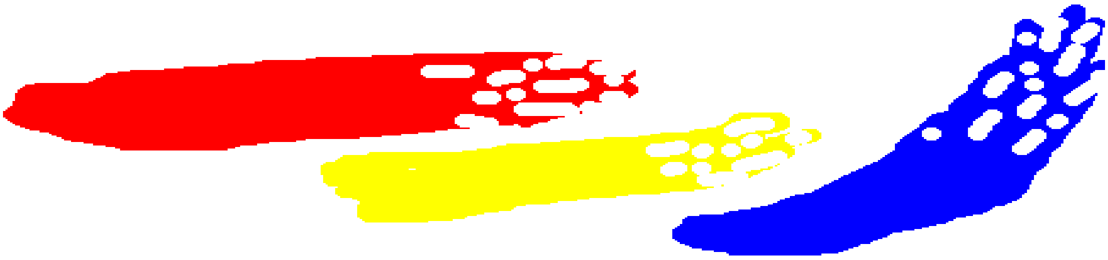


APPLICATION NOTE

for NAND Flash Memory

(Revision 2.0)

Memory Product & Technology Division

1999. 12. 28

TABLE OF CONTENTS

¡ áINTRODUCTION

The need for nonvolatile data Flash Memory storage is ever increasing as the computer systems are shifting toward the mobile domain. Ruggedness, low power consumption, small and light forms are the obvious advantage of Flash-based storage over disk-based technology. As the bit-cost of Flash decreases continuously, the Flash storage is poised to benefit the performance and functionality of portable platforms and equipment. NAND Flash is becoming the core of this Flash disk technology.

1. SYSTEM INTEFACING

This section provides a brief overview of implementing NAND Flash Memory into the system. It covers the following.

- NAND Flash Architecture
- Designing for systems with O/S
- Designing for O/S-less systems
- ECC design guide

1-1. NAND Flash Architecture

NAND Flash is optimized for storage applications. 16 neighboring cells are serially connected without any contact area in-between to offer high density. Electron tunneling is used for both programming and erasing, providing low power consumption. This cell structure provides high scalability, making NAND Flash as the leader in the capacity race. Because the electron tunneling occurs across the whole channel area of cell, charge trapping density per unit area is lower than other Flash technologies, resulting in superior endurance in Program/Erase cycles. Programming and reading are performed on a sector or a page unit of 512 byte to emulate the popular sector size.

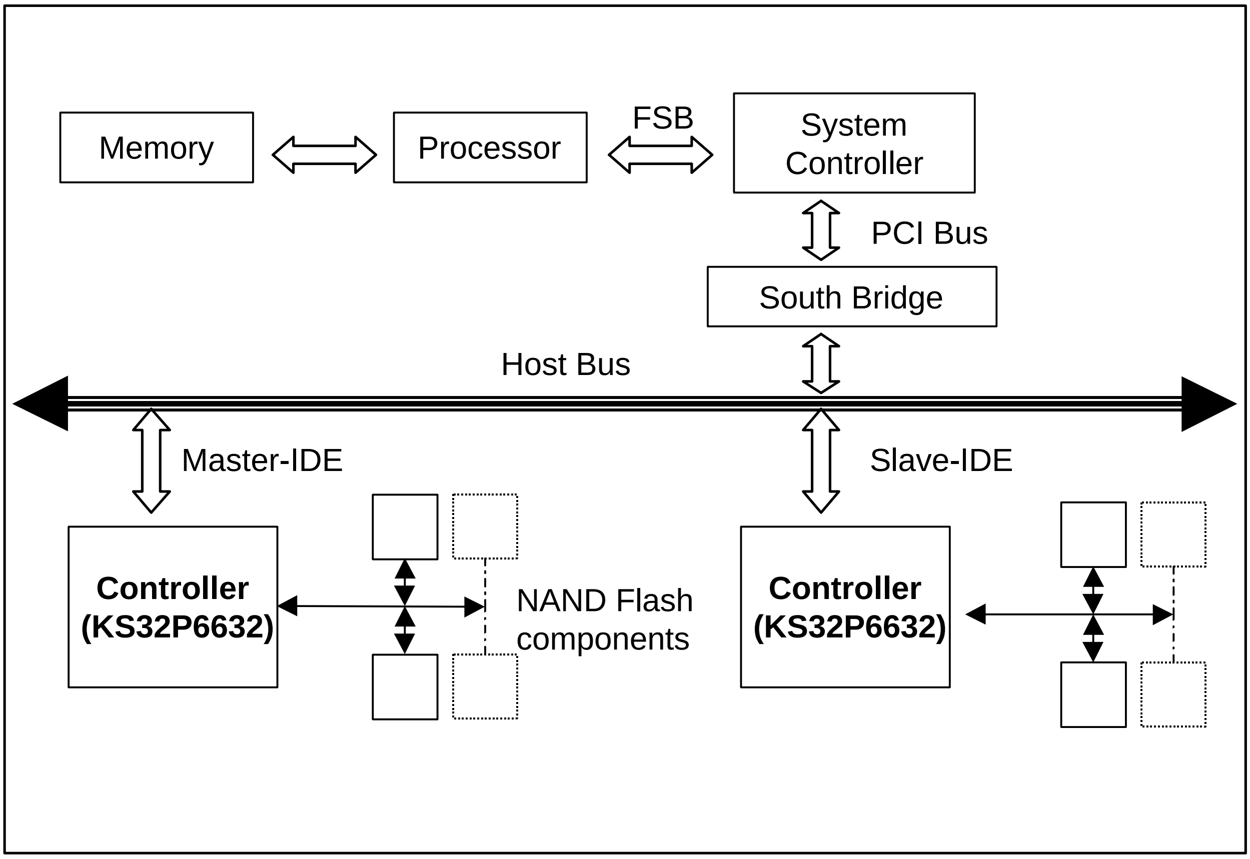
1.2 Designing for systems with O/S

Samsung's NAND Flash provides the highest performance and the most cost-effective solution for solid-state mass storage. When the host system has an operating system , the Flash disk operates as one of standard peripheral components and thus be connected to a host CPU's system bus through I/O interface like a IDE-disk drive. Since the host sees the Flash disk no different from hard disk drive, the native file system such as FAT (File Allocation Table) resides on the host operating system and a translator of host requests to the Flash disk is needed. The translator, embedded into the firmware of a controller directly interfacing Flash Memory, is commonly called Flash-Translation-Layer (FTL) and includes the following features.

- Converts the sector addresses addressed by the host to physical addresses of Flash Memory
- Converts host requests into the programming/erasing algorithms of associated Flash technology
- Detects the error and replaces the encountered bad sectors with the good by mapping them out

NOR-type Flash requires a very long erasing time. Block-erasure of NOR-type Flash takes about several-hundreds of millisecond while less than 2 ms with NAND Flash. To reduce the time overhead, FTL for NOR-type Flash normally includes the background erasure and garbage collection. When data of a block needs to be updated, the block is copied to another free or erased block with the updates. The new block is mapped as valid and the old as deleted. The deleted blocks are not actually erased. As the remaining erased blocks are decreased and below the pre-defined limit, the deleted blocks need to be reclaimed as free by being erased. Since the erasing takes too much time, it is processed as background operation when the controller is free from other tasks. This reclaiming process is called garbage collection. Because of its fast erasing time, the garbage-collection is not usually used with NAND Flash unless a very high performance is required. Figure 1-1 shows a typical block diagram of Flash disk as I/O device.

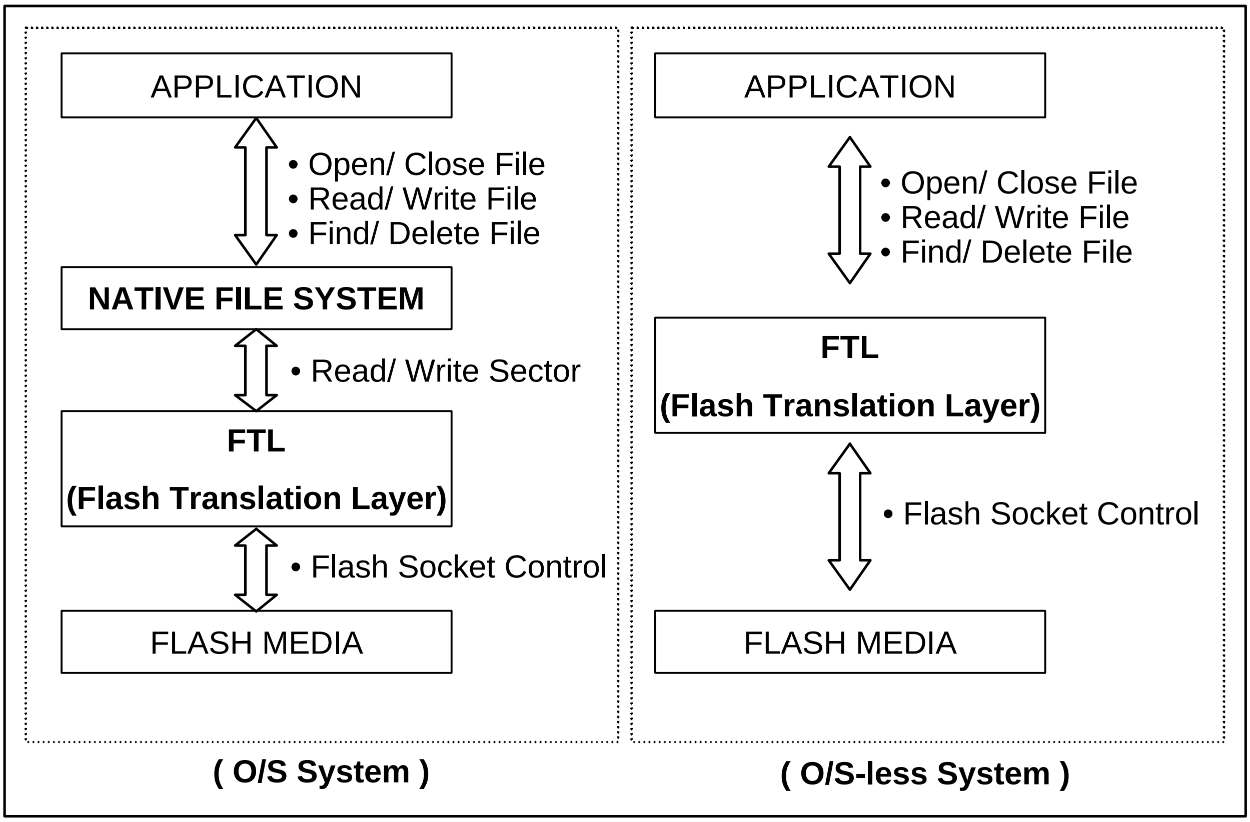
FIG. 1-1 TYPICAL BLOCK DIAGRAM of FLASH DISK as I/O DEVICE



1.3 Designing for O/S-less systems

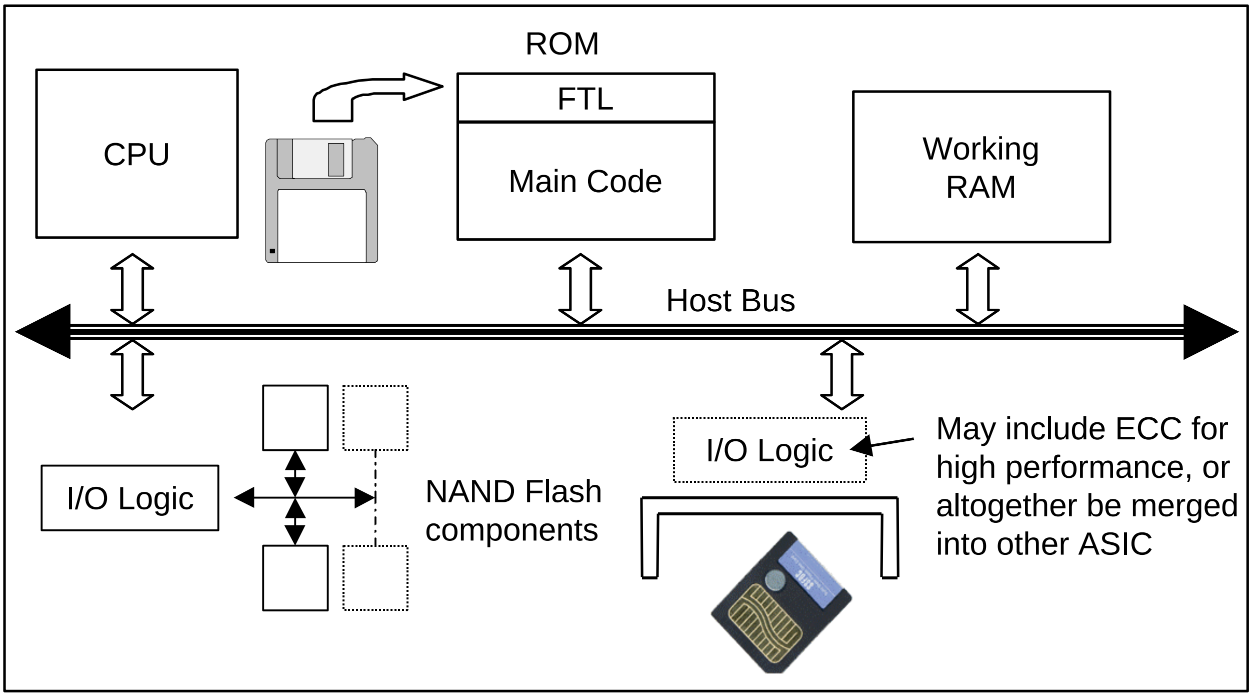
When the Flash Memory is used as storage disk for embedded systems, FTL incorporating a complete file system is necessary. The translation table from logical to physical address - necessary for FTL with O/S environment - is not necessary. Thus FTL could be in much smaller footprint. Figure 1-2 is a comparison in architecture of Flash file system between for O/S system and O/S-less.

FIG. 1-2 FILE SYSTEM COMPARISON for O/S and O/S-less SYSTEM



For its implementations, host MCU interfaces Flash directly and executes FTL using the working RAM. The FTL is stored in the firmware ROM. Figure 1-3 illustrates a typical structure of Flash disk in embedded systems.

FIG. 1-3 FLASH DISK TYPICAL STRUCTURE in EMBEDDED SYSTEM

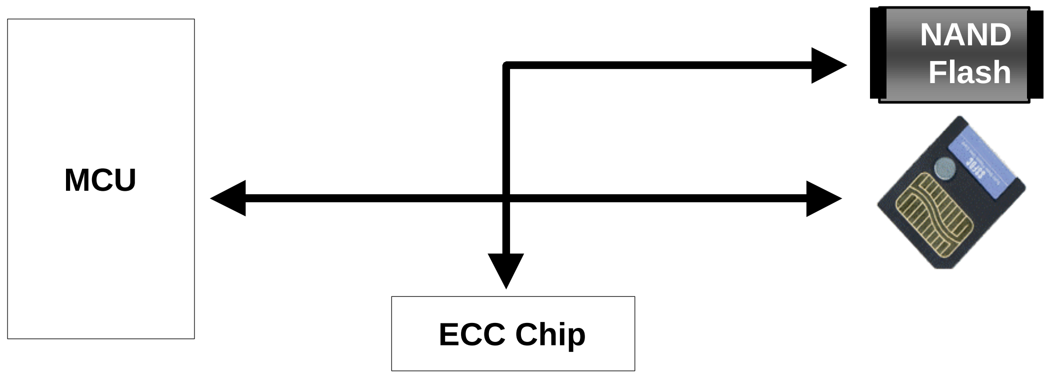

ROM
CPU
FTL
Main Code
Working RAM
Host Bus
I/O Logic
NAND Flash components
I/O Logic
May include ECC for high performance, or altogether be merged into other ASIC

The interface and associated timings for NAND Flash and the features of FTL need to be customized depending on the CPU performance and system requirements. NAND Flash requires only eight I/O's and five control clocks for operation. Standard WE/ and OE/ available from MCU may be also easily used for WE/ and RE/ of NAND Flash. CE/, CLE and ALE may be controlled by spare addresses. Because CE/ may be inactive during the sequential data-loading or burst-reading, interleaving between NAND Flash and other memory devices can be easily implemented. Spare 16 bytes of each page of NAND Flash can be utilized for keeping block status information such as good or bad status of each page, ECC parity and the number of P/E cycles. FTL for NAND Flash can be developed by system developers to tailor the system need or can be purchased from several companies including from M-Systems, Phoenix Technologies and Tokyo-Electron Device.

1.4 ECC Design Guide

To ensure highly reliable Flash disk, ECC is required for NAND Flash. One bit error correction per each page or sector would serve the purpose given the failure modes of long-term life test. Two types of failures could occur : write-failure or read-failure. Write-failure is the case when the programming of a page or erasing of a block returns the failure status. This is easily detected by checking the write status bit from Flash. Read-failure is the case when the written data changes after the successful writing. The failure mode is single bit error, mostly due to leakage of the oxide surrounding floating gate of cell. Hamming code ECC is recommended to recover the error. If the target system has the right performance, the algorithm can be implemented in software. A reference algorithm can be found in the Samsung Flash web-site. Since it only requires a few thousand gates, ECC may be integrated into other ASIC. A recommended block diagram is shown in Figure 1-4 where ECC block interfaces host MCU in parallel with NAND Flash, providing minimal footprint and flexibility for supporting multiple NAND Flash chips. The operating timing diagram is shown in Figure 1-5 and 1-6. VHDL model of this reference architecture may be provided by Samsung. In addition, an ASIC for this ECC is also available from Samsung.

FIG. 1-4 ECC BLOCK DIAGRAM



ECC chip executes only parity generation and Flash/ SmartMedia is directly controlled by MCU.

FIG. 1-5 OPERATING TIMING DIAGRAM (PROGRAM)

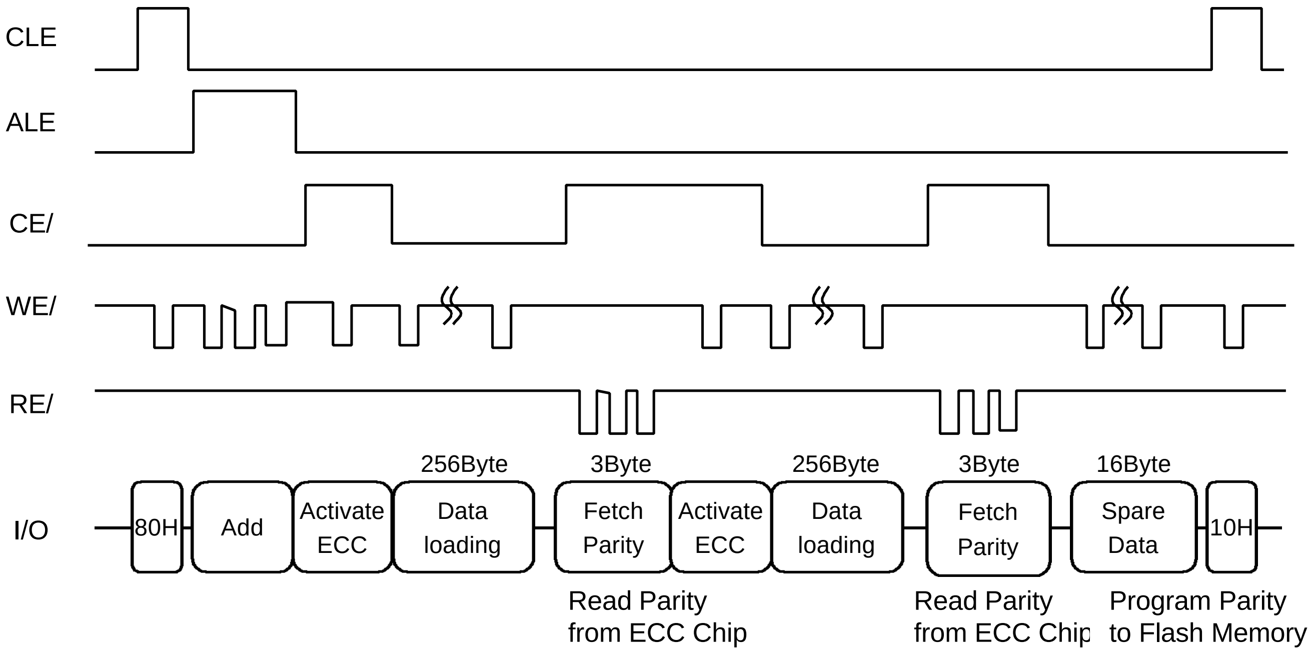
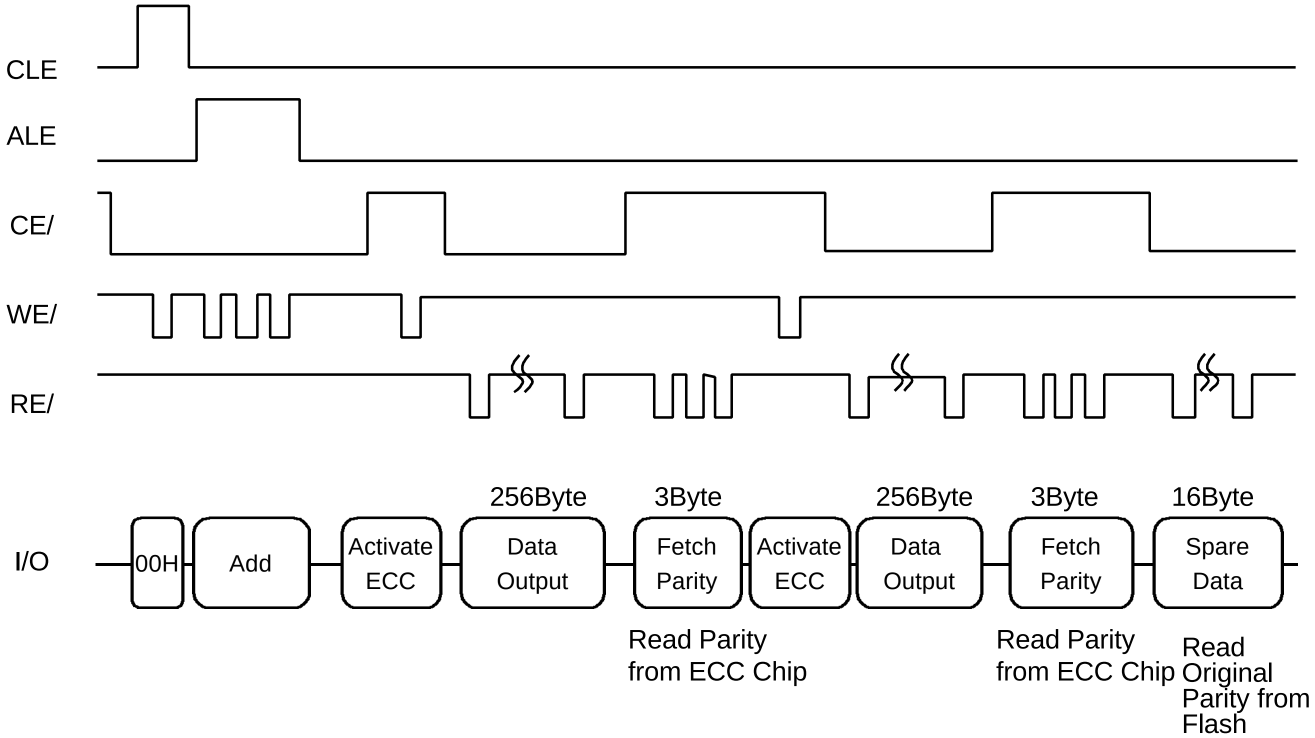


FIG. 1-6 OPERATING TIMING DIAGRAM (READ)



1.5 Development Tool

In order to help development of the software to be used in a system, Samsung is providing ISA board and sample source code which can control a Flash Memory in basic operations such as erase, program and read. For the details, please refer to the Flash section of Samsung`s web site (www.intl.samsungsemi.com) which branched to memory\flash\application note\available software.

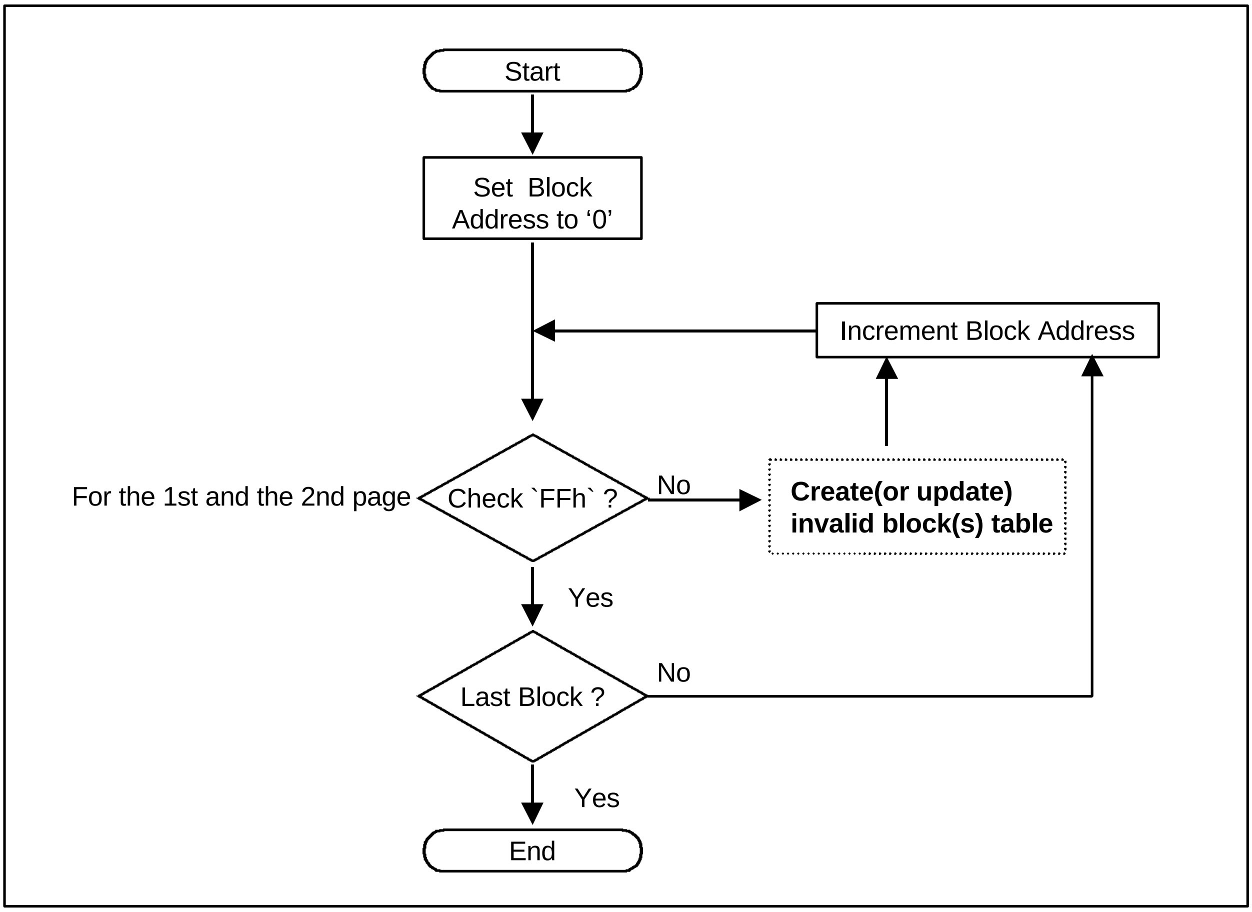
2. INVALID BLOCK(S) MANAGEMENT

The SAMSUNG NAND Flash may contain invalid blocks. The maximum number of invalid blocks is 3 to 35 depending on the device density(Refer to the Data Sheet). The system allowing imperfect memory cells is to maximize manufacturing yield and minimize the die costs. The devices with invalid blocks have the same quality levels as devices with all valid blocks and have the same all AC and DC characteristics. **The system designer must be able to mask out the invalid block(s) from address mapping.** An invalid block(s) do not affect the performance of valid block(s) because it could be isolated from the bit line and the common source line by a select transistor.

2-1. Identifying Initial Invalid Block(s)

All device locations are erased(FFh) except locations where the invalid block information is written prior to shipping. **Since the invalid block information is also erasable in most cases, it is impossible to recover the information once it has been erased.** Therefore, the system must be able to recognize the invalid block(s) based on the original invalid block information and create the invalid block table via the following suggested flow chart(Fig.2-1). Any intentional erasure of the original invalid block information is prohibited.

FIG. 2-1 FLOW CHART TO CREATE INVALID BLOCK TABLE



2-2. Management of Additional Invalid Block(s)

Failure Modes

The bad blocks other than those identified by factory can be increased at the customer side. The system should be designed to replace some additional failing blocks with the reserved blocks. As described in the datasheet, the error-detection mechanism is needed for both write (program/erase) and read. After program or erase, the cells must be verified to detect any write-related failure. **The following possible failure modes should be considered when implementing a highly reliable system.**

TABLE 2-1 FAILURE MODE AND COUNTERMEASURE

Failure Mode		Detection and Countermeasure sequence
Write	Erase Failure	Status Read after Erase → Block Replacement
	Program Failure	Status Read after Program → Block Replacement
Read	Single Bit Failure	Verify ECC → ECC Correction

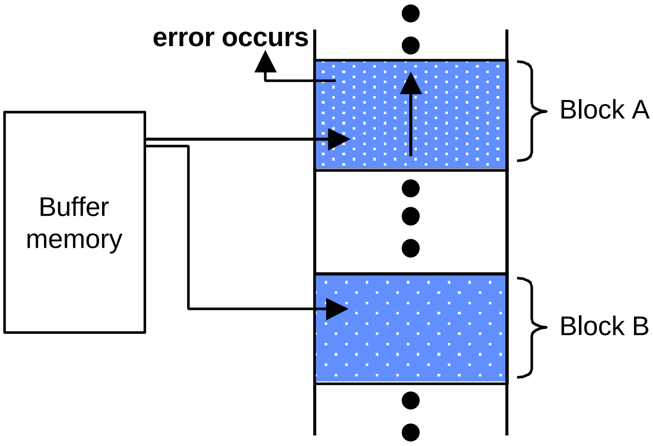
Countermeasure

Erase Failure

When an error occurs after an erase operation, prevent future accesses to this bad block by creating a table within the system or by using other appropriate scheme.

Program Failure (Page)

FIG. 2-2 BLOCK REPLACEMENT DURING OPERATION



When an error happens in block A, try to reprogram the data into another block B by loading it from an external buffer. Then, prevent further system access to block A by creating a 'Bad Block' table or other appropriate scheme.

Program Failure (Single Bit)

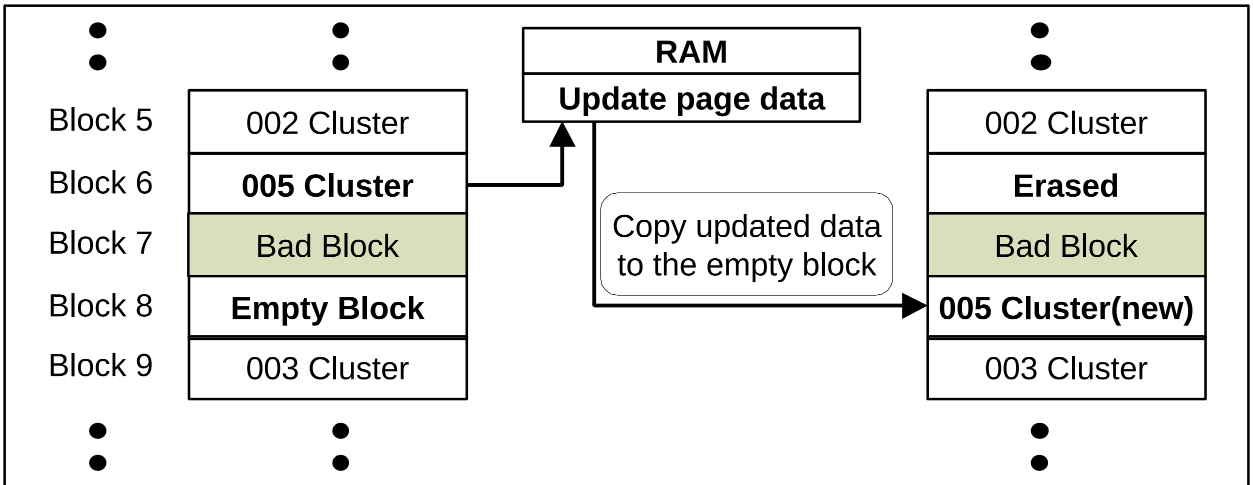
After write, a single bit can be changed due to the loss of charge, which should be detected and corrected by ECC. Regarding ECC, we generally recommend single bit correction for a sector (512bytes). Sample ECC based on Hamming Code is available.

ECC : Error Correcting Code → Hamming Code etc.
Example : 1 bit correction & 2 bit detection

2-3. Wear-Leveling Algorithm

In NAND flash memory, erase/ program operation is accomplished by the F-N (Fowler - Nordheim) tunneling algorithm using high voltage of about 20V. Erase/ program operation by the high voltage can cause oxide degradation and a failure if it is repeated without limit. By this reason, the number of program/ erase cycling(called as endurance) is guaranteed up to 1E6 cycles if the system adopts an ECC and a real time mapping algorithm. The designer should be careful not to exceed the endurance. Most applications would not need this much endurance. For write-intensive applications, a popular wear-leveling algorithm like monitoring erase/ program cycles per block or spreading write algorithm may be utilized. Updating data of a block into a physically different block with the new link list also help alleviates repetitive cycling.

FIG. 2-3 WEAR-LEVELING



The controller should monitor each block counter not to use specific blocks repeatedly.

2-4. Data Retention

Data retention, in flash memory, is classified by two modes. One is pure data retention without any bias and the other is read retention without erase/ program operation. Pure data retention of NAND Flash products is guaranteed for more than 10 years as documented by HTS (High Temperature Storage Test) data contained in Samsung`s qualification report. Repetitive reading, without erasing the blocks, hardly affects the data integrity. The number of read retention test is verified up to 1E6 cycles during qualification. The readout cycles may be reduced with high density products due to the excessively long test time. For the details, please refer to the qualification report.

2-5. Effect of P/E Cycling on Performance

NAND Flash goes through internal qualification process with endurance up to 1E6 cycles. The erase time does not change at all regardless of endurance up to 1E6 cycles.

FIG. 2-4 Cell Vth Shift vs. No of P/E Cycles

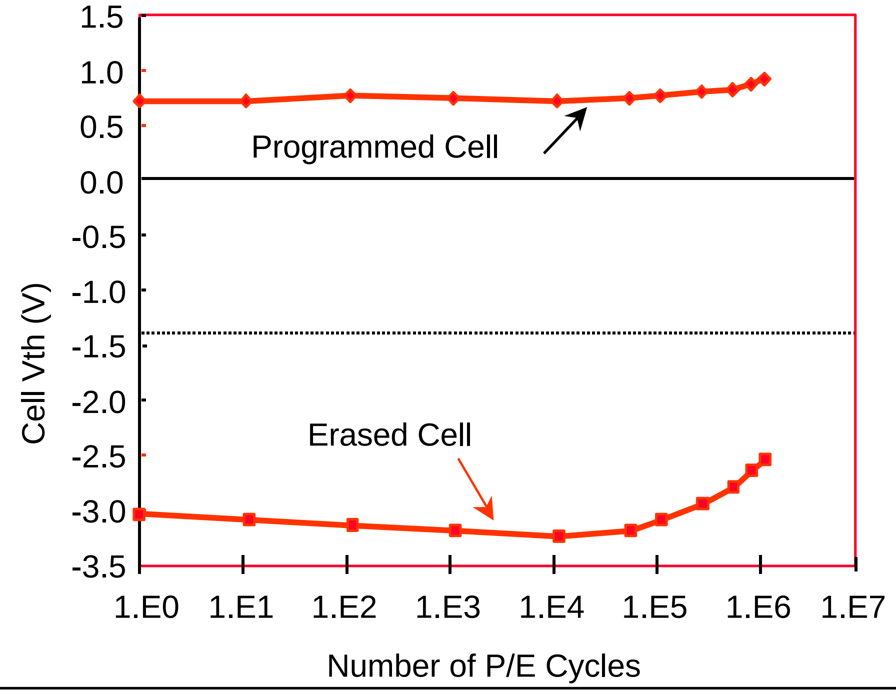
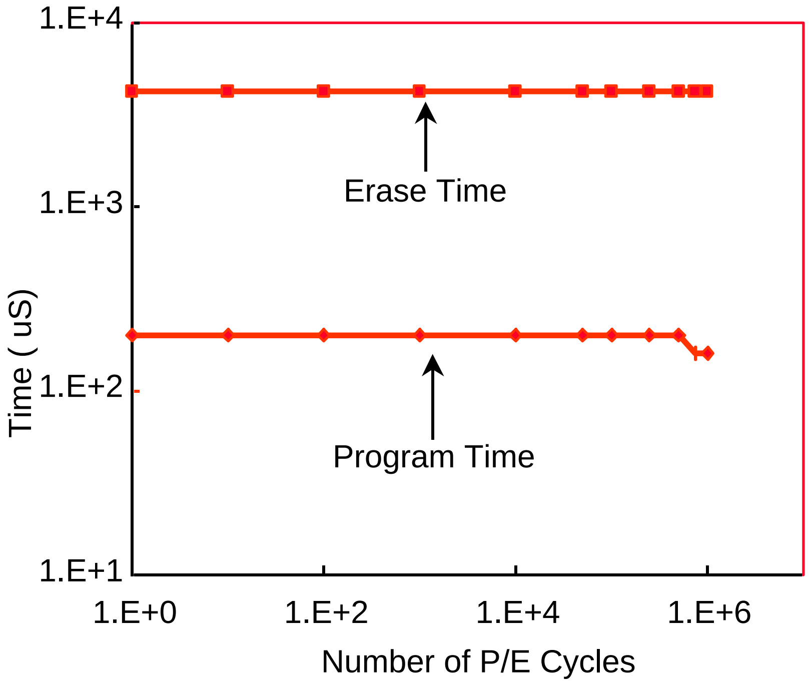
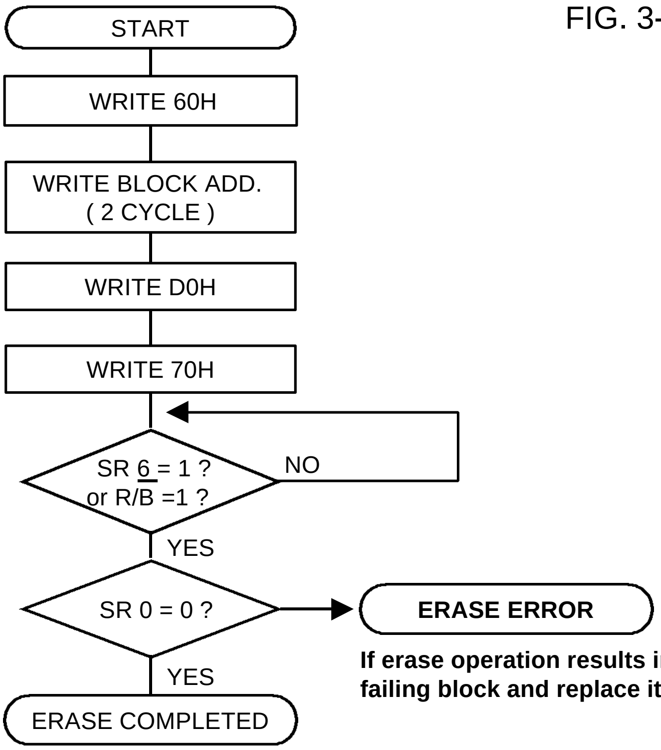


FIG. 2-5 Program/ Erase Time Variation vs. No of P/E Cycles



3. PROGRAM/ERASE ALOGORITHM

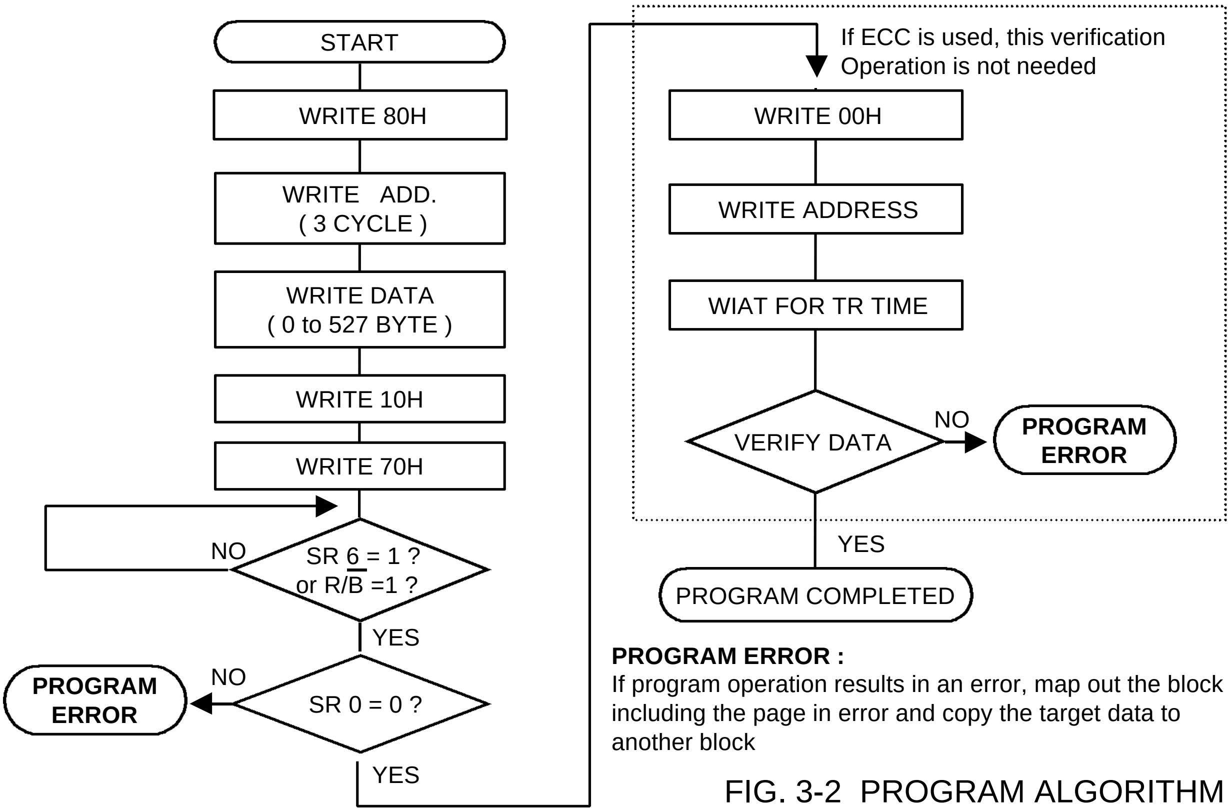
3-1. Erase Algorithm



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

FIG. 3-1 ERASE ALGORITHM

3-2. Program Algorithm



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

FIG. 3-2 PROGRAM ALGORITHM

3-3. Status Read Operation

The device automatically implements the execution and verification of the program and erase operations. The status read function is used to monitor the Ready/Busy status of the device, determines the pass/fail result of a program or erase operation. The device status is can be read through the I/O port using the RE/ clock after a “70H” command input. The Pass/Fail status on I/O 0 is only valid when the device is in the ready state. The resulting information is outlined in Table 3-1.

TABLE 3-1 STATUS REGISTER DEFINITIION

Status Register	Status	Definition
I/O0	Program/Erase	“0” : Successful Program / Erase(Pass) “1” : Error in Program / Erase(Fail)
I/O1	Reserved for Future use	“0”
I/O2		“0”
I/O3		“0”
I/O4		“0”
I/O5		“0”
I/O6	Device Operation	“0” : Busy “1” : Ready
I/O7	Write Protect	“0” : Protected “1” : Not Protected

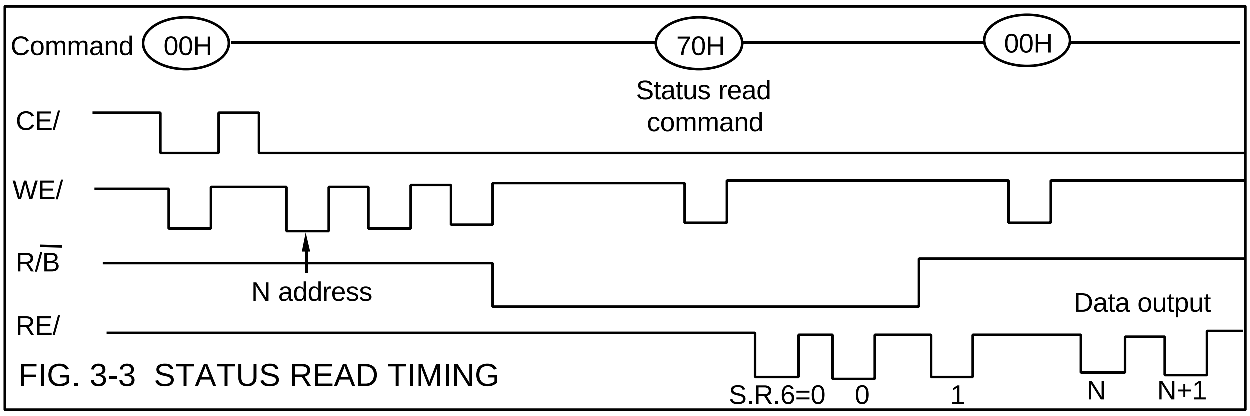


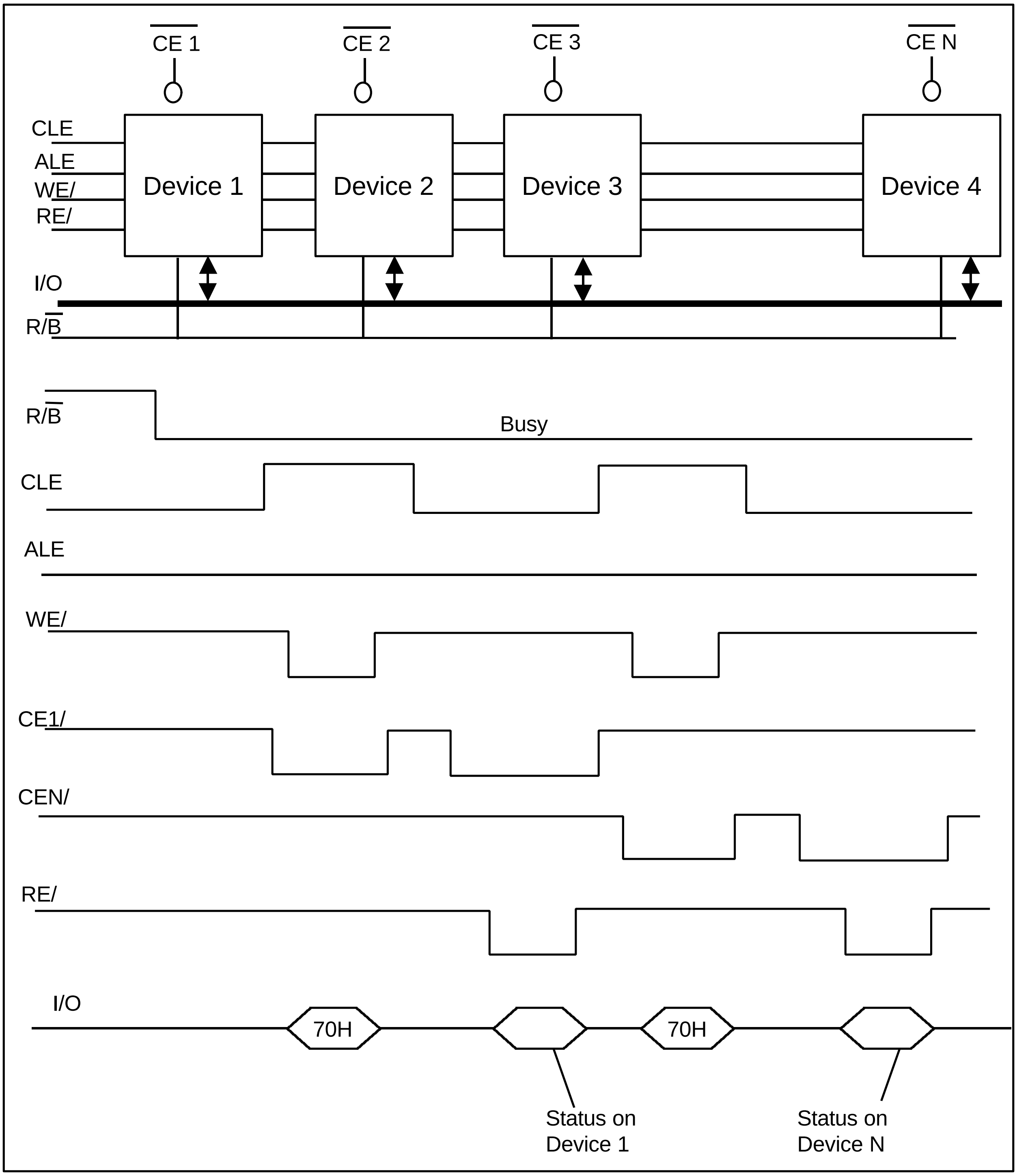
FIG. 3-3 STATUS READ TIMING

The Ready/Busy(S.R.6) status can be read by inputting the status read command ‘70H’ during the Read Mode. Once the device is set to the status Read Mode after '70H' command input, the device does not return to the read mode. In order to restore the device to the read mode after the device changes from Busy to Ready state, the read command ‘00H’ without address input must be written. In this case, the data output starts from address N.

The status bit "0" indicates that the internal verification circuit did not find error in programming. But, the internal verification circuit inside Flash detects only "1"s that are not successfully programmed to "0". If a cell is to retain the erase state of "1" but changed to “0" by any reason, this error is not detected by the internal circuit. This is called program-disturb, very rare cases. If it occurs, it is single bit error in a page. Therefore, if single bit error-detection and correction (ECC) is adopted, compare after programming is not needed. And in case of voice or graphic application, single bit error may be ignored : so for those application, neither comparing nor ECC is needed. However, if the data integrity is very critical, comparing is recommended, otherwise ECC should be used.

An application example with multiple devices is shown in Figure 3-4 :

FIG. 3-4 STATUS READ TIMING APPLICATION EXAMPLE

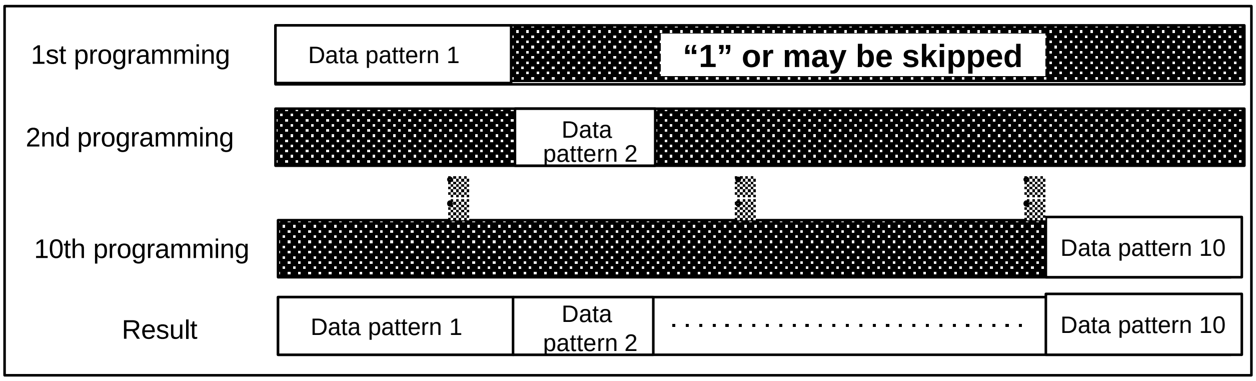


System Design Note : If the Ready/Busy pin signals of multiple devices are common-wired as shown in the diagram, the status read function can be used to determine the status of each individually selected device.

3-4. Partial Page Program

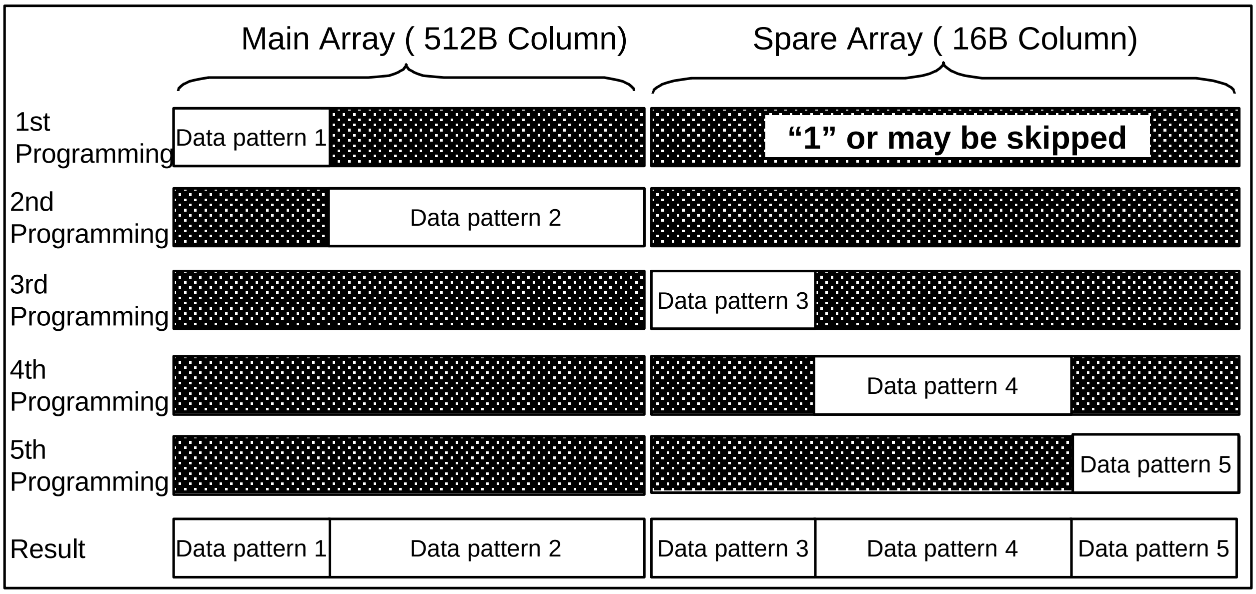
The flash memory allows a page to be divided into several segments. The number of segmentation is limited and varies by product densities as follows :

FIG. 3-5 PARTIAL PAGE PROGRAM for **4Mb/ 16Mb/ 32Mb/ 64Mb**



Regarding the data for unprogrammed or previously programmed page segments, there are two ways to take care of this. First is to input "1" for all the other bytes than the currently selected segment. Second is to skip this dummy data input cycles by specifying the column address to the start of selected segment. Upon the command input of 80H(Sequential Data Input command) the internal circuitry of NAND Flash sets the data of all page buffers to "1" by default. The page buffers with no data input keep the "1" state.Therefore only the bytes to be programmed need to be loaded and the others bytes outside data pattern do not need to be loaded,with the data of "1". If the start column address is higher than 255(as explained in Section 3-2),an appropriate pointer command should be inputted before 80H command.

FIG. 3-6 PARTIAL PAGE PROGRAM for **128Mb/ 256Mb**



The number of consecutive partial page programming operation within the same page without an intervening erase operation should not exceed 2 for main array and 3 for spare array. **If some parts of spare array were used during the main cell being programmed, the additional NOP of spare array should be restricted by 2.**

4. OPERATING WITH COMMANDS

4-1. Prohibition of Unspecified Commands

The operation commands are listed in Table 4-1. Input of a command other than those specified in Table 4-1 is prohibited. Stored data may be corrupted if an unspecified command is entered during the command cycle.

Table 4-1 COMMAND SETS

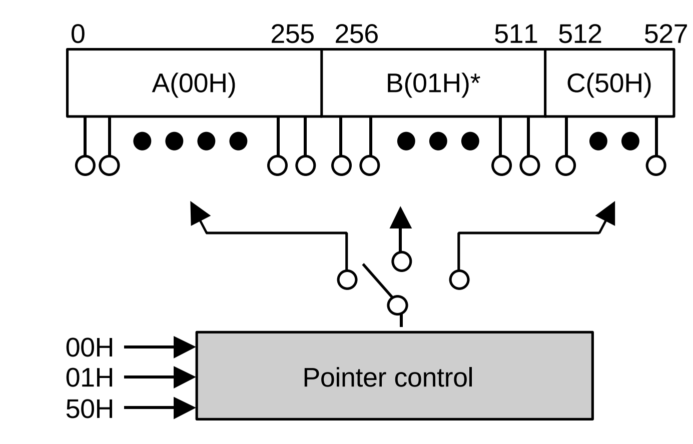
Function	1st. Cycle	2nd. Cycle	Acceptable Command during Busy State
Sequential Data Input	80H	-	
Read 1	00/01H [1]	-	
Read 2	50H [2]	-	
Read ID	90H	-	
Reset	FFH	-	○
Auto Page Program	10H	-	
Auto Block Erase	60H	D0H	
Read Status	70H	-	○

Note : 1) The 00H Command defines starting Address on the 1st half of Registers.
The 01H Command defines starting Address on the 2nd half of Registers.
After data access on the 2nd half of register by the 01H command, the status pointer is automatically moved to the 1st half register (00H) on the next cycle. 4Mb,8Mb and 16Mb NAND Flash do not support 01H command.
2) The 50H command is valid only When the SE/ is low level.(4Mb NAND Flash do not support 50H command.)

4-2. Pointer Control for '00H', '01H', '50H'

Three Read Modes are available with 32Mb~256Mb to set the destination of the pointer in either the main memory area of a page or the spare area. The pointer can be designated at any location between 0 and 255 in Read Mode 1 and between 256 and 511 in Read Mode 2 and between 512 and 527 in Read Mode 3. Figure 4-1 shows the block diagram of pointer operation for 32Mb,64Mb, 128Mb and 256Mb NAND Flash.

FIG. 4-1 POINTER CONTROL : 32Mb~256Mb

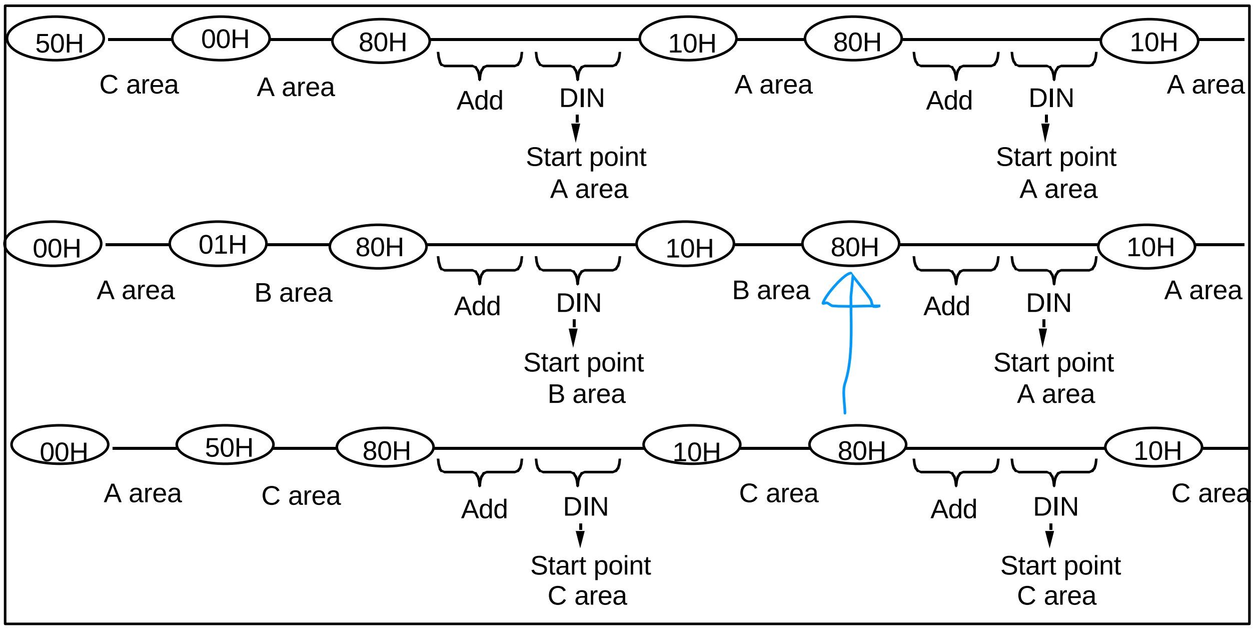


A,B : Main Memory Area
C : Spare Area

*Because 01H command is temporary, the device is set to 00H at the next page sequential read.

The pointer is set to region 'A' by the '00H' command and to region B by the “01H” command, and to region 'C' by the '50H'command. The '00H’ command must be input to set the pointer back to region 'A' when the pointer is in region 'C'.

FIG. 4-2 EXAMPLE FOR POINTER SET : 32Mb~256Mb

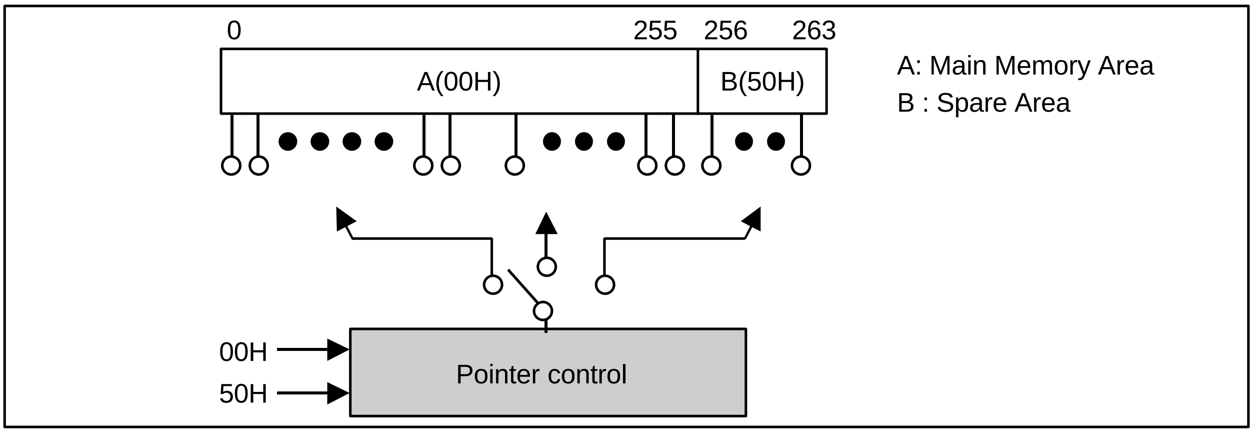


Operation	Pointer status after operation
Program/Erase	With previous 00H, Device is set to 00H Plane With previous 01H, Device is set to 00H Plane (1) With previous 50H, Device is set to 50H Plane
Reset	“00H” Plane(“A” area)
Power Up	“00H” Plane(“A” area)

Note 1) 01H command is valid just one time when it is used as a pointer for program/erase.

8Mb and 16Mb do not have 01H pointer and 4Mb has no pointer. Figure 4-3 shows the block diagram of pointer operation for 8Mb and 16Mb NAND Flash.

FIG. 4-3 POINTER CONTROL : 8Mb,16Mb



4-3. Device Status after Read/Program/Erase/Reset and Power-on

TABLE 4-2. STATUS DESCRIPTION

Execution Command	Device Status after executed CMD.
Read 1(00H or 01H)	Read 1 (00H Plane) *[1]
Read 2 (50H)	Read 2 (50H Plane)
Program	Wait *[2]
Erase	Wait *[2]
Reset	Wait *[2]
Power on	Read 1(00H Plane)

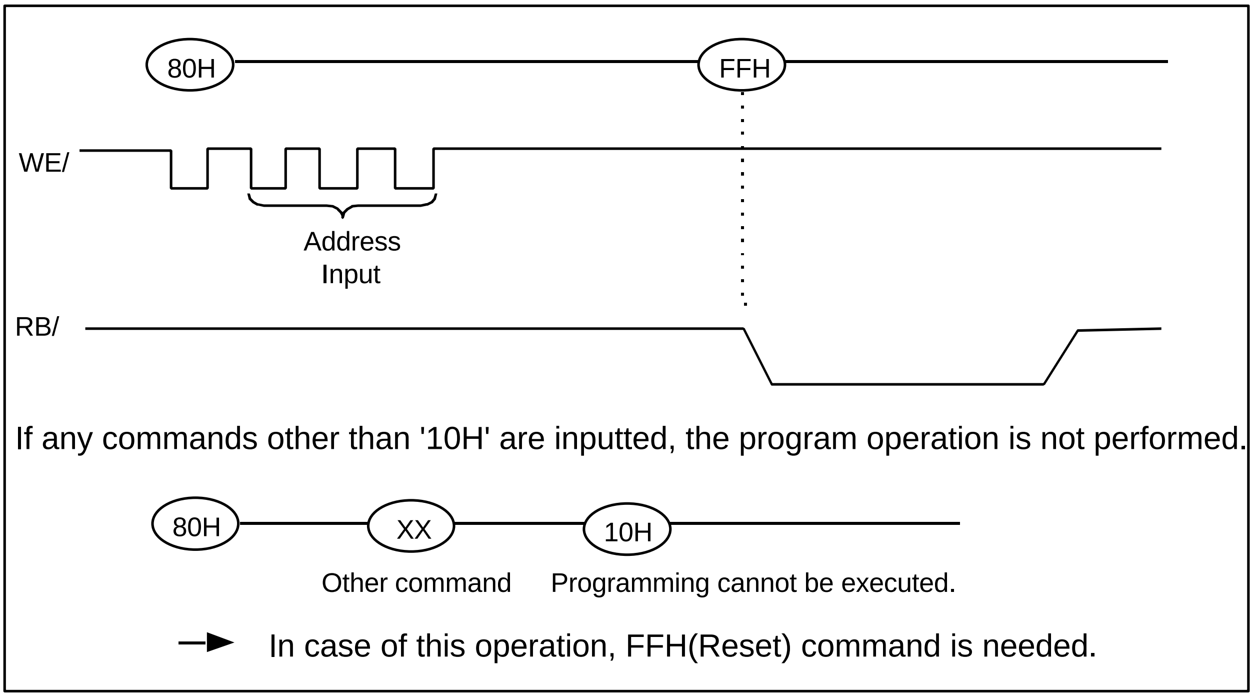
*(1) : 01H Command is valid just one time when it is used as a pointer for program/erase.

*(2) : After execution of Program/Erase/Reset command, '00H' '01H' or '50H' read commands are required to select either Read 1 or Read 2 mode.

4-4. Acceptable Commands after Sequential Input Command of '80H'

After the sequential input command (80H) input, do not input any commands other than the program execution command (10H) or the reset command (FFH) during programming.

FIG. 4-4 RIGHT USING OF PROGRAMMING COMMAND



If any commands other than '10H' are inputted, the program operation is not performed.

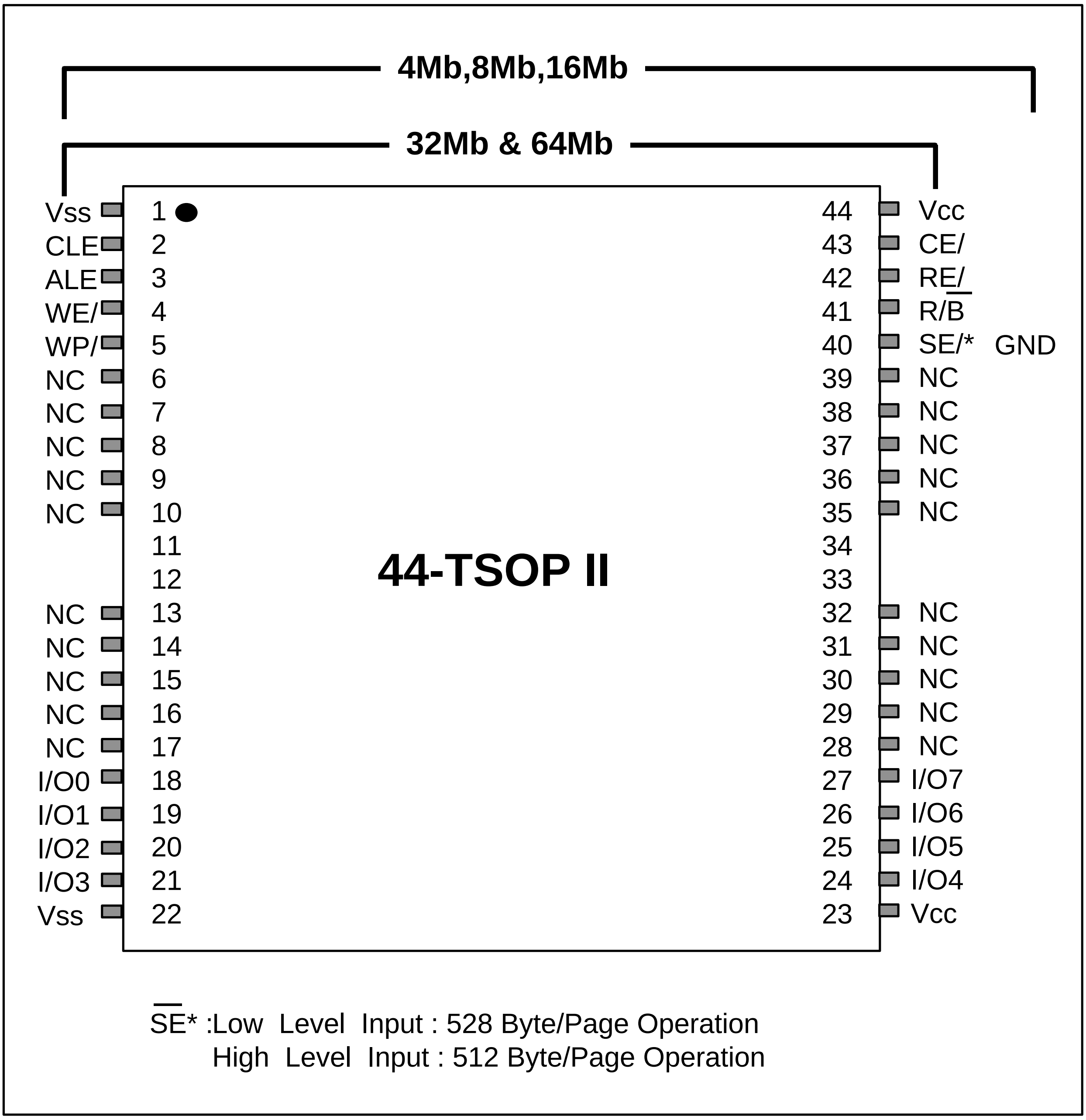
→ In case of this operation, FFH(Reset) command is needed.

5. UTILIZING THE DEVICE IN THE SAME SYSTEM DESIGN

5-1. Pin Assignment(4Mb,8Mb,16Mb,32Mb,64Mb)

SAMSUNG has been manufacturing the 4Mb,8Mb,16Mb, 32Mb and 64Mb NAND Flash in the same package, 44(40) TSOP 2. It allows the system developer to utilize the 4Mb~64Mb NAND devices in Flash design boards without any hardware changes. The following is a pin assignment comparison between the SAMSUNG`s 4Mb~64Mb NAND Flash devices.

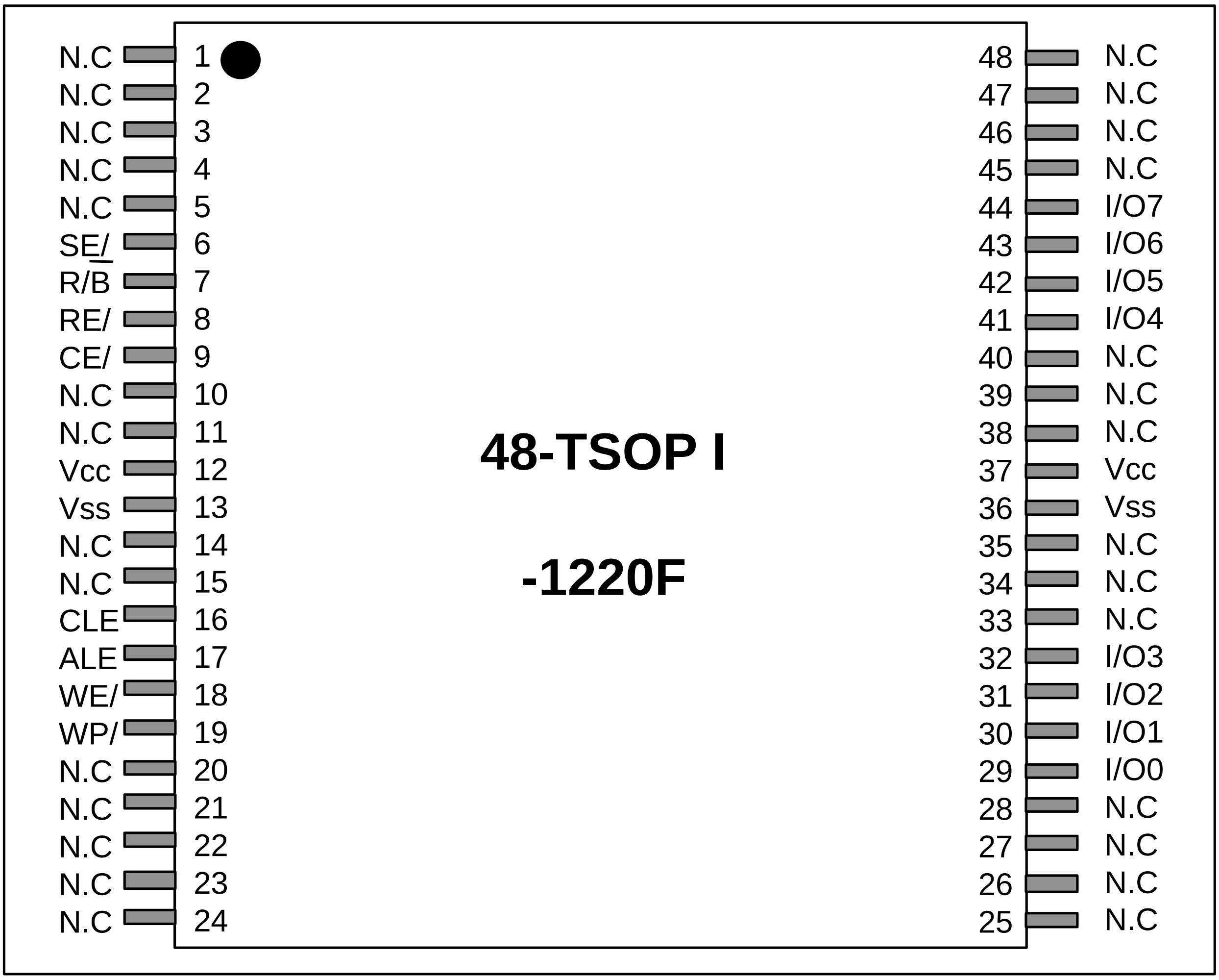
FIG. 5-1 PIN ASSIGNMENT for **4Mb,8Mb,16Mb,32Mb,64Mb**



SE* :Low Level Input : 528 Byte/Page Operation
High Level Input : 512 Byte/Page Operation

5-2. Pin Assignment(128Mb ~ 256Mb)

FIG. 5-2 PIN ASSIGNMENT for **128Mb ~ 256Mb**



5-3. Device ID Information

TABLE 5-1. DEVICE ID

Density	**SAMSUNG**			
	3.3V	5.0V	2.7V~3.6V	2.7V~5.5V
4Mb	A4H	A4H	-	A4H
8Mb	-	-	-	6EH
16Mb	EAH	64H	-	EAH
32Mb	E3H	E5H	-	E3H
64Mb	E6H	-	E6H	-
128Mb	-	-	73H	-
256Mb	-	-	75H	-

Device Part Number => 3.3V : KM29V ~, 5.0V : KM29N ~, 2.7V~3.6V : KM29U ~, 2.7V~5.5V : KM29W ~ Device ID of 256Mb and beyond may be changed according to their feature.

5-4. Addressing Map : 4Mb ~ 256Mb

The address map for the 4Mb ~ 256Mb devices are shown in table 5-2 :

TABLE 5-2. ADDRESS MAP for **4Mb ~ 256Mb**

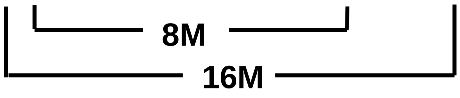
- 4M Flash

	I/O0	I/O1	I/O2	I/O3	I/O4	I/O5	I/O6	I/O7	
1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7	Col. Add.(A0~A4) Frame Add.(A5~A6)
2nd Cycle	A8	A9	A10	A11	A12	A13	A14	A15	Row Add.(A7~A11) Block Add.(A12~A18)
3rd Cycle	A16	A17	A18	X*	X*	X*	X*	X*	

X* : Can be High or Low.

- 8M,16M Flash

	I/O0	I/O1	I/O2	I/O3	I/O4	I/O5	I/O6	I/O7	
1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7	Col. Add.(A0~A7)
2nd Cycle	A8	A9	A10	A11	A12	A13	A14	A15	Row Add.(A8~A11)
3rd Cycle	A16	A17	A18	A19	A20	X*	X*	X*	

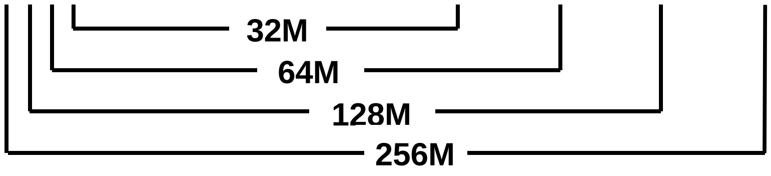


8M Block Address : A12 ~ A19(A20 : X*), 16M Block Address : A12 ~ A20

X* : Can be High or Low.

- 32M,64M,128M,256M Flash

	I/O0	I/O1	I/O2	I/O3	I/O4	I/O5	I/O6	I/O7	
1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7	Col. Add.(A0~A7)
2nd Cycle	A9	A10	A11	A12	A13	A14	A15	A16	
3rd Cycle	A17	A18	A19	A20	A21	A22	A23	A24	



32M Row Address : A9 ~ A12, 32M Block Address : A13 ~ A21(A22 ~ A24 : X*)
64M Row Address : A9 ~ A12, 64M Block Address : A13 ~ A22(A23, A24 : X*)
128M Row Address : A9 ~ A13, 128M Block Address : A14 ~ A23(A24 : X*),
256M Row Address : A9 ~ A13, 256M Block Address : A14 ~ A24
A8 : Initially set to "Low" or "High" by the 00H or 01H Command

5-5. Specification Comparison : 4Mb ~ 256Mb

TABLE 5-3. SPECIFICATION COMPARISON for **4Mb ~ 16Mb**

		4Mb	**8Mb**	**16Mb**
Organization	Page size	32 Byte(Frame size)*	(256 + 8) Byte	(256 + 8) Byte
	Block size	4K Byte	(4K + 128) Byte	(4K+128) Byte
	Number of page per block	32 pages	16 pages	16 pages
Operating voltage		3.3V,5V,2.7V~5.5V	2.7V~5.5V	3.3V,5V,2.7V~5.5V
Package		44 (40) Pin 400 mil Width 0.8mm Pitch TSOP Type II		
Main characteristics	tPROG	0.5 ms Typ.	0.25 ms Typ.	0.25 ms Typ.
	tBERS	6.0 ms Typ.	2.0 ms Typ.	2.0 ms Typ.
	tR	15 us Max.	10 us Max.	10 us Max.
	Cycle time	120 ns Min.	80 ns Min.	80 ns Min.

Note * : 1 Page = 4 Frames = 128 Byte

TABLE 5-4. SPECIFICATION COMPARISON for **32Mb ~ 256Mb**

		32Mb	**64Mb**	**128Mb/256Mb**
Organization	Page size	(512 + 16) Byte	(512 + 16) Byte	(512 + 16) Byte
	Block size	(8K + 256) Byte	(8K + 256) Byte	(16K+512) Byte
	Number of page per block	16 pages	16 pages	32 pages
Operating voltage		3.3V,5V,2.7V~5.5V	2.7V~3.6V	2.7V ~ 3.6V
Package		44 (40) Pin 400 mil 0.8mm Pitch TSOP Type II		48 TSOP I
Main characteristics	tPROG	0.25 ms Typ.	0.2 ms Typ.	0.2 ms Typ.
	tBERS	2.0 ms Typ.	2.0 ms Typ.	2.0 ms Typ.
	tR	10 us Max.	7 us Max.	10 us Max.
	Cycle time	50 ns Min.	50 ns Min.	50 ns Min.

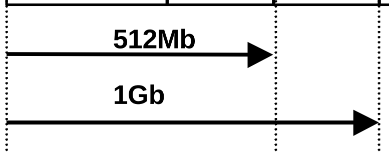
5-6. Upgrading to 512Mb ~ 1Gb

All for 256Mb except following items, hardware and software conditions, is effective in 512Mb to 1Gb flash memory.

TABLE 5-5. EXLPANSION for **256Mb/1Gb**

- Address Map

	I/O0	I/O1	I/O2	I/O3	I/O4	I/O5	I/O6	I/O7
1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd Cycle	A9	A10	A11	A12	A13	A14	A15	A16
3rd Cycle	A17	A18	A19	A20	A21	A22	A23	A24
4th Cycle	A25	A26						



A8 : Initially set to "Low" or "High" by the 00H or 01H Command

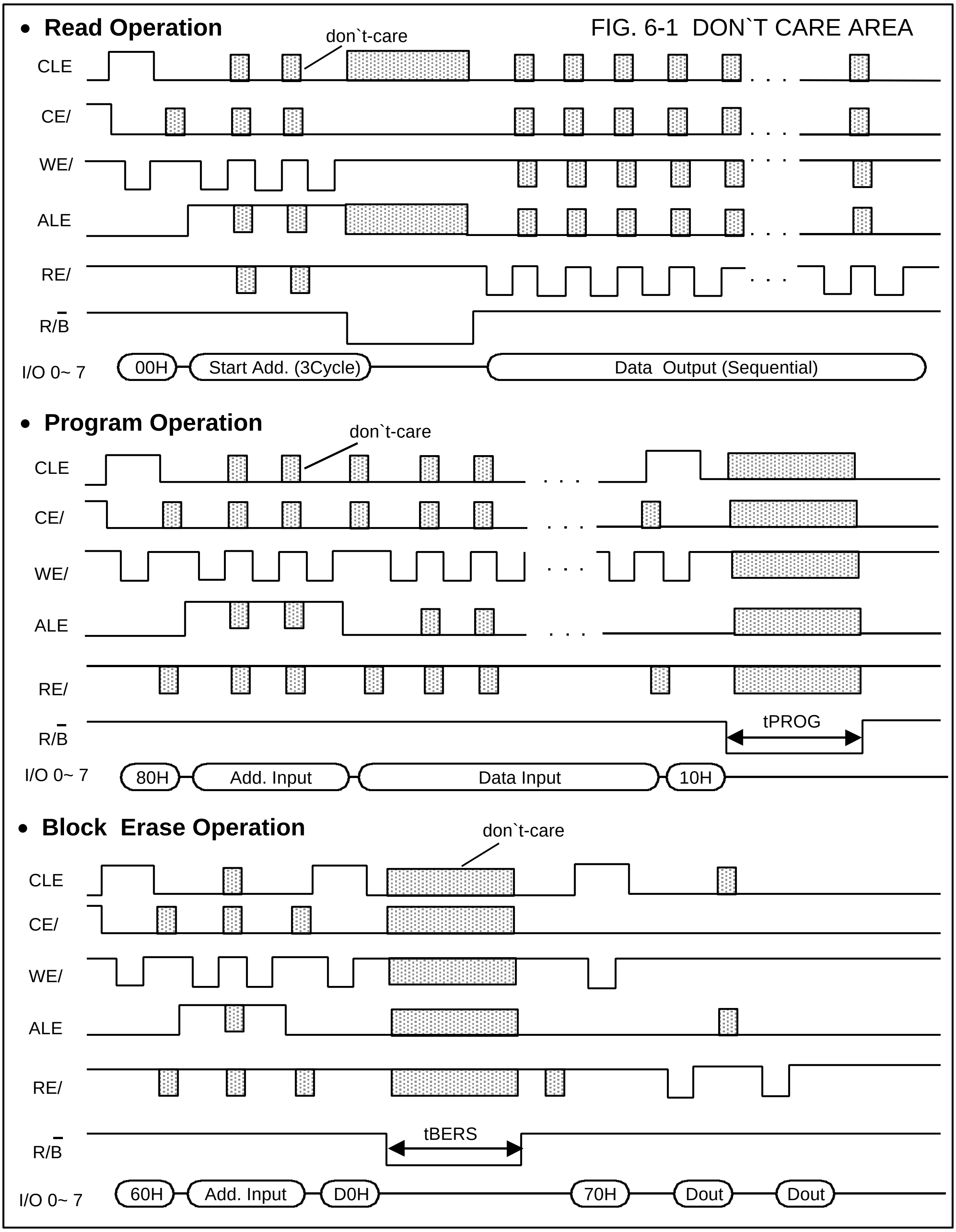
- Device ID

	512Mb	1Gb
ID	76h	79h

6. Hardware Design Considerations

6-1. Acceptable Don`t-Care Area in each Operation

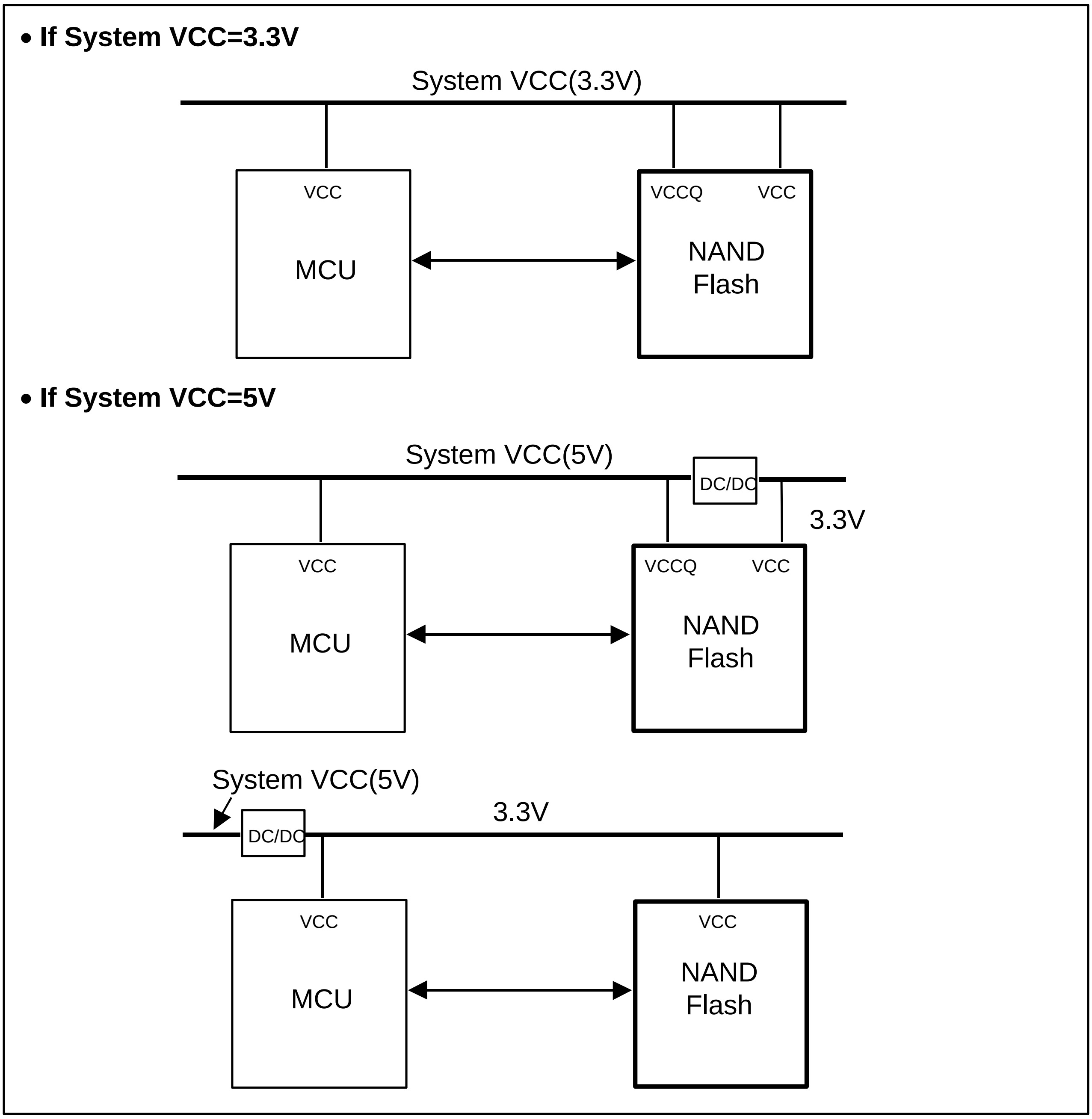
The device allows any don`t-care states during read, program and erase operation as follows.



6-2. VccQ/Vcc Power Connecting

The NAND Flash can be utilized on either 3.3V or 5.0V interface by connecting 3.3V or 5.0V power supply on the VccQ. The VccQ for I/O interface power line is electrically isolated from main chip's power line. It is not required to shift logic level between the 5V system and 3.3V NAND Flash. The Vcc power line has to be connected to 3.3V power supply on the system side. If system Vcc could change between 3.3V and 5V depending on its host system power, wide range products(2.7V~5.5V) available from 4Mb ~ 32Mb NAND Flash can be proper choices to eliminate DC-DC converter. Some densities and generations do not support 5V I/O interface. To get exact information for a specific product, refer to the specifications on Flash section of Samsung`s web-site.

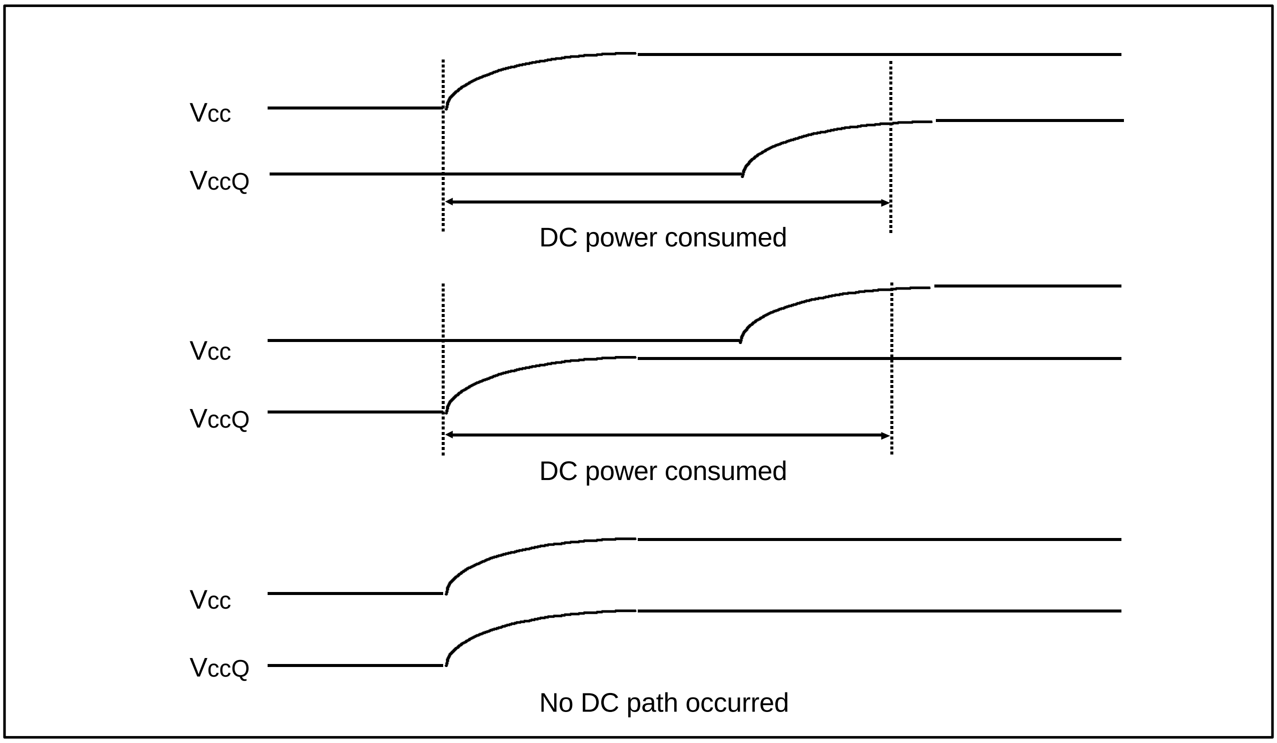
FIG. 6-2 POWER CONNECTING



- **VccQ/Vcc Power Start Up Sequence**

When the system is started up, Flash device may be powered with each of VccQ and Vcc at different instant. At this case, DC power under 20mA range may be consumed until the power reach their final level. It is recommended to start each Vcc and VccQ power simultaneously.

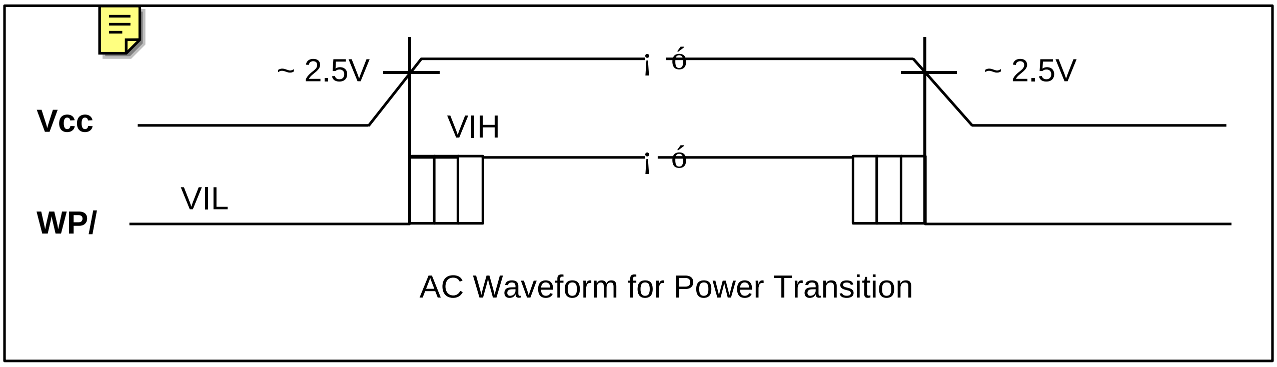
FIG. 6-3 POWER SEQUENCE



6-3. Data Protection

NAND Flash is designed to offer protection from any involuntary program/ erase during power transitions. An internal voltage detector disables all functions whenever Vcc is below about 2V. WP/ pin provides hardware protection and is recommended to be kept at VIL during power-up and power down as shown following figure. The 2step command sequence for program/ erase provides additional software protection.

FIG. 6-4 DATA PROTECTION

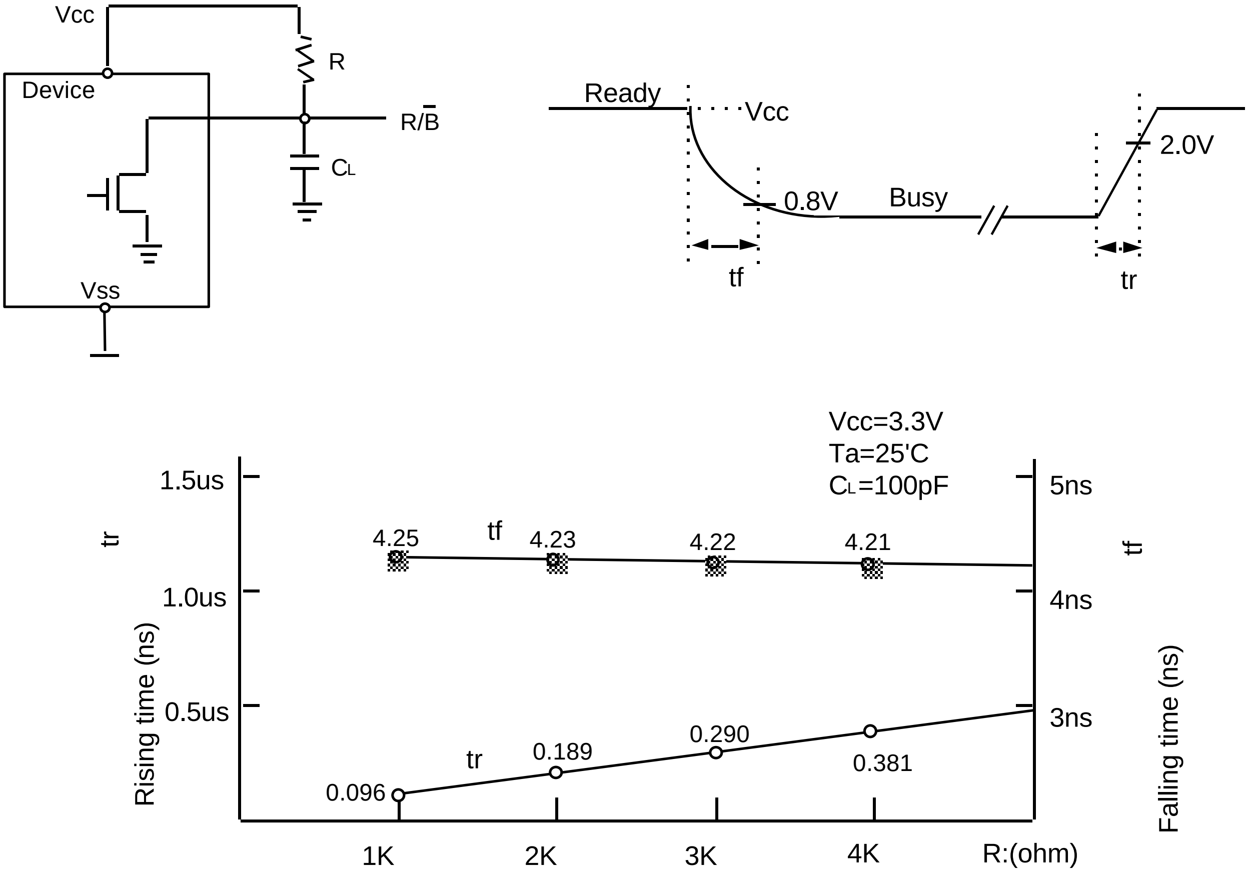


AC Waveform for Power Transition

6-4. R/ $\overline{\text{B}}$: Termination for the Ready/$\overline{\text{Busy}}$ Pin

A pull-up resistor needs to be used for termination because the Ready/$\overline{\text{Busy}}$ buffer consist of an open drain circuit.

FIG. 6-5 READY/BUSY SIGNAL SIMULATION



Falling time(tf) may vary a little by device. We recommend that you use this data as a reference when selecting a resistor value.

7. Miscellaneous

7-1. Device Behavior for WP/ Signal

Erase and program operations are compulsively reset if WP/ goes low when the command sets are given as following Fig. 7-1 and 7-2.

FIG. 7-1 WP/ GOES LOW DURING COMMAND INPUT

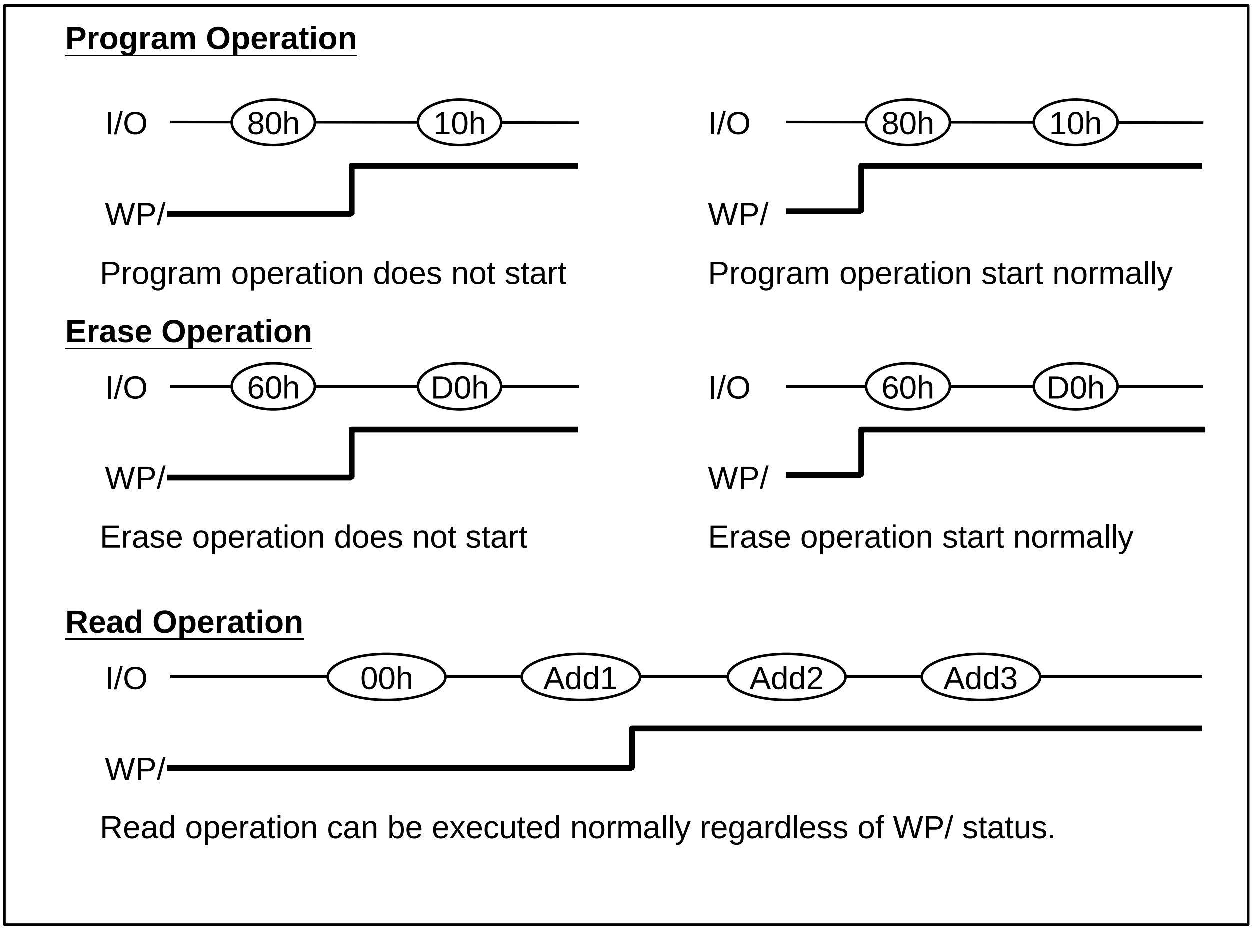
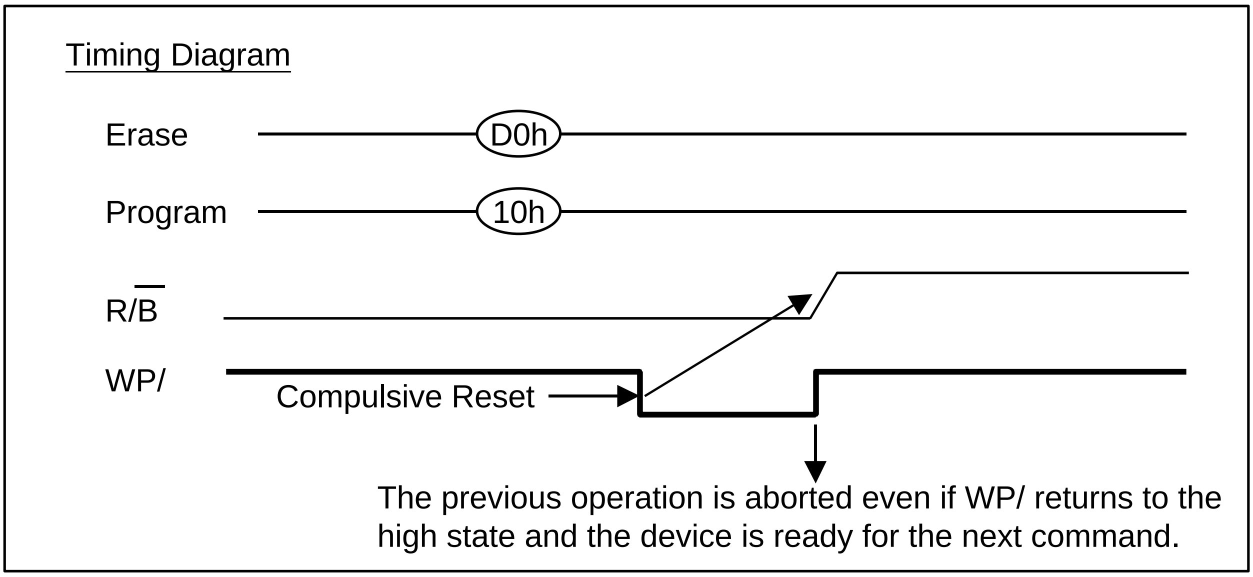

Program Operation
I/O
80h
10h
WP/
Program operation does not start
I/O
80h
10h
WP/
Program operation start normally
Erase Operation
I/O
60h
D0h
WP/
Erase operation does not start
I/O
60h
D0h
WP/
Erase operation start normally
Read Operation
I/O
00h
Add1
Add2
Add3
WP/
Read operation can be executed normally regardless of WP/ status.

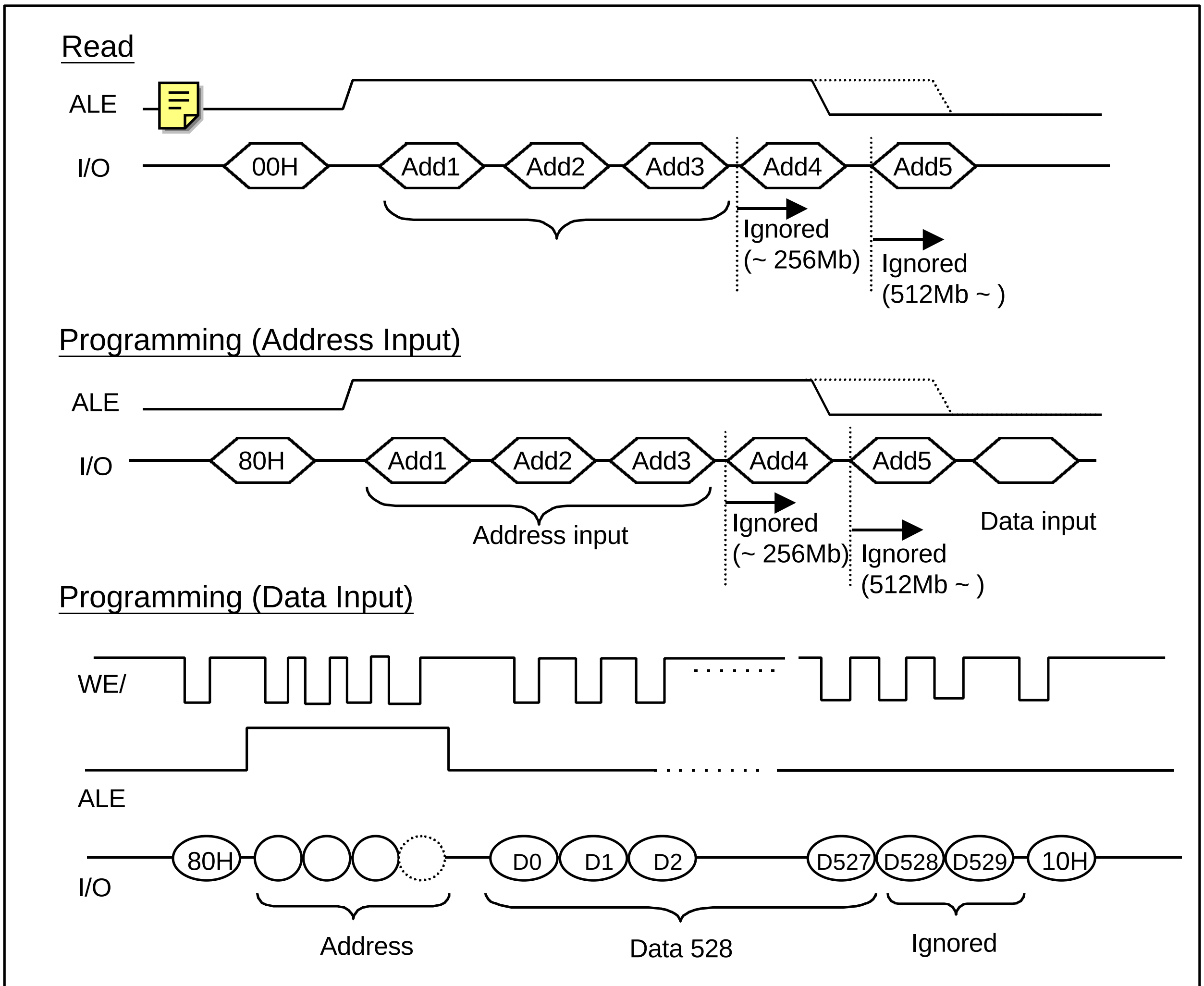
FIG. 7-2 WP/ GOES LOW DURING ERASE/PROGRAM OPERATION


Timing Diagram
Erase
D0h
Program
10h
R/$\overline{B}$
WP/
Compulsive Reset
The previous operation is aborted even if WP/ returns to the high state and the device is ready for the next command.

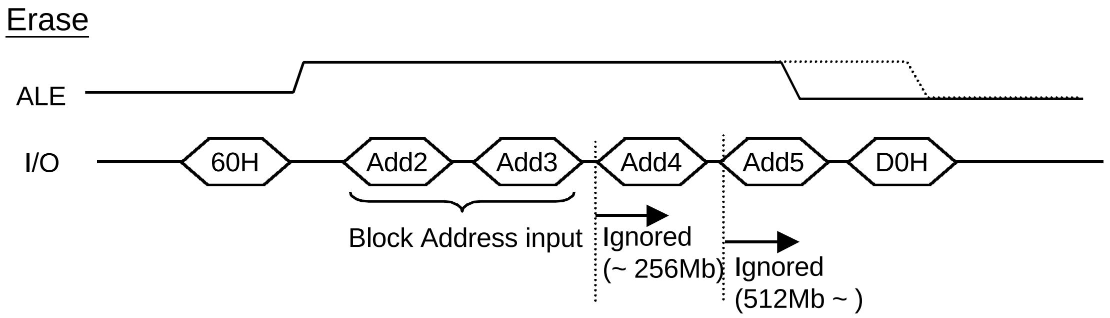
7-2. Device Behavior for Excess Data Input Cycles

In erase/program/read operation, the extra data are ignored if extra data are given as following Fig. 7-3.

FIG. 7-3 EXCESS DATA INPUT



Instead of 528 toggles with 32Mb and beyond, 8Mb and 16Mb NAND Flash function the same way after 264 toggles.



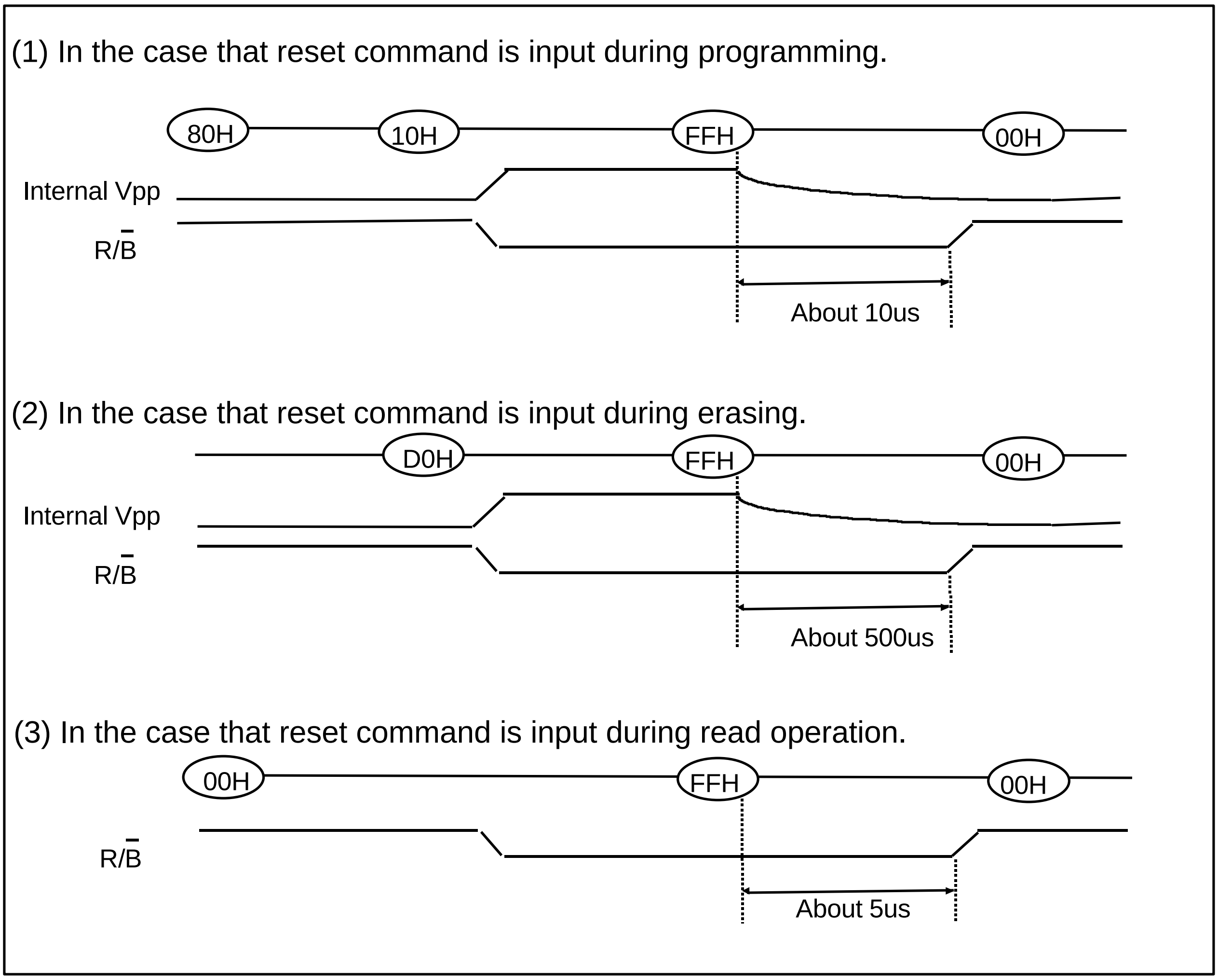
7-3. Reset Operation

Reset mode stops all operations compulsorily. For example, in the case of Program or Erase operation, the regulated voltage is discharged to 0 volts and the device will go to the wait state. The address and data registers are set after a reset as follows.

- Address Register : All '0
- Data Register : All '1
- Operation Mode : Wait State

The response after 'FFH' reset command input during each operation is as follows :

FIG. 7-4 RESET OPERATION


(1) In the case that reset command is input during programming.
80H
10H
FFH
00H
Internal Vpp
R/B̄
About 10us
(2) In the case that reset command is input during erasing.
D0H
FFH
00H
Internal Vpp
R/B̄
About 500us
(3) In the case that reset command is input during read operation.
00H
FFH
00H
R/B̄
About 5us