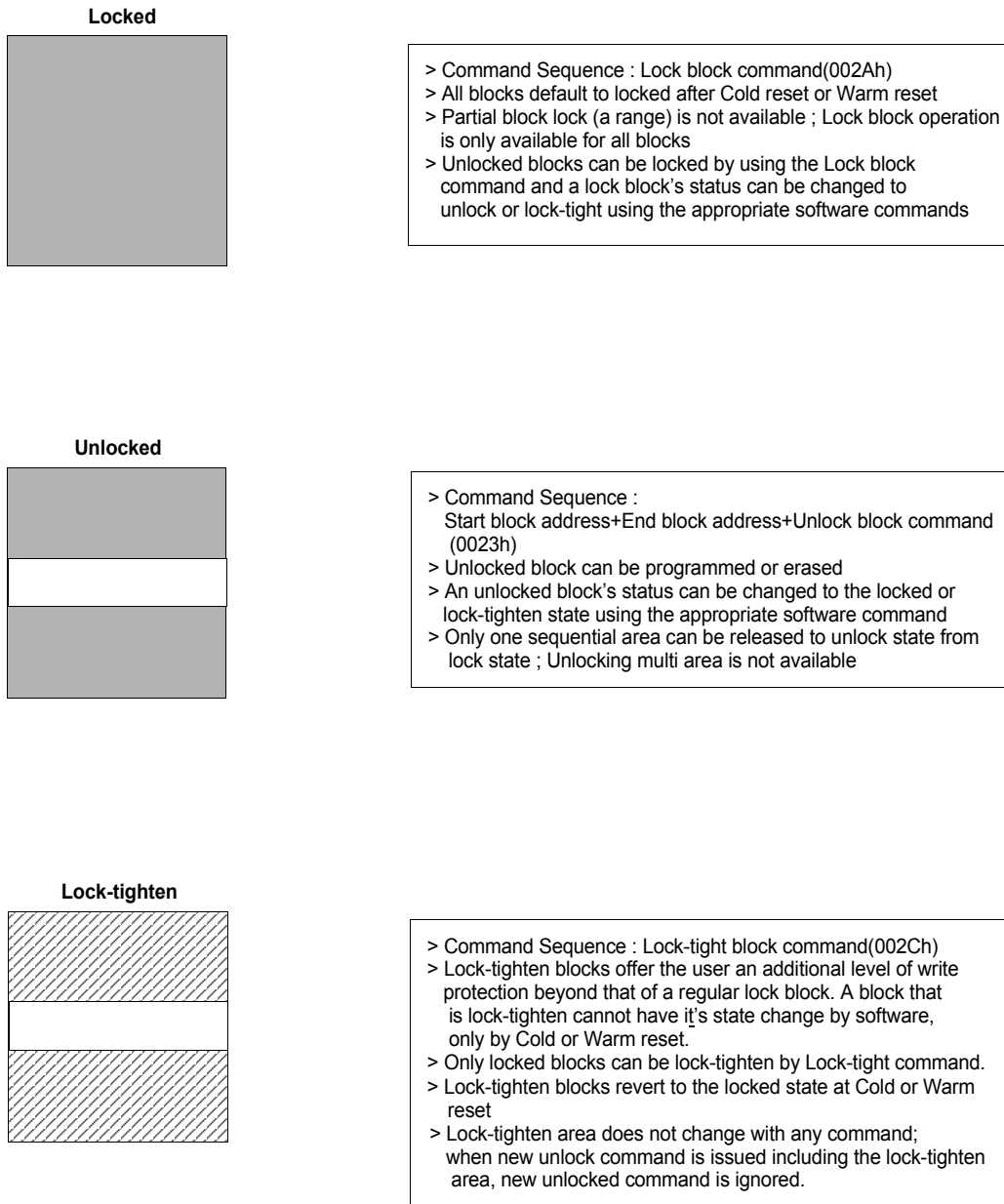


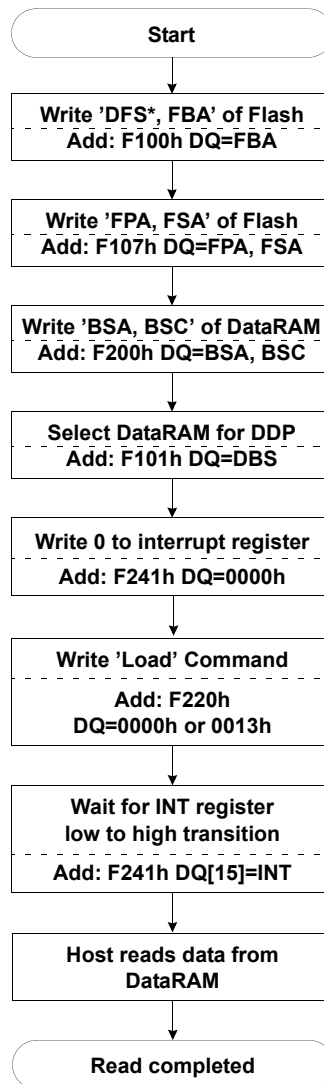
Figure 10. Operations of NAND Flash Write Protection



## Load Operation

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. The device transfers the data from NAND Flash array into the BufferRAM. The ECC is checked and any detected and corrected error is reported in the status response as well as any unrecoverable error. When the BufferRAM has been filled an interrupt is issued to the host in order to read the contents of the BufferRAM. The read from the BufferRAM consist of asynchronous read mode or synchronous read mode. The status information related to the BufferRAM fill operation can be checked by the host if required.

The device provides dual data buffer memory architecture. The device is capable of data-read operation from one data buffer and data-load operation to the other data buffer simultaneously. Refer to the information for more details in "Read while Load operation".



\* DBS, DFS is for DDP

Figure 11. Load operation flow-chart

**Read Operation**

The device has two read configurations ; Asynchronous read and Synchronous burst read. The initial state machine makes the device to be automatically entered into asynchronous read mode to prevent the memory content from spurious altering upon device power up or after a hardware reset. No commands are required to retrieve data in asynchronous mode. The synchronous mode will be enabled by setting RM bit of System configuration1 register to Synchronous read mode.

**Asynchronous Read Mode (RM = 0)**

For the asynchronous read mode a valid address should be asserted on ADD0-ADD15, while driving  $\overline{AVD}$  and  $\overline{CE}$  to  $V_{IL}$ .  $\overline{WE}$  should remain at  $V_{IH}$ . The data will appear on ADD15-ADD0. Address access time ( $t_{AA}$ ) is equal to the delay from valid addresses to valid output data. The chip enable access time( $t_{cE}$ ) is the delay from the falling edge of  $\overline{CE}$  to valid data at the outputs. The output enable access time( $t_{oE}$ ) is the delay from the falling edge of  $\overline{OE}$  to valid data at the output.

**Synchronous (Burst) Read Mode (RM = 1)**

The device is capable of continuous linear burst operation and linear burst operation of a preset length. For the burst mode, the host should determine how many clock cycles are desired for the initial word( $t_{iAA}$ ) of each burst access using BRL bit of System configuration 1 register. The registers also can be read during burst read mode by using  $\overline{AVD}$  signal with a address. To initiate the synchronous read again, a new address during  $\overline{CE}$  low toggle is needed after the host has completed status reads or the device has completed the program or erase operation.

**Continuous Linear Burst Read**

The initial word is output  $t_{iAA}$  after the rising edge of the first CLK cycle. Subsequent words are output  $t_{bA}$  after the rising edge of each successive clock cycle, which automatically increments the internal address counter. The RDY output indicates this condition to the system by pulsing low. The device will continue to output sequential burst data, wrapping around after it reaches the designated location(See Figure 12 for address map information) until the system asserts  $\overline{CE}$  high,  $\overline{RP}$  low or  $\overline{AVD}$  low in conjunction with a new address. The cold/warm/hot reset or asserting  $\overline{CE}$  high or  $\overline{WE}$  low pulse terminate the burst read operation.

**If the device is accessed synchronously while it is set to asynchronous read mode, it is possible to read out the first data without problems.**

| Division       | Add.map(word order) |          |
|----------------|---------------------|----------|
| BootM(0.5Kw)   | 0000h~01FFh         | Buffer0  |
| BufM 0(1Kw)    | 0200h~05FFh         |          |
| BufM 1(1Kw)    | 0600h~09FFh         | Buffer1  |
| Reserved Main  | 0A00h~7FFFh         | N/A Reg. |
| BootS(16w)     | 8000H~800Fh         | Buffer0  |
| BufS 0(32w)    | 8010h~802Fh         |          |
| BufS 1(32w)    | 8030h~804Fh         | Buffer1  |
| Reserved Spare | 8050h~8FFFh         | N/A Reg. |
| Reserved Reg.  | 9000h~EFFFh         |          |
| Register(4Kw)  | F000h~FFFFh         | Reg.     |

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

**Figure 12. The boundary of synchronous read**

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

As well as the Continuous Linear Burst Mode, there are four(4 & 8 & 16 & 32 word) (Note1) linear wrap-around mode, in which a fixed number of words are read from consecutive addresses. When the last word in the burst mode is reached, assert /CE and /OE high to terminate the operation. In these modes, the start address for burst read can be any address of address map.

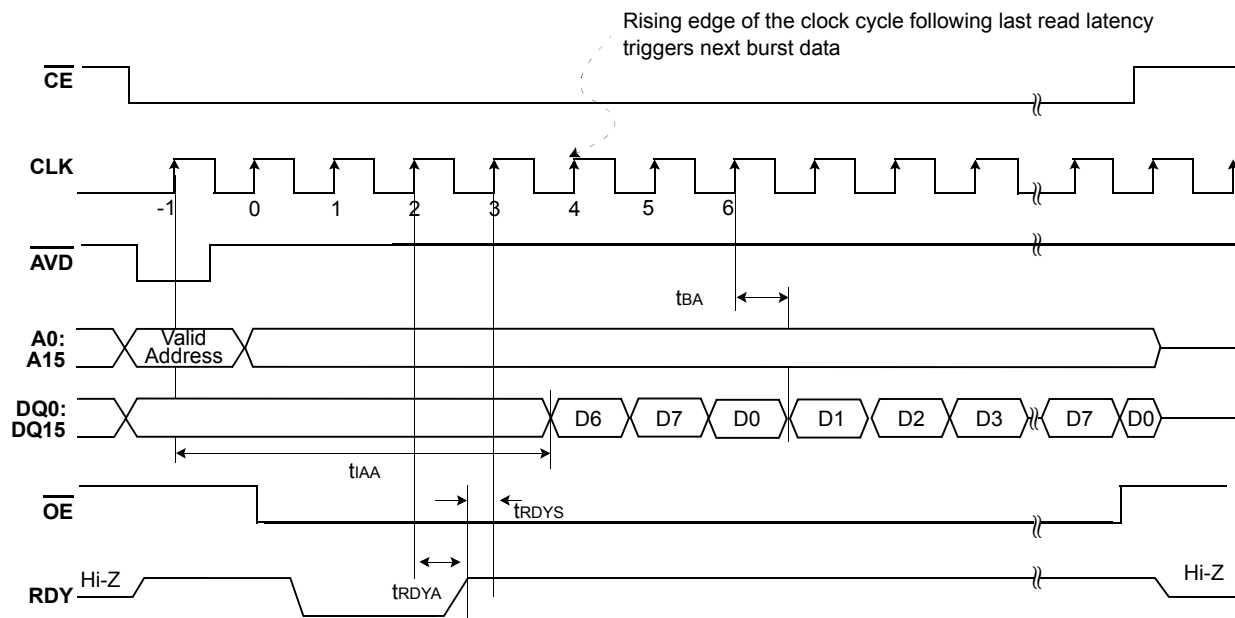
(Note 1) 32 word linear burst read isn't available on spare area BufferRAM

**Table 6. 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...   |
|             | .           | .                               | .            | .                    | .                             | .                             |

### Programmable Burst Read Latency

The programmable burst read latency feature indicates to the device the number of additional clock cycles that must elapse after  $\overline{AVD}$  is driven active before data will be available. Upon power up, the number of total initial access cycles defaults to four clocks. The number of total initial access cycles is programmable from four to seven cycles.



**Figure 13. Example of 4clock Burst Read Latency**

### Handshaking

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 cycle for optimal burst mode, the host should use the programmable burst read latency configuration.(See "System Configuration1 Register" for details.) The rising edge of RDY which is derived from 1 clock ahead of data fetch clock indicates the initial word of valid burst data.

### Output Disable Mode

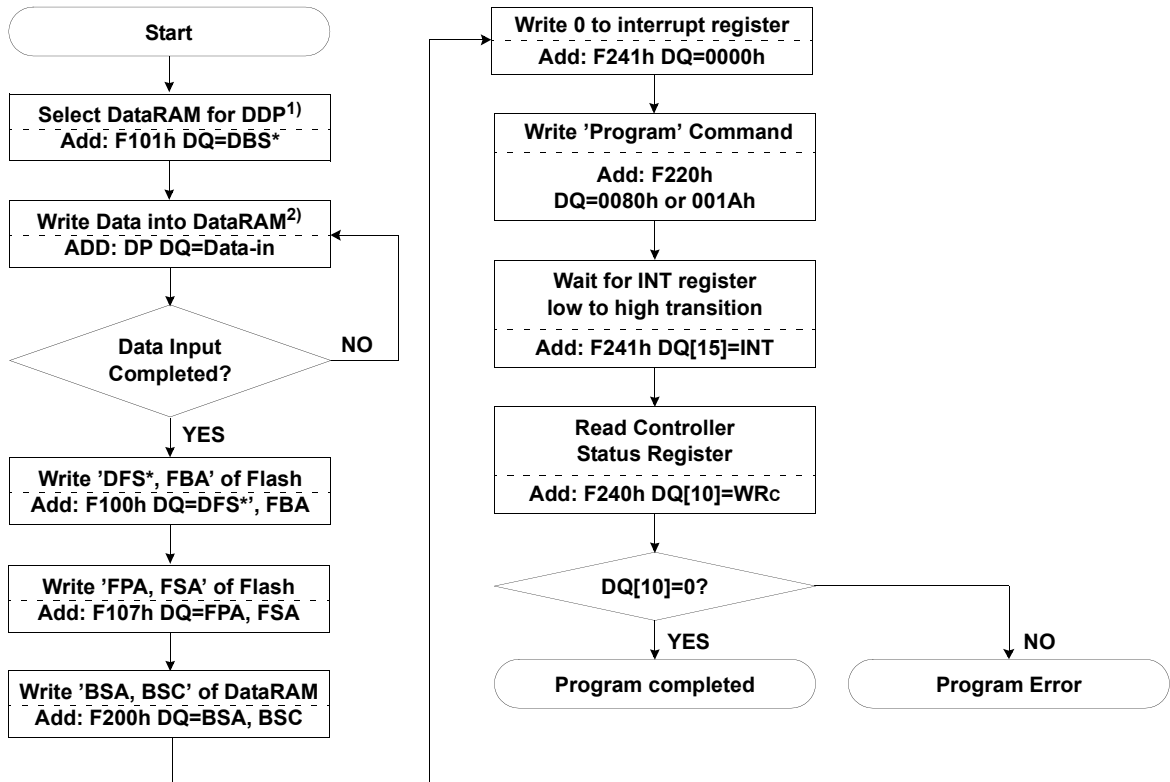
When the CE or OE input is at  $V_{IH}$ , output from the device is disabled. The outputs are placed in the high impedance state.

**Program Operation**

The device can be programmed in data unit. Programming is writing 0's into the memory array by executing the internal program routine. In order to perform the Internal Program Routine, command sequence is necessary. First, host sets the address of the BufferRAM and the memory location and loads the data to be programmed into the BufferRAM. Second, program command initiates the internal program routine. During the execution of the Routine, the host is not required to provide further controls or timings. During the Internal Program Routine, commands except reset command written to the device will be ignored.

Note that a reset during a program operation will cause data corruption at the corresponding location.

The device provides dual data buffer memory architecture. The device is capable of data-write operation from host to one of data buffers during program operation from another data buffer to Flash simultaneously. Refer to the information for more details in "Read while Load operation".



\* 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.

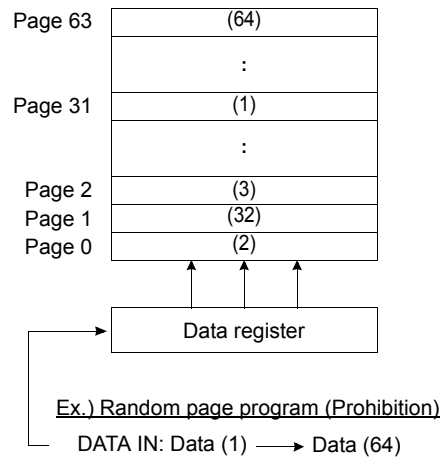
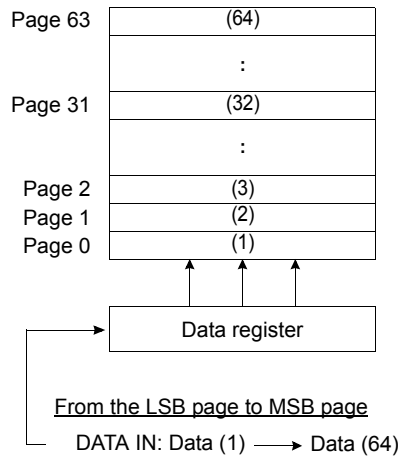
Note 1) This must happen before data input

2) Data input could be done anywhere between "Start" and "Write Program Command".

Figure 14. Program operation flow-chart

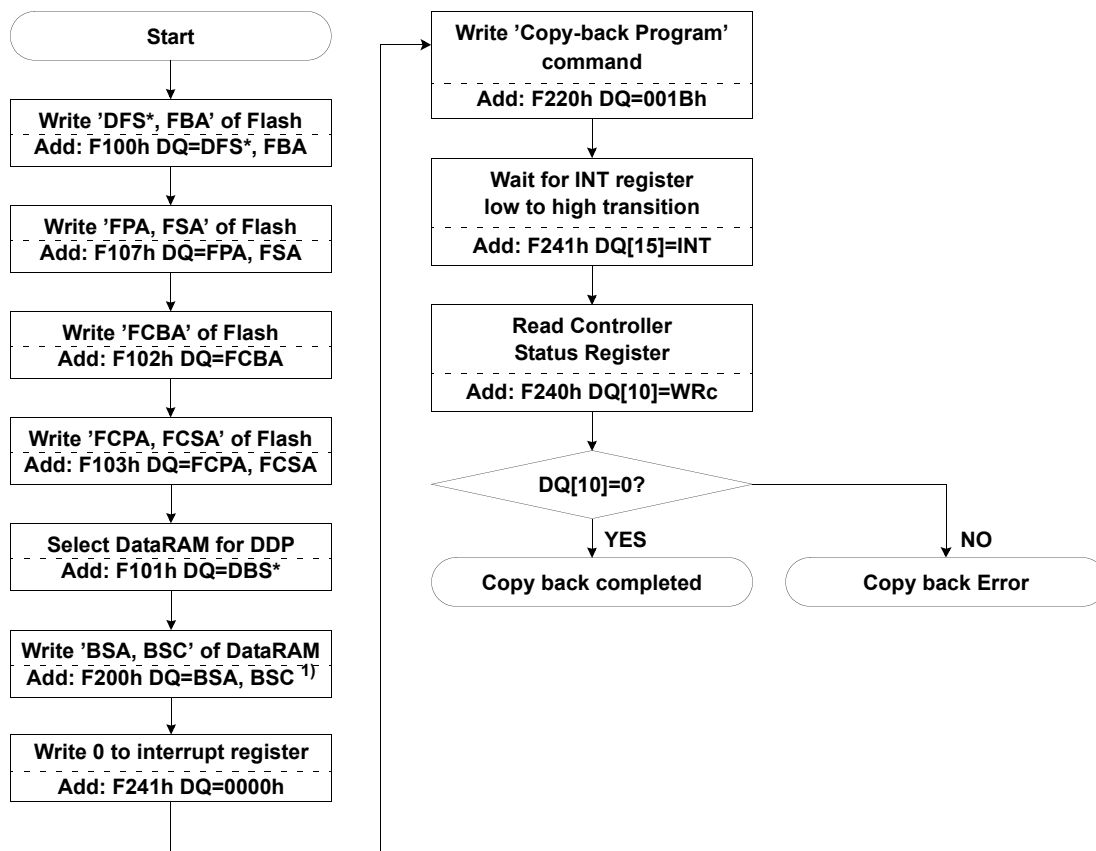
**Addressing for program operation**

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.



**Copy-back Program Operation**

The copy-back program is configured to quickly and efficiently rewrite data stored in one page by sector unit(1/2/3/4 sector) without utilizing an external memory. Since the time-consuming cycles of serial access and re-loading cycles are removed, the system performance is improved. The benefit is especially obvious when a portion of a block is updated and the rest of the block also need to be copied to the newly assigned free block. The operation for performing a copy-back program is a sequential execution of page-read without serial access and copying-program with the address of destination page.



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

\* DBS, DFS is for DDP

Note 1) Selected DataRAM by BSA & BSC is used for Copy back operation, so previous data is overwritten.

Figure 15. Copy back program operation flow-chart

Copy-Back Program Operation with Random Data Input

The Copy-Back Program Operation with Random Data Input in OneNAND consists of 2 phase, 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 OneNAND, accumulation of 1 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.

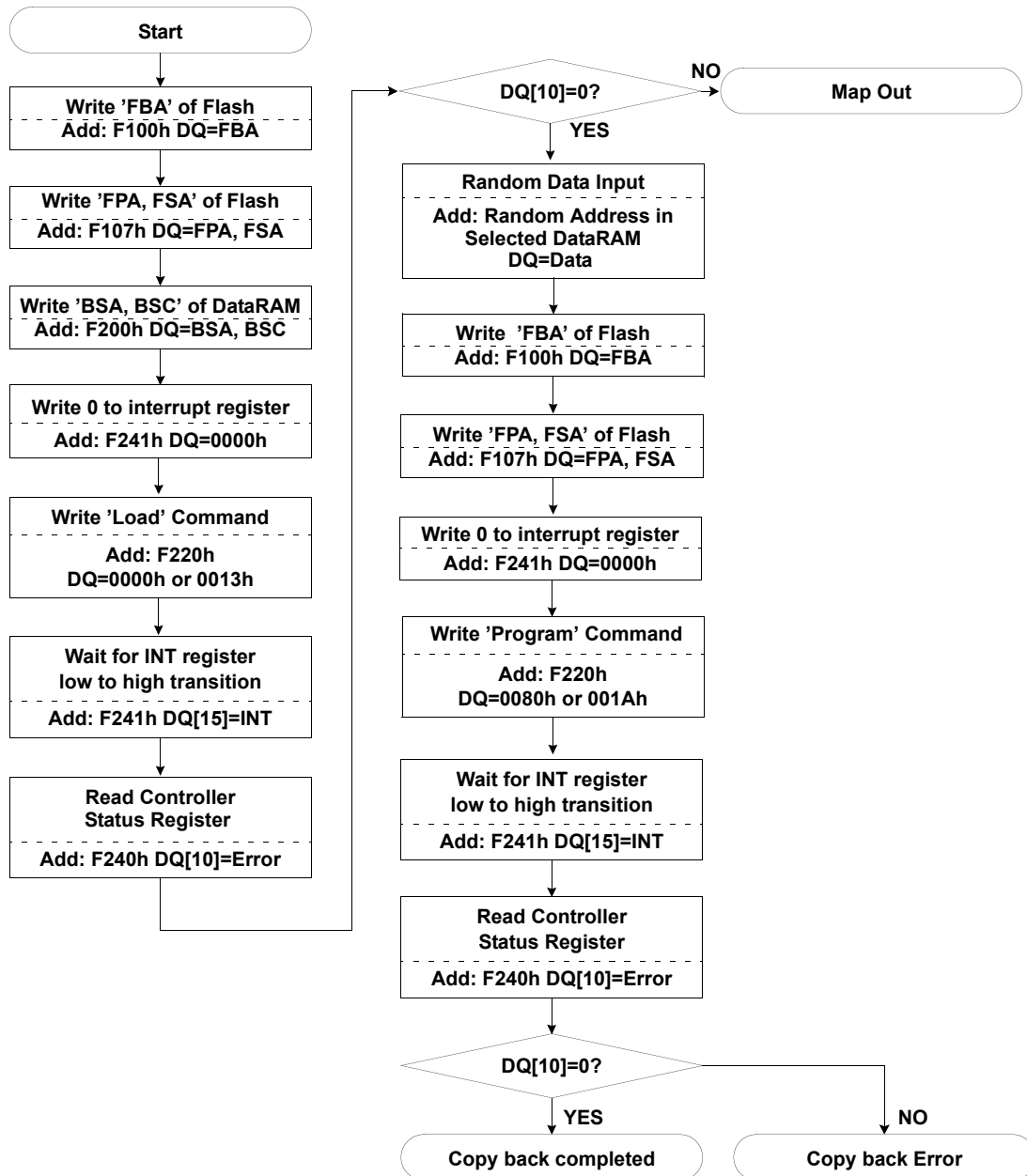


Figure 16. Copy-Back Program Operation with Random Data Input Flow Chart

Erase Operation

The device can be erased in block unit. To erase a block is to write 1's into the desired memory block by executing the Internal Erase Routine. In order to perform the Internal Erase Routine, command sequence is necessary. First, host sets the block address of the memory location. Second, 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, commands except reset command written to the device will be ignored.

Note that a reset during a erase operation will cause data corruption at the corresponding location.

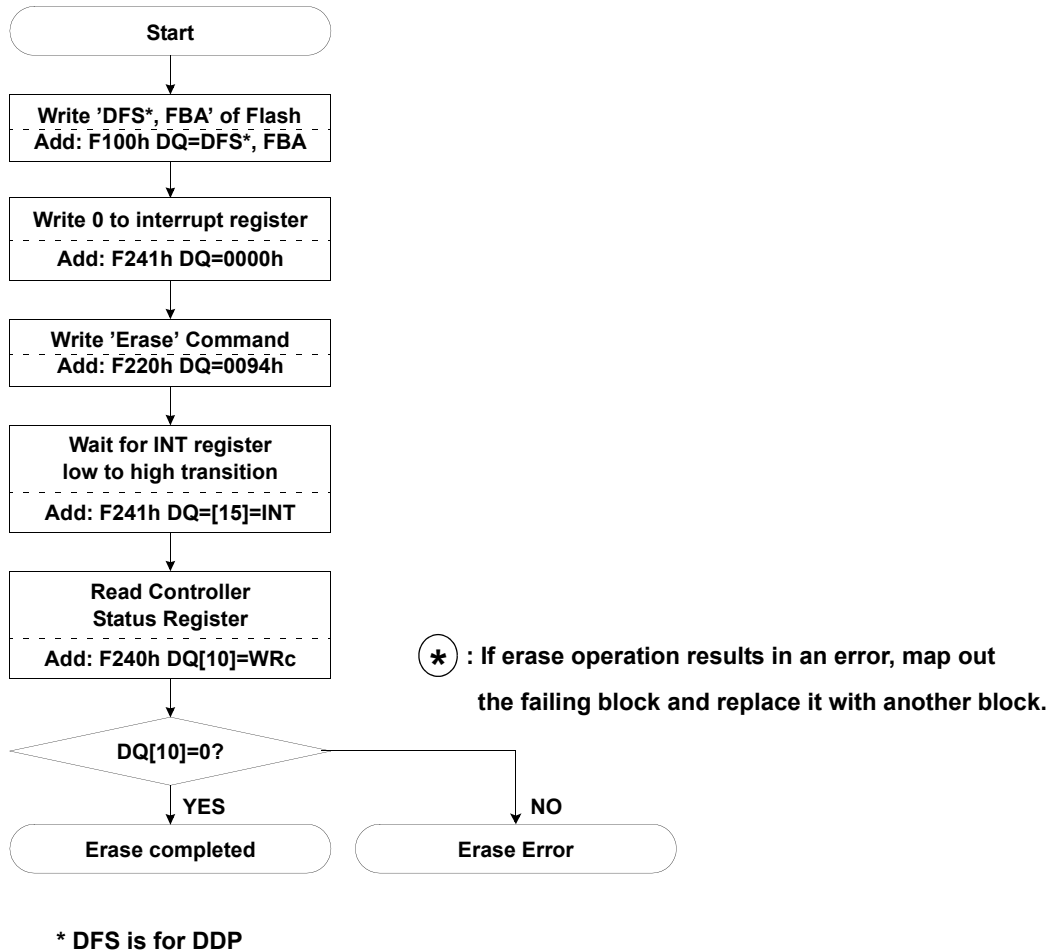


Figure 17. Erase operation flow-chart

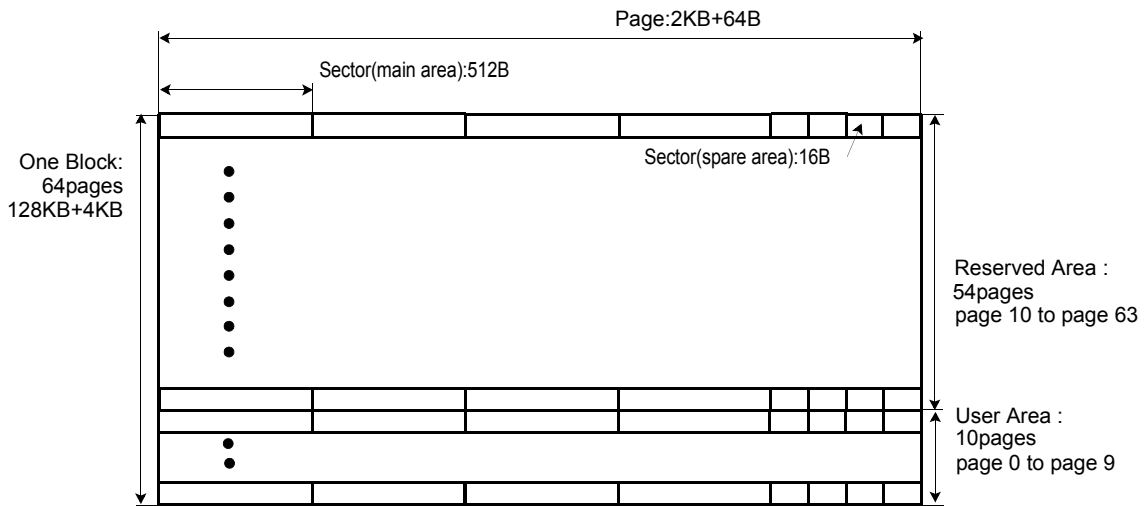


**OTP Operation**

The device supports one block sized OTP area, which can be read and programmed with the same sequence as normal operation. But this OTP block could not be erased. This block is separated from NAND Flash Array, so it could be accessed by OTP Access command instead of FBA. If user wants to exit from OTP access mode, Cold, Warm and Hot Reset operation should be done. OTP area is one block size(128KB, 64pages) and is divided by two areas. The first area from page 0 to page 9, total 10pages, is assigned for user and the second area from page 10 to page 63, total 54pages, are occupied for the device manufacturer. The manufacturer area is programmed prior to shipping, so this area could not be used by user. This block is fully guaranteed to be a valid block.

**OTP Block Page Allocation Information**

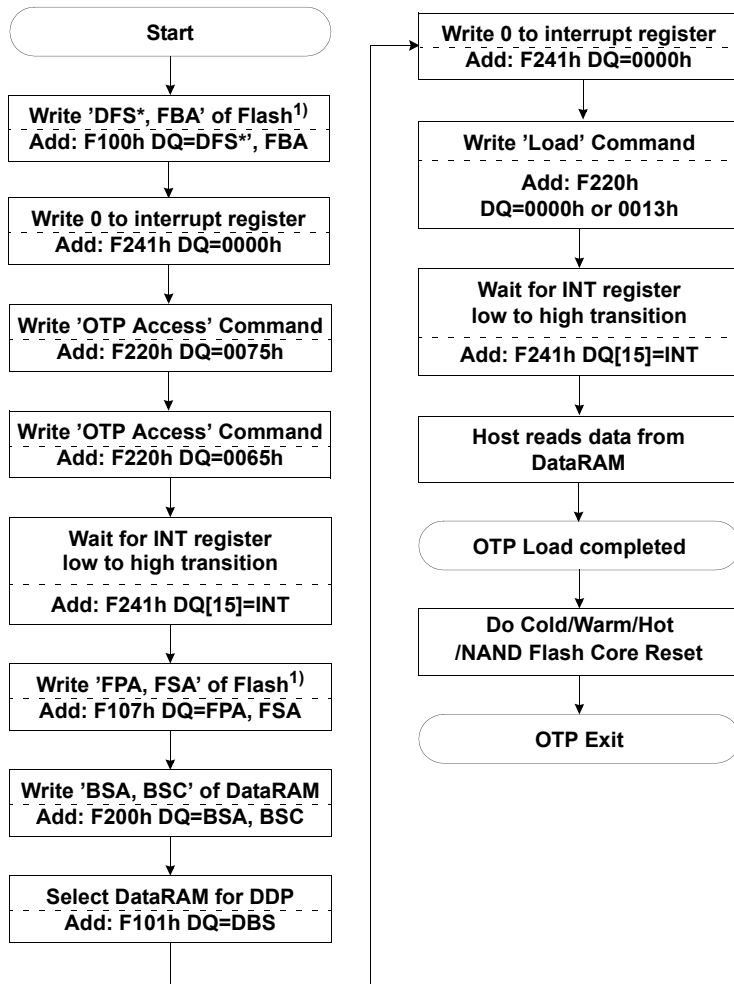
| Area         | Page               | Use                             |
|--------------|--------------------|---------------------------------|
| User         | 0 ~ 9 (10 pages)   | Designated as user area         |
| Manufacturer | 10 ~ 63 (54 pages) | Used by the device manufacturer |



**Figure 18. OTP area structure and assignment**

**OTP Load(OTP Access+Load NAND)**

OTP area is separated from NAND Flash Array, so it is accessed by OTP Access command instead of FBA. The content of OTP could be loaded with the same sequence as normal load operation after being accessed by the command. If user wants to exit from OTP access mode, Cold, Warm, Hot or NAND Flash Core Reset operation should be done.



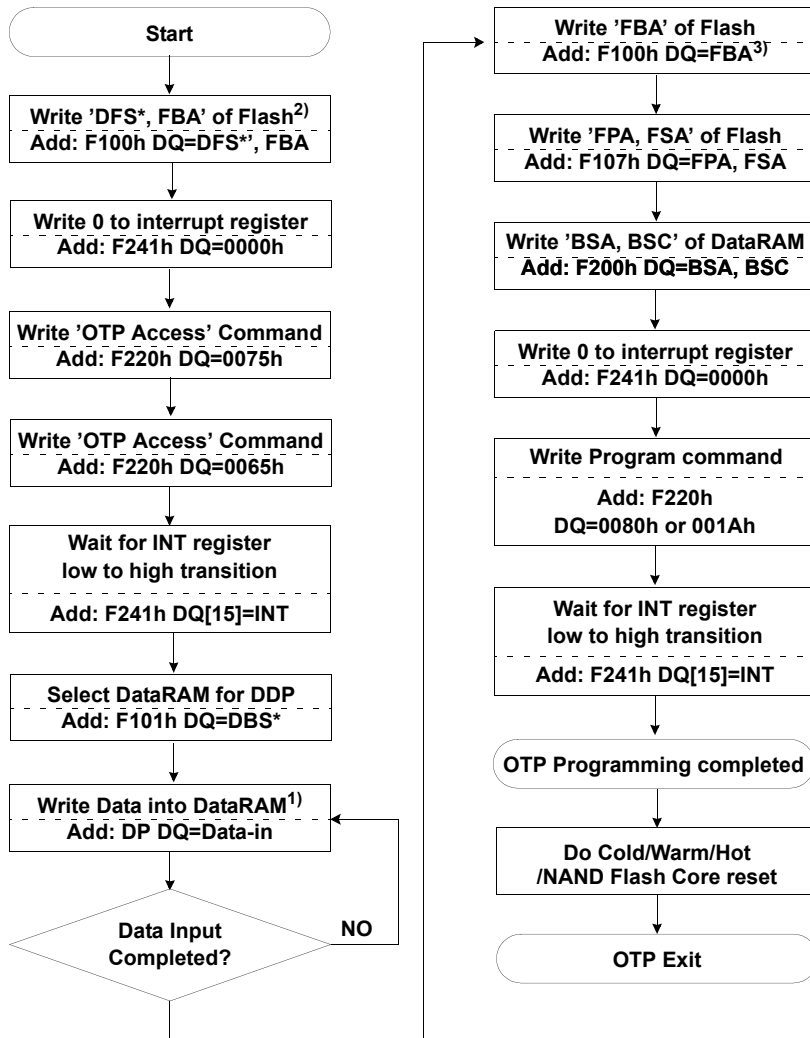
\* DBS, DFS is for DDP

Note 1) FBA(NAND Flash Block Address) could be omitted or any address.

Figure 19. OTP load operation flow-chart

OTP Programming(OTP Access+Program NAND)

OTP area could be programmed with the same sequence as normal program operation after being accessed by the command. But in case of OTP area program, OTP area is not a real OTP area but can be programmed more than once. And 2 command sequence is used to avoid the accidental write. To avoid the accidental write, FBA should point the unlocked area address among NAND Flash Array address map even though OTP area is separated from NAND Flash Array.



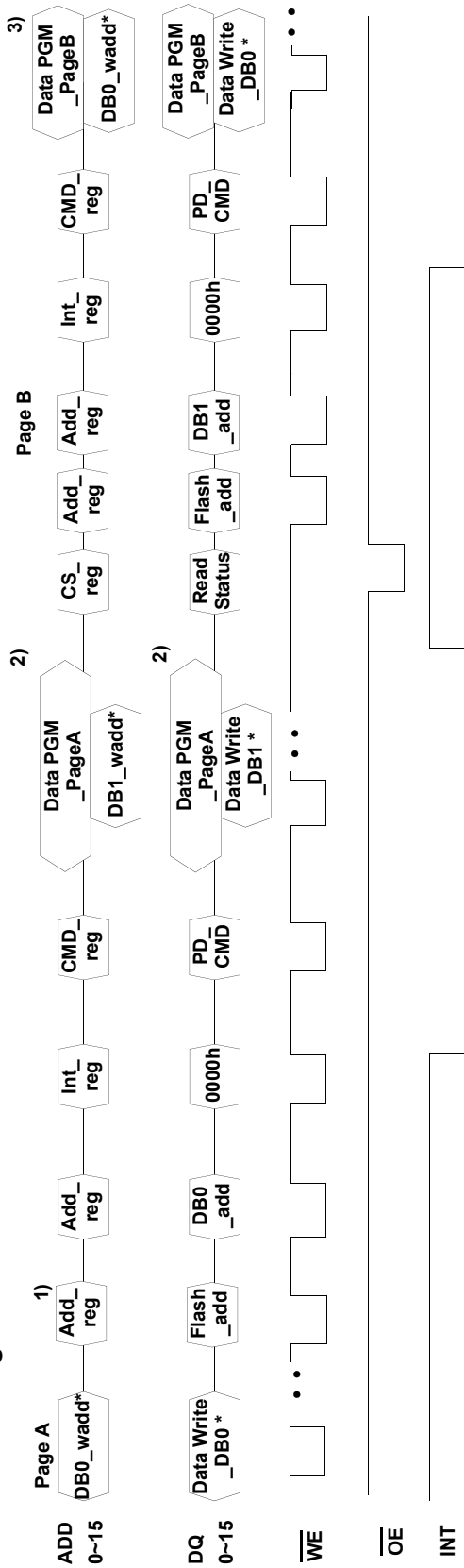
\* DBS, DFS is for DDP

- Note 1) FBA(NAND Flash Block Address) could be any address.
- 2) Data input could be done anywhere between "Start" and "Write Program Command".
- 3) FBA should point the unlocked area address among NADND Flash Array address map.

Figure 20. OTP program operation flow-chart



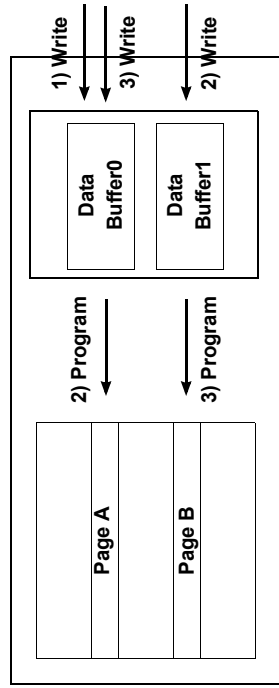
Write While Program



\* DBS should be set before accessing DataRAM for DDP

- Add\_reg : Address Register Address
- DBn\_wadd : DataRAM Address to be programmed
- DBn\_wadd : DataRAM Address to be written
- Data\_Write\_DBn : Write Data to DataRAMn
- Flash\_add : Flash Address to be programmed
- Int\_reg : Interrupt Register Address
- CMD\_reg : Command Register Address
- PD\_CMD : Program Command
- Data\_PGM\_PageA : Program Data from DataRAM to PageA
- CS\_reg : Controller Status Register Address

The device provides dual data buffer memory architecture. The device is capable of write operation and program operation simultaneously. This is so called the write while program operation with dual data buffer architecture, this feature provides the capability of executing write operation from host to one of data buffers during program operation from another data buffer to Flash simultaneously. Refer to the information for more details in "Program operation" before performing write while program operation. Simultaneous program and write operation to same data buffer is prohibited.



**ECC Operation**

While the device transfers data from BufferRAM to NAND Flash Array Page Buffer for Program Operation, the device hiddenly generates ECC(24bits for main area data and 10bits for 2nd and 3rd word data of each sector spare area) and while Load operation, hiddenly generates ECC and detects error number and position and corrects 1bit error. ECC is updated by the device automatically. After Load Operation, host can know whether there is error or not by reading 'ECC Status Register'(refer to ECC Status Register Table). In addition, OneNAND supports 2bit EDC even though it is little probable that 2bit error occurs. Hence, it is not recommended that Host reads 'ECC Status Register' for checking ECC error because the built-in Error Correction Logic of OneNAND finds out and corrects ECC error.

When the device loads NAND Flash Array main and spare area data with ECC operation, the device does not place the newly generated ECC for main and spare area into the buffer but places ECC which was generated and written in program operation into the buffer.

Ecc operation is done during the boot loading operation.

**ECC Bypass Operation**

ECC bypass operation is set by 9th bit of System Configuration 1 register. In ECC Bypass operation, the device neither generates ECC result which indicates error position nor updates ECC code to NAND Flash array spare area in program operation(refer to ECC Result Register Tables). During Load operation, the on-chip ECC engine does not generate a new ECC internally and the values of ECC Status and Result Registers are invalid. Hence, in ECC Bypass operation, the error cannot be detected and corrected by Mux-OneNAND itself. ECC Bypass operation is not recommended to host.

**Table 7. ECC Code & Result Status by ECC operation mode**

| 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 Error  |
| ECC operation | Update   | Pre-written ECC code <sup>(1)</sup> loaded | Update                                  | Correct     |
| ECC bypass    | Not update                                     | Pre-written code loaded                    | Invalid                                 | Not correct |

**NOTE:**

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

Data Protection during Power Down

The device is designed to offer protection from any involuntary program/erase during power-transitions. An internal voltage detector disables all functions whenever Vcc is below about 1.3V. RP pin provides hardware protection and is recommended to be kept at VIL before power-down.

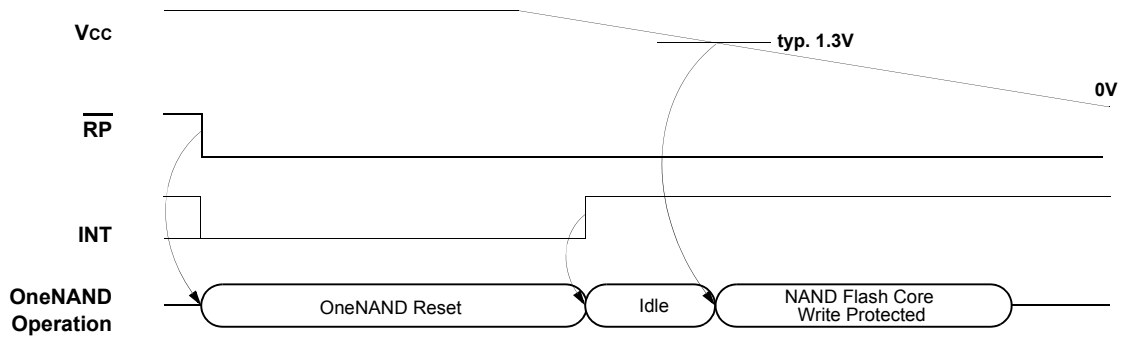


Figure 21. Data Protection during Power Down

## Technical Notes

### Invalid Block(s)

Invalid blocks are defined as blocks that contain one or more invalid bits whose reliability is not guaranteed by Samsung. The information regarding the invalid block(s) is so called as 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 design must be able to mask out the invalid block(s) via address mapping. The 1st block, which is placed on 00h block address, is fully guaranteed to be a valid block.

### Identifying Invalid Block(s)

All device locations are erased(FFFFh) except locations where the invalid block(s) information is written prior to shipping. The invalid block(s) status is defined by the 1st word in the spare area. Samsung makes sure that either the 1st or 2nd page of every invalid block has non-FFFFh data at the 1st word of sector0 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. Therefore, the system must be able to recognize the invalid block(s) based on the original invalid block information and create the invalid block table via the following suggested flow chart. Any intentional erasure of the original invalid block information is prohibited.

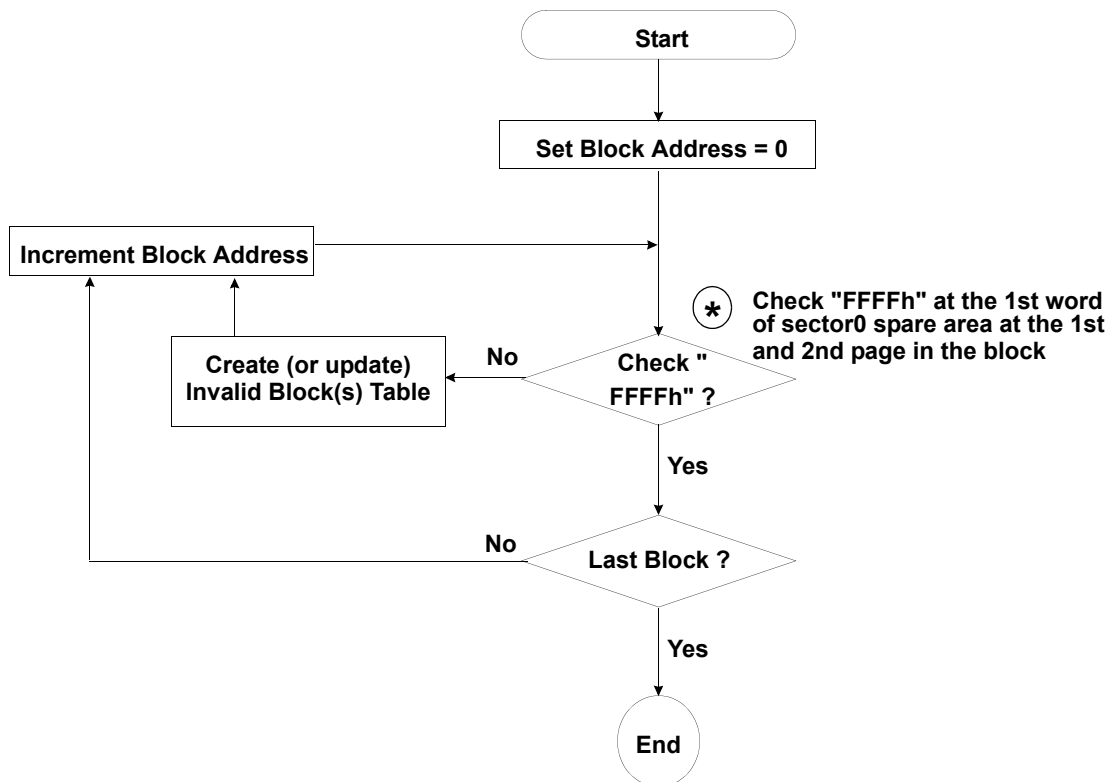


Figure 22. Flow chart to create invalid block table.



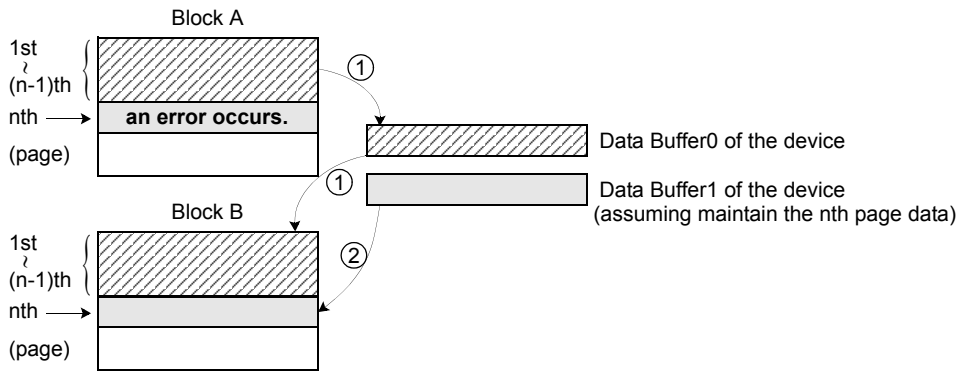
**Technical Notes** (Continued)

**Error in write or load operation**

Within its life time, additional invalid blocks may develop with the device. Refer to the qualification report for the actual data. The following possible failure modes should be considered to implement a highly reliable system. In the case of status read failure after erase or program, block replacement should be done. Because program status fail during a page program does not affect the data of the other pages in the same block, block replacement can be executed with a page-sized buffer by finding an erased empty block and reprogramming the current target data and copying the rest of the replaced block.

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

**Block Replacement**



When an error happens in the nth page of the Block 'A' during program operation.

\* Step1

Then, copy the data in the 1st ~ (n-1)th page to the same location of the Block 'B' via data buffer0.

\* Step2

Copy the nth page data of the Block 'A' in the data buffer1 to the nth page of another free block. (Block 'B')

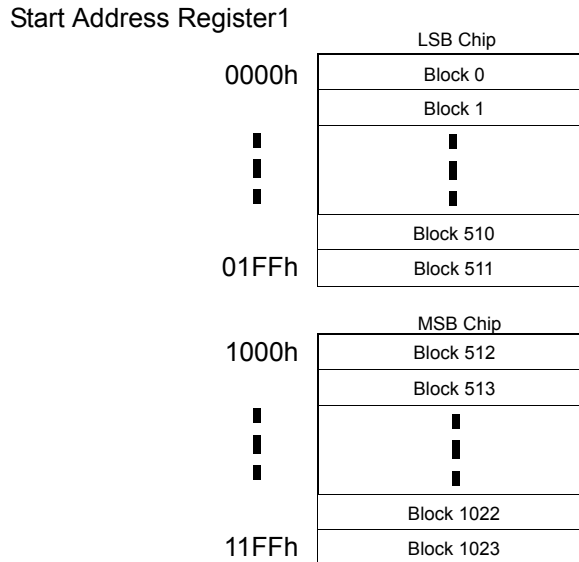
Do not further erase or program Block 'A' but create an 'invalid Block' table or other appropriate scheme.

## Technical Notes (Continued)

### OneNAND DDP Technical Note

#### DDP Chip Selection Register

OneNAND DDP configuration does not require additional pins. NAND Flash Block Address is consecutive between LSB and MSB chips. As seen in the figure below, the LSB Block Address ends at 01FFh(Block 511) and the MSB Block Address begins at 1000h(Block 512). The Device Flash Core Select (DFS) of Start Address 1 Register and the Device BufferRAM Select (DBS) of Start Address 2 Register are used to select the desired LSB or MSB Flash Core and BufferRAM in the DDP.



**Figure 23. Flash Block Address Map in DDP**

#### Start Address1 Register (R/W): F100h, default=0000h

This Read/Write register is used to select the Flash Core of the LSB or MSB device (DFS).

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

**DFS** (Device Flash Core Select): it selects Flash Core in two Flash Core of DDP

**FBA** (NAND Flash Block Address): NAND Flash block address which will be read or programmed or erased.

| Chip     | Start Address1 Register | Block Number         |
|----------|-------------------------|----------------------|
| LSB Chip | 0000h ~ 01FFh           | Block0 ~ Block511    |
| MSB Chip | 1000h ~ 11FFh           | Block512 ~ Block1023 |

#### Start Address2 Register (R/W): F101h, default=0000h

This Read/Write register is used to select the BufferRAM of the LSB or MSB device (DBS).

|     |                            |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|----------------------------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15  | 14                         | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBS | Reserved(0000000000000000) |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**DBS** (Device BufferRAM Select): it selects BufferRAM in two BufferRAM of DDP

## Technical Notes (Continued)

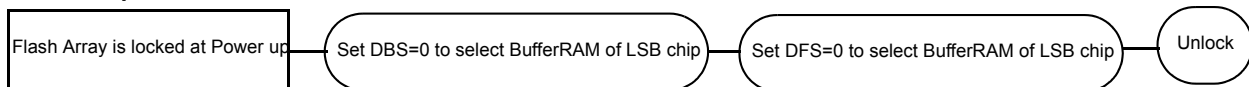
### DDP Chip Selection Operation

#### Flash Core Array Selection for Unlock Operation

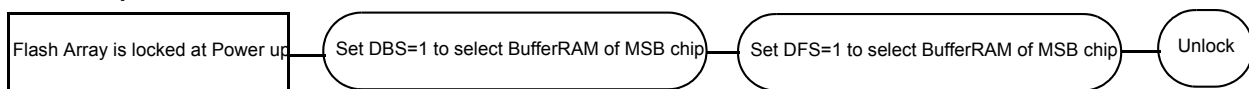
The LSB and MSB Flash Cores in a OneNAND DDP configuration power-up in a locked state and must be unlocked before operation. Set DFS and DBS = "0" to select the LSB Chip or "1" to select the MSB chip, then issue Unlock Command Sequence: Start block address+End block address+Unlock block command (0023h).

Note that the LSB and MSB chips must each be unlocked. The default state for Device BufferRAM Select (DBS) and Device Flash Core Select (DFS) = "0", which selects the LSB chip. For more information on lock operations.

#### . LSB Chip Case



#### . MSB Chip Case

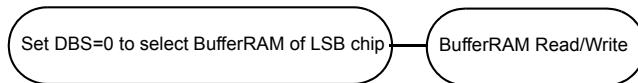


#### BufferRAM Selection for Read/Write Operation

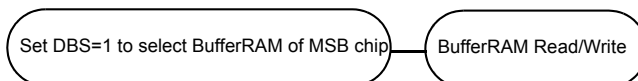
The LSB and MSB chip BufferRAMs operate independently. Select the desired LSB or MSB chip and then execute a BufferRAM Read/Write operation. The default state for Device BufferRAM Select (DBS) and Device Flash Core Select (DFS) = "0", which selects the LSB chip.

For more information on read/write operations.

#### . LSB Chip Case



#### . MSB Chip Case

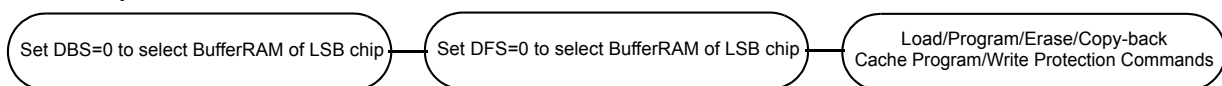


#### BufferRAM and Flash Core Selection for Load/Program/Erase/Copy-back/Cache Program/Write Protect Operation

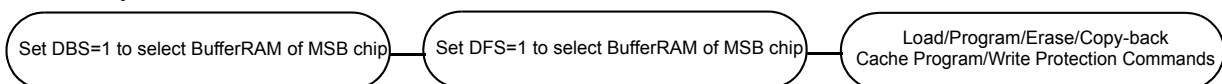
The LSB and MSB chip BufferRAMs operate independently. Select the desired LSB or MSB chip and then execute a BufferRAM Read/Write operation. The default state for Device BufferRAM Select (DBS) and Device Flash Core Select (DFS) = "0", which selects the LSB chip.

For more information on read/write operations.

#### . LSB Chip Case



#### . MSB Chip Case



**Technical Notes** (Continued)**Boot Sequence**

One of the best features OneNAND 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, OneNAND 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 NAND Flash Array 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 using dual buffering and taken by CPU to reduce CPU fetch time.

A typical boot scheme usually used to boot the system with OneNAND is explained at Figure 24 and Figure 25. 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.

**Boot Loaders in OneNAND**

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

NAND Flash Array of OneNAND is divided into the partitions as described at Figure 24 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 Figure 25.

**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 using dual buffering by BL1
3. BL2 is executed in DRAM  
OS image is loaded into DRAM through two DataRAMs using dual buffering by BL2
4. OS is running

Technical Notes (Continued)

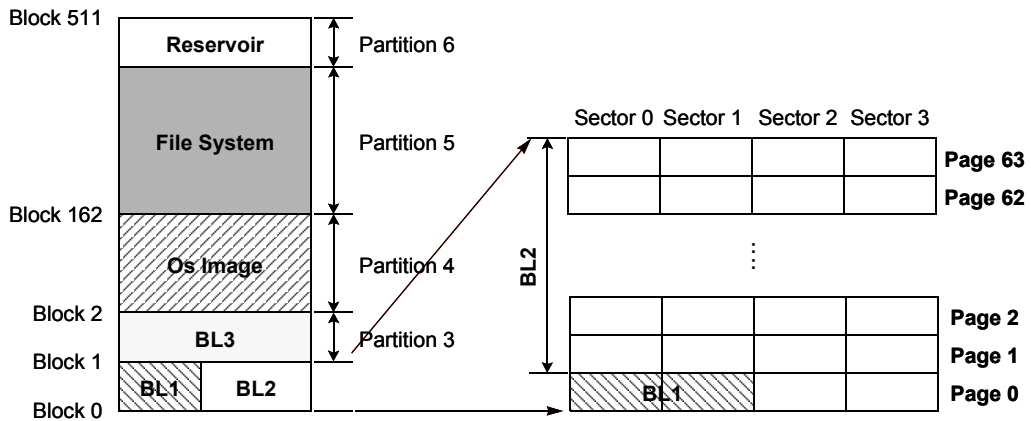
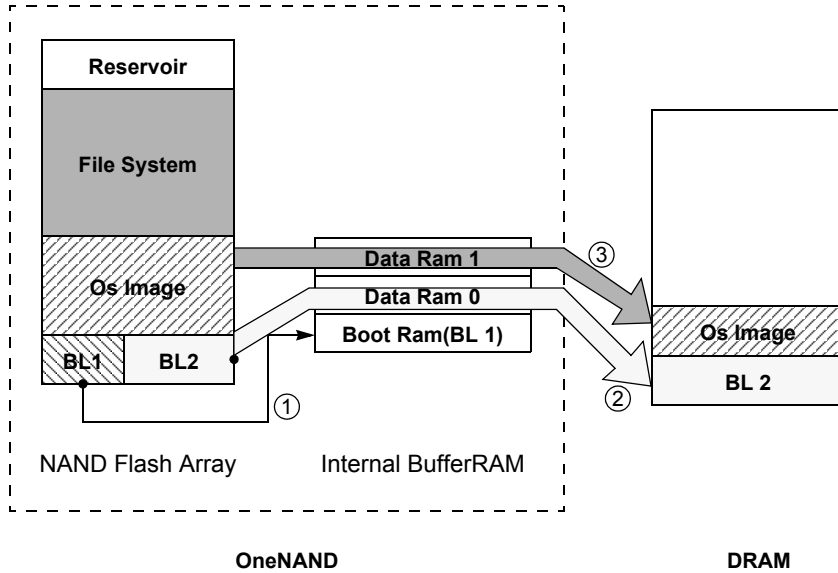


Figure 24. Partition of NAND Flash array



NOTE:  
 ② and ③ can be copied into DRAM through two DataRAMs using dual buffering

Figure 25. OneNAND Boot Sequence

**Technical Notes** (Continued)**Methods of Determining Interrupt Status**

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

The OneNAND 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.

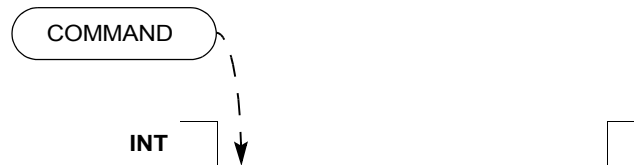
In its normal state, the INT pin is high if the INT polarity bit is default. Before a command is written to the command register, the INT bit must be written to '0' so the INT pin transitions to a low state indicating start of the operation. Upon completion of the command operation by the OneNAND's internal controller, INT returns to a high state.

INT is an open drain output allowing multiple INT outputs to be Or-tied together. INT does not float to a hi-Z condition when the chip is deselected or when outputs are 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.

**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.

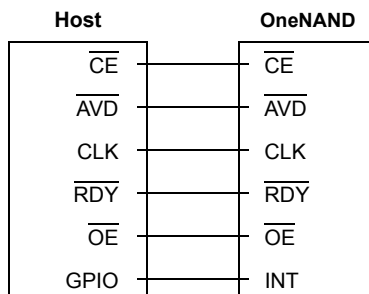


This can be configured to operate either synchronously or asynchronously as shown in the diagrams below.

## Technical Notes (Continued)

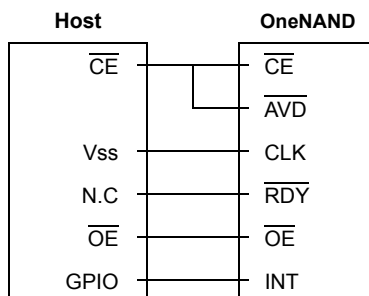
### Synchronous Mode Using the INT Pin

When operating synchronously, INT is tied directly to a Host GPIO.



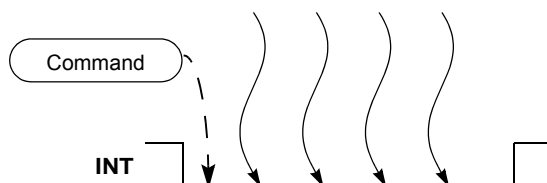
### Asynchronous Mode Using the INT Pin

When configured to operate in an asynchronous mode, /CE and /AVD of the OneNAND are tied to /CE of the Host. CLK is tied to the Host Vss (Ground). /RDY is tied to a no-connect. /OE of the OneNAND and Host are tied together and INT is tied to a GPIO.



### 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.

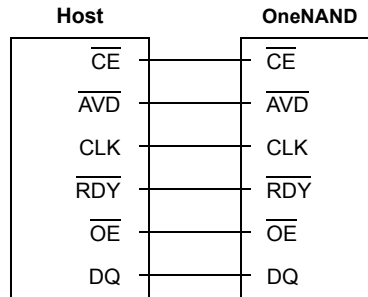


This can be configured in either a synchronous mode or an asynchronous mode.

## Technical Notes (Continued)

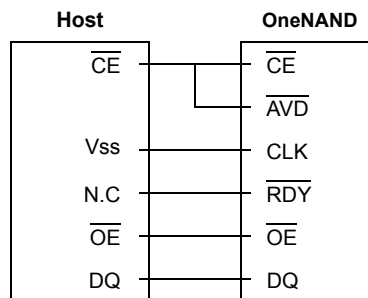
### Synchronous Mode Using Interrupt Status Register Bit Polling

When operating synchronously, /CE, /AVD, CLK, /RDY, /OE, and DQ pins on the host and OneNAND are tied together.



### Asynchronous Mode Using Interrupt Status Register Bit Polling

When configured to operate in an asynchronous mode, /CE and /AVD of the OneNAND are tied to /CE of the Host. CLK is tied to the Host Vss (Ground). /RDY is tied to a no-connect. /OE and DQ of the OneNAND and Host are tied together.



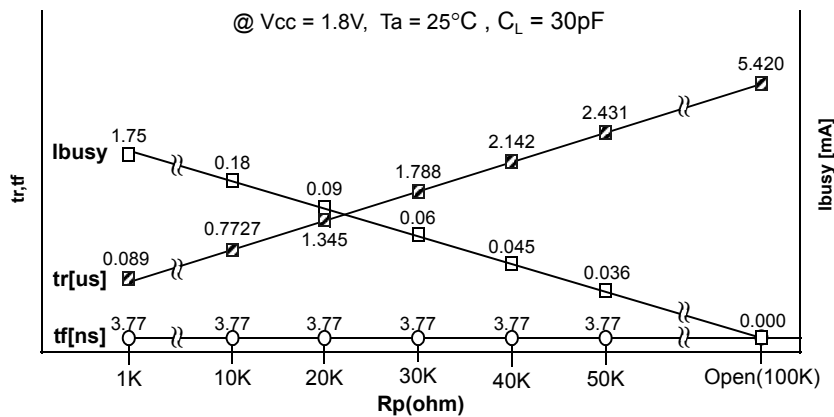
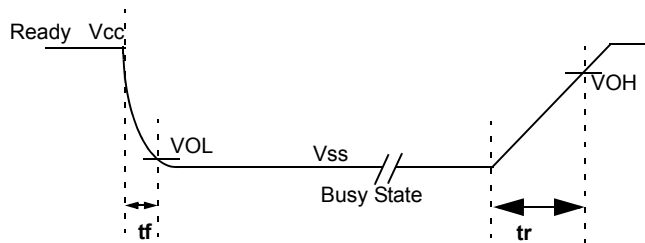
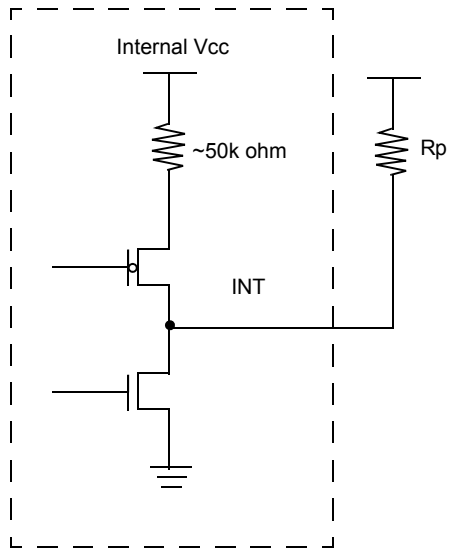


## Technical Notes (Continued)

### Determining Rp Value

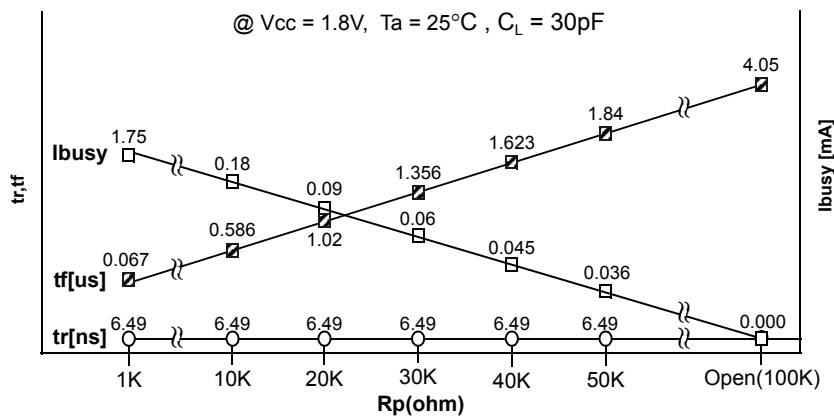
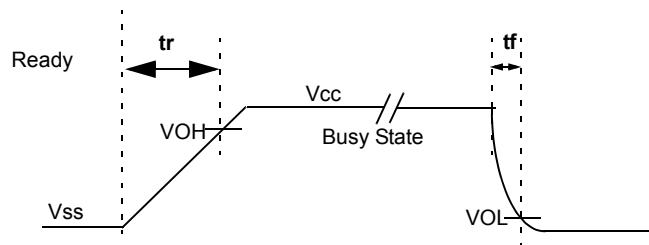
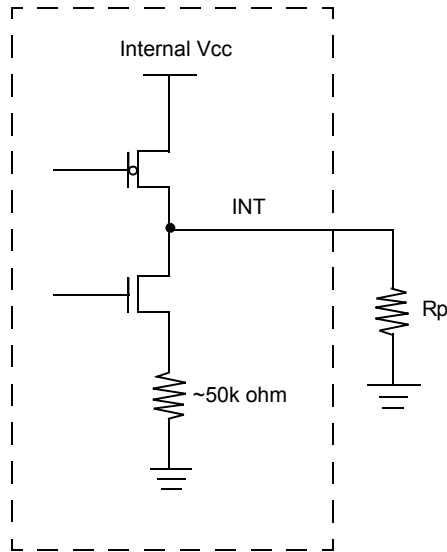
Because the pull-up resistor value is related to  $t_r(\text{INT})$ , an appropriate value can be obtained by the following reference charts.

### INT pol = 'High'



Technical Notes (Continued)

INT pol = 'Low'



## ABSOLUTE MAXIMUM RATINGS

| Parameter                          |            | Symbol                            | Rating                   |                          |                          | Unit |
|------------------------------------|------------|-----------------------------------|--------------------------|--------------------------|--------------------------|------|
|                                    |            |                                   | KFG1216Q2M<br>KFH1G16Q2M | KFG1216D2M<br>KFH1G16D2M | KFG1216U2M<br>KFH1G16U2M |      |
| Voltage on any pin relative to Vss | Vcc        | Vcc                               | -0.5 to + 2.45           | -0.6 to + 4.6            | -0.6 to + 4.6            | V    |
|                                    | All Pins   | VIN                               | -0.5 to + 2.45           | -0.6 to + 4.6            | -0.6 to + 4.6            |      |
| Temperature Under Bias             | Extended   | T <sub>bias</sub>                 | -30 to +125              | -30 to +125              | -30 to +125              | °C   |
|                                    | Industrial |                                   | -                        | -                        | -40 to +125              |      |
| Storage Temperature                |            | T <sub>stg</sub>                  | -65 to +150              | -65 to +150              | -65 to +150              | °C   |
| Short Circuit Output Current       |            | I <sub>os</sub>                   | 5                        | 5                        | 5                        | mA   |
| Operating Temperature              |            | T <sub>A</sub> (Extended Temp.)   | -30 to + 85              | -30 to + 85              | -                        | °C   |
|                                    |            | T <sub>A</sub> (Industrial Temp.) | -                        | -                        | -40 to + 85              |      |

## NOTES:

- Minimum DC voltage is -0.5V on Input/ Output pins. During transitions, this level should not fall to POR level(typ. 1.5V) .  
Maximum DC voltage is Vcc+0.6V on input / output pins which, during transitions, may overshoot to Vcc+2.0V for periods <20ns.
- 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.

## RECOMMENDED OPERATING CONDITIONS ( Voltage reference to GND )

| Parameter      | Symbol              | 1.8V Device |      |      | 2.65V Device |      |     | 3.3V Device |      |     | Unit |
|----------------|---------------------|-------------|------|------|--------------|------|-----|-------------|------|-----|------|
|                |                     | Min         | Typ. | Max  | Min          | Typ. | Max | Min         | Typ. | Max |      |
| Supply Voltage | Vcc-Core/<br>Vcc-IO | 1.7         | 1.8  | 1.95 | 2.4          | 2.65 | 2.9 | 2.7         | 3.3  | 3.6 | V    |
| Supply Voltage | Vss                 | 0           | 0    | 0    | 0            | 0    | 0   | 0           | 0    | 0   | V    |

## NOTES:

- The system power should reach 1.7V after POR triggering level(typ. 1.5V) within 400us.
- Vcc-Core should reach the operating voltage level prior to Vcc-IO.

## DC CHARACTERISTICS

| Parameter                                 | Symbol           | Test Conditions  | 1.8V device            |     |                        | 2.65V device           |     |                        | 3.3V device               |     |                            | Unit |    |
|---|------------------|--|------------------------|-----|------------------------|------------------------|-----|------------------------|---------------------------|-----|----------------------------|------|----|
|   |                  |  | Min                    | Typ | Max                    | Min                    | Typ | Max                    | Min                       | Typ | Max                        |      |    |
| Input Leakage Current                     | I <sub>LI</sub>  | V <sub>IN</sub> =V <sub>SS</sub> to V <sub>CC</sub> , V <sub>CC</sub> =V <sub>CCmax</sub>  | -1.0                   | -   | +1.0                   | -1.0                   | -   | +1.0                   | -1.0                      | -   | +1.0                       | μA   |    |
| Output Leakage Current                    | I <sub>LO</sub>  | V <sub>OUT</sub> =V <sub>SS</sub> to V <sub>CC</sub> , V <sub>CC</sub> =V <sub>CCmax</sub> , $\overline{CE}$ or $\overline{OE}$ =V <sub>IH</sub> (Note 1)    | -1.0                   | -   | +1.0                   | -1.0                   | -   | +1.0                   | -1.0                      | -   | +1.0                       | μA   |    |
| Active Asynchronous Read Current (Note 2) | I <sub>CC1</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub>  | -                      | 8   | 15                     | -                      | 10  | 20                     | -                         | 10  | 20                         | mA   |    |
| Active Burst Read Current (Note 2)        | I <sub>CC2</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub>  | 54MHz                  | -   | 12                     | 20                     | -   | 20                     | 30                        | -   | 20                         | 30   | mA |
|   |                  |  | 1MHz                   | -   | 3                      | 4                      | -   | 4                      | 6                         | -   | 4                          | 6    | mA |
| Active Write Current (Note 2)             | I <sub>CC3</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub>  | -                      | 8   | 15                     | -                      | 10  | 20                     | -                         | 10  | 20                         | mA   |    |
| Active Load Current (Note 3)              | I <sub>CC4</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub> , $\overline{WE}$ =V <sub>IH</sub> , V <sub>IN</sub> =V <sub>IH</sub> or V <sub>IL</sub> | -                      | 20  | 25                     | -                      | 25  | 30                     | -                         | 25  | 30                         | mA   |    |
| Active Program Current (Note 3)           | I <sub>CC5</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub> , $\overline{WE}$ =V <sub>IH</sub> , V <sub>IN</sub> =V <sub>IH</sub> or V <sub>IL</sub> | -                      | 20  | 25                     | -                      | 25  | 30                     | -                         | 25  | 30                         | mA   |    |
| Active Erase Current (Note3)              | I <sub>CC6</sub> | $\overline{CE}$ =V <sub>IL</sub> , $\overline{OE}$ =V <sub>IH</sub> , $\overline{WE}$ =V <sub>IH</sub> , V <sub>IN</sub> =V <sub>IH</sub> or V <sub>IL</sub> | -                      | 15  | 20                     | -                      | 20  | 25                     | -                         | 20  | 25                         | mA   |    |
| Standby Current                           | I <sub>SB</sub>  | $\overline{CE} = \overline{RP} = V_{CC} \pm 0.2V$  | Single                 | -   | 10                     | 50                     | -   | 20                     | 50                        | -   | 20                         | 50   | μA |
|   |                  |  | DDP                    | -   | 20                     | 100                    | -   | 40                     | 100                       | -   | 40                         | 100  |    |
| Input Low Voltage                         | V <sub>IL</sub>  | -  | -0.5                   | -   | 0.4                    | -0.5                   | -   | 0.4                    | 0                         | -   | 0.8                        | V    |    |
| Input High Voltage                        | V <sub>IH</sub>  | -  | V <sub>CCIO</sub> -0.4 | -   | V <sub>CCIO</sub> +0.4 | V <sub>CCIO</sub> -0.4 | -   | V <sub>CCIO</sub> +0.4 | 0.7*<br>V <sub>CCIO</sub> | -   | 0.7*<br>V <sub>CCIO</sub>  | V    |    |
| Output Low Voltage                        | V <sub>OL</sub>  | I <sub>OL</sub> = 100 μA, V <sub>CC</sub> =V <sub>CCmin</sub> , V <sub>CCq</sub> =V <sub>CCqmin</sub>  | -                      | -   | 0.2                    | -                      | -   | 0.2                    | -                         | -   | 0.22*<br>V <sub>CCIO</sub> | V    |    |
| Output High Voltage                       | V <sub>OH</sub>  | I <sub>OH</sub> = -100 μA, V <sub>CC</sub> =V <sub>CCmin</sub> , V <sub>CCq</sub> =V <sub>CCqmin</sub>   | V <sub>CCIO</sub> -0.1 | -   | -                      | V <sub>CCIO</sub> -0.4 | -   | -                      | 0.8*<br>V <sub>CCIO</sub> | -   | -                          | V    |    |

## NOTES:

- $\overline{CE}$  should be V<sub>IH</sub> for RDY. IOBE should be '0' for INT
- I<sub>CC</sub> active for Host access
- I<sub>CC</sub> active while Internal operation is in progress

## VALID BLOCK

| Parameter          | Symbol | Min | Typ. | Max | Unit   |
|--------------------|--------|-----|------|-----|--------|
| Valid Block Number | NvB    | 502 | -    | 512 | Blocks |

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 one or more bad bits. Do not erase or program factory-marked bad blocks.
2. The 1st block, which is placed on 00h block address, is fully guaranteed to be a valid block.

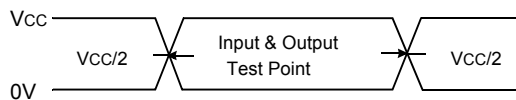
## CAPACITANCE (T<sub>A</sub> = 25 °C, V<sub>CC</sub> = 1.8V/2.65V/3.3V, f = 1.0MHz)

| Item                    | Symbol           | Test Condition       | Single |     | DDP |     | Unit |
|-------------------------|------------------|----------------------|--------|-----|-----|-----|------|
|                         |                  |                      | Min    | Max | Min | Max |      |
| Input Capacitance       | C <sub>IN1</sub> | V <sub>IN</sub> =0V  | -      | 10  | -   | 20  | pF   |
| Control Pin Capacitance | C <sub>IN2</sub> | V <sub>IN</sub> =0V  | -      | 10  | -   | 20  | pF   |
| Output Capacitance      | C <sub>OUT</sub> | V <sub>OUT</sub> =0V | -      | 10  | -   | 20  | pF   |

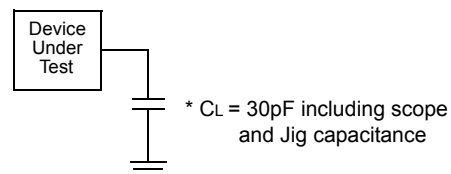
**NOTE** : Capacitance is periodically sampled and not 100% tested.

## AC TEST CONDITION (V<sub>CC</sub> = 1.8V/2.65V/3.3V)

| Parameter                      | Value                 |     |
|--------------------------------|-----------------------|-----|
| Input Pulse Levels             | 0V to V <sub>CC</sub> |     |
| Input Rise and Fall Times      | CLK                   | 3ns |
|                                | other inputs          | 5ns |
| Input and Output Timing Levels | V <sub>CC</sub> /2    |     |
| Output Load                    | C <sub>L</sub> = 30pF |     |



**Input Pulse and Test Point**



**Output Load**

## AC CHARACTERISTICS

### Synchronous Burst Read

| Parameter                                       | Symbol             | KFG1216X2M/KFH1G16X2M |      | Unit |
|---|--------------------|-----------------------|------|------|
|   |                    | Min                   | Max  |      |
| Clock   | CLK                | 1                     | 54   | MHz  |
| Clock Cycle                                     | tCLK               | 18.5                  | -    | ns   |
| Initial Access Time(at 54MHz)                   | tIAA               | -                     | 88.5 | ns   |
| Burst Access Time Valid Clock to Output Delay   | tBA                | -                     | 14.5 | ns   |
| $\overline{\text{AVD}}$ Setup Time to CLK       | tAVDS              | 7                     | -    | ns   |
| $\overline{\text{AVD}}$ Hold Time from CLK      | tAVDH              | 7                     | -    | ns   |
| Address Setup Time to CLK                       | tACS               | 7                     | -    | ns   |
| Address Hold Time from CLK                      | tACH               | 7                     | -    | ns   |
| Data Hold Time from Next Clock Cycle            | tBDH               | 4                     | -    | ns   |
| Output Enable to Data                           | tOE                | -                     | 20   | ns   |
| $\overline{\text{CE}}$ Disable to Output High Z | tCEZ <sup>1)</sup> | -                     | 20   | ns   |
| $\overline{\text{OE}}$ Disable to Output High Z | tOEZ <sup>1)</sup> | -                     | 17   | ns   |
| $\overline{\text{CE}}$ Setup Time to CLK        | tCES               | 7                     | -    | ns   |
| CLK High or Low Time                            | tCLKH/L            | tCLK/3                | -    | ns   |
| CLK <sup>2)</sup> to RDY valid                  | tRDYO              | -                     | 14.5 | ns   |
| CLK to RDY Setup Time                           | tRDYA              | -                     | 14.5 | ns   |
| RDY Setup Time to CLK                           | tRDYS              | 4                     | -    | ns   |
| $\overline{\text{CE}}$ low to RDY valid         | tCER               | -                     | 15   | ns   |

#### Note

1. If  $\overline{\text{OE}}$  is disabled before  $\overline{\text{CE}}$  is disabled, the output will go to high-z by tOEZ(max. 17ns).  
If  $\overline{\text{CE}}$  is disabled before  $\overline{\text{OE}}$  is disabled, the output will go to high-z by tCEZ(max. 20ns).  
If  $\overline{\text{CE}}$  and  $\overline{\text{OE}}$  are disabled at the same time, the output will go to high-z by tOEZ(max. 17ns).
2. It is the following clock of address fetch clock.

SWITCHING WAVEFORMS

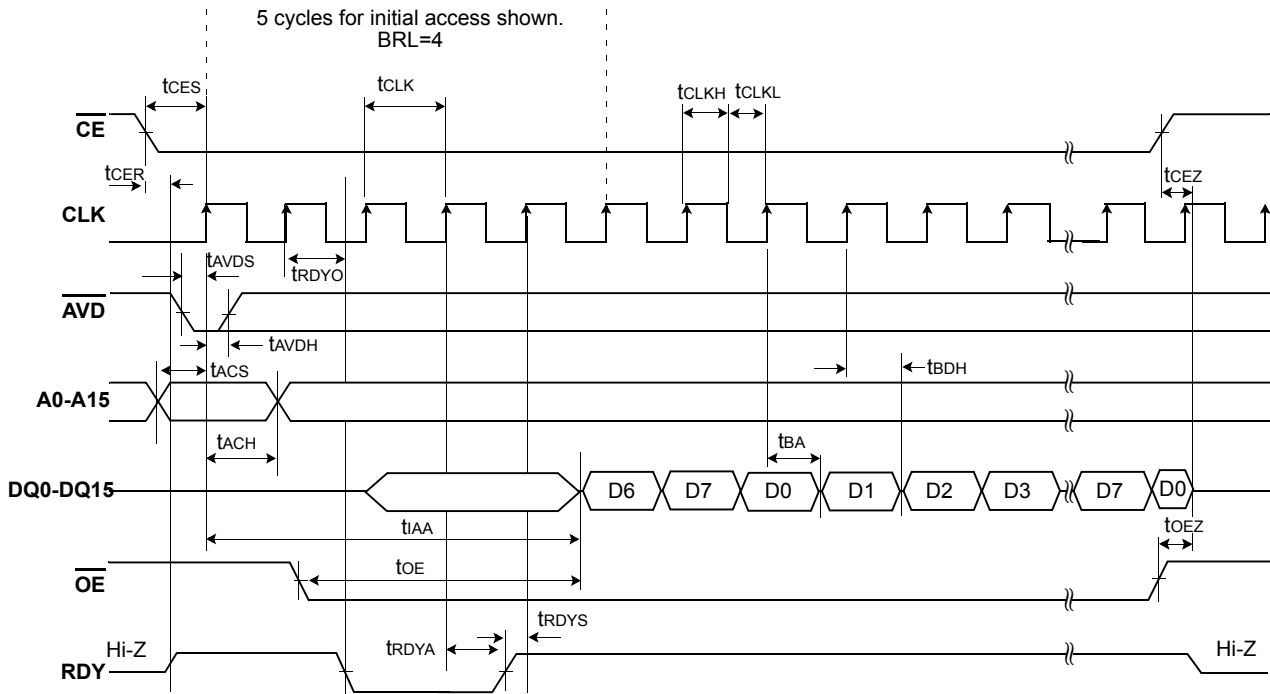


Figure 26. 8 Word Linear Burst Mode with Wrap Around

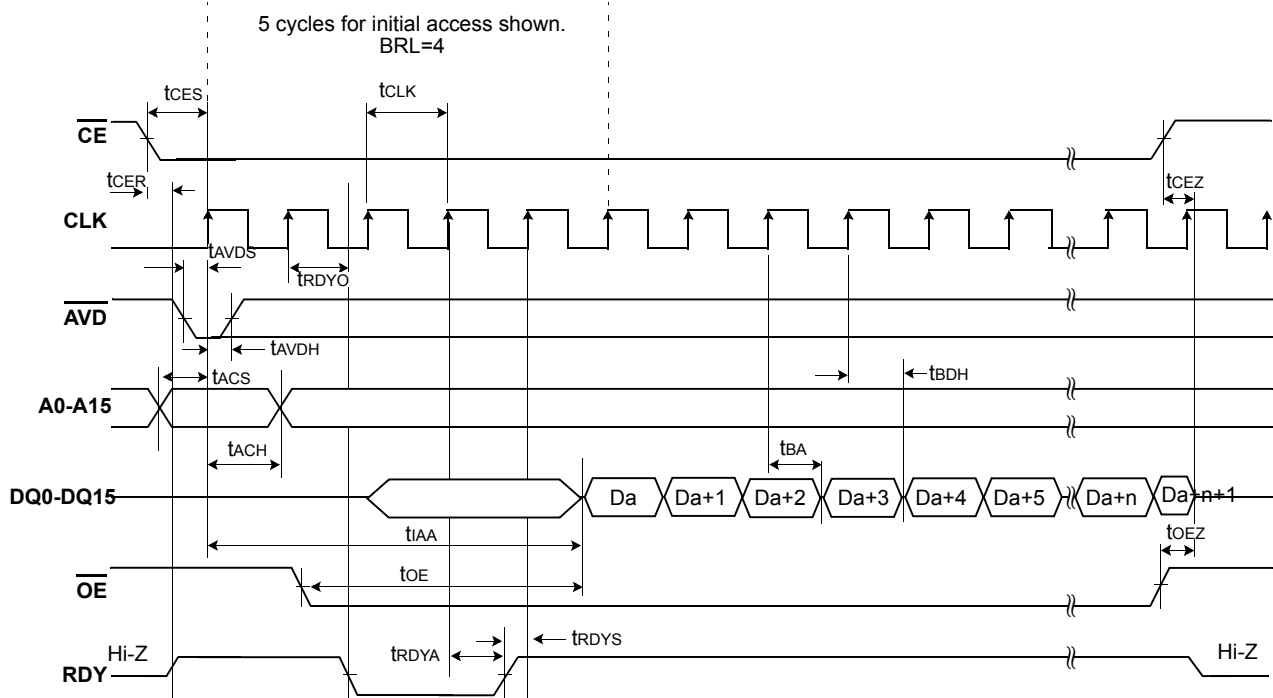


Figure 27. Continuous Linear Burst Mode with Wrap Around

NOTE: In order to avoid a bus conflict the  $\overline{OE}$  signal is enabled on the next rising edge after  $\overline{AVD}$  is going high.

**AC CHARACTERISTICS**  
**Asynchronous Read**

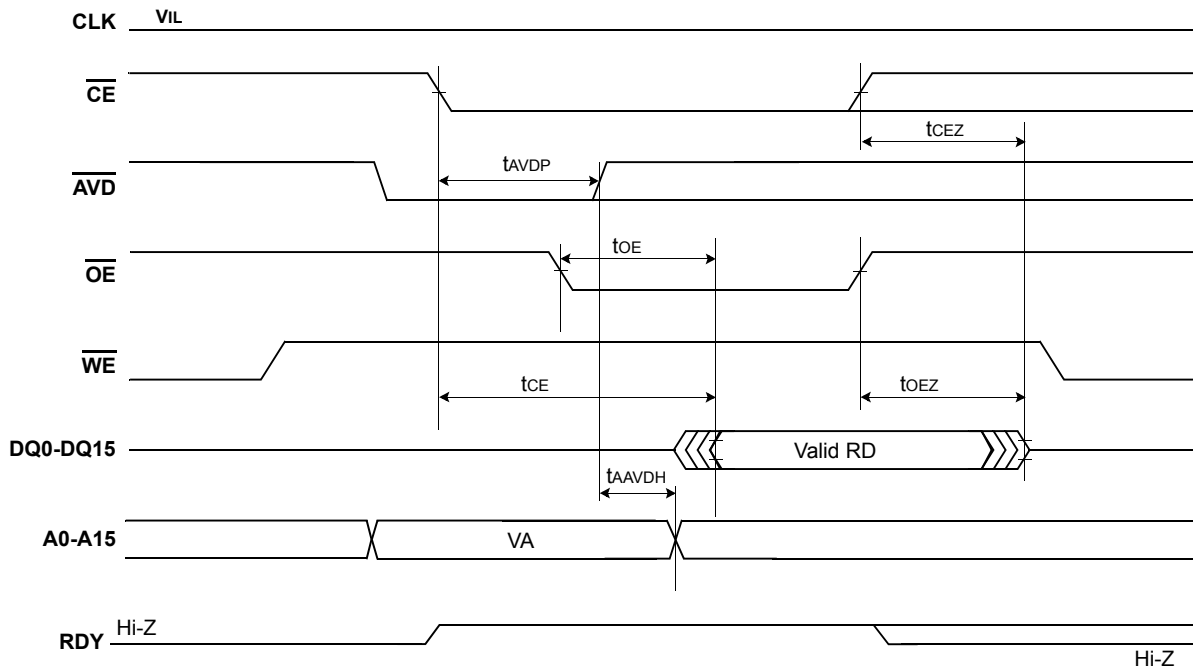
| Parameter  | Symbol | KFG1216X2M/KFH1G16X2M |     | Unit |
|--|--------|-----------------------|-----|------|
|  |        | Min                   | Max |      |
| Access Time from $\overline{CE}$ Low                         | tCE    | -                     | 76  | ns   |
| Asynchronous Access Time from $\overline{AVD}$ Low           | tAA    | -                     | 76  | ns   |
| Asynchronous Access Time from address valid                  | tACC   | -                     | 76  | ns   |
| Read Cycle Time  | tRC    | 76                    | -   | ns   |
| $\overline{AVD}$ Low Time                                    | tAVDP  | 12                    | -   | ns   |
| Address Setup to rising edge of $\overline{AVD}$             | tAAVDS | 7                     | -   | ns   |
| Address Hold from rising edge of $\overline{AVD}$            | tAAVDH | 7                     | -   | ns   |
| Output Enable to Output Valid                                | tOE    | -                     | 20  | ns   |
| $\overline{WE}$ Disable to $\overline{AVD}$ Enable           | tWEA   | 15                    | -   | ns   |
| $\overline{CE}$ Setup to $\overline{AVD}$ falling edge       | tCA    | 0                     | -   | ns   |
| $\overline{CE}$ Disable to Output & RDY High Z <sup>1)</sup> | tCEZ   | -                     | 20  | ns   |
| $\overline{OE}$ Disable to Output & RDY High Z <sup>1)</sup> | tOEZ   | -                     | 17  | ns   |

**NOTE:**

1. If  $\overline{OE}$  is disabled before  $\overline{CE}$  is disabled, the output will go to high-z by tOEZ(max. 17ns).  
 If  $\overline{CE}$  is disabled before  $\overline{OE}$  is disabled, the output will go to high-z by tCEZ(max. 20ns).  
 If  $\overline{CE}$  and  $\overline{OE}$  are disabled at the same time, the output will go to high-z by tOEZ(max. 17ns).  
 These parameters are not tested 100%.

**SWITCHING WAVEFORMS**

Case 1 : Valid Address and  $\overline{AVD}$  Transition occur before  $\overline{CE}$  is driven to Low



NOTE: VA=Valid Read Address, RD=Read Data.

**Figure 28 . Asynchronous Read Mode( $\overline{AVD}$  toggling)**



Case 2 :  $\overline{\text{AVD}}$  Transition occurs after  $\overline{\text{CE}}$  is driven to Low and Valid Address Transition occurs before  $\overline{\text{AVD}}$  is driven to Low

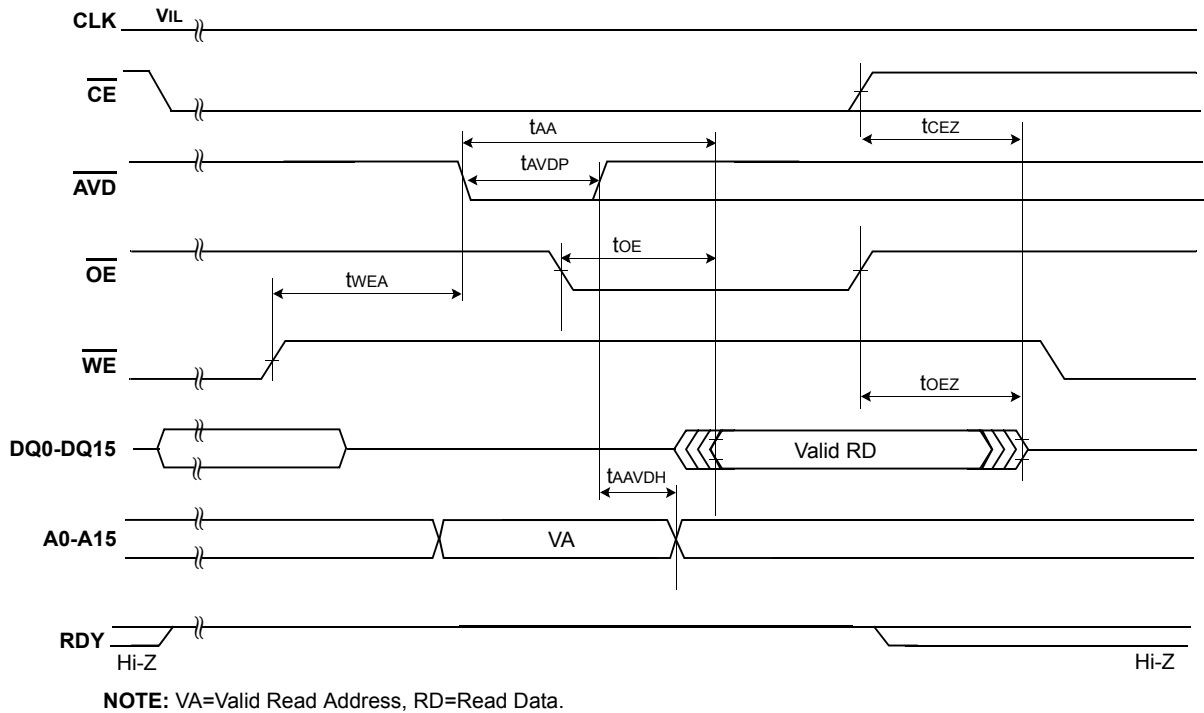


Figure 29. Asynchronous Read Mode( $\overline{\text{AVD}}$  toggling)

Case 3 :  $\overline{\text{AVD}}$  Transition occur after  $\overline{\text{CE}}$  is driven to Low and Valid Address Transition occurs after  $\overline{\text{AVD}}$  is driven to Low

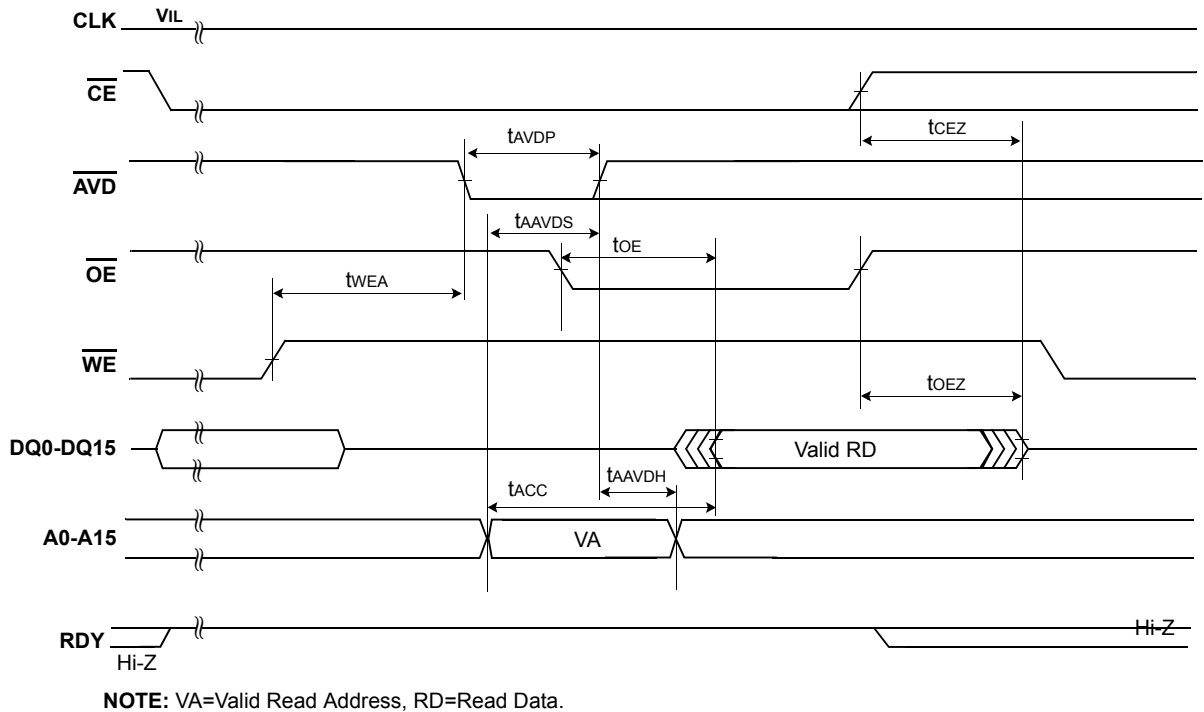
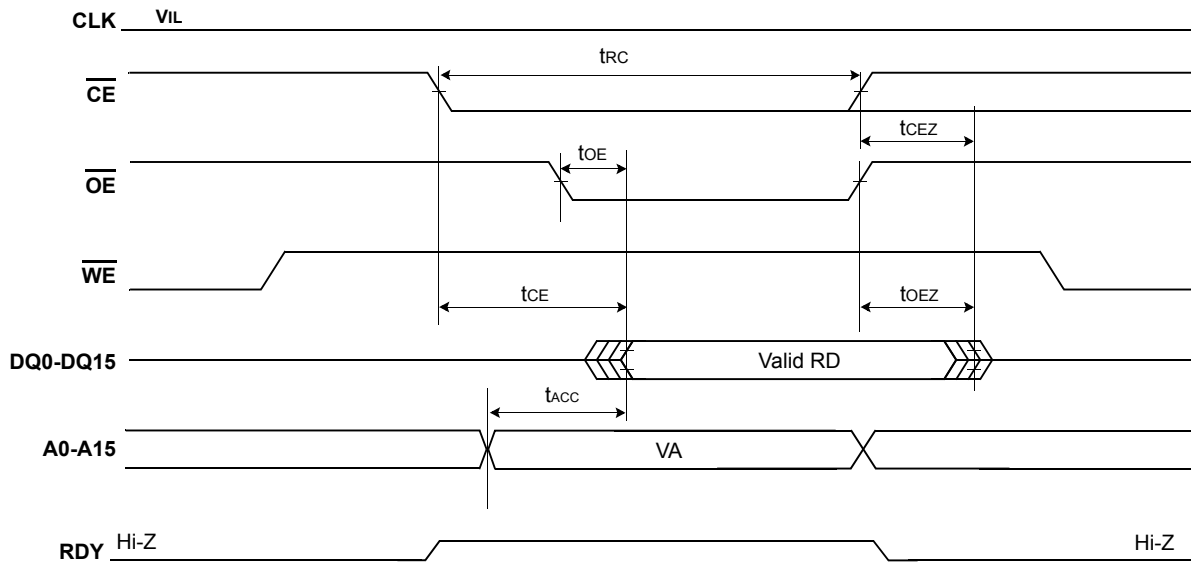


Figure 30. Asynchronous Read Mode( $\overline{\text{AVD}}$  toggling)

Case 4 :  $\overline{\text{AVD}}$  is tied to  $\overline{\text{CE}}$



NOTE: VA=Valid Read Address, RD=Read Data.

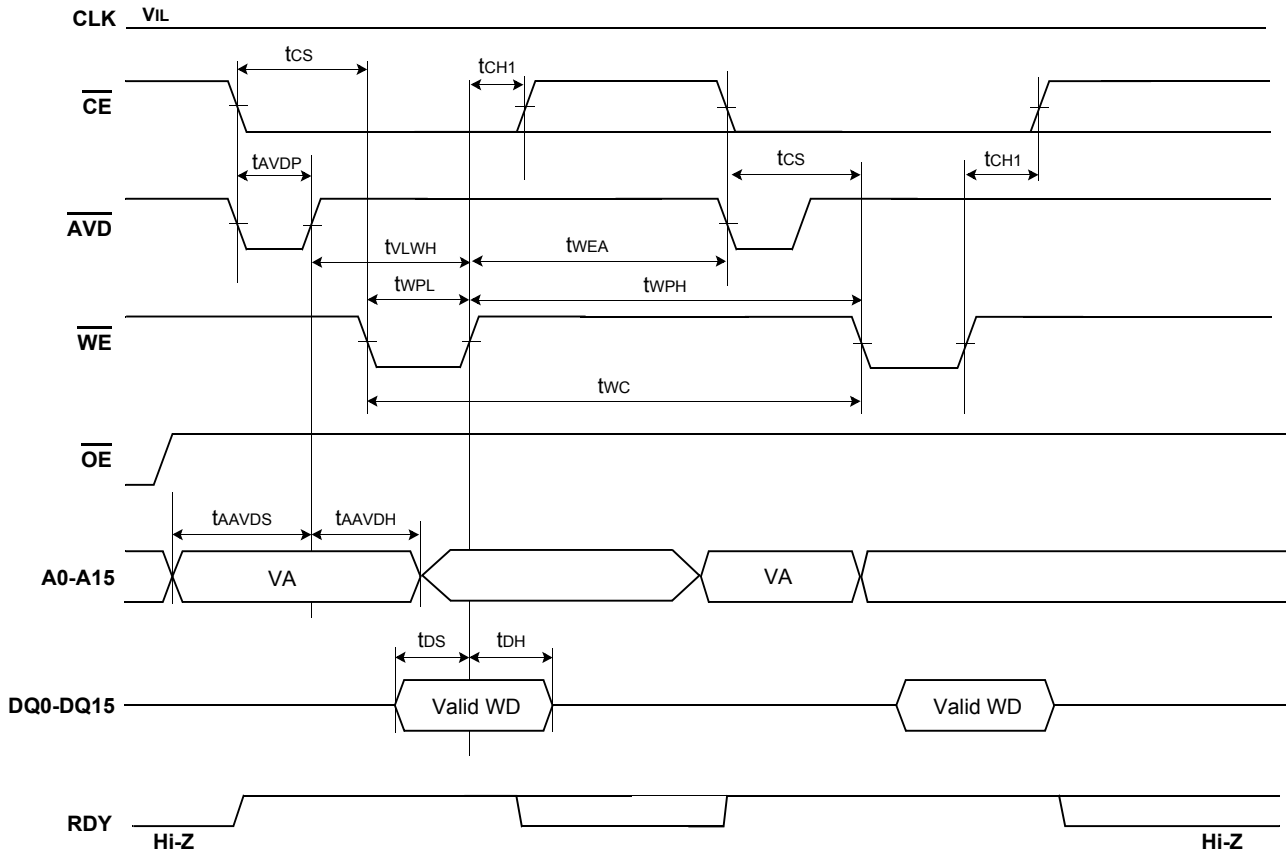
Figure 31. Asynchronous Read Mode( $\overline{\text{AVD}}$  tied to  $\overline{\text{CE}}$ )

## AC CHARACTERISTICS

## Asynchronous write operation

| Parameter  | Symbol | KFG1216X2M/KFH1G16X2M |     |     | Unit |
|--|--------|-----------------------|-----|-----|------|
|  |        | Min                   | Typ | Max |      |
| $\overline{WE}$ Cycle Time                               | tWC    | 70                    | -   | -   | ns   |
| $\overline{AVD}$ low pulse width                         | tAVDP  | 12                    | -   | -   | ns   |
| Address Setup to rising edge of $\overline{AVD}$         | tAAVDS | 7                     | -   | -   | ns   |
| Address Setup to falling edge of $\overline{WE}$         | tAWES  | 0                     |     |     |      |
| Address Hold to rising edge of $\overline{AVD}$          | tAAVDH | 7                     | -   | -   | ns   |
| Address Hold to falling edge of $\overline{WE}$          | tAH    | 10                    |     |     | ns   |
| Data Setup to rising edge of $\overline{WE}$             | tDS    | 10                    | -   | -   | ns   |
| Data Hold from rising edge of $\overline{WE}$            | tDH    | 4                     | -   | -   | ns   |
| $\overline{CE}$ Setup to falling edge of $\overline{WE}$ | tCS    | 0                     | -   | -   | ns   |
| $\overline{CE}$ Hold from rising edge of $\overline{WE}$ | tCH1   | 0                     | -   | -   | ns   |
| $\overline{CE}$ Hold from rising edge of $\overline{WE}$ |        |                       |     |     |      |
| $\overline{CE}$ Hold from rising edge of $\overline{WE}$ | tCH2   | 10                    | -   | -   | ns   |
| $\overline{WE}$ Pulse Width                              | tWPL   | 40                    | -   | -   | ns   |
| $\overline{WE}$ Pulse Width High                         | tWPH   | 30                    | -   | -   | ns   |
| $\overline{AVD}$ Disable to $\overline{WE}$ Disable      | tVLWH  | 15                    | -   | -   | ns   |
| $\overline{WE}$ Disable to $\overline{AVD}$ Enable       | tWEA   | 15                    | -   | -   | ns   |

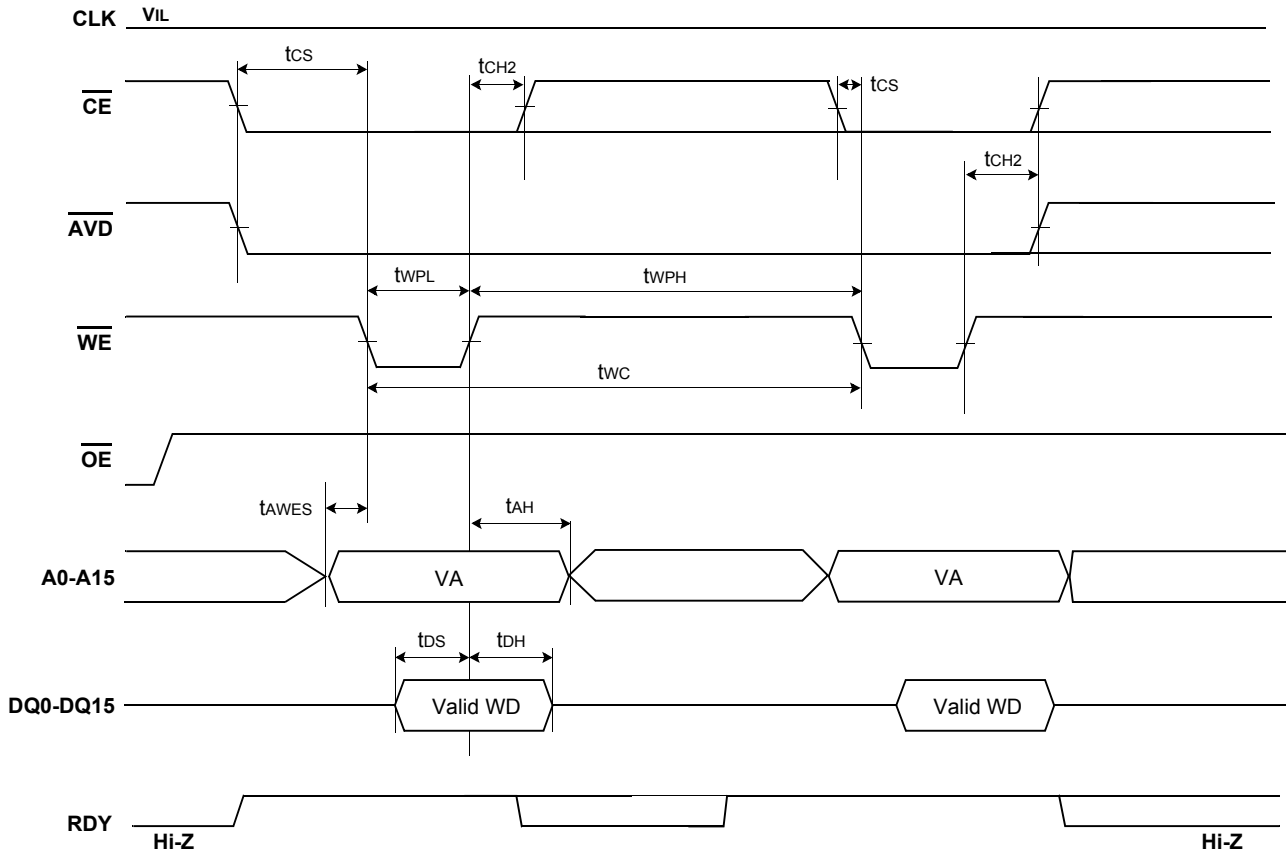
Case 1 :  $\overline{\text{AVD}}$  is toggled every write cycle



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

Figure 32. Latched Asynchronous Write Mode( $\overline{\text{AVD}}$  toggling)

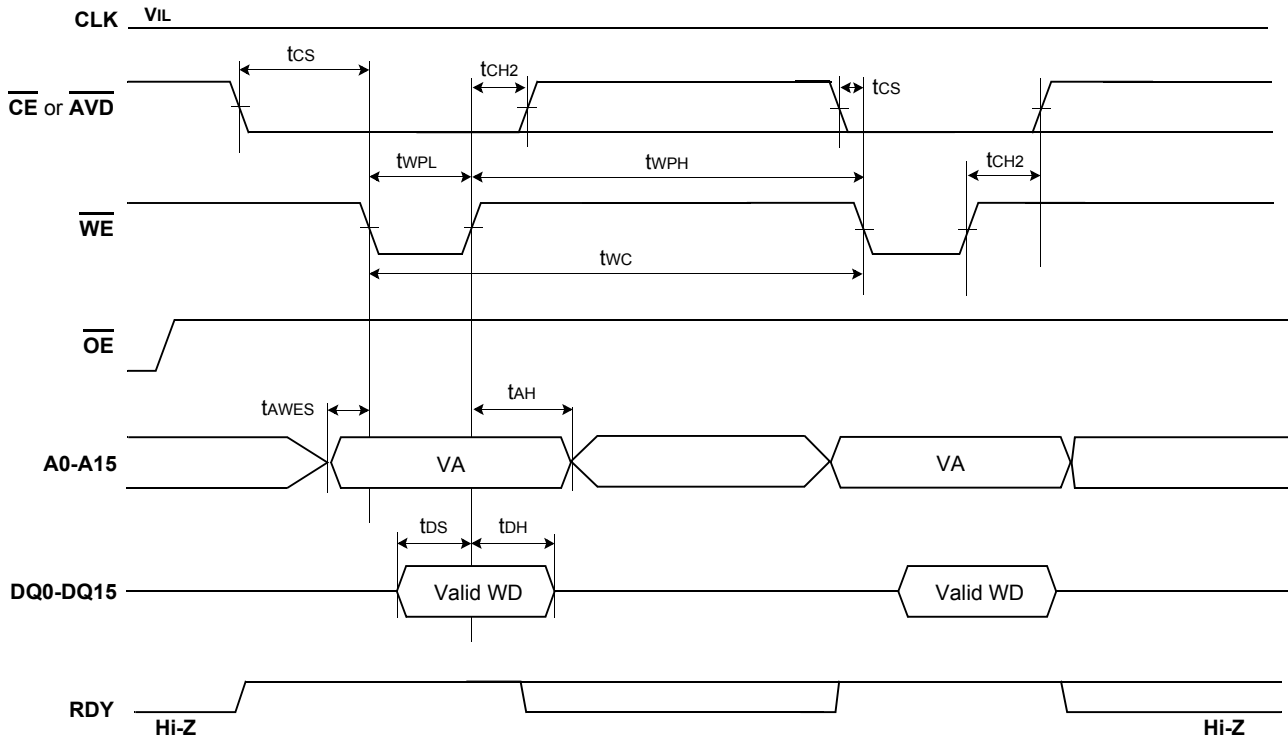
Case 2 :  $\overline{\text{AVD}}$  is synchronized with  $\overline{\text{CE}}$



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

Figure 33. Asynchronous Write Mode( $\overline{\text{AVD}}$  toggling)

Case 3 :  $\overline{\text{AVD}}$  is tied to  $\overline{\text{CE}}$



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

Figure 34. Asynchronous Write Mode( $\overline{\text{AVD}}$  tied to  $\overline{\text{CE}}$ )

AC CHARACTERISTICS

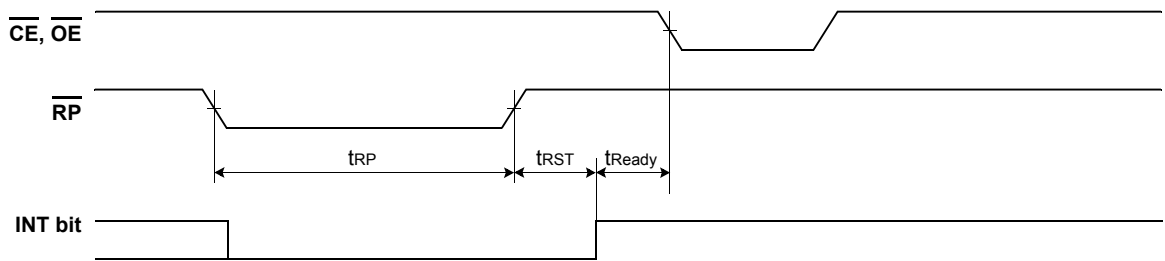
Reset

| Parameter   | Symbol | KFG1216U2M/<br>KFH1G16X2M |     | Unit    |
|---|--------|---------------------------|-----|---------|
|   |        | Min                       | Max |         |
| $\overline{RP}$ & Reset Command Latch(During Load Routines) to INT High (Note)          | tRST   | -                         | 10  | $\mu$ s |
| $\overline{RP}$ & Reset Command Latch(During Program Routines) to INT High (Note)       | tRST   | -                         | 20  | $\mu$ s |
| $\overline{RP}$ & Reset Command Latch(During Erase Routines) to INT High (Note)         | tRST   | -                         | 500 | $\mu$ s |
| $\overline{RP}$ & Reset Command Latch(NOT During Internal Routines) to Read Mode (Note) | tRST   | -                         | 10  | $\mu$ s |
| INT High to Read Mode (Note)  | tReady | 200                       | -   | ns      |
| $\overline{RP}$ Pulse Width   | tRP    | 200                       | -   | ns      |

NOTE: 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. Please refer to page 73 and 74.

SWITCHING WAVEFORMS

Warm Reset



Hot Reset

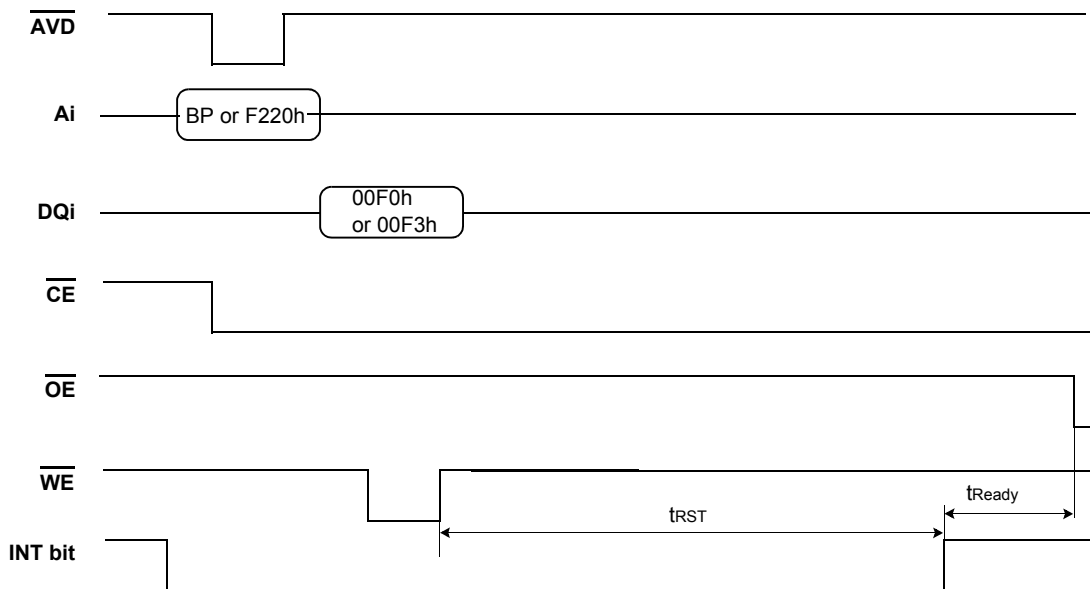


Figure 35. Reset Timing

**Performance**

| Parameter   | Symbol | Min | Typ | Max  | Unit   |
|---|--------|-----|-----|------|--------|
| Sector Load time (Note 1)   | tRD1   | -   | 40  | 45   | μs     |
| Page Load time (Note 1)   | tRD2   | -   | 85  | 100  | μs     |
| Sector Program time (Note 1)  | tPGM1  | -   | 320 | 720  | μs     |
| Page Program time (Note 1)  | tPGM2  | -   | 350 | 750  | μs     |
| OTP Access time(Note 1)   | tOTP   | -   | 600 | 1000 | ns     |
| Lock/Unlock/Lock-tight time (Note 1)  | tLOCK  | -   | 600 | 1000 | ns     |
| Number of Partial Program Cycles in the sector<br>(Including main and spare area) | NOP    | -   | -   | 2    | cycles |
| Block Erase time (Note 1)   | tBERS1 | -   | 2   | 3    | ms     |

**NOTES:** 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. Please refer to page 73 and 74.



SWITCHING WAVEFORMS

Load Operations

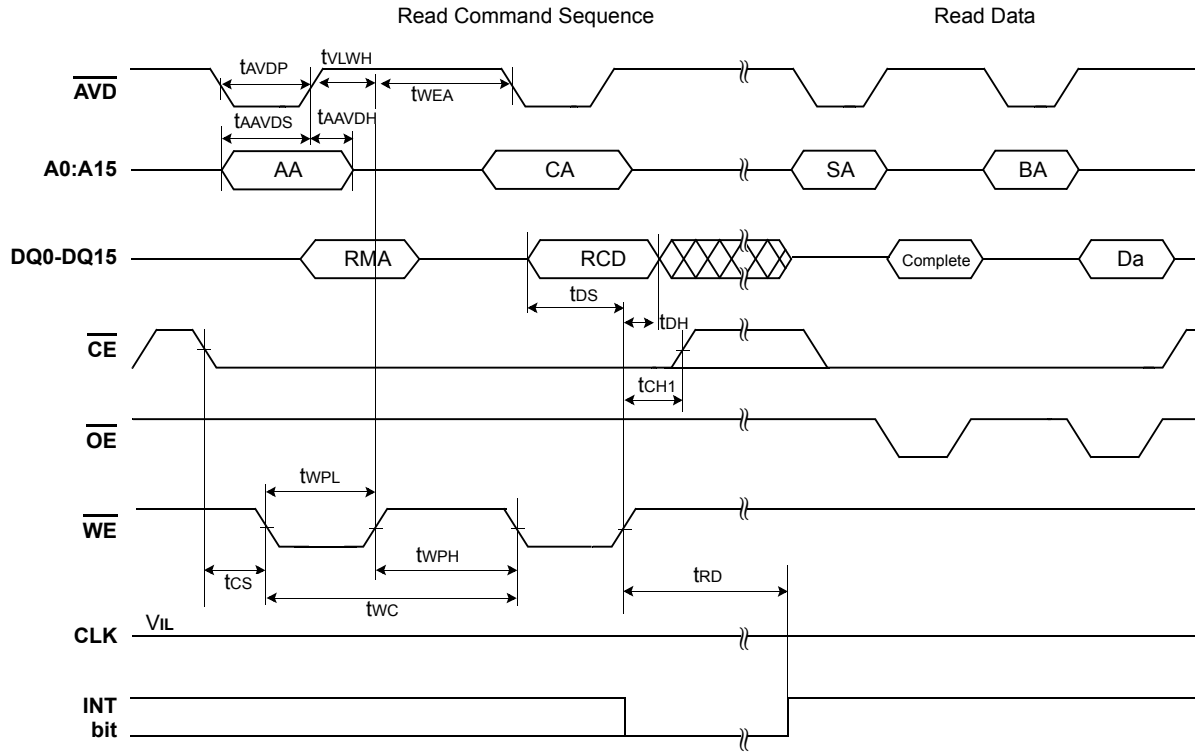


Figure 36. Load Operation Timing

NOTES:

1. AA = Address of address register  
 CA = Address of command register  
 RCD = Read Command  
 RMA = Address of memory to be read  
 BA = Address of BufferRAM to load 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.

SWITCHING WAVEFORMS

Program Operations

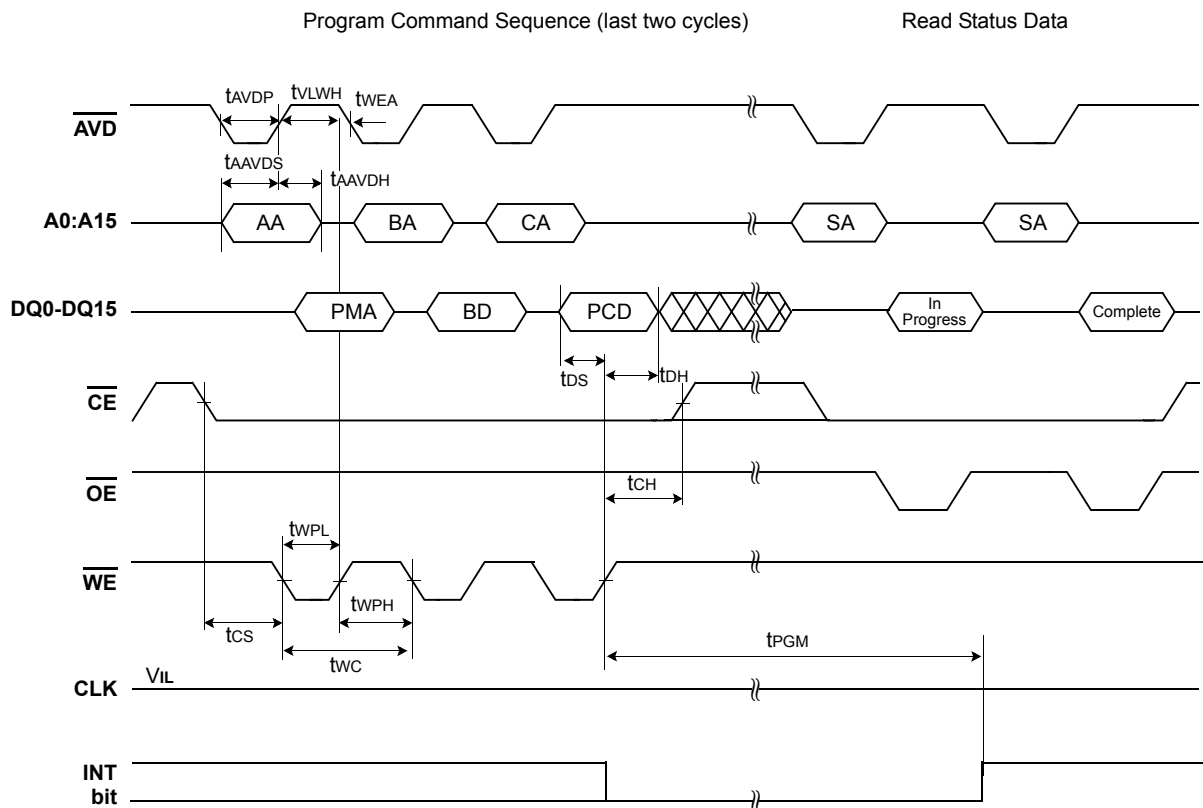


Figure 37 . Program Operation Timing

NOTES:

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 load 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.

SWITCHING WAVEFORMS

Erase Operation

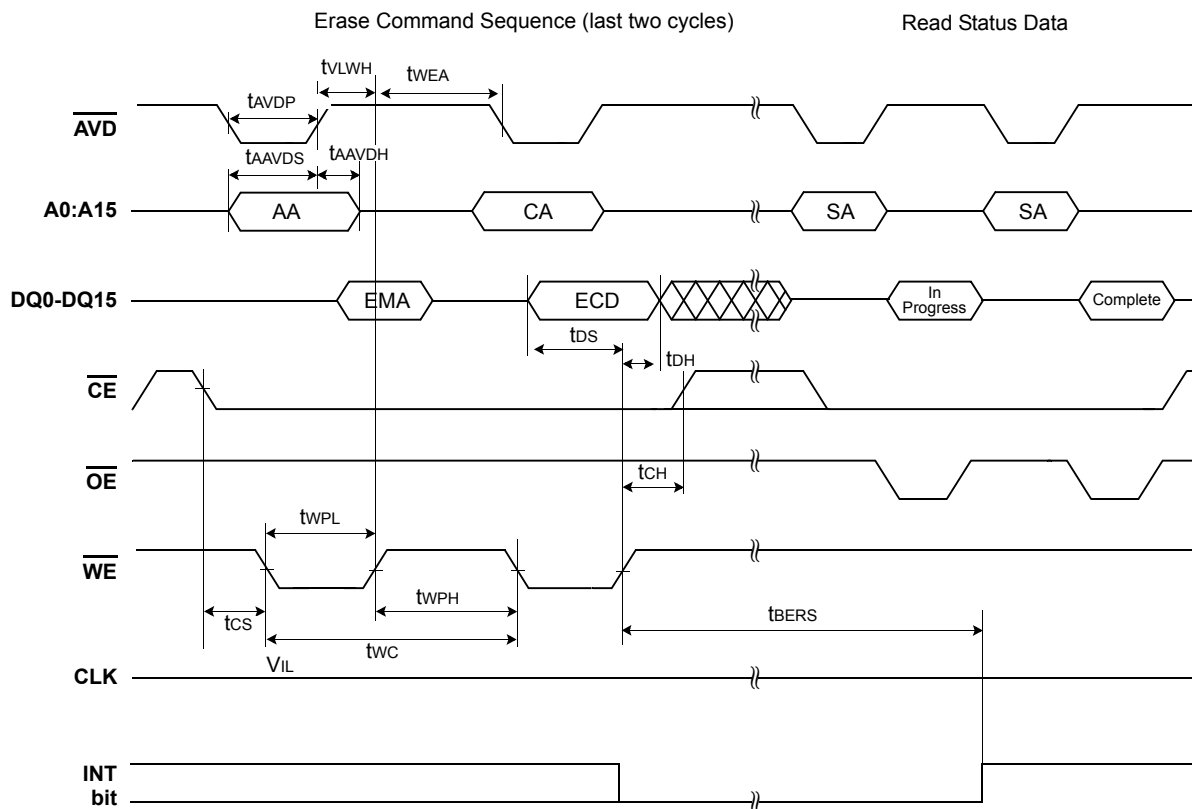
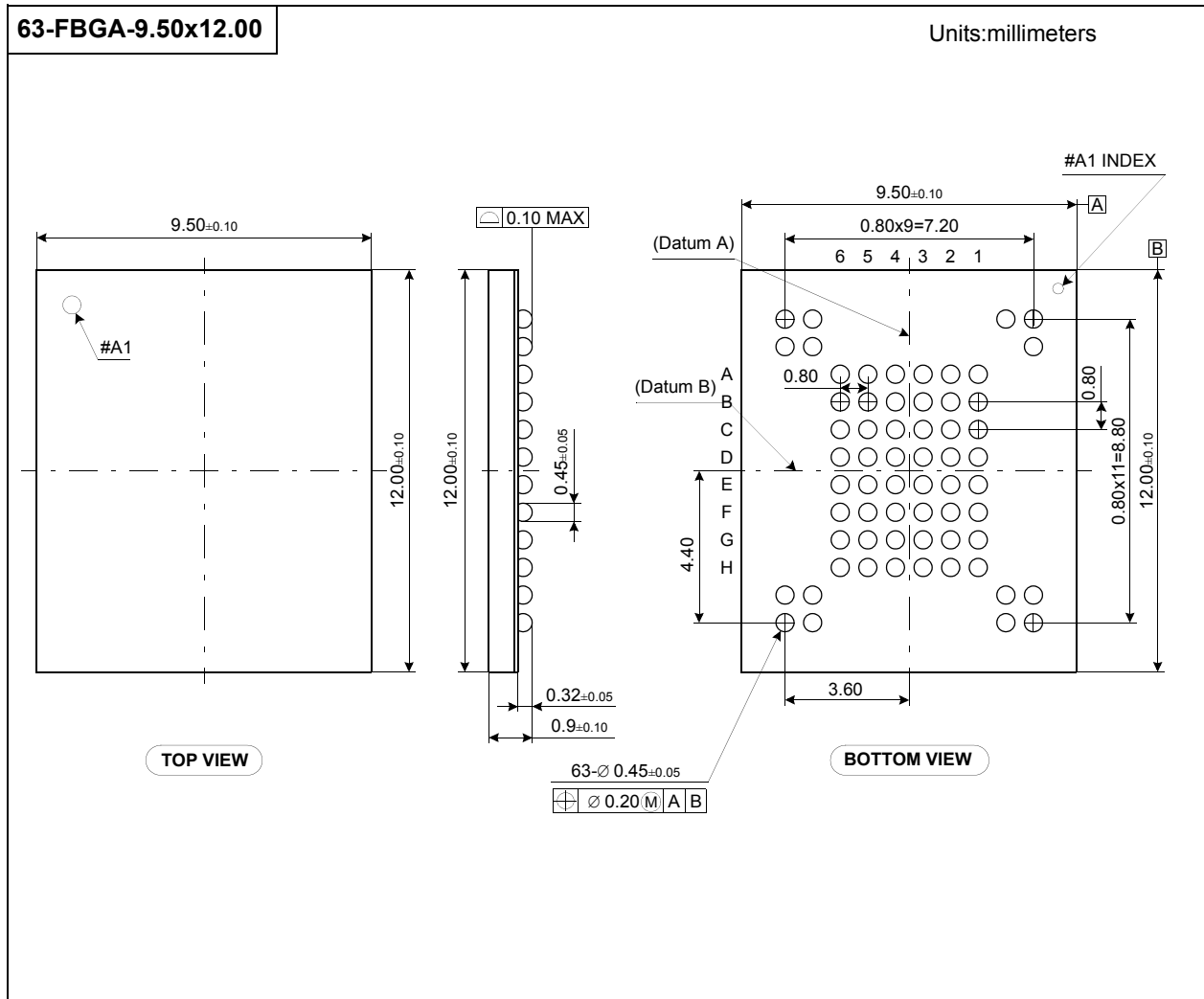


Figure 38. Block Erase Operations

NOTES:

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. "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.

OneNAND512 PACKAGE DIMENSIONS



ORDERING INFORMATION

