

OneNAND™ Specification

| Density | Part No. | Vcc(core & IO) | Temperature | PKG |
|---------|-----------------|------------------|-------------|------------|
| 1Gb | KFG1G16Q2M-DEB6 | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |
| 2Gb | KFH2G16Q2M-DEB6 | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |
| 4Gb | KFW4G16Q2M-DEB6 | 1.8V(1.7V~1.95V) | Extended | 63FBGA(LF) |

Version: Ver. 1.1
Date: Aug 26, 2005

1.0 INTRODUCTION

This specification contains information about the Samsung Electronics Company OneNAND™, Flash memory product family. Section 1.0 includes a general overview, revision history, and product ordering information.

Section 2.0 describes the OneNAND device. Section 3.0 provides information about device operation. Electrical specifications and timing waveforms are in Sections 4.0 through 6.0. Section 7.0 provides additional application and technical notes pertaining to use of the OneNAND. Package dimensions are found in Section 8.0

**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.

MuxOneNAND™, 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

1.1 Revision History

[Document Title](#)

OneNAND

[Revision History](#)

| <u>Revision No.</u> | <u>History</u> | <u>Draft Date</u> | <u>Remark</u> |
|---------------------|--|-------------------|---------------|
| 0.0 | Initial Issue. | Oct. 26, 2004 | Preliminary |
| 0.1 | 1. Corrected Errata 2. Revised cache read flow chart 3. Revised standby current 4. Revised spare area description 5. Added CE don't care state for Asynch Write, Load, Program, and Block Erase timing diagram | Dec. 7, 2004 | Preliminary |
| 0.2 | 1. Corrected Errata 2. Added Copy-back Program Operation With Random Data Input 3. Pended Active Erase Current 4. Changed tBA from 11ns to 11.5ns | Dec. 24, 2004 | Preliminary |
| 0.3 | 1. Corrected the errata 2. Revised typical value of ISB from 50uA to 10uA 3. Revised maximum value of ISB from 100uA to 50uA 4. Revised erase current as TBD 5. Revised maximum value of tCE, tAA and tACC from 70ns to 76ns 6. Revised Vcc-IO description 7. Revised Spare Area description 8. Added extra information on Controller Status Register 9. Added commands related to Interrupt Status Register bits 10. Revised Write Protection Status on Chapter 3.4.3 11. Revised Copy-Back Program Operation description 12. Added extra information on Multi-Block Erase Operation 13. Disabled FBA restriction in OTP operation 14. Revised Cache Read Flow Chart 15. Added ISB information on DDP 16. Revised Reset Parameter descriptions 17. Added RDY information on Warm Reset Timing diagram 18. Added information on Data Protection Timing During Power Down 19. Revised Interrupt pin rise and falling slope graph 20. Added restriction on address register setting on Dual Operations 21. Added restriction on address register setting on Cache Read Operation | Jan. 10, 2005 | Preliminary |
| 0.4 | 1. Corrected the errata 2. Updated DC parameters to RMS values 3. Added QDP feature 4. Added Speed Information on Product Number 5. Revised tOEZ description 6. Revised OTP register setting restriction 7. Added ILI, ILO and ISB information on QDP 8. Added Boot Sequence Information on Technical Notes 9. Added Cint Information | Feb. 25, 2005 | Preliminary |
| 0.4.1 | 1. Revised OTP Load Operation Flow Chart | Mar. 2, 2005 | Preliminary |

Revision History

| <u>Revision No.</u> | <u>History</u> | <u>Draft Date</u> | <u>Remark</u> |
|---------------------|---|-------------------|---------------|
| 1.0 | <ol style="list-style-type: none">1. Corrected the errata2. Added DFS restriction to Multi Block Erase.3. Added Data Protection flow chart.4. Removed Cache Read Operation.5. Added additional information on command register.6. Revised Interrupt status register information.7. Added INT pin schematic.8. Changed tPGM1 to 205 from 320us, tPGM2 to 220 from 350us.9. Revised ECC Bypass description10. Revised AC/DC parameters11. Revised Reset Parameters and Timing Diagrams. | May. 17, 2005 | Final |
| 1.1 | <ol style="list-style-type: none">1. Corrected the errata2. Deleted 2.65V/3.3V Device Descriptions.3. Revised Data Protection Flow Chart.4. Revised Invalid Block Table Creation Flow Chart.5. Revised Multi Block Erase Description6. Revised Reset Mode Operation. | Aug. 26, 2005 | Final |

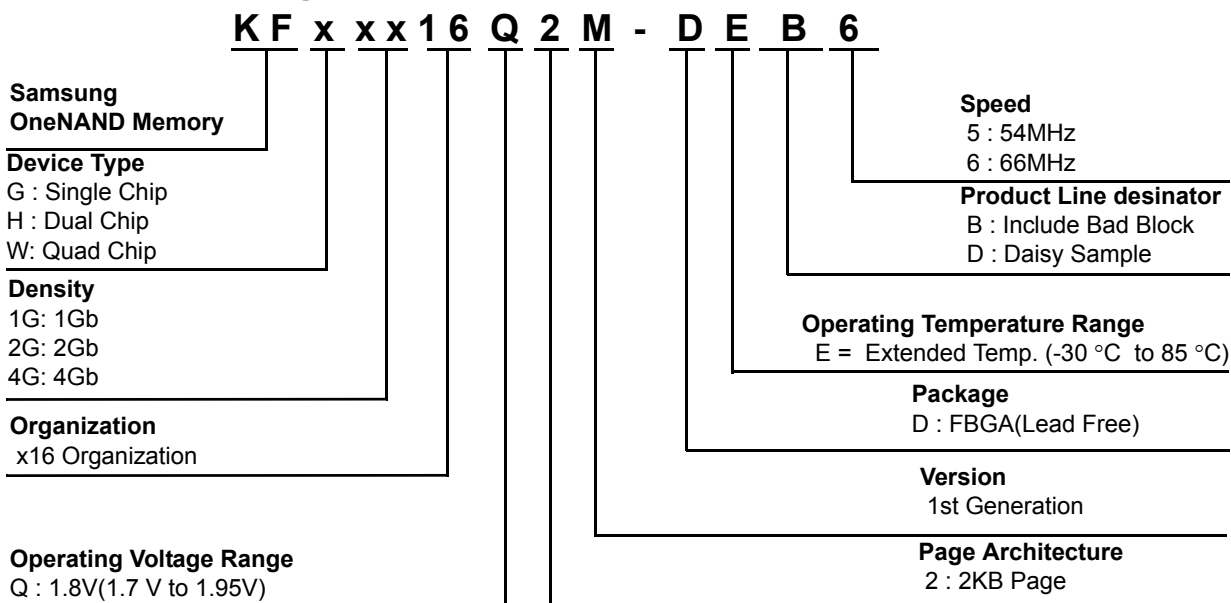
1.2 Flash Product Type Selector

Samsung offers a variety of Flash solutions including NAND Flash, OneNAND™ and NOR Flash. Samsung offers Flash products both component and a variety of card formats including RS-MMC, MMC, CompactFlash, and SmartMedia.

To determine which Samsung Flash product solution is best for your application, refer the product selector chart.

| Application Requires | Samsung Flash Products | | |
|------------------------|------------------------------|-------------------|-----|
| | NAND | OneNAND | NOR |
| Fast Random Read | | | • |
| Fast Sequential Read | • | • | |
| Fast Write/Program | • | • | |
| Multi Block Erase | | • (Max 64 Blocks) | • |
| Erase Suspend/Resume | | • | • |
| Copyback | • (EDC) | • (ECC) | |
| Lock/Unlock/Lock-Tight | | • | • |
| ECC | External (Hardware/Software) | Internal | X |
| Scalability | • | • | |

1.3 Ordering Information



1.4 Architectural Benefits

OneNAND is a highly integrated non-volatile memory solution based around a NAND Flash memory array.

The chip integrates system features including:

- A BootRAM and bootloader
- Two independent bi-directional 2KB DataRAM buffers
- A High-Speed x16 Host Interface
- On-chip Error Correction
- On-chip NOR interface controller

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

OneNAND takes advantage of the higher performance NAND program time, low power, and high density and combines it with the synchronous read performance of NOR. The NOR Flash host interface makes OneNAND an ideal solution for applications like G3 Smart Phones, Camera Phones, and mobile applications that have large, advanced multimedia applications and operating systems, but lack a NAND controller.

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

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.

1.5 Product Features

Device Architecture

- Design Technology: 90nm
- Supply Voltage: 1.8V (1.7V ~ 1.95V)
- Host Interface: 16 bit
- 5KB Internal BufferRAM: 1KB BootRAM, 4KB DataRAM
- SLC NAND Array: (2K+64)B Page Size, (128K+4K)B Block Size

Device Performance

- Host Interface Type: Synchronous Burst Read
 - Up to 66MHz clock frequency
 - Linear Burst 4-, 8-, 16-, 32-words with wrap around
 - Continuous 1K word Sequential BurstAsynchronous Random Read
 - 76ns access timeAsynchronous Random Write
 - Latency 3(up to 40MHz), 4, 5, 6, and 7
 - Up to 4 sectors using Sector Count Register
 - Cold/Warm/Hot/NAND Flash Core Resets
 - Up to 64 Blocks
- Programmable Burst Read Latency: Typical Power,
 - Standby current : 10uA @ Single , 20uA @ DDP, 40uA @ QDP
- Multiple Sector Read/Write: - Synchronous Burst Read current(66MHz) : 15mA @ Single, 20mA @ DDP/QDP
- Multiple Reset Modes: - Load current : 30mA
- Multi Block Erase: - Program current : 25mA
- Low Power Dissipation: - Erase current : 20mA
 - Multi Block Erase current : 20mA
 - Endurance : 100K Program/Erase Cycles
 - Data Retention : 10 Years

System Hardware

- Voltage detector generating internal reset signal from Vcc
- Hardware reset input (RP)
- Data Protection Modes
 - Write Protection for BootRAM
 - Write Protection for NAND Flash Array
 - Write Protection during power-up
 - Write Protection during power-down
- User-controlled One Time Programmable(OTP) area
- Internal 2bit EDC / 1bit ECC
- Internal Bootloader supports Booting Solution in system
- Handshaking Feature
 - INT pin indicates Ready / Busy
 - Polling the interrupt register status bit
 - by ID register
- Detailed chip information

Packaging

- 1G products 63ball, 10mm x 13mm x max 1.0mmt, 0.8mm ball pitch FBGA
- 2G DDP products 63ball, 11mm x 13mm x max 1.2mmt, 0.8mm ball pitch FBGA
- 4G QDP products 63ball, 11mm x 13mm x max 1.4mmt, 0.8mm ball pitch FBGA

1.6 General Overview

OneNAND™ is a monolithic integrated circuit with a NAND Flash array using a NOR Flash interface. This device includes control logic, a NAND Flash array, and 5KB of internal BufferRAM. The BufferRAM reserves 1KB for boot code buffering (BootRAM) and 4KB for data buffering (DataRAM), split between 2 independent buffers. It has a x16 Host Interface and a random access time speed of ~76ns.

The device operates up to a maximum host-driven clock frequency of 66MHz for synchronous reads at Vcc(or Vccq). Refer to chapter 4.2) with minimum 4-clock latency. Below 40MHz it is accessible with minimum 3-clock latency. Appropriate wait cycles are determined by programmable read latency.

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

2.0 DEVICE DESCRIPTION

2.1 Detailed Product Description

The OneNAND is an advanced generation, high-performance NAND-based Flash memory.

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

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

Clock speeds up to 66MHz with a x16 wide I/O yields a 68MByte/second bandwidth.

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

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

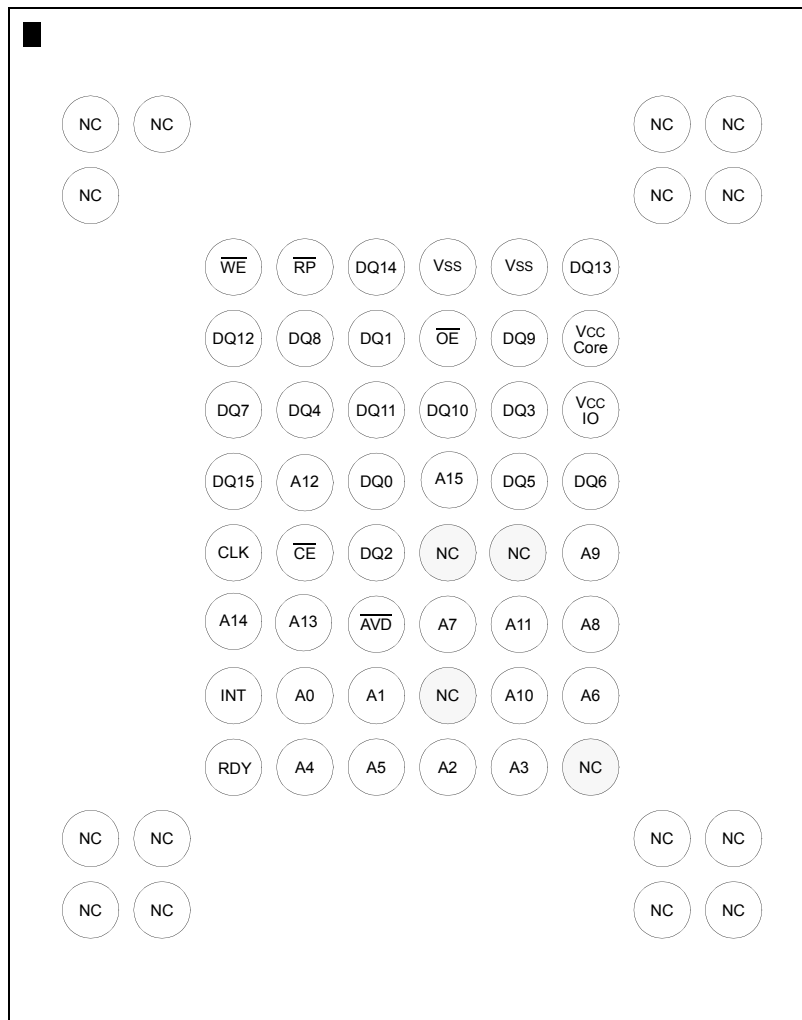
On-chip controller interfaces enable the device to operate in systems without NAND Host controllers.

2.2 Definitions

| | |
|-----------------------|---|
| B (capital letter) | Byte, 8bits |
| W (capital letter) | Word, 16bits |
| b (lower-case letter) | Bit |
| ECC | Error Correction Code |
| Calculated ECC | ECC that has been calculated during a load or program access |
| Written ECC | ECC that has been stored as data in the NAND Flash array or in the BufferRAM |
| BufferRAM | On-chip internal buffer consisting of BootRAM and DataRAM |
| BootRAM | A 1KB portion of the BufferRAM reserved for Boot Code buffering |
| DataRAM | A 4KB portion of the BufferRAM reserved for Data buffering |
| Sector | Part of a Page ; 512B for the main data area and 16B for the spare area. It is also the minimum Load/Program/Copy-Back Program unit during a 1~4 sector operation is available. |
| Data unit | Possible data unit to be read from memory to BufferRAM or to be programmed to memory. - 528B of which 512B is in main area and 16B in spare area - 1056B of which 1024B is in main area and 32B in spare area - 1584B of which 1536B is in main area and 48B in spare area - 2112B of which 2048B is in main area and 64B in spare area |

2.3 Pin Configuration

2.3.1 1Gb Product (KFG1G16Q2M)

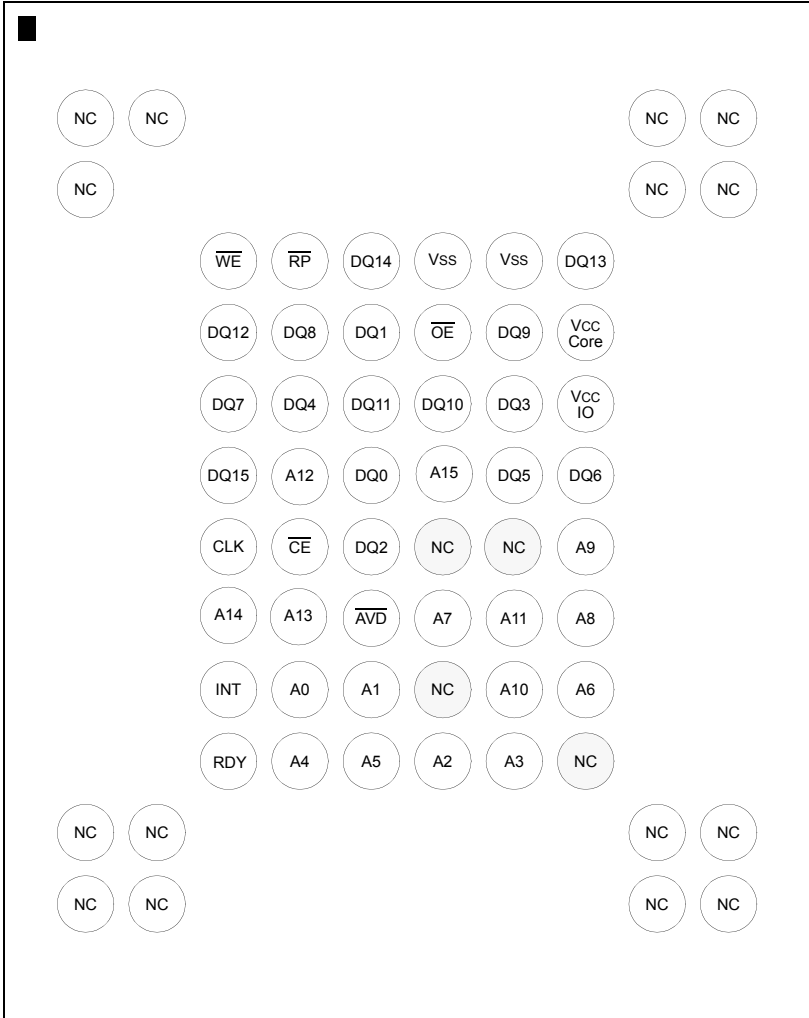


(TOP VIEW, Balls Facing Down)

63ball FBGA OneNAND Chip

63ball, 10mm x 13mm x max 1.0mm, 0.8mm ball pitch FBGA

2.3.2 2Gb Product (KFH2G16Q2M)

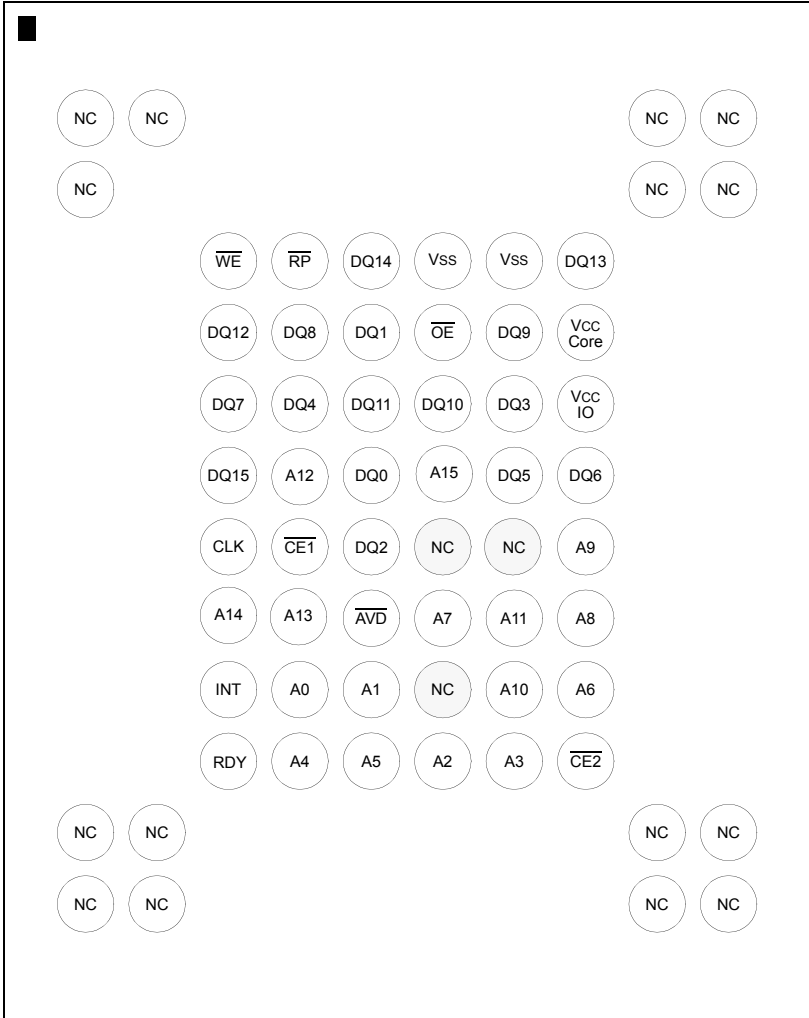


(TOP VIEW, Balls Facing Down)

63ball FBGA OneNAND Chip

63ball, 11mm x 13mm x max 1.2mm, 0.8mm ball pitch FBGA

2.3.3 4Gb Product (KFW4G16Q2M)



(TOP VIEW, Balls Facing Down)

63ball FBGA OneNAND Chip

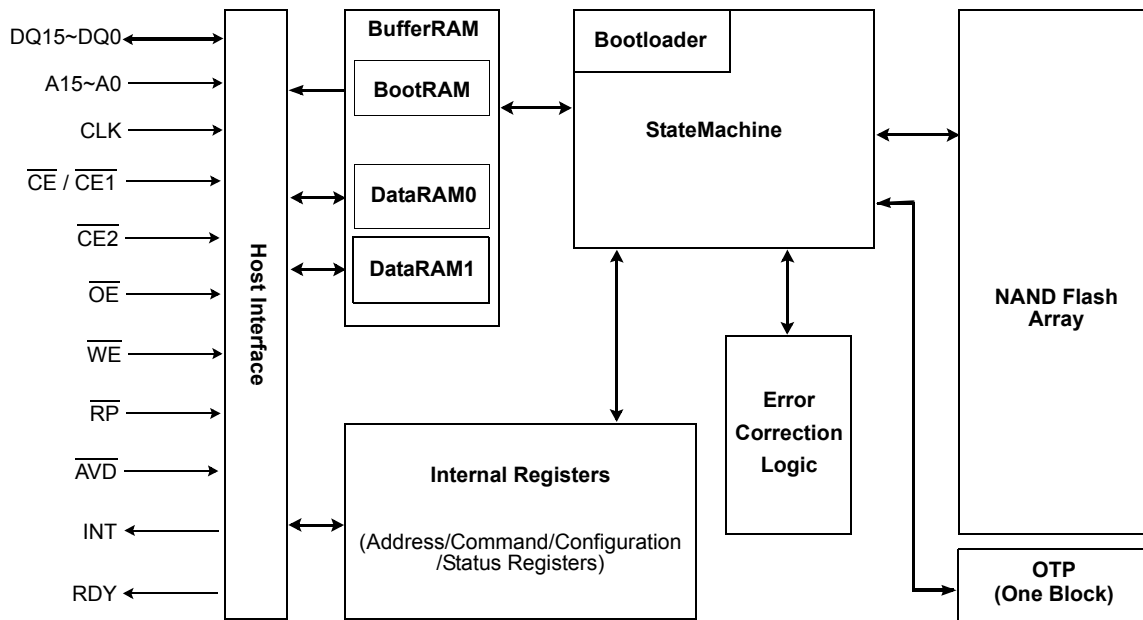
63ball, 11mm x 13mm x max 1.4mm, 0.8mm ball pitch FBGA

2.4 Pin Description

| Pin Name | Type | Name and Description |
|----------------------------------|------|---|
| Host Interface | | |
| A15~A0 | I | Address Inputs - Inputs for addresses during read and write operation, which are for addressing BufferRAM & Register. |
| DQ15~DQ0 | I/O | Data Inputs/Outputs - Inputs data during program and commands for all operations, outputs data during memory array/register read cycles. Data pins float to high-impedance when the chip is deselected or outputs are disabled. |
| INT | O | Interrupt Notifies the Host when a command is 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 | Ready Indicates data valid in synchronous read modes and is activated while \overline{CE} is low |
| CLK | I | Clock 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 | Write Enable \overline{WE} controls writes to the bufferRAM and registers. Datas are latched on the \overline{WE} pulse's rising edge |
| \overline{AVD} | I | Address Valid Detect 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. > Low : for asynchronous mode, indicates valid address; for burst mode, causes starting address to be latched on rising edge on CLK > High : device ignores address inputs |
| \overline{RP} | I | Reset Pin When low, \overline{RP} resets internal operation of OneNAND. \overline{RP} status is don't care during power-up and bootloading. |
| $\overline{CE} / \overline{CE1}$ | I | Chip Enable \overline{CE} -low activates internal control logic, and \overline{CE} -high deselected the device, places it in standby state, and places DQ in Hi-Z. The \overline{CE} input enables device for Single or DDP . The $\overline{CE1}$ input enables the first DDP device(KFH2G16Q2M) for QDP(KFW4G16Q2M) |
| $\overline{CE2}$ | I | Chip Enable The $\overline{CE2}$ input enables the second DDP device(KFH2G16Q2M) for QDP(KFW4G16Q2M) |
| \overline{OE} | I | Output Enable \overline{OE} -low enables the device's output data buffers during a read cycle. |
| Power Supply | | |
| Vcc-Core / Vcc | | Power for OneNAND Core This is the power supply for OneNAND Core. |
| Vcc-IO / Vccq | | Power for OneNAND I/O This is the power supply for OneNAND I/O Vcc-IO / Vccq is internally separated from Vcc-Core / Vcc. |
| Vss | | Ground for OneNAND |
| etc. | | |
| DNU | | Do Not Use Leave it disconnected. These pins are used for testing. |
| NC | | No Connection Lead is not internally connected. |

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

2.5 Block Diagram



2.6 Memory Array Organization

The OneNAND architecture integrates several memory areas on a single chip.

2.6.1 Internal (NAND Array) Memory Organization

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

Main Area

The main area is the primary memory array. This main area is divided into Blocks of 64 Pages. Within a Block, each Page is 2KB and is comprised of 4 Sectors. Within a Page, each Sector is 512B and is comprised of 256 Words.

Spare Area

The spare area is used for invalid block information and ECC storage. Spare area of internal memory is associated with corresponding main area of internal memory. Within a Block, each Page has four 16B Sectors of spare area. Each spare area Sector is 8 words.

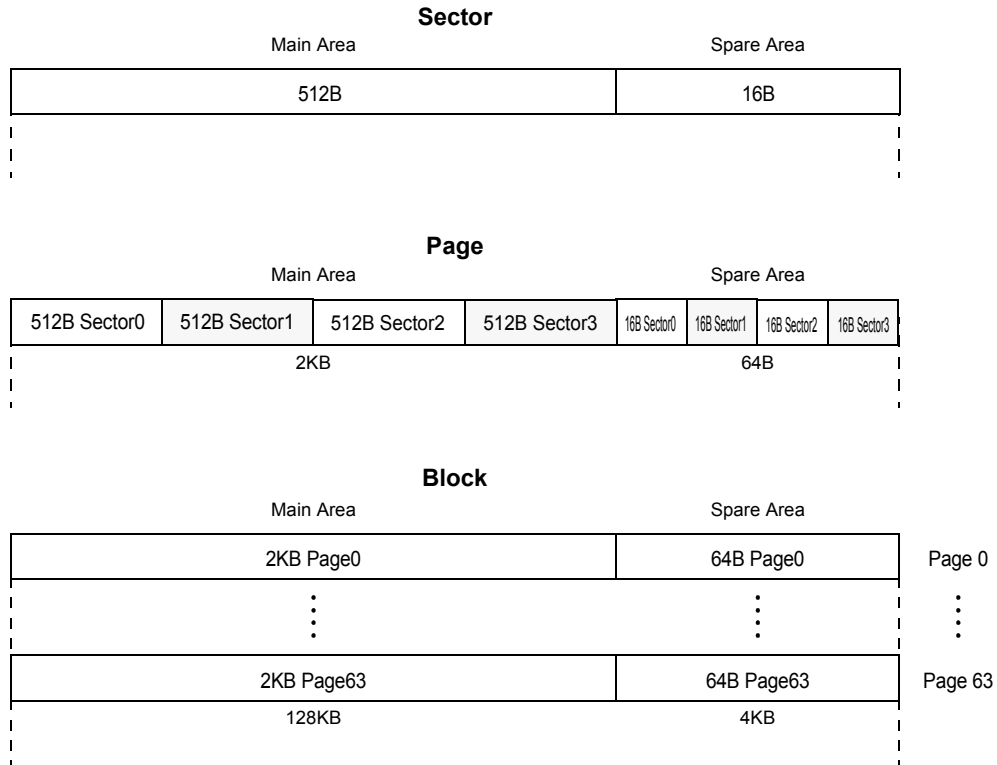
OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

Internal Memory Array Information

| Area | Block | Page | Sector |
|-------|-------|------|--------|
| Main | 128KB | 2KB | 512B |
| Spare | 4KB | 64B | 16B |

Internal Memory Array Organization

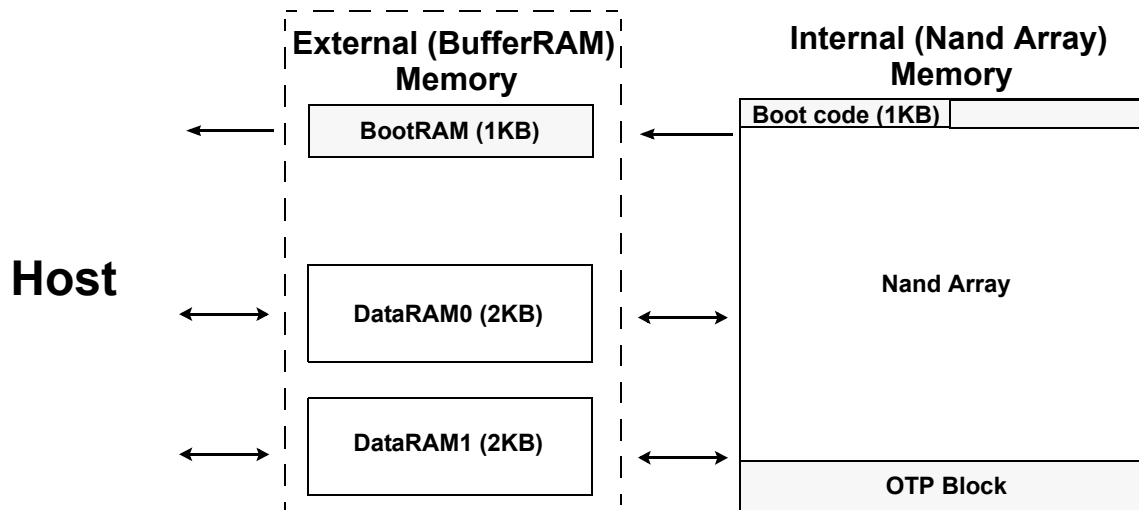


2.6.2 External (BufferRAM) Memory Organization

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

The BootRAM is a 1KB buffer that receives Boot Code from the internal memory and makes it available to the host at start up.

There are two independent 2KB bi-directional data buffers, DataRAM0 and DataRAM1. These dual buffers enable the host to execute simultaneous Read-While load, and Write-While-program operations after Boot Up. During Boot Up, the BootRam is used by the host to initialize the main memory, and deliver boot code from NAND Flash core to host.

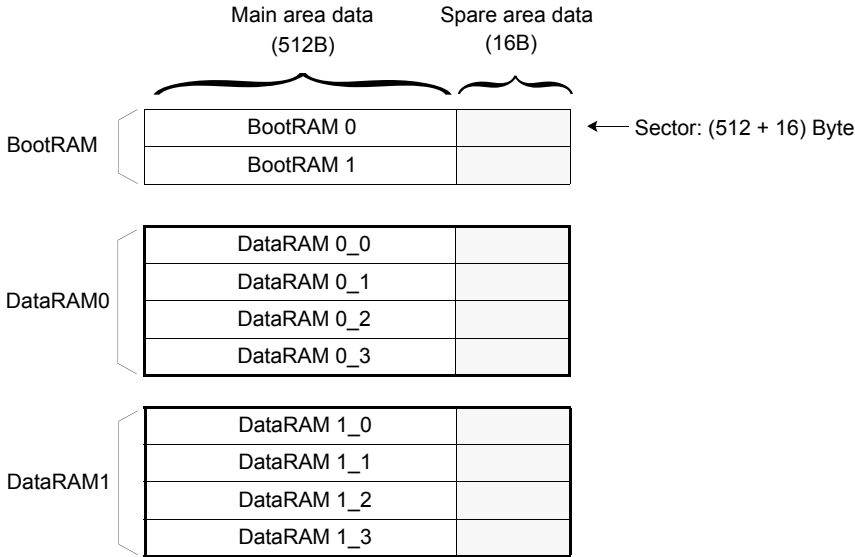


The external memory is divided into a main area and a spare area. Each buffer is the equivalent size of a Sector. The main area data is 512B. The spare area data is 16B.

External Memory Array Information

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

External Memory Array Organization



2.7 Memory Map

The following tables are the memory maps for the OneNAND.

2.7.1 Internal (NAND Array) Memory Organization

The following tables show the Internal Memory address map in 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block512 | 0200h | 0000h~00FFh | 128KB | Block544 | 0220h | 0000h~00FFh | 128KB |
| Block513 | 0201h | 0000h~00FFh | 128KB | Block545 | 0221h | 0000h~00FFh | 128KB |
| Block514 | 0202h | 0000h~00FFh | 128KB | Block546 | 0222h | 0000h~00FFh | 128KB |
| Block515 | 0203h | 0000h~00FFh | 128KB | Block547 | 0223h | 0000h~00FFh | 128KB |
| Block516 | 0204h | 0000h~00FFh | 128KB | Block548 | 0224h | 0000h~00FFh | 128KB |
| Block517 | 0205h | 0000h~00FFh | 128KB | Block549 | 0225h | 0000h~00FFh | 128KB |
| Block518 | 0206h | 0000h~00FFh | 128KB | Block550 | 0226h | 0000h~00FFh | 128KB |
| Block519 | 0207h | 0000h~00FFh | 128KB | Block551 | 0227h | 0000h~00FFh | 128KB |
| Block520 | 0208h | 0000h~00FFh | 128KB | Block552 | 0228h | 0000h~00FFh | 128KB |
| Block521 | 0209h | 0000h~00FFh | 128KB | Block553 | 0229h | 0000h~00FFh | 128KB |
| Block522 | 020Ah | 0000h~00FFh | 128KB | Block554 | 022Ah | 0000h~00FFh | 128KB |
| Block523 | 020Bh | 0000h~00FFh | 128KB | Block555 | 022Bh | 0000h~00FFh | 128KB |
| Block524 | 020Ch | 0000h~00FFh | 128KB | Block556 | 022Ch | 0000h~00FFh | 128KB |
| Block525 | 020Dh | 0000h~00FFh | 128KB | Block557 | 022Dh | 0000h~00FFh | 128KB |
| Block526 | 020Eh | 0000h~00FFh | 128KB | Block558 | 022Eh | 0000h~00FFh | 128KB |
| Block527 | 020Fh | 0000h~00FFh | 128KB | Block559 | 022Fh | 0000h~00FFh | 128KB |
| Block528 | 0210h | 0000h~00FFh | 128KB | Block560 | 0230h | 0000h~00FFh | 128KB |
| Block529 | 0211h | 0000h~00FFh | 128KB | Block561 | 0231h | 0000h~00FFh | 128KB |
| Block530 | 0212h | 0000h~00FFh | 128KB | Block562 | 0232h | 0000h~00FFh | 128KB |
| Block531 | 0213h | 0000h~00FFh | 128KB | Block563 | 0233h | 0000h~00FFh | 128KB |
| Block532 | 0214h | 0000h~00FFh | 128KB | Block564 | 0234h | 0000h~00FFh | 128KB |
| Block533 | 0215h | 0000h~00FFh | 128KB | Block565 | 0235h | 0000h~00FFh | 128KB |
| Block534 | 0216h | 0000h~00FFh | 128KB | Block566 | 0236h | 0000h~00FFh | 128KB |
| Block535 | 0217h | 0000h~00FFh | 128KB | Block567 | 0237h | 0000h~00FFh | 128KB |
| Block536 | 0218h | 0000h~00FFh | 128KB | Block568 | 0238h | 0000h~00FFh | 128KB |
| Block537 | 0219h | 0000h~00FFh | 128KB | Block569 | 0239h | 0000h~00FFh | 128KB |
| Block538 | 021Ah | 0000h~00FFh | 128KB | Block570 | 023Ah | 0000h~00FFh | 128KB |
| Block539 | 021Bh | 0000h~00FFh | 128KB | Block571 | 023Bh | 0000h~00FFh | 128KB |
| Block540 | 021Ch | 0000h~00FFh | 128KB | Block572 | 023Ch | 0000h~00FFh | 128KB |
| Block541 | 021Dh | 0000h~00FFh | 128KB | Block573 | 023Dh | 0000h~00FFh | 128KB |
| Block542 | 021Eh | 0000h~00FFh | 128KB | Block574 | 023Eh | 0000h~00FFh | 128KB |
| Block543 | 021Fh | 0000h~00FFh | 128KB | Block575 | 023Fh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block576 | 0240h | 0000h~00FFh | 128KB | Block608 | 0260h | 0000h~00FFh | 128KB |
| Block577 | 0241h | 0000h~00FFh | 128KB | Block609 | 0261h | 0000h~00FFh | 128KB |
| Block578 | 0242h | 0000h~00FFh | 128KB | Block610 | 0262h | 0000h~00FFh | 128KB |
| Block579 | 0243h | 0000h~00FFh | 128KB | Block611 | 0263h | 0000h~00FFh | 128KB |
| Block580 | 0244h | 0000h~00FFh | 128KB | Block612 | 0264h | 0000h~00FFh | 128KB |
| Block581 | 0245h | 0000h~00FFh | 128KB | Block613 | 0265h | 0000h~00FFh | 128KB |
| Block582 | 0246h | 0000h~00FFh | 128KB | Block614 | 0266h | 0000h~00FFh | 128KB |
| Block583 | 0247h | 0000h~00FFh | 128KB | Block615 | 0267h | 0000h~00FFh | 128KB |
| Block584 | 0248h | 0000h~00FFh | 128KB | Block616 | 0268h | 0000h~00FFh | 128KB |
| Block585 | 0249h | 0000h~00FFh | 128KB | Block617 | 0269h | 0000h~00FFh | 128KB |
| Block586 | 024Ah | 0000h~00FFh | 128KB | Block618 | 026Ah | 0000h~00FFh | 128KB |
| Block587 | 024Bh | 0000h~00FFh | 128KB | Block619 | 026Bh | 0000h~00FFh | 128KB |
| Block588 | 024Ch | 0000h~00FFh | 128KB | Block620 | 026Ch | 0000h~00FFh | 128KB |
| Block589 | 024Dh | 0000h~00FFh | 128KB | Block621 | 026Dh | 0000h~00FFh | 128KB |
| Block590 | 024Eh | 0000h~00FFh | 128KB | Block622 | 026Eh | 0000h~00FFh | 128KB |
| Block591 | 024Fh | 0000h~00FFh | 128KB | Block623 | 026Fh | 0000h~00FFh | 128KB |
| Block592 | 0250h | 0000h~00FFh | 128KB | Block624 | 0270h | 0000h~00FFh | 128KB |
| Block593 | 0251h | 0000h~00FFh | 128KB | Block625 | 0271h | 0000h~00FFh | 128KB |
| Block594 | 0252h | 0000h~00FFh | 128KB | Block626 | 0272h | 0000h~00FFh | 128KB |
| Block595 | 0253h | 0000h~00FFh | 128KB | Block627 | 0273h | 0000h~00FFh | 128KB |
| Block596 | 0254h | 0000h~00FFh | 128KB | Block628 | 0274h | 0000h~00FFh | 128KB |
| Block597 | 0255h | 0000h~00FFh | 128KB | Block629 | 0275h | 0000h~00FFh | 128KB |
| Block598 | 0256h | 0000h~00FFh | 128KB | Block630 | 0276h | 0000h~00FFh | 128KB |
| Block599 | 0257h | 0000h~00FFh | 128KB | Block631 | 0277h | 0000h~00FFh | 128KB |
| Block600 | 0258h | 0000h~00FFh | 128KB | Block632 | 0278h | 0000h~00FFh | 128KB |
| Block601 | 0259h | 0000h~00FFh | 128KB | Block633 | 0279h | 0000h~00FFh | 128KB |
| Block602 | 025Ah | 0000h~00FFh | 128KB | Block634 | 027Ah | 0000h~00FFh | 128KB |
| Block603 | 025Bh | 0000h~00FFh | 128KB | Block635 | 027Bh | 0000h~00FFh | 128KB |
| Block604 | 025Ch | 0000h~00FFh | 128KB | Block636 | 027Ch | 0000h~00FFh | 128KB |
| Block605 | 025Dh | 0000h~00FFh | 128KB | Block637 | 027Dh | 0000h~00FFh | 128KB |
| Block606 | 025Eh | 0000h~00FFh | 128KB | Block638 | 027Eh | 0000h~00FFh | 128KB |
| Block607 | 025Fh | 0000h~00FFh | 128KB | Block639 | 027Fh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block640 | 0280h | 0000h~00FFh | 128KB | Block672 | 02A0h | 0000h~00FFh | 128KB |
| Block641 | 0281h | 0000h~00FFh | 128KB | Block673 | 02A1h | 0000h~00FFh | 128KB |
| Block642 | 0282h | 0000h~00FFh | 128KB | Block674 | 02A2h | 0000h~00FFh | 128KB |
| Block643 | 0283h | 0000h~00FFh | 128KB | Block675 | 02A3h | 0000h~00FFh | 128KB |
| Block644 | 0284h | 0000h~00FFh | 128KB | Block676 | 02A4h | 0000h~00FFh | 128KB |
| Block645 | 0285h | 0000h~00FFh | 128KB | Block677 | 02A5h | 0000h~00FFh | 128KB |
| Block646 | 0286h | 0000h~00FFh | 128KB | Block678 | 02A6h | 0000h~00FFh | 128KB |
| Block647 | 0287h | 0000h~00FFh | 128KB | Block679 | 02A7h | 0000h~00FFh | 128KB |
| Block648 | 0288h | 0000h~00FFh | 128KB | Block680 | 02A8h | 0000h~00FFh | 128KB |
| Block649 | 0289h | 0000h~00FFh | 128KB | Block681 | 02A9h | 0000h~00FFh | 128KB |
| Block650 | 028Ah | 0000h~00FFh | 128KB | Block682 | 02AAh | 0000h~00FFh | 128KB |
| Block651 | 028Bh | 0000h~00FFh | 128KB | Block683 | 02ABh | 0000h~00FFh | 128KB |
| Block652 | 028Ch | 0000h~00FFh | 128KB | Block684 | 02ACh | 0000h~00FFh | 128KB |
| Block653 | 028Dh | 0000h~00FFh | 128KB | Block685 | 02ADh | 0000h~00FFh | 128KB |
| Block654 | 028Eh | 0000h~00FFh | 128KB | Block686 | 02AEh | 0000h~00FFh | 128KB |
| Block655 | 028Fh | 0000h~00FFh | 128KB | Block687 | 02AFh | 0000h~00FFh | 128KB |
| Block656 | 0290h | 0000h~00FFh | 128KB | Block688 | 02B0h | 0000h~00FFh | 128KB |
| Block657 | 0291h | 0000h~00FFh | 128KB | Block689 | 02B1h | 0000h~00FFh | 128KB |
| Block658 | 0292h | 0000h~00FFh | 128KB | Block690 | 02B2h | 0000h~00FFh | 128KB |
| Block659 | 0293h | 0000h~00FFh | 128KB | Block691 | 02B3h | 0000h~00FFh | 128KB |
| Block660 | 0294h | 0000h~00FFh | 128KB | Block692 | 02B4h | 0000h~00FFh | 128KB |
| Block661 | 0295h | 0000h~00FFh | 128KB | Block693 | 02B5h | 0000h~00FFh | 128KB |
| Block662 | 0296h | 0000h~00FFh | 128KB | Block694 | 02B6h | 0000h~00FFh | 128KB |
| Block663 | 0297h | 0000h~00FFh | 128KB | Block695 | 02B7h | 0000h~00FFh | 128KB |
| Block664 | 0298h | 0000h~00FFh | 128KB | Block696 | 02B8h | 0000h~00FFh | 128KB |
| Block665 | 0299h | 0000h~00FFh | 128KB | Block697 | 02B9h | 0000h~00FFh | 128KB |
| Block666 | 029Ah | 0000h~00FFh | 128KB | Block698 | 02BAh | 0000h~00FFh | 128KB |
| Block667 | 029Bh | 0000h~00FFh | 128KB | Block699 | 02BBh | 0000h~00FFh | 128KB |
| Block668 | 029Ch | 0000h~00FFh | 128KB | Block700 | 02BCh | 0000h~00FFh | 128KB |
| Block669 | 029Dh | 0000h~00FFh | 128KB | Block701 | 02BDh | 0000h~00FFh | 128KB |
| Block670 | 029Eh | 0000h~00FFh | 128KB | Block702 | 02BEh | 0000h~00FFh | 128KB |
| Block671 | 029Fh | 0000h~00FFh | 128KB | Block703 | 02BFh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block704 | 02C0h | 0000h~00FFh | 128KB | Block736 | 02E0h | 0000h~00FFh | 128KB |
| Block705 | 02C1h | 0000h~00FFh | 128KB | Block737 | 02E1h | 0000h~00FFh | 128KB |
| Block706 | 02C2h | 0000h~00FFh | 128KB | Block738 | 02E2h | 0000h~00FFh | 128KB |
| Block707 | 02C3h | 0000h~00FFh | 128KB | Block739 | 02E3h | 0000h~00FFh | 128KB |
| Block708 | 02C4h | 0000h~00FFh | 128KB | Block740 | 02E4h | 0000h~00FFh | 128KB |
| Block709 | 02C5h | 0000h~00FFh | 128KB | Block741 | 02E5h | 0000h~00FFh | 128KB |
| Block710 | 02C6h | 0000h~00FFh | 128KB | Block742 | 02E6h | 0000h~00FFh | 128KB |
| Block711 | 02C7h | 0000h~00FFh | 128KB | Block743 | 02E7h | 0000h~00FFh | 128KB |
| Block712 | 02C8h | 0000h~00FFh | 128KB | Block744 | 02E8h | 0000h~00FFh | 128KB |
| Block713 | 02C9h | 0000h~00FFh | 128KB | Block745 | 02E9h | 0000h~00FFh | 128KB |
| Block714 | 02CAh | 0000h~00FFh | 128KB | Block746 | 02EAh | 0000h~00FFh | 128KB |
| Block715 | 02CBh | 0000h~00FFh | 128KB | Block747 | 02EBh | 0000h~00FFh | 128KB |
| Block716 | 02CCh | 0000h~00FFh | 128KB | Block748 | 02ECh | 0000h~00FFh | 128KB |
| Block717 | 02CDh | 0000h~00FFh | 128KB | Block749 | 02EDh | 0000h~00FFh | 128KB |
| Block718 | 02CEh | 0000h~00FFh | 128KB | Block750 | 02EEh | 0000h~00FFh | 128KB |
| Block719 | 02CFh | 0000h~00FFh | 128KB | Block751 | 02EFh | 0000h~00FFh | 128KB |
| Block720 | 02D0h | 0000h~00FFh | 128KB | Block752 | 02F0h | 0000h~00FFh | 128KB |
| Block721 | 02D1h | 0000h~00FFh | 128KB | Block753 | 02F1h | 0000h~00FFh | 128KB |
| Block722 | 02D2h | 0000h~00FFh | 128KB | Block754 | 02F2h | 0000h~00FFh | 128KB |
| Block723 | 02D3h | 0000h~00FFh | 128KB | Block755 | 02F3h | 0000h~00FFh | 128KB |
| Block724 | 02D4h | 0000h~00FFh | 128KB | Block756 | 02F4h | 0000h~00FFh | 128KB |
| Block725 | 02D5h | 0000h~00FFh | 128KB | Block757 | 02F5h | 0000h~00FFh | 128KB |
| Block726 | 02D6h | 0000h~00FFh | 128KB | Block758 | 02F6h | 0000h~00FFh | 128KB |
| Block727 | 02D7h | 0000h~00FFh | 128KB | Block759 | 02F7h | 0000h~00FFh | 128KB |
| Block728 | 02D8h | 0000h~00FFh | 128KB | Block760 | 02F8h | 0000h~00FFh | 128KB |
| Block729 | 02D9h | 0000h~00FFh | 128KB | Block761 | 02F9h | 0000h~00FFh | 128KB |
| Block730 | 02DAh | 0000h~00FFh | 128KB | Block762 | 02FAh | 0000h~00FFh | 128KB |
| Block731 | 02DBh | 0000h~00FFh | 128KB | Block763 | 02FBh | 0000h~00FFh | 128KB |
| Block732 | 02DCh | 0000h~00FFh | 128KB | Block764 | 02FCh | 0000h~00FFh | 128KB |
| Block733 | 02DDh | 0000h~00FFh | 128KB | Block765 | 02FDh | 0000h~00FFh | 128KB |
| Block734 | 02DEh | 0000h~00FFh | 128KB | Block766 | 02FEh | 0000h~00FFh | 128KB |
| Block735 | 02DFh | 0000h~00FFh | 128KB | Block767 | 02FFh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block768 | 0300h | 0000h~00FFh | 128KB | Block800 | 0320h | 0000h~00FFh | 128KB |
| Block769 | 0301h | 0000h~00FFh | 128KB | Block801 | 0321h | 0000h~00FFh | 128KB |
| Block770 | 0302h | 0000h~00FFh | 128KB | Block802 | 0322h | 0000h~00FFh | 128KB |
| Block771 | 0303h | 0000h~00FFh | 128KB | Block803 | 0323h | 0000h~00FFh | 128KB |
| Block772 | 0304h | 0000h~00FFh | 128KB | Block804 | 0324h | 0000h~00FFh | 128KB |
| Block773 | 0305h | 0000h~00FFh | 128KB | Block805 | 0325h | 0000h~00FFh | 128KB |
| Block774 | 0306h | 0000h~00FFh | 128KB | Block806 | 0326h | 0000h~00FFh | 128KB |
| Block775 | 0307h | 0000h~00FFh | 128KB | Block807 | 0327h | 0000h~00FFh | 128KB |
| Block776 | 0308h | 0000h~00FFh | 128KB | Block808 | 0328h | 0000h~00FFh | 128KB |
| Block777 | 0309h | 0000h~00FFh | 128KB | Block809 | 0329h | 0000h~00FFh | 128KB |
| Block778 | 030Ah | 0000h~00FFh | 128KB | Block810 | 032Ah | 0000h~00FFh | 128KB |
| Block779 | 030Bh | 0000h~00FFh | 128KB | Block811 | 032Bh | 0000h~00FFh | 128KB |
| Block780 | 030Ch | 0000h~00FFh | 128KB | Block812 | 032Ch | 0000h~00FFh | 128KB |
| Block781 | 030Dh | 0000h~00FFh | 128KB | Block813 | 032Dh | 0000h~00FFh | 128KB |
| Block782 | 030Eh | 0000h~00FFh | 128KB | Block814 | 032Eh | 0000h~00FFh | 128KB |
| Block783 | 030Fh | 0000h~00FFh | 128KB | Block815 | 032Fh | 0000h~00FFh | 128KB |
| Block784 | 0310h | 0000h~00FFh | 128KB | Block816 | 0330h | 0000h~00FFh | 128KB |
| Block785 | 0311h | 0000h~00FFh | 128KB | Block817 | 0331h | 0000h~00FFh | 128KB |
| Block786 | 0312h | 0000h~00FFh | 128KB | Block818 | 0332h | 0000h~00FFh | 128KB |
| Block787 | 0313h | 0000h~00FFh | 128KB | Block819 | 0333h | 0000h~00FFh | 128KB |
| Block788 | 0314h | 0000h~00FFh | 128KB | Block820 | 0334h | 0000h~00FFh | 128KB |
| Block789 | 0315h | 0000h~00FFh | 128KB | Block821 | 0335h | 0000h~00FFh | 128KB |
| Block790 | 0316h | 0000h~00FFh | 128KB | Block822 | 0336h | 0000h~00FFh | 128KB |
| Block791 | 0317h | 0000h~00FFh | 128KB | Block823 | 0337h | 0000h~00FFh | 128KB |
| Block792 | 0318h | 0000h~00FFh | 128KB | Block824 | 0338h | 0000h~00FFh | 128KB |
| Block793 | 0319h | 0000h~00FFh | 128KB | Block825 | 0339h | 0000h~00FFh | 128KB |
| Block794 | 031Ah | 0000h~00FFh | 128KB | Block826 | 033Ah | 0000h~00FFh | 128KB |
| Block795 | 031Bh | 0000h~00FFh | 128KB | Block827 | 033Bh | 0000h~00FFh | 128KB |
| Block796 | 031Ch | 0000h~00FFh | 128KB | Block828 | 033Ch | 0000h~00FFh | 128KB |
| Block797 | 031Dh | 0000h~00FFh | 128KB | Block829 | 033Dh | 0000h~00FFh | 128KB |
| Block798 | 031Eh | 0000h~00FFh | 128KB | Block830 | 033Eh | 0000h~00FFh | 128KB |
| Block799 | 031Fh | 0000h~00FFh | 128KB | Block831 | 033Fh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block832 | 0340h | 0000h~00FFh | 128KB | Block864 | 0360h | 0000h~00FFh | 128KB |
| Block833 | 0341h | 0000h~00FFh | 128KB | Block865 | 0361h | 0000h~00FFh | 128KB |
| Block834 | 0342h | 0000h~00FFh | 128KB | Block866 | 0362h | 0000h~00FFh | 128KB |
| Block835 | 0343h | 0000h~00FFh | 128KB | Block867 | 0363h | 0000h~00FFh | 128KB |
| Block836 | 0344h | 0000h~00FFh | 128KB | Block868 | 0364h | 0000h~00FFh | 128KB |
| Block837 | 0345h | 0000h~00FFh | 128KB | Block869 | 0365h | 0000h~00FFh | 128KB |
| Block838 | 0346h | 0000h~00FFh | 128KB | Block870 | 0366h | 0000h~00FFh | 128KB |
| Block839 | 0347h | 0000h~00FFh | 128KB | Block871 | 0367h | 0000h~00FFh | 128KB |
| Block840 | 0348h | 0000h~00FFh | 128KB | Block872 | 0368h | 0000h~00FFh | 128KB |
| Block841 | 0349h | 0000h~00FFh | 128KB | Block873 | 0369h | 0000h~00FFh | 128KB |
| Block842 | 034Ah | 0000h~00FFh | 128KB | Block874 | 036Ah | 0000h~00FFh | 128KB |
| Block843 | 034Bh | 0000h~00FFh | 128KB | Block875 | 036Bh | 0000h~00FFh | 128KB |
| Block844 | 034Ch | 0000h~00FFh | 128KB | Block876 | 036Ch | 0000h~00FFh | 128KB |
| Block845 | 034Dh | 0000h~00FFh | 128KB | Block877 | 036Dh | 0000h~00FFh | 128KB |
| Block846 | 034Eh | 0000h~00FFh | 128KB | Block878 | 036Eh | 0000h~00FFh | 128KB |
| Block847 | 034Fh | 0000h~00FFh | 128KB | Block879 | 036Fh | 0000h~00FFh | 128KB |
| Block848 | 0350h | 0000h~00FFh | 128KB | Block880 | 0370h | 0000h~00FFh | 128KB |
| Block849 | 0351h | 0000h~00FFh | 128KB | Block881 | 0371h | 0000h~00FFh | 128KB |
| Block850 | 0352h | 0000h~00FFh | 128KB | Block882 | 0372h | 0000h~00FFh | 128KB |
| Block851 | 0353h | 0000h~00FFh | 128KB | Block883 | 0373h | 0000h~00FFh | 128KB |
| Block852 | 0354h | 0000h~00FFh | 128KB | Block884 | 0374h | 0000h~00FFh | 128KB |
| Block853 | 0355h | 0000h~00FFh | 128KB | Block885 | 0375h | 0000h~00FFh | 128KB |
| Block854 | 0356h | 0000h~00FFh | 128KB | Block886 | 0376h | 0000h~00FFh | 128KB |
| Block855 | 0357h | 0000h~00FFh | 128KB | Block887 | 0377h | 0000h~00FFh | 128KB |
| Block856 | 0358h | 0000h~00FFh | 128KB | Block888 | 0378h | 0000h~00FFh | 128KB |
| Block857 | 0359h | 0000h~00FFh | 128KB | Block889 | 0379h | 0000h~00FFh | 128KB |
| Block858 | 035Ah | 0000h~00FFh | 128KB | Block890 | 037Ah | 0000h~00FFh | 128KB |
| Block859 | 035Bh | 0000h~00FFh | 128KB | Block891 | 037Bh | 0000h~00FFh | 128KB |
| Block860 | 035Ch | 0000h~00FFh | 128KB | Block892 | 037Ch | 0000h~00FFh | 128KB |
| Block861 | 035Dh | 0000h~00FFh | 128KB | Block893 | 037Dh | 0000h~00FFh | 128KB |
| Block862 | 035Eh | 0000h~00FFh | 128KB | Block894 | 037Eh | 0000h~00FFh | 128KB |
| Block863 | 035Fh | 0000h~00FFh | 128KB | Block895 | 037Fh | 0000h~00FFh | 128KB |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|----------|---------------|-------------------------|-------|
| Block896 | 0380h | 0000h~00FFh | 128KB | Block928 | 03A0h | 0000h~00FFh | 128KB |
| Block897 | 0381h | 0000h~00FFh | 128KB | Block929 | 03A1h | 0000h~00FFh | 128KB |
| Block898 | 0382h | 0000h~00FFh | 128KB | Block930 | 03A2h | 0000h~00FFh | 128KB |
| Block899 | 0383h | 0000h~00FFh | 128KB | Block931 | 03A3h | 0000h~00FFh | 128KB |
| Block900 | 0384h | 0000h~00FFh | 128KB | Block932 | 03A4h | 0000h~00FFh | 128KB |
| Block901 | 0385h | 0000h~00FFh | 128KB | Block933 | 03A5h | 0000h~00FFh | 128KB |
| Block902 | 0386h | 0000h~00FFh | 128KB | Block934 | 03A6h | 0000h~00FFh | 128KB |
| Block903 | 0387h | 0000h~00FFh | 128KB | Block935 | 03A7h | 0000h~00FFh | 128KB |
| Block904 | 0388h | 0000h~00FFh | 128KB | Block936 | 03A8h | 0000h~00FFh | 128KB |
| Block905 | 0389h | 0000h~00FFh | 128KB | Block937 | 03A9h | 0000h~00FFh | 128KB |
| Block906 | 038Ah | 0000h~00FFh | 128KB | Block938 | 03AAh | 0000h~00FFh | 128KB |
| Block907 | 038Bh | 0000h~00FFh | 128KB | Block939 | 03ABh | 0000h~00FFh | 128KB |
| Block908 | 038Ch | 0000h~00FFh | 128KB | Block940 | 03ACh | 0000h~00FFh | 128KB |
| Block909 | 038Dh | 0000h~00FFh | 128KB | Block941 | 03ADh | 0000h~00FFh | 128KB |
| Block910 | 038Eh | 0000h~00FFh | 128KB | Block942 | 03AEh | 0000h~00FFh | 128KB |
| Block911 | 038Fh | 0000h~00FFh | 128KB | Block943 | 03AFh | 0000h~00FFh | 128KB |
| Block912 | 0390h | 0000h~00FFh | 128KB | Block944 | 03B0h | 0000h~00FFh | 128KB |
| Block913 | 0391h | 0000h~00FFh | 128KB | Block945 | 03B1h | 0000h~00FFh | 128KB |
| Block914 | 0392h | 0000h~00FFh | 128KB | Block946 | 03B2h | 0000h~00FFh | 128KB |
| Block915 | 0393h | 0000h~00FFh | 128KB | Block947 | 03B3h | 0000h~00FFh | 128KB |
| Block916 | 0394h | 0000h~00FFh | 128KB | Block948 | 03B4h | 0000h~00FFh | 128KB |
| Block917 | 0395h | 0000h~00FFh | 128KB | Block949 | 03B5h | 0000h~00FFh | 128KB |
| Block918 | 0396h | 0000h~00FFh | 128KB | Block950 | 03B6h | 0000h~00FFh | 128KB |
| Block919 | 0397h | 0000h~00FFh | 128KB | Block951 | 03B7h | 0000h~00FFh | 128KB |
| Block920 | 0398h | 0000h~00FFh | 128KB | Block952 | 03B8h | 0000h~00FFh | 128KB |
| Block921 | 0399h | 0000h~00FFh | 128KB | Block953 | 03B9h | 0000h~00FFh | 128KB |
| Block922 | 039Ah | 0000h~00FFh | 128KB | Block954 | 03BAh | 0000h~00FFh | 128KB |
| Block923 | 039Bh | 0000h~00FFh | 128KB | Block955 | 03BBh | 0000h~00FFh | 128KB |
| Block924 | 039Ch | 0000h~00FFh | 128KB | Block956 | 03BCh | 0000h~00FFh | 128KB |
| Block925 | 039Dh | 0000h~00FFh | 128KB | Block957 | 03BDh | 0000h~00FFh | 128KB |
| Block926 | 039Eh | 0000h~00FFh | 128KB | Block958 | 03BEh | 0000h~00FFh | 128KB |
| Block927 | 039Fh | 0000h~00FFh | 128KB | Block959 | 03BFh | 0000h~00FFh | 128KB |

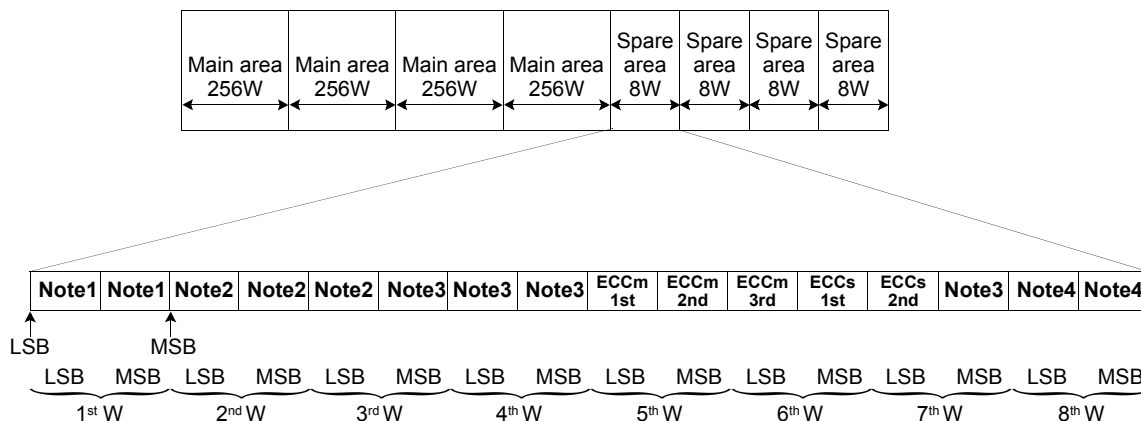
OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| Block | Block Address | Page and Sector Address | Size | Block | Block Address | Page and Sector Address | Size |
|----------|---------------|-------------------------|-------|-----------|---------------|-------------------------|-------|
| Block960 | 03C0h | 0000h~00FFh | 128KB | Block992 | 03E0h | 0000h~00FFh | 128KB |
| Block961 | 03C1h | 0000h~00FFh | 128KB | Block993 | 03E1h | 0000h~00FFh | 128KB |
| Block962 | 03C2h | 0000h~00FFh | 128KB | Block994 | 03E2h | 0000h~00FFh | 128KB |
| Block963 | 03C3h | 0000h~00FFh | 128KB | Block995 | 03E3h | 0000h~00FFh | 128KB |
| Block964 | 03C4h | 0000h~00FFh | 128KB | Block996 | 03E4h | 0000h~00FFh | 128KB |
| Block965 | 03C5h | 0000h~00FFh | 128KB | Block997 | 03E5h | 0000h~00FFh | 128KB |
| Block966 | 03C6h | 0000h~00FFh | 128KB | Block998 | 03E6h | 0000h~00FFh | 128KB |
| Block967 | 03C7h | 0000h~00FFh | 128KB | Block999 | 03E7h | 0000h~00FFh | 128KB |
| Block968 | 03C8h | 0000h~00FFh | 128KB | Block1000 | 03E8h | 0000h~00FFh | 128KB |
| Block969 | 03C9h | 0000h~00FFh | 128KB | Block1001 | 03E9h | 0000h~00FFh | 128KB |
| Block970 | 03CAh | 0000h~00FFh | 128KB | Block1002 | 03EAh | 0000h~00FFh | 128KB |
| Block971 | 03CBh | 0000h~00FFh | 128KB | Block1003 | 03EBh | 0000h~00FFh | 128KB |
| Block972 | 03CCh | 0000h~00FFh | 128KB | Block1004 | 03ECh | 0000h~00FFh | 128KB |
| Block973 | 03CDh | 0000h~00FFh | 128KB | Block1005 | 03EDh | 0000h~00FFh | 128KB |
| Block974 | 03CEh | 0000h~00FFh | 128KB | Block1006 | 03EEh | 0000h~00FFh | 128KB |
| Block975 | 03CFh | 0000h~00FFh | 128KB | Block1007 | 03EFh | 0000h~00FFh | 128KB |
| Block976 | 03D0h | 0000h~00FFh | 128KB | Block1008 | 03F0h | 0000h~00FFh | 128KB |
| Block977 | 03D1h | 0000h~00FFh | 128KB | Block1009 | 03F1h | 0000h~00FFh | 128KB |
| Block978 | 03D2h | 0000h~00FFh | 128KB | Block1010 | 03F2h | 0000h~00FFh | 128KB |
| Block979 | 03D3h | 0000h~00FFh | 128KB | Block1011 | 03F3h | 0000h~00FFh | 128KB |
| Block980 | 03D4h | 0000h~00FFh | 128KB | Block1012 | 03F4h | 0000h~00FFh | 128KB |
| Block981 | 03D5h | 0000h~00FFh | 128KB | Block1013 | 03F5h | 0000h~00FFh | 128KB |
| Block982 | 03D6h | 0000h~00FFh | 128KB | Block1014 | 03F6h | 0000h~00FFh | 128KB |
| Block983 | 03D7h | 0000h~00FFh | 128KB | Block1015 | 03F7h | 0000h~00FFh | 128KB |
| Block984 | 03D8h | 0000h~00FFh | 128KB | Block1016 | 03F8h | 0000h~00FFh | 128KB |
| Block985 | 03D9h | 0000h~00FFh | 128KB | Block1017 | 03F9h | 0000h~00FFh | 128KB |
| Block986 | 03DAh | 0000h~00FFh | 128KB | Block1018 | 03FAh | 0000h~00FFh | 128KB |
| Block987 | 03DBh | 0000h~00FFh | 128KB | Block1019 | 03FBh | 0000h~00FFh | 128KB |
| Block988 | 03DCh | 0000h~00FFh | 128KB | Block1020 | 03FCh | 0000h~00FFh | 128KB |
| Block989 | 03DDh | 0000h~00FFh | 128KB | Block1021 | 03FDh | 0000h~00FFh | 128KB |
| Block990 | 03DEh | 0000h~00FFh | 128KB | Block1022 | 03FEh | 0000h~00FFh | 128KB |
| Block991 | 03DFh | 0000h~00FFh | 128KB | Block1023 | 03FFh | 0000h~00FFh | 128KB |

2.7.2 Internal Memory Spare Area Assignment

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



Spare Area Assignment in the Internal Memory NAND Array Information

| Word | Byte | Note | Description |
|------|------|------|---|
| 1 | LSB | 1 | Invalid Block information in 1st and 2nd page of an invalid block |
| | MSB | | |
| 2 | LSB | 2 | Managed by internal ECC logic for Logical Sector Number data |
| | MSB | | |
| 3 | LSB | 3 | Reserved for future use |
| | MSB | | |
| 4 | LSB | 3 | Reserved for future use |
| | MSB | | |
| 5 | LSB | 3 | Dedicated to internal ECC logic. Read Only. ECCm 1st for main area data |
| | MSB | | Dedicated to internal ECC logic. Read Only. ECCm 2nd for main area data |
| 6 | LSB | 3 | Dedicated to internal ECC logic. Read Only. ECCm 3rd for main area data |
| | MSB | | Dedicated to internal ECC logic. Read Only. ECCs 1st for 2nd word of spare area data |
| 7 | LSB | 3 | Dedicated to internal ECC logic. Read Only. ECCs 2nd for 3rd word of spare area data |
| | MSB | | Reserved for future use |
| 8 | LSB | 4 | Available to the user |
| | MSB | | |

2.7.3 External Memory (BufferRAM) Address Map

The following table shows the External Memory address map in Word and Byte Order.
 Note that the data output is unknown while host reads a register bit of reserved area.

| Division | Address (word order) | Address (byte order) | Size (total 128KB) | | | Usage | Description |
|---------------------|-------------------------|-------------------------|-----------------------|---------|----------|-----------|-----------------------------|
| | | | | | | | |
| Main area (64KB) | 0000h~00FFh | 00000h~001FEh | 512B | 1KB | R | BootM 0 | BootRAM Main sector0 |
| | 0100h~01FFh | 00200h~003FEh | 512B | | | BootM 1 | BootRAM Main sector1 |
| | 0200h~02FFh | 00400h~005FEh | 512B | 4KB | R/W | 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 | 59K | 59K | - | Reserved | Reserved |
| | Spare area (8KB) | 8000h~8007h | 10000h~1000Eh | 16B | 32B | R | BootS 0 |
| 8008h~800Fh | | 10010h~1001Eh | 16B | BootS 1 | | | BootRAM Spare sector1 |
| 8010h~8017h | | 10020h~1002Eh | 16B | 128B | R/W | 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~11FFFEh | 8032B | | | 8032B | - |
| Reserved (24KB) | 9000h~BFFFh | 12000h~17FFEh | 24KB | 24KB | - | Reserved | Reserved |
| Reserved (8KB) | C000h~CFFFh | 18000h~19FFEh | 8KB | 8KB | - | Reserved | Reserved |
| Reserved (16KB) | D000h~EFFFh | 1A000h~1DFFEh | 16KB | 16KB | - | Reserved | Reserved |
| Registers (8KB) | F000h~FFFFh | 1E000h~1FFFEh | 8KB | 8KB | R or R/W | Registers | Registers |

2.7.4 External Memory Detail Map Information

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

• BootRAM(Main area)

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

| | |
|--|--|
| 0000h~00FFh(512B) BootM 0 (sector 0 of page 0) | 0100h~01FFh(512B) BootM 1 (sector 1 of page 0) |
|--|--|

• DataRAM(Main area)

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

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

• BootRAM(Spare area)

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

| | |
|---|---|
| 8000h~8007h(16B) BootS 0 (sector 0 of page 0) | 8008h~800Fh(16B) BootS 1 (sector 1 of page 0) |
|---|---|

• DataRAM(Spare area)

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

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

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

2.7.5 External Memory Spare Area Assignment

← Equivalent to 1word of NAND Flash →

| Buf. | Word Address | Byte Address | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|--------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BootS 0 | 8000h | 10000h | BI | | | | | | | | | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 8005h | 1000Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 8006h | 1000Ch | FFh(Reserved for the future use) | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 800Dh | 1001Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 800Eh | 1001Ch | FFh(Reserved for the future use) | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 8015h | 1002Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 8016h | 1002Ch | FFh(Reserved for the future use) | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 801Dh | 1003Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 801Eh | 1003Ch | FFh(Reserved for the future use) | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 801Fh | 1003Eh | Free Usage | | | | | | | | | | | | | | | |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

| 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 8025h | 1004Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 8026h | 1004Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 802Dh | 1005Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 802Eh | 1005Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 8035h | 1006Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 8036h | 1006Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 803Dh | 1007Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 803Eh | 1007Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 8045h | 1008Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 8046h | 1008Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 8047h | 1008Eh | Free Usage | | | | | | | | | | | | | | | |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

← 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 nd) | | | | | | | | ECC Code for Main area data (1 st) | | | | | | | |
| | 804Dh | 1009Ah | ECC Code for Spare area data (1 st) | | | | | | | | ECC Code for Main area data (3 rd) | | | | | | | |
| | 804Eh | 1009Ch | Reserved for the future use | | | | | | | | ECC Code for Spare area data (2 nd) | | | | | | | |
| | 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.

2.8 Registers

Section 2.8 of this specification provides information about the OneNAND1G registers.

2.8.1 Register Address Map

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

| Address (word order) | Address (byte order) | Name | Host Access | Description |
|-------------------------|-------------------------|---------------------------|----------------|---|
| F000h | 1E000h | Manufacturer ID | R | Manufacturer identification |
| F001h | 1E002h | Device ID | R | Device identification |
| F002h | 1E004h | Version ID | R | N/A |
| F003h | 1E006h | Data Buffer size | R | Data buffer size |
| F004h | 1E008h | Boot Buffer size | R | Boot buffer size |
| F005h | 1E00Ah | Amount of buffers | R | Amount of data/boot buffers |
| F006h | 1E00Ch | Technology | R | Info about technology |
| F007h~F0FFh | 1E00Eh~1E1FEh | Reserved | - | Reserved for user |
| F100h | 1E200h | Start address 1 | R/W | Chip address for selection of NAND Core in DDP & Block address |
| F101h | 1E202h | Start address 2 | R/W | Chip address for selection of BufferRAM in DDP |
| F102h | 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 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 memory and the start Buffer Address The meaning is with which buffer to start and how many 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 memory operation commands |
| F221h | 1E442h | System Configuration 1 | R, R/W | Memory and Host Interface Configuration |
| F222h | 1E444h | System 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 memory operation |
| F241h | 1E482h | Interrupt | R/W | Memory Command Completion Interrupt Status |
| F242h~F24Bh | 1E484h~1E496h | Reserved | - | Reserved for user |
| F24Ch | 1E498h | Start Block Address | R/W | Start memory block address in Write Protection mode |

| Address (word order) | Address (byte order) | Name | Host Access | Description |
|----------------------|----------------------|-------------------------------|-------------|--|
| F24Dh | 1E49Ah | Reserved | R/W | Reserved for user |
| F24Eh | 1E49Ch | Write Protection Status | R | Current memory Write Protection status (unlocked/locked/tight-locked) |
| F24Fh~FEFFh | 1E49Eh~1FDfEh | Reserved | - | Reserved for user |
| FF00h | 1FE00h | ECC Status Register | 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 |

2.8.2 Manufacturer ID Register F000h (R)

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

F000h, default = 00ECh

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

2.8.3 Device ID Register F001h (R)

This Read register describes the device.

F001h, see table for default.

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

Device Identification

| Device Identification | Description |
|----------------------------|--|
| DeviceID [1:0] Vcc | 00 = 1.8V, 01 = 2.65V/3.3V, 10/11 = reserved |
| DeviceID [2] Muxed/Demuxed | 0 = Muxed, 1 = Demuxed |
| DeviceID [3] Single/DPP | 0 = Single, 1 = DPP |
| DeviceID [6:4] Density | 000 = 128Mb, 001 = 256Mb, 010 = 512Mb, 011 = 1Gb, 100 = 2Gb, 101=4Gb |

Device ID Default

| Device | DeviceID[15:0] |
|------------|---------------------|
| KFG1G16Q2M | 0034h |
| KFH2G16Q2M | 004Ch |
| KFW4G16Q2M | 004Ch ¹⁾ |

Note1) Due to KFW4G16Q2M is QDP with dual KFH2G16Q2M, Device ID[15:0] is same as KFH2G16Q2M

2.8.4 Version ID Register F002h

This register is reserved for manufacturer

2.8.5 Data Buffer Size Register F003h (R)

This Read register describes the size of the Data Buffer.

F003h, default = 0800h

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

Data Buffer Size Information

| Version Identification | Description |
|------------------------|---|
| DataBufSize | Total data buffer size in Words equal to 2 buffers of 1024 Words each ($2 \times 1024 = 2^{11}$) in the memory interface |

2.8.6 Boot Buffer Size Register F004h (R)

This Read register describes the size of the Boot Buffer.

F004h, default = 0200h

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

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

2.8.7 Number of Buffers Register F005h (R)

This Read register describes the number of each Buffer.

F005h, default = 0201h

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

Number of Buffers Information

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

2.8.8 Technology Register F006h (R)

This Read register describes the internal NAND array technology.

F006h, default = 0000h

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

Technology Information

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

2.8.9 Start Address1 Register F100h (R/W)

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

F100h, default = 0000h

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

| Device | Number of Block | FBA |
|---------|-----------------|--------------------|
| 2Gb DDP | 2048 | DFS[15] & FBA[9:0] |
| 1Gb | 1024 | FBA[9:0] |

Start Address1 Information

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

2.8.10 Start Address2 Register F101h (R/W)

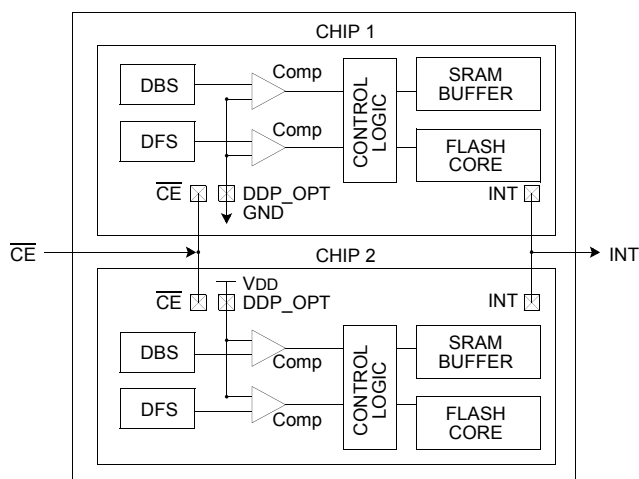
This Read/Write register describes the BufferRAM of DDP (Device BufferRAM Select)

F101h, default = 0000h

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

Start Address2 Information

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



2.8.11 Start Address3 Register F102h (R/W)

This Read/Write register describes the NAND Flash destination block address which will be copy back programmed.

F102h, default = 0000h

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

| Device | Number of Block | FBA |
|--------|-----------------|-----------|
| 1Gb | 1024 | FCBA[9:0] |

Start Address3 Information

| Register Information | Description |
|----------------------|------------------------------------|
| FCBA | NAND Flash Copy Back Block Address |

2.8.12 Start Address4 Register F103h (R/W)

This Read/Write register describes the NAND Flash destination page address in a block and the NAND Flash destination sector address in a page for copy back programming.

F103h, default = 0000h

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

Start Address4 Information

| Item | Description | Default Value | Range |
|------|-------------------------------------|---------------|---|
| FCPA | NAND Flash Copy Back Page Address | 000000 | 000000 ~ 111111, 6 bits for 64 pages |
| FCSA | NAND Flash Copy Back Sector Address | 00 | 00 ~ 11, 2 bits for 4 sectors |

2.8.13 Start Address5 Register F104h

This register is reserved for future use.

2.8.14 Start Address6 Register F105h

This register is reserved for future use.

2.8.15 Start Address7 Register F106h

This register is reserved for future use.

2.8.16 Start Address8 Register F107h (R/W)

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

F107h, default = 0000h

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

Start Address8 Information

| Item | Description | Default Value | Range |
|------|---------------------------|---------------|---|
| FPA | NAND Flash Page Address | 000000 | 000000 ~ 111111, 6 bits for 64 pages |
| FSA | NAND Flash Sector Address | 00 | 00 ~ 11, 2 bits for 4 sectors |

2.8.17 Start Buffer Register F200h (R/W)

This Read/Write register describes the BufferRAM Sector Count (BSC) and BufferRAM Sector Address (BSA).

The BufferRAM Sector Count (BSC) field specifies the number of sectors to be loaded, programmed, or copy back programmed. At 00 value (the default value), the number of sector is "4". If the internal RAM buffer reaches its maximum value of 11, it will count up to 0 value to meet the BSC value. For example, if BSA = 1101, BSC = 00, then the selected BufferRAM will count up from '1101 → 1110 → 1111 → 1100'.

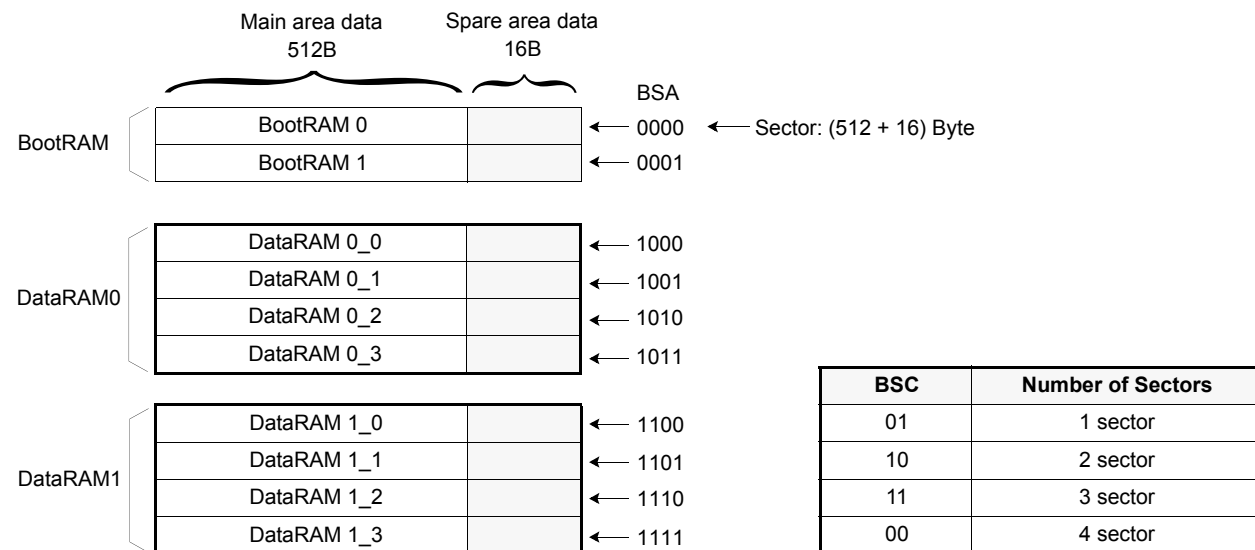
The BufferRAM Sector Address (BSA) is the sector 0~3 address in the internal BootRAM and DataRAM where data is placed.

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 | | | |

Start Buffer Register Information

| Item | Description |
|----------|---|
| BSA[3] | Selection bit between BootRAM and DataRAM |
| BSA[2] | Selection bit between DataRAM0 and DataRAM1 |
| BSA[1:0] | Selection bit between Sector0 and Sector1 in the internal BootRAM Selection bit between Sector0 to Sector3 in the internal DataRAM |



2.8.18 Command Register F220h (R/W)

This Read/Write register describes the operation of the MuxOneNAND 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 below.)

F220h, default = 0000h

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

| CMD | Operation | Acceptable command during busy |
|-------|--|--------------------------------|
| 0000h | Load 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 ¹⁾ | 00F0h, 00F3h |
| 001Ah | Program single/multiple spare data unit from buffer | 00F0h, 00F3h |
| 001Bh | Copy back Program operation | 00F0h, 00F3h |
| 0023h | Unlock NAND array a block | - |
| 002Ah | Lock NAND array a block | - |
| 002Ch | Lock-tight NAND array a block | |
| 0071h | Erase Verify Read | 00F0h, 00F3h |
| 0094h | Block Erase | 00F0h, 00F3h |
| 0095h | Multi-Block Erase | 00F0h, 00F3h |
| 00B0h | Erase Suspend | - |
| 0030h | Erase Resume | 00F0h, 00F3h |
| 00F0h | Reset NAND Flash Core | - |
| 00F3h | Reset OneNAND ²⁾ | - |
| 0065h | OTP Access | 00F0h, 00F3h |

NOTE:

1) 0080h programs both main and spare area, while 001Ah programs only spare area. Refer to chapter 5.8 for NOP limits in issuing these commands. When using 0080h and 001Ah command, Read-only part in spare area must be masked by FF. (Refer to chapter 2.7.2)

2) 'Reset OneNAND'(=Hot reset) command makes the registers and NAND Flash core into default state as the warm reset(=reset by \overline{RP} pin).

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

This Read/Write register describes the system configuration.

F221h, default = 40C0h

| | | | | | | | | | | | | | | | |
|-----|-----|----|-----|----|----|-----|------------|------------|------|-------------|---------------|---|---|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | R/W | | R/W | | | R/W | R/W | R/W | R/W | R/W | R/W | R | | | R |
| RM | BRL | | BL | | | ECC | RDY pol | INT pol | IOBE | RDY Conf | Reserved(000) | | | BWPS | |

Read Mode (RM)

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

Read Mode Information[15]

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

Burst Read Latency (BRL)

| BRL | Latency Cycles |
|-----|------------------|
| 000 | 8(N/A) |
| 001 | 9(N/A) |
| 010 | 10(N/A) |
| 011 | 3(up to 40MHz) |
| 100 | 4(default, min.) |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Burst Read Latency (BRL) Information[14:12]

| Item | Definition | Description |
|------|--------------------|--|
| BRL | Burst Read Latency | Specifies the access latency in the burst read transfer for the initial access |

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

Burst Length (BL)

| 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 | |

Burst Length (BL) Information[11:9]

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

Error Correction Code (ECC) Information[8]

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

RDY Polarity (RDYpol) Information[7]

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

INT Polarity (INTpol) Information[6]

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

I/O Buffer Enable (IOBE)

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

I/O Buffer Enable Information[5]

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

RDY Configuration (RDY conf)

RDY Configuration Information[4]

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

Boot Buffer Write Protect Status (BWPS)

Boot Buffer Write Protect Status Information[0]

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

2.8.20 System Configuration 2 Register F222h

This register is reserved for future use.

2.8.21 Controller Status Register F240h (R)

This Read register shows the overall internal status of the OneNAND and the controller.

F240h, default = 0000h

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|-------|-------|-----|-------------|------|------|------------------|---|---|---|---|--------|
| OnGo | Lock | Load | Prog | Erase | Error | Sus | Reserved(0) | RSTB | OTPL | Reserved(000000) | | | | | TO (0) |

OnGo

This bit shows the overall internal status of the OneNAND device.

OnGo Information[15]

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

Lock

This bit shows whether the host is loading data from the NAND Flash array into the locked BootRAM or whether the host is performing a program/erase of a locked block of the NAND Flash array.

Lock Information[14]

| Lock | Locked/Unlocked Check Result |
|------|------------------------------|
| 0 | Unlocked |
| 1 | Locked |

Load

This bit shows the Load Operation status.

Load Information[13]

| Item | Definition | Description |
|------|-----------------------|---|
| Load | Load Operation status | 0 = ready (default) 1 = busy or error (see controller status output modes) |

Program

This bit shows the Program Operation status.

Program Information[12]

| Item | Definition | Description |
|------|--------------------------|---|
| Prog | Program Operation status | 0 = ready (default) 1 = busy or error (see controller status output modes) |

Erase

This bit shows the Erase Operation status.

Erase Information[11]

| Item | Definition | Description |
|-------|------------------------|---|
| Erase | Erase Operation status | 0 = ready (default) 1 = busy or error (see controller status output modes) |

Error

This bit shows the overall Error status, including Load Reset, Program Reset, and Erase Reset status.

Error Information[10]

| Error | Current Sector/Page Load/Program/CopyBack. Program/ Erase Result and Invalid Command Input |
|-------|---|
| 0 | Pass |
| 1 | Fail |

Erase Suspend (Sus)

This bit shows the Erase Suspend status.

Sus Information[9]

| Sus | Erase Suspend Status |
|-----|--|
| 0 | Erase Resume(Default) |
| 1 | Erase Suspend, Program Ongoing(Susp.), Load Ongoing(Susp.), Program Fail(Susp.), Load Fail(Susp.), Invalid Command(Susp.) |

Reset / Busy (RSTB)

This bit shows the Reset Operation status.

RSTB Information[7]

| Item | Definition | Description |
|------|------------------------|--|
| RSTB | Reset Operation Status | 0 = ready (default) 1 = busy (see controller status output modes) |

OTP Lock Status (OTPL)

This bit shows whether the OTP block is locked or unlocked. Locking the OTP has the effect of a 'write-protect' to guard against accidental re-programming of data stored in the OTP block.

The OTPL status bit is automatically updated at power-on.

OTP Lock Information[6]

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

Time Out (TO)

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

TO Information[0]

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

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

Controller Status Register Output Modes

| Mode | Controller Status Register [15:0] | | | | | | | | | | | |
|--------------------------------------|-----------------------------------|------|------|------|-------|-------|-----|-------------|------|------|-------------|-----|
| | [15] | [14] | [13] | [12] | [11] | [10] | [9] | [8] | [7] | [6] | [5:1] | [0] |
| | OnGo | Lock | Load | Prog | Erase | Error | Sus | Reserved(0) | RSTB | OTPL | Reserved(0) | TO |
| Load Ongoing | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Program Ongoing | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Ongoing | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Reset Ongoing | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0/1 | 00000 | 0 |
| Multi-Block Erase Ongoing | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Verify Read Ongoing | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Load OK | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Program OK | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase OK | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Verify Read OK ³⁾ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Load Fail ¹⁾ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Program Fail | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Fail | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Verify Read Fail ³⁾ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Load Reset ²⁾ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0/1 | 00000 | 0 |
| Program Reset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0/1 | 00000 | 0 |
| Erase Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0/1 | 00000 | 0 |
| Erase Suspend | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0/1 | 00000 | 0 |
| Program Lock | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Erase Lock | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Load Lock(Buffer Lock) | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| OTP Program Fail(Lock) | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 00000 | 0 |
| OTP Program Fail | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 00000 | 0 |
| OTP Erase Fail | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Program Ongoing(Susp.) | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0/1 | 00000 | 0 |
| Load Ongoing(Susp.) | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0/1 | 00000 | 0 |
| Program Fail(Susp.) | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0/1 | 00000 | 0 |
| Load Fail(Susp.) | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0/1 | 00000 | 0 |
| Invalid Command | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0/1 | 00000 | 0 |
| Invalid Command(Susp.) | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0/1 | 00000 | 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. Multi Block Erase status should be checked by Erase Verify Read operation.



2.8.22 Interrupt Status Register F241h (R/W)

This Read/Write register shows status of the OneNAND interrupts.

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

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

Interrupt (INT)

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

INT Interrupt [15]

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|--------------------|--|---------------|----------|-------------|--------------------|
| | | Cold | Warm/hot | | |
| | | 1 | 1 | 0 | off |
| sets itself to '1' | One or more of RI, WI, RSTI and EI is set to '1', or 0065h, 0023h, 0071h, 002A and 002C commands are completed | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, or Cold/Warm/Hot reset is being performed | | | 1→0 | off |

Read Interrupt (RI)

This is the Read interrupt bit.

RI Interrupt [7]

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|--------------------|---|---------------|----------|-------------|--------------------|
| | | Cold | Warm/hot | | |
| | | 1 | 0 | 0 | off |
| sets itself to '1' | At the completion of an Load Operation (0000h, 0013h, Load Data into Buffer, or boot is done) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, or Cold/Warm/Hot reset is being performed | | | 1→0 | off |

Write Interrupt (WI)

This is the Write interrupt bit.

WI Interrupt [6]

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|--------------------|---|---------------|----------|-------------|--------------------|
| | | Cold | Warm/hot | | |
| | | 0 | 0 | 0 | off |
| sets itself to '1' | At the completion of an Program Operation (0080h, 001Ah, 001Bh) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, or Cold/Warm/Hot reset is being performed | | | 1→0 | off |

Erase Interrupt (EI)

This is the Erase interrupt bit.

EI Interrupt [5]

| Status | Conditions | Default State | | Valid State | Interrupt Function |
|--------------------|---|---------------|----------|-------------|--------------------|
| | | Cold | Warm/hot | | |
| | | 0 | 0 | 0 | off |
| sets itself to '1' | At the completion of an Erase Operation (0094h, 0095h, 0030h) | | | 0→1 | Pending |
| clears to '0' | '0' is written to this bit, or Cold/Warm/Hot reset is being performed | | | 1→0 | off |

Reset Interrupt (RSTI)

This is the Reset interrupt bit.

RSTI Interrupt [4]

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

2.8.23 Start Block Address Register F24Ch (R/W)

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

F24Ch, default = 0000h

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

| Device | Number of Block | SBA |
|--------|-----------------|-------|
| 1Gb | 1024 | [9:0] |

SBA Information[9:0]

| Item | Definition | Description |
|------|---------------------|---|
| SBA | Start Block Address | Precedes Lock Block, Unlock Block, or Lock-Tight commands |

2.8.24 End Block Address Register F24Dh

This register is reserved for future use.

2.8.25 NAND Flash Write Protection Status Register F24Eh (R)

This Read register shows the Write Protection Status of the NAND Flash memory array. To read the write protection status, FBA has to be set before reading the register.

F24Eh, default = 0002h

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

Write Protection Status Information[2:0]

| Item | Bit | Definition | Description |
|------|-----|---------------------|--|
| US | 2 | Unlocked Status | 1 = current NAND Flash block is unlocked |
| LS | 1 | Locked Status | 1 = current NAND Flash block is locked |
| LTS | 0 | Locked-Tight Status | 1 = current NAND Flash block is locked-tight |

2.8.26 ECC Status Register FF00h (R)

This Read register shows the Error Correction Status. The OneNAND can detect 1- or 2-bit errors and correct 1-bit errors. 3-bit or more error detection and correction is not supported.

ECC can be performed on the NAND Flash main and spare memory areas. The ECC status register can also show the number of errors in a sector as a result of an ECC check in during a load operation. ECC status bits are also updated during a boot loading operation.

FF00h, default = 0000h

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

Error Status

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

ECC Information[15:0]

| Item | Definition | Description |
|------|--|--|
| ERm0 | 1st selected sector of the main BufferRAM | Status of errors in the 1st selected sector of the main BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERm1 | 2nd selected sector of the main BufferRAM | Status of errors in the 2nd selected sector of the main BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERm2 | 3rd selected sector of the main BufferRAM | Status of errors in the 3rd selected sector of the main BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERm3 | 4th selected sector of the main BufferRAM | Status of errors in the 4th selected sector of the main BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERs0 | 1st selected sector of the spare BufferRAM | Status of errors in the 1st selected sector of the spare BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERs1 | 2nd selected sector of the spare BufferRAM | Status of errors in the 2nd selected sector of the spare BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERs2 | 3rd selected sector of the spare BufferRAM | Status of errors in the 3rd selected sector of the spare BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |
| ERs3 | 4th selected sector of the spare BufferRAM | Status of errors in the 4th selected sector of the spare BufferRAM as a result of an ECC check during a load operation. Also updated during a Bootload operation. |

2.8.27 ECC Result of 1st Selected Sector, Main Area Data **Register FF01h (R)**

This Read register shows the Error Correction result for the 1st selected sector of the main area data. ECCposWord0 is the error position address in the Main Area data of 256 words. ECCposIO0 is the error position address which selects 1 of 16 DQs. ECCposWord0 and ECCposIO0 are also updated at boot loading.

FF01h, default = 0000h

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

2.8.28 ECC Result of 1st Selected Sector, Spare Area Data **Register FF02h (R)**

This Read register shows the Error Correction result for the 1st selected sector of the spare area data. ECClogSector0 is the error position address for 1.5 words of 2nd and 3rd words in the spare area. ECCposIO0 is the error position address which selects 1 of 16 DQs. ECClogSector0 and ECCposIO0 are also updated at boot loading.

FF02h, default = 0000h

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

2.8.29 ECC Result of 2nd Selected Sector, Main Area Data Register FF03h (R)

This Read register shows the Error Correction result for the 2nd selected sector of the main area data. ECCposWord1 is the error position address in the Main Area data of 256 words. ECCposIO1 is the error position address which selects 1 of 16 DQs. ECCposWord1 and ECCposIO1 are also updated at boot loading.

FF03h, default = 0000h

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

2.8.30 ECC Result of 2nd Selected Sector, Spare Area Data Register FF04h (R)

This Read register shows the Error Correction result for the 2nd selected sector of the spare area data. ECClogSector1 is the error position address for 1.5 words of 2nd and 3rd words in the spare area. ECCposIO1 is the error position address which selects 1 of 16 DQs. ECClogSector1 and ECCposIO1 are also updated at boot loading.

FF04h, default = 0000h

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

2.8.31 ECC Result of 3rd Selected Sector, Main Area Data Register FF05h (R)

This Read register shows the Error Correction result for the 3rd selected sector of the main area data. ECCposWord2 is the error position address in the Main Area data of 256 words. ECCposIO2 is the error position address which selects 1 of 16 DQs. ECCposWord2 and ECCposIO2 are also updated at boot loading.

FF05h, default = 0000h

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

2.8.32 ECC Result of 3rd Selected Sector, Spare Area Data Register FF06h (R)

This Read register shows the Error Correction result for the 3rd selected sector of the spare area data. ECClogSector2 is the error position address for 1.5 words of 2nd and 3rd words in the spare area. ECCposIO2 is the error position address which selects 1 of 16 DQs. ECClogSector2 and ECCposIO2 are also updated at boot loading.

FF06h, default = 0000h

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

2.8.33 ECC Result of 4th Selected Sector, Main Area Data Register FF07h (R)

This Read register shows the Error Correction result for the 4th selected sector of the main area data. ECCposWord3 is the error position address in the Main Area data of 256 words. ECCposIO3 is the error position address which selects 1 of 16 DQs. ECCposWord3 and ECCposIO3 are also updated at boot loading.

FF07h, default = 0000h

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

2.8.34 ECC Result of 4th Selected Sector, Spare Area Data Register FF08h (R)

This Read register shows the Error Correction result for the 4th selected sector of the spare area data. ECClogSector3 is the error position address for 1.5 words of 2nd and 3rd words in the spare area. ECCposIO3 is the error position address which selects 1 of 16 DQs. ECClogSector3 and ECCposIO3 are also updated at boot loading.

FF08h, default = 0000h

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

ECC Log Sector

ECClogSector0~ECClogSector3 indicates the error position in the 2nd word and LSB of 3rd word in the spare area. Refer to note 2 in chapter 2.7.2

ECClogSector Information [5:4]

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

3.0 DEVICE OPERATION

This section of the datasheet discusses the operation of the OneNAND device. It is followed by AC/DC Characteristics and Timing Diagrams which may be consulted for further information.

The OneNAND supports either a command-based or a register-based interface for performing operations on the device including reading device ID, writing data to buffer etc.

3.1 Command Based Operation

The command-based interface is active in the boot partition. 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.

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 including address and data to the boot partition or writing an improper command will terminate the previous command sequence and make the device enter the ready status.

The defined valid command based sequences are stated in Command Sequences Table.

Command Sequences

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

NOTE:

- 1) DP(Data Partition) : DataRAM Area
- 2) BP(Boot Partition) : BootRAM Area [0000h ~ 01FFh, 8000h ~ 800Fh].
- 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
 0002h -> Current Block Write Protection Status
- 6) WE toggling can terminate 'Read Identification Data' operation.

3.1.1 Reading Data From Buffer

The buffer memory can be read by addressing a Read to the desired buffer area.

3.1.2 Writing Data to Buffer

The buffer memory can be written to by addressing a Write to a desired buffer area.

3.1.3 Reset OneNAND Command

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

3.1.4 Load Data Into Buffer Command

Load Data into Buffer command is a two-cycle command. Two sequential designated command activates this operation. Sequentially writing 00E0h and 0000h to the boot partition [0000h~01FFh, 8000h~800Fh] will load one page to DataRAM0. 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.

3.1.5 Read Identification Data Command

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



Identification Data Description

| Address | Data Out |
|---------|---|
| 0000h | Manufacturer ID : 00ECh |
| 0001h | Device ID : refer to chapter 2.8.3 |
| 0002h | Current Block Write Protection Status ¹⁾ |

Note 1) To read the write protection status, FBA has to be set before issuing this command.

3.2 Device Bus Operation

The device bus operations are shown in the table below.

| Operation | \overline{CE} | \overline{OE} | \overline{WE} | 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 | X |
| Asynchronous Read | L | L | H | Add. In | Data Out | H | L |  or L |
| Load Initial Burst Read | 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_{IL} (Low), H=V_{IH} (High), X=Don't Care.

3.3 Reset Mode Operation

The One NAND has 4 reset modes: Cold/Warm/Hot Reset, and NAND Flash Core Reset. Section 3.3 discusses the operation of these reset modes.

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

Internal Register Reset Table

| | Internal Registers | Default | Cold Reset | Warm Reset (RP) | Hot Reset (00F3h) | Hot Reset (BP-F0) | NAND Flash Core Reset (00F0h) |
|-------|---|----------|------------|-----------------|-------------------|-------------------|-------------------------------|
| F000h | Manufacturer ID Register (R) | 00ECh | N/A | N/A | N/A | N/A | N/A |
| F001h | Device ID Register (R): OneNAND | (Note 3) | N/A | N/A | N/A | N/A | N/A |
| F002h | Version ID Register (R): N/A | N/A | 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): 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 Address8 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 | (Note1) | (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 | 0000h | N/A | N/A |
| F24Dh | End Block Address: N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| F24Eh | NAND Flash Write Protection Status (R) | 0002h | 0002h | 0002h | 0002h | 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, IOBE is 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 Device ID Register F001h.

3.3.1 Cold Reset Mode Operation

See Timing Diagram 6.11

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

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

It takes approximately 70us to copy 1KB of bootcode. Upon completion of loading into the BootRAM, it is available to be read by the host. The INT pin is not available until after IOBE = 1 and IOBE bit can be changed by host.

3.3.2 Warm Reset Mode Operation

See Timing Diagrams 6.12

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

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

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

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

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

3.3.3 Hot Reset Mode Operation

See Timing Diagram 6.13

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

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

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

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

3.3.4 NAND Flash Core Reset Mode Operation

See Timing Diagram 6.14

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

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

3.4 Write Protection Operation

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

3.4.1 BootRAM Write Protection Operation

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

After the bootcode loading is completed, the BootRAM is always locked to protect the boot code from the accidental write.

3.4.2 NAND Flash Array Write Protection Operation

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

Hardware Write Protection Operation

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

Software Write Protection Operation

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

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

3.4.3 NAND Array Write Protection States

There are three lock states in the NAND Array: unlocked, locked, and locked-tight.

OneNAND1G supports lock/unlock/lock-tight by **one block**, so each block should be locked/unlocked/locked-tight individually.

Write Protection Status

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

The followings summarize locking status.

example)

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

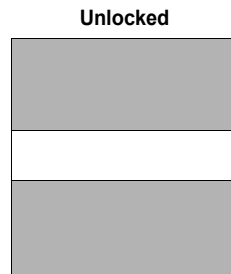
-> If host executes unlock block operation, then [2:0] values turn to 100.

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

3.4.3.1 Unlocked NAND Array Write Protection State

An Unlocked block can be programmed or erased. The status of an unlocked block can be changed to locked or locked-tight using the appropriate software command. (locked-tight state can be achieved via lock-tight command which follows lock command)

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



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

3.4.3.2 Locked NAND Array Write Protection State

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

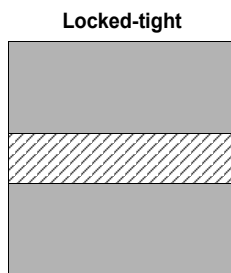


Lock Command Sequence:
Start block address+Lock block command (002Ah)

3.4.3.3 Locked-tight NAND Array Write Protection State

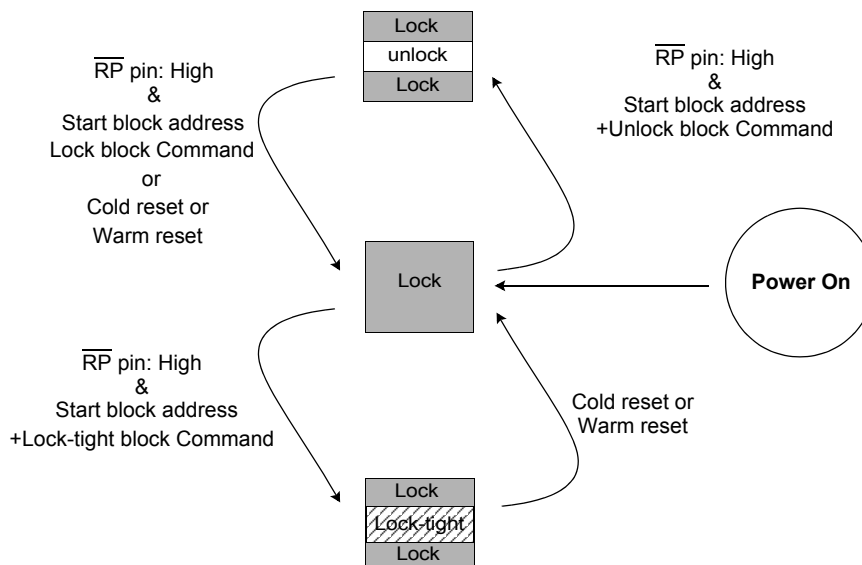
A block that is in a locked-tight state can only be changed to lock state after a Cold or Warm Reset. Unlock and Lock command sequences will not affect its state. This is an added level of write protection security.

A block must first be set to a locked state before it can be changed to locked-tight using the Lock-tight command. locked-tight blocks will revert to a locked state following a Cold or Warm Reset.

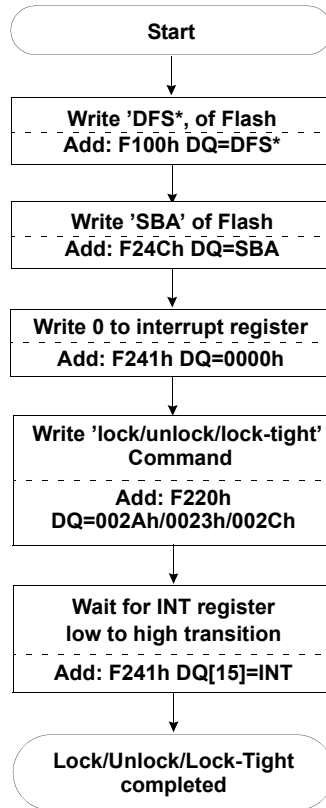


Lock-Tight Command Sequence:
 Start block address+Lock-tight block command (002Ch)

3.4.4 NAND Flash Array Write Protection State Diagram



Data Protection Operation Flow Diagram



* DFS is for DDP

Note) Samsung strongly recommends to follow the above flow chart

3.5 Data Protection During Power Down Operation

See Timing Diagram 6.15

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 POR level, about 1.3V. It is recommended that the \overline{RP} pin, which provides hardware protection, should be kept at VIL before power-down.

3.6 Load Operation

See Timing Diagram 6.8

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

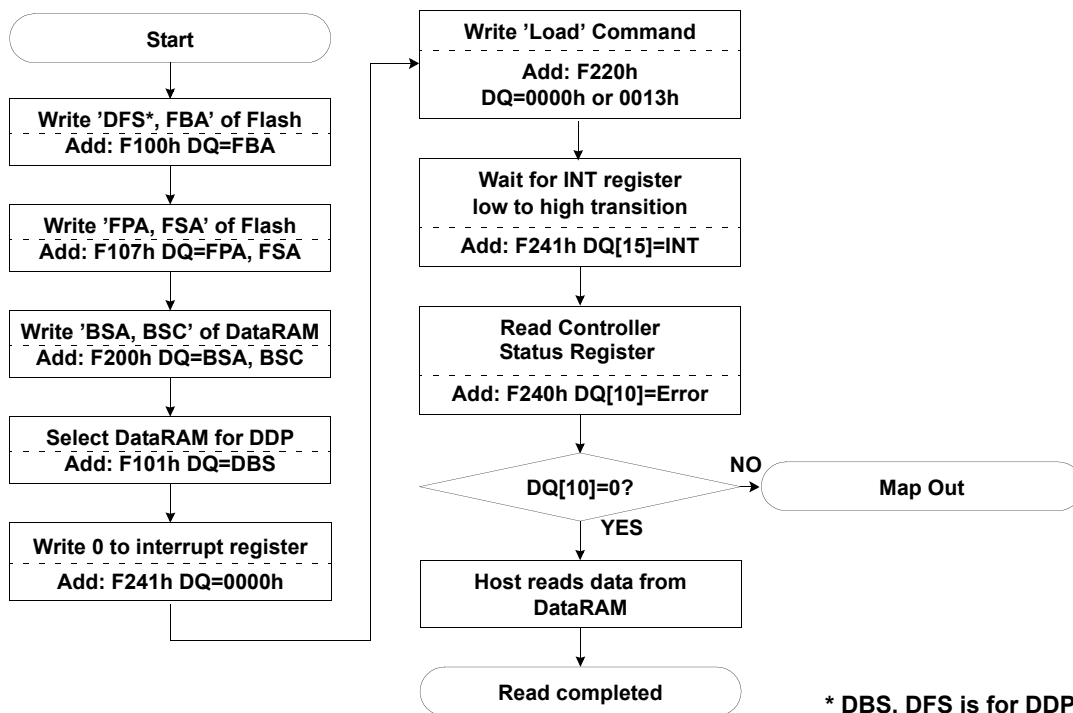
During a Load operation, the device:

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

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

The device has a dual data buffer memory architecture (DataRAM0, DataRAM1), each 2KB in size. Each DataRAM buffer has 4 Sectors. The device is capable of independent and simultaneous data-read operation from one data buffer and data-load operation to the other data buffer. Refer to the information for more details in section 3.12.1, "Read-While-Load Operation".

Load Operation Flow Chart Diagram



3.7 Read Operation

See Timing Diagrams 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6

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

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

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

3.7.1 Asynchronous Read Mode Operation (RM=0)

See Timing Diagrams 6.3, 6.4, 6.5 and 6.6

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

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

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

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

3.7.2 Synchronous Read Mode Operation (RM=1)

See Timing Diagrams 6.1 and 6.2

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

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

Burst Address Sequences

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

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

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

The BRL registers can be read during a burst read mode by using the \overline{AVD} signal with an address.

3.7.2.1 Continuous Linear Burst Read Operation

See Timing Diagram 6.2

First Clock Cycle

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

Subsequent Clock Cycles

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

Terminating Burst Read

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

Synchronous Read Boundary

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

* Reserved area is not available on Synchronous read

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

See Timing Diagram 6.1

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

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

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

3.7.2.3 Programmable Burst Read Latency Operation

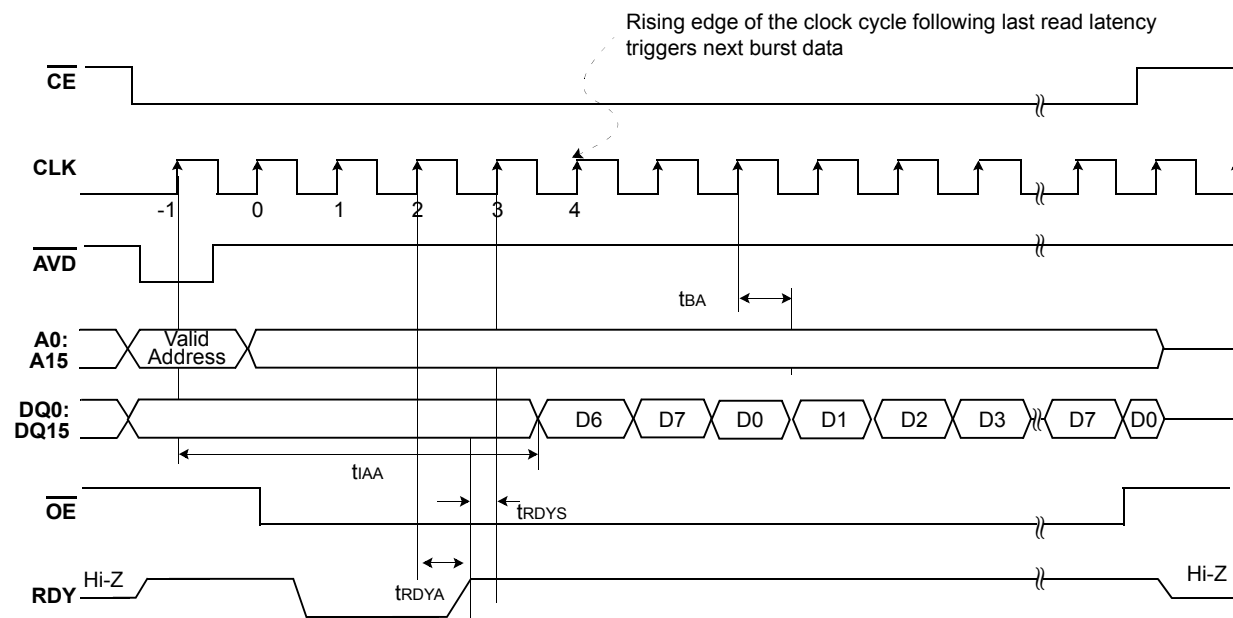
See Timing Diagrams 6.1 and 6.2

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

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

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

Four Clock Burst Read Latency (default condition)



3.7.3 Handshaking Operation

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

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

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

3.7.4 Output Disable Mode Operation

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

3.8 Program Operation

See Timing Diagram 6.9

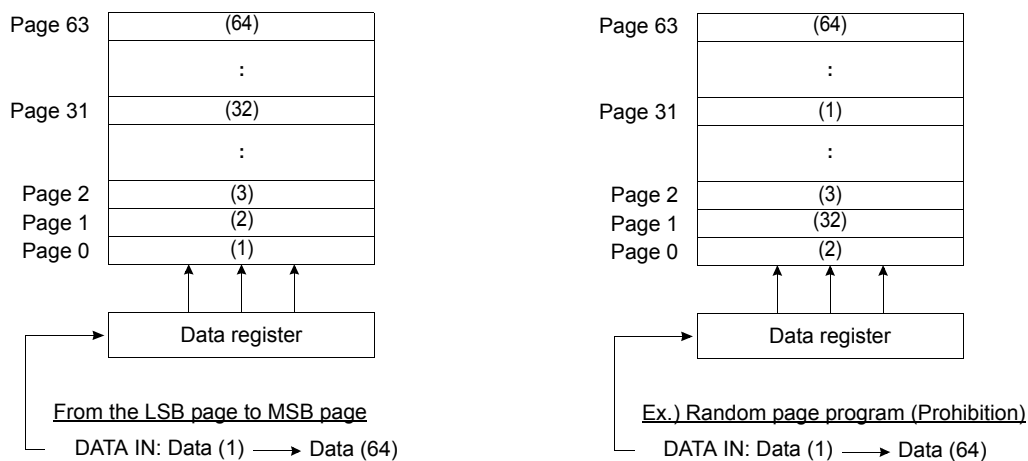
The Program operation is used to program data from the on-chip BufferRAMs into the NAND FLASH memory array.

The device has two 2KB data buffers, each 1 Page (2KB + 64B) in size. Each page has 4 sectors of 512B each main area and 16B spare area. The device can be programmed in units of 1~4 sectors.

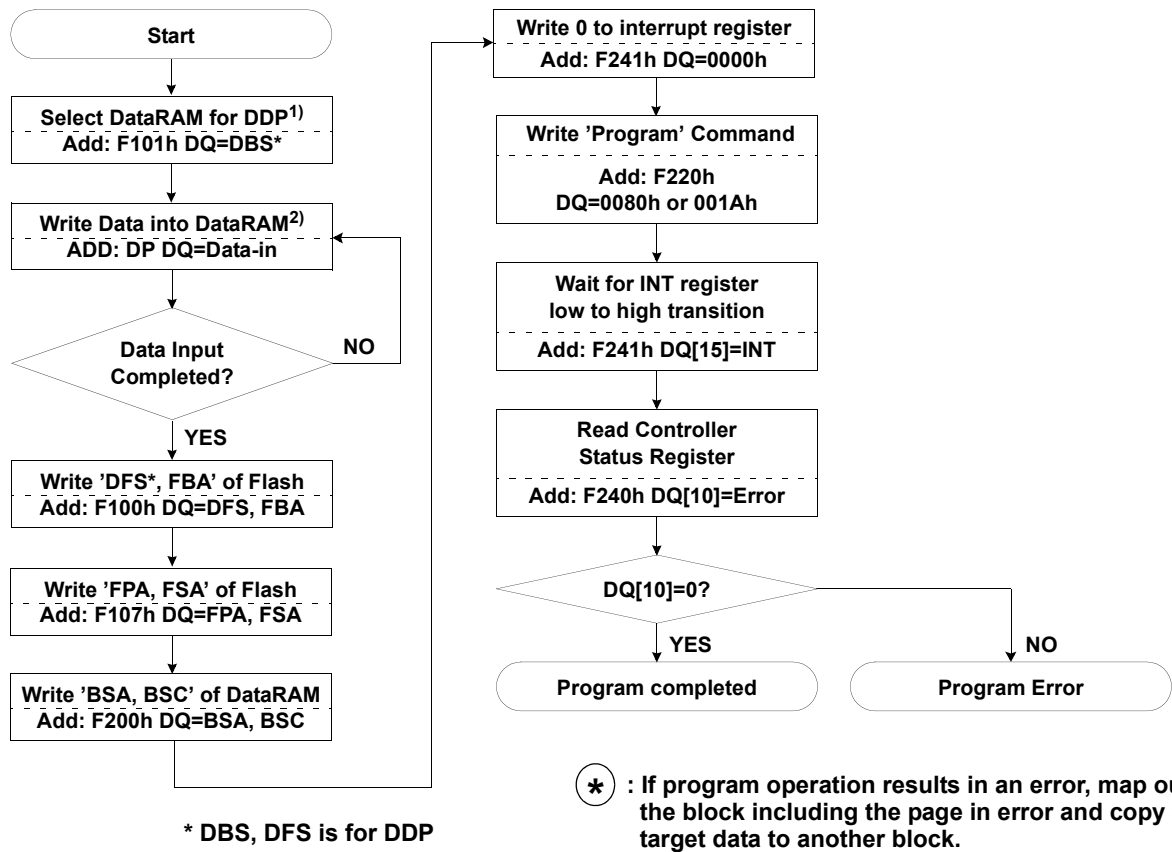
The architecture of the DataRAMs permits a simultaneous data-write operation from the Host to one of data buffers and a program operation from the other data buffer to the NAND Flash Array memory. Refer to Section 3.12.2, "Write While Program Operation", for more information.

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.



Program Operation Flow Diagram



- Note 1) This must happen before data input
 2) Data input could be done anywhere between "Start" and "Write Program Command".

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

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

Data input from the Host to the DataRAM can be done at any time during the Internal Program Routine after "Start" but before the "Write Program Command" is written.

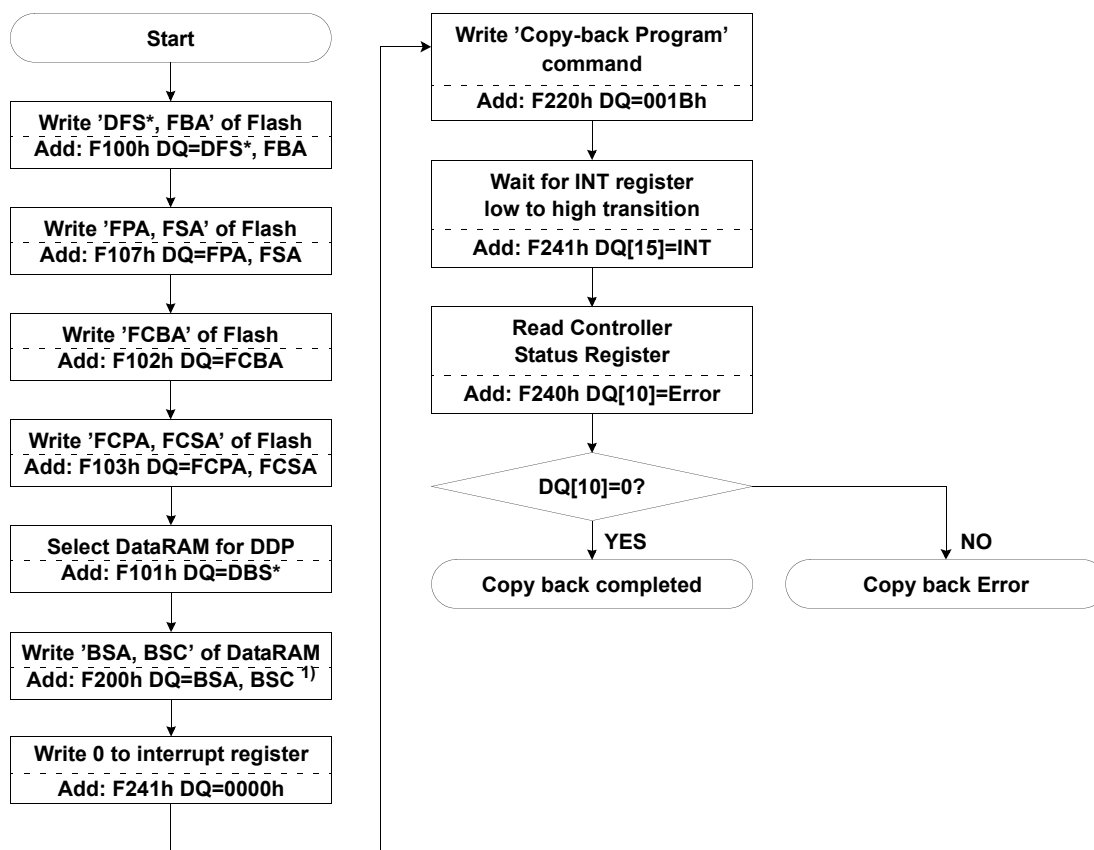
3.9 Copy-Back Program Operation

The Copy-Back program is configured to quickly rewrite data stored in one page without utilizing memory other than OneNAND. 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 block is updated and the rest of the block also need to be copied to the newly assigned free block.

Data from the source page is saved in one of the on-chip DataRAM buffers and then programmed directly into the destination page. The DataRAM overwrites the previous data using the Buffer Sector Address (BSA) and Buffer Sector Count (BSC).

The Copy-Back Program Operation does this by performing sequential page-reads without a serial access and executing a copy-program using the address of the destination page.

Copy-Back Program Operation Flow Chart



* : 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.
 2) FBA, FPA and FSA should be input prior to FCBA, FCPA and FCSA.

The Copy-Back steps shown in the flow chart are:

- Data is read from the NAND Array using Flash Block Address (FBA), Flash Page Address (FPA) and Flash Sector Address (FSA). FBA, FPA, and FSA identify the source address to read data from NAND Flash array.
- The BufferRAM Sector Count (BSC) and BufferRAM Sector Address (BSA) identifies how many sectors and the location of the sectors in DataRAM that are used.
- The destination address in the NAND Array is written using the Flash Copy-Back Block Address (FCBA), Flash Copy-Back Page Address (FCPA), and Flash Copy-Back Sector Address (FCSA).
- The Copy-Back Program command is issued to start programming.
- Upon completion of copy-back programming to the destination page address, the Host checks the status to see if the operation was successfully completed. If there was an error, map out the block including the page in error and copy the target data to another block.

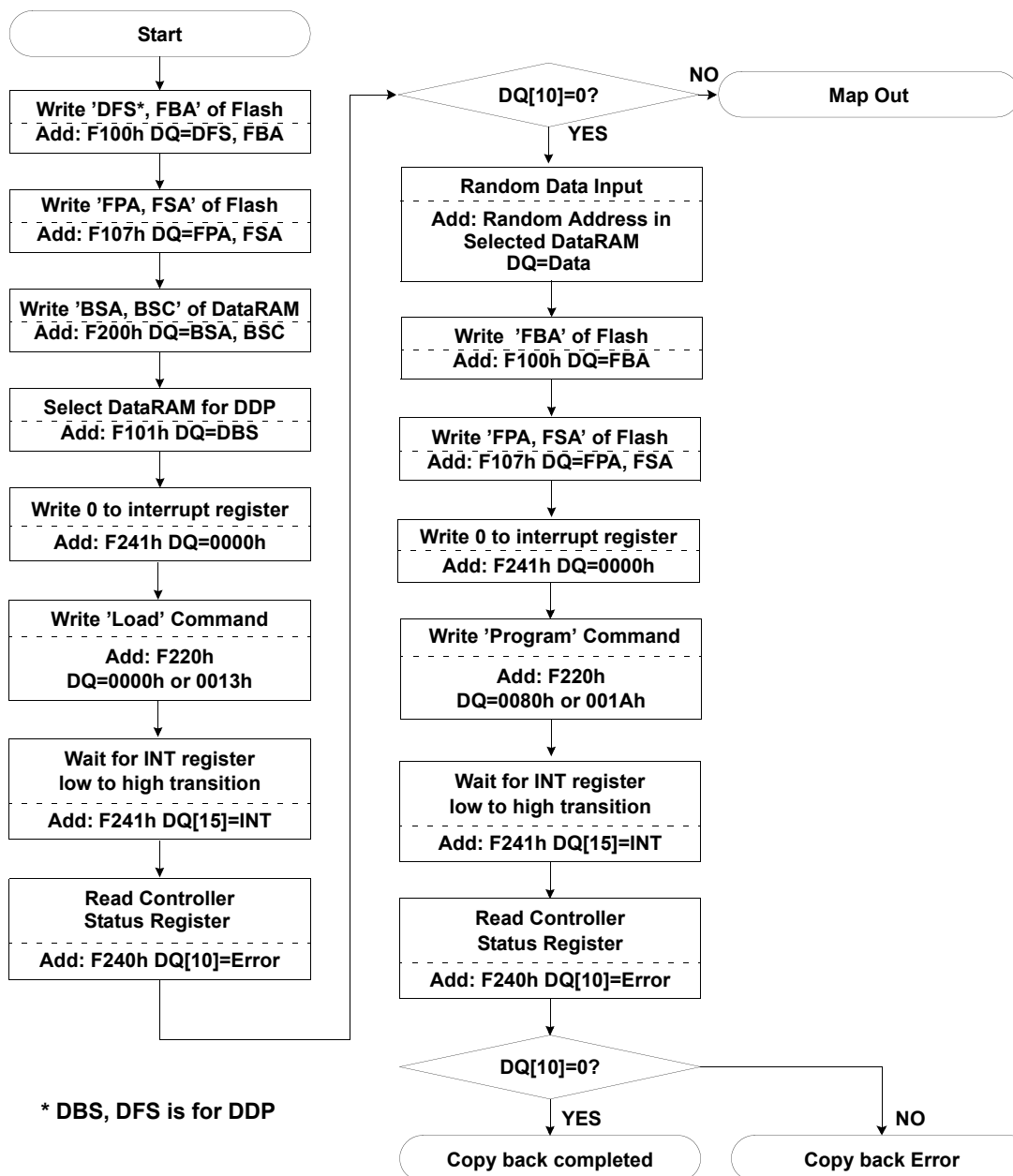
3.9.1 Copy-Back Program Operation with Random Data Input

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.

Copy-Back Program Operation with Random Data Input Flow Chart



3.10 Erase Operation

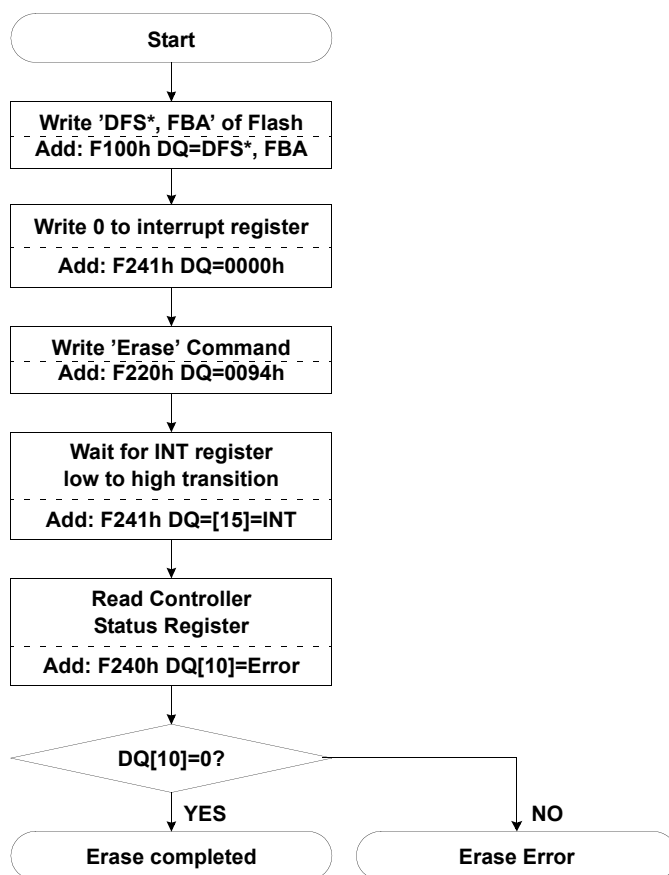
There are multiple methods for erasing data in the device including Block Erase and Multi-Block Erase.

3.10.1 Block Erase Operation

See Timing Diagram 6.10

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

Block Erase Operation Flow Chart



DFS* is for DDP

⊛ : If erase operation results in an error, map out the failing block and replace it with another block.

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

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

A reset during an erase operation will cause data corruption at the corresponding location.

3.10.2 Multi-Block Erase Operation

See Timing Diagram 6.10

Using Multi-Block Erase, the device can erase up to 64 multiple blocks simultaneously.

Multiple blocks can be erased by issuing a Multi-Block Erase command and writing the block address of the memory location to be erased. The final Flash Block Address (FBA) and Block Erase command initiate the internal multi block erase routine. During a Multi-Block Erase, the OnGo bit of the Controller Status Register is set to '1'(busy) from the time first block address to be latched is written until the actual erase has finished.

During block address latch sequence, issuing of other commands except Block Erase and Multi Block Erase at INT=High will abort the current operation. So to speak, It will cancel the previously latched addresses of Multi Block Erase Operation. On the other hand, Other command issue at INT=low will be ignored.

A reset during an erase operation will cause data corruption at the address location being operated on during the reset.

Despite a failed block during Multi-Block Erase operation, the device will continue the erase operation until all other specified blocks are erased.

Erase Suspend Command issue during Multi Block Erase Address latch sequence is prohibited.

Locked Blocks

If there are locked blocks in the specified range, the Multi-Block Erase operation works as the follows.

Case 1: All specified blocks except BA(2) will be erased.

[BA(1)+0095h] + **[BA((2), locked)+0095h]** + ... + [BA(N-1)+0095h] + [BA(N)+0094h]

Case 2: Multi-Block Erase Operation is suspended and fails to start if the last Block Erase command is put together with the locked block address until right command and address input are issued.

[BA(1)+0095h] + [BA(2)+0095h] + ... + [BA(N-1)+0095h] + **[BA((N), locked)+0094h]**

Case 3: All specified blocks except BA(N) are erased.

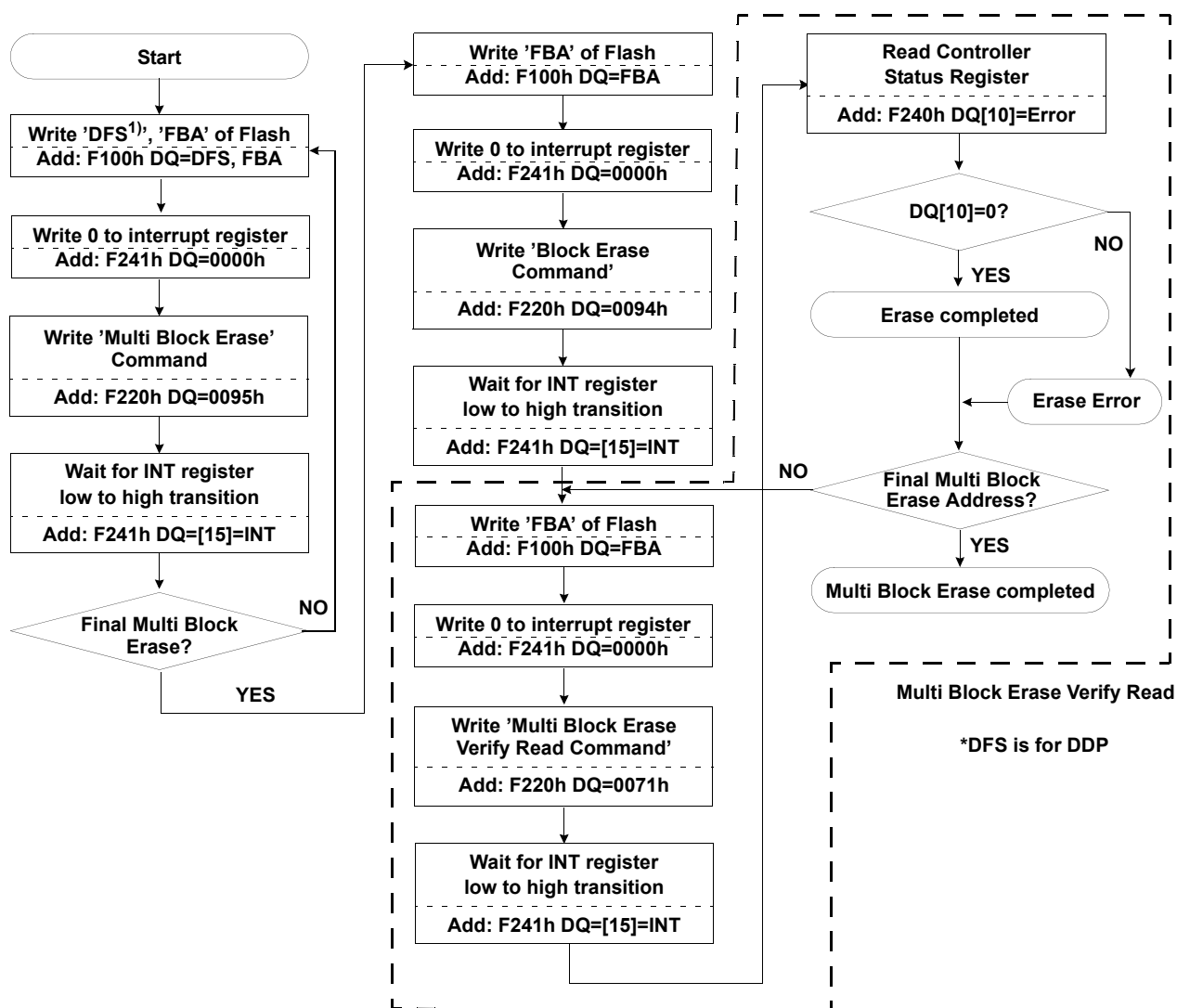
[BA(1)+0095h] + [BA(2)+0095h] + ... + [BA(N-1)+0095h] + **[BA((N), locked)+0094h]** + [BA(N+1)+0094h]

3.10.3 Multi-Block Erase Verify Read Operation

After a Multi-Block Erase Operation, verify Erase Operation result of each block with Multi-Block Erase Verify Command combined with address of each block.

If a failed address is identified, it must be managed in firmware.

Multi Block Erase/ Multi Block Erase Verify Read Flow Chart



Note 1) DFS should be a fixed value, for Multi Block Erase is performed within a single chip.

3.10.4 Erase Suspend / Erase Resume Operation

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

Erase Suspend During a Block Erase Operation

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

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

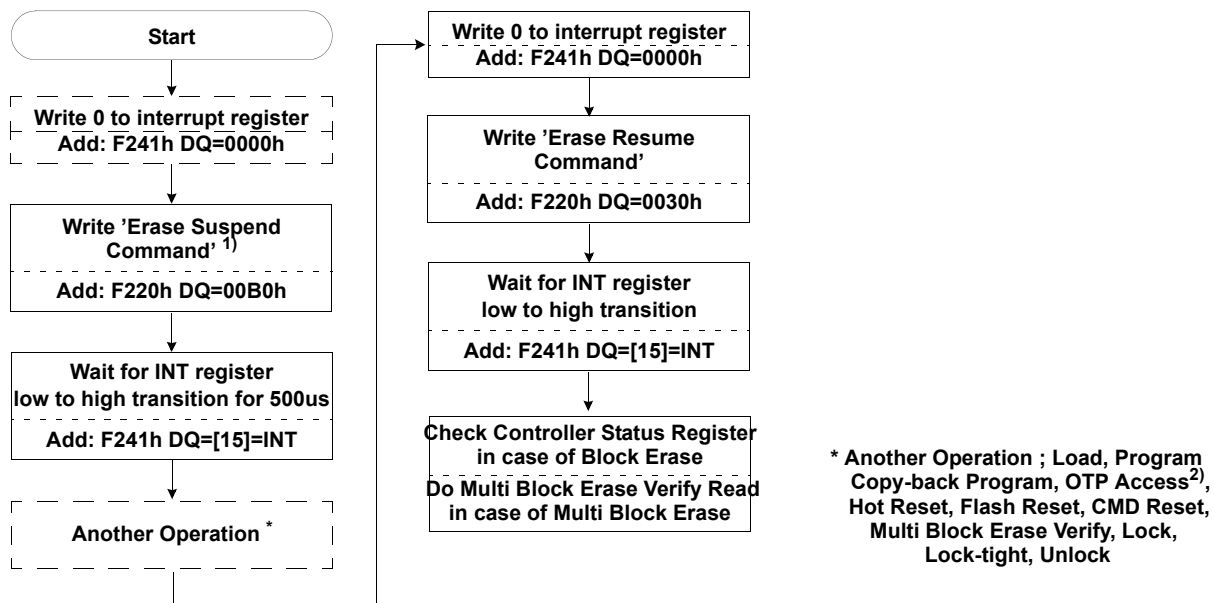
The subsequent operation can be to any block that was NOT being erased.

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

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

- Block Erase/Multi-Block Erase/Erase Suspend

Erase Suspend and Erase Resume Operation Flow Chart



Note 1) Erase Suspend command input is prohibited during Multi Block Erase address latch period.

2) If OTP access mode exit happens with Reset operation during Erase Suspend mode, Reset operation could hurt the erase operation. So if a user wants to exit from OTP access mode without the erase operation stop, Reset NAND Flash Core command should be used.

Erase Resume

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

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

3.11 OTP Operation

One Block of the NAND Flash Array memory is reserved as a One-Time Programmable Block memory area.

The OTP block can be read, programmed and locked using the same operations as any other NAND Flash Array memory block. OTP block cannot be erased.

OTP block is fully-guaranteed to be a valid block.

Entering the OTP Block

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

Exiting the OTP Block

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

Exiting the OTP Block during an Erase Operation

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

The OTP Block Page Assignments

OTP area is one block size (128KB+4KB, 64 Pages) and is divided into two areas. The 10-page User Area is available as an OTP storage area. The 54-page Manufacturer Area is programmed by the manufacturer prior to shipping the device to the user.

OTP Block Page Allocation Information

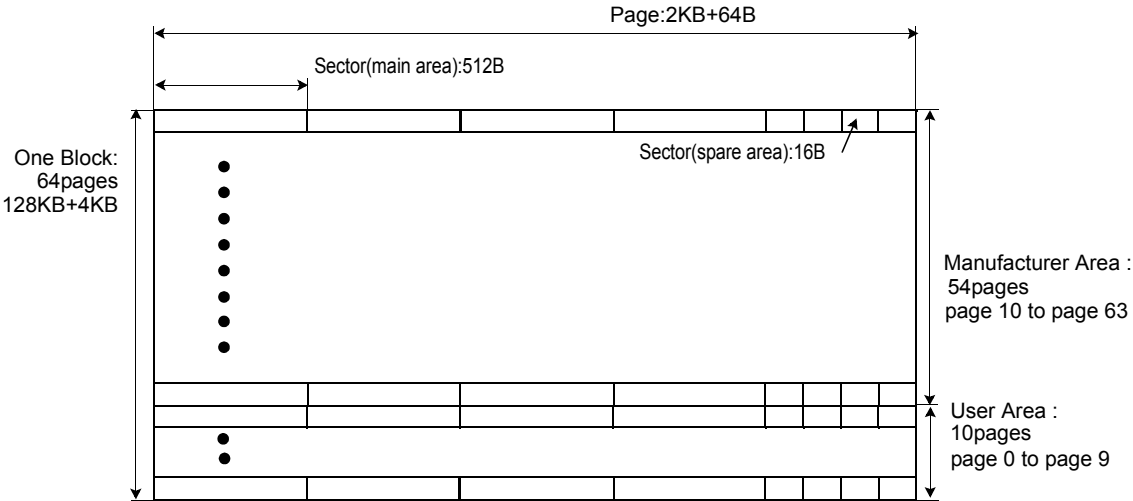
| Area | Page | Use |
|--------------|--------------------|---------------------------------|
| User | 0 ~ 9 (10 pages) | Designated as user area |
| Manufacturer | 10 ~ 63 (54 pages) | Used by the device manufacturer |

Note) For DDP and QDP products, One block OTP is available in each chip. i.e. 2 OTP blocks in DDP, 4 OTP blocks in QDP

OneNAND4G(KFW4G16Q2M-DEB6)
 OneNAND2G(KFH2G16Q2M-DEB6)
 OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

OTP Area Structure



3.11.1 OTP Load Operation

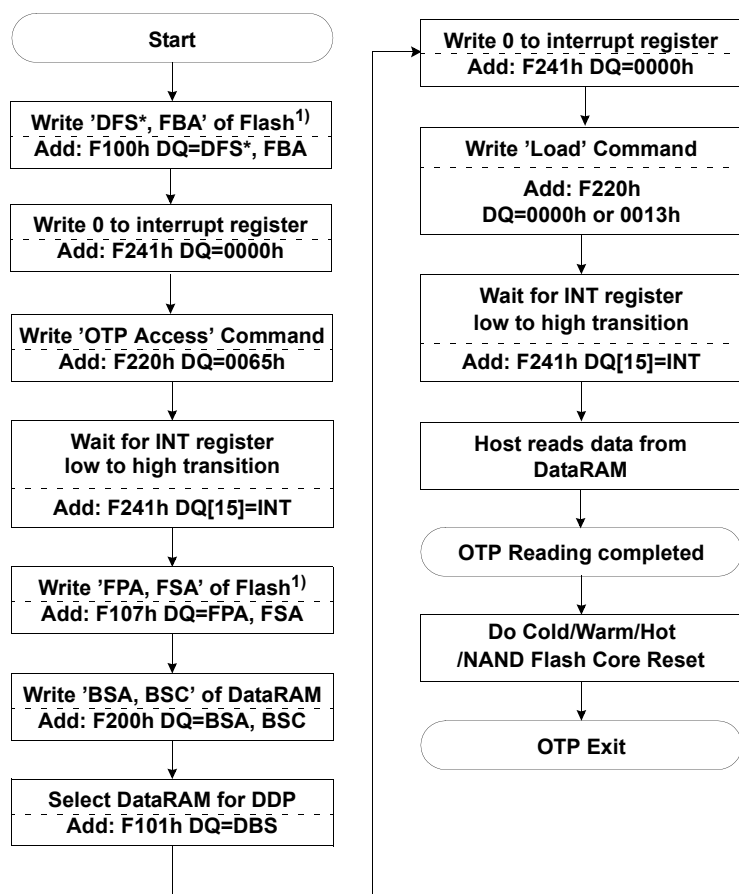
An OTP Load Operation accesses the OTP area and transfers identified content from the OTP to the DataRAM on-chip buffer, thus making the OTP contents available to the Host.

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

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

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

OTP Read Operation Flow Chart



* DBS, DFS is for DDP

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

3.11.2 OTP Program Operation

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

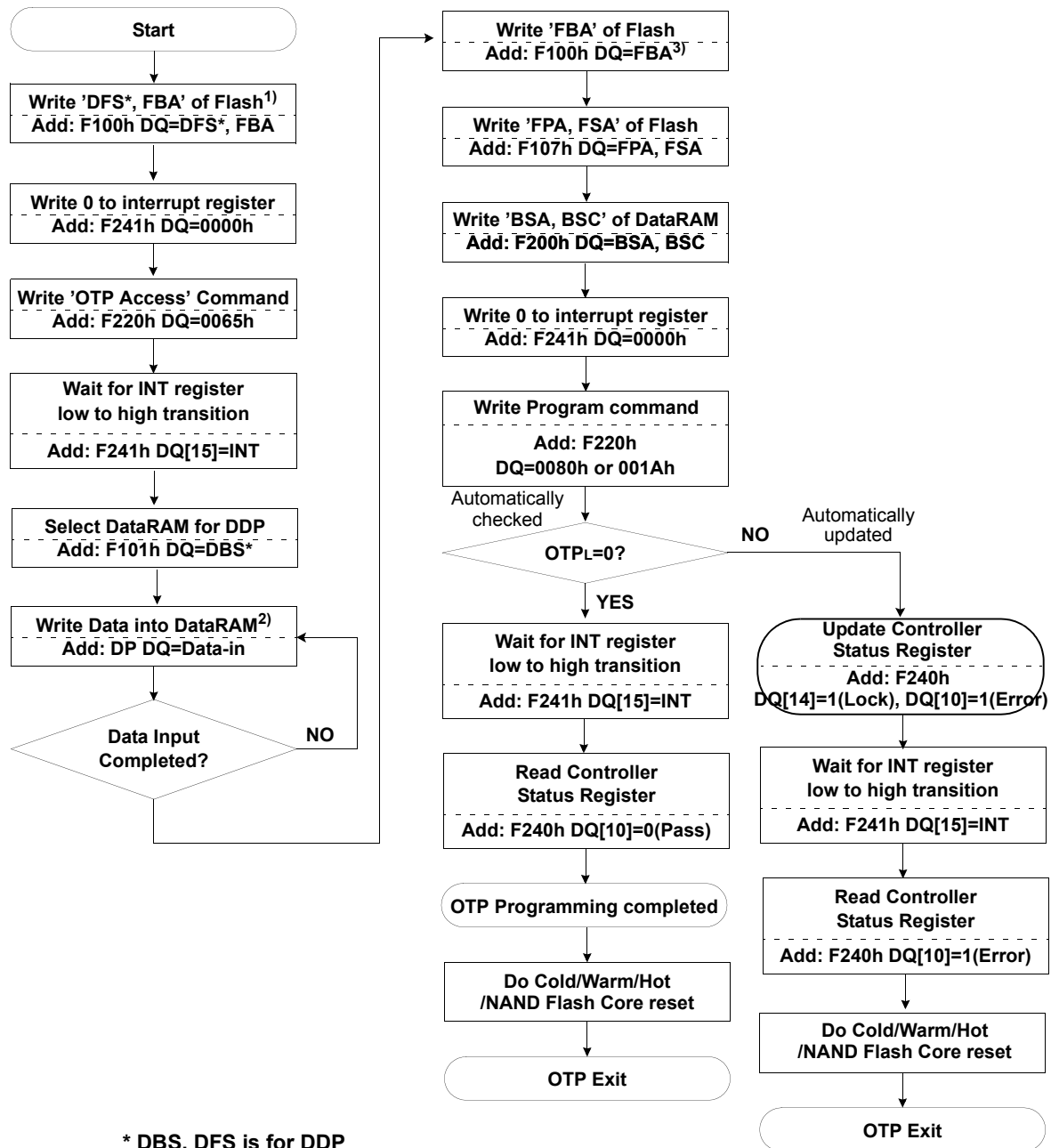
A memory location in the OTP area can be programmed only one time (no erase operation permitted).

The OTP area is programmed using the same sequence as normal program operation after being accessed by the command (see section 3.8 for more information).

Programming the OTP Area

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

OTP Program Operation Flow Chart



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

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

3) FBA should point the unlocked area address among NAND Flash Array address map.

3.11.3 OTP Lock Operation

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

Unlike the main area of the NAND Flash Array memory, **once the OTP block is locked, it cannot be unlocked.**

Locking the OTP

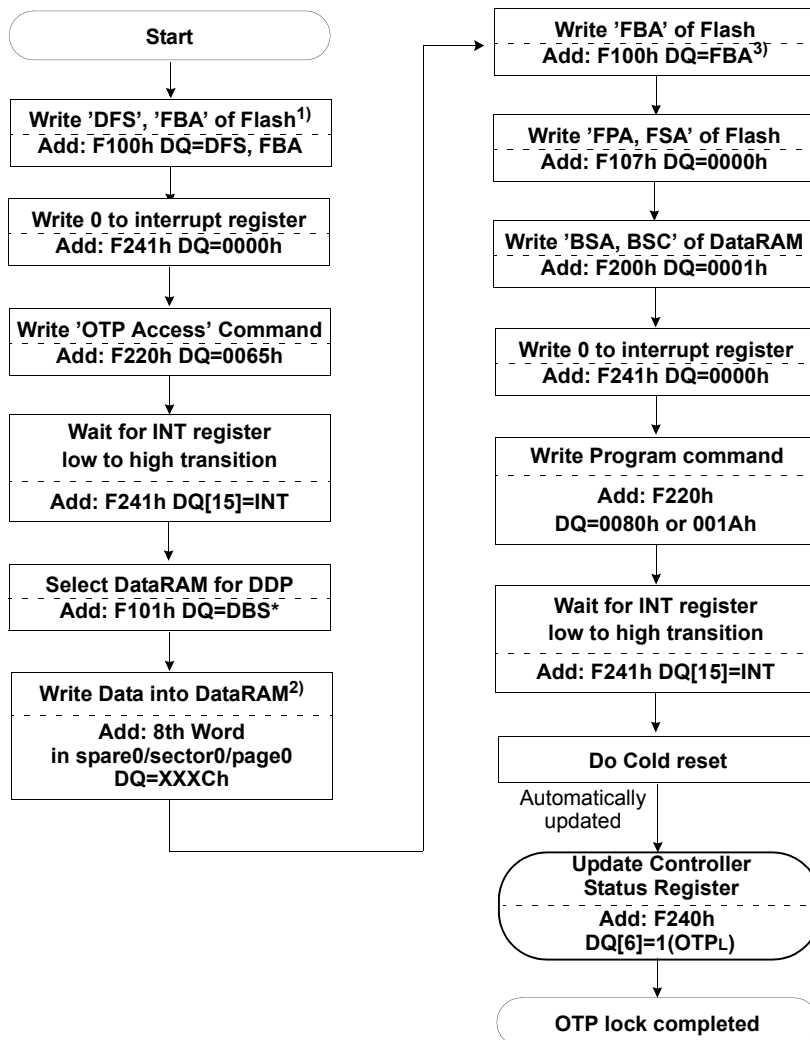
Programming to the OTP area can be prevented by locking the OTP area. Locking the OTP area is accomplished by programming XXXCh to 8th word of sector0 of page0 of the spare0 memory area.

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

OTP Lock Operation Steps

- Issue the OTP Access Command
- Fill data to be programmed into DataRAM (data can be input at anytime between the "Start" and "Write Program" commands)
- Write 'XXXCH' data into the 8th word of sector0 of page0 of the spare0 memory area of the DataRAM.
- Issue a Flash Block Address (FBA) which is unlocked area address of NAND Flash Array address map.
- Issue a Program command to program the data from the DataRAM into the OTP
- When the OTP lock is complete, do a Cold Reset to exit the OTP Access mode and update OTP lock bit[6].
- OTP lock bit[6] of the Controller Status Register will be set to "1" and the OTP will be locked.

OTP Lock Operation Flow Chart



* DBS, DFS is for DDP

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

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

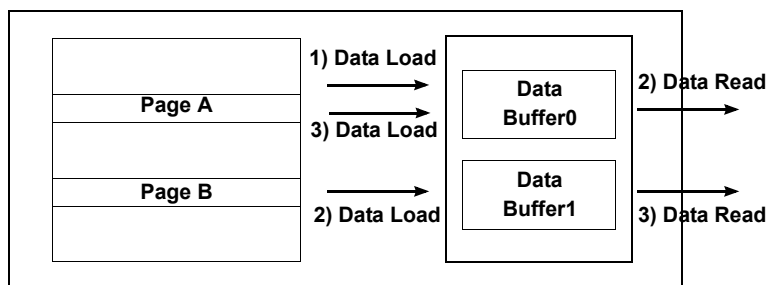
3) FBA should point the unlocked area address among NAND Flash Array address map.

3.12 Dual Operations

The device has independent dual data buffers on-chip (except during the Boot Load period) that enables higher performance read and program operation.

3.12.1 Read-While-Load Operation

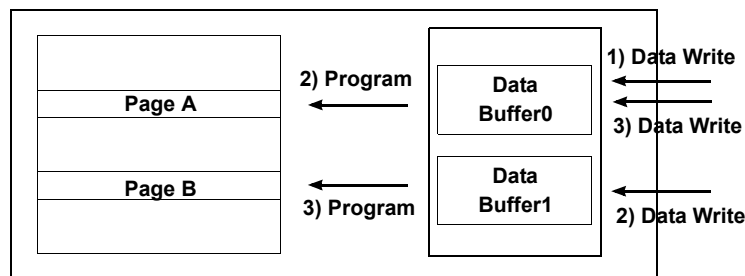
This operation accelerates the read performance of the device by enabling data to be read out by the host from one DataRAM buffer while the other DataRAM buffer is being loaded with data from the NAND Flash Array memory.



The dual data buffer architecture provides the capability of executing a data-read operation from one of DataRAM buffers during a simultaneous data-load operation from Flash to the other buffer. Simultaneous load and read operation to same data buffer is prohibited. See sections 3.6 and 3.7 for more information on Load and Read Operations. If host sets FBA, FSA, or FPA while loading into designated page, it will fail the internal load operation. Address registers should not be updated until internal operation is completed.

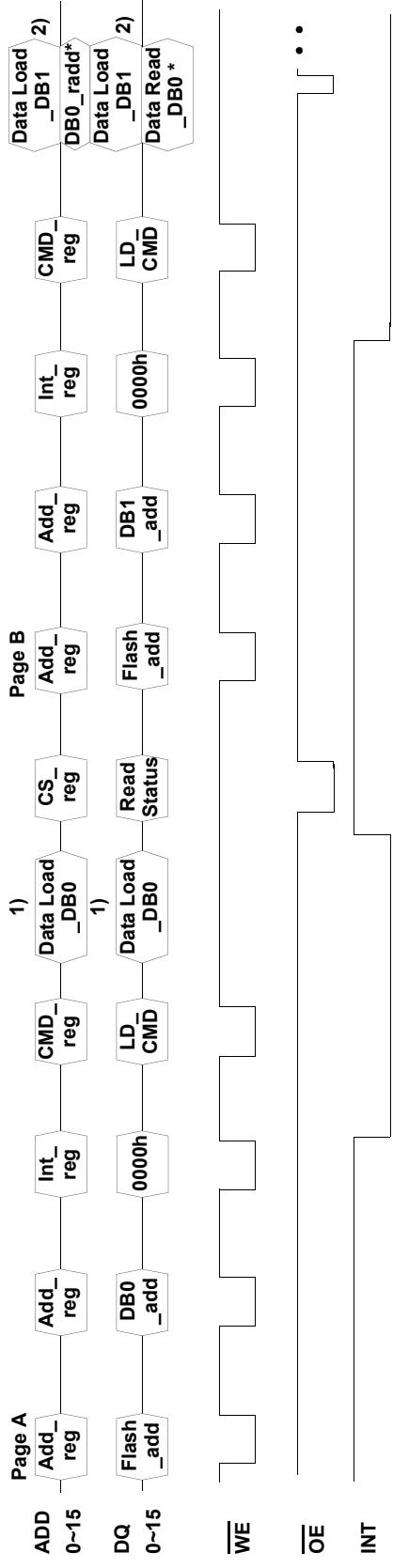
3.12.2 Write-While-Program Operation

This operation accelerates the programming performance of the device by enabling data to be written by the host into one DataRAM buffer while the NAND Flash Array memory is being programmed with data from the other DataRAM buffer.



The dual data buffer architecture provides the capability of executing a data-write operation to one of DataRAM buffers during simultaneous data-program operation to Flash from the other buffer. Simultaneous program and write operation to same data buffer is prohibited. See sections 3.8 for more information on Program Operation. If host sets FBA, FSA, or FPA while programming into designated page, it will fail the internal program operation. Address registers should not be updated until internal operation is completed.

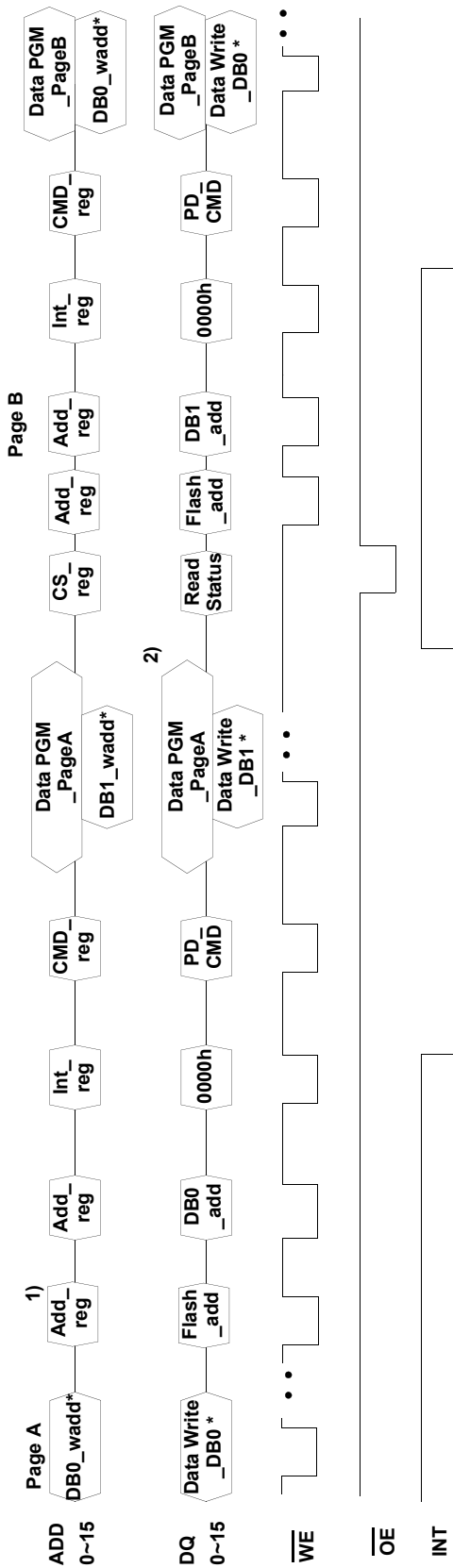
Read While Load Diagram



* DBS should be set before accessing DataRAM for DDP

- Int_reg : Interrupt Register Address
- Add_reg : Address Register Address
- Flash_add : Flash Address to be loaded
- DBn_add : DataRAM Address to be loaded
- CMD_reg : Command Register Address
- LD_CMD : Load Command
- Data Load_DBn : Load Data from NAND Flash Array to DataRAMn
- CS_reg : Controller Status Register Address
- Data Read_DBn : Read Data from DBn
- DBn_radd : DataRAM Address to be read

Write While Program Diagram



* DBS should be set before accessing DataRAM for DDP

Add_reg : Address Register Address
 DBn_add : 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

3.13 ECC Operation

The OneNAND device has on-chip ECC with the capability of detecting 2 bit errors and correcting 1-bit errors in the NAND Flash Array memory main and spare areas.

As the device transfers data from a BufferRAM to the NAND Flash Array memory Page Buffer for Program Operation, the device initiates a background operation which generates an Error Correction Code (ECC) of 24bits for each sector main area data and 10bits for 2nd and 3rd word data of each sector spare area.

During a Load operation from the NAND Flash Array memory Page, the on-chip ECC engine generates a new ECC. The 'Load ECC result' is compared to the originally 'Program ECC' thus detecting the number and position of errors. Single-bit error is corrected.

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

Error types are divided into 'no error', '1bit correctable error', and '2bit error uncorrectable error'.

OneNAND supports 2bit EDC even though 2bit error seldom or never occurs. Hence, it is not recommended for Host to read 'ECC Status Register' for checking ECC error because the built-in Error Correction Logic of OneNAND automatically corrects ECC error.

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

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

3.13.1 ECC Bypass Operation

In an ECC bypass operation, the device does not generate ECC as a background operation. The result does not indicate error position (refer to the ECC Result Table).

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

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

ECC Code and ECC Result by ECC Operation

| Operation | Program operation | Load operation | | |
|---------------|--|--|---|-------------|
| | ECC Code Update to NAND Flash Array Spare Area | ECC Code at BufferRAM Spare Area | ECC Status & Result Update to Registers | 1bit Error |
| ECC operation | Update | Pre-written ECC code ⁽¹⁾ 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.

3.14 Invalid Block Operation

Invalid blocks are defined as blocks in the device's NAND Flash Array memory that contain one or more invalid bits whose reliability is not guaranteed by Samsung.

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

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

The system 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 always fully guaranteed to be a valid block.

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

3.14.1 Invalid Block Identification Table Operation

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

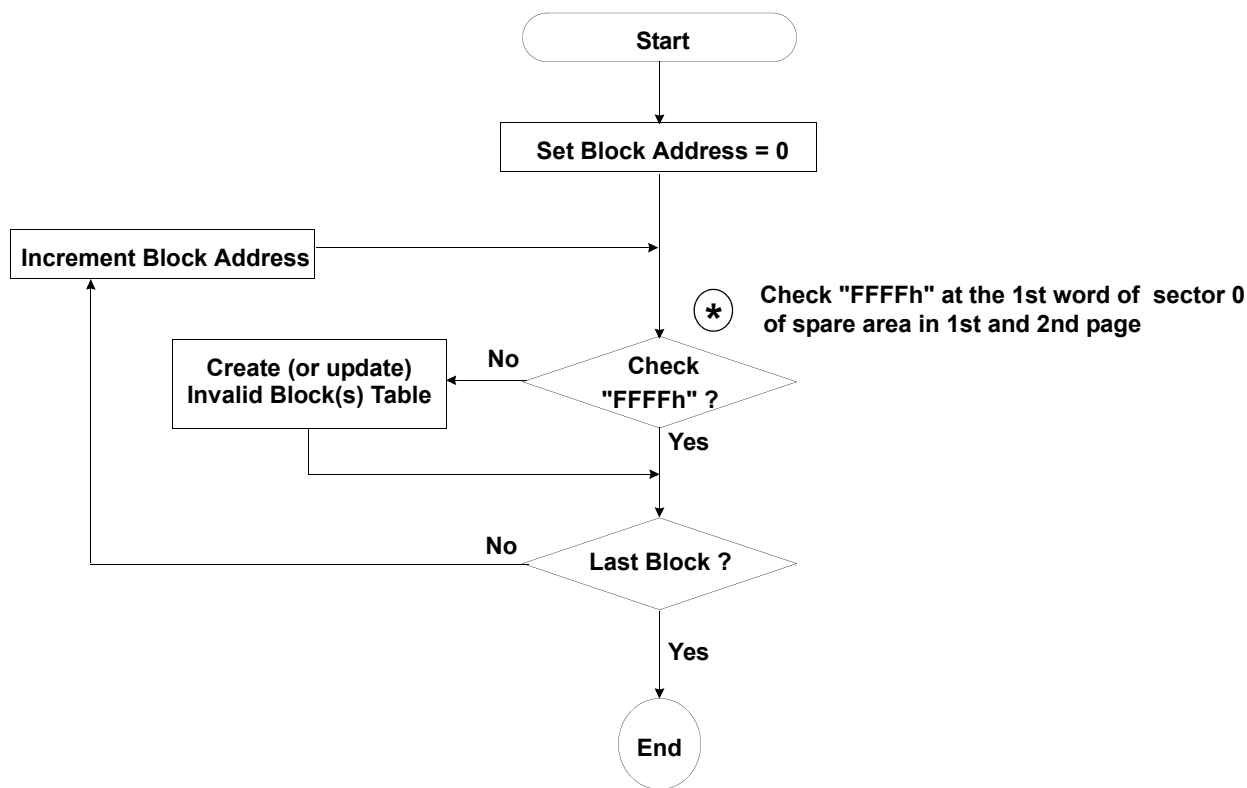
Invalid blocks are identified by erasing all address locations in the NAND Flash Array memory except locations where the invalid block(s) information is written prior to shipping.

An invalid block(s) status is defined by the 1st word in the spare area. Samsung makes sure that either the 1st or 2nd page of every invalid block has non-FFFFh data at the 1st word of sector0.

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

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

Invalid Block Table Creation Flow Chart



3.14.2 Invalid Block Replacement Operation

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

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

In the case of a status read failure after erase or program, a block replacement should be done. Because program status failure during a page program does not affect the data of the other pages in the same block, a 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.

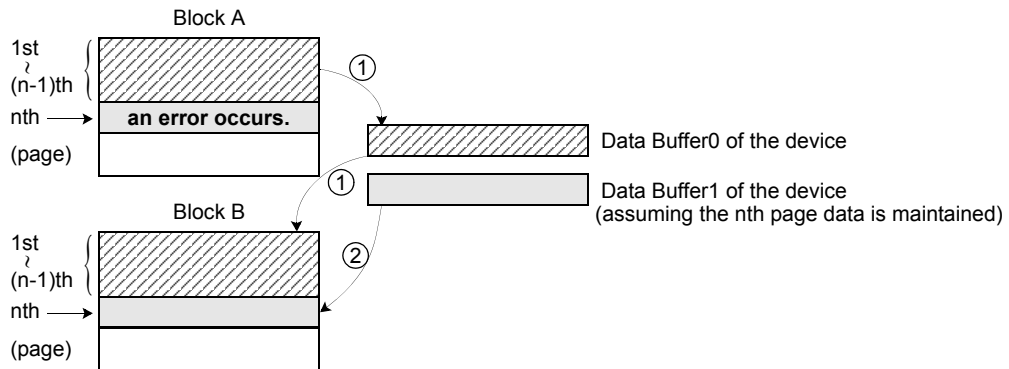
Block Failure Modes and Countermeasures

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

Referring to the diagram for further illustration, when an error happens in the nth page of block 'A' during program operation, copy the data in the 1st ~ (n-1)th page to the same location of block 'B' via data buffer0.

Then copy the nth page data of block 'A' in the data buffer1 to the nth page of block 'B' or any free block. Do not further erase or program block 'A' but instead complete the operation by creating an 'Invalid Block Table' or other appropriate scheme.

Block Replacement Operation Sequence



4.0 DC CHARACTERISTICS

4.1 Absolute Maximum Ratings

| Parameter | | Symbol | Rating | Unit |
|------------------------------------|----------|---------------------------------|----------------|------|
| Voltage on any pin relative to Vss | Vcc | Vcc | -0.5 to + 2.45 | V |
| | All Pins | V _{IN} | -0.5 to + 2.45 | |
| Temperature Under Bias | Extended | T _{bias} | -30 to +125 | °C |
| Storage Temperature | | T _{stg} | -65 to +150 | °C |
| Short Circuit Output Current | | I _{os} | 5 | mA |
| Recommended Operating Temperature | | T _A (Extended Temp.) | -30 to +85 | °C |

NOTES:

1. 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.
2. Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to the conditions detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

4.2 Operating Conditions

Voltage reference to GND

| Parameter | Symbol | 1.8V Device | | | Unit |
|----------------|----------------|-------------|------|------|------|
| | | Min | Typ. | Max | |
| Supply Voltage | Vcc-core / Vcc | 1.7 | 1.8 | 1.95 | V |
| | Vcc-IO / Vccq | | | | |
| | Vss | 0 | 0 | 0 | |

NOTES:

1. The system power should reach 1.7V after POR triggering level(typ. 1.5V) within 400us.
2. Vcc-Core (or Vcc) should reach the operating voltage level prior to or at the same time as Vcc-IO (or Vccq).

4.3 DC Characteristics

| Parameter | Symbol | Test Conditions | 1.8V device | | | Unit | |
|---|------------------|---|-----------------------|-------|-----------------------|-------|----|
| | | | Min | Typ | Max | | |
| Input Leakage Current | I _{LI} | V _{IN} =V _{SS} to V _{CC} , V _{CC} =V _{CCmax} | Single | - 1.0 | - | + 1.0 | μA |
| | | | DDP | -2.0 | - | + 2.0 | |
| | | | QDP | -4.0 | - | + 4.0 | |
| Output Leakage Current | I _{LO} | V _{OUT} =V _{SS} to V _{CC} , V _{CC} =V _{CCmax} , CE or OE=V _{IH} (Note 1) | Single | - 1.0 | - | + 1.0 | μA |
| | | | DDP | -2.0 | - | + 2.0 | |
| | | | QDP | -4.0 | - | + 4.0 | |
| Active Asynchronous Read Current (Note 2) | I _{CC1} | CE=V _{IL} , OE=V _{IH} | - | 8 | 15 | mA | |
| Active Burst Read Current (Note 2) | I _{CC2} | CE=V _{IL} , OE=V _{IH} | 66Mhz | - | 15 | 25 | mA |
| | | | 1MHz | - | 3 | 4 | mA |
| | | | 66Mhz (DDP/QDP) | - | 20 | 25 | mA |
| | | | 1MHz (DDP/QDP) | - | 3 | 4 | mA |
| Active Write Current (Note 2) | I _{CC3} | CE=V _{IL} , OE=V _{IH} | Single | - | 8 | 15 | mA |
| | | | DDP | - | 13 | 20 | mA |
| Active Load Current (Note 3) | I _{CC4} | CE=V _{IL} , OE=V _{IH} , WE=V _{IH} | - | 30 | 40 | mA | |
| Active Program Current (Note 3) | I _{CC5} | CE=V _{IL} , OE=V _{IH} , WE=V _{IH} | - | 25 | 30 | mA | |
| Active Erase Current (Note 3) | I _{CC6} | CE=V _{IL} , OE=V _{IH} , WE=V _{IH} | - | 20 | 25 | mA | |
| Multi Block Erase Current (Note 3) | I _{CC7} | CE=V _{IL} , OE=V _{IH} , WE=V _{IH} , 64blocks | - | 20 | 25 | mA | |
| Standby Current | I _{SB} | CE = RP=V _{CC} ± 0.2V | Single | - | 10 | 50 | μA |
| | | | DDP | - | 20 | 100 | |
| | | | QDP | - | 40 | 180 | |
| Input Low Voltage | V _{IL} | - | -0.5 | - | 0.4 | V | |
| Input High Voltage (Note4) | V _{IH} | - | V _{CCq} -0.4 | - | V _{CCq} +0.4 | V | |
| Output Low Voltage | V _{OL} | I _{OL} = 100 μA, V _{CC} =V _{CCmin} , V _{CCq} =V _{CCqmin} | - | - | 0.2 | V | |
| Output High Voltage | V _{OH} | I _{OH} = -100 μA, V _{CC} =V _{CCmin} , V _{CCq} =V _{CCqmin} | V _{CCq} -0.1 | - | - | V | |

Note 1. CE should be V_{IH} for RDY. IOBE should be '0' for INT.

Note 2. I_{CC} active for Host access

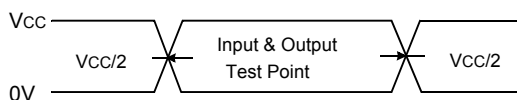
Note 3. I_{CC} active for Internal operation. (without host access)

Note 4. V_{CCq} is equivalent to V_{CC}-IO

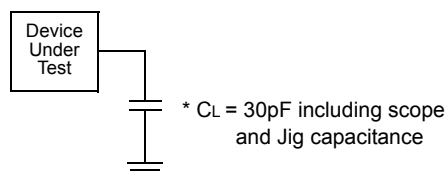
5.0 AC CHARACTERISTICS

5.1 AC Test Conditions

| Parameter | | Value |
|--------------------------------|--------------|-----------------------|
| Input Pulse Levels | | 0V to V _{CC} |
| Input Rise and Fall Times | CLK | 3ns |
| | other inputs | 5ns |
| Input and Output Timing Levels | | V _{CC} /2 |
| Output Load | | C _L = 30pF |



Input Pulse and Test Point



Output Load

5.2 Device Capacitance

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

| Item | Symbol | Test Condition | Single | | DDP | | QDP | | Unit |
|-------------------------|------------------|----------------------|--------|-----|-----|-----|-----|-----|------|
| | | | Min | Max | Min | Max | Min | Max | |
| Input Capacitance | C _{IN1} | V _{IN} =0V | - | 10 | - | 20 | - | 40 | pF |
| Control Pin Capacitance | C _{IN2} | V _{IN} =0V | - | 10 | - | 20 | - | 40 | |
| Output Capacitance | C _{OUT} | V _{OUT} =0V | - | 10 | - | 20 | - | 40 | |
| INT Capacitance | C _{INT} | V _{OUT} =0V | - | 15 | - | 30 | - | 60 | |

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

5.3 Valid Block Characteristics

| Parameter | Symbol | Min | Typ. | Max | Unit |
|-----------------------------|-----------------|-------|------|-------|--------|
| Valid Block Number (Single) | N _{VB} | 1004 | - | 1024 | Blocks |
| Valid Block Number (DDP) | N _{VB} | 2008 | - | 2048 | Blocks |
| Valid Block Number (QDP) | N _{VB} | 4016* | - | 4096* | Blocks |

NOTES:

- 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.
 - The 1st block, which is placed on 00h block address, is fully guaranteed to be a valid block.
- * Each KFG1G16Q2M chip in the KFH2G16Q2M has Maximum 20 invalid blocks.

5.4 AC Characteristics for Synchronous Burst Read

See Timing Diagrams 6.1 and 6.2

| Parameter | Symbol | KFG1G16Q2M/KFH2G16Q2M /KFW4G16Q2M | | Unit |
|---|--------------------|--------------------------------------|------|------|
| | | Min | Max | |
| Clock | CLK | 1 | 66 | MHz |
| Clock Cycle | tCLK | 15 | - | ns |
| Initial Access Time | tIAA | - | 70 | ns |
| Burst Access Time Valid Clock to Output Delay | tBA | - | 11.5 | ns |
| $\overline{\text{AVD}}$ Setup Time to CLK | tAVDS | 5 | - | ns |
| $\overline{\text{AVD}}$ Hold Time from CLK | tAVDH | 6 | - | ns |
| Address Setup Time to CLK | tACS | 5 | - | ns |
| Address Hold Time from CLK | tACH | 6 | - | 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 ¹⁾ | - | 20 | ns |
| $\overline{\text{OE}}$ Disable to Output High Z | tOEZ ¹⁾ | - | 15 | ns |
| $\overline{\text{CE}}$ Setup Time to CLK | tCES | 6 | - | ns |
| CLK High or Low Time | tCLKH/L | tCLK/3 | - | ns |
| CLK ²⁾ to RDY valid | trDYO | - | 11.5 | ns |
| CLK to RDY Setup Time | trDYA | - | 11.5 | ns |
| RDY Setup Time to CLK | trDYS | 3.5 | - | ns |
| $\overline{\text{CE}}$ low to RDY valid | tCER | - | 15 | ns |

Note

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

5.5 AC Characteristics for Asynchronous Read

See Timing Diagrams 6.3, 6.4, 6.5 and 6.6

| Parameter | Symbol | KFG1G16Q2M/KFH2G16Q2M /KFW4G16Q2M | | Unit |
|--|--------|--------------------------------------|-----|------|
| | | Min | Max | Unit |
| 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{OE} enable | tOEH | 0 | - | ns |
| \overline{CE} Setup to \overline{AVD} falling edge | tCA | 0 | - | ns |
| \overline{CE} Disable to Output & RDY High Z ¹⁾ | tCEZ | - | 20 | ns |
| \overline{OE} Disable to Output High Z ¹⁾ | tOEZ | - | 15 | ns |
| \overline{WE} Disable to \overline{AVD} Enable | tWEA | 15 | - | ns |

NOTE:

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

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

See Timing Diagrams 6.12, 6.13 and 6.14

| Parameter | Symbol | Min | Max | Unit |
|---|-------------------------------|-----|-----|------|
| \overline{RP} & Reset Command Latch to BootRAM Access | tReady1 (BufferRAM) | - | 5 | μs |
| \overline{RP} & Reset Command Latch(During Load Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 10 | μs |
| \overline{RP} & Reset Command Latch(During Program Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 20 | μs |
| \overline{RP} & Reset Command Latch(During Erase Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 500 | μs |
| \overline{RP} & Reset Command Latch(NOT During Internal Routines) to INT High (Note1) | tReady2 (NAND Flash Array) | - | 10 | μs |
| \overline{RP} Pulse Width (Note2) | tRP | 200 | - | ns |

Note

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

5.7 AC Characteristics for Asynchronous Write/Load/ Program/Erase Operation

See Timing Diagrams 6.7, 6.8 and 6.9

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|-------------------|-----|-----|-----|------|
| \overline{WE} Cycle Time | t _{wc} | 70 | - | - | ns |
| \overline{AVD} low pulse width | t _{avdp} | 12 | - | - | ns |
| Address Setup Time | t _{awes} | 0 | - | - | ns |
| Address Hold Time | t _{ah} | 30 | - | - | ns |
| Data Setup Time | t _{ds} | 25 | - | - | ns |
| Data Hold Time | t _{dh} | 0 | - | - | ns |
| \overline{CE} Setup Time | t _{cs} | 0 | - | - | ns |
| \overline{CE} Hold Time | t _{ch} | 10 | - | - | ns |
| \overline{WE} Pulse Width | t _{wpl} | 40 | - | - | ns |
| \overline{WE} Pulse Width High | t _{wph} | 30 | - | - | ns |
| \overline{AVD} Disable to \overline{WE} Disable | t _{vlwh} | 15 | - | - | ns |
| \overline{WE} Disable to \overline{AVD} Enable | t _{wea} | 15 | - | - | ns |

5.8 AC Characteristics for Load/Program/Erase Performance

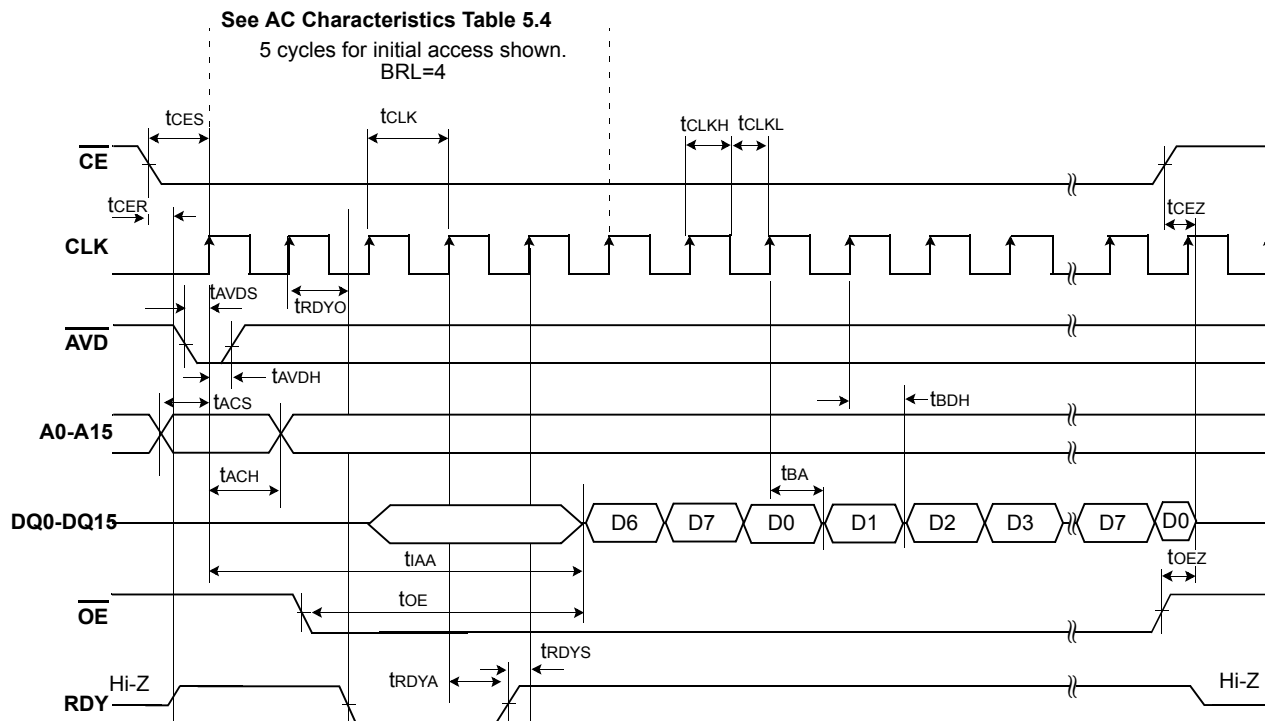
See Timing Diagrams 6.8, 6.9, and 6.10

| Parameter | Symbol | Min | Typ | Max | Unit | |
|--|-------------------|--------------------|-----|-----|--------|----|
| Sector Load time(Note 1) | t _{rd1} | - | 23 | 35 | μs | |
| Page Load time(Note 1) | t _{rd2} | - | 30 | 45 | μs | |
| Sector Program time(Note 1) | t _{pgm1} | - | 205 | 720 | μs | |
| Page Program time(Note 1) | t _{pgm2} | - | 220 | 750 | μs | |
| OTP Access Time(Note 1) | t _{otp} | - | 500 | 700 | ns | |
| Lock/Unlock/Lock-tight Time(Note 1) | t _{lock} | - | 500 | 700 | ns | |
| Erase Suspend Time(Note 1) | t _{esp} | - | 400 | 500 | μs | |
| Erase Resume Time(Note 1) | 1 Block | t _{ers1} | - | 2 | 3 | ms |
| | 2~64 Blocks | t _{ers2} | - | 4 | 6 | ms |
| Number of Partial Program Cycles in the sector (Including main and spare area) | NOP | - | - | 2 | cycles | |
| Block Erase time (Note 1) | 1 Block | t _{bers1} | - | 2 | 3 | ms |
| | 2~64 Blocks | t _{bers2} | - | 4 | 6 | ms |
| Multi Block Erase Verify Read time(Note 1) | t _{rd3} | - | 70 | 100 | μs | |

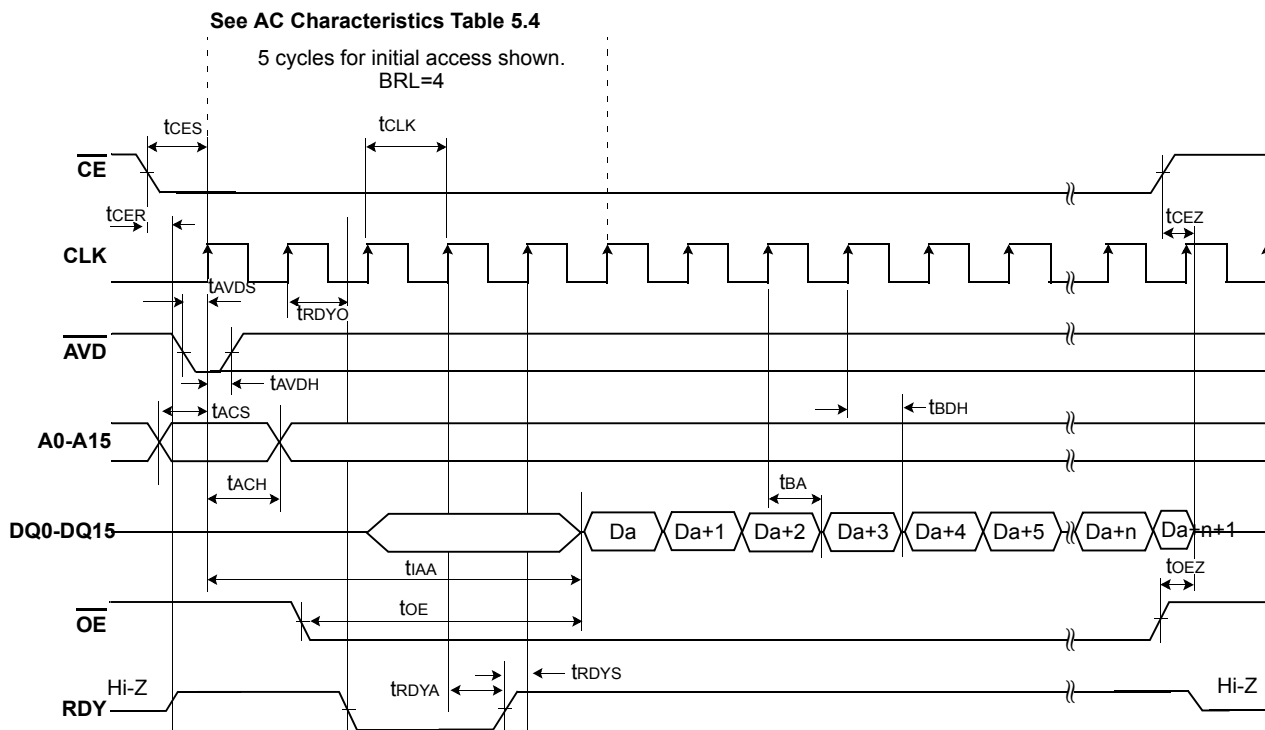
Note 1) These parameters are tested based on INT bit of interrupt register. Because the time on INT pin is related to the pull-up and pull-down resistor value.

6.0 TIMING DIAGRAMS

6.1 8-Word Linear Burst Mode with Wrap Around

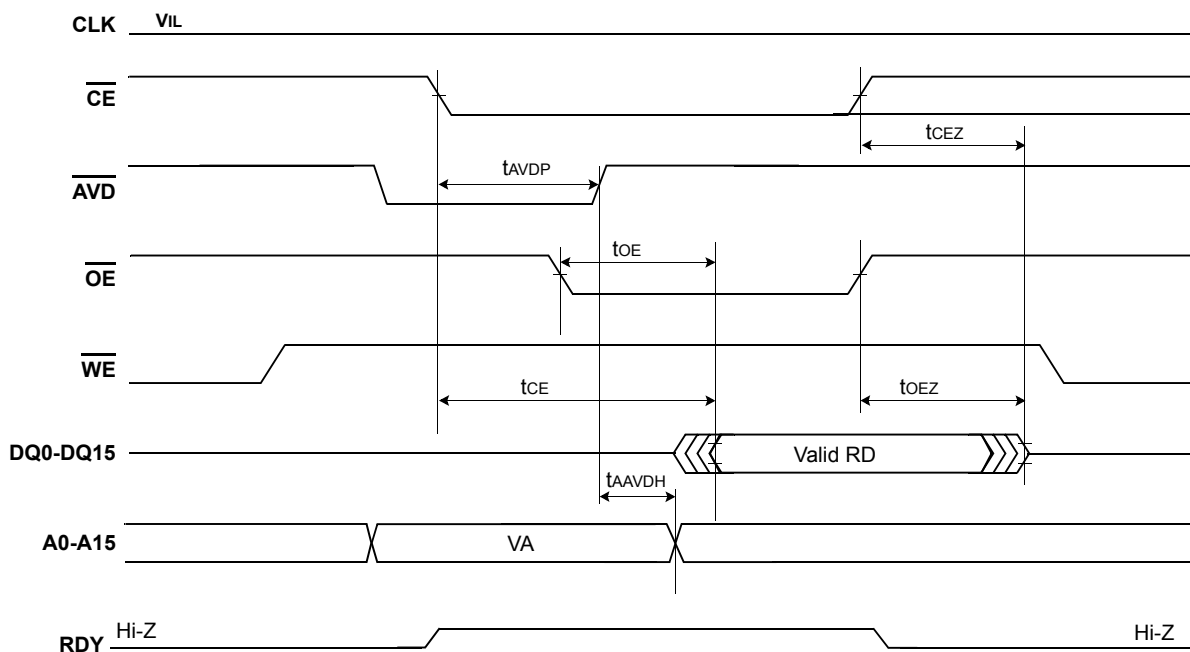


6.2 Continuous Linear Burst Mode with Wrap Around



6.3 Asynchronous Read (VA and \overline{AVD} Transition Before \overline{CE} Low)

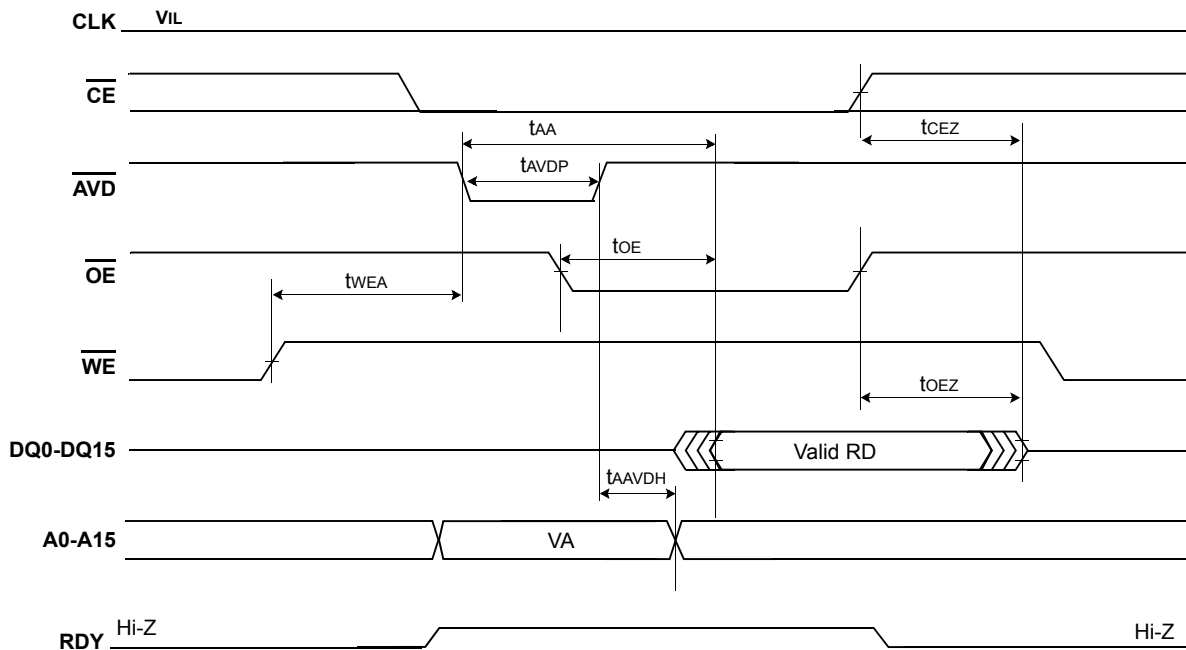
See AC Characteristics Table 5.5



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

6.4 Asynchronous Read (VA and \overline{AVD} Transition After \overline{CE} Low)

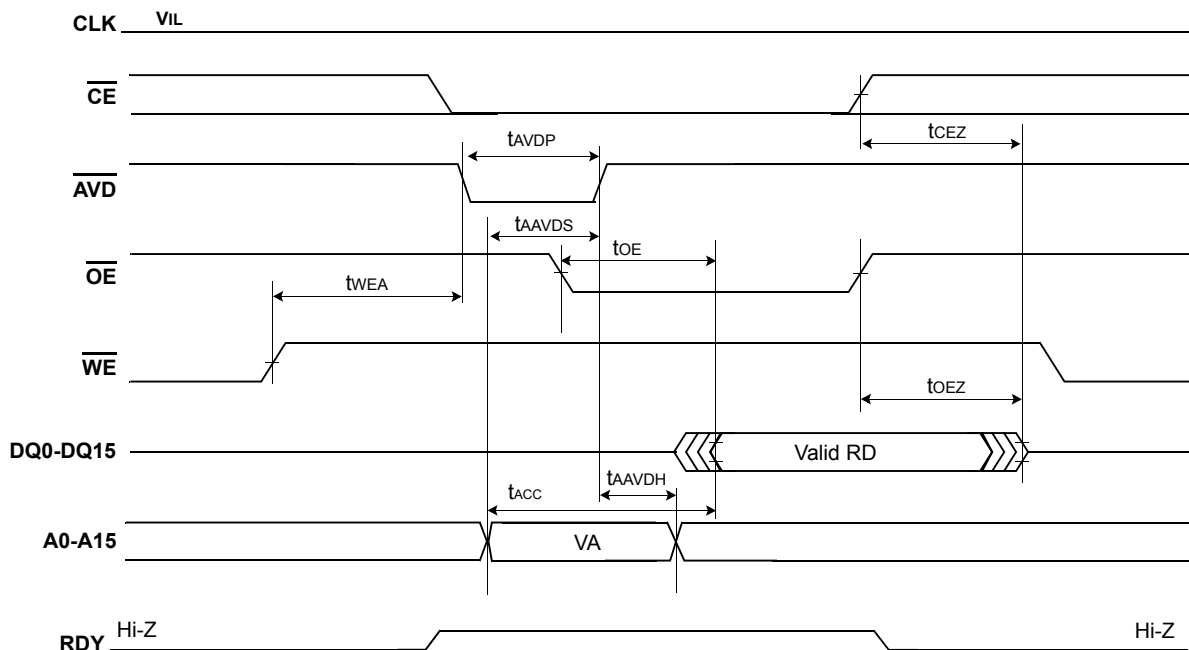
See AC Characteristics Table 5.5



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

6.5 Asynchronous Read (\overline{VA} and \overline{AVD} Transition After \overline{CE} Low)

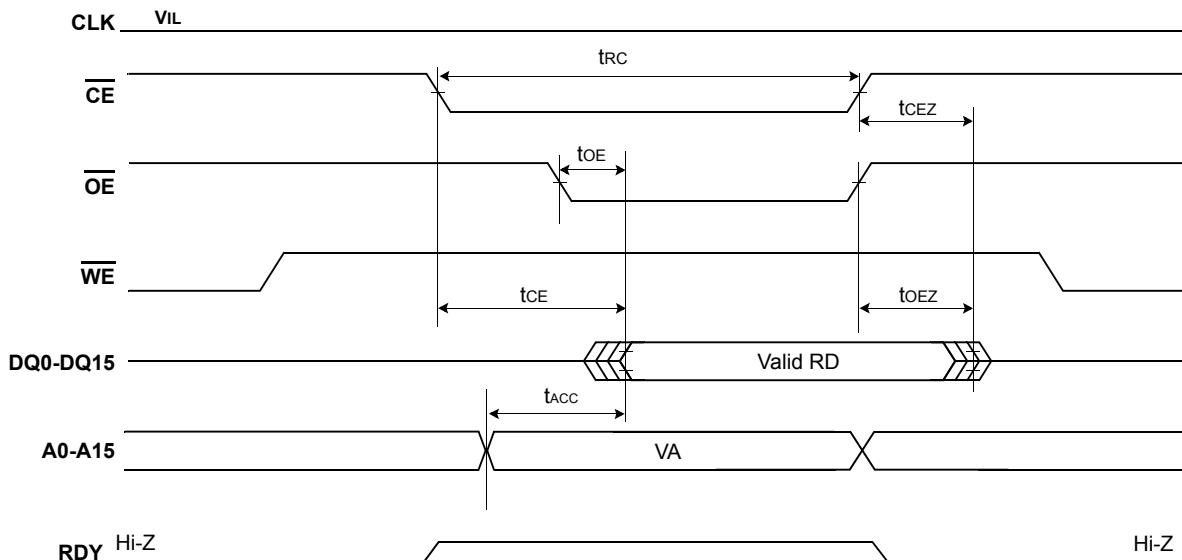
See AC Characteristics Table 5.5



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

6.6 Asynchronous Read (\overline{AVD} is tied to \overline{CE})

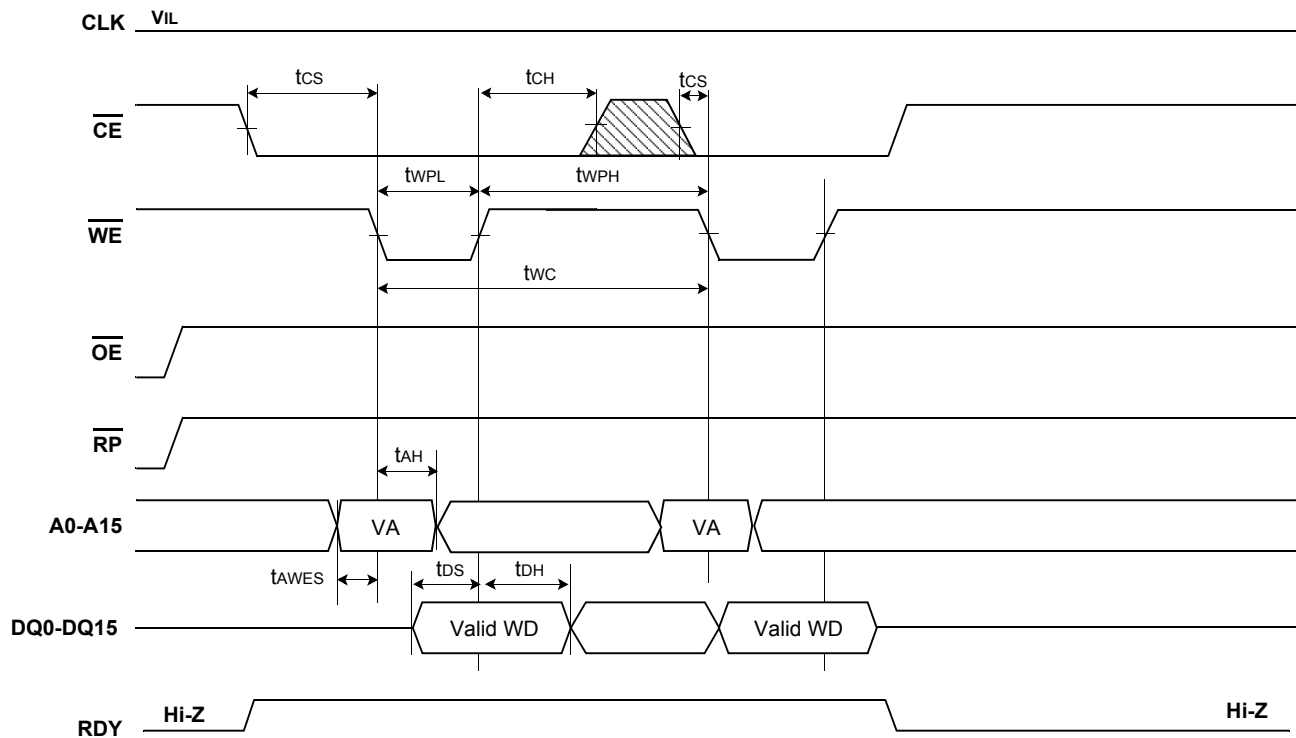
See AC Characteristics Table 5.5



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

6.7 Asynchronous Write

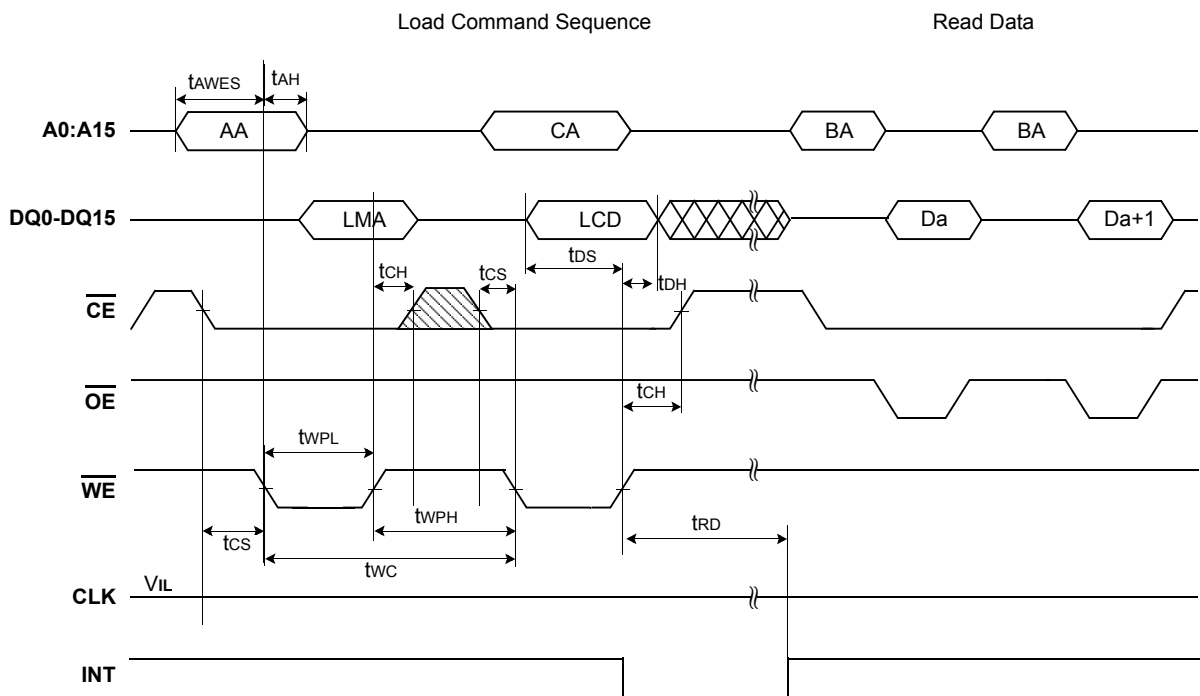
See AC Characteristics Table 5.7



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

6.8 Load Operation Timing

See AC Characteristics Tables 5.7 and 5.8

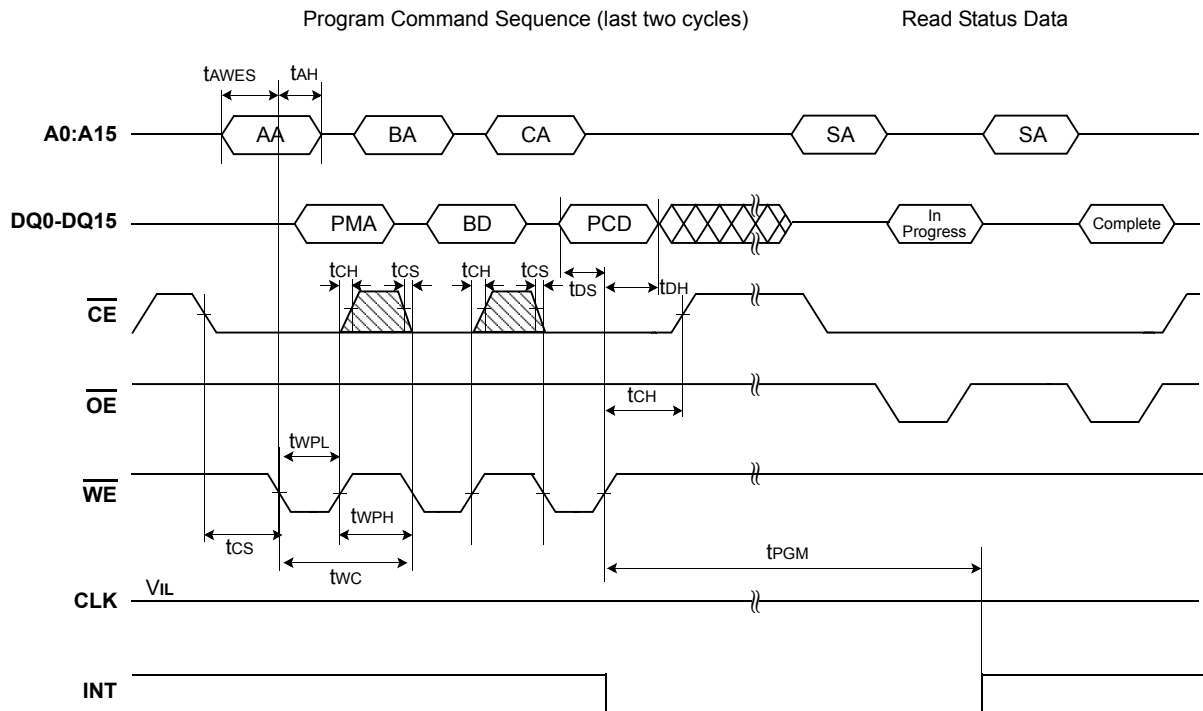


NOTES:

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

6.9 Program Operation Timing

See AC Characteristics Tables 5.7 and 5.8

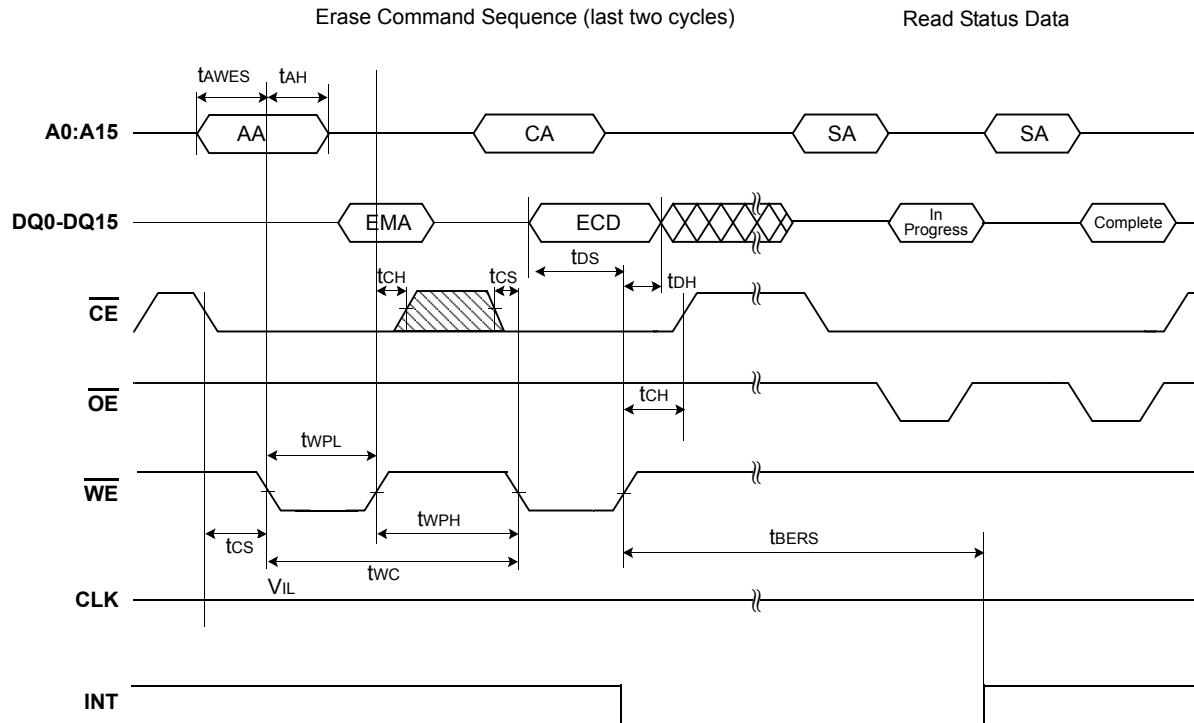


NOTES:

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.

6.10 Block Erase Operation Timing

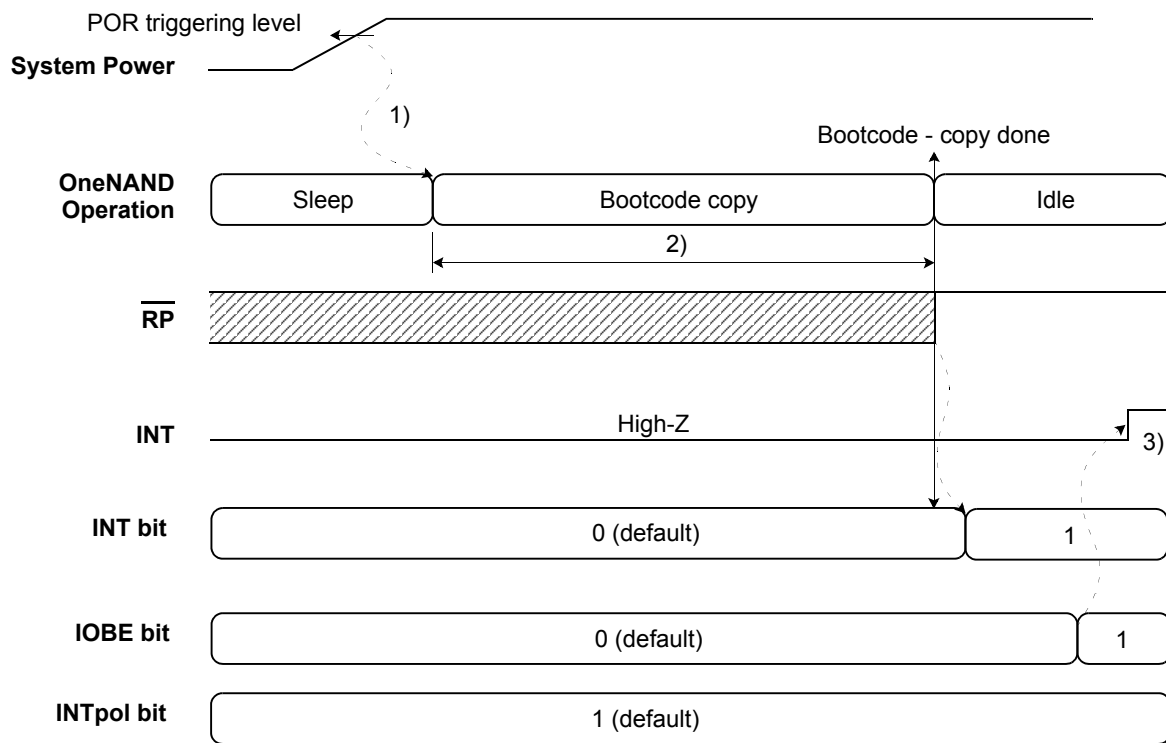
See AC Characteristics Tables 5.7 and 5.8



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.

6.11 Cold Reset Timing



Note: 1) Bootcode copy operation starts 400us later than POR activation.

The system power should reach Vcc after POR triggering level(typ. 1.5V) within 400us for valid boot code data.

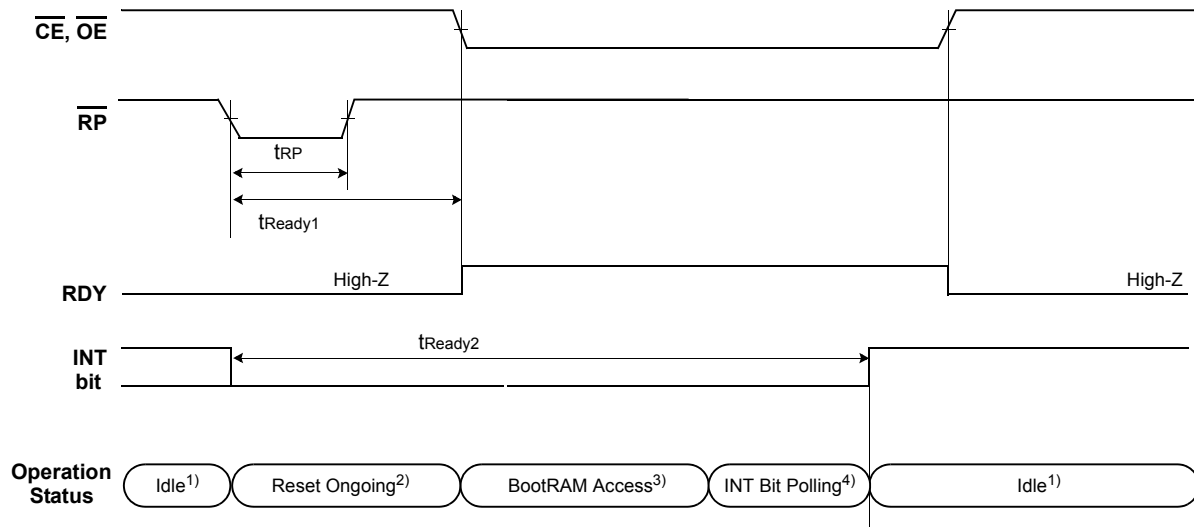
2) 1K bytes Bootcode copy takes 70us(estimated) from sector0 and sector1/page0/block0 of NAND Flash array to BootRAM. Host can read Bootcode in BootRAM(1K bytes) after Bootcode copy completion.

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'

6.12 Warm Reset Timing

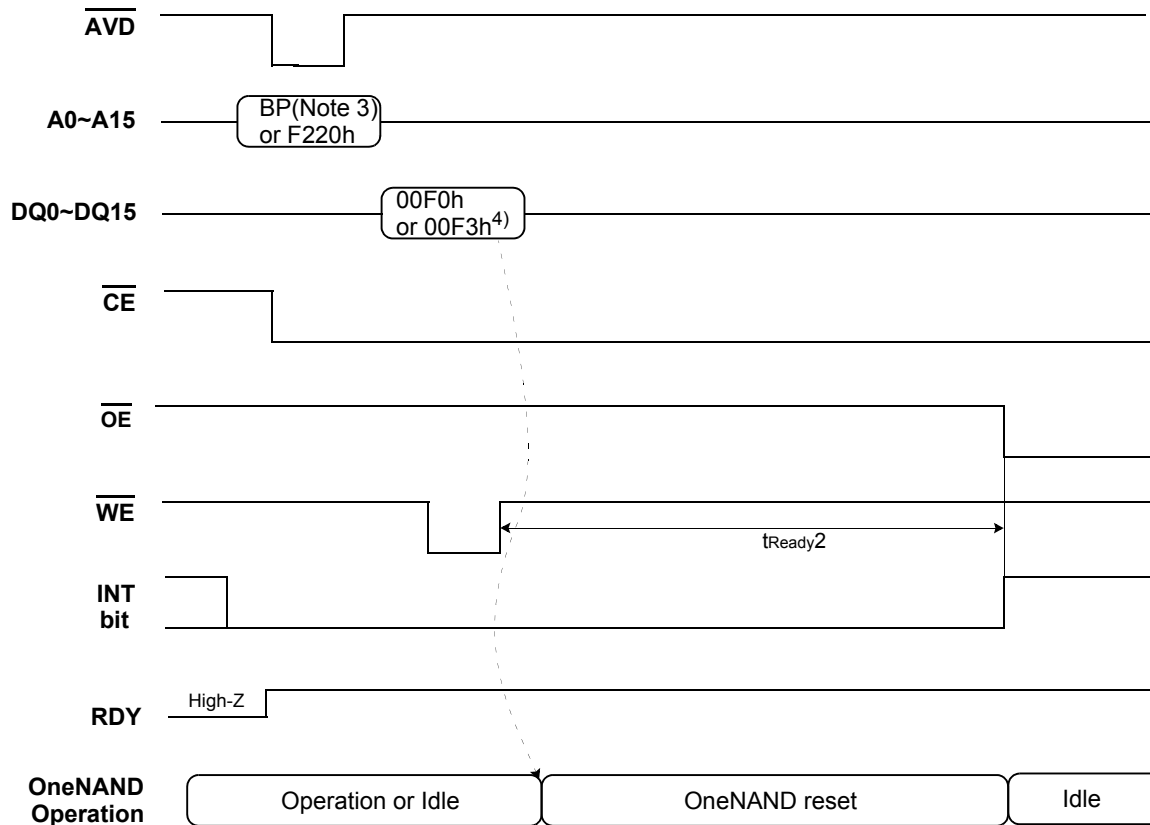
See AC Characteristics Tables 5.6



NOTES:

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

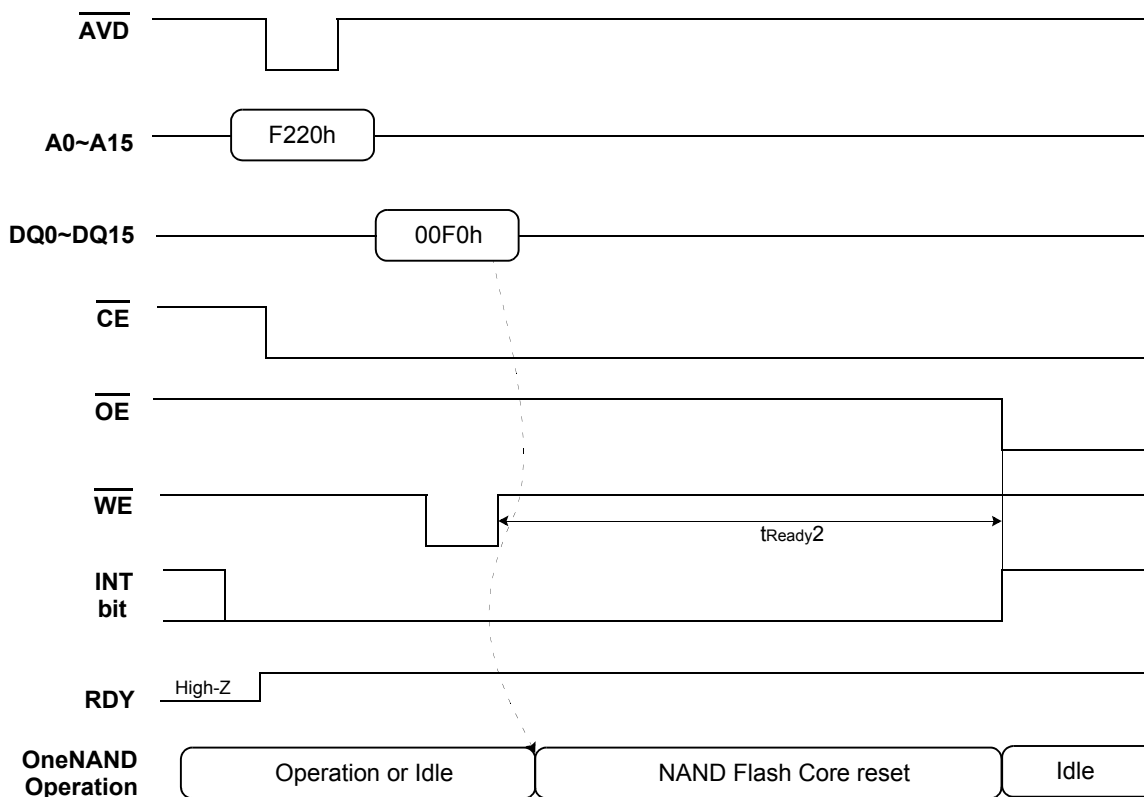
6.13 Hot Reset Timing



NOTE:

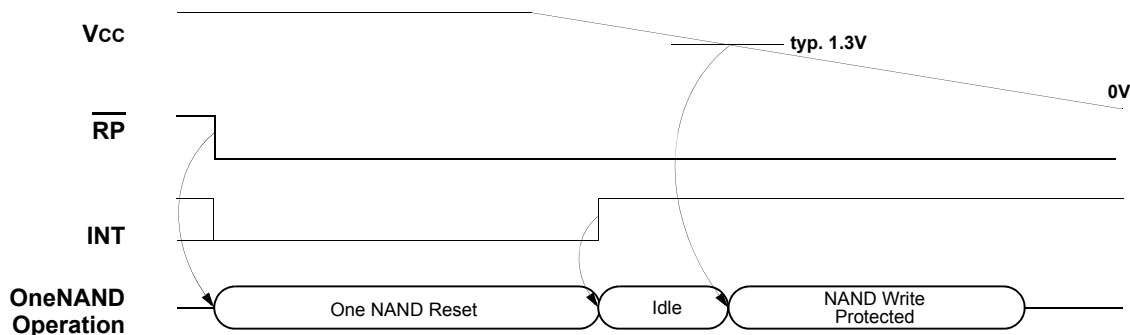
1. Internal reset operation means that the device initializes internal registers and makes output signals go to default status and bufferRAM data are kept unchanged after Warm/Hot reset operations.
2. Reset command : Command based reset or Register based reset
3. BP(Boot Partition): BootRAM area [0000h~01FFh,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

6.14 NAND Flash Core Reset Timing



6.15 Data Protection Timing 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 V_{cc} is below about 1.3V. \overline{RP} pin provides hardware protection and is recommended to be kept at V_{IL} before power-down.



7.0 TECHNICAL AND APPLICATION NOTES

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

7.1 Methods of Determining Interrupt Status

There are two methods of determining Interrupt Status on the 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.

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

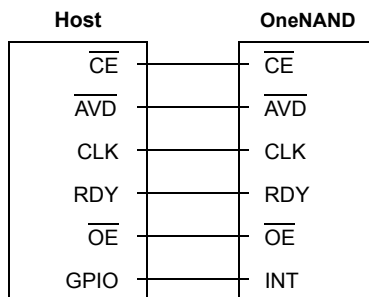
INT can be tied to a Host GPIO to detect the rising edge of INT, signaling the end of a command operation.



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

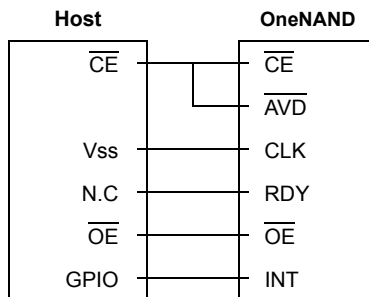
Synchronous Mode Using the INT Pin

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



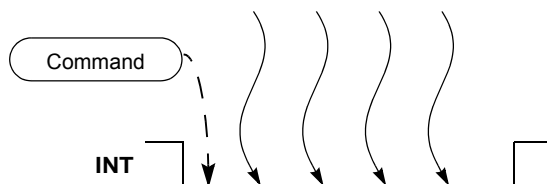
Asynchronous Mode Using the INT Pin

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



7.1.2 Polling the Interrupt Register Status Bit

An alternate method of determining the end of an operation is to continuously monitor the Interrupt Status Register Bit instead of using the INT pin.



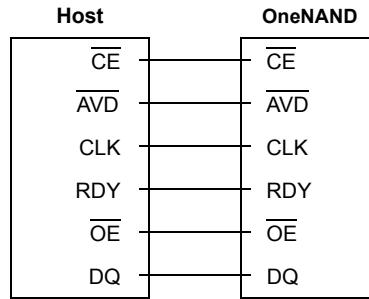
This can be configured in either a synchronous mode or an asynchronous mode.

OneNAND4G(KFW4G16Q2M-DEB6)
OneNAND2G(KFH2G16Q2M-DEB6)
OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

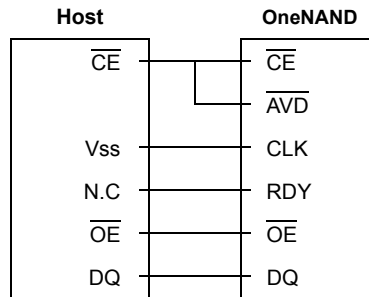
Synchronous Mode Using Interrupt Status Register Bit Polling

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



Asynchronous Mode Using Interrupt Status Register Bit Polling

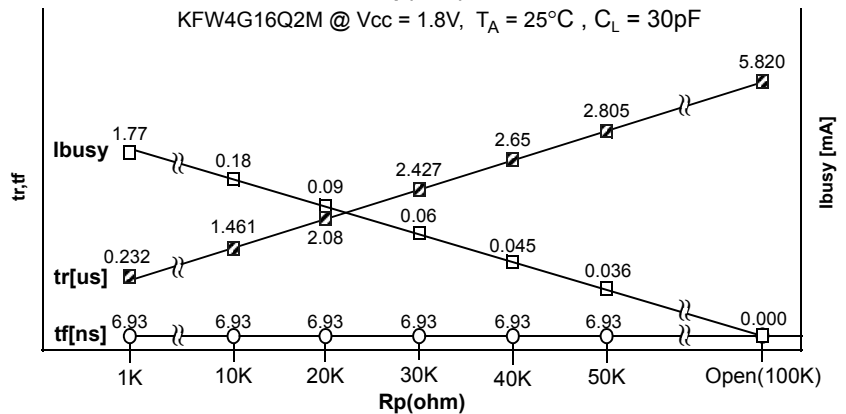
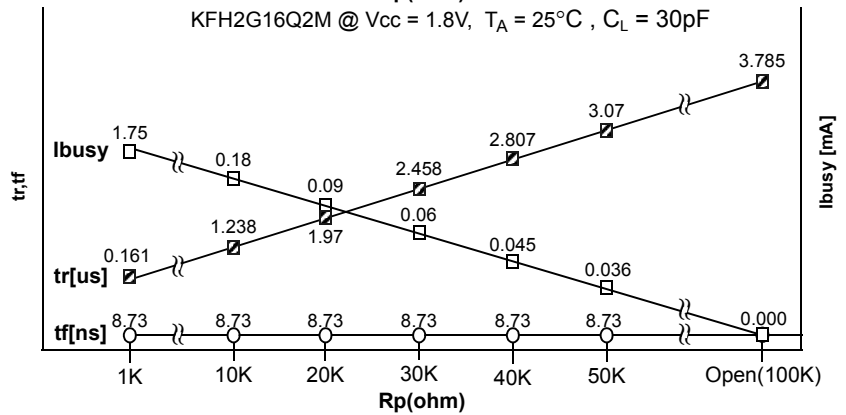
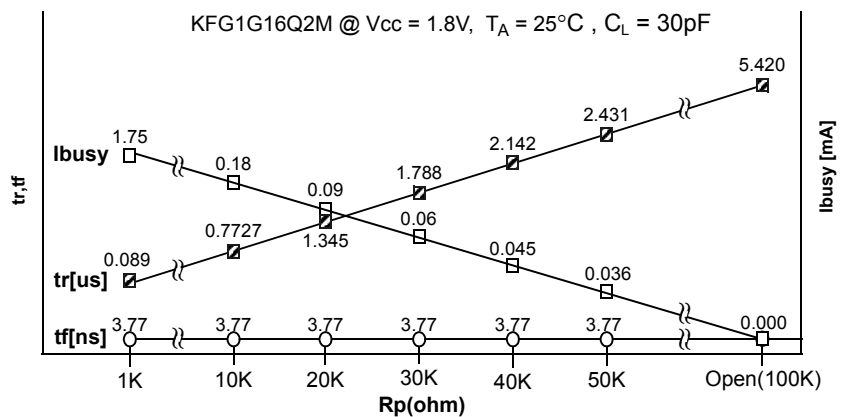
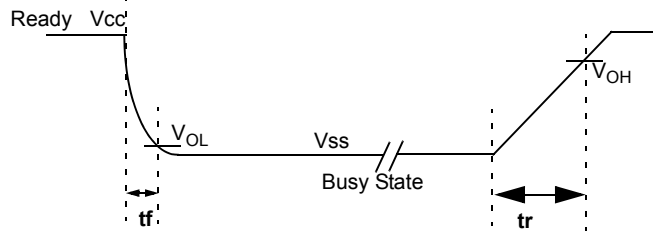
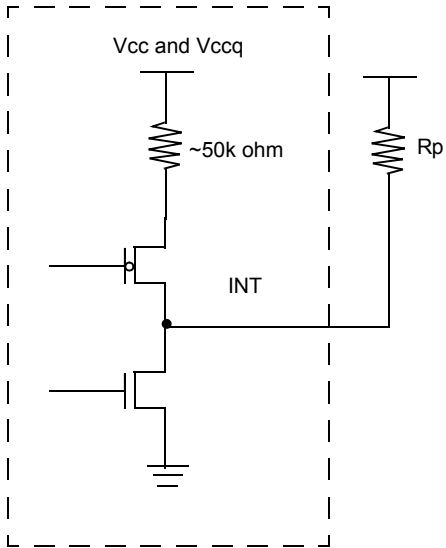
When configured to operate in an asynchronous mode, \overline{CE} and \overline{AVD} of the OneNAND are tied to \overline{CE} of the Host. CLK is tied to the Host Vss (Ground). RDY is tied to a no-connect. \overline{OE} and DQ of the OneNAND and Host are tied together.



7.1.3 Determining Rp Value

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

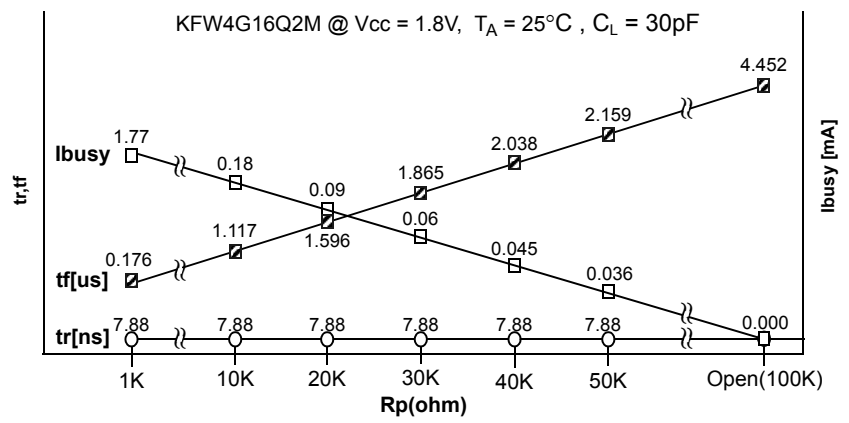
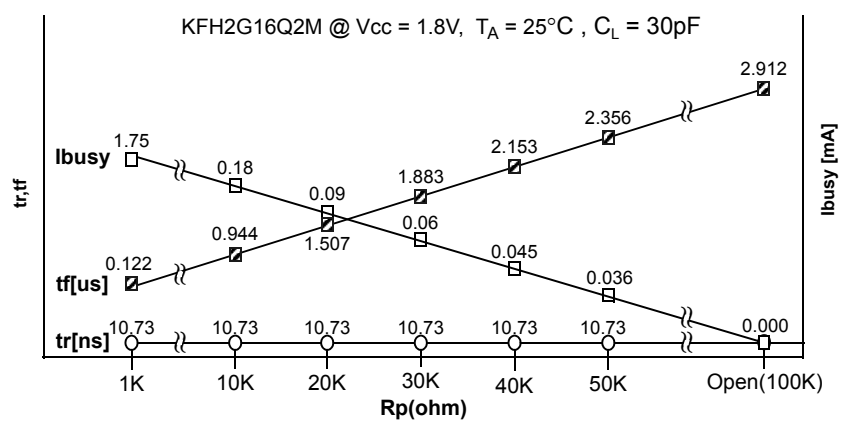
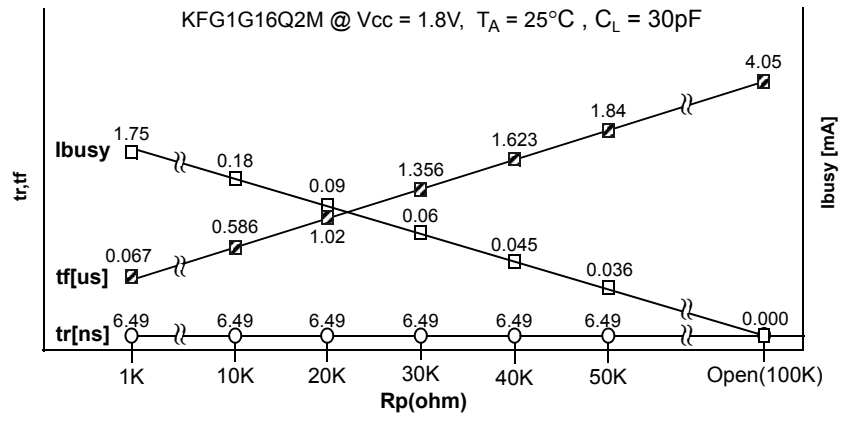
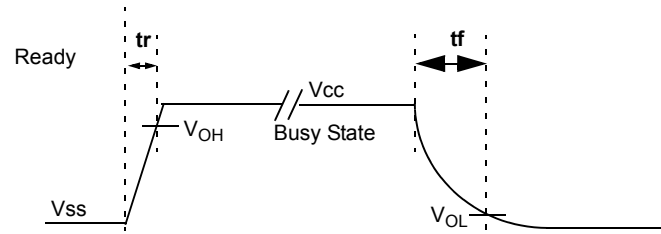
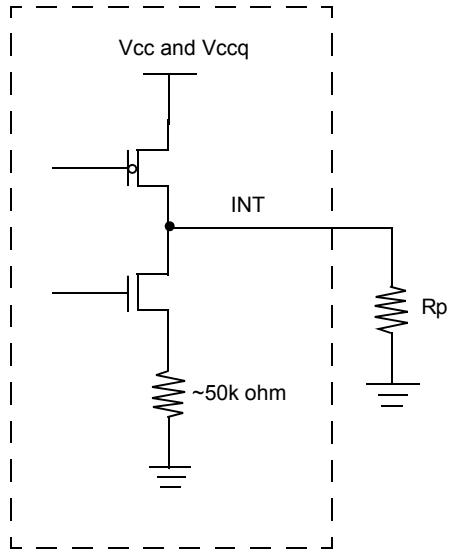
INT pol = 'High'



OneNAND4G(KFW4G16Q2M-DEB6)
 OneNAND2G(KFH2G16Q2M-DEB6)
 OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY

INT pol = 'Low'



7.2 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 Partition of NAND Flash Array and OneNAND Boot Sequence. In this boot scheme, boot code is comprised of BL1, where BL stands for Boot Loader, BL2, and BL3. Moreover, the size of the boot code is larger than 3KB (the 3rd case above). BL1 is called primary boot loader in other words. Here is the table of detailed explanations about the function of each boot loader in this specific boot scheme.

7.2.1 Boot Loaders in OneNAND

Boot Loaders in OneNAND

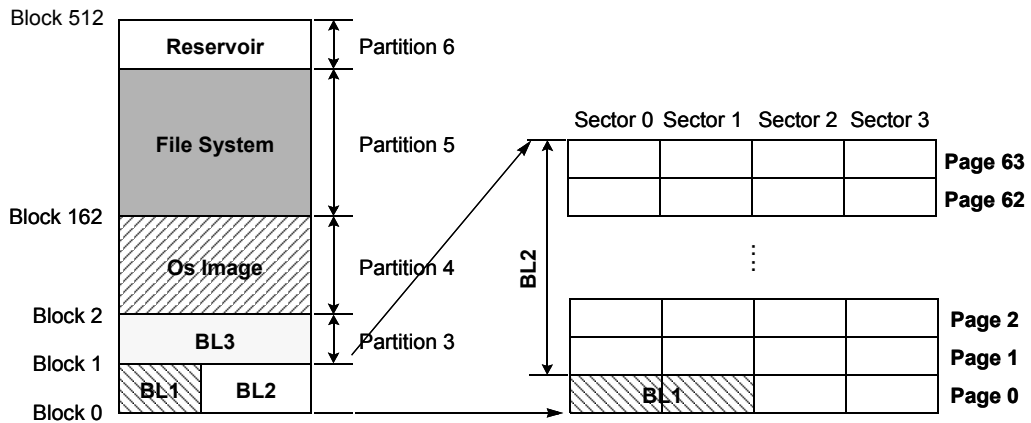
| Boot Loader | Description |
|----------------|--|
| BL1 | Moves BL2 from NAND Flash Array to DRAM through two DataRAMs using dual buffering |
| BL2 | Moves OS image (or BL3 optionally) from NAND Flash Array to DRAM through two DataRAMs using dual buffering |
| BL3 (Optional) | Moves or writes the image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

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

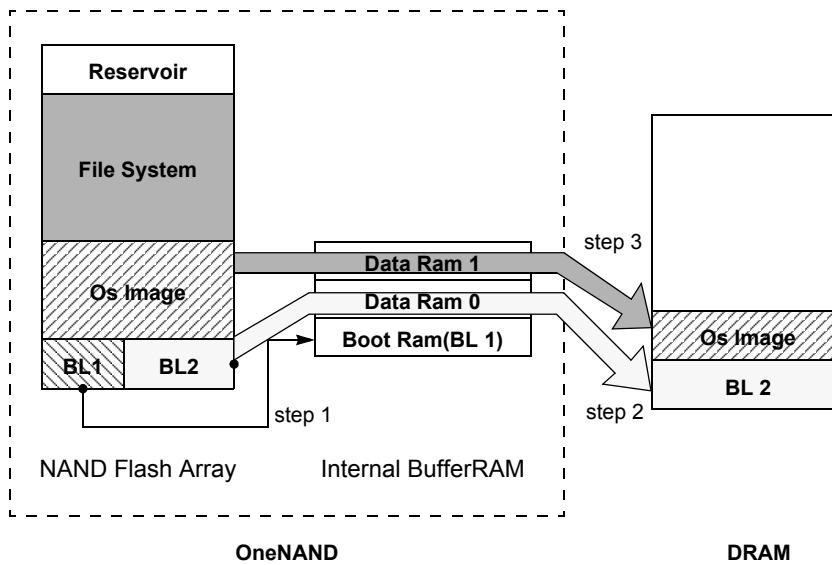
7.2.2 Boot Sequence

Boot Sequence :

1. Power is on
BL1 is loaded into BootRAM
2. BL1 is executed in BootRAM
BL2 is loaded into DRAM through two DataRAMs 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



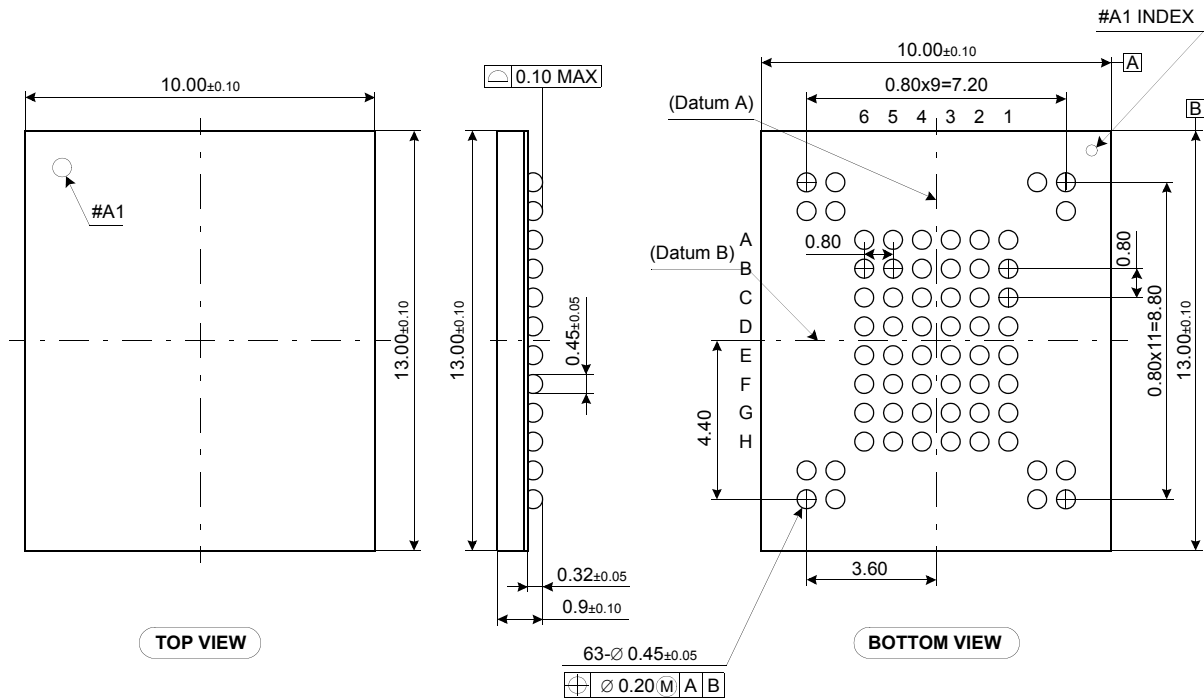
Partition of NAND Flash array



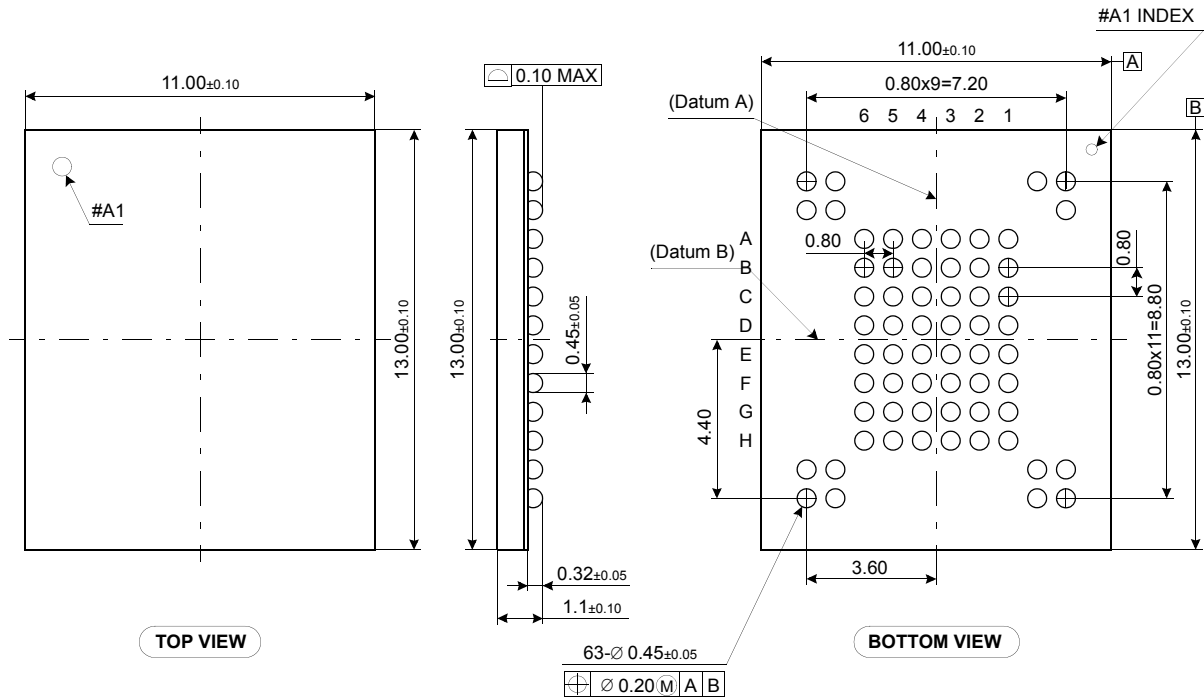
NOTE:
 Step 2 and Step 3 can be copied into DRAM through two DataRAMs using dual buffering

OneNAND Boot Sequence

8.0 PACKAGE DIMENSIONS



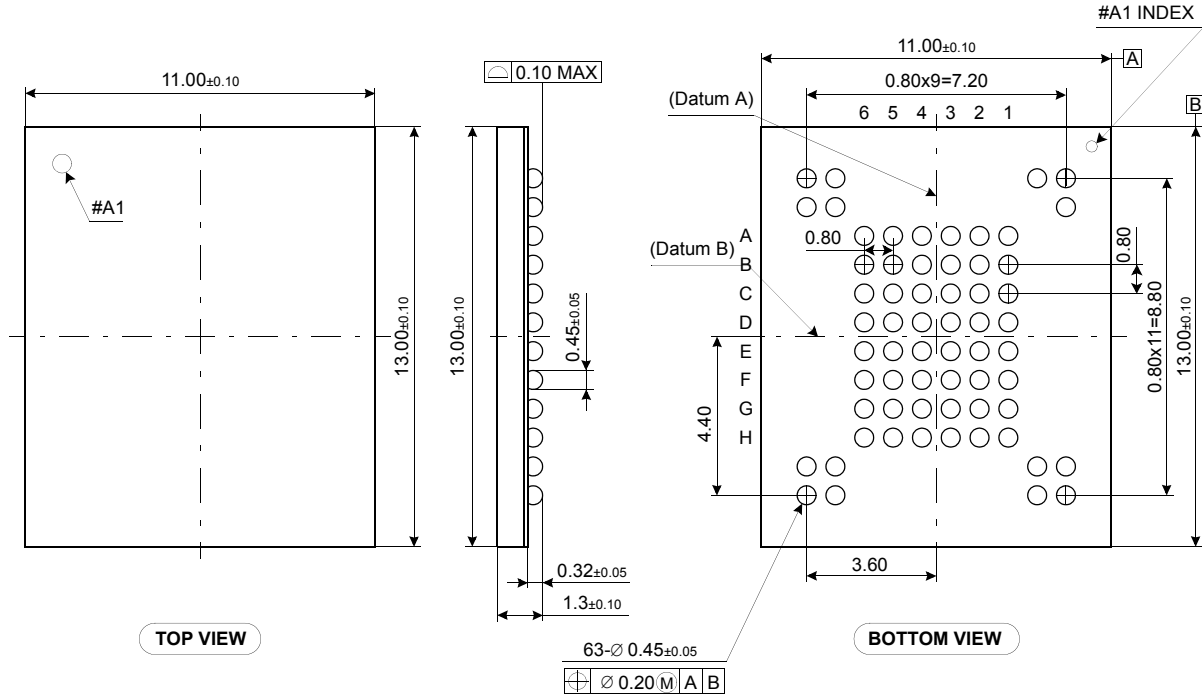
1G product (KFG1G16Q2M)



2G product (KFH2G16Q2M)

OneNAND4G(KFW4G16Q2M-DEB6)
 OneNAND2G(KFH2G16Q2M-DEB6)
 OneNAND1G(KFG1G16Q2M-DEB6)

FLASH MEMORY



4G product (KFW4G16Q2M)