DS05-20887-1E

FLASH MEMORY

CMOS

64 M (8 M × 8/4 M × 16) BIT

Dual Operation

MBM29DL640E80/90/12

■ DESCRIPTION

The MBM29DL640E is a 64 M-bit, 3.0 V-only Flash memory organized as 8 Mbytes of 8 bits each or 4 Mwords of 16 bits each. The device is offered in 48-pin TSOP (I) and 63-ball FBGA packages. This device is designed to be programmed in system with 3.0 V V_{CC} supply. 12.0 V V_{PP} and 5.0 V V_{CC} are not required for write or erase operations. The device can also be reprogrammed in standard EPROM programmers.

The device is organized into four physical banks: Bank A, Bank B, Bank C and Bank D, which can be considered to be four separate memory arrays as far as certain operations are concerned. This device is the same as Fujitsu's standard 3 V only Flash memories with the additional capability of allowing a normal non-delayed read access from a non-busy bank of the array while an embedded write (either a program or an erase) operation is simultaneously taking place on the other bank.

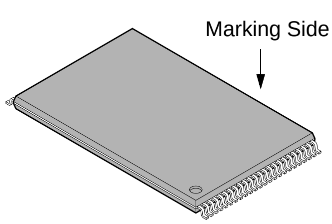
(Continued)

■ PRODUCT LINE UP

Part No.		MBM29DL640E		
Ordering Part No.	$V_{CC} = 3.3\ V^{+0.3\ V}_{-0.3\ V}$	80	—	—
	$V_{CC} = 3.0\ V^{+0.6\ V}_{-0.3\ V}$	—	90	12
Max. Address Access Time (ns)		80	90	120
Max. $\overline{CE}$ Access Time (ns)		80	90	120
Max. $\overline{OE}$ Access Time (ns)		30	35	50

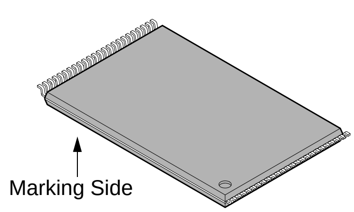
■ PACKAGES

48-pin plastic TSOP (I)



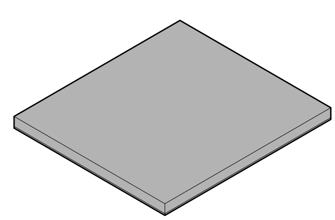
(FPT-48P-M19)

48-pin plastic TSOP (I)



(FPT-48P-M20)

63-ball plastic FBGA



(BGA-63P-M02)

(Continued)

In the device, a new design concept called FlexBank™ *1 Architecture is implemented. Using this concept the device can execute simultaneous operation between Bank 1, a bank chosen from among the four banks, and Bank 2, a bank consisting of the three remaining banks. This means that any bank can be chosen as Bank 1. (Refer to FUNCTIONAL DESCRIPTION for Simultaneous Operation.)

The standard device offers access times 80 ns, 90 ns and 120 ns, allowing operation of high-speed microprocessors without the wait. To eliminate bus contention the device has separate chip enable ($\overline{\text{CE}}$) , write enable ($\overline{\text{WE}}$) and output enable ($\overline{\text{OE}}$) controls.

This device consists of pin and command set compatible with JEDEC standard E^2PROMs. Commands are written to the command register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for the programming and erase operations. Reading data out of the device is similar to reading from 5.0 V and 12.0 V Flash or EPROM devices.

The device is programmed by executing the program command sequence. This will invoke the Embedded Program Algorithm™ which is an internal algorithm that automatically times the program pulse widths and verifies proper cell margin. Typically each sector can be programmed and verified in about 0.5 seconds. Erase is accomplished by executing the erase command sequence. This will invoke the Embedded Erase Algorithm™ which is an internal algorithm that automatically preprograms the array if it is not already programmed before executing the erase operation. During erase, the device automatically times the erase pulse widths and verifies the proper cell margin.

A sector is typically erased and verified in 1.0 second (if already completely preprogrammed) .

The device also features a sector erase architecture. The sector mode allows each sector to be erased and reprogrammed without affecting other sectors. The device is erased when shipped from the factory.

The device features single 3.0 V power supply operation for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations. A low V_{CC} detector automatically inhibits write operations on the loss of power. The end of program or erase is detected by $\overline{\text{Data}}$ Polling of DQ_7, by the Toggle Bit feature on DQ_6, or the RY/$\overline{\text{BY}}$ output pin. Once a program or erase cycle has been completed, the device internally resets to the read mode.

The device also has a hardware $\overline{\text{RESET}}$ pin. When this pin is driven low, execution of any Embedded Program Algorithm or Embedded Erase Algorithm is terminated. The internal state machine is then reset to the read mode. The $\overline{\text{RESET}}$ pin may be tied to the system reset circuitry. Therefore if a system reset occurs during the Embedded Program™ *2 Algorithm or Embedded Erase™ *2 Algorithm, the device is automatically reset to the read mode and have erroneous data stored in the address locations being programmed or erased. These locations need rewriting after the Reset. Resetting the device enables the system's microprocessor to read the boot-up firmware from the Flash memory.

Fujitsu's Flash technology combines years of EPROM and E^2PROM experience to produce the highest levels of quality, reliability, and cost effectiveness. The device memory electrically erases the entire chip or all bits within a sector simultaneously via Fowler-Nordhiem tunneling. The bytes/words are programmed one byte/word at a time using the EPROM programming mechanism of hot electron injection.

*1: FlexBank™ is a trademark of Fujitsu Limited.

*2: Embedded Erase™ and Embedded Program™ are trademarks of Advanced Micro Devices, Inc.

■ FEATURES

- **0.23 μm Process Technology**
- **Simultaneous Read/Write operations (Dual Bank)**
- **FlexBank™**
  Bank A : 8 Mbit (8 KB × 8 and 64 KB × 15)
  Bank B : 24 Mbit (64 KB × 48)
  Bank C : 24 Mbit (64 KB × 48)
  Bank D : 8 Mbit (8 KB × 8 and 64 KB × 15)
  Two virtual Banks are chosen from the combination of four physical banks (Refer to Table 9, 10)
  Host system can program or erase in one bank, and then read immediately and simultaneously from the other bank with zero latency between read and write operations.
  Read-while-erase
  Read-while-program
- **Single 3.0 V read, program, and erase**
  Minimized system level power requirements
- **Compatible with JEDEC-standard commands**
  Uses the same software commands as E^2PROMs
- **Compatible with JEDEC-standard world-wide pinouts**
  48-pin TSOP (I) (Package suffix : TN – Normal Bend Type, TR – Reversed Bend Type)
  63-ball FBGA (Package suffix : PBT)
- **Minimum 100,000 program/erase cycles**
- **High performance**
  80 ns maximum access time
- **Sector erase architecture**
  Sixteen 4 Kword and one hundred twenty-six 32 Kword sectors in word mode
  Sixteen 8 Kbyte and one hundred twenty-six 64 Kbyte sectors in byte mode
  Any combination of sectors can be concurrently erased. It also supports full chip erase.
- **Hidden ROM (Hi-ROM) region**
  256 byte of Hi-ROM, accessible through a new “Hi-ROM Enable” command sequence
  Factory serialized and protected to provide a secure electronic serial number (ESN)
- **$\overline{\text{WP}}$/ACC input pin**
  At V_{IL}, allows protection of “outermost” 2 × 8 Kbytes on both ends of boot sectors, regardless of sector protection/unprotection status
  At V_{IH}, allows removal of boot sector protection
  At V_{ACC}, increases program performance
- **Embedded Erase™ Algorithms**
  Automatically preprograms and erases the chip or any sector
- **Embedded Program™ Algorithms**
  Automatically writes and verifies data at specified address

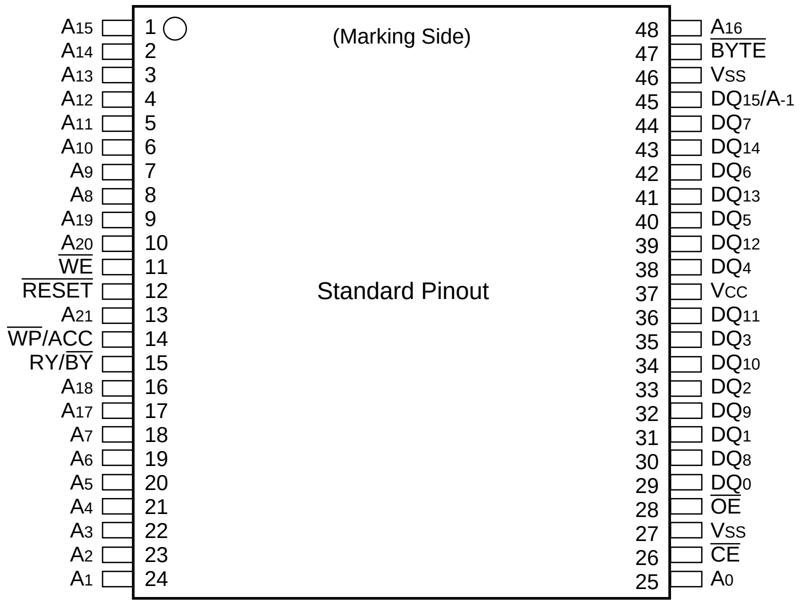
(Continued)

(Continued)

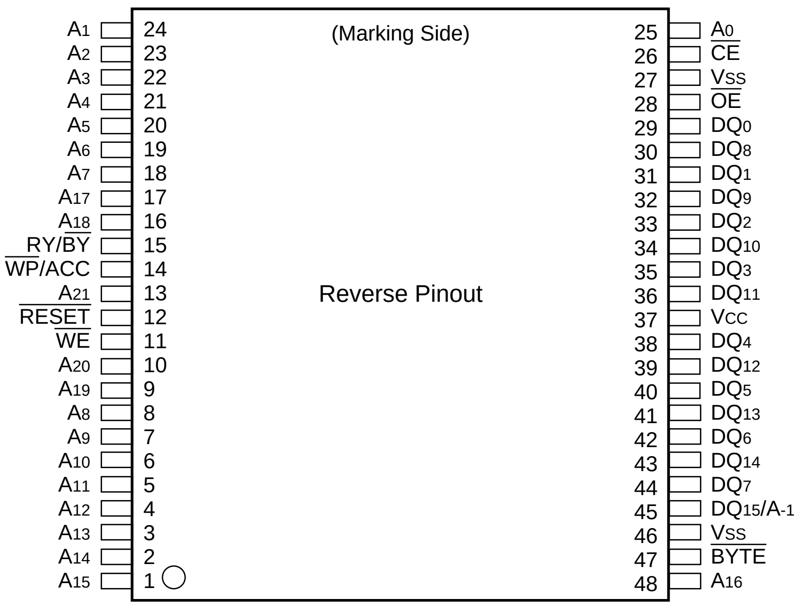
- **$\overline{\text{Data}}$ Polling and Toggle Bit feature for detection of program or erase cycle completion**
- **Ready/Busy output (RY/$\overline{\text{BY}}$)**
  Hardware method for detection of program or erase cycle completion
- **Automatic sleep mode**
  When addresses remain stable, the device automatically switches itself to low power mode.
- **Low V_{CC} write inhibit $\leq$ 2.5 V**
- **Program Suspend/Resume**
  Suspends the program operation to allow a read in another byte
- **Erase Suspend/Resume**
  Suspends the erase operation to allow a read data and/or program in another sector within the same device
- **Sector group protection**
  Hardware method disables any combination of sector groups from program or erase operations
- **Sector Group Protection Set function by Extended sector group protection command**
- **Fast Programming Function by Extended Command**
- **Temporary sector group unprotection**
  Temporary sector group unprotection via the $\overline{\text{RESET}}$ pin.
- **In accordance with CFI (Common Flash Memory Interface)**

■ PIN ASSIGNMENTS

TSOP (I)



(FPT-48P-M19)

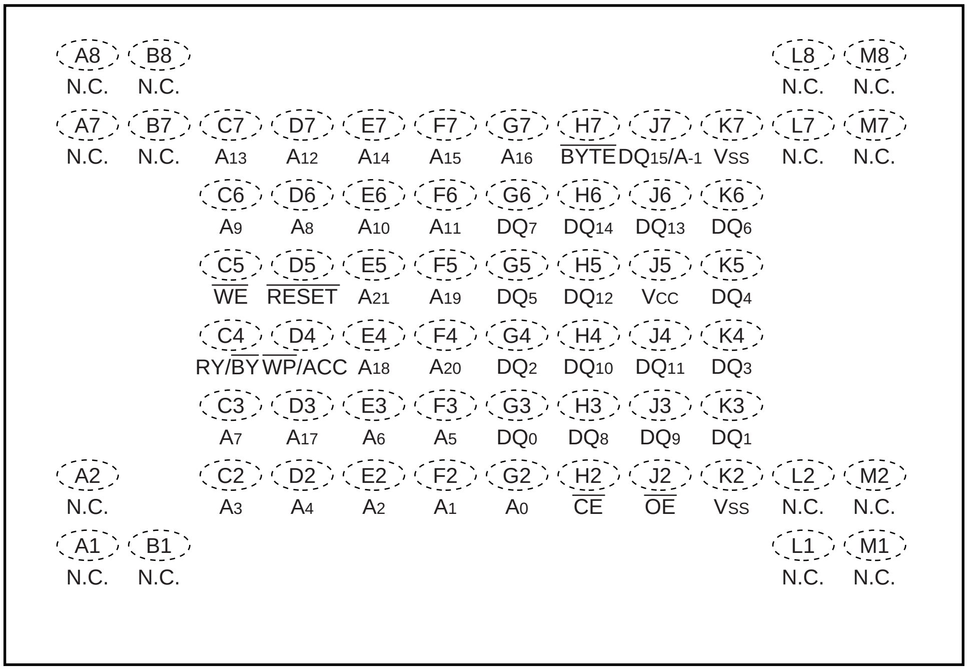


(FPT-48P-M20)

(Continued)

(Continued)

FBGA
(TOP VIEW)
(Marking Side)



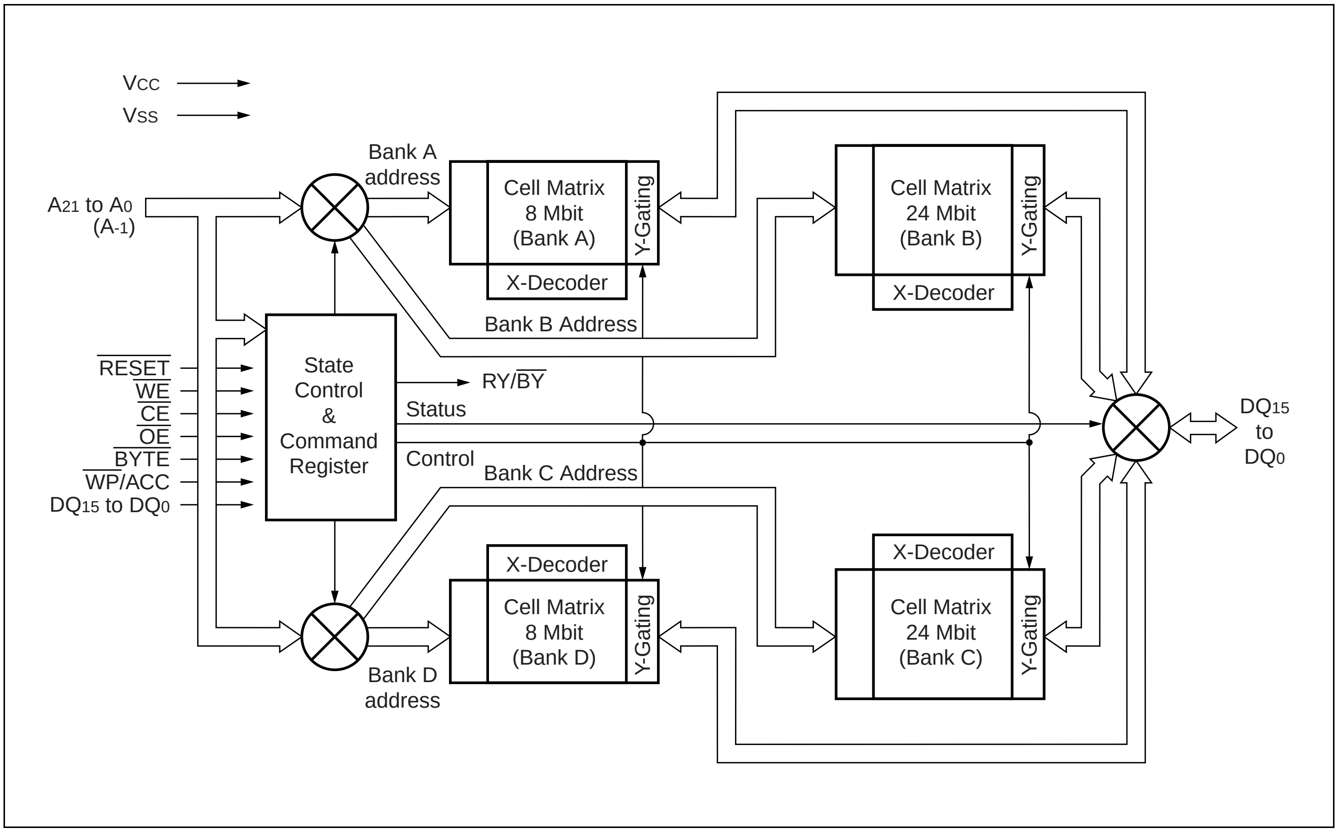
(BGA-63P-M02)

■ PIN DESCRIPTIONS

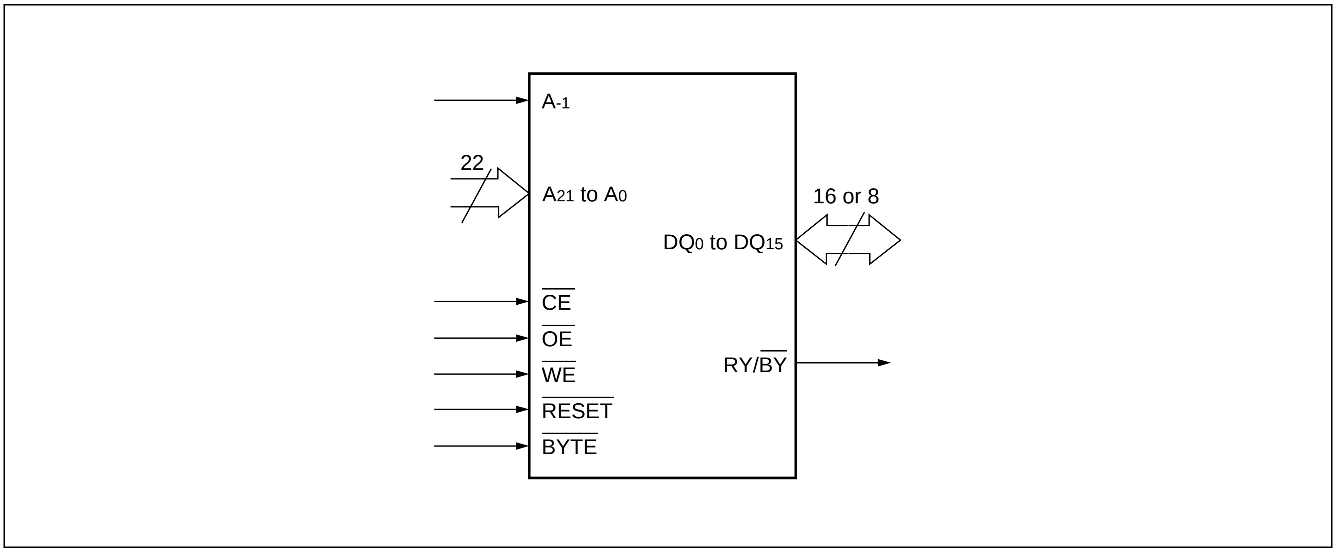
Table 1　MBM29DL640E Pin Configuration

Pin	Function
A_{21} to A_0, A_{-1}	Address Input
DQ_{15} to DQ_0	Data Input/Output
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
RY/$\overline{BY}$	Ready/Busy Output
$\overline{RESET}$	Hardware Reset Pin/Temporary Sector Group Unprotection
$\overline{BYTE}$	Selects 8-bit or 16-bit mode
$\overline{WP}$/ACC	Hardware Write Protection/Program Acceleration
V_{SS}	Device Ground
V_{CC}	Device Power Supply
N.C.	No Internal Connection

■ BLOCK DIAGRAM



■ LOGIC SYMBOL



■ DEVICE BUS OPERATION

Table 2 MBM29DL640E User Bus Operations ($\overline{\text{BYTE}}$ = V_{IH})

Operation	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{WE}}$	A_0	A_1	A_2	A_3	A_6	A_9	DQ_{15} to DQ_0	$\overline{\text{RESET}}$	$\overline{\text{WP}}$/ ACC
Auto-Select Manufacturer Code *1	L	L	H	L	L	L	L	L	V_{ID}	Code	H	X
Auto-Select Device Code *1	L	L	H	H	L	L	L	L	V_{ID}	Code	H	X
Extended Auto-Select Device Code *1	L	L	H	L	H	H	H	L	V_{ID}	Code	H	X
	L	L	H	H	H	H	H	L	V_{ID}	Code	H	X
Read *3	L	L	H	A_0	A_1	A_2	A_3	A_6	A_9	D_{OUT}	H	X
Standby	H	X	X	X	X	X	X	X	X	High-Z	H	X
Output Disable	L	H	H	X	X	X	X	X	X	High-Z	H	X
Write (Program/Erase)	L	H	L	A_0	A_1	A_2	A_3	A_6	A_9	D_{IN}	H	X
Enable Sector Group Protection *2, *4	L	V_{ID}	$\sqcup$	L	H	L	L	L	V_{ID}	X	H	X
Verify Sector Group Protection *2, *4	L	L	H	L	H	L	L	L	V_{ID}	Code	H	X
Temporary Sector Group Unprotection *5	X	X	X	X	X	X	X	X	X	X	V_{ID}	X
Reset (Hardware) /Standby	X	X	X	X	X	X	X	X	X	High-Z	L	X
Boot Block Sector Write Protection *6	X	X	X	X	X	X	X	X	X	X	X	L

Legend : L = V_{IL}, H = V_{IH}, X = V_{IL} or V_{IH}, $\sqcup$ = Pulse input. See DC Characteristics for voltage levels.

*1: Manufacturer and device codes may also be accessed via a command register write sequence. SeeTable 4.

*2: Refer to section on Sector Group Protection.

*3: $\overline{\text{WE}}$ can be V_{IL} if $\overline{\text{OE}}$ is V_{IL}, $\overline{\text{OE}}$ at V_{IH} initiates the write operations.

*4: V_{CC} = 3.3 V ± 10%

*5: It is also used for the extended sector group protection.

*6: Protect “outermost” 2 × 8 Kbytes (4 Kwords) on both ends of the boot block sectors.

(Continued)

(Continued)

Table 3 MBM29DL640E User Bus Operations ($\overline{BYTE}$ = V_{IL})

Operation	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_{15} /A_{-1}	A_0	A_1	A_2	A_3	A_6	A_9	DQ_7 to DQ_0	$\overline{RESET}$	$\overline{WP}$/ ACC
Auto-Select Manufacturer Code *1	L	L	H	L	L	L	L	L	L	V_{ID}	Code	H	X
Auto-Select Device Code *1	L	L	H	L	H	L	L	L	L	V_{ID}	Code	H	X
Extended Auto-Select Device Code *1	L	L	H	L	L	H	H	H	L	V_{ID}	Code	H	X
	L	L	H	L	H	H	H	H	L	V_{ID}	Code	H	X
Read *3	L	L	H	A_{-1}	A_0	A_1	A_2	A_3	A_6	A_9	D_{OUT}	H	X
Standby	H	X	X	X	X	X	X	X	X	X	High-Z	H	X
Output Disable	L	H	H	X	X	X	X	X	X	X	High-Z	H	X
Write (Program/Erase)	L	H	L	A_{-1}	A_0	A_1	A_2	A_3	A_6	A_9	D_{IN}	H	X
Enable Sector Group Protection *2, *4	L	V_{ID}	⊔	L	L	H	L	L	L	V_{ID}	X	H	X
Verify Sector Group Protection *2, *4	L	L	H	L	L	H	L	L	L	V_{ID}	Code	H	X
Temporary Sector Group Unprotection *5	X	X	X	X	X	X	X	X	X	X	X	V_{ID}	X
Reset (Hardware) / Standby	X	X	X	X	X	X	X	X	X	X	High-Z	L	X
Boot Block Sector Write Protection *6	X	X	X	X	X	X	X	X	X	X	X	X	L

Legend : L = V_{IL}, H = V_{IH}, X = V_{IL} or V_{IH}, ⊔ = Pulse input. See DC Characteristics for voltage levels.

*1: Manufacturer and device codes may also be accessed via a command register write sequence. See Table 4.

*2: Refer to section on Sector Group Protection.

*3: $\overline{WE}$ can be V_{IL} if $\overline{OE}$ is V_{IL}, $\overline{OE}$ at V_{IH} initiates the write operations.

*4: V_{CC} = 3.3 V ± 10%

*5: Also used for extended sector group protection.

*6: Protects “outermost” 2 × 8 Kbytes (4 Kwords) on both ends of the boot block sectors.

Table 4 MBM29DL640E Command Definitions

Command Sequence		Bus Write Cycles Req'd	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Fourth Bus Read/Write Cycle		Fifth Bus Write Cycle		Sixth Bus Write Cycle	
			Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read/ Reset	Word	1	XXXh	F0h	—	—	—	—	—	—	—	—	—	—
	Byte													
Read/ Reset	Word	3	555h	AAh	2AAh	55h	555h	F0h	RA	RD	—	—	—	—
	Byte		AAAh		555h		AAAh							
Autoselect	Word	3	555h	AAh	2AAh	55h	(BA) 555h	90h	—	—	—	—	—	—
	Byte		AAAh		555h		(BA) AAAh							
Program	Word	4	555h	AAh	2AAh	55h	555h	A0h	PA	PD	—	—	—	—
	Byte		AAAh		555h		AAAh							
Program Suspend		1	BA	B0h	—	—	—	—	—	—	—	—	—	—
Program Resume		1	BA	30h	—	—	—	—	—	—	—	—	—	—
Chip Erase	Word	6	555h	AAh	2AAh	55h	555h	80h	555h	AAh	2AAh	55h	555h	10h
	Byte		AAAh		555h		AAAh		AAAh		555h		AAAh	
Sector Erase	Word	6	555h	AAh	2AAh	55h	555h	80h	555h	AAh	2AAh	55h	SA	30h
	Byte		AAAh		555h		AAAh		AAAh		555h			
Erase Suspend		1	BA	B0h	—	—	—	—	—	—	—	—	—	—
Erase Resume		1	BA	30h	—	—	—	—	—	—	—	—	—	—
Set to Fast Mode	Word	3	555h	AAh	2AAh	55h	555h	20h	—	—	—	—	—	—
	Byte		AAAh		555h		AAAh							
Fast Program *1	Word	2	XXXh	A0h	PA	PD	—	—	—	—	—	—	—	—
	Byte		XXXh											
Reset from Fast Mode *1	Word	2	BA	90h	XXXh	*4 F0h	—	—	—	—	—	—	—	—
	Byte		BA		XXXh									
Extended Sector Group Protection *2	Word	4	XXXh	60h	SPA	60h	SPA	40h	SPA	SD	—	—	—	—
	Byte													
Query	Word	1	(BA) 55h	98h	—	—	—	—	—	—	—	—	—	—
	Byte		(BA) AAh											

(Continued)

(Continued)

Command Sequence		Bus Write Cycles Req'd	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Fourth Bus Read/Write Cycle		Fifth Bus Write Cycle		Sixth Bus Write Cycle	
			Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Hi-ROM Entry	Word	3	555h	AAh	2AAh	55h	555h	88h	—	—	—	—	—	—
	Byte		AAAh		555h		AAAh							
Hi-ROM Program *3	Word	4	555h	AAh	2AAh	55h	555h	A0h	(HRA) PA	PD	—	—	—	—
	Byte		AAAh		555h		AAAh							
Hi-ROM Exit *3	Word	4	555h	AAh	2AAh	55h	(HRBA) 555h	90h	XXXh	00h	—	—	—	—
	Byte		AAAh		555h		(HRBA) AAAh							

*1: This command is valid during Fast Mode.

*2: This command is valid while $\overline{RESET} = V_{ID}$.

*3: This command is valid during Hi-ROM mode.

*4: The data “00h” is also acceptable.

Notes : 1. Address bits A_{21} to A_{11} = X = “H” or “L” for all address commands except or Program Address (PA) , Sector Address (SA) , Bank Address (BA) and Sector Group Address (SPA) .

2. Bus operations are defined in Tables 2 and 3.
3. RA = Address of the memory location to be read
   PA = Address of the memory location to be programmed
   Addresses are latched on the falling edge of the write pulse.
   SA = Address of the sector to be erased. The combination of A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13} and A_{12} will uniquely select any sector.
   BA = Bank Address. Address setted by A_{21}, A_{20}, A_{19} will select Bank A, Bank B, Bank C and Bank D.
4. RD = Data read from location RA during read operation.
   PD = Data to be programmed at location PA. Data is latched on the falling edge of write pulse.
5. SPA = Sector group address to be protected. Set sector group address and (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) .
   SD = Sector group protection verify data. Output 01h at protected sector group addresses and output 00h at unprotected sector group addresses.
6. HRA = Address of the Hi-ROM area Word Mode : 000000h to 00007Fh
   Byte Mode : 000000h to 0000FFh
7. HRBA = Bank Address of the Hi-ROM area ($A_{21} = A_{20} = A_{19} = V_{IL}$)
8. The system should generate the following address patterns:
   Word Mode : 555h or 2AAh to addresses A_{10} to A_0
   Byte Mode : AAAh or 555h to addresses A_{10} to A_0, and A_{-1}
9. Both Read/Reset commands are functionally equivalent, resetting the device to the read mode.

Table 5.1 MBM29DL640E Sector Group Protection Verify Autoselect Codes

Type		A_{21} to A_{12}	A_6	A_3	A_2	A_1	A_0	A_{-1}[*1]	Code (HEX)
Manufacture’s Code		BA[*3]	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IL}	04h
Device Code	Byte	BA[*3]	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	7Eh
	Word							X	227Eh
Extended Device Code[*4]	Byte	BA[*3]	V_{IL}	V_{IH}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	02h
	Word							X	2202h
	Byte	BA[*3]	V_{IL}	V_{IH}	V_{IH}	V_{IH}	V_{IH}	V_{IL}	01h
	Word							X	2201h
Sector Group Protection		Sector Group Addresses	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	01h[*2]

*1 : A_{-1} is for Byte mode.

*2 : Outputs 01h at protected sector group addresses and outputs 00h at unprotected sector group addresses.

*3 : When V_{ID} is applied to A_9, both Bank 1 and Bank 2 are put into Autoselect mode, which makes simultaneous operation unable to be executed. Consequently, specifying the bank address is not required. However, the bank address needs to be indicated when Autoselect mode is read out at command mode, because then it becomes possible to activate simultaneous operation.

*4 : At WORD mode, a read cycle at address (BA) 01h (at BYTE mode, (BA) 02h) outputs device code. When 227Eh (at BYTE mode, 7Eh) is output, it indicates that two additional codes, called Extended Device Codes, will be required. Therefore the system may continue reading out these Extended Device Codes at the address of (BA) 0Eh (at BYTE mode, (BA) 1Ch) , as well as at (BA) 0Fh (at BYTE mode, (BA) 1Eh) .

Table 5.2 Expanded Autoselect Code Table

Type		Code	DQ_{15}	DQ_{14}	DQ_{13}	DQ_{12}	DQ_{11}	DQ_{10}	DQ_9	DQ_8	DQ_7	DQ_6	DQ_5	DQ_4	DQ_3	DQ_2	DQ_1	DQ_0
Manufacturer’s Code		04h	A_{-1}/0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Device Code	(B)	7Eh	A_{-1}	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	0	1	1	1	1	1	1	0
	(W)	227Eh	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0
Extended Device Code	(B)	02h	A_{-1}	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	0	0	0	0	0	0	1	0
	(W)	2202h	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0
	(B)	01h	A_{-1}	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	0	0	0	0	0	0	0	1
	(W)	2201h	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1
Sector Group Protection		01h	A_{-1}/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(B) : Byte mode
(W) : Word mode

■ FLEXIBLE SECTOR-ERASE ARCHITECTURE

Table 6.1 Sector Address Tables (Bank A)

Bank	Sector	Sector Address: Bank Address A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	Sector Size (Kbytes/ Kwords)	(× 8) Address Range	(× 16) Address Range
Bank A	SA0	0	0	0	0	0	0	0	0	0	0	8/4	000000h to 001FFFh	000000h to 000FFFh
	SA1	0	0	0	0	0	0	0	0	0	1	8/4	002000h to 003FFFh	001000h to 001FFFh
	SA2	0	0	0	0	0	0	0	0	1	0	8/4	004000h to 005FFFh	002000h to 002FFFh
	SA3	0	0	0	0	0	0	0	0	1	1	8/4	006000h to 007FFFh	003000h to 003FFFh
	SA4	0	0	0	0	0	0	0	1	0	0	8/4	008000h to 009FFFh	004000h to 004FFFh
	SA5	0	0	0	0	0	0	0	1	0	1	8/4	00A000h to 00BFFFh	005000h to 005FFFh
	SA6	0	0	0	0	0	0	0	1	1	0	8/4	00C000h to 00DFFFh	006000h to 006FFFh
	SA7	0	0	0	0	0	0	0	1	1	1	8/4	00E000h to 00FFFFh	007000h to 007FFFh
	SA8	0	0	0	0	0	0	1	X	X	X	64/32	010000h to 01FFFFh	008000h to 00FFFFh
	SA9	0	0	0	0	0	1	0	X	X	X	64/32	020000h to 02FFFFh	010000h to 017FFFh
	SA10	0	0	0	0	0	1	1	X	X	X	64/32	030000h to 03FFFFh	018000h to 01FFFFh
	SA11	0	0	0	0	1	0	0	X	X	X	64/32	040000h to 04FFFFh	020000h to 027FFFh
	SA12	0	0	0	0	1	0	1	X	X	X	64/32	050000h to 05FFFFh	028000h to 02FFFFh
	SA13	0	0	0	0	1	1	0	X	X	X	64/32	060000h to 06FFFFh	030000h to 037FFFh
	SA14	0	0	0	0	1	1	1	X	X	X	64/32	070000h to 07FFFFh	038000h to 03FFFFh
	SA15	0	0	0	1	0	0	0	X	X	X	64/32	080000h to 08FFFFh	040000h to 047FFFh
	SA16	0	0	0	1	0	0	1	X	X	X	64/32	090000h to 09FFFFh	048000h to 04FFFFh
	SA17	0	0	0	1	0	1	0	X	X	X	64/32	0A0000h to 0AFFFFh	050000h to 057FFFh
	SA18	0	0	0	1	0	1	1	X	X	X	64/32	0B0000h to 0BFFFFh	058000h to 06FFFFh
	SA19	0	0	0	1	1	0	0	X	X	X	64/32	0C0000h to 0CFFFFh	060000h to 067FFFh
	SA20	0	0	0	1	1	0	1	X	X	X	64/32	0D0000h to 0DFFFFh	068000h to 06FFFFh
	SA21	0	0	0	1	1	1	0	X	X	X	64/32	0E0000h to 0EFFFFh	070000h to 077FFFh
	SA22	0	0	0	1	1	1	1	X	X	X	64/32	0F0000h to 0FFFFFh	078000h to 07FFFFh

Note : The address range is A_{21} : A_{-1} if in byte mode ($\overline{BYTE} = V_{IL}$) .
The address range is A_{21} : A_0 if in word mode ($\overline{BYTE} = V_{IH}$) .

Table 6.2 Sector Address Tables (Bank B)

Bank	Sector	Sector Address										Sector Size (Kbytes/ Kwords)	(× 8) Address Range	(× 16) Address Range
		Bank Address												
		A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}			
Bank B	SA23	0	0	1	0	0	0	0	X	X	X	64/32	100000h to 10FFFFh	080000h to 087FFFh
	SA24	0	0	1	0	0	0	1	X	X	X	64/32	110000h to 11FFFFh	088000h to 08FFFFh
	SA25	0	0	1	0	0	1	0	X	X	X	64/32	120000h to 12FFFFh	090000h to 097FFFh
	SA26	0	0	1	0	0	1	1	X	X	X	64/32	130000h to 13FFFFh	098000h to 09FFFFh
	SA27	0	0	1	0	1	0	0	X	X	X	64/32	140000h to 14FFFFh	0A0000h to 0A7FFFh
	SA28	0	0	1	0	1	0	1	X	X	X	64/32	150000h to 15FFFFh	0A8000h to 0AFFFFh
	SA29	0	0	1	0	1	1	0	X	X	X	64/32	160000h to 16FFFFh	0B0000h to 0B7FFFh
	SA30	0	0	1	0	1	1	1	X	X	X	64/32	170000h to 17FFFFh	0B8000h to 0BFFFFh
	SA31	0	0	1	1	0	0	0	X	X	X	64/32	180000h to 18FFFFh	0C0000h to 0C7FFFh
	SA32	0	0	1	1	0	0	1	X	X	X	64/32	190000h to 19FFFFh	0C8000h to 0CFFFFh
	SA33	0	0	1	1	0	1	0	X	X	X	64/32	1A0000h to 1AFFFFh	0D0000h to 0D7FFFh
	SA34	0	0	1	1	0	1	1	X	X	X	64/32	1B0000h to 1BFFFFh	0D8000h to 0DFFFFh
	SA35	0	0	1	1	1	0	0	X	X	X	64/32	1C0000h to 1CFFFFh	0E0000h to 0E7FFFh
	SA36	0	0	1	1	1	0	1	X	X	X	64/32	1D0000h to 1DFFFFh	0E8000h to 0EFFFFh
	SA37	0	0	1	1	1	1	0	X	X	X	64/32	1E0000h to 1EFFFFh	0F0000h to 0F7FFFh
	SA38	0	0	1	1	1	1	1	X	X	X	64/32	1F0000h to 1FFFFFh	0F8000h to 0FFFFFh
	SA39	0	1	0	0	0	0	0	X	X	X	64/32	200000h to 20FFFFh	100000h to 107FFFh
	SA40	0	1	0	0	0	0	1	X	X	X	64/32	210000h to 21FFFFh	108000h to 10FFFFh
	SA41	0	1	0	0	0	1	0	X	X	X	64/32	220000h to 22FFFFh	110000h to 117FFFh
	SA42	0	1	0	0	0	1	1	X	X	X	64/32	230000h to 23FFFFh	118000h to 11FFFFh
	SA43	0	1	0	0	1	0	0	X	X	X	64/32	240000h to 24FFFFh	120000h to 127FFFh
	SA44	0	1	0	0	1	0	1	X	X	X	64/32	250000h to 25FFFFh	128000h to 12FFFFh
	SA45	0	1	0	0	1	1	0	X	X	X	64/32	260000h to 26FFFFh	130000h to 137FFFh
	SA46	0	1	0	0	1	1	1	X	X	X	64/32	270000h to 27FFFFh	138000h to 13FFFFh
	SA47	0	1	0	1	0	0	0	X	X	X	64/32	280000h to 28FFFFh	140000h to 147FFFh
	SA48	0	1	0	1	0	0	1	X	X	X	64/32	290000h to 29FFFFh	148000h to 14FFFFh
	SA49	0	1	0	1	0	1	0	X	X	X	64/32	2A0000h to 2AFFFFh	150000h to 157FFFh
	SA50	0	1	0	1	0	1	1	X	X	X	64/32	2B0000h to 2BFFFFh	158000h to 15FFFFh
	SA51	0	1	0	1	1	0	0	X	X	X	64/32	2C0000h to 2CFFFFh	160000h to 167FFFh
	SA52	0	1	0	1	1	0	1	X	X	X	64/32	2D0000h to 2DFFFFh	168000h to 16FFFFh
	SA53	0	1	0	1	1	1	0	X	X	X	64/32	2E0000h to 2EFFFFh	170000h to 177FFFh

(Continued)

(Continued)

Bank	Sector	Sector Address										Sector Size (Kbytes/ Kwords)	(× 8) Address Range	(× 16) Address Range
		Bank Address												
		A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}			
Bank B	SA54	0	1	0	1	1	1	1	X	X	X	64/32	2F0000h to 2FFFFFh	178000h to 17FFFFh
	SA55	0	1	1	0	0	0	0	X	X	X	64/32	300000h to 30FFFFh	180000h to 187FFFh
	SA56	0	1	1	0	0	0	1	X	X	X	64/32	310000h to 31FFFFh	188000h to 18FFFFh
	SA57	0	1	1	0	0	1	0	X	X	X	64/32	320000h to 32FFFFh	190000h to 197FFFh
	SA58	0	1	1	0	0	1	1	X	X	X	64/32	330000h to 33FFFFh	198000h to 19FFFFh
	SA59	0	1	1	0	1	0	0	X	X	X	64/32	340000h to 34FFFFh	1A0000h to 1A7FFFh
	SA60	0	1	1	0	1	0	1	X	X	X	64/32	350000h to 35FFFFh	1A8000h to 1AFFFFh
	SA61	0	1	1	0	1	1	0	X	X	X	64/32	360000h to 36FFFFh	1B0000h to 1B7FFFh
	SA62	0	1	1	0	1	1	1	X	X	X	64/32	370000h to 37FFFFh	1B8000h to 1BFFFFh
	SA63	0	1	1	1	0	0	0	X	X	X	64/32	380000h to 38FFFFh	1C0000h to 1C7FFFh
	SA64	0	1	1	1	0	0	1	X	X	X	64/32	390000h to 39FFFFh	1C8000h to 1CFFFFh
	SA65	0	1	1	1	0	1	0	X	X	X	64/32	3A0000h to 3AFFFFh	1D0000h to 1D7FFFh
	SA66	0	1	1	1	0	1	1	X	X	X	64/32	3B0000h to 3BFFFFh	1D8000h to 1DFFFFh
	SA67	0	1	1	1	1	0	0	X	X	X	64/32	3C0000h to 3CFFFFh	1E0000h to 1E7FFFh
	SA68	0	1	1	1	1	0	1	X	X	X	64/32	3D0000h to 3DFFFFh	1E8000h to 1EFFFFh
	SA69	0	1	1	1	1	1	0	X	X	X	64/32	3E0000h to 3EFFFFh	1F0000h to 1F7FFFh
	SA70	0	1	1	1	1	1	1	X	X	X	64/32	3F0000h to 3FFFFFh	1F8000h to 1FFFFFh

Note : The address range is A_{21} : A_{-1} if in byte mode ($\overline{BYTE} = V_{IL}$) .
The address range is A_{21} : A_0 if in word mode ($\overline{BYTE} = V_{IH}$) .

Table 6.3 Sector Address Tables (Bank C)

Bank	Sector	Sector Address: Bank Address A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	Sector Size (Kbytes/ Kwords)	(× 8) Address Range	(× 16) Address Range
Bank C	SA71	1	0	0	0	0	0	0	X	X	X	64/32	400000h to 40FFFFh	200000h to 207FFFh
	SA72	1	0	0	0	0	0	1	X	X	X	64/32	410000h to 41FFFFh	208000h to 20FFFFh
	SA73	1	0	0	0	0	1	0	X	X	X	64/32	420000h to 42FFFFh	210000h to 217FFFh
	SA74	1	0	0	0	0	1	1	X	X	X	64/32	430000h to 43FFFFh	218000h to 21FFFFh
	SA75	1	0	0	0	1	0	0	X	X	X	64/32	440000h to 44FFFFh	220000h to 227FFFh
	SA76	1	0	0	0	1	0	1	X	X	X	64/32	450000h to 45FFFFh	228000h to 22FFFFh
	SA77	1	0	0	0	1	1	0	X	X	X	64/32	460000h to 46FFFFh	230000h to 237FFFh
	SA78	1	0	0	0	1	1	1	X	X	X	64/32	470000h to 47FFFFh	238000h to 23FFFFh
	SA79	1	0	0	1	0	0	0	X	X	X	64/32	480000h to 48FFFFh	240000h to 247FFFh
	SA80	1	0	0	1	0	0	1	X	X	X	64/32	490000h to 49FFFFh	248000h to 24FFFFh
	SA81	1	0	0	1	0	1	0	X	X	X	64/32	4A0000h to 4AFFFFh	250000h to 257FFFh
	SA82	1	0	0	1	0	1	1	X	X	X	64/32	4B0000h to 4BFFFFh	258000h to 25FFFFh
	SA83	1	0	0	1	1	0	0	X	X	X	64/32	4C0000h to 4CFFFFh	260000h to 267FFFh
	SA84	1	0	0	1	1	0	1	X	X	X	64/32	4D0000h to 4DFFFFh	268000h to 26FFFFh
	SA85	1	0	0	1	1	1	0	X	X	X	64/32	4E0000h to 4EFFFFh	270000h to 277FFFh
	SA86	1	0	0	1	1	1	1	X	X	X	64/32	4F0000h to 4FFFFFh	278000h to 27FFFFh
	SA87	1	0	1	0	0	0	0	X	X	X	64/32	500000h to 50FFFFh	280000h to 287FFFh
	SA88	1	0	1	0	0	0	1	X	X	X	64/32	510000h to 51FFFFh	288000h to 28FFFFh
	SA89	1	0	1	0	0	1	0	X	X	X	64/32	520000h to 52FFFFh	290000h to 297FFFh
	SA90	1	0	1	0	0	1	1	X	X	X	64/32	530000h to 53FFFFh	298000h to 29FFFFh
	SA91	1	0	1	0	1	0	0	X	X	X	64/32	540000h to 54FFFFh	2A0000h to 2A7FFFh
	SA92	1	0	1	0	1	0	1	X	X	X	64/32	550000h to 55FFFFh	2A8000h to 2AFFFFh
	SA93	1	0	1	0	1	1	0	X	X	X	64/32	560000h to 56FFFFh	2B0000h to 2B7FFFh
	SA94	1	0	1	0	1	1	1	X	X	X	64/32	570000h to 57FFFFh	2B8000h to 2BFFFFh
	SA95	1	0	1	1	0	0	0	X	X	X	64/32	580000h to 58FFFFh	2C0000h to 2C7FFFh
	SA96	1	0	1	1	0	0	1	X	X	X	64/32	590000h to 59FFFFh	2C8000h to 2CFFFFh
	SA97	1	0	1	1	0	1	0	X	X	X	64/32	5A0000h to 5AFFFFh	2D0000h to 2D7FFFh
	SA98	1	0	1	1	0	1	1	X	X	X	64/32	5B0000h to 5BFFFFh	2D8000h to 2DFFFFh
	SA99	1	0	1	1	1	0	0	X	X	X	64/32	5C0000h to 5CFFFFh	2E0000h to 2EE7FFh
	SA100	1	0	1	1	1	0	1	X	X	X	64/32	5D0000h to 5DFFFFh	2E8000h to 2EFFFFh
	SA101	1	0	1	1	1	1	0	X	X	X	64/32	5E0000h to 5EFFFFh	2F0000h to 2F7FFFh
	SA102	1	0	1	1	1	1	1	X	X	X	64/32	5F0000h to 5FFFFFh	2F8000h to 2FFFFFh

(Continued)

(Continued)

Bank	Sector	Sector Address										Sector Size (Kbytes/ Kwords)	(×8) Address Range	(×16) Address Range
		Bank Address												
		A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}			
Bank C	SA103	1	1	0	0	0	0	0	X	X	X	64/32	600000h to 60FFFFh	300000h to 307FFFh
	SA104	1	1	0	0	0	0	1	X	X	X	64/32	610000h to 61FFFFh	308000h to 30FFFFh
	SA105	1	1	0	0	0	1	0	X	X	X	64/32	620000h to 62FFFFh	310000h to 317FFFh
	SA106	1	1	0	0	0	1	1	X	X	X	64/32	630000h to 63FFFFh	318000h to 31FFFFh
	SA107	1	1	0	0	1	0	0	X	X	X	64/32	640000h to 64FFFFh	320000h to 327FFFh
	SA108	1	1	0	0	1	0	1	X	X	X	64/32	650000h to 65FFFFh	328000h to 32FFFFh
	SA109	1	1	0	0	1	1	0	X	X	X	64/32	660000h to 66FFFFh	330000h to 337FFFh
	SA110	1	1	0	0	1	1	1	X	X	X	64/32	670000h to 67FFFFh	338000h to 33FFFFh
	SA111	1	1	0	1	0	0	0	X	X	X	64/32	680000h to 68FFFFh	340000h to 347FFFh
	SA112	1	1	0	1	0	0	1	X	X	X	64/32	690000h to 69FFFFh	348000h to 34FFFFh
	SA113	1	1	0	1	0	1	0	X	X	X	64/32	6A0000h to 6AFFFFh	350000h to 357FFFh
	SA114	1	1	0	1	0	1	1	X	X	X	64/32	6B0000h to 6BFFFFh	358000h to 35FFFFh
	SA115	1	1	0	1	1	0	0	X	X	X	64/32	6C0000h to 6CFFFFh	360000h to 367FFFh
	SA116	1	1	0	1	1	0	1	X	X	X	64/32	6D0000h to 6DFFFFh	368000h to 36FFFFh
	SA117	1	1	0	1	1	1	0	X	X	X	64/32	6E0000h to 6EFFFFh	370000h to 377FFFh
	SA118	1	1	0	1	1	1	1	X	X	X	64/32	6F0000h to 6FFFFFh	378000h to 37FFFFh

Note : The address range is A_{21} : A_{-1} if in byte mode ($\overline{BYTE} = V_{IL}$) .
The address range is A_{21} : A_0 if in word mode ($\overline{BYTE} = V_{IH}$) .

Table 6.4 Sector Address Tables (Bank D)

Bank	Sector	Sector Address										Sector Size (Kbytes/Kwords)	(×8) Address Range	(×16) Address Range
		Bank Address												
		A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}			
Bank D	SA119	1	1	1	0	0	0	0	X	X	X	64/32	700000h to 70FFFFh	380000h to 387FFFh
	SA120	1	1	1	0	0	0	1	X	X	X	64/32	710000h to 71FFFFh	388000h to 38FFFFh
	SA121	1	1	1	0	0	1	0	X	X	X	64/32	720000h to 72FFFFh	390000h to 397FFFh
	SA122	1	1	1	0	0	1	1	X	X	X	64/32	730000h to 73FFFFh	398000h to 39FFFFh
	SA123	1	1	1	0	1	0	0	X	X	X	64/32	740000h to 74FFFFh	3A0000h to 3A7FFFh
	SA124	1	1	1	0	1	0	1	X	X	X	64/32	750000h to 75FFFFh	3A8000h to 3AFFFFh
	SA125	1	1	1	0	1	1	0	X	X	X	64/32	760000h to 76FFFFh	3B0000h to 3B7FFFh
	SA126	1	1	1	0	1	1	1	X	X	X	64/32	770000h to 77FFFFh	3B8000h to 3BFFFFh
	SA127	1	1	1	1	0	0	0	X	X	X	64/32	780000h to 78FFFFh	3C0000h to 3C7FFFh
	SA128	1	1	1	1	0	0	1	X	X	X	64/32	790000h to 79FFFFh	3C8000h to 3CFFFFh
	SA129	1	1	1	1	0	1	0	X	X	X	64/32	7A0000h to 7AFFFFh	3D0000h to 3D7FFFh
	SA130	1	1	1	1	0	1	1	X	X	X	64/32	7B0000h to 7BFFFFh	3D8000h to 3DFFFFh
	SA131	1	1	1	1	1	0	0	X	X	X	64/32	7C0000h to 7CFFFFh	3E0000h to 3E7FFFh
	SA132	1	1	1	1	1	0	1	X	X	X	64/32	7D0000h to 7DFFFFh	3E8000h to 3EFFFFh
	SA133	1	1	1	1	1	1	0	X	X	X	64/32	7E0000h to 7EFFFFh	3F0000h to 3F7FFFh
	SA134	1	1	1	1	1	1	1	0	0	0	8/4	7F0000h to 7F1FFFh	3F8000h to 3F8FFFh
	SA135	1	1	1	1	1	1	1	0	0	1	8/4	7F2000h to 7F3FFFh	3F9000h to 3F9FFFh
	SA136	1	1	1	1	1	1	1	0	1	0	8/4	7F4000h to 7F5FFFh	3FA000h to 3FAFFFh
	SA137	1	1	1	1	1	1	1	0	1	1	8/4	7F6000h to 7F7FFFh	3FB000h to 3FBFFFh
	SA138	1	1	1	1	1	1	1	1	0	0	8/4	7F8000h to 7F9FFFh	3FC000h to 3FCFFFh
	SA139	1	1	1	1	1	1	1	1	0	1	8/4	7FA000h to 7FBFFFh	3FD000h to 3FDFFFh
	SA140	1	1	1	1	1	1	1	1	1	0	8/4	7FC000h to 7FDFFFh	3FE000h to 3FEFFFh
	SA141	1	1	1	1	1	1	1	1	1	1	8/4	7FE000h to 7FFFFFh	3FF000h to 3FFFFFh

Note : The address range is A_{21} : A_{-1} if in byte mode ($\overline{BYTE} = V_{IL}$) .
The address range is A_{21} : A_0 if in word mode ($\overline{BYTE} = V_{IH}$) .

Table 7　Sector Group Address Table

Sector Group	A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	Sectors
SGA0	0	0	0	0	0	0	0	0	0	0	SA0
SGA1	0	0	0	0	0	0	0	0	0	1	SA1
SGA2	0	0	0	0	0	0	0	0	1	0	SA2
SGA3	0	0	0	0	0	0	0	0	1	1	SA3
SGA4	0	0	0	0	0	0	0	1	0	0	SA4
SGA5	0	0	0	0	0	0	0	1	0	1	SA5
SGA6	0	0	0	0	0	0	0	1	1	0	SA6
SGA7	0	0	0	0	0	0	0	1	1	1	SA7
SGA8	0	0	0	0	0	0	0	X	X	X	SA8 to SA10
						0	1				
						1	0				
SGA9	0	0	0	0	1	X	X	X	X	X	SA11 to SA14
SGA10	0	0	0	1	0	X	X	X	X	X	SA15 to SA18
SGA11	0	0	0	1	1	X	X	X	X	X	SA19 to SA22
SGA12	0	0	1	0	0	X	X	X	X	X	SA23 to SA26
SGA13	0	0	1	0	1	X	X	X	X	X	SA27 to SA30
SGA14	0	0	1	1	0	X	X	X	X	X	SA31 to SA34
SGA15	0	0	1	1	1	X	X	X	X	X	SA35 to SA38
SGA16	0	1	0	0	0	X	X	X	X	X	SA39 to SA42
SGA17	0	1	0	0	1	X	X	X	X	X	SA43 to SA46
SGA18	0	1	0	1	0	X	X	X	X	X	SA47 to SA50
SGA19	0	1	0	1	1	X	X	X	X	X	SA51 to SA54
SGA20	0	1	1	0	0	X	X	X	X	X	SA55 to SA58
SGA21	0	1	1	0	1	X	X	X	X	X	SA59 to SA62
SGA22	0	1	1	1	0	X	X	X	X	X	SA63 to SA66
SGA23	0	1	1	1	1	X	X	X	X	X	SA67 to SA70

(Continued)

(Continued)

Sector Group	A_{21}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	Sectors
SGA24	1	0	0	0	0	X	X	X	X	X	SA71 to SA74
SGA25	1	0	0	0	1	X	X	X	X	X	SA75 to SA78
SGA26	1	0	0	1	0	X	X	X	X	X	SA79 to SA82
SGA27	1	0	0	1	1	X	X	X	X	X	SA83 to SA86
SGA28	1	0	0	0	0	X	X	X	X	X	SA87 to SA90
SGA29	1	0	1	0	1	X	X	X	X	X	SA91 to SA94
SGA30	1	0	1	1	0	X	X	X	X	X	SA95 to SA98
SGA31	1	0	1	1	1	X	X	X	X	X	SA99 to SA102
SGA32	1	1	0	0	0	X	X	X	X	X	SA103 to SA106
SGA33	1	1	0	0	1	X	X	X	X	X	SA107 to SA110
SGA34	1	1	0	1	0	X	X	X	X	X	SA111 to SA114
SGA35	1	1	0	1	1	X	X	X	X	X	SA115 to SA118
SGA36	1	1	1	0	0	X	X	X	X	X	SA119 to SA122
SGA37	1	1	1	0	1	X	X	X	X	X	SA123 to SA126
SGA38	1	1	1	1	0	X	X	X	X	X	SA127 to SA130
SGA39	1	1	1	1	1	0 0 1	0 1 0	X	X	X	SA131 to SA133
SGA40	1	1	1	1	1	1	1	0	0	0	SA134
SGA41	1	1	1	1	1	1	1	0	0	1	SA135
SGA42	1	1	1	1	1	1	1	0	1	0	SA136
SGA43	1	1	1	1	1	1	1	0	1	1	SA137
SGA44	1	1	1	1	1	1	1	1	0	0	SA138
SGA45	1	1	1	1	1	1	1	1	0	1	SA139
SGA46	1	1	1	1	1	1	1	1	1	0	SA140
SGA47	1	1	1	1	1	1	1	1	1	1	SA141

Table 8 Common Flash Memory Interface Code

Description	A_6 to A_0	DQ_{15} to DQ_0
Query-unique ASCII string "QRY"	10h 11h 12h	0051h 0052h 0059h
Primary OEM Command Set 2h : AMD/FJ standard type	13h 14h	0002h 0000h
Address for Primary Extended Table	15h 16h	0040h 0000h
Alternate OEM Command Set (00h = not applicable)	17h 18h	0000h 0000h
Address for Alternate OEM Extended Table	19h 1Ah	0000h 0000h
V_{CC} Min. (write/erase) D7-4 : 1 V, D3-0 : 100 mV	1Bh	0027h
V_{CC} Max. (write/erase) D7-4 : 1 V, D3-0 : 100 mV	1Ch	0036h
V_{PP} Min. voltage	1Dh	0000h
V_{PP} Max. voltage	1Eh	0000h
Typical timeout per single byte/word write 2^N µs	1Fh	0004h
Typical timeout for Min. size buffer write 2^N µs	20h	0000h
Typical timeout per individual block erase 2^N ms	21h	000Ah
Typical timeout for full chip erase 2^N ms	22h	0000h
Max. timeout for byte/word write 2^N times typical	23h	0005h
Max. timeout for buffer write 2^N times typical	24h	0000h
Max. timeout per individual block erase 2^N times typical	25h	0004h
Max. timeout for full chip erase 2^N times typical	26h	0000h
Device Size = 2^N byte	27h	0017h
Flash Device Interface description ×: ×8 / ×16	28h 29h	0002h 0000h
Max. number of bytes in multi-byte write = 2^N	2Ah 2Bh	0000h 0000h
Number of Erase Block Regions within device	2Ch	0003h
Erase Block Region 1 Information bit 0 to 15: y = number of sectors bit 16 to 31: z = size (z × 256 bytes)	2Dh 2Eh 2Fh 30h	0007h 0000h 0020h 0000h
Erase Block Region 2 Information bit 0 to 15: y = number of sectors bit 16 to 31: z = size (z × 256 bytes)	31h 32h 33h 34h	007Dh 0000h 0000h 0001h
Erase Block Region 3 Information bit 0 to 15: y = number of sectors bit 16 to 31: z = size (z × 256 bytes)	35h 36h 37h 38h	0007h 0000h 0020h 0000h

Description	A_6 to A_0	DQ_{15} to DQ_0
Query-unique ASCII string "PRI"	40h 41h 42h	0050h 0052h 0049h
Major version number, ASCII	43h	0031h
Minor version number, ASCII	44h	0033h
Address Sensitive Unlock 0h = Required 1h = Not Required	45h	0000h
Erase Suspend 0h = Not Supported 1h = To Read Only 2h = To Read & Write	46h	0002h
Sector Protection 0h = Not Supported X = Number of sectors per group	47h	0001h
Sector Temporary Unprotection 00h = Not Supported 01h = Supported	48h	0001h
Sector Protection Algorithm	49h	0004h
Simultaneous Operation 00h = Not Supported X = Total number of sectors in all banks except Bank 1	4Ah	0077h
Burst Mode Type 00h = Not Supported	4Bh	0000h
Page Mode Type 00h = Not Supported	4Ch	0000h
ACC (Acceleration) Supply Minimum 00h = Not Supported, D7-4 : 1 V, D3-0 : 100 mV	4Dh	0085h
ACC (Acceleration) Supply Maximum 00h = Not Supported, D7-4 : 1 V, D3-0 : 100 mV	4Eh	0095h
Boot Type	4Fh	0001h
Program Suspend 00h = Not Supported 01h = Supported	50h	0001h
Bank Organization 00h = If data at 4Ah is zero. X = Number of Banks	57h	0004h
Bank A Region Information X = Number of sectors in Bank A	58h	0017h
Bank B Region Information X = Number of sectors in Bank B	59h	0030h
Bank C Region Information X = Number of sectors in Bank C	5Ah	0030h
Bank D Region Information X = Number of sectors in Bank D	5Bh	0017h

■ FUNCTIONAL DESCRIPTION

Simultaneous Operation

The device features functions that enable reading of data from one memory bank while a program or erase operation is in progress in the other memory bank (simultaneous operation) , in addition to conventional features (read, program, erase, erase-suspend read, and erase-suspend program) . The bank can be selected by bank address (A_{21}, A_{20}, A_{19}) with zero latency. The device consists of the following four banks :
Bank A : 8 × 8 KB and 15 × 64 KB; Bank B : 48 × 64 KB; Bank C : 48 × 64 KB; Bank D : 8 × 8 KB and 15 × 64 KB.
The device can execute simultaneous operations between Bank 1, a bank chosen from among the four banks, and Bank 2, a bank consisting of the three remaining banks. (See Table 9.) This is what we call a “FlexBank”, for example, the rest of banks B, C and D to let the system read while Bank A is in the process of program (or erase) operation. However, the different types of operations for the three banks are impossible, e.g. Bank A writing, Bank B erasing, and Bank C reading out. With this “FlexBank”, as described in Table 10, the system gets to select from four combinations of data volume for Bank 1 and Bank 2, which works well to meet the system requirement. The simultaneous operation cannot execute multi-function mode in the same bank. Table 11 shows the possible combinations for simultaneous operation. (Refer to Figure 11 Bank-to-Bank Read/Write Timing Diagram.)

Table 9 FlexBank™ Architecture

Bank Splits	Bank 1		Bank 2	
	Volume	Combination	Volume	Combination
1	8 Mbit	Bank A	56 Mbit	Remainder (Bank B, C, D)
2	24 Mbit	Bank B	40 Mbit	Remainder (Bank A, C, D)
3	24 Mbit	Bank C	40 Mbit	Remainder (Bank A, B, D)
4	8 Mbit	Bank D	56 Mbit	Remainder (Bank A, B, C)

Table 10 Example of Virtual Banks Combination

Bank Splits	Bank 1			Bank 2		
	Volume	Combination	Sector Size	Volume	Combination	Sector Size
1	8 Mbit	Bank A	8 × 8 Kbyte/4 Kword + 15 × 64 Kbyte/32 Kword	56 Mbit	Bank B + Bank C + Bank D	8 × 8 Kbyte/4 Kword + 111 × 64 Kbyte/32 Kword
2	16 Mbit	Bank A + Bank D	16 × 8 Kbyte/4 Kword + 30 × 64 Kbyte/32 Kword	48 Mbit	Bank B + Bank C	96 × 64 Kbyte/32 Kword
3	24 Mbit	Bank B	48 × 64 Kbyte/32 Kword	40 Mbit	Bank A + Bank C + Bank D	16 × 8 Kbyte/4 Kword + 78 × 64 Kbyte/32 Kword
4	32 Mbit	Bank A + Bank B	8 × 8 Kbyte/4 Kword + 63 × 64 Kbyte/32 Kword	32 Mbit	Bank C + Bank D	8 × 8 Kbyte/4 Kword + 63 × 64 Kbyte/32 Kword

Note : When multiple sector erase over several banks is operated, the system cannot read out of the bank to which a sector being erased belongs. For example, suppose that erasing is taking place at both Bank A and Bank B, neither Bank A nor Bank B is read out (they would output the sequence flag once they were selected.) Meanwhile the system would get to read from either Bank C or Bank D.

Table 11 Simultaneous Operation

Case	Bank 1 Status	Bank 2 Status
1	Read mode	Read mode
2	Read mode	Autoselect mode
3	Read mode	Program mode
4	Read mode	Erase mode *
5	Autoselect mode	Read mode
6	Program mode	Read mode
7	Erase mode *	Read mode

* : By writing erase suspend command on the bank address of sector being erased, the erase operation gets suspended so that it enables reading from or programming the remaining sectors.

Note: Bank 1 and Bank 2 are divided for the sake of convenience at Simultaneous Operation. Actually, the Bank consists of 4 banks, Bank A, Bank B, BankC and Bank D. Bank Address (BA) meant to specify each of the Banks.

Read Mode

The device has two control functions which are required in order to obtain data at the outputs. $\overline{CE}$ is the power control and should be used for a device selection. $\overline{OE}$ is the output control and should be used to gate data to the output pins.

Address access time (t_{ACC}) is equal to delay from stable addresses to valid output data. The chip enable access time (t_{CE}) is the delay from stable addresses and stable $\overline{CE}$ to valid data at the output pins. The output enable access time is the delay from the falling edge of $\overline{OE}$ to valid data at the output pins (assuming the addresses have been stable for at least t_{ACC}-t_{OE} time) . When reading out data without changing addresses after power-up, it is necessary to input hardware reset or to change $\overline{CE}$ pin from “H” or “L”

Standby Mode

There are two ways to implement the standby mode on the device, one using both the $\overline{CE}$ and $\overline{RESET}$ pins, and the other via the $\overline{RESET}$ pin only.

When using both pins, a CMOS standby mode is achieved with $\overline{CE}$ and $\overline{RESET}$ input held at $V_{CC} \pm 0.3$ V. Under this condition the current consumed is less than 5 µA Max. During Embedded Algorithm operation, V_{CC} active current (I_{CC2}) is required even if $\overline{CE}$ = “H”. The device can be read with standard access time (t_{CE}) from either of these standby modes.

When using the $\overline{RESET}$ pin only, a CMOS standby mode is achieved with $\overline{RESET}$ input held at $V_{SS} \pm 0.3$ V ($\overline{CE}$ = “H” or “L”) . Under this condition the current consumed is less than 5 µA Max. Once the $\overline{RESET}$ pin is set high, the device requires t_{RH} as a wake-up time for output to be valid for read access.

During standby mode, the output is in the high impedance state, regardless of $\overline{OE}$ input.

Automatic Sleep Mode

Automatic sleep mode works to restrain power consumption during read-out of device data. It can be useful in applications such as handy terminal, which requires low power consumption.

To activate this mode, the device automatically switches itself to low power mode when the device addresses remain stable during access time of 150 ns. It is not necessary to control $\overline{CE}$, $\overline{WE}$ and $\overline{OE}$ in this mode. In this mode the current consumed is typically 1 µA (CMOS Level) .

During simultaneous operation, V_{CC} active current (I_{CC2}) is required.

Since the data are latched during this mode, the data are continuously read out. When the addresses are changed, the mode is automatically canceled and the device reads the data for changed addresses.

Output Disable

With the $\overline{OE}$ input at a logic high level (V_{IH}) , output from the device is disabled. This will cause the output pins to be in a high impedance state.

Autoselect

The Autoselect mode allows the reading out of a binary code and identifies its manufacturer and type.It is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional over the entire temperature range of the device.

To activate this mode, the programming equipment must force V_{ID} on address pin A_9. Three identifier bytes may then be sequenced from the device outputs by toggling addresses. All addresses are DON'T CARES except A_6, A_3, A_2, A_1 and A_0 (A_{-1}) . (See Tables 2 and 3.)

The manufacturer and device codes may also be read via the command register, for instances when the device is erased or programmed in a system without access to high voltage on the A_9 pin. The command sequence is illustrated in Table 4. (Refer to Autoselect Command section.)

In the command Autoselect mode, the bank addresses BA; (A_{21}, A_{20}, A_{19}) must point to a specific bank during the third write bus cycle of the Autoselect command. Then the Autoselect data will be read from that bank while array data can be read from the other bank.

In WORD mode, a read cycle from address 00h returns the manufacturer's code (Fujitsu = 04h) . A read cycle at address 01h outputs device code. When 227Eh is output, it indicates that two additional codes, called Extended Device Codes will be required. Therefore the system may continue reading out these Extended Device Codes at addresses of 0Eh and 0Fh. Notice that the above applies to WORD mode; the addresses and codes differ from those of BYTE mode. (Refer to Table 5.1 and 5.2.)

In the case of applying V_{ID} on A_9, since both Bank 1 and Bank 2 enter Autoselect mode, simultanous operation cannot be executed.

Write

Device erasure and programming are accomplished via the command register. The contents of the register serve as input to the internal state machine. The state machine output dictates the function of the device.

The command register itself does not occupy any addressable memory location. The register is a latch used to store the commands, along with the address and data information needed to execute the command. The command register is written by bringing $\overline{WE}$ to V_{IL}, while $\overline{CE}$ is at V_{IL} and $\overline{OE}$ is at V_{IH}. Addresses are latched on the falling edge of $\overline{WE}$ or $\overline{CE}$, whichever happens later, while data is latched on the rising edge of $\overline{WE}$ or $\overline{CE}$, whichever happens first. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

Sector Group Protection

The device features hardware sector group protection. This feature will disable both program and erase operations in any combination of forty eight sector groups of memory. (See Table 7) . The user's side can use the sector group protection using programming equipment. The device is shipped with all sector groups that are unprotected.

To activate this mode, the programming equipment must force V_{ID} on address pin A_9 and control pin $\overline{OE}$, $\overline{CE}$ = V_{IL} and $A_6 = A_3 = A_2 = A_0 = V_{IL}$, $A_1 = V_{IH}$. The sector group addresses (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13} and A_{12}) should be set to the sector to be protected. Tables 6.1 to 6.4 define the sector address for each of the one hundred forty-two (142) individual sectors, and Table 7 defines the sector group address for each of the forty eight (48) individual group sectors. Programming of the protection circuitry begins on the falling edge of the $\overline{WE}$ pulse and is terminated with the rising edge of the same. Sector group addresses must be held constant during the $\overline{WE}$ pulse. See Figures 18 and 26 for sector group protection waveforms and algorithms.

To verify programming of the protection circuitry, the programming equipment must force V_{ID} on address pin A_9 with $\overline{CE}$ and $\overline{OE}$ at V_{IL} and $\overline{WE}$ at V_{IH}. Scanning the sector group addresses (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13} and A_{12}) while (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) will produce a logical "1" code at device output DQ_0 for a protected sector. Otherwise the device will produce "0" for unprotected sectors. In this mode, the lower order addresses, except for A_0, A_1, A_2, A_3 and A_6 are DON'T CARES. Address locations with $A_1 = V_{IL}$ are reserved for Autoselect manufacturer and device codes. A_{-1} requires applying to V_{IL} on byte mode.

Whether the sector group is protected in the system can be determined by writing an Autoselect command. Performing a read operation at the address location (BA) XX02h, where the higher order addresses (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13}, and A_{12}) are the desired sector group address, will produce a logical "1" at DQ_0 for a protected sector group. Note that the bank addresses (A_{21}, A_{20}, A_{19}) must be pointing to a specific bank during the third write bus cycle of the Autoselect command. Then the Autoselect data can be read from that bank while array data can still be read from the other bank. To read Autoselect data from the other bank, it must be reset to read mode and then write the Autoselect command to the other bank. See Tables 5.1 and 5.2 for Autoselect codes.

Temporary Sector Group Unprotection

This feature allows temporary unprotection of previously protected sector groups of the device in order to change data. The Sector Group Unprotection mode is activated by setting the $\overline{RESET}$ pin to high voltage (V_{ID}) . During this mode, formerly protected sector groups can be programmed or erased by selecting the sector group addresses. Once the V_{ID} is taken away from the $\overline{RESET}$ pin, all the previously protected sector groups will be protected again. Refer to Figures 19 and 27.

Extended Sector Group Protection

In addition to normal sector group protection, the device has Extended Sector Group Protection as extended function. This function enables protection of the sector group by forcing V_{ID} on $\overline{RESET}$ pin and writes a command sequence. Unlike conventional procedures, it is not necessary to force V_{ID} and control timing for control pins. The only $\overline{RESET}$ pin requires V_{ID} for sector group protection in this mode. The extended sector group protection requires V_{ID} on $\overline{RESET}$ pin. With this condition the operation is initiated by writing the set-up command (60h) in the command register. Then the sector group addresses pins (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13} and A_{12}) and (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) should be set to the sector group to be protected (setting V_{IL} for the other addresses pins is recommended) , and an extended sector group protection command (60h) should be written. A sector group is typically protected in 250 μs. To verify programming of the protection circuitry, the sector group addresses pins (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13} and A_{12}) and (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) should be set a command (40h) should be written. Following the command write, a logical "1" at device output DQ_0 will produce a protected sector in the read operation. If the output is logical "0", write the extended sector group protection command (60h) again. To terminate the operation, it is necessary to set $\overline{RESET}$ pin to V_{IH}. (Refer to Figures 20 and 28.)

$\overline{\text{RESET}}$

Hardware Reset

The device may be reset by driving the $\overline{\text{RESET}}$ pin to V_{IL}. The $\overline{\text{RESET}}$ pin has a pulse requirement and has to be kept low (V_{IL}) for at least "t_{RP}" in order to properly reset the internal state machine. Any operation in the process of being executed will be terminated and the internal state machine will be reset to the read mode "t_{READY}" after the $\overline{\text{RESET}}$ pin is driven low. Furthermore, once the $\overline{\text{RESET}}$ pin goes high the device requires an additional "t_{RH}" before it will allow read access. When the $\overline{\text{RESET}}$ pin is low, the device will be in the standby mode for the duration of the pulse and all the data output pins will be tri-stated. If a hardware reset occurs during a program or erase operation, the data at that particular location will be corrupted. Please note that the RY/$\overline{\text{BY}}$ output signal should be ignored during the $\overline{\text{RESET}}$ pulse. See Figure 14 for the timing diagram. Refer to Temporary Sector Group Unprotection for additional functionality.

Boot Block Sector Protection

The Write Protection function provides a hardware method of protecting certain boot sectors without using V_{ID}. This function is one of two provided by the $\overline{\text{WP}}$/ACC pin.

If the system asserts V_{IL} on the $\overline{\text{WP}}$/ACC pin, the device disables program and erase functions in the two outermost 8 Kbytes on both ends of boot sectors independently of whether those sectors are protected or unprotected using the method described in "Sector Protection/Unprotection."
(MBM29DL640E : SA0, SA1, SA140, and SA141)

If the system asserts V_{IH} on the $\overline{\text{WP}}$/ACC pin, the device reverts to whether the two outermost 8 Kbyte on both ends of boot sectors were last set to be protected or unprotected. Sector protection or unprotection for these four sectors depends on whether they were last protected or unprotected using the method described in "Sector protection/unprotection."

Accelerated Program Operation

The device offers accelerated program operation which enables programming in high speed. If the system asserts V_{ACC} to the $\overline{\text{WP}}$/ACC pin, the device automatically enters the acceleration mode and the time required for program operation will reduce to about 60%. This function is primarily intended to allow high speed programming, so caution is needed as the sector group will temporarily be unprotected.

The system would use a fast program command sequence when programming during acceleration mode.
Set command to fast mode and reset command from fast mode are not necessary. When the device enters the acceleration mode, the device is automatically set to fast mode. Therefore, the present sequence could be used for programming and detection of completion during acceleration mode.

Removing V_{ACC} from the $\overline{\text{WP}}$/ACC pin returns the device to normal operation. Do not remove V_{ACC} from $\overline{\text{WP}}$/ACC pin while programming. See Figure 21.

■ COMMAND DEFINITIONS

Device operations are selected by writing specific address and data sequences into the command register. Some commands require Bank Address (BA) input. When command sequences are input into a bank reading, the commands have priority over the reading. Table 4 shows the valid register command sequences. Note that the Erase Suspend (B0h) and Erase Resume (30h) commands are valid only while the Sector Erase operation is in progress. Also the Program Suspend (B0h) and Program Resume (30h) commands are valid only while the Program operation is in progress. Moreover, Read/Reset commands are functionally equivalent, resetting the device to the read mode. Please note that commands are always written at DQ_7 to DQ_0 and DQ_{15} to DQ_8 bits are ignored.

Read/Reset Command

In order to return from Autoselect mode or Exceeded Timing Limits ($DQ_5 = 1$) to Read/Reset mode, the Read/Reset operation is initiated by writing the Read/Reset command sequence into the command register. Microprocessor read cycles retrieve array data from the memory. The device remains enabled for reads until the command register contents are altered.

The device will automatically power-up in the Read/Reset state. In this case a command sequence is not required in order to read data. Standard microprocessor read cycles will retrieve array data. This default value ensures that no spurious alteration of the memory content occurs during the power transition. Refer to AC Read Characteristics and Timing Diagram for the specific timing parameters.

Autoselect Command

Flash memories are intended for use in applications where the local CPU alters memory contents. Therefore, manufacture and device codes must be accessible while the device resides in the target system. PROM programmers typically access the signature codes by raising A_9 to a higher voltage. However, multiplexing high voltage onto the address lines is not generally desired system design practice.

The device contains an Autoselect command operation to supplement traditional PROM programming methodology. The operation is initiated by writing the Autoselect command sequence into the command register.

The Autoselect command sequence is initiated first by writing two unlock cycles. This is followed by a third write cycle that contains the bank address (BA) and the Autoselect command. Then the manufacture and device codes can be read from the bank, and actual data from the memory cell can be read from another bank. The higher order address (A_{21}, A_{20}, A_{19}) required for reading out the manufacture and device codes demands the bank address (BA) set at the third write cycle.

Following the command write, in WORD mode, a read cycle from address (BA) 00h returns the manufacturer's code (Fujitsu = 04h) . And a read cycle at address (BA) 01h outputs device code. When 227Eh was output, this indicates that two additional codes, called Extended Device Codes will be required. Therefore the system may continue reading out these Extended Device Codes at the address of (BA) 0Eh, as well as at (BA) 0Fh. Notice that the above applies to WORD mode. The addresses and codes differ from those of BYTE mode. (Refer to Table 5.1 and 5.2.)

The sector state (protection or unprotection) will be informed by address (BA) 02h for $\times$ 16 ((BA) 04h for $\times$ 8) . Scanning the sector group addresses (A_{21}, A_{20}, A_{19}, A_{18}, A_{17}, A_{16}, A_{15}, A_{14}, A_{13}, and A_{12}) while (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) will produce a logical "1" at device output DQ_0 for a protected sector group. The programming verification should be performed by verifying sector group protection on the protected sector. (See Tables 2 and 3.)

The manufacture and device codes can be read from the selected bank. To read the manufacture and device codes and sector protection status from a non-selected bank, it is necessary to write the Read/Reset command sequence into the register. Autoselect command should then be written into the bank to be read.

If the software (program code) for Autoselect command is stored in the Flash memory, the device and manufacture codes should be read from the other bank, which does not contain the software.

To terminate the operation, it is necessary to write the Read/Reset command sequence into the register. To execute the Autoselect command during the operation, Read/Reset command sequence must be written before the Autoselect command.

Byte/Word Programming

The device is programmed on a byte-by-byte (or word-by-word) basis. Programming is a four bus cycle operation. There are two “unlock” write cycles. These are followed by the program set-up command and data write cycles. Addresses are latched on the falling edge of $\overline{CE}$ or $\overline{WE}$, whichever happens later, and the data is latched on the rising edge of $\overline{CE}$ or $\overline{WE}$, whichever happens first. The rising edge of $\overline{CE}$ or $\overline{WE}$ (whichever happens first) starts programming. Upon executing the Embedded Program Algorithm command sequence, the system is not required to provide further controls or timings. The device will automatically provide adequate internally generated program pulses and verify the programmed cell margin.

The system can determine the status of the program operation by using DQ_7 ($\overline{\text{Data}}$ Polling) , DQ_6 (Toggle Bit) or RY/$\overline{BY}$. The $\overline{\text{Data}}$ Polling and Toggle Bit must be performed at the memory location which is being programmed.

The automatic programming operation is completed when the data on DQ_7 is equivalent to data written to this bit at which time the device returns to the read mode and addresses are no longer latched (see Table 12, Hardware Sequence Flags). Therefore, the device requires that a valid address to the device be supplied by the system in this particular instance. Hence, $\overline{\text{Data}}$ Polling must be performed at the memory location which is being programmed.

If hardware reset occurs during the programming operation, the data being written is not guaranteed.

Programming is allowed in any sequence and across sector boundaries. Beware that a data “0” cannot be programmed back to a “1”. Attempting to do so may either hang up the device or result in an apparent success according to the data polling algorithm but a read from Read/Reset mode will show that the data is still “0”. Only erase operations can convert from “0”s to “1”s.

Figure 22 illustrates the Embedded Program™ Algorithm using typical command strings and bus operations.

Program Suspend/Resume

The Program Suspend command allows the system to interrupt a program operation so that data can be read from any address. Writing the Program Suspend command (B0h) during the Embedded Program operation immediately suspends the programming. The Program Suspend command may also be issued during a programming operation while an erase is suspended. The bank addresses of sector being programmed should be set when writing the Program Suspend command.

When the Program Suspend command is written during a programming process, the device halts the program operation within 1 µs and updates the status bits.

After the program operation has been suspended, the system can read data from any address. The data at program-suspended address is not valid. Normal read timing and command definitions apply.

After the Program Resume command (30h) is written, the device reverts to programming. The bank addresses of sectors being suspended should be set when writing the Program Resume command. The system can determine the status of the program operation using the DQ_7 or DQ_6 status bits, just as in the standard program operation. See “Write Operation Status” for more information.

The system may also write the Autoselect command sequence when the device in the Program Suspend mode. The device allows reading Autoselect codes at the addresses within programming sectors, since the codes are not stored in the memory. When the device exits the Autoselect mode, the device reverts to the Program Suspend mode, and is ready for another valid operation. See “Autoselect Command Sequence” for more information.

The system must write the Program Resume command (address bits are “Bank Address”) to exit from the Program Suspend mode and continue the programming operation. Further writes of the Resume command are ignored. Another Program Suspend command can be written after the device has resumed programming.

Chip Erase

Chip erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the chip erase command.

Chip erase does not require the user to program the device prior to erase. Upon executing the Embedded Erase Algorithm command sequence, the device will automatically program and verify the entire memory for an all-zero data pattern prior to electrical erase (Preprogram function) . The system is not required to provide any controls or timings during these operations.

The system can determine the status of the erase operation by using DQ_7 ($\overline{\text{Data}}$ Polling) , DQ_6 (Toggle Bit) or RY/$\overline{\text{BY}}$. The chip erase begins on the rising edge of the last $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens first in the command sequence, and terminates when the data on DQ_7 is “1” (see Write Operation Status section), at which time the device returns to the read mode.

Chip Erase Time; Sector Erase Time × All sectors + Chip Program Time (Preprogramming)

Figure 23 illustrates the Embedded Erase™ Algorithm using typical command strings and bus operations.

Sector Erase

Sector erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the Sector Erase command. The sector address (any address location within the desired sector) is latched on the falling edge of $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens later, while the command (Data = 30h) is latched on the rising edge of $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens first. After time-out of “t_{TOW}” from the rising edge of the last sector erase command, the sector erase operation begins.

Multiple sectors may be erased concurrently by writing the six bus cycle operations on Table 4. This sequence is followed by writes of the Sector Erase command to addresses in other sectors desired to be concurrently erased. The time between writes must be less than “t_{TOW}”. Otherwise, that command will not be accepted and erasure will not start. It is recommended that processor interrupts be disabled during this time to guarantee such a condition. The interrupts can reoccur after the last Sector Erase command is written. A time-out of “t_{TOW}” from the rising edge of last $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens first, will initiate the execution of the Sector Erase command (s) . If another falling edge of $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens first occurs within the “t_{TOW}” time-out window, the timer is reset (monitor DQ_3 to determine if the sector erase timer window is still open, see section DQ_3, Sector Erase Timer). Resetting the device once execution has begun will corrupt the data in the sector. In that case, restart the erase on those sectors and allow them to complete (refer to Write Operation Status section for Sector Erase Timer operation). Loading the sector erase buffer may be done in any sequence and with any number of sectors (0 to 141) .

Sector erase does not require the user to program the device before erase. The device automatically programs all memory locations in the sector (s) to be erased prior to electrical erase (Preprogram function) . When erasing a sector, the rest remain unaffected. The system is not required to provide any controls or timings during these operations.

The system can determine the status of the erase operation by using DQ_7 ($\overline{\text{Data}}$ Polling) , DQ_6 (Toggle Bit) or RY/$\overline{\text{BY}}$.

The sector erase begins after the “t_{TOW}” time-out from the rising edge of $\overline{\text{CE}}$ or $\overline{\text{WE}}$, whichever happens first, for the last sector erase command pulse and terminates when the data on DQ_7 is “1” (see Write Operation Status section), at which time the device returns to the read mode. $\overline{\text{Data}}$ polling and Toggle Bit must be performed at an address within any of the sectors being erased.

Multiple Sector Erase Time = [Sector Erase Time + Sector Program Time (Preprogramming)] × Number of Sector Erase

In case of multiple sector erase across bank boundaries, a read from the bank (read-while-erase) to which sectors being erased belong cannot be performed.

Figure 23 illustrates the Embedded Erase™ Algorithm using typical command strings and bus operations.

Erase Suspend/Resume

The Erase Suspend command allows the user to interrupt Sector Erase operation and then reads data from or programs to a sector not being erased. This command is applicable ONLY during the Sector Erase operation which includes the time-out period for sector erase. Writing the Erase Suspend command (B0h) during the Sector Erase time-out results in immediate termination of the time-out period and suspension of the erase operation.

Writing the Erase Resume command (30h) resumes the erase operation. The bank address of sector being erased or erase-suspended should be set when writing the Erase Suspend or Erase Resume command.

When the Erase Suspend command is written during the Sector Erase operation, the device takes a maximum of "t_{SPD}" to suspend the erase operation. When the device has entered the erase-suspended mode, the RY/$\overline{BY}$ output pin will be at Hi-Z and the DQ_7 bit will be at logic "1", and DQ_6 will stop toggling. The user must use the address of the erasing sector for reading DQ_6 and DQ_7 to determine if the erase operation has been suspended. Further writes of the Erase Suspend command are ignored.

When the erase operation has been suspended, the device defaults to the erase-suspend-read mode. Reading data in this mode is the same as reading from the standard read mode, except that the data must be read from sectors that have not been erase-suspended. Reading successively from the erase-suspended sector while the device is in the erase-suspend-read mode will cause DQ_2 to toggle (see the section on DQ_2).

After entering the erase-suspend-read mode, the user can program the device by writing the appropriate command sequence for Program. This program mode is known as the erase-suspend-program mode. Again, it is the same as programming in the regular Program mode, except that the data must be programmed to sectors that are not erase-suspended. Reading successively from the erase-suspended sector while the device is in the erase-suspend-program mode will cause DQ_2 to toggle. The end of the erase-suspended Program operation is detected by the RY/$\overline{BY}$ output pin, $\overline{Data}$ polling of DQ_7 or by the Toggle Bit I (DQ_6), which is the same as the regular Program operation. Note that DQ_7 must be read from the Program address while DQ_6 can be read from any address within bank being erase-suspended.

To resume the operation of Sector Erase, the Resume command (30h) should be written to the bank being erase suspended. Any further writes of the Resume command at this point will be ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

Extended Command

(1) Fast Mode

The device has a Fast Mode function. It dispenses with the initial two unclock cycles required in the standard program command sequence by writing the Fast Mode command into the command register. In this mode, the required bus cycle for programming consists of two bus cycles instead of four in standard program command. Do not write erase command in this mode. The read operation is also executed after exiting from the fast mode. To exit from this mode, it is necessary to write Fast Mode Reset command into the command register. The first cycle must contain the bank address (see Figure 29) .The V_{CC} active current is required even if $\overline{CE} = V_{IH}$ during Fast Mode.

(2) Fast Programming

During Fast Mode, programming can be executed with two bus cycle operation. The Embedded Program Algorithm is executed by writing program set-up command (A0h) and data write cycles (PA/PD) (see Figure 29) .

(3) CFI (Common Flash Memory Interface)

The CFI (Common Flash Memory Interface) specification outlines device and the host system software interrogation handshake, which allows specific vendor-specified software algorithms to be used for entire families of devices. This allows device-independent, JEDEC ID-independent and forward-and backward-compatible software support for the specified flash device families. Refer to CFI specification in detail.

The operation is initiated by writing the query command (98h) into the command register. The bank address should be set when writing this command. Then the device information can be read from the bank, and data from the memory cell can be read from the another bank. The higher order address (A_{21}, A_{20}, A_{19}) required for reading out the CFI Codes requires that the bank address (BA) be set at the write cycle. Following the command write, a read cycle from specific address retrieves device information. Please note that output data of upper byte (DQ_{15} to DQ_8) is "0" in word mode (16 bit) read. Refer to CFI code table (Table 12) . To terminate operation, it is necessary to write the read/reset command sequence into the register.

Hidden ROM (Hi-ROM) Region

The Hi-ROM feature provides a Flash memory region that the system may access through a new command sequence. This is primarily intended for customers who wish to use an Electronic Serial Number (ESN) in the device with the ESN protected against modification. Once the Hi-ROM region is protected, any further modification of that region becomes impossible. This ensures the security of the ESN once the product is shipped to the field.

The Hi-ROM region is 256 bytes in length and is stored at the same address of the “outermost” 8 Kbyte boot sector in Bank A. The device occupies the address of the byte mode 000000h to 0000FFh (word mode 000000h to 00007Fh) . After the system has written the Enter Hi-ROM command sequence, the system may read the Hi-ROM region by using the addresses normally occupied by the boot sector (particular area of SA0) . That is, the device sends all commands that would normally be sent to the boot sector (particular area of SA0) to the Hi-ROM region. This mode of operation continues until the system issues the Exit Hi-ROM command sequence, or until power is removed from the device. On power-up, or following a hardware reset, the device reverts to sending commands to the boot sector.

When reading the Hi-ROM region, either change addresses or change $\overline{CE}$ pin from “H” to “L”. The same procedure should be taken (changing addresses or $\overline{CE}$ pin from “H” to “L”) after the system issues the Exit Hi-ROM command sequence to read actual memory cell data.

Hidden ROM (Hi-ROM) Entry Command

The device has a Hidden ROM area with One Time Protect function. This area is to enter the security code and to unable the change of the code once set. Programming is allowed in this area until it is protected. However, once it gets protected, it is impossible to unprotect. Therefore, extreme caution is required.

The hidden ROM area is 256 bytes. This area is normally the “outermost” 8 Kbyte boot block area in Bank A. Therefore, write the Hidden ROM entry command sequence to enter the Hidden ROM area. It is called Hidden ROM mode when the Hidden ROM area appears.

Sectors other than the boot block area SA0 can be read during Hidden ROM mode. Read/program of the Hidden ROM area is possible during Hidden ROM mode. Write the Hidden ROM reset command sequence to exit the Hidden ROM mode. The bank address of the Hidden ROM should be set on the third cycle of this reset command sequence.

In Hidden ROM mode, the simultaneous operation cannot be executed multi-function mode between the Hidden ROM area and the Bank A.

Hidden ROM (Hi-ROM) Program Command

To program the data to the Hidden ROM area, write the Hidden ROM program command sequence during Hidden ROM mode. This command is the same as the usual program command, except that it needs to write the command during Hidden ROM mode. Therefore the detection of completion method is the same as in the past, using the DQ_7 data pooling, DQ_6 toggle bit and RY/$\overline{BY}$ pin. You should pay attention to the address to be programmed. If an address not in the Hidden ROM area is selected, the previous data will be deleted.

Hidden ROM (Hi-ROM) Protect Command

There are two methods to protect the Hidden ROM area. One is to write the sector group protect setup command (60h) , set the sector address in the Hidden ROM area and (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) , and write the sector group protect command (60h) during the Hidden ROM mode. The same command sequence may be used because it is the same as the extension sector group protect in the past, except that it is in the Hidden ROM mode and does not apply high voltage to the $\overline{RESET}$ pin. Please refer to “Function Explanation Extended Sector Group Protection” for details of extension sector group protect setting.

The other method is to apply high voltage (V_{ID}) to A_9 and $\overline{OE}$, set the sector address in the Hidden ROM area and (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) , and apply the write pulse during the Hidden ROM mode. To verify the protect circuit, apply high voltage (V_{ID}) to A_9, specify (A_6, A_3, A_2, A_1, A_0) = (0, 0, 0, 1, 0) and the sector address in the Hidden ROM area, and read. When “1” appears on DQ_0, the protect setting is completed. “0” will appear on DQ_0 if it is not protected. Apply write pulse again. The same command sequence could be used for the above method because other than the Hidden ROM mode, it is the same as the sector group protect previously mentioned. Refer to “Function Explanation Secor Group Protection” for details of the sector group protect setting.

Take note that other sector groups will be affected if an address other than those for the Hidden ROM area is selected for the sector group address, so please be careful. Pay close attention that once it is protected, protection CANNOT BE CANCELLED.

Write Operation Status

Detailed in Table 12 are all the status flags which can determine the status of the bank for the current mode operation. The read operation from the bank which doesn’t operate Embedded Algorithm returns data of memory cells. These bits offer a method for determining whether an Embedded Algorithm is properly completed. The information on DQ_2 is address-sensitive. This means that if an address from an erasing sector is consecutively read, the DQ_2 bit will toggle. However, DQ_2 will not toggle if an address from a non-erasing sector is consecutively read. This allows users to determine which sectors are in erase and which are not.

The status flag is not output from banks (non-busy banks) which do not execute Embedded Algorithms. For example, a bank (busy bank) is executing an Embedded Algorithm. When the read sequence is [1] < busy bank > , [2] < non-busy bank > , [3] < busy bank > , the DQ_6 toggles in the case of [1] and [3]. In case of [2], the data of memory cells are output. In the erase-suspend read mode with the same read sequence, DQ_6 will not be toggled in [1] and [3].

In the erase suspend read mode, DQ_2 is toggled in [1] and [3]. In case of [2], the data of memory cell is output.

Table 12 Hardware Sequence Flags

Status			DQ_7	DQ_6	DQ_5	DQ_3	DQ_2
In Progress	Embedded Program Algorithm		$\overline{DQ_7}$	Toggle	0	0	1
	Embedded Erase Algorithm		0	Toggle	0	1	Toggle *1
	Program Suspended Mode	Program Suspend Read (Program Suspended Sector)	Data	Data	Data	Data	Data
		Program Suspend Read (Non-Program Suspended Sector)	Data	Data	Data	Data	Data
	Erase Suspended Mode	Erase Suspend Read (Erase Suspended Sector)	1	1	0	0	Toggle
		Erase Suspend Read (Non-Erase Suspended Sector)	Data	Data	Data	Data	Data
		Erase Suspend Program (Non-Erase Suspended Sector)	$\overline{DQ_7}$	Toggle	0	0	1 *2
Exceeded Time Limits	Embedded Program Algorithm		$\overline{DQ_7}$	Toggle	1	0	1
	Embedded Erase Algorithm		0	Toggle	1	1	N/A
	Erase Suspended Mode	Erase Suspend Program (Non-Erase Suspended Sector)	$\overline{DQ_7}$	Toggle	1	0	N/A

*1: Successive reads from the erasing or erase-suspend sector will cause DQ_2 to toggle.

*2: Reading from non-erase suspend sector address will indicate logic “1” at the DQ_2 bit.

Notes: 1. DQ_0 and DQ_1 are reserve pins for future use.
2. DQ_4 is limited to Fujitsu internal use.

DQ_7
$\overline{\text{Data}}$ Polling

The device features $\overline{\text{Data}}$ Polling as a method to indicate to the host that the Embedded Algorithms are in progress or completed. During the Embedded Program Algorithm, an attempt to read the device will produce a complement of data last written to DQ_7. Upon completion of the Embedded Program Algorithm, an attempt to read the device will produce true data last written to DQ_7. During the Embedded Erase Algorithm, an attempt to read the device will produce a “0” at the DQ_7 output. Upon completion of the Embedded Erase Algorithm, an attempt to read device will produce a “1” on DQ_7. The flowchart for $\overline{\text{Data}}$ Polling (DQ_7) is shown in Figure 24.

For programming, the $\overline{\text{Data}}$ Polling is valid after the rising edge of the fourth write pulse in the four write pulse sequences.

For chip erase and sector erase, the $\overline{\text{Data}}$ Polling is valid after the rising edge of the sixth write pulse in the six write pulse sequences. $\overline{\text{Data}}$ Polling must be performed at sector addresses of sectors being erased, not protected sectors. Otherwise the status may become invalid.

If a program address falls within a protected sector, $\overline{\text{Data}}$ Polling on DQ_7 is active for approximately 1 µs, then that bank returns to the read mode. After an erase command sequence is written, if all sectors selected for erasing are protected, $\overline{\text{Data}}$ Polling on DQ_7 is active for approximately 400 µs, then the bank returns to read mode.

Once the Embedded Algorithm operation is close to being completed, the device data pins (DQ_7) may change asynchronously while the output enable ($\overline{OE}$) is asserted low. This means that device is driving status information on DQ_7 at one instant, and then that byte’s valid data at the next instant. Depending on when the system samples the DQ_7 output, it may read the status or valid data. Even if device has completed the Embedded Algorithm operation and DQ_7 has a valid data, data outputs on DQ_0 to DQ_6 may still be invalid. The valid data on DQ_0 to DQ_7 will be read on successive read attempts.

The $\overline{\text{Data}}$ Polling feature is active only during the Embedded Programming Algorithm, Embedded Erase Algorithm or sector erase time-out. (See Table 12.)

See Figure 9 for the $\overline{\text{Data}}$ Polling timing specifications and diagrams.

DQ_6
Toggle Bit I

The device also features the “Toggle Bit I” as a method to indicate to the host system that the Embedded Algorithms are in progress or completed.

During Embedded Program or Erase Algorithm cycle, successive attempts to read ($\overline{OE}$ toggling) data from the busy bank will result in DQ_6 toggling between one and zero. Once the Embedded Program or Erase Algorithm cycle is completed, DQ_6 will stop toggling and valid data will be read on the next successive attempts. During programming, the Toggle Bit I is valid after the rising edge of the fourth write pulse in the four write pulse sequences. For chip erase and sector erase, the Toggle Bit I is valid after the rising edge of the sixth write pulse in the six write pulse sequences. The Toggle Bit I is active during the sector time out.

In programming, if the sector being written is protected, the toggle bit will toggle for about 1 µs and then stop toggling with data unchanged. In erase, the device will erase all selected sectors except for protected ones. If all selected sectors are protected, the chip will toggle the toggle bit for about 400 µs and then drop back into read mode, having data kept remained.

Either $\overline{CE}$ or $\overline{OE}$ toggling will cause DQ_6 to toggle. In addition, an Erase Suspend/Resume command will cause DQ_6 to toggle.

The system can use DQ_6 to determine whether a sector is actively erased or is erase-suspended. When a bank is actively erased (that is, the Embedded Erase Algorithm is in progress) , DQ_6 toggles. When a bank enters the Erase Suspend mode, DQ_6 stops toggling. Successive read cycles during erase-suspend-program cause DQ_6 to toggle.

To operate toggle bit function properly, $\overline{CE}$ or $\overline{OE}$ must be high when bank address is changed.

See Figure 10 for the Toggle Bit I timing specifications and diagrams.

DQ_5

Exceeded Timing Limits

DQ_5 will indicate if the program or erase time has exceeded the specified limits (internal pulse count) . Under these conditions DQ_5 will produce "1". This is a failure condition indicating that the program or erase cycle was not successfully completed. $\overline{\text{Data}}$ Polling is only operating function of the device under this condition. The $\overline{CE}$ circuit will partially power down device under these conditions (to approximately 2 mA) . The $\overline{OE}$ and $\overline{WE}$ pins will control the output disable functions as described in Tables 2 and 3.

The DQ_5 failure condition may also appear if a user tries to program a non-blank location without pre-erase. In this case the device locks out and never completes the Embedded Algorithm operation. Hence, the system never reads valid data on DQ_7 bit and DQ_6 never stop toggling. Once the device has exceeded timing limits, the DQ_5 bit will indicate a "1." Please note that this is not a device failure condition since the device was incorrectly used. If this occurs, reset device with the command sequence.

DQ_3

Sector Erase Timer

After completion of the initial sector erase command sequence, sector erase time-out begins. DQ_3 will remain low until the time-out is completed. $\overline{\text{Data}}$ Polling and Toggle Bit are valid after the initial sector erase command sequence.

If $\overline{\text{Data}}$ Polling or the Toggle Bit I indicates that a valid erase command has been written, DQ_3 may be used to determine whether the sector erase timer window is still open. If DQ_3 is high ("1") the internally controlled erase cycle has begun. If DQ_3 is low ("0") , the device will accept additional sector erase commands. To insure the command has been accepted, the system software should check the status of DQ_3 prior to and following each subsequent Sector Erase command. If DQ_3 were high on the second status check, the command may not have been accepted.

See Table 12 : Hardware Sequence Flags.

DQ_2

Toggle Bit II

This toggle bit II, along with DQ_6, can be used to determine whether the device is in the Embedded Erase Algorithm or in Erase Suspend.

Successive reads from the erasing sector will cause DQ_2 to toggle during the Embedded Erase Algorithm. If the device is in the erase-suspended-read mode, successive reads from the erase-suspended sector will cause DQ_2 to toggle. When the device is in the erase-suspended-program mode, successive reads from the non-erase suspended sector will indicate a logic "1" at the DQ_2 bit.

DQ_6 is different from DQ_2 in that DQ_6 toggles only when the standard program or Erase, or Erase Suspend Program operation is in progress. The behavior of these two status bits, along with that of DQ_7, is summarized as follows :

For example, DQ_2 and DQ_6 can be used together to determine if the erase-suspend-read mode is in progress. (DQ_2 toggles while DQ_6 does not.) See also Table 13 and Figure 12.

Furthermore DQ_2 can also be used to determine which sector is being erased. At the erase mode, DQ_2 toggles if this bit is read from an erasing sector.

To operate toggle bit function properly, $\overline{CE}$ or $\overline{OE}$ must be high when bank address is changed.

Reading Toggle Bits DQ_6/DQ_2

Whenever the system initially begins reading toggle bit status, it must read DQ_7 to DQ_0 at least twice in a row to determine whether a toggle bit is toggling. Typically a system would note and store the value of the toggle bit after the first read. After the second read, the system would compare the new value of the toggle bit with the first. If the toggle bit is not toggling, the device has completed the program or erase operation. The system can read array data on DQ_7 to DQ_0 on the following read cycle.

However, if, after the initial two read cycles, the system determines that the toggle bit is still toggling, the system also should note whether the value of DQ_5 is high (see the section on DQ_5) . If it is, the system should then determine again whether the toggle bit is toggling, since the toggle bit may have stopped toggling just as DQ_5 went high. If the toggle bit is no longer toggling, the device has successfully completed the program or erase operation. If it is still toggling, the device did not complete the operation successfully, and the system must write the reset command to return to reading array data.

The remaining scenario is that the system initially determines that the toggle bit is toggling and DQ_5 has not gone high. The system may continue to monitor the toggle bit and DQ_5 through successive read cycles, determining the status as described in the previous paragraph. Alternatively, it may choose to perform other system tasks. In this case, the system must start at the beginning of the algorithm when it returns to determine the status of the operation. (Refer to Figure 25.)

Table 13 Toggle Bit Status

Mode	DQ_7	DQ_6	DQ_2
Program	$\overline{DQ_7}$	Toggle	1
Erase	0	Toggle	Toggle (Note)
Erase-Suspend Read (Erase-Suspended Sector)	1	1	Toggle
Erase-Suspend Program	$\overline{DQ_7}$	Toggle	1 (Note)

Note : Successive reads from the erasing or erase-suspend sector will cause DQ_2 to toggle. Reading from the non-erase suspend sector address will indicate logic “1” at the DQ_2 bit.

RY/$\overline{\text{BY}}$

Ready/Busy

The device provides a RY/$\overline{\text{BY}}$ open-drain output pin as a way to indicate to the host system that Embedded Algorithms are either in progress or have been completed. If output is low, the device is busy with either a program or erase operation. If output is high, the device is ready to accept any read/write or erase operation. If the device is placed in an Erase Suspend mode, RY/$\overline{\text{BY}}$ output will be high.

During programming, the RY/$\overline{\text{BY}}$ pin is driven low after the rising edge of the fourth write pulse. During an erase operation, the RY/$\overline{\text{BY}}$ pin is driven low after the rising edge of the sixth write pulse. The RY/$\overline{\text{BY}}$ pin will indicate a busy condition during $\overline{\text{RESET}}$ pulse. Refer to Figures 13 and 14 for a detailed timing diagram. The RY/$\overline{\text{BY}}$ pin is pulled high in standby mode.

Since this is an open-drain output, RY/$\overline{\text{BY}}$ pins can be tied together in parallel with a pull-up resistor to V_{CC}.

Byte/Word Configuration

$\overline{\text{BYTE}}$ pin selects byte (8-bit) mode or word (16-bit) mode for the device. When this pin is driven high, the device operates in word (16-bit) mode. Data is read and programmed at DQ_{15} to DQ_0. When this pin is driven low, the device operates in byte (8-bit) mode. In this mode, the DQ_{15}/A_{-1} pin becomes the lowest address bit, and DQ_{14} to DQ_8 bits are tri-stated. However, the command bus cycle is always an 8-bit operation and hence commands are written at DQ_7 to DQ_0 and DQ_{15} to DQ_8 bits are ignored. Refer to Figures 15, 16 and 17 for the timing diagram.

Data Protection

The device is designed to offer protection against accidental erasure or programming caused by spurious system level signals that may exist during power transitions. During power-up, the device automatically resets the internal state machine in Read mode. Also, with its control register architecture, alteration of memory contents only occurs after successful completion of specific multi-bus cycle command sequences.

The device also incorporates several features to prevent inadvertent write cycles resulting from V_{CC} power-up and power-down transitions or system noise.

Low V_{CC} Write Inhibit

To avoid initiation of a write cycle during V_{CC} power-up and power-down, a write cycle is locked out for V_{CC} less than V_{LKO} (Min.) . If $V_{CC} < V_{LKO}$, the command register is disabled and all internal program/erase circuits are disabled. Under this condition, the device will reset to the read mode. Subsequent writes will be ignored until the V_{CC} level is greater than V_{LKO}. It is the user's responsibility to ensure that the control pins are logically correct to prevent unintentional writes when V_{CC} is above V_{LKO} (Min.) .

If the Embedded Erase Algorithm is interrupted, the intervened erasing sector (s) is (are) not valid.

Write Pulse "Glitch" Protection

Noise pulses of less than 3 ns (typical) on $\overline{OE}$, $\overline{CE}$ or $\overline{WE}$ will not initiate a write cycle.

Logical Inhibit

Writing is inhibited by holding any one of $\overline{OE} = V_{IL}$, $\overline{CE} = V_{IH}$ or $\overline{WE} = V_{IH}$. To initiate a write cycle $\overline{CE}$ and $\overline{WE}$ must be a logical zero while $\overline{OE}$ is a logical one.

Power-Up Write Inhibit

Power-up of the device with $\overline{WE} = \overline{CE} = V_{IL}$ and $\overline{OE} = V_{IH}$ will not accept commands on the rising edge of $\overline{WE}$. The internal state machine is automatically reset to the read mode on power-up.

■ ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Rating		Unit
		Min.	Max.	
Storage Temperature	Tstg	–55	+125	°C
Ambient Temperature with Power Applied	Ta	–40	+85	°C
Voltage with Respect to Ground All pins except A_9, $\overline{OE}$, and $\overline{RESET}$ (Note 1)	V_{IN}, V_{OUT}	–0.5	V_{CC} + 0.5	V
Power Supply Voltage (Note 1)	V_{CC}	–0.5	+4.0	V
A_9, $\overline{OE}$, and $\overline{RESET}$ (Note 2)	V_{IN}	–0.5	+13.0	V
$\overline{WP}$/ACC (Note 3)	V_{ACC}	–0.5	+10.5	V

Notes : 1. Minimum DC voltage on input or I/O pins is –0.5 V. During voltage transitions, input or I/O pins may undershoot V_{SS} to –2.0 V for periods of up to 20 ns. Maximum DC voltage on input or I/O pins is V_{CC}+0.5 V. During voltage transitions, input or I/O pins may overshoot to V_{CC}+2.0 V for periods of up to 20 ns.

2. Minimum DC input voltage on A_9, $\overline{OE}$ and $\overline{RESET}$ pins is –0.5 V. During voltage transitions, A_9, $\overline{OE}$ and $\overline{RESET}$ pins may undershoot V_{SS} to –2.0 V for periods of up to 20 ns. Voltage difference between input and supply voltage (V_{IN} - V_{CC}) does not exceed +9.0 V. Maximum DC input voltage on A_9, $\overline{OE}$ and $\overline{RESET}$ pins is +13.0 V which may overshoot to +14.0 V for periods of up to 20 ns.

3. Minimum DC input voltage on $\overline{WP}$/ACC pin is –0.5 V. During voltage transitions, $\overline{WP}$/ACC pin may undershoot V_{SS} to –2.0 V for periods of up to 20 ns. Maximum DC input voltage on $\overline{WP}$/ACC pin is +10.5 V which may overshoot to +12.0 V for periods of up to 20 ns when Vcc is applied.

WARNING: Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

■ RECOMMENDED OPERATING RANGES

Parameter	Symbol	Part No.	Ranges		Unit
			Min.	Max.	
Ambient Temperature	Ta	MBM29DL640E 80	–20	+70	°C
		MBM29DL640E 90/12	–40	+85	°C
Power Supply Voltage	V_{CC}	MBM29DL640E 80	+3.0	+3.6	V
		MBM29DL640E 90/12	+2.7	+3.6	V

WARNING: The recommended operating conditions are required in order to ensure the normal operation of the semiconductor device. All of the device's electrical characteristics are warranted when the device is operated within these ranges.

Always use semiconductor devices within their recommended operating condition ranges. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their FUJITSU representatives beforehand.

Operating ranges define those limits between which the proper device function is guaranteed.

■ MAXIMUM OVERSHOOT/UNDERSHOOT

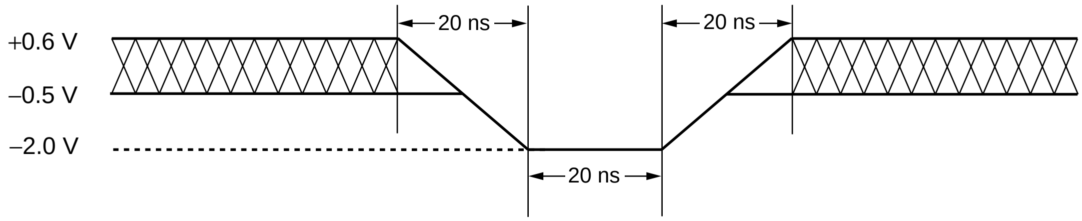


Figure 1   Maximum Undershoot Waveform

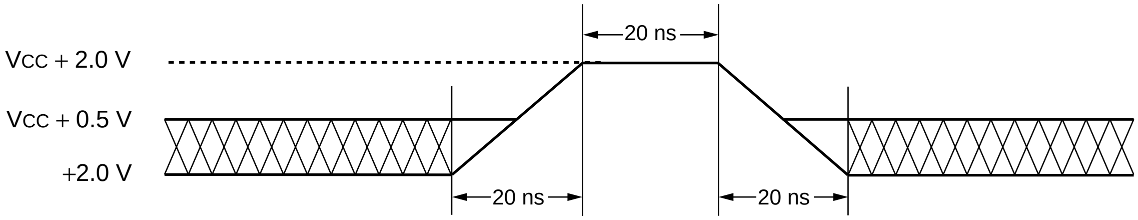
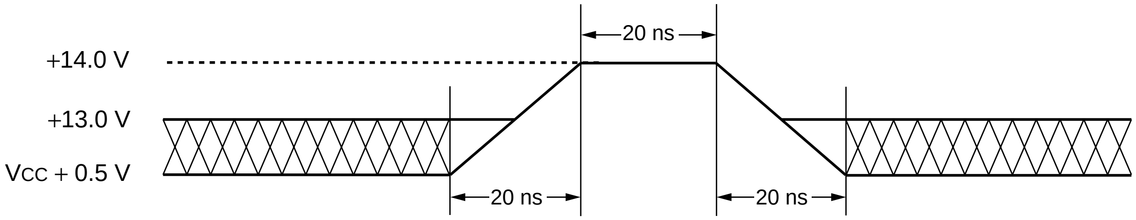


Figure 2   Maximum Overshoot Waveform 1



Note : This waveform is applied for A9, $\overline{\text{OE}}$ and $\overline{\text{RESET}}$.

Figure 3   Maximum Overshoot Waveform 2

■ ELECTRICAL CHARACTERISTICS

1. DC Characteristics

Parameter	Symbol	Conditions		Value Min.	Value Typ.	Value Max.	Unit
Input Leakage Current	I_{LI}	$V_{IN} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$ Max.		–1.0	—	+1.0	µA
Output Leakage Current	I_{LO}	$V_{OUT} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$ Max.		–1.0	—	+1.0	µA
A_9, $\overline{OE}$, $\overline{RESET}$ Inputs Leakage Current	I_{LIT}	$V_{CC} = V_{CC}$ Max., A_9, $\overline{OE}$, $\overline{RESET}$ = 12.5 V		—	—	+35	µA
$\overline{WP}$/ACC Accelerated Program Current	I_{LIA}	$V_{CC} = V_{CC}$ Max. $\overline{WP}$/ACC = V_{ACC} Max.		—	—	20	mA
V_{CC} Active Current *1	I_{CC1}	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$, f = 5 MHz	Byte	—	—	16	mA
			Word	—	—	18	
		$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$, f = 1 MHz	Byte	—	—	7	mA
			Word	—	—	7	
V_{CC} Active Current *2	I_{CC2}	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$		—	—	40	mA
V_{CC} Current (Standby)	I_{CC3}	$V_{CC} = V_{CC}$ Max., $\overline{CE} = V_{CC} \pm 0.3$ V, $\overline{RESET} = V_{CC} \pm 0.3$ V $\overline{WP}$/ACC = $V_{CC} \pm 0.3$ V		—	1	5	µA
V_{CC} Current (Standby, Reset)	I_{CC4}	$V_{CC} = V_{CC}$ Max., $\overline{RESET} = V_{SS} \pm 0.3$ V		—	1	5	µA
V_{CC} Current (Automatic Sleep Mode) *3	I_{CC5}	$V_{CC} = V_{CC}$ Max., $\overline{CE} = V_{SS} \pm 0.3$ V, $\overline{RESET} = V_{CC} \pm 0.3$ V $V_{IN} = V_{CC} \pm 0.3$ V or $V_{SS} \pm 0.3$ V		—	1	5	µA
V_{CC} Active Current *5 (Read-While-Program)	I_{CC6}	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$	Byte	—	—	56	mA
			Word	—	—	58	
V_{CC} Active Current *5 (Read-While-Erase)	I_{CC7}	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$	Byte	—	—	56	mA
			Word	—	—	58	
V_{CC} Active Current (Erase-Suspend-Program)	I_{CC8}	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IH}$		—	—	40	mA
Input Low Level	V_{IL}	—		–0.5	—	0.6	V
Input High Level	V_{IH}	—		2.0	—	$V_{CC} + 0.3$	V
Voltage for Autoselect and Sector Protection (A_9, $\overline{OE}$, $\overline{RESET}$) *4	V_{ID}	—		11.5	12	12.5	V
Voltage for $\overline{WP}$/ACC Sector Protection/Unprotection and Program Acceleration *4	V_{ACC}	—		8.5	9.0	9.5	V
Output Low Voltage Level	V_{OL}	I_{OL} = 100 µA, $V_{CC} = V_{CC}$ Min.		—	—	0.45	V
Output High Voltage Level	V_{OH1}	I_{OH} = –2.0 mA, $V_{CC} = V_{CC}$ Min.		2.4	—	—	V
	V_{OH2}	I_{OH} = –100 µA		$V_{CC} - 0.4$	—	—	V
Low V_{CC} Lock-Out Voltage	V_{LKO}	—		2.3	2.4	2.5	V

*1: The I_{CC} current listed includes both the DC operating current and the frequency dependent component.

*2: I_{CC} active while Embedded Algorithm (program or erase) is in progress.

*3: Automatic sleep mode enables the low power mode when address remain stable for 150 ns.

*4: Applicable for only V_{CC}.

*5: Embedded Algorithm (program or erase) is in progress. (@5 MHz)

2. AC Characteristics

- **Read Only Operations Characteristics**

Parameter	Symbol		Condition	Value (Note)						Unit
				80		90		12		
	JEDEC	Standard		Min.	Max.	Min.	Max.	Min.	Max.	
Read Cycle Time	t_{AVAV}	t_{RC}	—	80	—	90	—	120	—	ns
Address to Output Delay	t_{AVQV}	t_{ACC}	$\overline{CE} = V_{IL}$ $\overline{OE} = V_{IL}$	—	80	—	90	—	120	ns
Chip Enable to Output Delay	t_{ELQV}	t_{CE}	$\overline{OE} = V_{IL}$	—	80	—	90	—	120	ns
Output Enable to Output Delay	t_{GLQV}	t_{OE}	—	—	30	—	35	—	50	ns
Chip Enable to Output High-Z	t_{EHQZ}	t_{DF}	—	—	25	—	30	—	30	ns
Output Enable to Output High-Z	t_{GHQZ}	t_{DF}	—	—	25	—	30	—	30	ns
Output Hold Time From Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurs First	t_{AXQX}	t_{OH}	—	0	—	0		0	—	ns
$\overline{RESET}$ Pin Low to Read Mode	—	t_{READY}	—	—	20	—	20	—	20	µs
$\overline{CE}$ to $\overline{BYTE}$ Switching Low or High	—	t_{ELFL} t_{ELFH}	—	—	5	—	5	—	5	ns

Note : Test Conditions :
Output Load : 1 TTL gate and 30 pF (MBM29DL640E-80)
1 TTL gate and 100 pF (MBM29DL640E-90/120)
Input rise and fall times : 5 ns
Input pulse levels : 0.0 V to 3.0 V
Timing measurement reference level
Input : 1.5 V
Output : 1.5 V

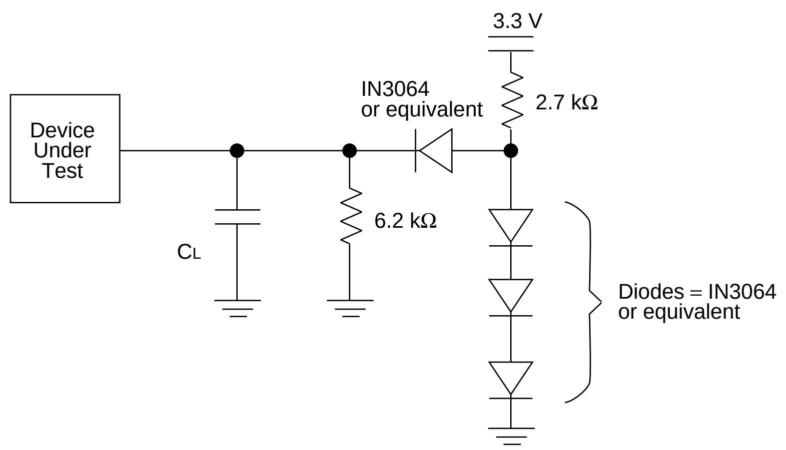


Figure 4 Test Conditions

• **Write/Erase/Program Operations**

Parameter		Symbol		Value									Unit
				80			90			12			
		JEDEC	Standard	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Write Cycle Time		t_{AVAV}	t_{WC}	80	—	—	90	—	—	120	—	—	ns
Address Setup Time		t_{AVWL}	t_{AS}	0	—	—	0	—	—	0	—	—	ns
Address Setup Time to $\overline{OE}$ Low During Toggle Bit Polling		—	t_{ASO}	12	—	—	15	—	—	15	—	—	ns
Address Hold Time		t_{WLAX}	t_{AH}	45	—	—	45	—	—	50	—	—	ns
Address Hold Time from $\overline{CE}$ or $\overline{OE}$ High During Toggle Bit Polling		—	t_{AHT}	0	—	—	0	—	—	0	—	—	ns
Data Setup Time		t_{DVWH}	t_{DS}	30	—	—	35	—	—	50	—	—	ns
Data Hold Time		t_{WHDX}	t_{DH}	0	—	—	0	—	—	0	—	—	ns
Output Enable Hold Time	Read	—	t_{OEH}	0	—	—	0	—	—	0	—	—	ns
	Toggle and $\overline{Data}$ Polling			10	—	—	10	—	—	10	—	—	ns
$\overline{CE}$ High During Toggle Bit Polling		—	t_{CEPH}	20	—	—	20	—	—	20	—	—	ns
$\overline{OE}$ High During Toggle Bit Polling		—	t_{OEPH}	20	—	—	20	—	—	20	—	—	ns
Read Recover Time Before Write		t_{GHWL}	t_{GHWL}	0	—	—	0	—	—	0	—	—	ns
Read Recover Time Before Write		t_{GHEL}	t_{GHEL}	0	—	—	0	—	—	0	—	—	ns
$\overline{CE}$ Setup Time		t_{ELWL}	t_{CS}	0	—	—	0	—	—	0	—	—	ns
$\overline{WE}$ Setup Time		t_{WLEL}	t_{WS}	0	—	—	0	—	—	0	—	—	ns
$\overline{CE}$ Hold Time		t_{WHEH}	t_{CH}	0	—	—	0	—	—	0	—	—	ns
$\overline{WE}$ Hold Time		t_{EHWH}	t_{WH}	0	—	—	0	—	—	0	—	—	ns
Write Pulse Width		t_{WLWH}	t_{WP}	35	—	—	35	—	—	50	—	—	ns
$\overline{CE}$ Pulse Width		t_{ELEH}	t_{CP}	35	—	—	35	—	—	50	—	—	ns
Write Pulse Width High		t_{WHWL}	t_{WPH}	25	—	—	30	—	—	30	—	—	ns
$\overline{CE}$ Pulse Width High		t_{EHEL}	t_{CPH}	25	—	—	30	—	—	30	—	—	ns
Programming Operation	Byte	t_{WHWH1}	t_{WHWH1}	—	8	—	—	8	—	—	8	—	µs
	Word			—	16	—	—	16	—	—	16	—	µs
Sector Erase Operation *1		t_{WHWH2}	t_{WHWH2}	—	1	—	—	1	—	—	1	—	s
V_{CC} Setup Time		—	t_{VCS}	50	—	—	50	—	—	50	—	—	µs
Rise Time to V_{ID} *2		—	t_{VIDR}	500	—	—	500	—	—	500	—	—	ns
Rise Time to V_{ACC} *3		—	t_{VACCR}	500	—	—	500	—	—	500	—	—	ns
Voltage Transition Time *2		—	t_{VLHT}	4	—	—	4	—	—	4	—	—	µs
Write Pulse Width *2		—	t_{WPP}	100	—	—	100	—	—	100	—	—	µs

(Continued)

(Continued)

Parameter	Symbol		Value									Unit
			80			90			12			
	JEDEC	Standard	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
$\overline{OE}$ Setup Time to $\overline{WE}$ Active *2	—	t_{OESP}	4	—	—	4	—	—	4	—	—	µs
$\overline{CE}$ Setup Time to $\overline{WE}$ Active *2	—	t_{CSP}	4	—	—	4	—	—	4	—	—	µs
Recover Time from RY/$\overline{BY}$	—	t_{RB}	0	—	—	0	—	—	0	—	—	ns
$\overline{RESET}$ Pulse Width	—	t_{RP}	500	—	—	500	—	—	500	—	—	ns
$\overline{RESET}$ High Level Period Before Read	—	t_{RH}	200	—	—	200	—	—	200	—	—	ns
$\overline{BYTE}$ Switching Low to Output High-Z	—	t_{FLQZ}	—	—	30	—	—	30	—	—	40	ns
$\overline{BYTE}$ Switching High to Output Active	—	t_{FHQV}	—	—	80	—	—	90	—	—	120	ns
Program/Erase Valid to RY/$\overline{BY}$ Delay	—	t_{BUSY}	—	—	90	—	—	90	—	—	90	ns
Delay Time from Embedded Output Enable	—	t_{EOE}	—	—	80	—	—	90	—	—	120	ns
Erase Time-out Time	—	t_{TOW}	50	—	—	50		—	50	—	—	µs
Erase Suspend Transition Time	—	t_{SPD}	—	—	20	—	—	20	—	—	20	µs

*1: This does not include preprogramming time.

*2: This timing is for Sector Group Protection operation.

*3: This timing is for Accelerated Program operation.

■ ERASE AND PROGRAMMING PERFORMANCE

Parameter	Limits			Unit	Comments
	Min.	Typ.	Max.		
Sector Erase Time	—	1	10	s	Excludes programming time prior to erasure
Word Programming Time	—	16	360	µs	Excludes system-level overhead
Byte Programming Time	—	8	300	µs	
Chip Programming Time	—	—	200	s	Excludes system-level overhead
Program/Erase Cycle	100,000	—	—	cycle	—

■ TSOP (I) PIN CAPACITANCE

Parameter	Symbol	Condition	Value		Unit
			Typ.	Max.	
Input Capacitance	C_{IN}	$V_{IN} = 0$	6	7.5	pF
Output Capacitance	C_{OUT}	$V_{OUT} = 0$	8.5	12	pF
Control Pin Capacitance	C_{IN2}	$V_{IN} = 0$	8	11	pF
$\overline{WP}$/ACC Pin Capacitance	C_{IN3}	$V_{IN} = 0$	9	11	pF

Note : Test conditions Ta = 25 °C, f = 1.0 MHz

■ FBGA PIN CAPACITANCE

Parameter	Symbol	Condition	Value		Unit
			Typ.	Max.	
Input Capacitance	C_{IN}	$V_{IN} = 0$	TBD	TBD	pF
Output Capacitance	C_{OUT}	$V_{OUT} = 0$	TBD	TBD	pF
Control Pin Capacitance	C_{IN2}	$V_{IN} = 0$	TBD	TBD	pF
$\overline{WP}$/ACC Pin Capacitance	C_{IN3}	$V_{IN} = 0$	TBD	TBD	pF

Note : Test conditions Ta = 25 °C, f = 1.0 MHz

■ TIMING DIAGRAM

- **Key to Switching Waveforms**

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Change from H to L
	May Change from L to H	Will Change from L to H
	"H" or "L": Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

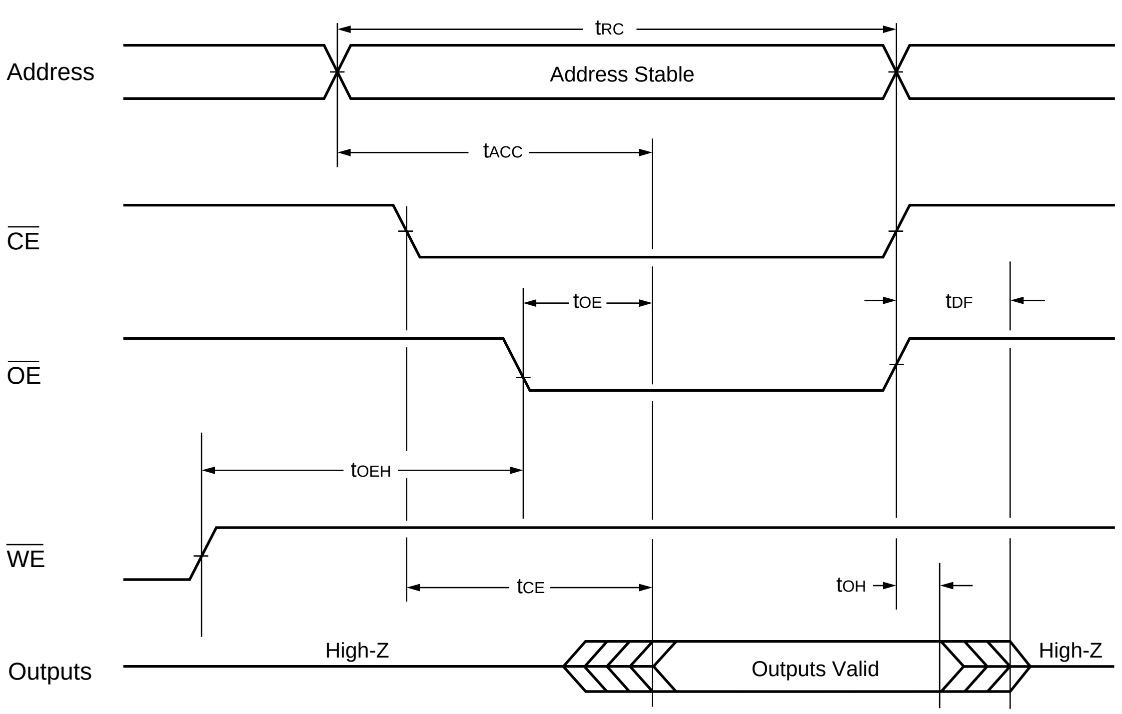


Figure 5.1 Read Operation Timing Diagram

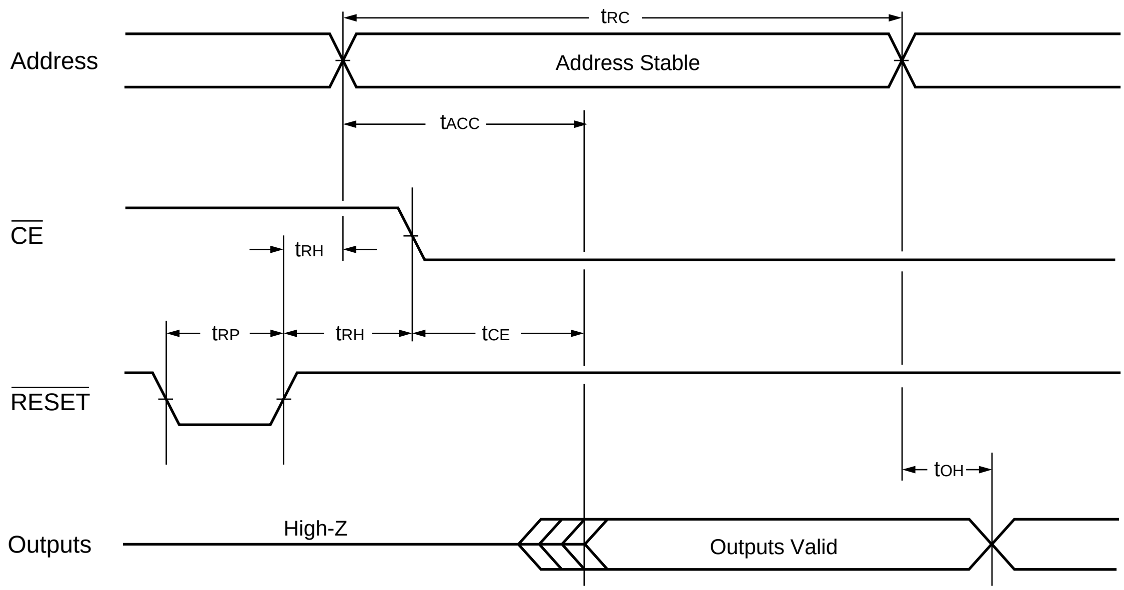
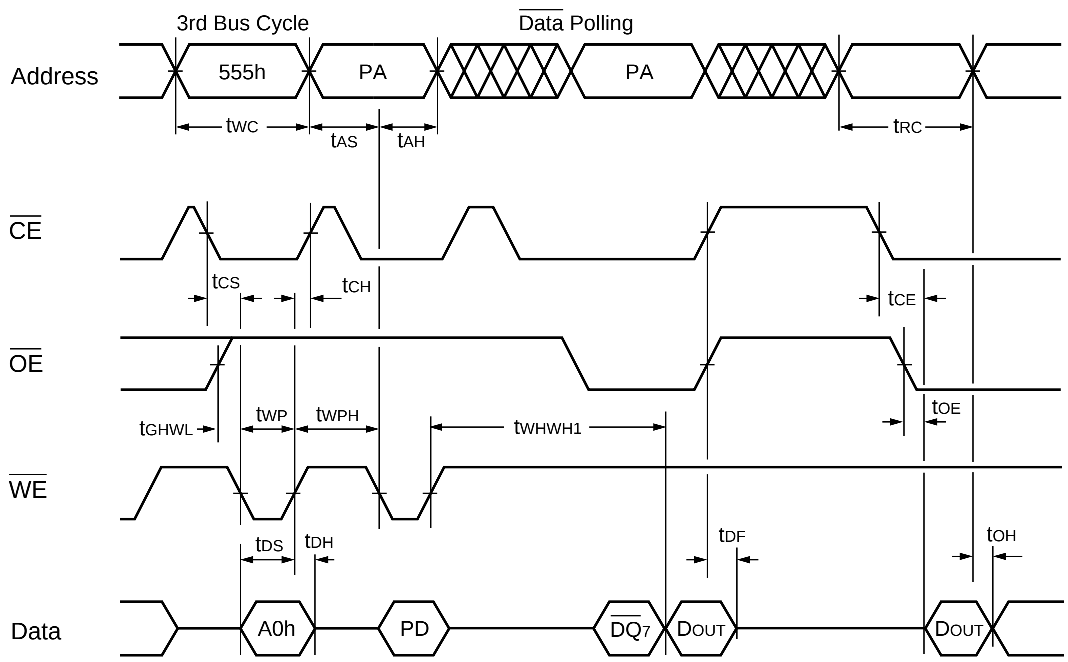
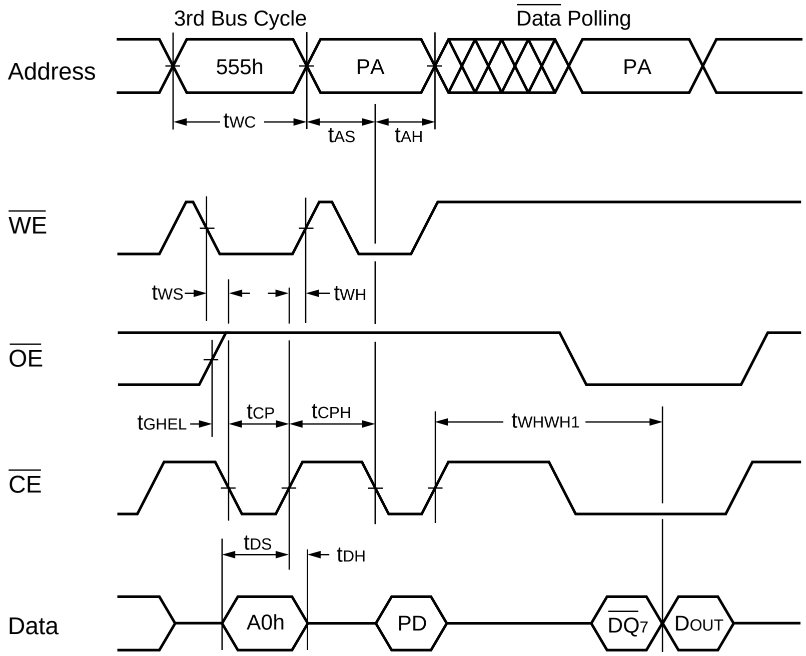


Figure 5.2 Hardware Reset/Read Operation Timing Diagram



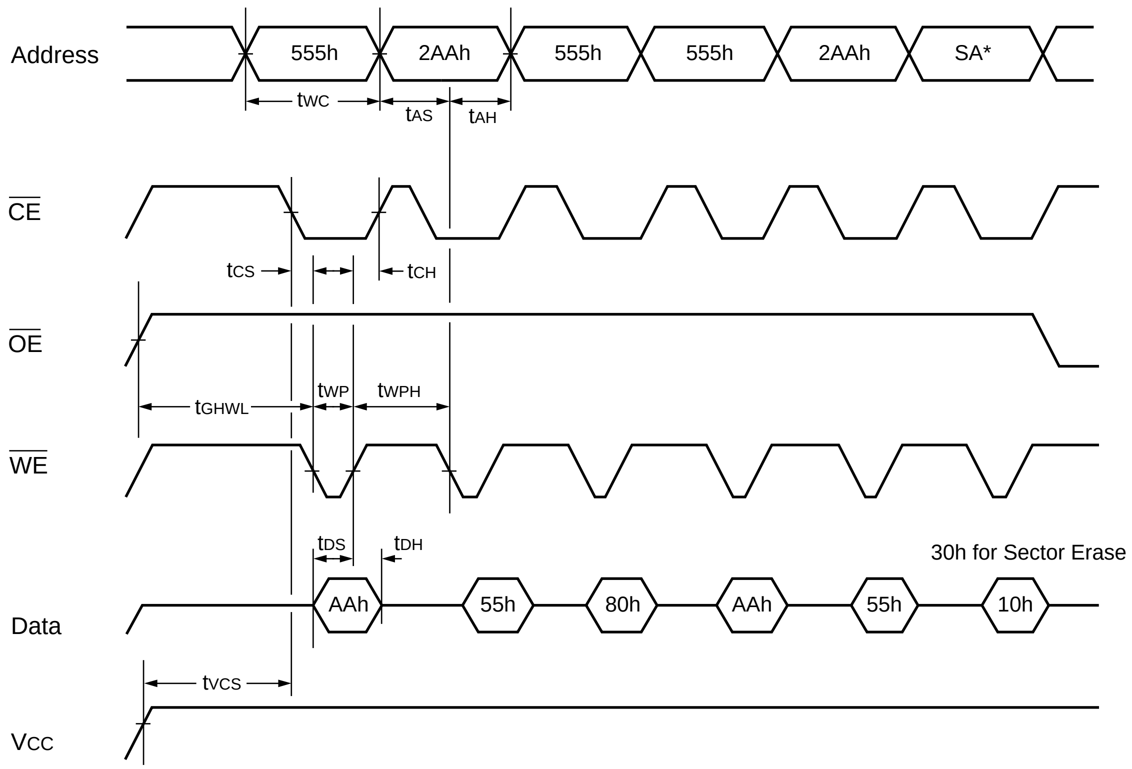
Notes : 1. PA is address of the memory location to be programmed.
2. PD is data to be programmed at word address.
3. $\overline{DQ_7}$ is the output of the complement of the data written to the device.
4. D_{OUT} is the output of the data written to the device.
5. Figure indicates last two bus cycles out of four bus cycle sequence.
6. These waveforms are for the × 16 mode. (The addresses differ from × 8 mode.)

Figure 6 Alternate $\overline{\text{WE}}$ Controlled Program Operation Timing Diagram



Notes : 1. PA is address of the memory location to be programmed.
2. PD is data to be programmed at word address.
3. $\overline{DQ_7}$ is the output of the complement of the data written to the device.
4. D_{OUT} is the output of the data written to the device.
5. Figure indicates last two bus cycles out of four bus cycle sequence.
6. These waveforms are for the × 16 mode. (The addresses differ from × 8 mode.)

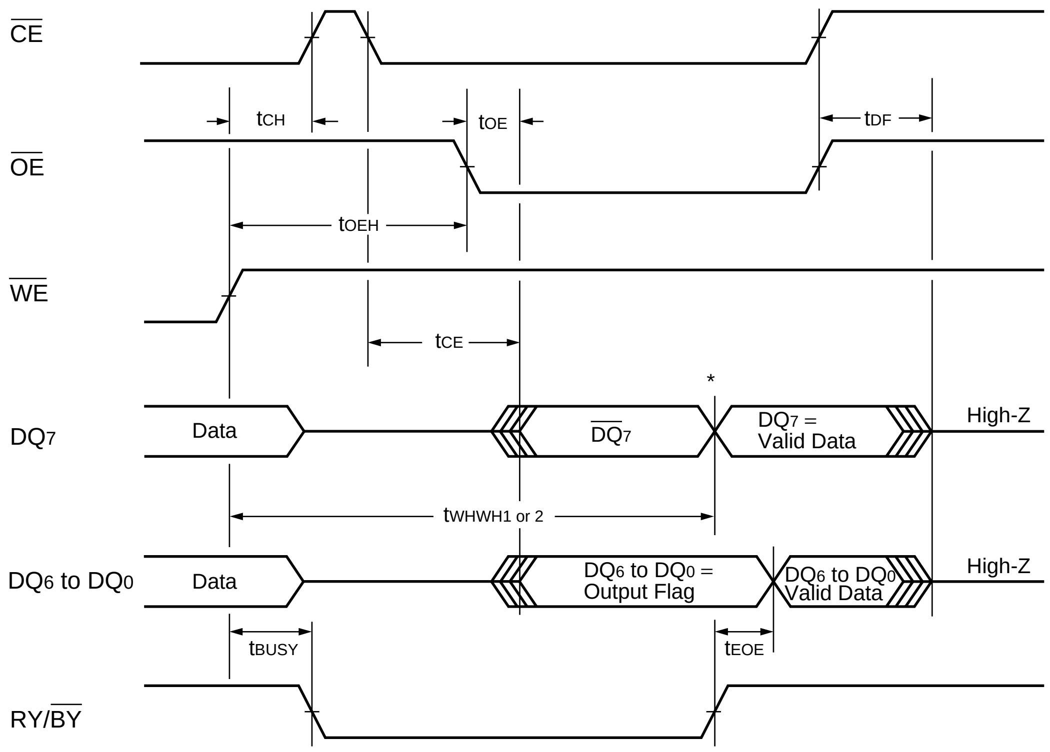
Figure 7 Alternate $\overline{CE}$ Controlled Program Operation Timing Diagram



* : SA is the sector address for Sector Erase. Addresses = 555h (Word) for Chip Erase.

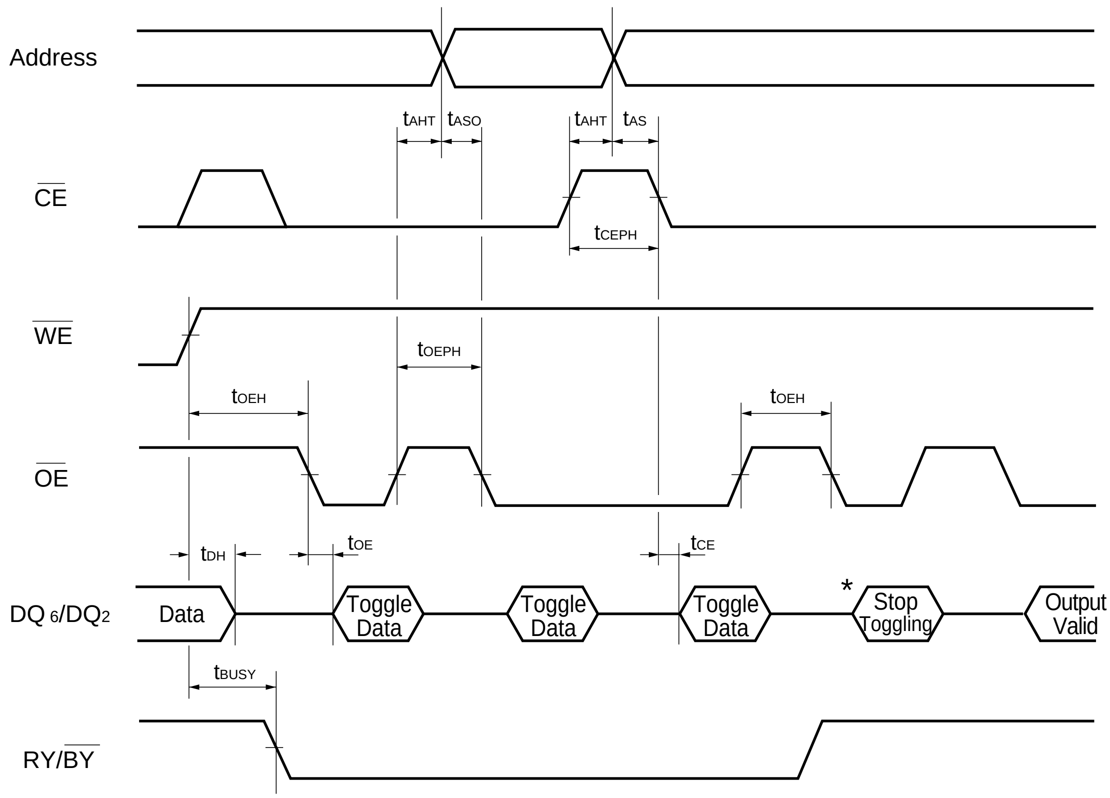
Note : These waveforms are for the × 16 mode. (The addresses differ from × 8 mode.)

Figure 8　Chip/Sector Erase Operation Timing Diagram



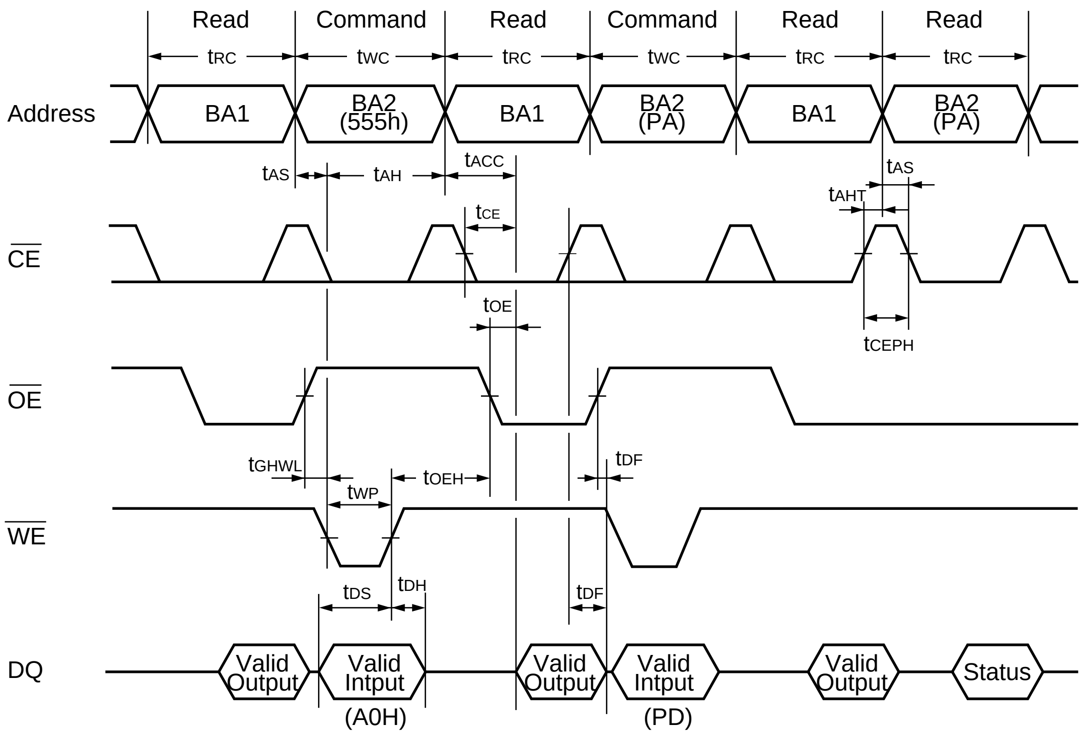
* : DQ_7 = Valid Data (The device has completed the Embedded operation) .

Figure 9 Data Polling during Embedded Algorithm Operation Timing Diagram



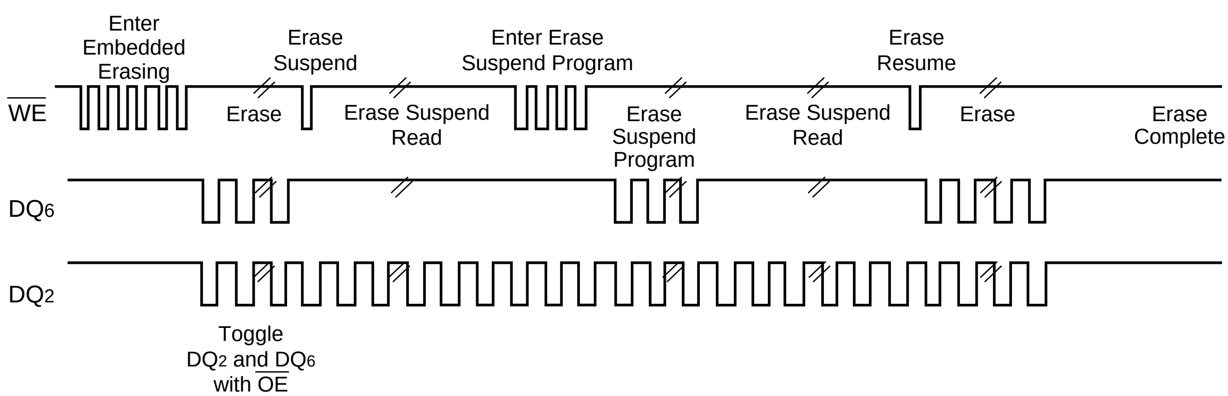
* : DQ6 stops toggling (The device has completed the Embedded operation).

Figure 10 AC Waveforms for Toggle Bit I during Embedded Algorithm Operations



Note : This is example of Read for Bank 1 and Embedded Algorithm (program) for Bank 2.
BA1 : Address corresponding to Bank 1
BA2 : Address corresponding to Bank 2

Figure 11 Bank-to-Bank Read/Write Timing Diagram



Note : DQ_2 is read from the erase-suspended sector.

Figure 12 DQ_2 vs. DQ_6

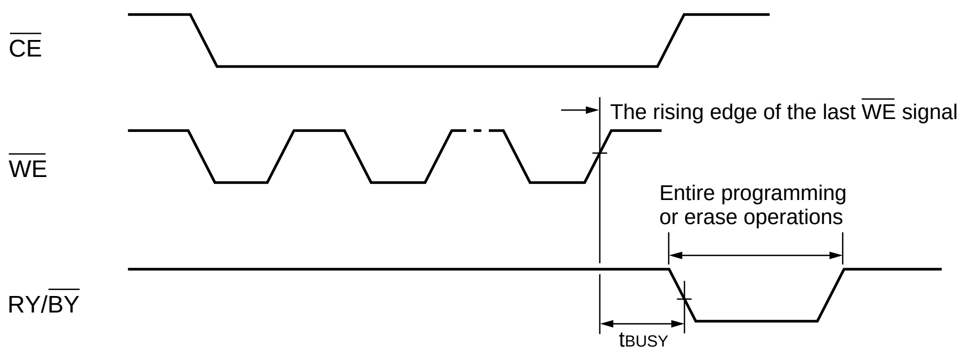


Figure 13 RY/BY Timing Diagram during Program/Erase Operation Timing Diagram

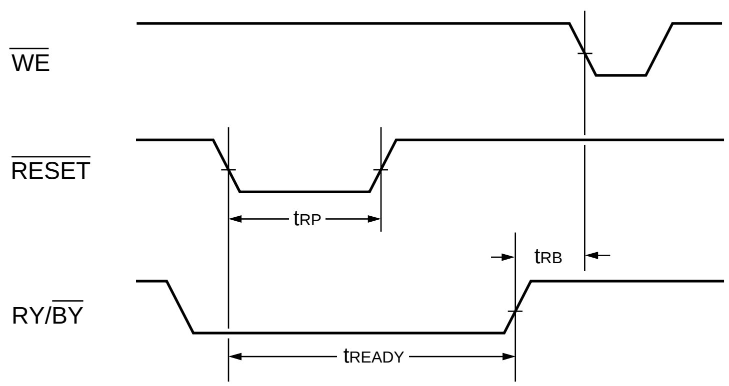


Figure 14 RESET, RY/BY Timing Diagram

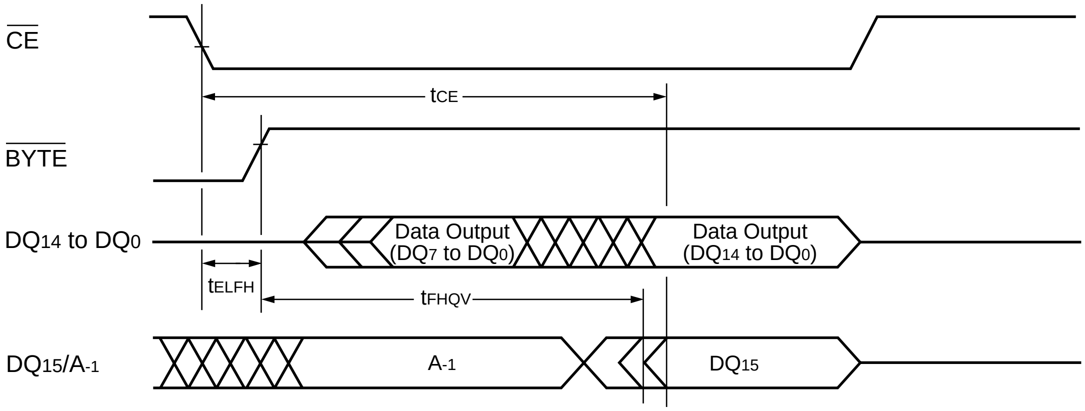


Figure 15 Timing Diagram for Word Mode Configuration

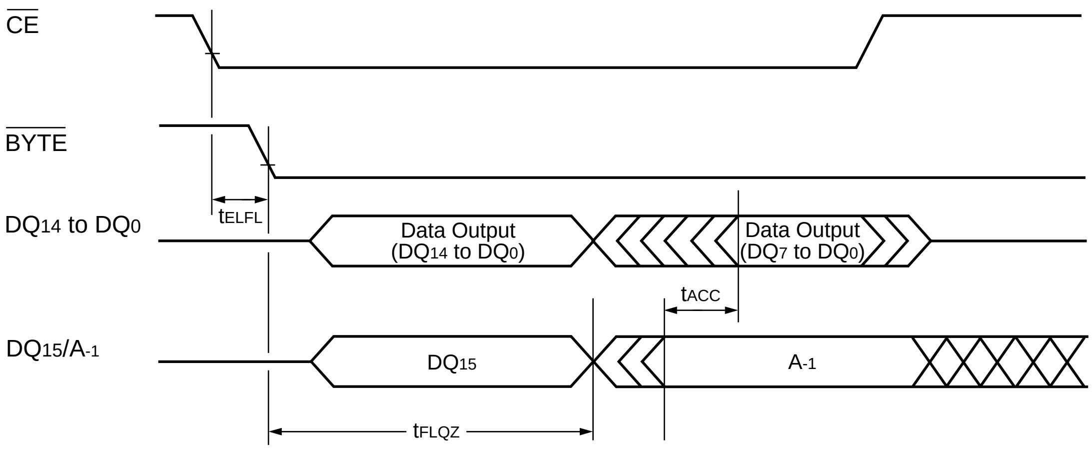


Figure 16 Timing Diagram for Byte Mode Configuration

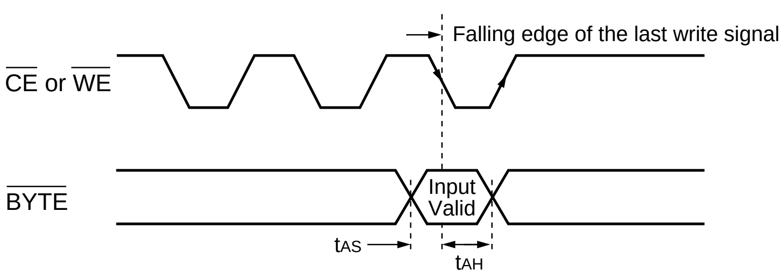
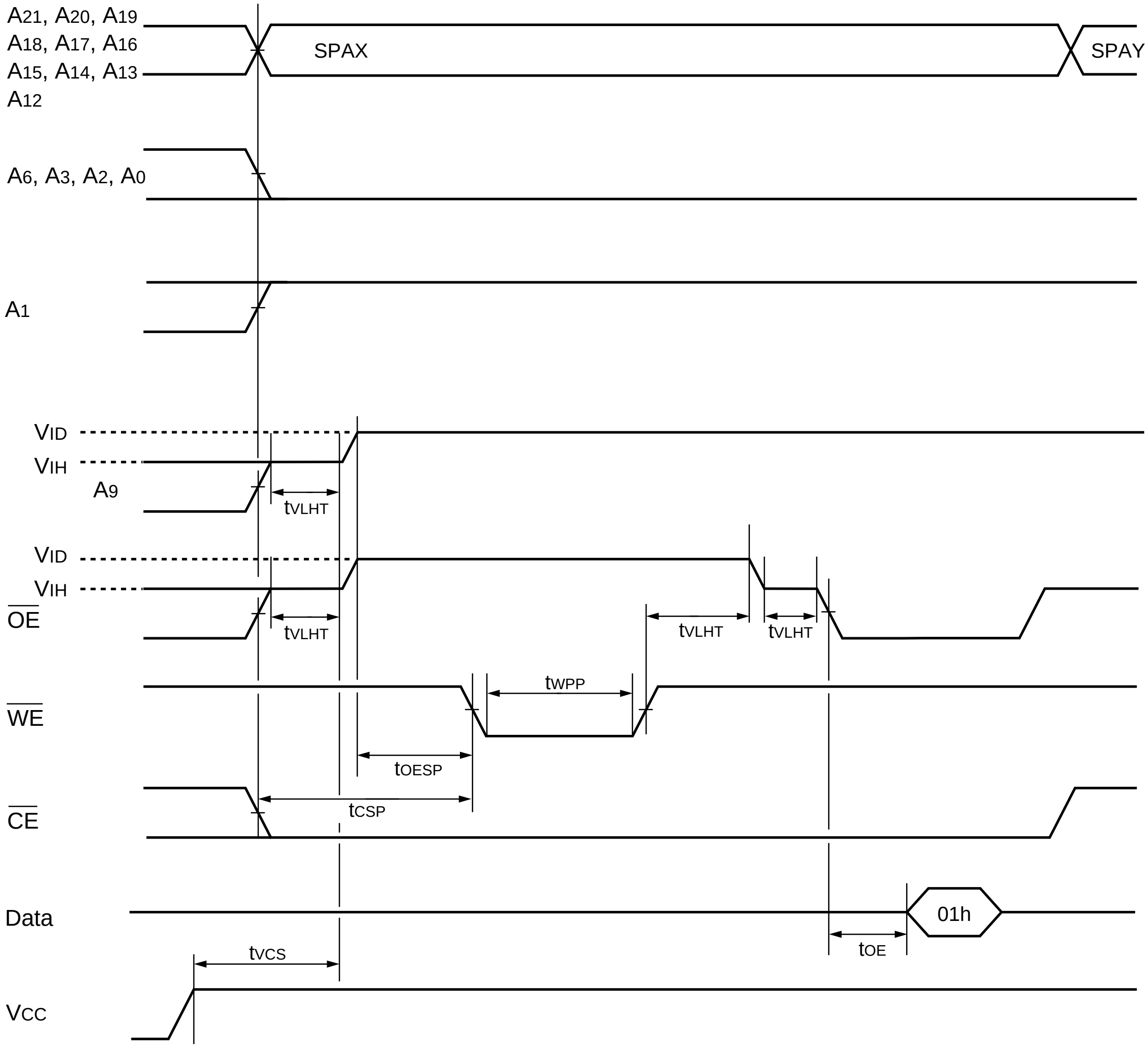


Figure 17 BYTE Timing Diagram for Write Operations



SPAX : Sector Group Address to be protected
SPAY : Next Sector Group Address to be protected

Note : A_{-1} is V_{IL} on byte mode.

Figure 18 Sector Group Protection Timing Diagram

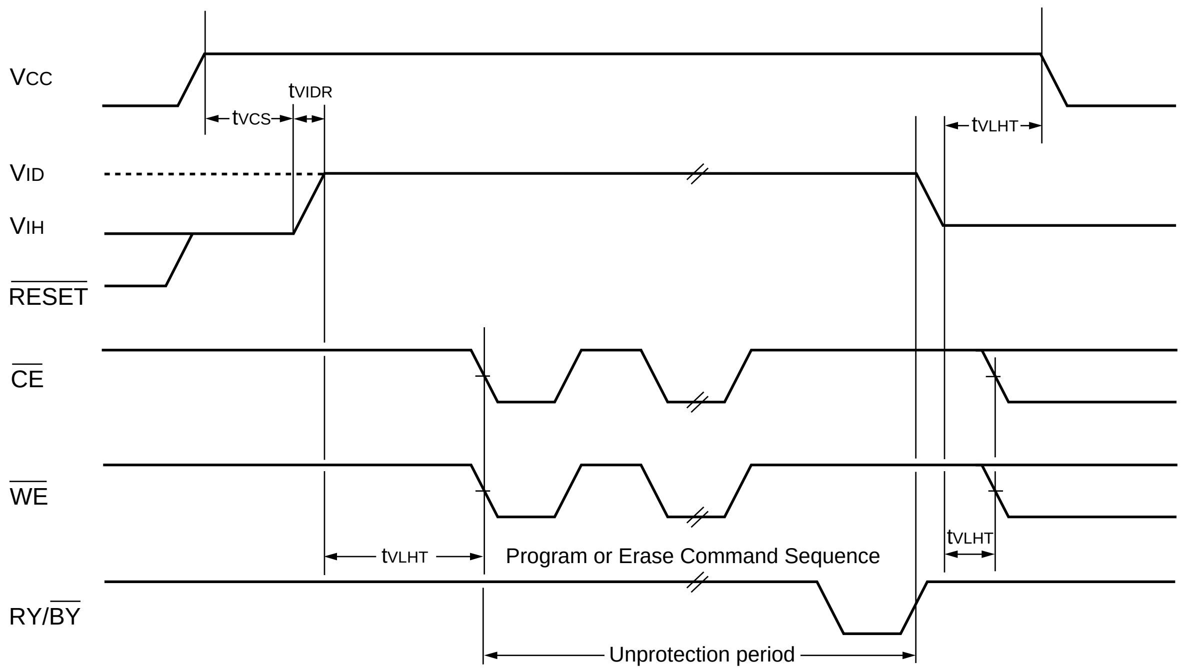
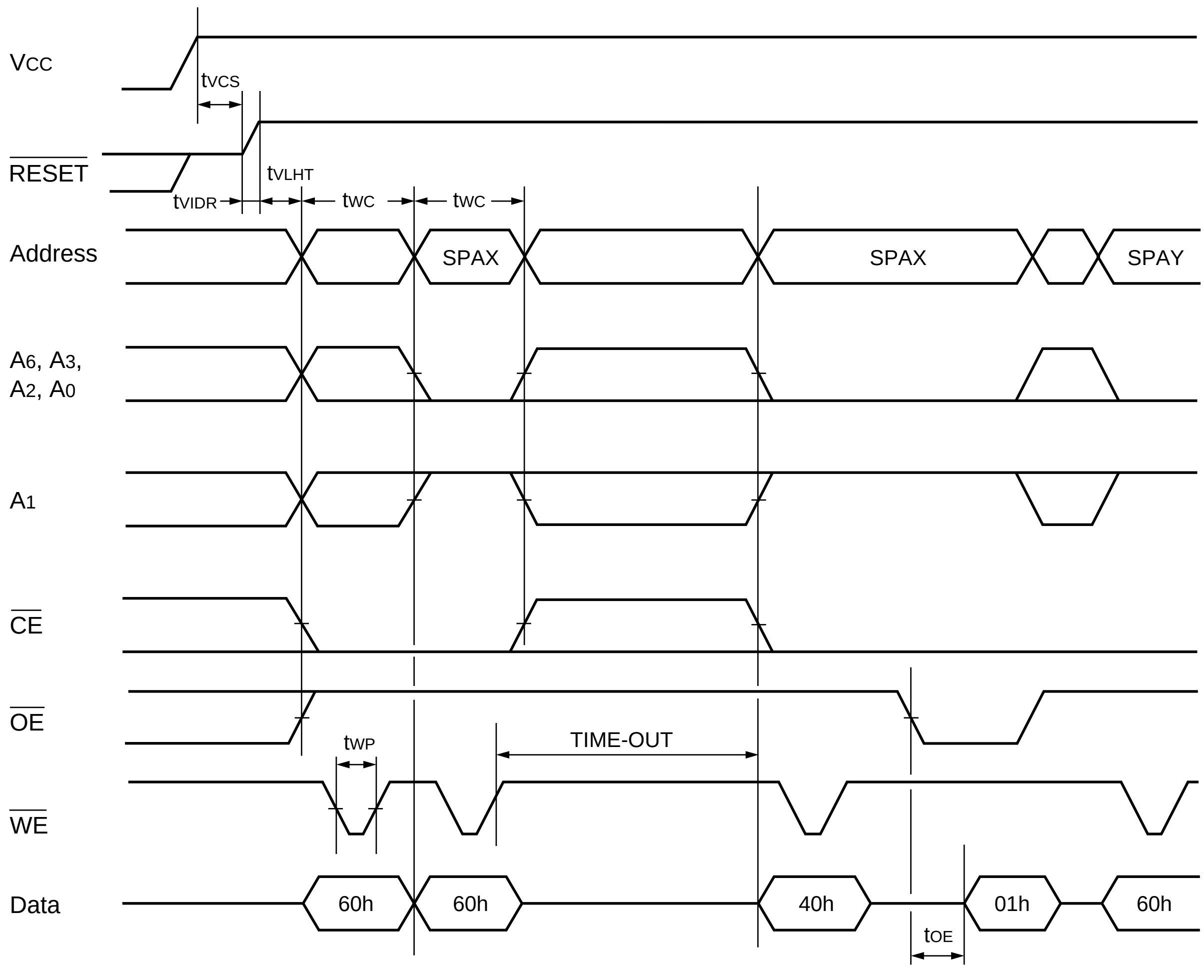


Figure 19 Temporary Sector Group Unprotection Timing Diagram



SPAX : Sector Group Address to be protected
SPAY : Next Sector Group Address to be protected
TIME-OUT : Time-Out window = 250 µs (Min.)

Figure 20 Extended Sector Group Protection Timing Diagram

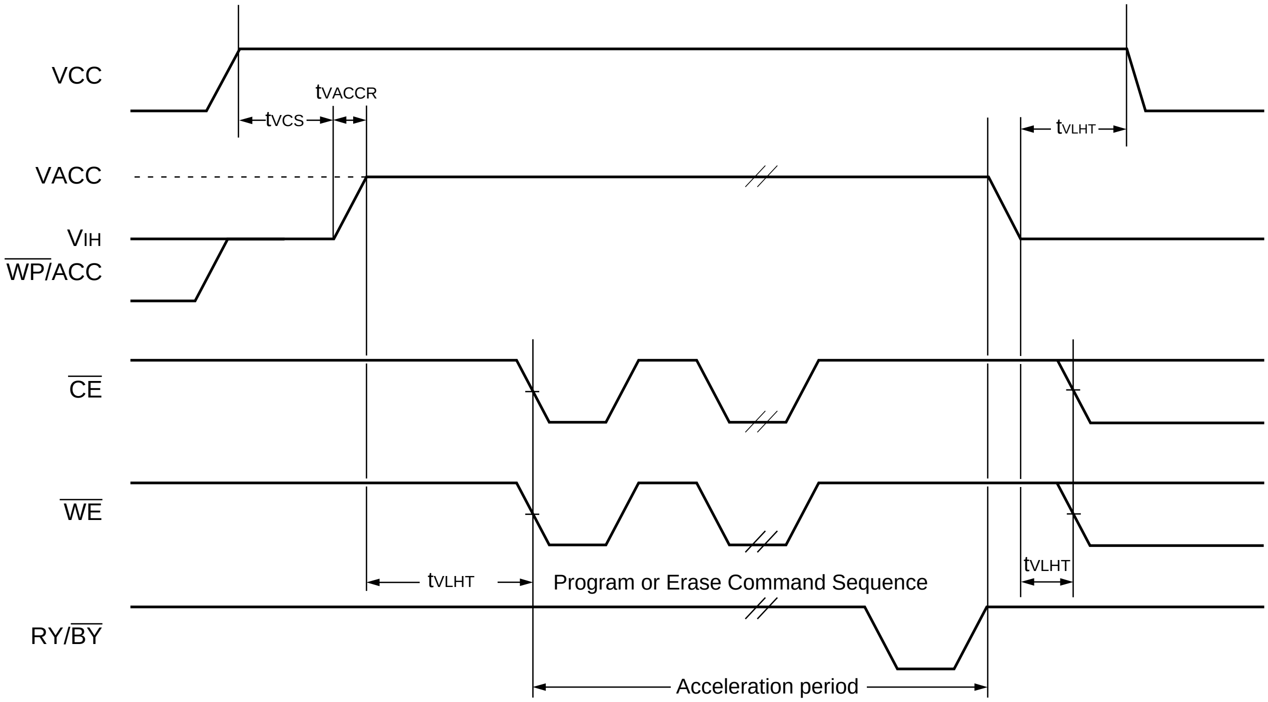
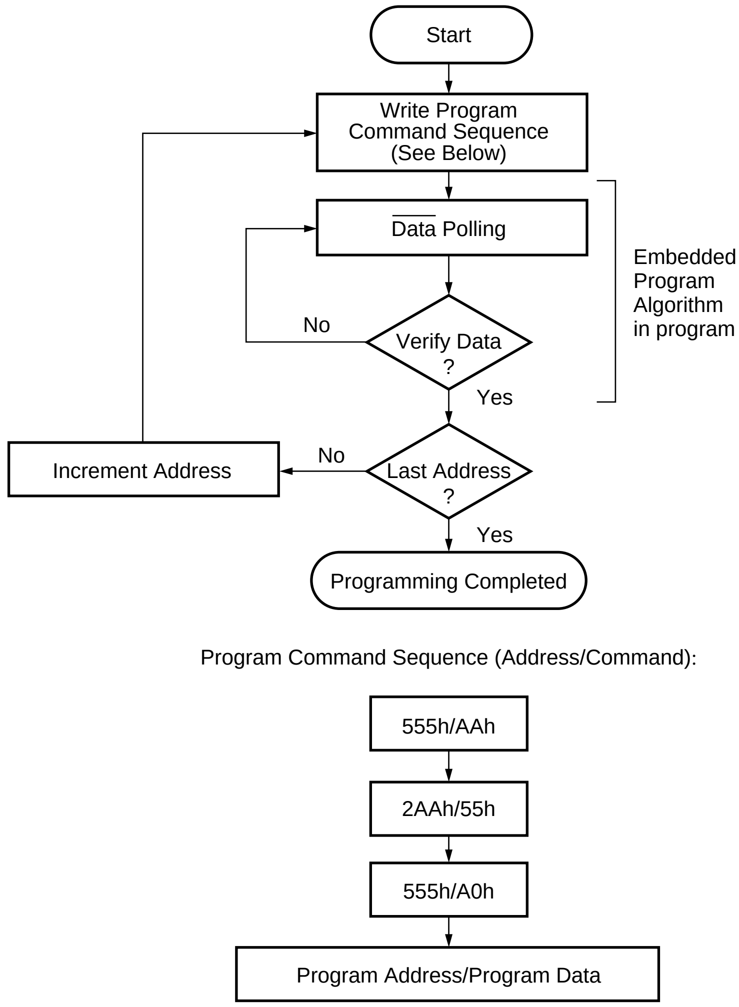


Figure 21 Accelerated Program Timing Diagram

■ FLOW CHART

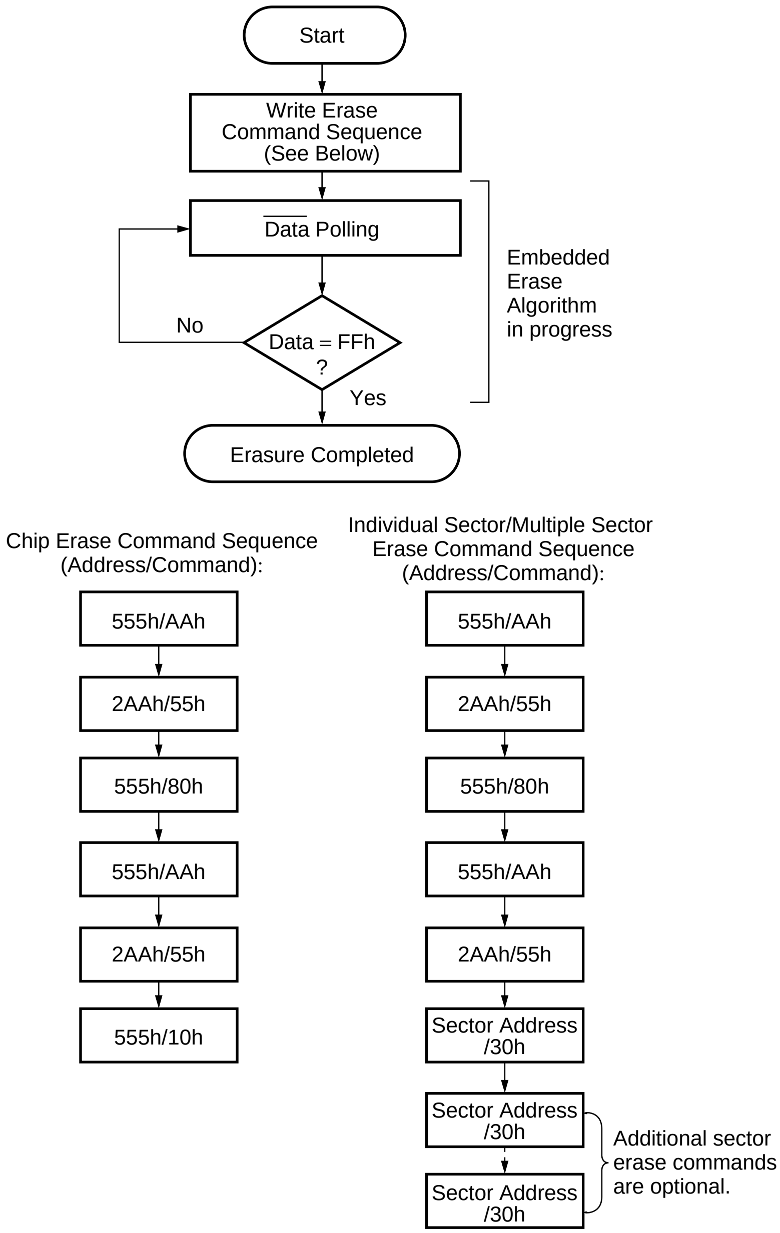
EMBEDDED ALGORITHM



Note : The sequence is applied for × 16 mode.
The addresses differ from × 8 mode.

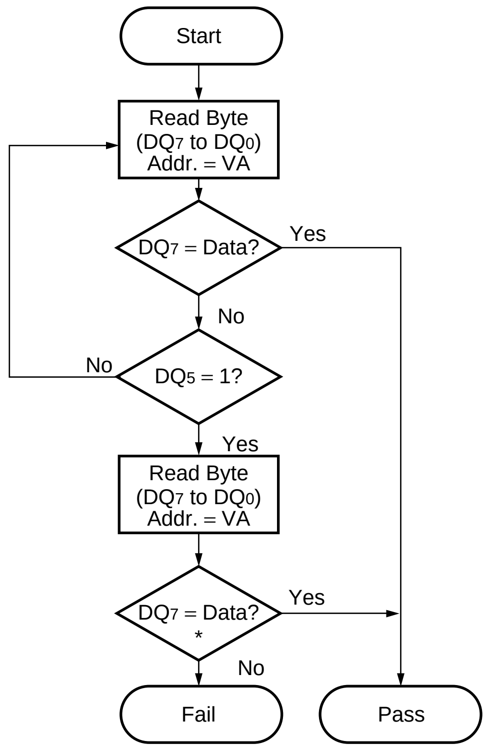
Figure 22 Embedded Program™ Algorithm

EMBEDDED ALGORITHM



Note : The sequence is applied for × 16 mode.
The addresses differ from × 8 mode.

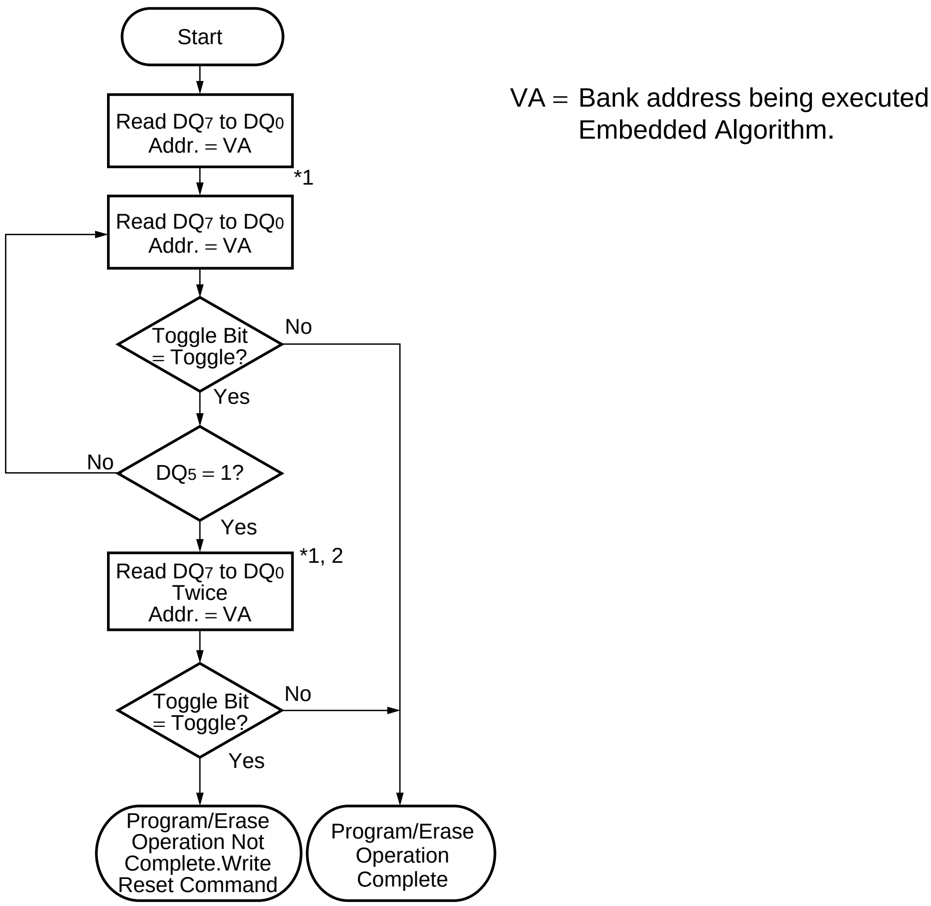
Figure 23 Embedded Erase™ Algorithm



VA = Address for programming
= Any of the sector addresses within the sector being erased during sector erase or multiple erases operation.
= Any of the sector addresses within the sector not being protected during sector erase or multiple sector erases operation.

* : DQ_7 is rechecked even if DQ_5 = “1” because DQ_7 may change simultaneously with DQ_5.

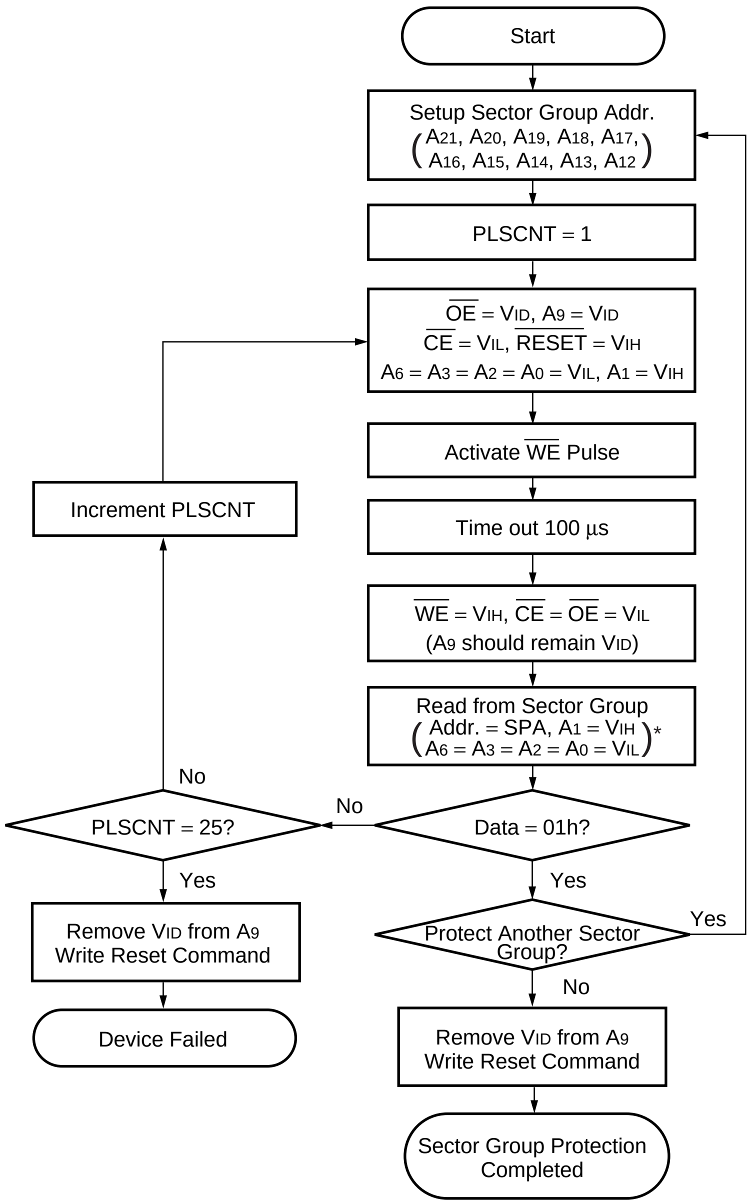
Figure 24 $\overline{\text{Data}}$ Polling Algorithm



*1 : Read toggle bit twice to determine whether or not it is toggling.

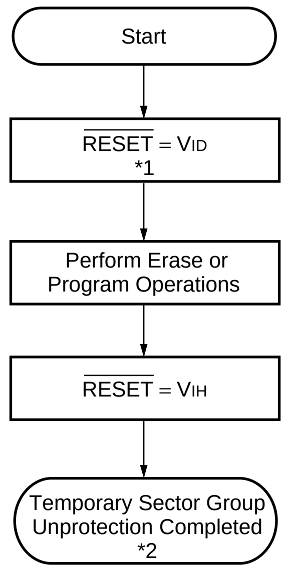
*2 : Recheck toggle bit because it may stop toggling as DQ_5 changes to “1”.

Figure 25　Toggle Bit Algorithm



* : A_{-1} is V_{IL} in byte mode.

Figure 26 Sector Group Protection Algorithm



*1 : All protected sectors are unprotected.

*2 : All previously protected sectors are reprotected.

Figure 27   Temporary Sector Group Unprotection Algorithm

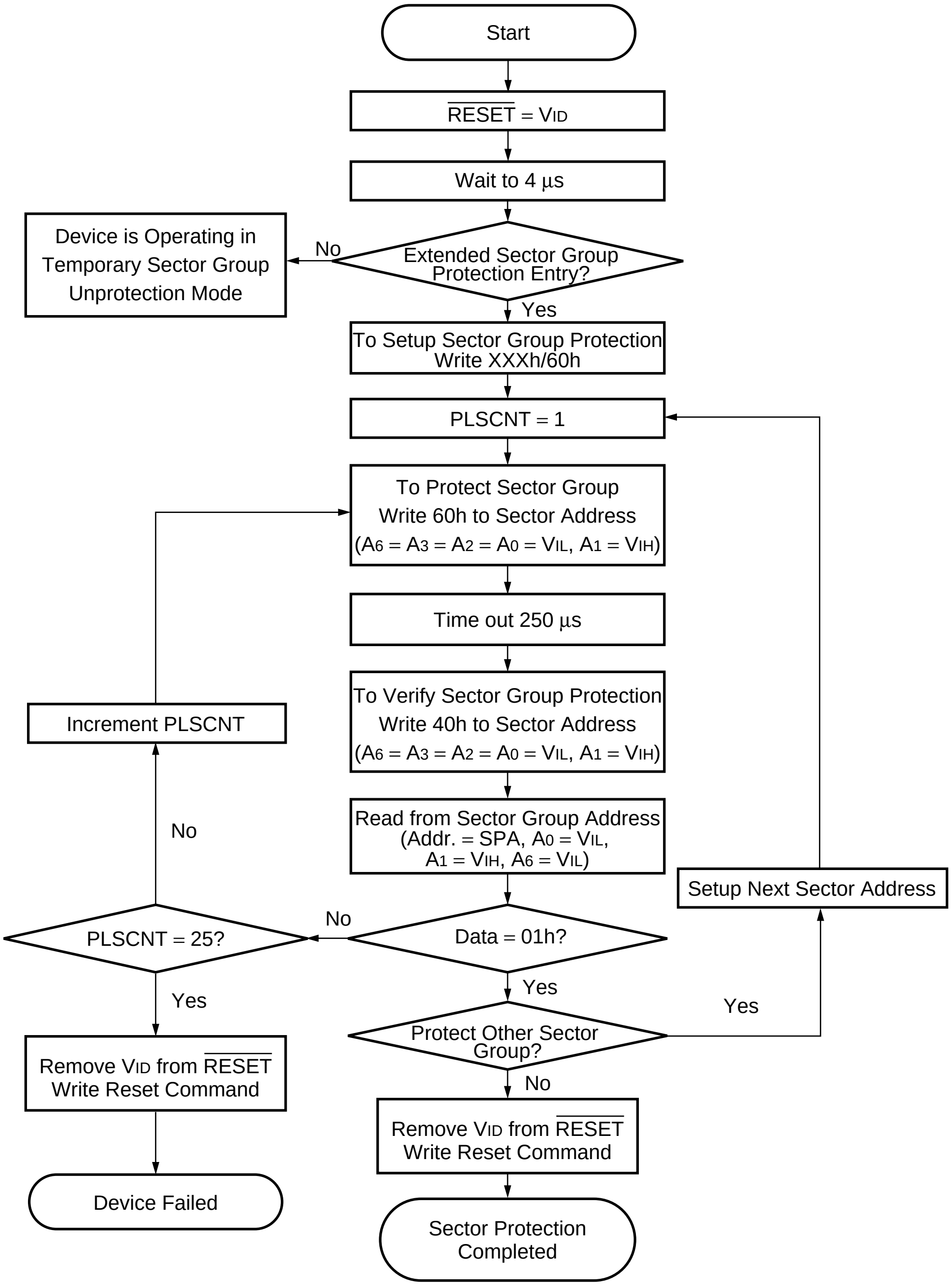
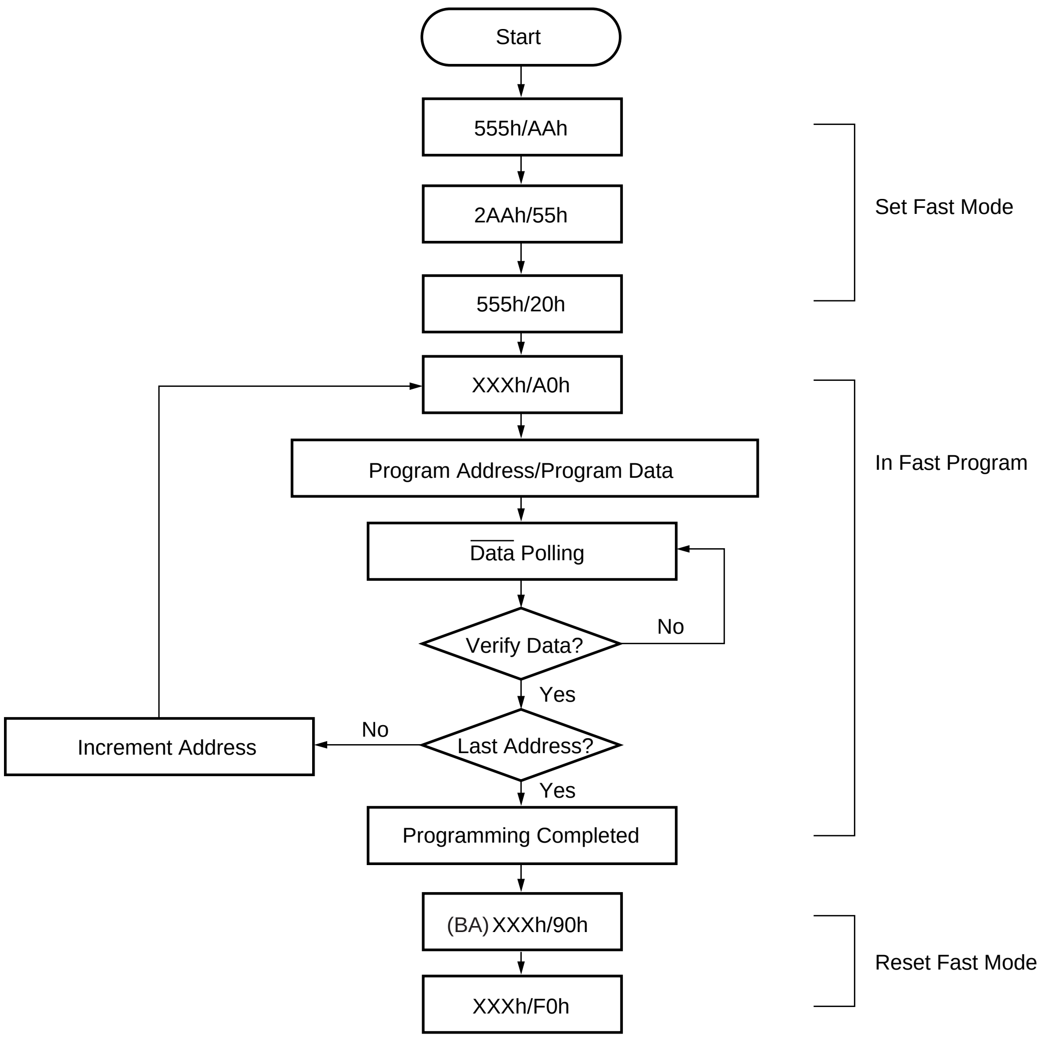


Figure 28 Extended Sector Group Protection Algorithm

FAST MODE ALGORITHM



Notes: • The sequence is applied for × 16 mode.
• The addresses differ from × 8 mode.

Figure 29 Embedded Programming Algorithm for Fast Mode

■ ORDERING INFORMATION

Standard Products

Fujitsu standard products are available in several packages. The order number is formed by a combination of :

MBM29DL640 E 80 TN

PACKAGE TYPE
TN = 48-Pin Thin Small Outline Package (TSOP) Standard Pinout
TR = 48-Pin Thin Small Outline Package (TSOP) Reverse Pinout
PBT = 63-Ball Fine pitch Ball Grid Array Package (FBGA)

SPEED OPTION
See Product Selector Guide

DEVICE REVISION

DEVICE NUMBER/DESCRIPTION
MBM29DL640
64 Mega-bit (8 M × 8-Bit or 4 M × 16-Bit) CMOS Flash Memory
3.0 V-only Read, Program, and Erase

Valid Combinations		
MBM29DL640E	80 90 12	TN TR PBT

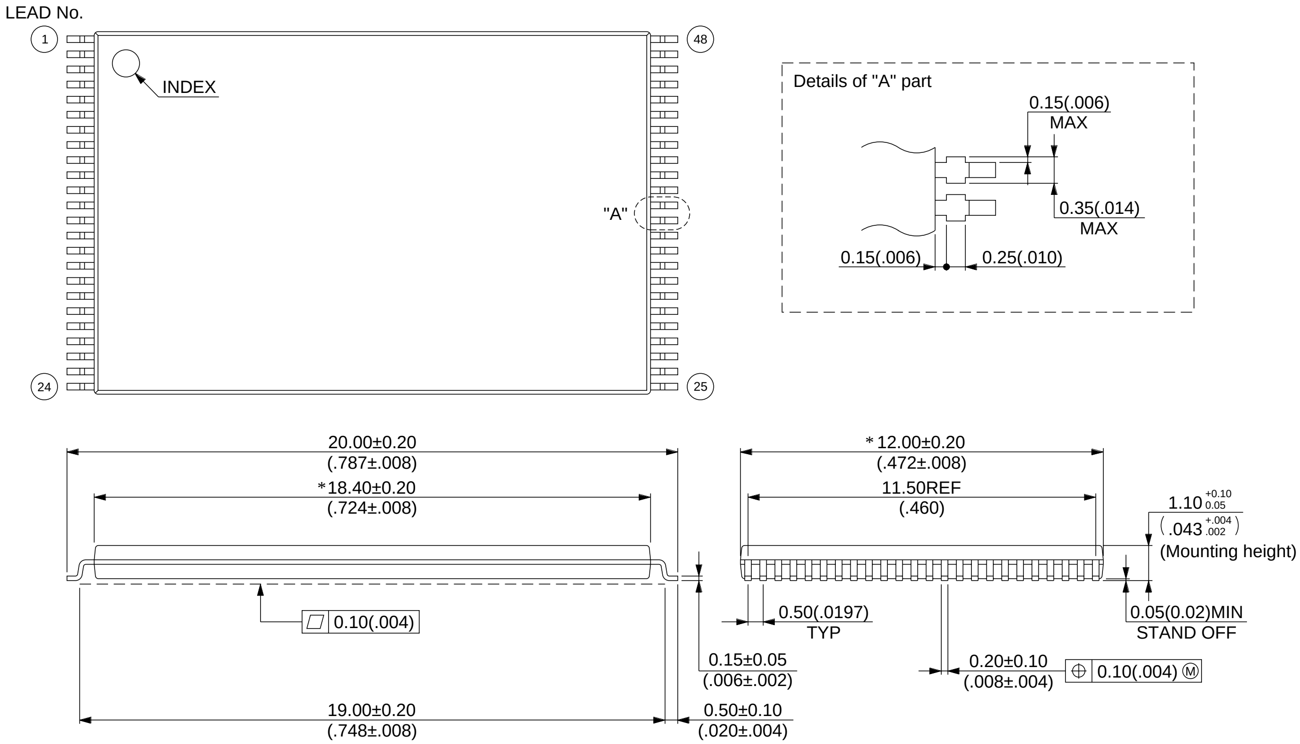
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local Fujitsu sales office to confirm availability of specific valid combinations and to check on newly released combinations.

■ PACKAGE DIMENSIONS

48-pin plastic TSOP (I)
(FPT-48P-M19)

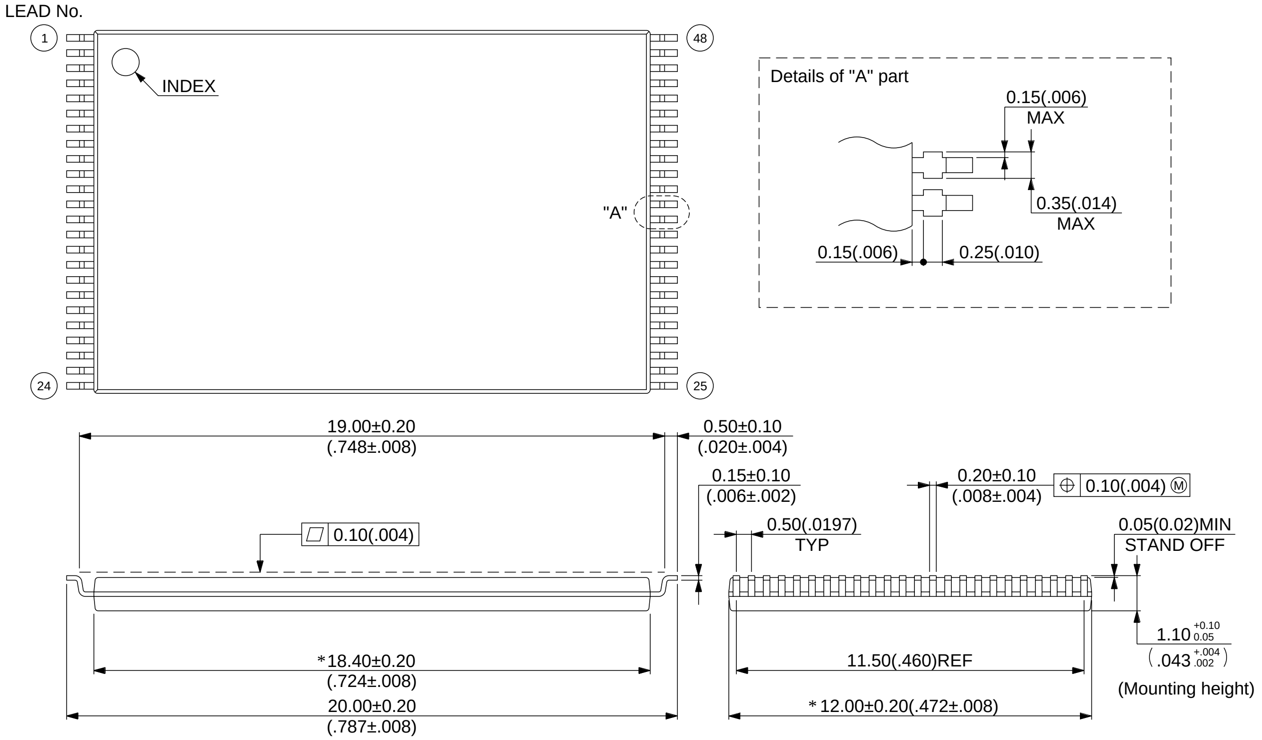
* Resin Protrusion. (Each Side : 0.15 (.006) Max)



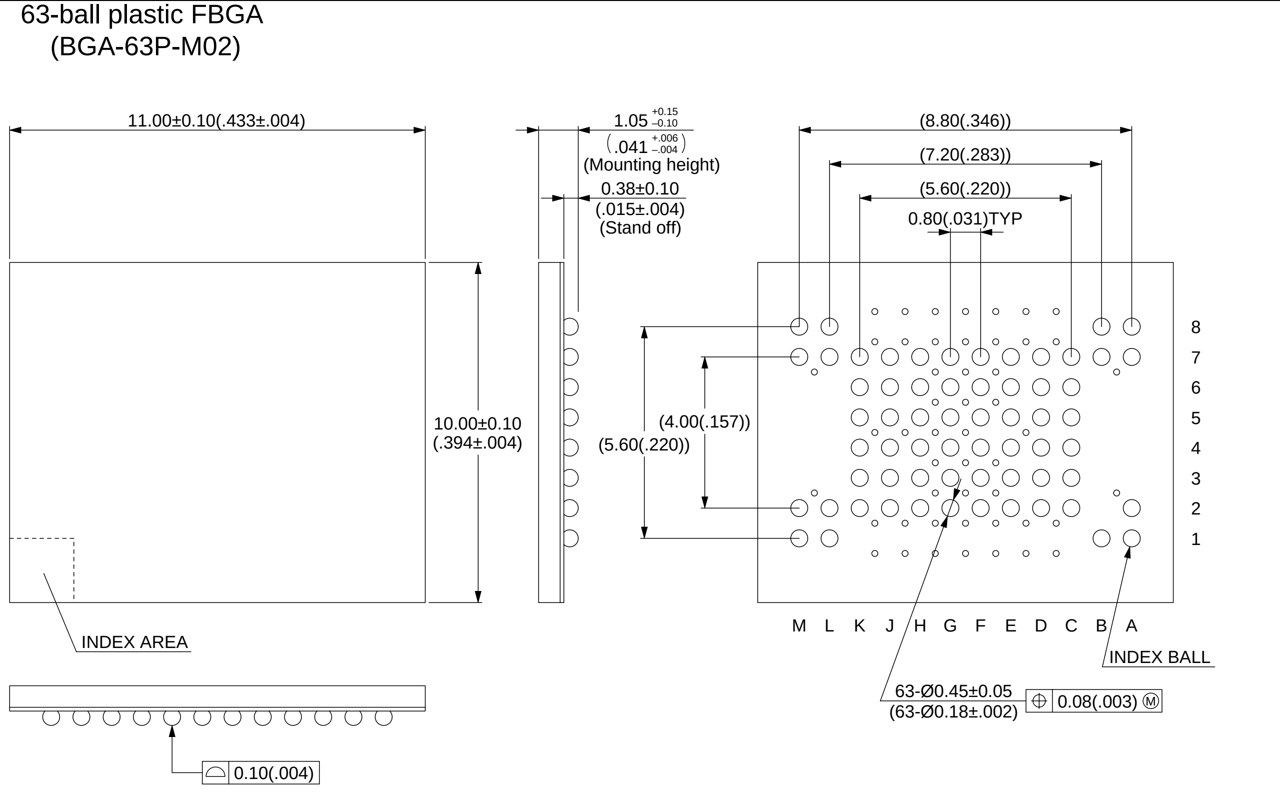
Dimensions in mm (inches)

48-pin plastic TSOP (I)
(FPT-48P-M20)

* Resin Protrusion. (Each Side : 0.15 (.006) Max)



Dimensions in mm (inches)



Dimensions in mm (inches)

FUJITSU LIMITED

For further information please contact:

Japan

FUJITSU LIMITED
Corporate Global Business Support Division
Electronic Devices
Shinjuku Dai-Ichi Seimei Bldg. 7-1,
Nishishinjuku 2-chome, Shinjuku-ku,
Tokyo 163-0721, Japan
Tel: +81-3-5322-3347
Fax: +81-3-5322-3386

http://edevice.fujitsu.com/

North and South America

FUJITSU MICROELECTRONICS, INC.
3545 North First Street,
San Jose, CA 95134-1804, U.S.A.
Tel: +1-408-922-9000
Fax: +1-408-922-9179

Customer Response Center
Mon. - Fri.: 7 am - 5 pm (PST)
Tel: +1-800-866-8608
Fax: +1-408-922-9179

http://www.fujitsumicro.com/

Europe

FUJITSU MICROELECTRONICS EUROPE GmbH
Am Siebenstein 6-10,
D-63303 Dreieich-Buchschlag,
Germany
Tel: +49-6103-690-0
Fax: +49-6103-690-122

http://www.fujitsu-fme.com/

Asia Pacific

FUJITSU MICROELECTRONICS ASIA PTE. LTD.
#05-08, 151 Lorong Chuan,
New Tech Park,
Singapore 556741
Tel: +65-281-0770
Fax: +65-281-0220

http://www.fmap.com.sg/

Korea

FUJITSU MICROELECTRONICS KOREA LTD.
1702 KOSMO TOWER, 1002 Daechi-Dong,
Kangnam-Gu,Seoul 135-280
Korea
Tel: +82-2-3484-7100
Fax: +82-2-3484-7111

F0101

The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.

The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.

The contents of this document may not be reproduced or copied without the permission of FUJITSU LIMITED.

FUJITSU semiconductor devices are intended for use in standard applications (computers, office automation and other office equipments, industrial, communications, and measurement equipments, personal or household devices, etc.).
CAUTION:
Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with FUJITSU sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Control Law of Japan, the prior authorization by Japanese government should be required for export of those products from Japan.