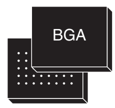
M58LT128HST
M58LT128HSB

128 Mbit (8 Mb ×16, multiple bank, multilevel interface, burst) 1.8 V supply, secure Flash memories

Features

- Supply voltage
  - V_{DD} = 1.7 V to 2.0 V for Program, Erase and Read
  - V_{DDQ} = 2.7 V to 3.6 V for I/O buffers
  - V_{PP} = 9 V for fast program
- Synchronous/Asynchronous Read
  - Synchronous Burst Read mode: 52 MHz
  - Asynchronous Page Read mode
  - Random access: 85 ns
- Synchronous Burst Read Suspend
- Programming time
  - 2.5 µs typical word program time using Buffer Enhanced Factory Program command
- Memory organization
  - Multiple bank memory array: 8-Mbit banks
  - Parameter blocks (top or bottom location)
- Dual operations
  - program/erase in one bank while read in others
  - No delay between Read and Write operations
- Block protection
  - All blocks protected at power-up
  - Any combination of blocks can be protected with zero latency
  - Absolute write protection with $V_{PP} = V_{SS}$
- Security
  - Software security features
  - 64-bit unique device number
  - 2112-bit user programmable OTP Cells
- Common flash interface (CFI)
- 100 000 program/erase cycles per block



TBGA64 (ZA)
10 × 13 mm

- Electronic signature
  - Manufacturer code: 20h
  - Top device codes: M58LT128HST: 88D6h
  - Bottom device codes M58LT128HSB: 88D7h
- TBGA64 package
  - ECOPACK® available

Contents

1 **Description 7**

2 **Signal descriptions 12**

2.1 Address inputs (A0-A22) 12

2.2 Data inputs/outputs (DQ0-DQ15) 12

2.3 Chip Enable ($\overline{E}$) 12

2.4 Output Enable ($\overline{G}$) 12

2.5 Write Enable ($\overline{W}$) 12

2.6 Reset ($\overline{RP}$) 12

2.7 Latch Enable ($\overline{L}$) 13

2.8 Clock (K) 13

2.9 Wait (WAIT) 13

2.10 V_{DD} supply voltage 13

2.11 V_{DDQ} supply voltage 13

2.12 V_{PP} program supply voltage 13

2.13 V_{SS} ground 14

2.14 V_{SSQ} ground 14

3 **Bus operations 15**

3.1 Bus Read 15

3.2 Bus Write 15

3.3 Address Latch 15

3.4 Output Disable 15

3.5 Standby 16

3.6 Reset 16

4 **Command interface 17**

4.1 Read Array command 18

4.2 Read Status Register command 18

4.3 Read Electronic Signature command 19

4.4 Read CFI Query command 19

4.5 Clear Status Register command 20

4.6 Block Erase command 20
4.7 Blank Check command 21
4.8 Program command 22
4.9 Buffer Program command 23
4.10 Buffer Enhanced Factory Program command 24
4.10.1 Setup phase 24
4.10.2 Program and verify phase 25
4.10.3 Exit phase 25
4.11 Program/Erase Suspend command 26
4.12 Program/Erase Resume command 27
4.13 Protection Register Program command 27
4.14 Set Configuration Register command 28
4.15 Block Protect command 28
4.16 Block Unprotect command 28

5 Status Register 33
5.1 Program/Erase Controller status bit (SR7) 33
5.2 Erase Suspend status bit (SR6) 33
5.3 Erase/Blank Check status bit (SR5) 34
5.4 Program status bit (SR4) 34
5.5 V_{PP} status bit (SR3) 34
5.6 Program Suspend status bit (SR2) 35
5.7 Block Protection status bit (SR1) 35
5.8 Bank Write/Multiple Word Program status bit (SR0) 35

6 Configuration Register 37
6.1 Read Select bit (CR15) 37
6.2 X-Latency bits (CR13-CR11) 37
6.3 Wait Polarity bit (CR10) 38
6.4 Data Output Configuration bit (CR9) 38
6.5 Wait Configuration bit (CR8) 38
6.6 Burst Type bit (CR7) 38
6.7 Valid Clock Edge bit (CR6) 39
6.8 Wrap Burst bit (CR3) 39

6.9 Burst length bits (CR2-CR0) 39

7 Read modes 44

7.1 Asynchronous Read mode 44

7.2 Synchronous Burst Read mode 45

7.2.1 Synchronous Burst Read Suspend 46

7.3 Single Synchronous Read mode 46

8 Dual operations and multiple bank architecture 47

9 Block protection 49

9.1 Protection status 49

9.2 Protected state 49

9.3 Unprotected state 49

9.4 Protection operations during Erase Suspend 50

10 Program and erase times and endurance cycles 51

11 Maximum rating 53

12 DC and AC parameters 54

13 Package mechanical 70

14 Part numbering 72

Appendix A Block address tables 73

Appendix B Common Flash Interface 81

Appendix C Flowcharts and pseudo codes 91

Appendix D Command Interface state tables 100

Revision history 109

List of tables

Table 1. Signal names . . . 9
Table 2. Bank architecture . . . 11
Table 3. Bus operations . . . 16
Table 4. Command codes . . . 17
Table 5. Standard commands . . . 29
Table 6. Factory commands . . . 30
Table 7. Electronic signature codes . . . 30
Table 8. Protection Register locks . . . 32
Table 9. Status Register bits . . . 36
Table 10. X-Latency Settings . . . 37
Table 11. Configuration Register . . . 40
Table 12. Burst type definition . . . 41
Table 13. Dual operations allowed in other banks . . . 47
Table 14. Dual operations allowed in same bank . . . 48
Table 15. Dual operation limitations . . . 48
Table 16. Program/erase times and endurance cycles . . . 51
Table 17. Absolute maximum ratings . . . 53
Table 18. Operating and AC measurement conditions . . . 54
Table 19. Capacitance . . . 55
Table 20. DC characteristics - currents . . . 56
Table 21. DC characteristics - voltages . . . 57
Table 22. Asynchronous Read AC characteristics . . . 60
Table 23. Synchronous Read AC characteristics . . . 64
Table 24. Write AC characteristics, Write Enable controlled . . . 66
Table 25. Write AC characteristics, Chip Enable controlled . . . 68
Table 26. Reset and power-up AC characteristics . . . 69
Table 27. TBGA64 10 × 13 mm - 8 × 8 active ball array, 1 mm pitch, package mechanical data . . . 71
Table 28. Ordering information scheme . . . 72
Table 29. Top boot block addresses, M58LT128HST . . . 73
Table 30. Bottom boot block addresses, M58LT128HSB . . . 77
Table 31. Query structure overview . . . 81
Table 32. CFI query identification string . . . 82
Table 33. CFI query system interface information . . . 83
Table 34. Device geometry definition . . . 84
Table 35. Primary algorithm-specific extended query table . . . 85
Table 36. Protection Register information . . . 86
Table 37. Burst Read information . . . 86
Table 38. Bank and Erase Block region information . . . 87
Table 39. Bank and Erase Block region 1 information . . . 87
Table 40. Bank and Erase Block region 2 Information . . . 89
Table 41. Command Interface states - modify table, next state . . . 100
Table 42. Command Interface states - modify table, next output state . . . 103
Table 43. Command interface states - lock table, next state . . . 105
Table 44. Command interface states - lock table, next output state . . . 107
Table 45. Document revision history . . . 109

List of figures

Figure 1. Logic diagram . . . 9
Figure 2. TBGA64 package connections (top view through package) . . . 10
Figure 3. Memory map . . . 11
Figure 4. Protection Register memory map . . . 31
Figure 5. X-latency and data output configuration example . . . 43
Figure 6. Wait configuration example . . . 43
Figure 7. AC measurement I/O waveform . . . 54
Figure 8. AC measurement load circuit . . . 55
Figure 9. Asynchronous random access Read AC waveforms . . . 58
Figure 10. Asynchronous Page Read AC waveforms . . . 59
Figure 11. Synchronous Burst Read AC waveforms . . . 61
Figure 12. Single Synchronous Read AC waveforms . . . 62
Figure 13. Synchronous Burst Read Suspend AC waveforms . . . 63
Figure 14. Clock input AC waveform . . . 64
Figure 15. Write AC waveforms, Write Enable controlled . . . 65
Figure 16. Write AC waveforms, Chip Enable controlled . . . 67
Figure 17. Reset and power-up AC waveforms . . . 69
Figure 18. TBGA64 10 × 13 mm - 8 × 8 active ball array, 1 mm pitch, bottom view package outline . . . 70
Figure 19. Program flowchart and pseudo code . . . 91
Figure 20. Blank Check flowchart and pseudo code . . . 92
Figure 21. Buffer Program flowchart and pseudo code . . . 93
Figure 22. Program Suspend & Resume flowchart and pseudo code . . . 94
Figure 23. Block Erase flowchart and pseudo code . . . 95
Figure 24. Erase Suspend & Resume flowchart and pseudo code . . . 96
Figure 25. Protect/Unprotect operation flowchart and pseudo code . . . 97
Figure 26. Protection Register Program flowchart and pseudo code . . . 98
Figure 27. Buffer Enhanced Factory Program flowchart and pseudo code . . . 99

1 Description

The M58LT128HST/B are 128 Mbit (8 Mbit x 16) non-volatile secure Flash memories. They may be erased electrically at block level and programmed in system on a word-by-word basis using a 1.7 V to 2.0 V V_{DD} supply for the circuitry and a 2.7 V to 3.6 V V_{DDQ} supply for the Input/Output pins. An optional 9 V V_{PP} power supply is provided to accelerate factory programming.

The devices feature an asymmetrical block architecture, with an array of 131 blocks, divided into 8 Mbit banks. There are 15 banks each containing 8 main blocks of 64 Kwords, and one parameter bank containing 4 parameter blocks of 16 Kwords and 7 main blocks of 64 Kwords.

The multiple bank architecture allows dual operations, while programming or erasing in one bank, Read operations are possible in other banks. Only one bank at a time is allowed to be in Program or Erase mode. It is possible to perform burst reads that cross bank boundaries. The bank architecture is summarized in *Table 2*, and the memory map is shown in *Figure 3*. The parameter blocks are located at the top of the memory address space for the M58LT128HST, and at the bottom for the M58LT128HSB.

Each block can be erased separately. Erase can be suspended to perform a program or read operation in any other block, and then resumed. Program can be suspended to read data at any memory location except for the one being programmed, and then resumed. Each block can be programmed and erased over 100,000 cycles using the supply voltage V_{DD}. There is a buffer-enhanced factory programming command available to accelerate programming.

Program And Erase Commands Are Written To The command interface of the memory. An internal Program/Erase Controller manages the timings necessary for program and erase operations. The end of a program or erase operation can be detected and any error conditions identified in the Status Register. The command set required to control the memory is consistent with JEDEC standards.

The device supports Synchronous Burst Read and Asynchronous Read from all blocks of the memory array; at power-up the device is configured for Asynchronous Read. In Synchronous Burst Read mode, data is output on each clock cycle at frequencies of up to 52 MHz. The Synchronous Burst Read operation can be suspended and resumed.

The device features an Automatic Standby mode. When the bus is inactive during Asynchronous Read operations, the device automatically switches to the Automatic Standby mode. In this condition the power consumption is reduced to the standby value and the outputs are still driven.

The M58LT128HST/B features an instant, individual block protection scheme that allows any block to be protected or unprotected with no latency, enabling instant code and data protection. They can be protected individually preventing any accidental programming or erasure. There is an additional hardware protection against program and erase. When $V_{PP} \leq V_{PPLK}$ all blocks are protected against program or erase. All blocks are protected at power-up.

The device includes 17 Protection Registers and 2 Protection Register locks, one for the first Protection Register and the other for the 16 one-time-programmable (OTP) Protection Registers of 128 bits each. The first Protection Register is divided into two segments: a 64 bit segment containing a unique device number written by Numonyx, and a 64 bit segment OTP by the user. The user programmable segment can be permanently protected. *Figure 4*, shows the Protection Register memory map.

The M58LT128HST/B also has a full set of software security features that are not described in this datasheet, but are documented in a dedicated application note. For further information, please contact Numonyx.

The M58LT128HST/B are offered in a TBGA64, 10 × 13 mm, 1 mm pitch package. They are supplied with all the bits erased (set to ’1’).

Figure 1. Logic diagram

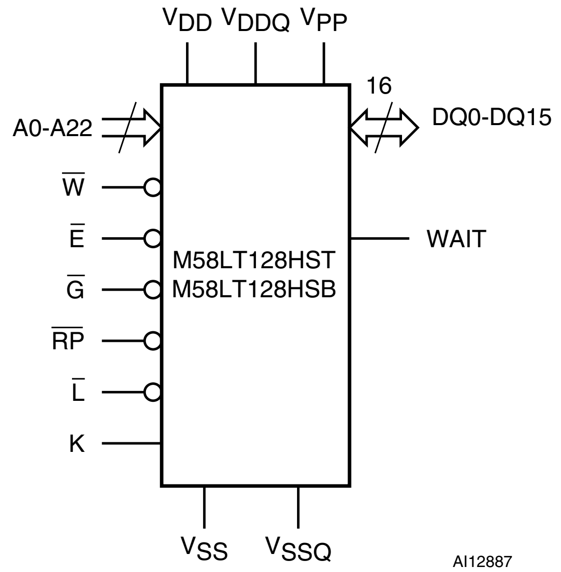


Table 1. Signal names

Signal name	Function	Direction
A0-A22	Address inputs	Inputs
DQ0-DQ15	Data input/outputs, command inputs	I/O
$\overline{E}$	Chip Enable	Input
$\overline{G}$	Output Enable	Input
$\overline{W}$	Write Enable	Input
$\overline{RP}$	Reset	Input
K	Clock	Input
$\overline{L}$	Latch Enable	Input
WAIT	Wait	Output
V_{DD}	Supply voltage	Input
V_{DDQ}	Supply voltage for input/output buffers	Input
V_{PP}	Optional supply voltage for fast program & erase	Input
V_{SS}	Ground	
V_{SSQ}	Ground input/output supply	Input
NC	Not Connected Internally	
DU	Do Not Use	

Figure 2. TBGA64 package connections (top view through package)

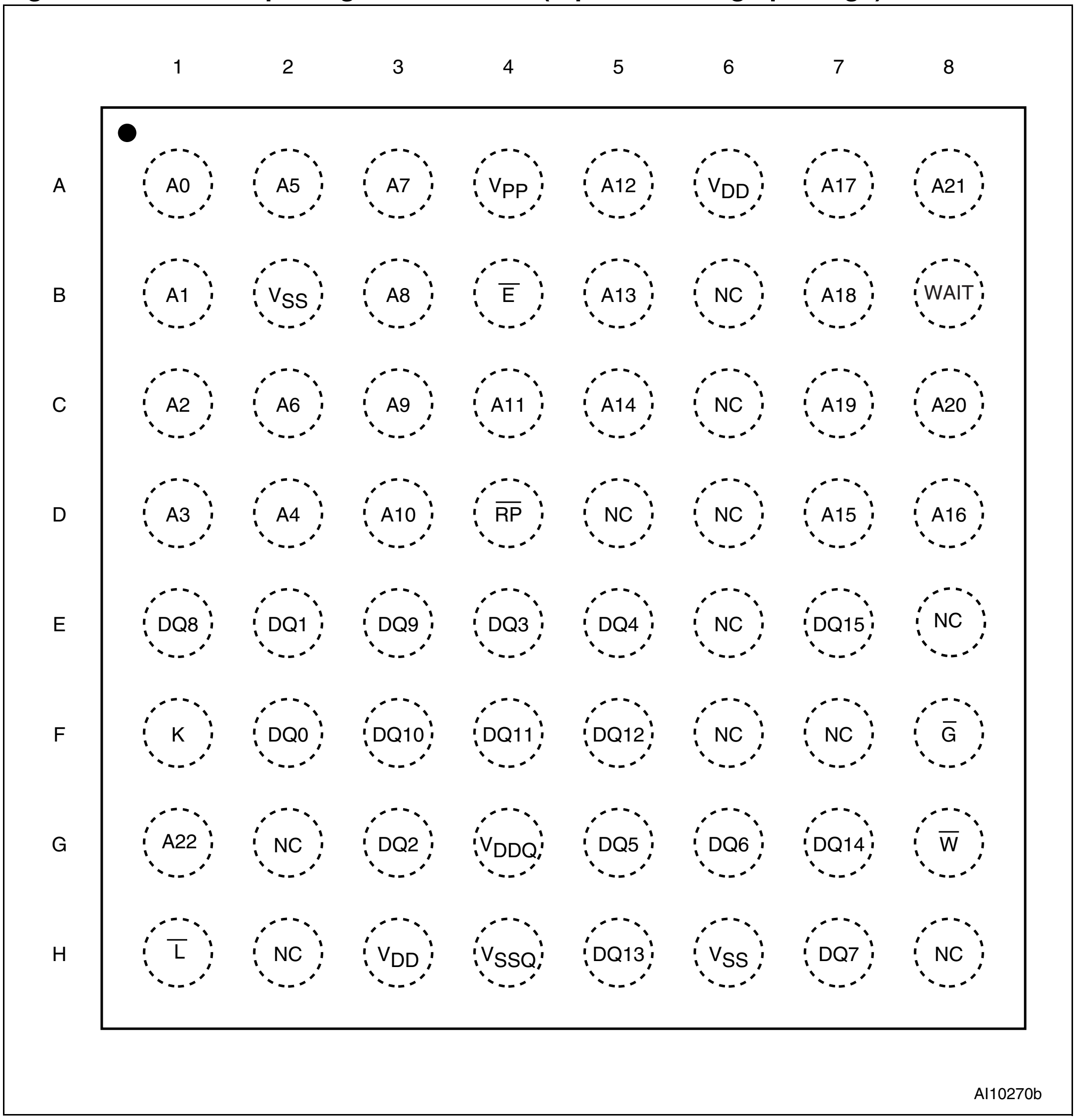
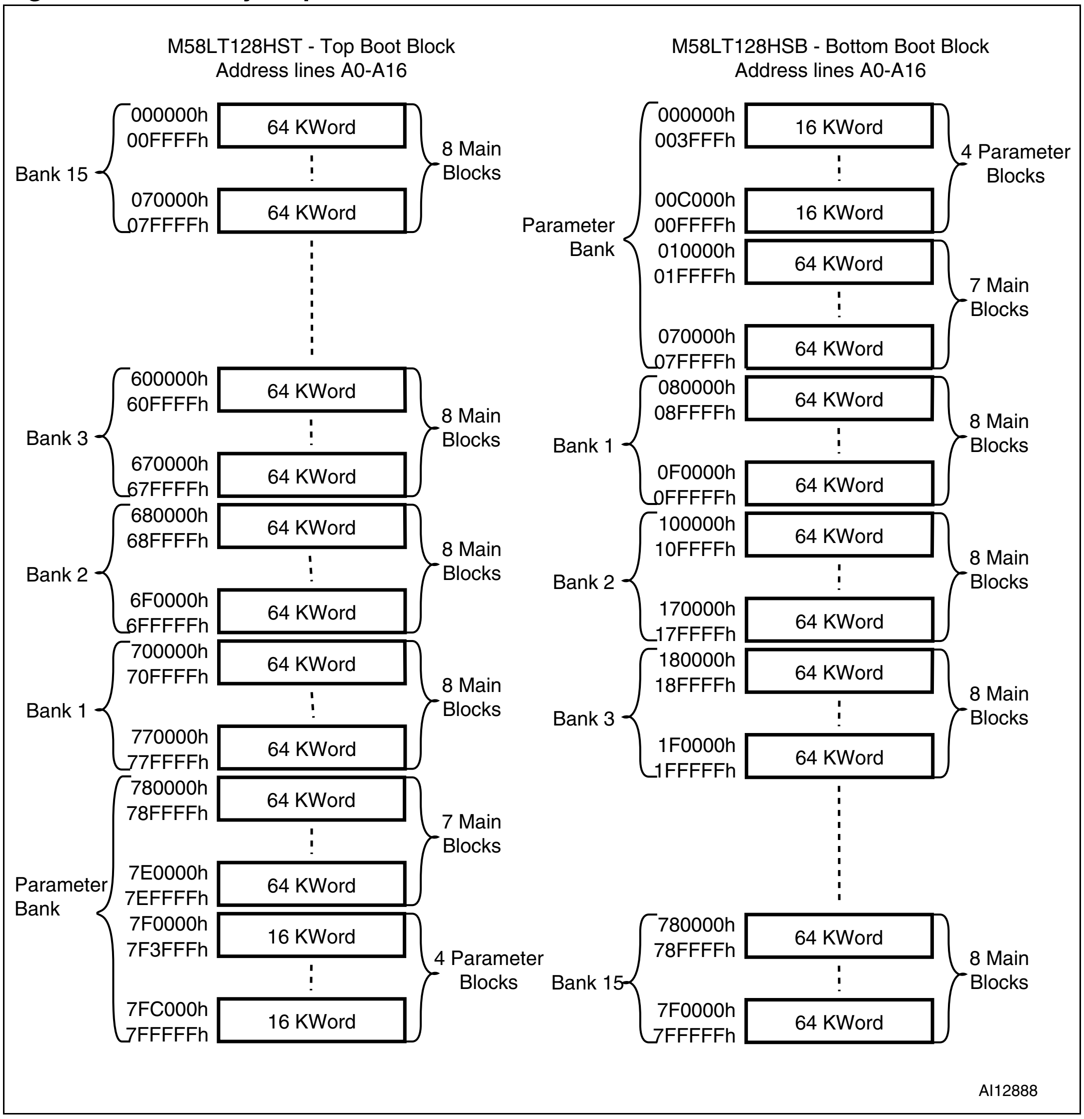


Table 2. Bank architecture

Number	Bank size	Parameter blocks	Main blocks
Parameter bank	8 Mbits	4 blocks of 16 Kwords	7 blocks of 64 Kwords
Bank 1	8 Mbits	-	8 blocks of 64 Kwords
Bank 2	8 Mbits	-	8 blocks of 64 Kwords
Bank 3	8 Mbits	-	8 blocks of 64 Kwords
⋮	⋮	⋮	⋮
Bank 14	8 Mbits	-	8 blocks of 64 Kwords
Bank 15	8 Mbits	-	8 blocks of 64 Kwords

Figure 3. Memory map



2 Signal descriptions

See *Figure 1: Logic diagram* and *Table 1: Signal names*, for a brief overview of the signals connected to this device.

2.1 Address inputs (A0-A22)

The Address Inputs select the cells in the memory array to access during Bus Read operations. During Bus Write operations they control the commands sent to the command interface of the Program/Erase Controller.

2.2 Data inputs/outputs (DQ0-DQ15)

The Data I/O output the data stored at the selected address during a Bus Read operation or input a command or the data to be programmed during a Bus Write operation.

2.3 Chip Enable ($\overline{E}$)

The Chip Enable input activates the memory control logic, input buffers, decoders and sense amplifiers. When Chip Enable is at V_{IL} and Reset is at V_{IH} the device is in active mode. When Chip Enable is at V_{IH} the memory is deselected, the outputs are high impedance and the power consumption is reduced to the standby level.

2.4 Output Enable ($\overline{G}$)

The Output Enable input controls data outputs during the Bus Read operation of the memory.

2.5 Write Enable ($\overline{W}$)

The Write Enable input controls the Bus Write operation of the memory's Command Interface. The data and address inputs are latched on the rising edge of Chip Enable or Write Enable whichever occurs first.

2.6 Reset ($\overline{RP}$)

The Reset input provides a hardware reset of the memory. When Reset is at V_{IL}, the memory is in reset mode: the outputs are high impedance and the current consumption is reduced to the Reset Supply Current I_{DD2}. Refer to *Table 20: DC characteristics - currents*, for the value of $I_{DD2.}$ After Reset, all blocks are in the protected state and the Configuration Register is reset. When Reset is at V_{IH}, the device is in normal operation. When exiting Reset mode the device enters Asynchronous Read mode, but a negative transition of Chip Enable or Latch Enable is required to ensure valid data outputs.

2.7 Latch Enable ($\overline{L}$)

Latch Enable latches the address bits on its rising edge. The address latch is transparent when Latch Enable is at V_{IL} and it is inhibited when Latch Enable is at V_{IH}.

2.8 Clock (K)

The Clock input synchronizes the memory to the microcontroller during Synchronous Read operations; the address is latched on a Clock edge (rising or falling, according to the configuration settings) when Latch Enable is at V_{IL}. Clock is ignored during Asynchronous Read and in Write operations.

2.9 Wait (WAIT)

Wait is an output signal used during synchronous read to indicate whether the data on the output bus are valid. This output is high impedance when Chip Enable is at V_{IH}, Output Enable is at V_{IH} or Reset is at V_{IL}. It can be configured to be active during the wait cycle or one clock cycle in advance.

2.10 V_{DD} supply voltage

V_{DD} provides the power supply to the internal core of the memory device. It is the main power supply for all operations (Read, Program, and Erase).

2.11 V_{DDQ} supply voltage

V_{DDQ} provides the power supply to the I/O pins and enables all outputs to be powered independently from V_{DD}.

2.12 V_{PP} program supply voltage

V_{PP} is both a control input and a power supply pin. The two functions are selected by the voltage range applied to the pin.

If V_{PP} is kept in a low voltage range (0V to V_{DDQ}) V_{PP} is seen as a control input. In this case a voltage lower than V_{PPLK} gives absolute protection against program or erase, while V_{PP} in the V_{PP1} range enables these functions (see Tables *20* and *21*, DC Characteristics for the relevant values). V_{PP} is only sampled at the beginning of a program or erase; a change in its value after the operation has started does not have any effect and program or erase operations continue.

If V_{PP} is in the range of V_{PPH} it acts as a power supply pin. In this condition V_{PP} must be stable until the Program/Erase algorithm is completed.

2.13 V_{SS} ground

V_{SS} ground is the reference for the core supply. It must be connected to the system ground.

2.14 V_{SSQ} ground

V_{SSQ} ground is the reference for the input/output circuitry driven by V_{DDQ}. V_{SSQ} must be connected to V_{SS}

Note: Each device in a system should have V_{DD}, V_{DDQ} and V_{PP} decoupled with a 0.1µF ceramic capacitor close to the pin (high-frequency, inherently low inductance capacitors should be as close as possible to the package). See Figure 8: AC measurement load circuit. The PCB track widths should be sufficient to carry the required V_{PP} program and erase currents.

3 Bus operations

There are six standard bus operations that control the device. These are Bus Read, Bus Write, Address Latch, Output Disable, Standby and Reset. See *Table 3: Bus operations*, for a summary.

Typically glitches of less than 5 ns on Chip Enable or Write Enable are ignored by the memory and do not affect Bus Write operations.

3.1 Bus Read

Bus Read operations are used to output the contents of the Memory Array, the Electronic Signature, the Status Register and the Common Flash Interface. Both Chip Enable and Output Enable must be at V_{IL} to perform a Read operation. The Chip Enable input is used to enable the device. Output Enable is used to gate data onto the output. The data read depends on the previous command written to the memory (see Command Interface section). See Figures *9*, *10* and *11* Read AC Waveforms, and Tables *22* and *23* Read AC Characteristics, for details of when the output becomes valid.

3.2 Bus Write

Bus Write operations write commands to the memory or latch input data to be programmed. A Bus Write operation is initiated when Chip Enable and Write Enable are at V_{IL} with Output Enable at V_{IH}. Commands, input data and addresses are latched on the rising edge of Write Enable or Chip Enable, whichever occurs first. The addresses must be latched prior to the write operation by toggling Latch Enable (when Chip Enable is at V_{IL}). The Latch Enable must be tied to V_{IH} during the Bus Write operation.

See Figures *15* and *16*, Write AC waveforms, and Tables *24* and *25*, Write AC characteristics, for details of the timing requirements.

3.3 Address Latch

Address Latch operations input valid addresses. Both Chip Enable and Latch Enable must be at V_{IL} during Address Latch operations. The addresses are latched on the rising edge of Latch Enable.

3.4 Output Disable

The outputs are high impedance when the Output Enable is at V_{IH}.

3.5 Standby

Standby disables most of the internal circuitry, allowing a substantial reduction of the current consumption. The memory is in Standby when Chip Enable and Reset are at V_{IH}. The power consumption is reduced to the standby level I_{DD3} and the outputs are set to high impedance, independently from the Output Enable or Write Enable inputs. If Chip Enable switches to V_{IH} during a Program or Erase operation, the device enters Standby mode when finished.

3.6 Reset

During Reset mode the memory is deselected and the outputs are high impedance. The memory is in Reset mode when Reset is at V_{IL}. The power consumption is reduced to the Reset level, independently from the Chip Enable, Output Enable, or Write Enable inputs. If Reset is pulled to V_{SS} during a Program or Erase, this operation is aborted and the memory content is no longer valid.

Table 3. Bus operations(1)

Operation	$\overline{E}$	$\overline{G}$	$\overline{W}$	$\overline{L}$	$\overline{RP}$	WAIT(2)	DQ15-DQ0
Bus Read	V_{IL}	V_{IL}	V_{IH}	V_{IL}(3)	V_{IH}		Data Output
Bus Write	V_{IL}	V_{IH}	V_{IL}	V_{IL}(3)	V_{IH}		Data Input
Address Latch	V_{IL}	X	V_{IH}	V_{IL}	V_{IH}		Data Output or Hi-Z(4)
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{IH}	Hi-Z	Hi-Z
Standby	V_{IH}	X	X	X	V_{IH}	Hi-Z	Hi-Z
Reset	X	X	X	X	V_{IL}	Hi-Z	Hi-Z

1. X = 'Don't care'.
2. WAIT signal polarity is configured using the Set Configuration Register command.
3. $\overline{L}$ can be tied to V_{IH} if the valid address has been previously latched.
4. Depends on $\overline{G}$.

4 Command interface

All Bus Write operations to the memory are interpreted by the Command Interface. Commands consist of one or more sequential Bus Write operations. An internal Program/Erase Controller handles all timings and verifies the correct execution of the program and erase commands. The Program/Erase Controller provides a Status Register whose output may be read at any time to monitor the progress or the result of the operation.

When exiting from Reset or whenever V_{DD} is lower than V_{LKO}, the Command Interface is reset to Read mode when power is first applied. Command sequences must be followed exactly. Any invalid combination of commands is ignored.

Refer to *Table 4: Command codes*, *Table 5: Standard commands*, *Table 6: Factory commands*, and *Appendix D: Command Interface state tables* for a summary of the Command Interface.

Table 4. Command codes

Hex Code	Command
01h	Block Protect Confirm
03h	Set Configuration Register Confirm
10h	Alternative Program Setup
20h	Block Erase Setup
40h	Program Setup
50h	Clear Status Register
60h	Block Protect Setup, Block Unprotect Setup and Set Configuration Register Setup
70h	Read Status Register
80h	Buffer Enhanced Factory Program Setup
90h	Read Electronic Signature
98h	Read CFI Query
B0h	Program/Erase Suspend
BCh	Blank Check Setup
C0h	Protection Register Program
CBh	Blank Check Confirm
D0h	Program/Erase Resume, Block Erase Confirm, Block Unprotect Confirm, Buffer Program or Buffer Enhanced Factory Program Confirm
E8h	Buffer Program
FFh	Read Array

4.1 Read Array command

The Read Array command returns the addressed bank to Read Array mode.

One Bus Write cycle is required to issue the Read Array command. Once a bank is in Read Array mode, subsequent read operations outputs the data from the memory array.

A Read Array command can be issued to any banks while programming or erasing in another bank. If the Read Array command is issued to a bank currently executing a program or erase operation, the bank returns to Read Array mode, but the Program or Erase operation continues. However, the data output from the bank is not guaranteed until the Program or Erase operation is finished. The Read modes of other banks are not affected.

4.2 Read Status Register command

The device contains a Status Register that is used to monitor program or erase operations.

The Read Status Register command is used to read the contents of the Status Register for the addressed bank.

One Bus Write cycle is required to issue the Read Status Register command. Once a bank is in Read Status Register mode, subsequent Read operations output the contents of the Status Register.

The Status Register data is latched on the falling edge of the Chip Enable or Output Enable signals. Either Chip Enable or Output Enable must be toggled to update the Status Register data

The Read Status Register command can be issued at any time, even during Program or Erase operations. The Read Status Register command only changes the Read mode of the addressed bank. The Read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read the Status Register. A Read Array command is required to return the bank to Read Array mode.

See *Table 9* for the description of the Status Register Bits.

4.3 Read Electronic Signature command

The Read Electronic Signature command is used to read the manufacturer and device codes, the protection status of the addressed bank, the Protection Register, and the Configuration Register.

One Bus Write cycle is required to issue the Read Electronic Signature command. Once a bank is in Read Electronic Signature mode, subsequent Read operations in the same bank output the manufacturer code, the device code, the protection status of the addressed bank, the Protection Register, or the Configuration Register (see *Table 8*).

The Read Electronic Signature command can be issued at any time, even during Program or Erase operations, except during Protection Register Program operations. Dual operations between the parameter bank and the electronic signature location are not allowed (see *Table 15: Dual operation limitations* for details).

If a Read Electronic Signature command is issued to a bank that is executing a Program or Erase operation the bank go into Read Electronic Signature mode. Subsequent Bus Read cycles output the Electronic Signature data and the Program/Erase Controller continues to program or erase in the background.

The Read Electronic Signature command only changes the Read mode of the addressed bank. The Read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read the Electronic Signature. A Read Array command is required to return the bank to Read Array mode.

4.4 Read CFI Query command

The Read CFI Query command is used to read data from the Common Flash Interface (CFI).

One Bus Write cycle is required to issue the Read CFI Query command. Once a bank is in Read CFI Query mode, subsequent Bus Read operations in the same bank read from the Common Flash Interface. The Read CFI Query command can be issued at any time, even during Program or Erase operations.

If a Read CFI Query command is issued to a bank that is executing a Program or Erase operation the bank goes into Read CFI Query mode. Subsequent Bus Read cycles output the CFI data and the Program/Erase controller continues to Program or Erase in the background.

The Read CFI Query command only changes the Read mode of the addressed bank. The Read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read from the CFI. A Read Array command is required to return the bank to Read Array mode. Dual operations between the Parameter Bank and the CFI memory space are not allowed (see *Table 15: Dual operation limitations* for details).

See *Appendix B: Common Flash Interface*, Tables *31*, *32*, *33*, *34*, *35*, *36*, *37*, *38*, *39* and *40* for details on the information contained in the Common Flash Interface memory area.

4.5 Clear Status Register command

The Clear Status Register command can be used to reset (set to ‘0’) all error bits (SR1, 3, 4 and 5) in the Status Register.

One Bus Write cycle is required to issue the Clear Status Register command. The Clear Status Register command does not affect the Read mode of the bank.

The error bits in the Status Register do not automatically return to ‘0’ when a new command is issued. The error bits in the Status Register should be cleared before attempting a new Program or Erase command.

4.6 Block Erase command

The Block Erase command is used to erase a block. It sets all the bits within the selected block to ’1’. All previous data in the block is lost.

If the block is protected then the erase operation aborts, the data in the block is not changed, and the Status Register outputs the error.

The following two Bus Write cycles are required to issue the command:

- The first bus cycle sets up the Block Erase command.
- The second latches the block address and starts the Program/Erase Controller.

If the second bus cycle is not the Block Erase Confirm code, Status Register bits SR4 and SR5 are set and the command is aborted.

Once the command is issued, the bank enters Read Status Register mode and any Read operation within the addressed bank outputs the contents of the Status Register. A Read Array command is required to return the bank to Read Array mode.

During Block Erase operations the bank containing the block being erased only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query, and the Program/Erase Suspend command; all other commands are ignored.

The Block Erase operation aborts if Reset, $\overline{RP}$, goes to V_{IL}. As data integrity cannot be guaranteed when the Block Erase operation is aborted, the block must be erased again.

Refer to *Chapter 8: Dual operations and multiple bank architecture* for detailed information about simultaneous operations allowed in banks not being erased.

Typical erase times are given in *Table 16: Program/erase times and endurance cycles*.

See *Appendix C*, *Figure 23: Block Erase flowchart and pseudo code* for a suggested flowchart for using the Block Erase command.

4.7 Blank Check command

The Blank Check command is used to check whether a block has been completely erased. Only one block at a time can be checked. To use the Blank Check command, V_{PP} must be equal to V_{PPH}. If V_{PP} is not equal to V_{PPH}, the device ignores the command and no error is shown in the Status Register.

The following two bus cycles are required to issue the Blank Check command:

- The first bus cycle writes the Blank Check command (BCh) to any address in the block to be checked.
- The second bus cycle writes the Blank Check Confirm command (CBh) to any address in the block to be checked and starts the Blank Check operation.

If the second bus cycle is not Blank Check Confirm, Status Register bits SR4 and SR5 are set to '1' and the command aborts.

Once the command is issued the addressed bank automatically enters the Status Register mode and further reads the Status Register contents within the bank output.

The only operation permitted during Blank Check is Read Status Register. Dual operations are not supported while a Blank Check operation is in progress. Blank Check operations cannot be suspended and are not allowed while the device is in Program/Erase Suspend.

The SR7 Status Register bit indicates the status of the Blank Check operation in progress: SR7 = '0' means that the Blank Check operation is still ongoing, and SR7 = '1' means that the operation is complete.

The SR5 Status Register bit goes High (SR5 = '1') to indicate if the Blank Check operation has failed.

At the end of the operation the bank remains in the Read Status Register mode until another command is written to the Command Interface.

See *Appendix C*, *Figure 20: Blank Check flowchart and pseudo code* for a suggested flowchart for using the Blank Check command.

Typical Blank Check times are given in *Table 16: Program/erase times and endurance cycles*.

4.8 Program command

The Program command is used to program a single word to the memory array.

If the block being programmed is protected, then the Program operation aborts, the data in the block is not changed, and the Status Register outputs the error.

The following two Bus Write cycles are required to issue the Program Command:

- The first bus cycle sets up the Program command.
- The second latches the address and data to be programmed and starts the Program/Erase Controller.

Once the programming has started, Read operations in the bank being programmed output the Status Register content.

During a Program operation, the bank containing the word being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and the Program/Erase Suspend commands; all other commands are ignored. A Read Array command is required to return the bank to Read Array mode.

Refer to *Chapter 8: Dual operations and multiple bank architecture* for detailed information about simultaneous operations allowed in banks not being programmed.

Typical Program times are given in *Table 16: Program/erase times and endurance cycles*.

The Program operation aborts if Reset, $\overline{RP}$, goes to V_{IL}. As data integrity cannot be guaranteed when the Program operation is aborted, the word must be reprogrammed.

See *Appendix C*, *Figure 19: Program flowchart and pseudo code* for the flowchart for using the Program command.

4.9 Buffer Program command

The Buffer Program Command makes use of the device's 32-word Write Buffer to accelerate programming. Up to 32 words can be loaded into the Write Buffer, which can dramatically reduce in-system programming time compared to the standard non-buffered Program command.

Four successive steps are required to issue the Buffer Program command:

1. The first Bus Write cycle sets up the Buffer Program command. The setup code can be addressed to any location within the targeted block.

After the first Bus Write cycle, Read operations in the bank output the contents of the Status Register. Status Register bit SR7 should be read to check that the buffer is available (SR7 = '1'). If the buffer is not available (SR7 = '0'), re-issue the Buffer Program command to update the Status Register contents.

2. The second Bus Write cycle sets up the number of words to be programmed. Value "n" is written to the same block address, where n+1 is the number of words to be programmed.
3. Use n+1 Bus Write cycles to load the address and data for each word into the Write Buffer. Addresses must lie within the range from the start address to the start address + n, where the start address is the location of the first data to be programmed. Optimum performance is obtained when the start address corresponds to a 32-word boundary.
4. The final Bus Write cycle confirms the Buffer Program command and starts the program operation.

All the addresses used in the Buffer Program operation must lie within the same block.

Invalid address combinations or an incorrect sequence of Bus Write cycles sets an error in the Status Register and aborts the operation without affecting the data in the memory array.

If the block being programmed is protected an error is set in the Status Register, and the operation aborts without affecting the data in the memory array.

During Buffer Program operations the bank being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query, and the Program/Erase Suspend command; all other commands are ignored.

Refer to *Chapter 8: Dual operations and multiple bank architecture* for detailed information about simultaneous operations allowed in banks not being programmed.

See *Appendix C*, *Figure 21: Buffer Program flowchart and pseudo code* for a suggested flowchart on using the Buffer Program command.

4.10 Buffer Enhanced Factory Program command

The Buffer Enhanced Factory Program command has been specially developed to accelerate programming in manufacturing environments where the programming time is critical. It is used to program one or more Write Buffer(s) of 32 words to a block. Once the device enters Buffer Enhanced Factory Program mode, the Write Buffer can be reloaded any number of times as long as the address remains within the same block. Only one block can be programmed at a time.

If the block being programmed is protected, then the Program operation aborts, the data in the block is not changed and the Status Register outputs the error.

The use of the Buffer Enhanced Factory Program command requires the following operating conditions:

- V_{PP} must be set to V_{PPH}
- V_{DD} must be within operating range
- Ambient temperature T_A must be 30°C ± 10°C
- The targeted block must be unprotected
- The start address must be aligned with the start of a 32-word buffer boundary
- The address must remain the Start Address throughout programming.

Dual operations are not supported during the Buffer Enhanced Factory Program operation and the command cannot be suspended.

The Buffer Enhanced Factory Program Command consists of three phases: the Setup Phase, the Program and Verify Phase, and the Exit Phase. Refer to *Table 6: Factory commands* for detailed information.

4.10.1 Setup phase

The Buffer Enhanced Factory Program command requires the following two Bus Write cycles to initiate the command.

- The first Bus Write cycle sets up the Buffer Enhanced Factory Program command.
- The second Bus Write cycle confirms the command.

After the Confirm command is issued, Read operations output the contents of the Status Register. The read Status Register command must not be issued, othewise it is interpreted as data to program.

The Status Register Program/Erase Controller bit SR7 should be read to check that the Program/Erase Controller is ready to proceed to the next phase.

If an error is detected, SR4 goes high (set to ‘1’) and the Buffer Enhanced Factory Program operation is terminated. See *Chapter 5: Status Register* for details on the error.

4.10.2 Program and verify phase

The program and verify phase requires 32 cycles to program the 32 words to the Write Buffer. The data is stored sequentially, starting at the first address of the Write Buffer, until the Write Buffer is full (32 words). To program less than 32 words, the remaining words should be programmed with FFFFh.

The following three successive steps are required to issue and execute the program and verify phase of the command.

1. Use one Bus Write operation to latch the Start Address and the first word to be programmed. The Status Register Bank Write status bit SR0 should be read to check that the Program/Erase Controller is ready for the next word.
2. Each subsequent word to be programmed is latched with a new Bus Write operation. The address must remain the start address as the Program/Erase Controller increments the address location. If any address is given that is not in the same block as the start address, the program and verify phase terminates. Status Register bit SR0 should be read between each Bus Write cycle to check that the Program/Erase Controller is ready for the next word.
3. Once the Write Buffer is full, the data is programmed sequentially to the memory array. After the Program operation, the device automatically verifies the data and reprograms, if necessary.

The program and verify phase can be repeated, without re-issuing the command, to program additional 32 word locations as long as the address remains in the same block.

4. Finally, after all words, or the entire block has been programmed, write one Bus Write operation to any address outside the block containing the start address, to terminate program and verify phase.

Status Register bit SR0 must be checked to determine whether the Program operation is finished. The Status Register may be checked for errors at any time but it must be checked after the entire block has been programmed.

4.10.3 Exit phase

Status Register Program/Erase Controller bit SR7 set to ‘1’ indicates that the device has exited the Buffer Enhanced Factory Program operation and returned to Read Status Register mode. A full Status Register check should be done to ensure that the block has been successfully programmed. See Section *Table 5: Status Register* for more details.

For optimum performance the Buffer Enhanced Factory Program command should be limited to a maximum of 100 program/erase cycles per block. If this limit is exceeded, the internal algorithm continues to work properly but some degradation in performance is possible. Typical program times are given in *Table 16*.

See *Appendix C*, *Figure 27: Buffer Enhanced Factory Program flowchart and pseudo code* for a suggested flowchart on using the Buffer Enhanced Factory Program command.

4.11 Program/Erase Suspend command

The Program/Erase Suspend command is used to pause a Program or Block Erase operation. The command can be addressed to any bank. The Program/Erase Resume command is required to restart the suspended operation.

One Bus Write cycle is required to issue the Program/Erase Suspend command. Once the Program/Erase Controller has paused, bits SR7, SR6, and/or SR2 of the Status Register are set to ‘1’.

The following commands are accepted during Program/Erase Suspend:

– Program/Erase Resume
– Read Array (data from erase-suspended blocks or program-suspended words is not valid)
– Read Status Register
– Read Electronic Signature
– Read CFI Query

In addition, if the suspended operation is a Block Erase, then the following commands are also accepted:

– Clear Status Register
– Program (except in erase-suspended blocks)
– Buffer Program (except in erase suspended blocks)
– Block Protect
– Block Unprotect
– Set Configuration Register

During an Erase Suspend, the block being erased can be protected by issuing the Block Protect command. When the Program/Erase Resume command is issued, the operation completes.

It is possible to accumulate multiple suspend operations. For example, suspend an Erase operation, start a Program operation, suspend the Program operation, and then read the array.

If a Program command is issued during a Block Erase Suspend, the Erase operation cannot be resumed until the program operation has completed.

The Program/Erase Suspend command does not change the read mode of the banks. If the suspended bank was in Read Status Register, Read Electronic signature, or Read CFI Query mode, the bank remains in that mode and outputs the corresponding data.

Refer to Section *Table 8: Dual operations and multiple bank architecture* for detailed information about simultaneous operations allowed during Program/Erase Suspend.

During a Program/Erase Suspend, the device can be placed in Standby mode by driving Chip Enable to V_{IH}. Program/Erase is aborted if Reset, $\overline{RP}$, goes to V_{IL}.

See *Appendix C*, *Figure 22: Program Suspend & Resume flowchart and pseudo code* and *Figure 24: Erase Suspend & Resume flowchart and pseudo code* for flowcharts for using the Program/Erase Suspend command.

4.12 Program/Erase Resume command

The Program/Erase Resume command is used to restart the Program or Erase operation suspended by the Program/Erase Suspend command. One Bus Write cycle is required to issue the command. The command can be issued to any address.

The Program/Erase Resume command does not change the Read mode of the banks. If the suspended bank was in Read Status Register, Read Electronic signature, or Read CFI Query mode, the bank remains in that mode and outputs the corresponding data.

If a Program command is issued during a Block Erase Suspend, then the erase cannot be resumed until the program operation has completed.

See *Appendix C*, *Figure 22: Program Suspend & Resume flowchart and pseudo code* and *Figure 24: Erase Suspend & Resume flowchart and pseudo code* for flowcharts for using the Program/Erase Resume command.

4.13 Protection Register Program command

The Protection Register Program command is used to program the user segments of the Protection Register and the two Protection Register Locks.

The device features 16 OTP segments of 128 bits and one OTP segment of 64 bits, as shown in *Figure 4: Protection Register memory map*.

The segments are programmed one word at a time. When shipped, all bits in the segment are set to ‘1’. The user can only program the bits to ‘0’.

The following two Bus Write cycles are required to issue the Protection Register Program command:

- The first bus cycle sets up the Protection Register Program command.
- The second latches the address and data to be programmed to the Protection Register and starts the Program/Erase Controller.

Read operations to the bank being programmed output the Status Register content after the Program operation has started. Attempting to program a previously protected Protection Register results in a Status Register error.

The Protection Register Program cannot be suspended. Dual operations between the Parameter Bank and the Protection Register memory space are not allowed (see *Table 15: Dual operation limitations* for details).

The two Protection Register Locks are used to protect the OTP segments from further modification. The protection of the OTP segments is not reversible. Refer to *Figure 4: Protection Register memory map* and *Table 8: Protection Register locks* for details on the Lock bits.

See *Appendix C*, *Figure 26: Protection Register Program flowchart and pseudo code* for a flowchart for using the Protection Register Program command.

4.14 Set Configuration Register command

The Set Configuration Register command is used to write a new value to the Configuration Register.

The following two Bus Write cycles are required to issue the Set Configuration Register command.

- The first cycle sets up the Set Configuration Register command and the address corresponding to the Configuration Register content.
- The second cycle writes the Configuration Register data and the Confirm command.

The Configuration Register data must be written as an address during the bus write cycles, such as A0 = CR0, A1 = CR1, …, A15 = CR15. Addresses A16-A22 are ignored.

Read operations output the array content after the Set Configuration Register command is issued. The Read Electronic Signature command is required to read the updated contents of the Configuration Register.

4.15 Block Protect command

The Block Protect command is used to protect a block and prevent Program or Erase operations from changing the data in it. All blocks are protected after power-up or reset.

The following two Bus Write cycles are required to issue the Block Protect command:

- The first bus cycle sets up the Block Protect command.
- The second Bus Write cycle latches the block address and protects the block.

Once the command has been issued, subsequent Bus Read operations read the Status Register. The protection status can be monitored for each block using the Read Electronic Signature command.

Refer to *Section 9: Block protection* for a detailed explanation. See *Appendix C*, *Figure 25: Protect/Unprotect operation flowchart and pseudo code* for a flowchart for using the Block Protect command.

4.16 Block Unprotect command

The Block Unprotect command is used to unprotect a block, allowing the block to be programmed or erased.

The following two Bus Write cycles are required to issue the Block Unprotect command:

- The first bus cycle sets up the Block Unprotect command.
- The second Bus Write cycle latches the block address and unprotects the block.

Once the command has been issued, subsequent Bus Read operations read the Status Register. The protection status can be monitored for each block using the Read Electronic Signature command.

Refer to *Section 9: Block protection* for a detailed explanation and *Appendix C*, *Figure 25: Protect/Unprotect operation flowchart and pseudo code* for a flowchart for using the Block Unprotect command.

Table 5. Standard commands(1)

Commands	Cycles	Bus operations					
		1st cycle			2nd cycle		
		Op.	Add	Data	Op.	Add	Data
Read Array	1+	Write	BKA	FFh	Read	WA	RD
Read Status Register	1+	Write	BKA	70h	Read	BKA(2)	SRD
Read Electronic Signature	1+	Write	BKA	90h	Read	BKA(2)	ESD
Read CFI Query	1+	Write	BKA	98h	Read	BKA(2)	QD
Clear Status Register	1	Write	X	50h			
Block Erase	2	Write	BKA or BA(3)	20h	Write	BA	D0h
Program	2	Write	BKA or WA(3)	40h or 10h	Write	WA	PD
Buffer Program(4)	n+4	Write	BA	E8h	Write	BA	n
		Write	PA_1	PD_1	Write	PA_2	PD_2
		Write	PA_{n+1}	PD_{n+1}	Write	X	D0h
Program/Erase Suspend	1	Write	X	B0h			
Program/Erase Resume	1	Write	X	D0h			
Protection Register Program	2	Write	PRA	C0h	Write	PRA	PRD
Set Configuration Register	2	Write	CRD	60h	Write	CRD	03h
Block Protect	2	Write	BKA or BA(3)	60h	Write	BA	01h
Block Unprotect	2	Write	BKA or BA(3)	60h	Write	BA	D0h

1. X = 'Don't Care', WA = word Address in targeted bank, RD = Read Data, SRD = Status Register Data, ESD = Electronic Signature Data, QD = Query Data, BA = Block Address, BKA = Bank Address, PD = Program Data, PRA = Protection Register Address, PRD = Protection Register Data, CRD = Configuration Register Data.
2. Must be same bank as in the first cycle. The signature addresses are listed in *Table 7*.
3. Any address within the bank can be used.
4. n+1 is the number of words to be programmed.

Table 6. Factory commands

Command	Phase	Cycles	Bus Write operations(1)									
			1st		2nd		3rd		Final -1		Final	
			Add	Data	Add	Data	Add	Data	Add	Data	Add	Data
Blank Check		2	BA	BCh	BA	CBh						
Buffer Enhanced Factory Program	Setup	2	BKA or WA(2)	80h	WA_1	D0h						
	Program/ Verify(3)	≥32	WA_1	PD_1	WA_1	PD_2	WA_1	PD_3	WA_1	PD_{31}	WA_1	PD_{32}
	Exit	1	NOT BA_1(4)	X								

1. WA = word Address in targeted bank, BKA = Bank Address, PD = Program Data, BA = Block Address, X = 'Don't Care'.
2. Any address within the bank can be used.
3. The Program/Verify phase can be executed any number of times as long as the data is to be programmed to the same block.
4. WA_1 is the Start Address, NOT BA_1 = Not Block Address of WA_1.

Table 7. Electronic signature codes

Code		Address (h)	Data (h)
Manufacturer code		Bank Address + 000	0020
Device code	Top	Bank Address + 001	88D6 (M58LT128HST)
	Bottom	Bank Address + 001	88D7 (M58LT128HSB)
Block protection	Protected	Block Address + 002	0001
	Unprotected		0000
Configuration Register		Bank Address + 005	CR(1)
Protection Register PR0 Lock	Numonyx Factory Default	Bank Address + 080	0002
	OTP Area Permanently Protected		0000
Protection Register PR0		Bank Address + 081 Bank Address + 084	Unique Device Number
		Bank Address + 085 Bank Address + 088	OTP Area
Protection Register PR1 through PR16 Lock		Bank Address + 089	PRLD(1)
Protection Registers PR1-PR16		Bank Address + 08A Bank Address + 109	OTP Area

1. CR = Configuration Register, PRLD = Protection Register Lock Data.

Figure 4. Protection Register memory map

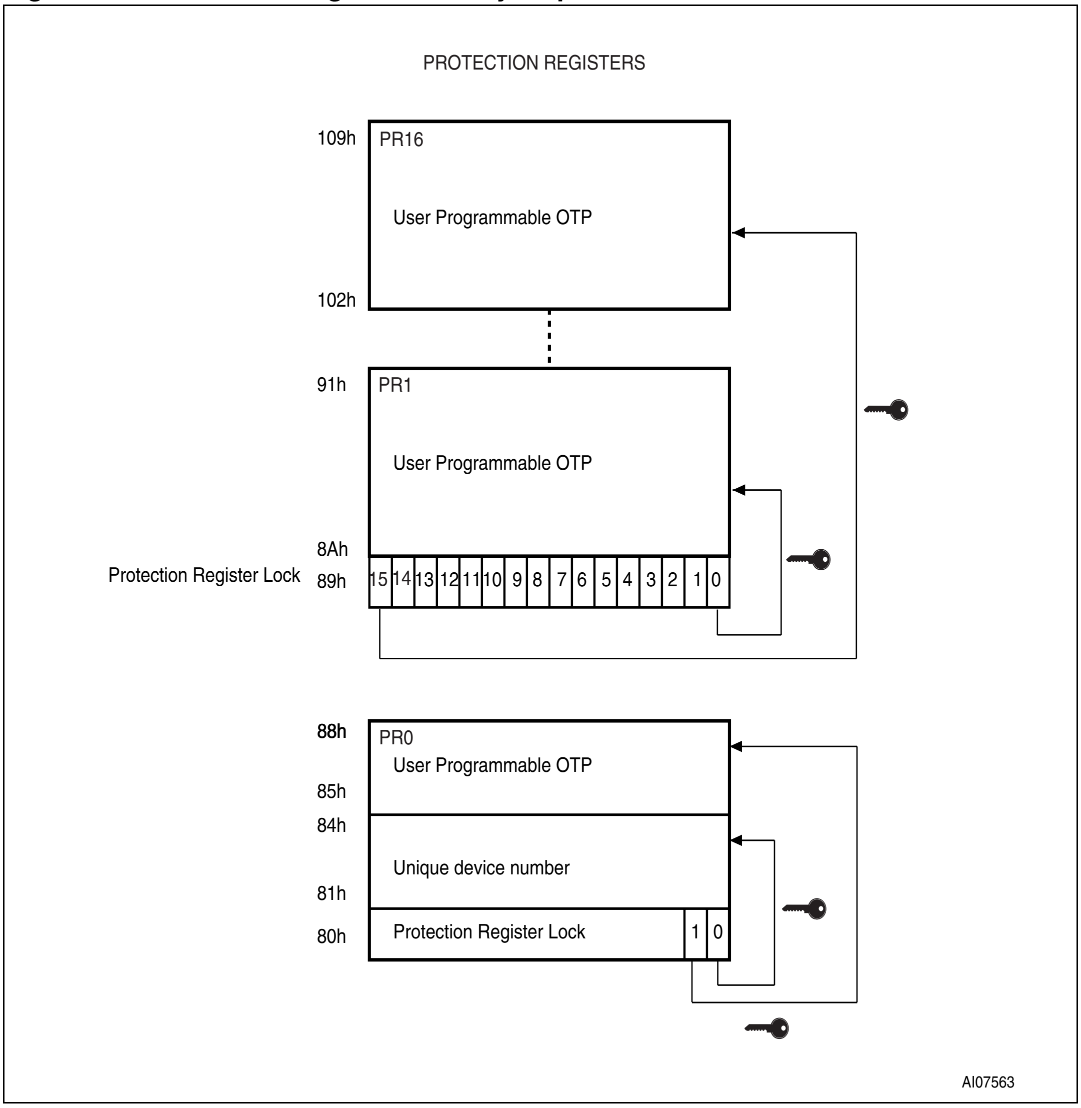


Table 8. Protection Register locks

Lock			Description
Number	Address	Bits	
Lock 1	80h	Bit 0	Preprogrammed to protect unique device number, address 81h to 84h in PR0
		Bit 1	Protects 64 bits of OTP segment, address 85h to 88h in PR0
		Bits 2 to 15	reserved
Lock 2	89h	Bit 0	Protects 128 bits of OTP segment PR1
		Bit 1	Protects 128 bits of OTP segment PR2
		Bit 2	Protects 128 bits of OTP segment PR3
		⋮	⋮
		Bit 13	Protects 128 bits of OTP segment PR14
		Bit 14	Protects 128 bits of OTP segment PR15
		Bit 15	Protects 128 bits of OTP segment PR16

5 Status Register

The Status Register provides information on the current or previous Program or Erase operations. The Read Status Register command reads the contents of the Status Register (refer to *Section 4.2: Read Status Register command* for more details. To output the contents, the Status Register is latched and updated on the falling edge of the Chip Enable or Output Enable signals, and can be read until Chip Enable or Output Enable returns to V_{IH}. The Status Register can only be read using single Asynchronous or Single Synchronous reads. If no Read Array command has been issued, Bus Read operations from any address within the bank always read the Status Register during Program and Erase operations.

The various bits convey information about the status and any errors of the operation. Bits SR7, SR6, SR2, and SR0 give information on the status of the device and are set and reset by the device. Bits SR5, SR4, SR3, and SR1 give information about any errors; they are set by the device but must be reset by issuing a Clear Status Register command or a hardware reset. If an error bit is set to ‘1’ the Status Register should be reset before issuing another command.

The bits in the Status Register are summarized in *Table 9: Status Register bits*. Refer to *Table 9* in conjunction with the following text descriptions.

5.1 Program/Erase Controller status bit (SR7)

The Program/Erase Controller status bit indicates whether the Program/Erase Controller is active or inactive in any bank.

When the Program/Erase Controller status bit is Low (set to ‘0’), the Program/Erase Controller is active. When the bit is High (set to ‘1’), the Program/Erase Controller is inactive and the device is ready to process a new command.

The Program/Erase Controller status bit is Low immediately after a Program/Erase Suspend command is issued, until the Program/Erase Controller pauses. After the Program/Erase Controller pauses, the bit is High.

5.2 Erase Suspend status bit (SR6)

The Erase Suspend status bit indicates that an erase operation has been suspended. When the Erase Suspend status bit is High (set to ‘1’), a Program/Erase Suspend command has been issued and the memory is waiting for a Program/Erase Resume command.

The Erase Suspend status bit should only be considered valid when the Program/Erase Controller status bit is High (Program/Erase Controller inactive). SR6 is set within the Erase Suspend latency time of the Program/Erase Suspend command being issued; therefore, the memory may still complete the operation rather than entering the Suspend mode.

When a Program/Erase Resume command is issued, the Erase Suspend status bit returns Low.

5.3 Erase/Blank Check status bit (SR5)

The Erase/Blank Check status bit is used to identify if there was an error during a Block Erase operation. When the Erase/Blank Check status bit is High (set to ‘1’), the Program/Erase Controller has applied the maximum number of pulses to the block and still failed to verify that it has erased correctly.

The Erase/Blank Check status bit should be read once the Program/Erase Controller status bit is High (Program/Erase Controller inactive).

The Erase/Blank Check status bit is also used to indicate whether an error occurred during the Blank Check operation. If the data at one or more locations in the block where the Blank Check command has been issued is different from FFFFh, SR5 is set to '1'.

Once set High, the Erase/Blank Check status bit must be set Low by a Clear Status Register command or a hardware reset before a new erase command is issued, otherwise the new command appears to fail.

5.4 Program status bit (SR4)

The Program status bit is used to identify if there was an error during a Program operation.

The Program status bit should be read once the Program/Erase Controller status bit is High (Program/Erase Controller inactive).

When the Program status bit is High (set to ‘1’), the Program/Erase Controller has applied the maximum number of pulses to the word and still failed to verify that it has programmed correctly. Attempting to program a '1' to an already programmed bit while $V_{PP} = V_{PPH}$ also sets the Program status bit High. If V_{PP} is different from V_{PPH}, SR4 remains Low (set to '0') and the attempt is not shown.

Once set High, the Program status bit must be set Low by a Clear Status Register command or a hardware reset before a new Program command is issued, otherwise the new command appears to fail.

5.5 V_{PP} status bit (SR3)

The V_{PP} status bit is used to identify an invalid voltage on the V_{PP} pin during Program and Erase operations. The V_{PP} pin is only sampled at the beginning of a Program or Erase operation. Program and Erase operations are not guaranteed if V_{PP} becomes invalid during an operation.

When the V_{PP} status bit is Low (set to ‘0’), the voltage on the V_{PP} pin was sampled at a valid voltage.

When the V_{PP} status bit is High (set to ‘1’), the V_{PP} pin has a voltage that is below the V_{PP} Lockout Voltage, V_{PPLK}. This means the memory is protected and Program and Erase operations cannot be performed.

Once set High, the V_{PP} status bit must be set Low by a Clear Status Register command or a hardware reset before a new Program or Erase command is issued, otherwise the new command appears to fail.

5.6 Program Suspend status bit (SR2)

The Program Suspend status bit indicates that a Program operation has been suspended. The Program Suspend status bit should only be considered valid when the Program/Erase Controller status bit is High (Program/Erase Controller inactive).

When the Program Suspend status bit is High (set to ‘1’), a Program/Erase Suspend command has been issued and the memory is waiting for a Program/Erase Resume command.

SR2 is set within the Program Suspend latency time of the Program/Erase Suspend command being issued; therefore,the memory may still complete the operation rather than entering Suspend mode.

When a Program/Erase Resume command is issued, the Program Suspend status bit returns Low.

5.7 Block Protection status bit (SR1)

The Block Protection status bit is used to identify if a Program or Block Erase operation has tried to modify the contents of a protected block.

When the Block Protection status bit is High (set to ‘1’), a Program or Erase operation has been attempted on a protected block.

Once set High, the Block Protection status bit must be set Low by a Clear Status Register command or a hardware reset before a new Program or Erase command is issued, otherwise the new command appears to fail.

5.8 Bank Write/Multiple Word Program status bit (SR0)

The Bank Write status bit indicates whether the addressed bank is programming or erasing. In Buffer Enhanced Factory Program mode the Multiple Word Program bit shows if the device is ready to accept a new word to be programmed to the memory array.

The Bank Write status bit should only be considered valid when the Program/Erase Controller Status SR7 is Low (set to ‘0’).

When both the Program/Erase Controller status bit and the Bank Write status bit are Low (set to ‘0’), the addressed bank is executing a Program or Erase operation. When the Program/Erase Controller status bit is Low (set to ‘0’) and the Bank Write status bit is High (set to ‘1’), a program or erase operation is being executed in a bank other than the one being addressed.

In Buffer Enhanced Factory Program mode, if Multiple Word Program status bit is Low (set to ‘0’), the device is ready for the next word. If the Multiple Word Program status bit is High (set to ‘1’), the device is not ready for the next word.

For further details on how to use the Status Register, see the Flowcharts and Pseudocodes provided in *Appendix C*.

Table 9. Status Register bits

<table>
<tr><th>Bit</th><th>Name</th><th>Type</th><th>Logic Level(1)</th><th colspan="2">Definition</th></tr>
<tr><td rowspan="2">SR7</td><td rowspan="2">P/E.C. Status</td><td rowspan="2">Status</td><td>'1'</td><td colspan="2">Ready</td></tr>
<tr><td>'0'</td><td colspan="2">Busy</td></tr>
<tr><td rowspan="2">SR6</td><td rowspan="2">Erase Suspend Status</td><td rowspan="2">Status</td><td>'1'</td><td colspan="2">Erase suspended</td></tr>
<tr><td>'0'</td><td colspan="2">Erase In progress or completed</td></tr>
<tr><td rowspan="2">SR5</td><td rowspan="2">Erase/Blank Check Status</td><td rowspan="2">Error</td><td>'1'</td><td colspan="2">Erase/Blank Check error</td></tr>
<tr><td>'0'</td><td colspan="2">Erase/Blank Check success</td></tr>
<tr><td rowspan="2">SR4</td><td rowspan="2">Program Status</td><td rowspan="2">Error</td><td>'1'</td><td colspan="2">Program error</td></tr>
<tr><td>'0'</td><td colspan="2">Program success</td></tr>
<tr><td rowspan="2">SR3</td><td rowspan="2">V_{PP} Status</td><td rowspan="2">Error</td><td>'1'</td><td colspan="2">V_{PP} invalid, abort</td></tr>
<tr><td>'0'</td><td colspan="2">V_{PP} OK</td></tr>
<tr><td rowspan="2">SR2</td><td rowspan="2">Program Suspend Status</td><td rowspan="2">Status</td><td>'1'</td><td colspan="2">Program suspended</td></tr>
<tr><td>'0'</td><td colspan="2">Program in progress or completed</td></tr>
<tr><td rowspan="2">SR1</td><td rowspan="2">Block Protection Status</td><td rowspan="2">Error</td><td>'1'</td><td colspan="2">Program/Erase on protected block, abort</td></tr>
<tr><td>'0'</td><td colspan="2">No operation to protected blocks</td></tr>
<tr><td rowspan="8">SR0</td><td rowspan="4">Bank Write Status</td><td rowspan="4">Status</td><td rowspan="2">'1'</td><td>SR7 = ‘1’</td><td>Not allowed</td></tr>
<tr><td>SR7 = ‘0’</td><td>Program or Erase operation in a bank other than the addressed bank</td></tr>
<tr><td rowspan="2">'0'</td><td>SR7 = ‘1’</td><td>No Program or Erase operation in the device</td></tr>
<tr><td>SR7 = ‘0’</td><td>Program or Erase operation in addressed bank</td></tr>
<tr><td rowspan="4">Multiple Word Program Status (Buffer Enhanced Factory Program mode)</td><td rowspan="4">Status</td><td rowspan="2">'1'</td><td>SR7 = ‘1’</td><td>Not allowed</td></tr>
<tr><td>SR7 = ‘0’</td><td>The device is NOT ready for the next buffer loading or is going to exit the BEFP mode</td></tr>
<tr><td rowspan="2">'0'</td><td>SR7 = ‘1’</td><td>The device has exited the BEFP mode</td></tr>
<tr><td>SR7 = ‘0’</td><td>The device is ready for the next Buffer loading</td></tr>
</table>

1. Logic level '1' is High, '0' is Low.

6 Configuration Register

The Configuration Register is used to configure the type of bus access that the memory performs. Refer to *Chapter 7: Read modes* for details on Read operations.

The Configuration Register is set through the Command Interface using the Set Configuration Register command. After a reset or power-up, the device is configured for Asynchronous Read (CR15 = 1). The Configuration Register bits are described in *Table 11* and specify the selection of the burst length, burst type, burst X latency, and the Read operation. Refer to Figures *5* and *6* for examples of synchronous burst configurations.

6.1 Read Select bit (CR15)

The Read Select bit, CR15, is used to switch between Asynchronous and Synchronous Read operations.

When the Read Select bit is set to ’1’, Read operations are asynchronous; when the Read Select bit is set to ’0’, Read operations are synchronous.

Synchronous Burst Read is supported in both parameter and main blocks, and can be performed across banks.

On reset or power-up the Read Select bit is set to ’1’ for asynchronous access.

6.2 X-Latency bits (CR13-CR11)

The X-Latency bits are used during Synchronous Read operations to set the number of clock cycles between the address being latched and the first data becoming available. Refer to *Figure 5: X-latency and data output configuration example*.

For correct operation the X-Latency bits can only assume the values in *Table 11: Configuration Register*.

Table 10 shows how to set the X-Latency parameter, taking into account the speed class of the device and the frequency used to read the Flash memory in Synchronous mode.

Table 10. X-Latency Settings

fmax	$t_K$min	X-Latency min
30 MHz	33 ns	3
40 MHz	25 ns	4
52 MHz	19 ns	5

6.3 Wait Polarity bit (CR10)

The Wait Polarity bit is used to set the polarity of the Wait signal used in Synchronous Burst Read mode. During Synchronous Burst Read mode the Wait signal indicates whether the data output is valid or a WAIT state must be inserted.

When the Wait Polarity bit is set to ‘0’ the Wait signal is active Low. When the Wait Polarity bit is set to ‘1’ the Wait signal is active High.

6.4 Data Output Configuration bit (CR9)

The Data Output Configuration bit is used to configure the output to remain valid for either one or two clock cycles during Synchronous mode.

When the Data Output Configuration bit is ’0’ the output data is valid for one clock cycle; when the Data Output Configuration bit is ’1’ the output data is valid for two clock cycles.

The Data Output Configuration bit must be configured using the following condition:

- $t_K > t_{KQV} + t_{QVK\_CPU}$

where:

- t_K is the clock period
- t_{QVK_CPU} is the data setup time required by the system CPU
- t_{KQV} is the clock to data valid time.

If this condition is not satisfied, the Data Output Configuration bit should be set to ‘1’ (two clock cycles). Refer to *Figure 5: X-latency and data output configuration example*.

6.5 Wait Configuration bit (CR8)

The Wait Configuration bit is used to control the timing of the Wait output pin, WAIT, in Synchronous Burst Read mode.

When WAIT is asserted, Data is Not Valid and when WAIT is de-asserted, Data is Valid.

When the Wait Configuration bit is Low (set to ’0’), the Wait output pin is asserted during the WAIT state. When the Wait Configuration bit is High (set to ’1’), the Wait output pin is asserted one data cycle before the WAIT state.

6.6 Burst Type bit (CR7)

The Burst Type bit determines the sequence of addresses read during Synchronous Burst Reads.

The Burst Type bit is High (set to ’1’) because the memory only outputs from sequential addresses.

See *Table 12: Burst type definition* for the sequence of addresses output from a given starting address in Sequential mode.

6.7 Valid Clock Edge bit (CR6)

The Valid Clock Edge bit, CR6, is used to configure the active edge of the Clock, K, during Synchronous Read operations. When the Valid Clock Edge bit is Low (set to ’0’) the falling edge of the Clock is the active edge. When the Valid Clock Edge bit is High (set to ’1’) the rising edge of the Clock is the active edge.

6.8 Wrap Burst bit (CR3)

The Wrap Burst bit, CR3, is used to select between wrap and no wrap. Synchronous Burst reads can be confined inside the 4, 8 or 16 word boundary (wrap) or overcome the boundary (no wrap).

When the Wrap Burst bit is Low (set to ‘0’) the Burst Read wraps. When it is High (set to ‘1’) the Burst Read does not wrap.

6.9 Burst length bits (CR2-CR0)

The Burst Length bits are used to set the number of words to be output during a Synchronous Burst Read operation as result of a single address latch cycle.

They can be set for 4 words, 8 words, 16 words or continuous burst, where all the words are read sequentially. In Continuous Burst mode the burst sequence can cross bank boundaries.

In continuous burst mode, in 4, 8 or 16 words no-wrap, depending on the starting address, the device asserts the WAIT signal to indicate that a delay is necessary before the data is output.

If the starting address is shifted by 1, 2 or 3 positions from the four-word boundary, WAIT is asserted for 1, 2 or 3 clock cycles, respectively, when the burst sequence crosses the first 16-word boundary. This indicates that the device needs an internal delay to read the successive words in the array. WAIT is only asserted once during a continuous burst access. See also *Table 12: Burst type definition*.

CR14, CR5 and **CR4** are reserved for future use.

Table 11. Configuration Register

Bit	Description	Value	Description
CR15	Read Select	0	Synchronous Read
		1	Asynchronous Read (Default at power-on)
CR14	Reserved		
CR13-CR11	X-Latency	010	2 clock latency(1)
		011	3 clock latency
		100	4 clock latency
		101	5 clock latency
		110	6 clock latency
		111	7 clock latency (default)
		Other configurations reserved	
CR10	Wait Polarity	0	WAIT is active Low
		1	WAIT is active High (default)
CR9	Data Output Configuration	0	Data held for one clock cycle
		1	Data held for two clock cycles (default)(1)
CR8	Wait Configuration	0	WAIT is active during WAIT state
		1	WAIT is active one data cycle before WAIT state(1) (default)
CR7	Burst Type	0	Reserved
		1	Sequential (default)
CR6	Valid Clock Edge	0	Falling Clock edge
		1	Rising Clock edge (default)
CR5-CR4	Reserved		
CR3	Wrap Burst	0	Wrap
		1	No Wrap (default)
CR2-CR0	Burst Length	001	4 words
		010	8 words
		011	16 words
		111	Continuous (default)

1. The combination X-Latency = 2, Data held for two clock cycles and Wait active one data cycle before the WAIT state is not supported.

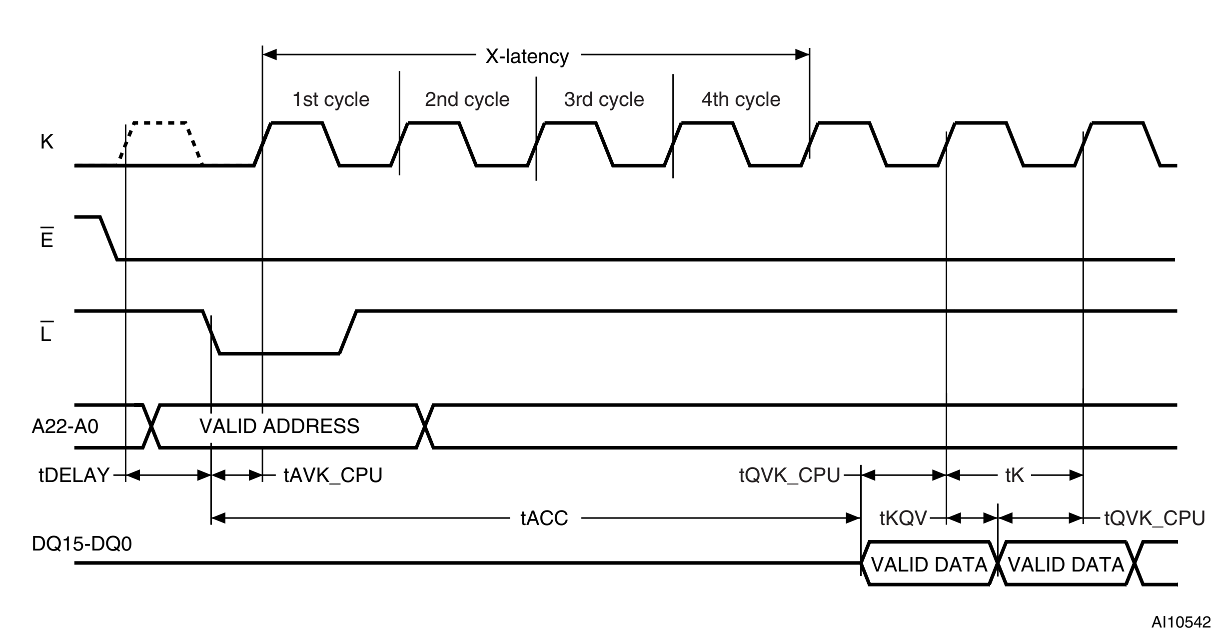
Table 12. Burst type definition

Mode	Start Add	Sequential			Continuous Burst
		4 words	8 words	16 words	
Wrap	0	0-1-2-3	0-1-2-3-4-5-6-7	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	0-1-2-3-4-5-6...
	1	1-2-3-0	1-2-3-4-5-6-7-0	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-0	1-2-3-4-5-6-7-...15-WAIT-16-17-18...
	2	2-3-0-1	2-3-4-5-6-7-0-1	2-3-4-5-6-7-8-9-10-11-12-13-14-15-0-1	2-3-4-5-6-7...15-WAIT-WAIT-16-17-18...
	3	3-0-1-2	3-4-5-6-7-0-1-2	3-4-5-6-7-8-9-10-11-12-13-14-15-0-1-2	3-4-5-6-7...15-WAIT-WAIT-WAIT-16-17-18...
	...				
	7	7-4-5-6	7-0-1-2-3-4-5-6	7-8-9-10-11-12-13-14-15-0-1-2-3-4-5-6	7-8-9-10-11-12-13-14-15-WAIT-WAIT-WAIT-16-17...
	...				
	12				12-13-14-15-16-17-18...
	13				13-14-15-WAIT-16-17-18...
	14				14-15-WAIT-WAIT-16-17-18....
	15				15-WAIT-WAIT-WAIT-16-17-18...

Table 12. Burst type definition (continued)

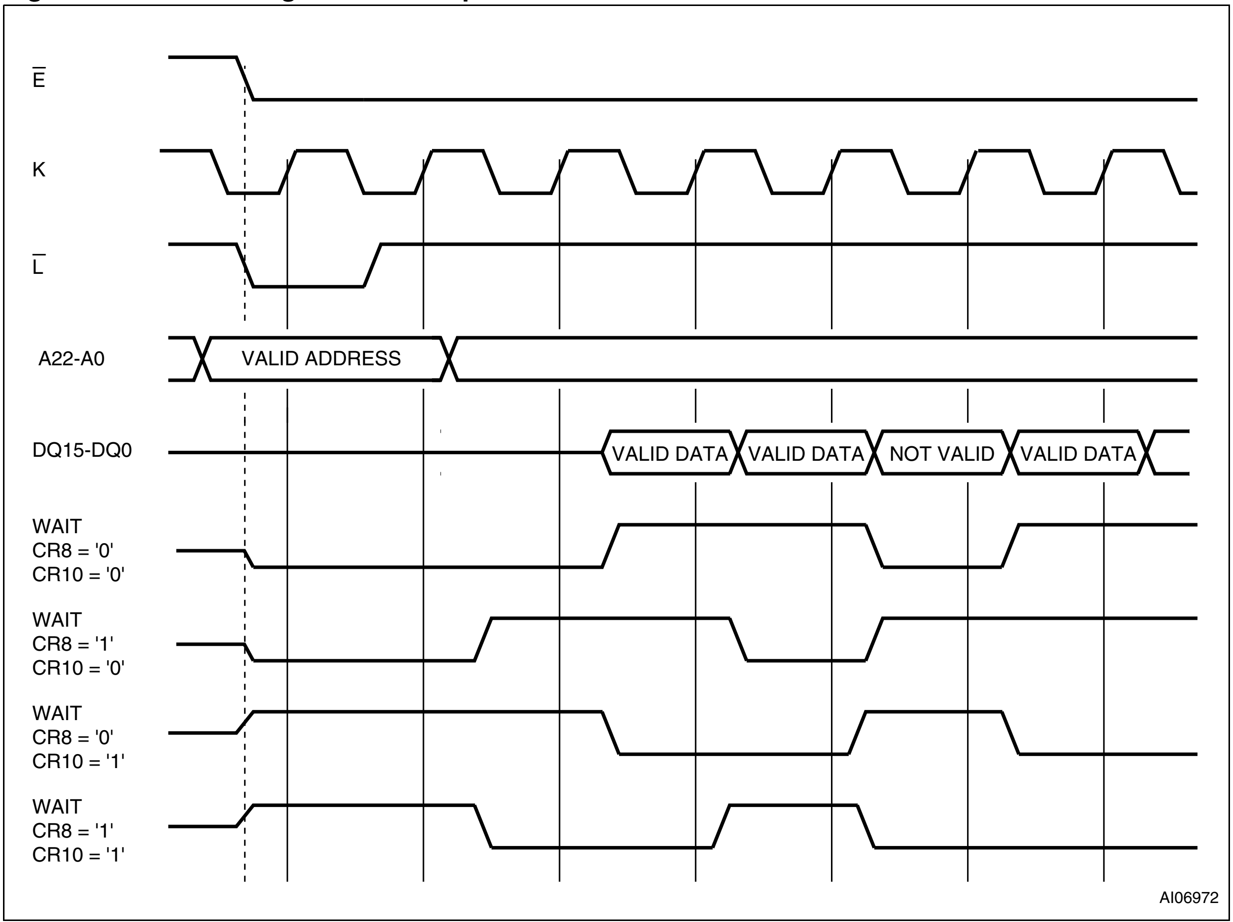
Mode	Start Add	Sequential			Continuous Burst
		4 words	8 words	16 words	
No-wrap	0	0-1-2-3	0-1-2-3-4-5-6-7	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	Same as for wrap (wrap /no wrap has no effect on continuous burst)
	1	1-2-3-4	1-2-3-4-5-6-7-8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-WAIT-16	
	2	2-3-4-5	2-3-4-5-6-7-8-9...	2-3-4-5-6-7-8-9-10-11-12-13-14-15-WAIT-WAIT-16-17	
	3	3-4-5-6	3-4-5-6-7-8-9-10	3-4-5-6-7-8-9-10-11-12-13-14-15-WAIT-WAIT-WAIT-16-17-18	
	...				
	7	7-8-9-10	7-8-9-10-11-12-13-14	7-8-9-10-11-12-13-14-15-WAIT-WAIT-WAIT-16-17-18-19-20-21-22	
	...				
	12	12-13-14-15	12-13-14-15-16-17-18-19	12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27	
	13	13-14-15-WAIT-16	13-14-15-WAIT-16-17-18-19-20	13-14-15-WAIT-16-17-18-19-20-21-22-23-24-25-26-27-28	
	14	14-15-WAIT-WAIT-16-17	14-15-WAIT-WAIT-16-17-18-19-20-21	14-15-WAIT-WAIT-16-17-18-19-20-21-22-23-24-25-26-27-28-29	
	15	15-WAIT-WAIT-WAIT-16-17-18	15-WAIT-WAIT-WAIT-16-17-18-19-20-21-22	15-WAIT-WAIT-WAIT-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30	

Figure 5. X-latency and data output configuration example



1. The settings shown are X-latency = 4, Data Output held for one clock cycle.

Figure 6. Wait configuration example



7 Read modes

Read operations can be performed in two different ways depending on the settings in the Configuration Register. If the clock signal is 'don't care' for the data output, the Read operation is asynchronous. If the data output is synchronized with clock, the Read operation is synchronous.

The Read mode and format of the data output are determined by the Configuration Register. (See *Section 6: Configuration Register* for details). All banks support both asynchronous and Synchronous Read operations.

7.1 Asynchronous Read mode

In Asynchronous Read operations the clock signal is 'don't care'. The device outputs the data corresponding to the address latched; that is the memory array, Status Register, Common Flash Interface, or Electronic Signature depending on the command issued. CR15 in the Configuration Register must be set to '1' for synchronous operations.

Asynchronous Read operations can be performed in two different ways, Asynchronous Random Access Read and Asynchronous Page Read. Only Asynchronous Page Read takes full advantage of the internal page storage so different timings are applied.

In Asynchronous Read mode a page of data is internally read and stored in a page buffer. The page has a size of 4 words and is addressed by address inputs A0 and A1. The first Read operation within the page has a longer access time (t_{AVQV}, random access time). Subsequent reads within the same page have much shorter access times (t_{AVQV1}, page access time). If the page changes, then the normal, longer timings apply again.

The device features an Automatic Standby mode. During Asynchronous Read operations, after a bus inactivity of 150 ns, the device automatically switches to the Automatic Standby mode. In this situation, the power consumption is reduced to the standby value and the outputs are still driven.

In Asynchronous Read mode, the WAIT signal is always de-asserted.

See *Table 22: Asynchronous Read AC characteristics*, *Figure 9: Asynchronous random access Read AC waveforms*, and *Figure 10: Asynchronous Page Read AC waveforms* for details.

7.2 Synchronous Burst Read mode

In Synchronous Burst Read mode the data is output in bursts synchronized with the clock. It is possible to perform Burst reads across bank boundaries.

Synchronous Burst Read mode can only be used to read the memory array. For other Read operations, such as Read Status Register, Read CFI and Read Electronic Signature, then Single Synchronous Read or Asynchronous Random Access Read must be used.

In Synchronous Burst Read mode the flow of the data output depends on parameters that are configured in the Configuration Register.

A burst sequence starts at the first clock edge (rising or falling depending on Valid Clock Edge bit CR6 in the Configuration Register) after the falling edge of Latch Enable or Chip Enable, whichever occurs last. Addresses are internally incremented and data is output on each data cycle after a delay, which depends on the X-latency bits CR13-CR11 of the Configuration Register.

The number of words to be output during a Synchronous Burst Read operation can be configured as 4 words, 8 words, 16 words or Continuous (Burst Length bits CR2-CR0). The data can be configured to remain valid for one or two clock cycles (Data Output Configuration bit CR9).

The order of the data output can be modified through the Wrap Burst bit in the Configuration Register. The burst sequence is sequential and can be confined inside the 4-, 8- or 16-word boundary (wrap) or overcome the boundary (no wrap).

The WAIT signal may be asserted to indicate to the system that an output delay occurs. This delay depends on the starting address of the burst sequence and on the burst configuration.

WAIT is asserted during the X-latency, the WAIT state, and at the end of a 4-, 8- and 16-word burst. It is only de-asserted when output data is valid. In Continuous Burst Read mode a WAIT state occurs when crossing the first 16-word boundary. If the starting address is aligned to the burst length (4-, 8- or 16-words), the wrapped configuration has no impact on the output sequence.

The WAIT signal can be configured to be active Low or active High by setting CR10 in the Configuration Register.

See *Table 23: Synchronous Read AC characteristics* and *Figure 11: Synchronous Burst Read AC waveforms* for details.

7.2.1 Synchronous Burst Read Suspend

A Synchronous Burst Read operation can be suspended, freeing the data bus for other higher priority devices. It can be suspended during the initial access latency time (before data is output) or after the device has output data. When the Synchronous Burst Read operation is suspended, internal array sensing continues and any previously latched internal data is retained. A burst sequence can be suspended and resumed as often as required as long as the operating conditions of the device are met.

A Synchronous Burst Read operation is suspended when Chip Enable, $\overline{E}$, is Low and the current address has been latched (on a Latch Enable rising edge or on a valid clock edge). The Clock signal is then halted at V_{IH} or at V_{IL}, and Output Enable, $\overline{G}$, goes High.

When Output Enable, $\overline{G}$, becomes Low again and the Clock signal restarts, the Synchronous Burst Read operation resumes exactly where it stopped.

WAIT reverts to high-impedance whenever Chip Enable, $\overline{E}$, or Output Enable, $\overline{G}$, goes High.

See *Table 23: Synchronous Read AC characteristics* and *Figure 13: Synchronous Burst Read Suspend AC waveforms* for details.

7.3 Single Synchronous Read mode

Single Synchronous Read operations are similar to Synchronous Burst Read operations except that the memory outputs the same data to the end of the operation.

Synchronous Single Reads are used to read the Electronic Signature, Status Register, CFI, Block Protection Status, Configuration Register Status or Protection Register. When the addressed bank is in Read CFI, Read Status Register, or Read Electronic Signature mode, the WAIT signal is asserted during the X-latency, the WAIT state, and at the end of a 4-, 8- and 16-word burst. It is only de-asserted when output data are valid.

See *Table 23: Synchronous Read AC characteristics* and *Figure 12: Single Synchronous Read AC waveforms* for details.

8 Dual operations and multiple bank architecture

The multiple bank architecture of the M58LT128HST/B gives greater flexibility for software developers to split the code and data spaces within the memory array. The dual operations feature simplifies the software management of the device by allowing code to be executed from one bank while another bank is being programmed or erased.

The dual operations feature means that while programming or erasing in one bank, read operations are possible in another bank with zero latency (only one bank at a time is allowed to be in Program or Erase mode).

If a read operation is required in a bank, which is programming or erasing, the program or erase operation can be suspended. Also if the suspended operation was Erase then a Program command can be issued to another block. This means the device can have one block in Erase Suspend mode, one programming, and other banks in Read mode.

Bus Read operations are allowed in another bank between setup and confirm cycles of Program or Erase operations.

By using a combination of these features, Read operations are possible at any moment in the M58LT128HST/B device.

Dual operations between the parameter bank and either of the CFI, the OTP or the electronic signature memory space are not allowed. *Table 15* shows which dual operations are allowed or not between the CFI, the OTP, the electronic signature locations, and the memory array.

Table 13 and *Table 14* show the dual operations possible in other banks and in the same bank.

Table 13. Dual operations allowed in other banks

Status of bank	Commands allowed in another bank							
	Read Array	Read Status Register	Read CFI Query	Read Electronic Signature	Program, Buffer Program	Block Erase	Program /Erase Suspend	Program /Erase Resume
Idle	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Programming	Yes	Yes	Yes	Yes	–	–	Yes	–
Erasing	Yes	Yes	Yes	Yes	–	–	Yes	–
Program suspended	Yes	Yes	Yes	Yes	–	–	–	Yes
Erase suspended	Yes	Yes	Yes	Yes	Yes	–	–	Yes

Table 14. Dual operations allowed in same bank

Status of bank	Commands allowed in same bank							
	Read Array	Read Status Register	Read CFI Query	Read Electronic Signature	Program, Buffer Program	Block Erase	Program /Erase Suspend	Program /Erase Resume
Idle	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Programming	–(1)	Yes	Yes	Yes	–	–	Yes	–
Erasing	–(1)	Yes	Yes	Yes	–	–	Yes	–
Program suspended	Yes(2)	Yes	Yes	Yes	–	–	–	Yes
Erase suspended	Yes(2)	Yes	Yes	Yes	Yes(3)	–	–	Yes

1. The Read Array command is accepted but the data output is not guaranteed until the Program or Erase has completed.
2. The Read Array command is accepted but the data output is not guaranteed in the block that is being erased, or in the word that is being programmed.
3. Not allowed in the block that is being erased or in the word that is being programmed.

Table 15. Dual operation limitations

Current status		Commands allowed			
		Read CFI/OTP/ electronic signature	Read parameter blocks	Read main blocks	
				Located in parameter bank	Not located in parameter bank
Programming/erasing parameter blocks		No	No	No	Yes
Programming/ erasing main blocks	Located in parameter bank	Yes	No	No	Yes
	Not located in parameter bank	Yes	Yes	Yes	In different bank only
Programming OTP		No	No	No	No

9 Block protection

The M58LT128HST/B features an instant, individual block protection scheme that allows any block to be protected or unprotected with no latency. This protection scheme has two levels of protection.

- Protect/unprotect: this first level allows software only control of block protection.
- $V_{PP} \leq V_{PPLK}$: the second level offers a complete hardware protection against Program and Erase operations on all blocks.

The protection status of each block can be set to protected and unprotected. *Appendix C*, *Figure 25* shows a flowchart for the protection operations.

9.1 Protection status

The protection status of every block can be read in the Read Electronic Signature mode of the device. To enter this mode, issue the Read Electronic Signature command. Subsequent reads at the address specified in *Table 7* output the protection status of that block.

The protection status is represented by DQ0. DQ0 indicates the block protect/unprotect status. It is set by the Protect command and cleared by the Unprotect command.

The following sections explain the operation of the protection system.

9.2 Protected state

The default state of all blocks on power-up or after a hardware reset is protected (state = 1). Protected blocks are fully protected from Program or Erase operations. Any Program or Erase operations attempted on a protected block return an error in the Status Register. The state of a protected block can be changed to unprotected using the appropriate software commands. An unprotected block can be protected by issuing the Protect command.

9.3 Unprotected state

Unprotected blocks (state = 0) can be programmed or erased. All unprotected blocks return to the protected state after a hardware reset or when the device is powered-down. The state of an unprotected block can be changed to protected using the appropriate software commands. A protected block can be unprotected by issuing the Unprotect command.

9.4 Protection operations during Erase Suspend

Changes to the block protection state can be made during an Erase Suspend by using the standard protection command sequences to unprotect or protect a block. This is useful in the case where another block needs to be updated while an Erase operation is in progress.

To change block protection during an Erase operation, first write the Erase Suspend command, then check the Status Register until it indicates that the Erase operation has been suspended. Next, write the desired Protect command sequence to a block and the protection status changes. After completing any desired Protect, Read, or Program operations, resume the Erase operation with the Erase Resume command.

If a block is protected during an Erase Suspend of the same block, the Erase operation completes when the erase is resumed. Protection operations cannot be performed during a Program Suspend operation.

10 Program and erase times and endurance cycles

The program and erase times and the number of program/erase cycles per block are shown in *Table 16*. Exact erase times may change depending on the memory array condition. The best case is when all the bits in the block are at ‘0’ (preprogrammed). The worst case is when all the bits in the block are at ‘1’ (not preprogrammed). Usually, the system overhead is negligible with respect to the erase time. In the M58LT128HST/B the maximum number of program/erase cycles depends on the V_{PP} voltage supply used.

Table 16. Program/erase times and endurance cycles(1) (2)

	Parameter	Condition		Min	Typ	Typical after 100kW/E Cycles	Max	Unit
$V_{PP} = V_{DD}$	Erase	Parameter block (16 Kword)			0.4	1	2.5	s
		Main block (64 Kword)	Preprogrammed		1.2	3	4	s
			Not preprogrammed		1.5		4	s
	Program(3)	Single word	Word program		12		180	µs
			Buffer program		12		180	µs
		Buffer (32 words) (Buffer Program)			384			µs
		Main block (64 Kword)			768			ms
	Suspend latency	Program			5		10	µs
		Erase			5		20	µs
	Program/erase cycles (per block)	Main blocks		100,000				cycles
		Parameter blocks		100,000				cycles

Table 16. Program/erase times and endurance cycles[1] [2] (continued)

Parameter		Condition		Min	Typ	Typical after 100kW/E Cycles	Max	Unit
$V_{PP} = V_{PPH}$	Erase	Parameter block (16 Kword)			0.4		2.5	s
		Main block (64 Kword)			1		4	s
	Program[3]	Single word	Word program		10		170	µs
			Buffer enhanced factory program[4]		2.5			µs
		Buffer (32 words)	Buffer program		80			µs
			Buffer enhanced factory program		80			µs
		Main Block (64 Kwords)	Buffer program		160			ms
			Buffer enhanced factory program		160			ms
		Bank (8 Mbits)	Buffer program		1.28			s
			Buffer enhanced factory program		1.28			s
	Program/erase cycles (per block)	Main blocks					1000	cycles
		Parameter blocks					2500	cycles
	Blank check	Main blocks			16			ms
		Parameter blocks			4			ms

1. T_A = –40 to 85°C; V_{DD} = 1.7 V to 2 V; V_{DDQ} = 2.7 V to 3.6 V.
2. Values are liable to change with the external system-level overhead (command sequence and Status Register polling execution).
3. Excludes the time needed to execute the command sequence.
4. This is an average value on the entire device.

11 Maximum rating

Stressing the device above the rating listed in the absolute maximum ratings table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the operating sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Refer to the Numonyx SURE Program and other relevant quality documents.

Table 17. Absolute maximum ratings

Symbol	Parameter	Value		Unit
		Min	Max	
T_A	Ambient operating temperature	–40	85	°C
T_{BIAS}	Temperature under bias	–40	85	°C
T_{STG}	Storage temperature	–65	125	°C
V_{IO}	Input or output voltage	–0.5	3.8	V
V_{DD}	Supply voltage	–0.2	2.5	V
V_{DDQ}	Input/output supply voltage	–0.2	4.2	V
V_{PP}	Program voltage	–0.2	10	V
I_O	Output short circuit current		100	mA
t_{VPPH}	Time for V_{PP} at V_{PPH}		100	hours

12 DC and AC parameters

This section summarizes the operating measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristics tables that follow are derived from tests performed under the measurement conditions summarized in *Table 18: Operating and AC measurement conditions*. Designers should check that the operating conditions in their circuit match the operating conditions when relying on the quoted parameters.

Table 18. Operating and AC measurement conditions

Parameter	M58LT128HST/B 85 Min	M58LT128HST/B 85 Max	Units
V_{DD} supply voltage	1.7	2.0	V
V_{DDQ} supply voltage	2.7	3.6	V
V_{PP} supply voltage (factory environment)	8.5	9.5	V
V_{PP} supply voltage (application environment)	–0.4	V_{DDQ}+0.4	V
Ambient operating temperature	–40	85	°C
Load capacitance (C_L)	30		pF
Input rise and fall times		5	ns
Input pulse voltages	0 to V_{DDQ}		V
Input and output timing ref. voltages	$V_{DDQ}/2$		V

Figure 7. AC measurement I/O waveform

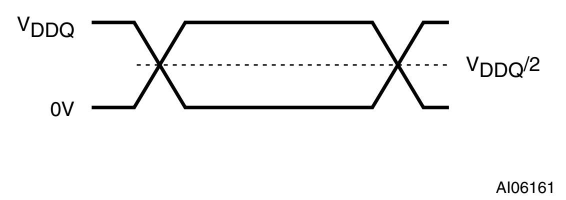


Figure 8. AC measurement load circuit

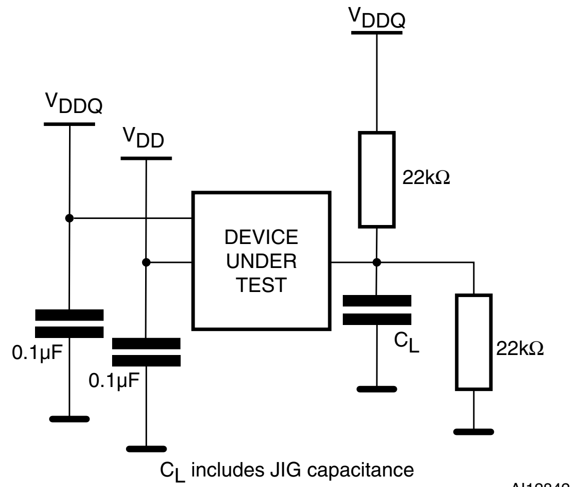


Table 19. Capacitance(1)

Symbol	Parameter	Test Condition	Min	Max	Unit
C_{IN}	Input capacitance	$V_{IN} = 0V$	6	8	pF
C_{OUT}	Output capacitance	$V_{OUT} = 0V$	8	12	pF

1. Sampled only, not 100% tested.

Table 20. DC characteristics - currents

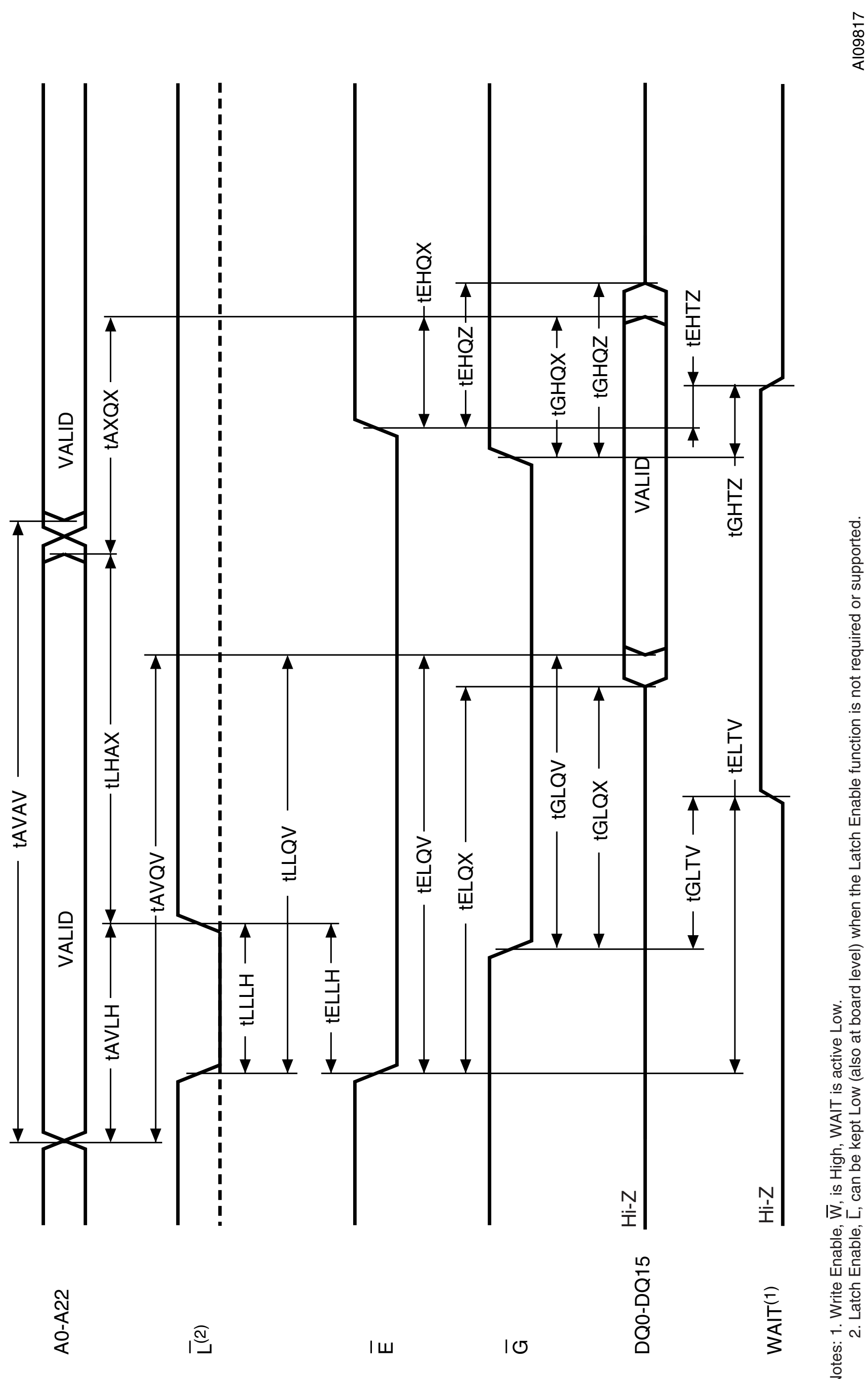
Symbol	Parameter	Test Condition	Typ	Max	Unit
I_{LI}	Input leakage current	$0V \leq V_{IN} \leq V_{DDQ}$		±1	µA
I_{LO}	Output leakage current	$0V \leq V_{OUT} \leq V_{DDQ}$		±1	µA
I_{DD1}	Supply current Asynchronous Read (f=5 MHz)	$\overline{E} = V_{IL}$, $\overline{G} = V_{IH}$	14	16	mA
	Supply current Synchronous Read (f = 40 MHz)	4 word	13	17	mA
		8 word	15	19	mA
		16 word	17	21	mA
		Continuous	21	26	mA
	Supply current Synchronous Read (f = 52 MHz)	4 word	16	19	mA
		8 word	19	23	mA
		16 word	22	26	mA
		Continuous	23	28	mA
I_{DD2}	Supply current (Reset)	$\overline{RP} = V_{SS} \pm 0.2V$	25	75	µA
I_{DD3}	Supply current (Standby)	$\overline{E} = V_{DDQ} \pm 0.2V$ $K=V_{SS}$	25	75	µA
I_{DD4}	Supply current (Automatic Standby)	$\overline{E} = V_{IL}$, $\overline{G} = V_{IH}$	25	75	µA
I_{DD5}(1)	Supply current (Program)	$V_{PP} = V_{PPH}$	8	20	mA
		$V_{PP} = V_{DD}$	10	25	mA
	Supply current (Erase)	$V_{PP} = V_{PPH}$	8	20	mA
		$V_{PP} = V_{DD}$	10	25	mA
I_{DD6}(1),(2)	Supply current (Dual operations)	Program/erase in one bank, Asynchronous Read in another bank	24	41	mA
		Program/erase in one bank, Synchronous Read (continuous f=52 MHz) in another bank	33	53	mA
I_{DD7}(1)	Supply current Program/Erase Suspended (standby)	$\overline{E} = V_{DDQ} \pm 0.2V$ $K=V_{SS}$	25	75	µA
I_{PP1}(1)	V_{PP} supply current (Program)	$V_{PP} = V_{PPH}$	2	5	mA
		$V_{PP} = V_{DD}$	0.2	5	µA
	V_{PP} supply current (Erase)	$V_{PP} = V_{PPH}$	2	5	mA
		$V_{PP} = V_{DD}$	0.2	5	µA
I_{PP2}	V_{PP} supply current (Read)	$V_{PP} \leq V_{DD}$	0.2	5	µA
I_{PP3}(1)	V_{PP} supply current (Standby)	$V_{PP} \leq V_{DD}$	0.2	5	µA

1. Sampled only, not 100% tested.
2. V_{DD} Dual Operation current is the sum of Read and Program or Erase currents.

Table 21. DC characteristics - voltages

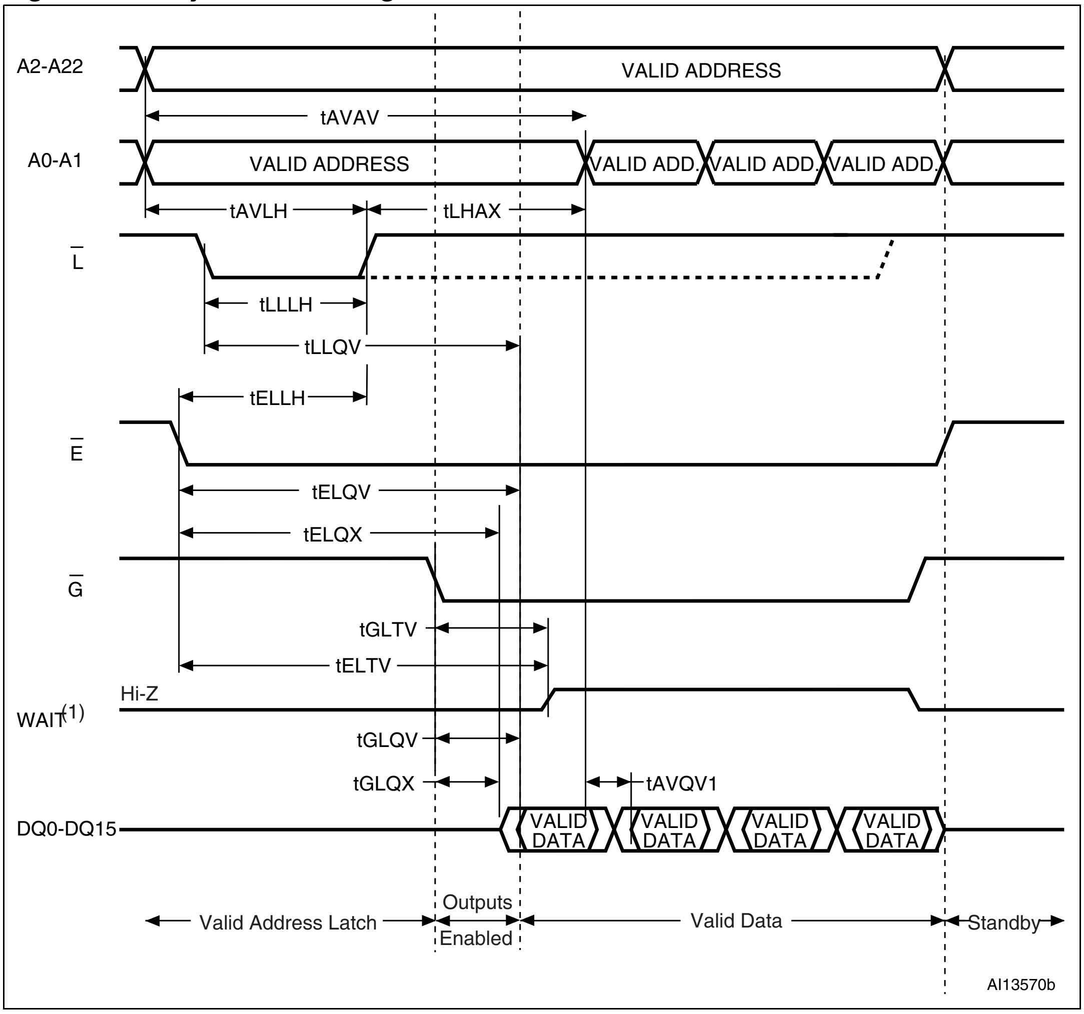
Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
V_{IL}	Input low voltage		0		0.4	V
V_{IH}	Input high voltage		V_{DDQ} –0.4		V_{DDQ} + 0.4	V
V_{OL}	Output low voltage	I_{OL} = 100µA			0.1	V
V_{OH}	Output high voltage	I_{OH} = –100µA	V_{DDQ} –0.1			V
V_{PP1}	V_{PP} program voltage-logic	Program, Erase	1.3	3	3.6	V
V_{PPH}	V_{PP} program voltage factory	Program, Erase	8.5	9.0	9.5	V
V_{PPLK}	Program or Erase lockout				0.4	V
V_{LKO}	V_{DD} lock voltage				1	V

Figure 9. Asynchronous random access Read AC waveforms



Notes: 1. Write Enable, $\overline{W}$, is High, WAIT is active Low.
2. Latch Enable, $\overline{L}$, can be kept Low (also at board level) when the Latch Enable function is not required or supported.

Figure 10. Asynchronous Page Read AC waveforms



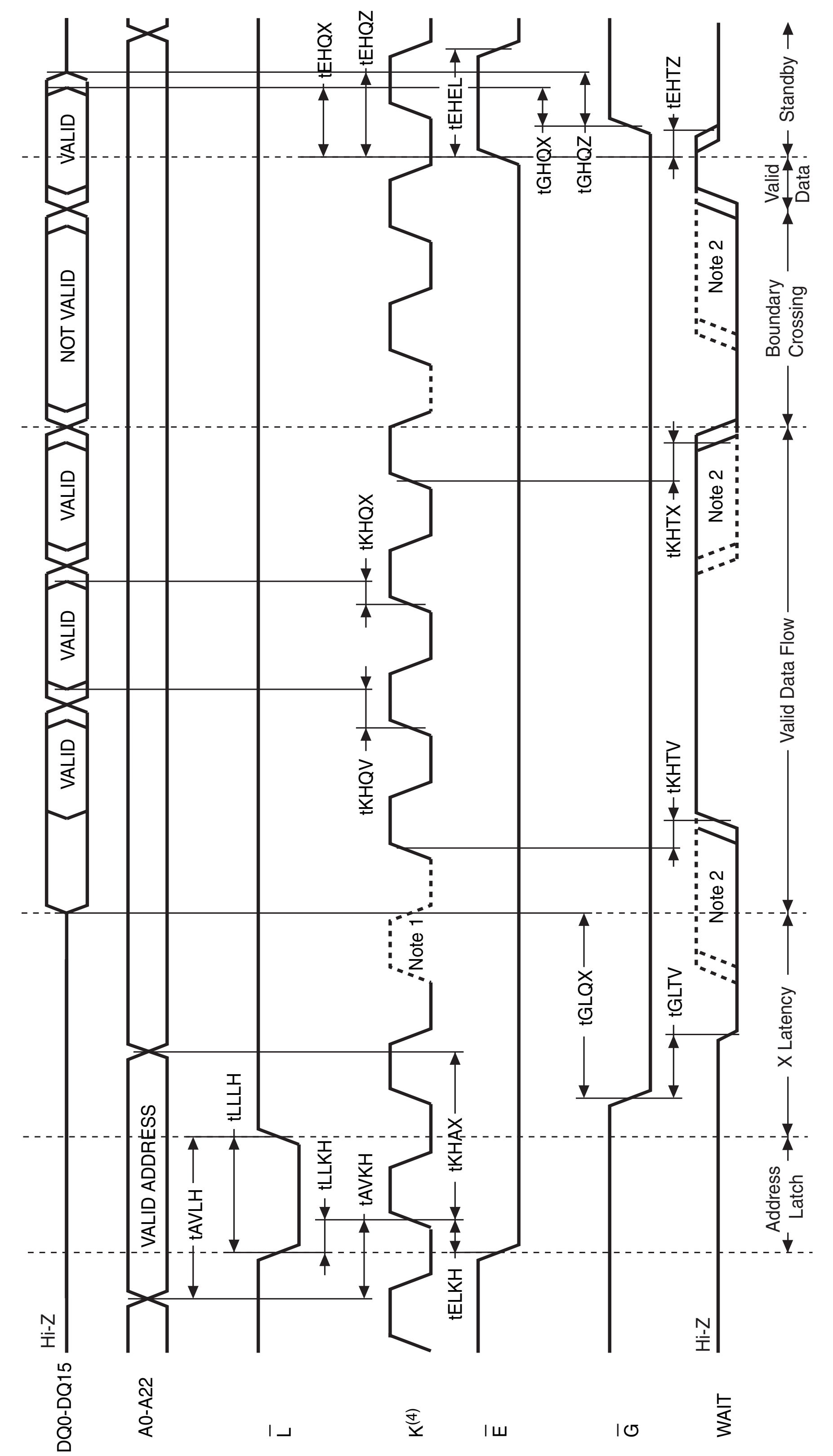
1. WAIT is active Low.

Table 22. Asynchronous Read AC characteristics

	Symbol	Alt	Parameter		M58LT128HST/B 85	Unit
Read Timings	t_{AVAV}	t_{RC}	Address Valid to Next Address Valid	Min	85	ns
	t_{AVQV}	t_{ACC}	Address Valid to Output Valid (Random)	Max	85	ns
	t_{AVQV1}	t_{PAGE}	Address Valid to Output Valid (Page)	Max	25	ns
	t_{AXQX}(1)	t_{OH}	Address Transition to Output Transition	Min	0	ns
	t_{ELTV}		Chip Enable Low to Wait Valid	Max	17	ns
	t_{ELQV}(2)	t_{CE}	Chip Enable Low to Output Valid	Max	85	ns
	t_{ELQX}(1)	t_{LZ}	Chip Enable Low to Output Transition	Min	0	ns
	t_{EHTZ}		Chip Enable High to Wait Hi-Z	Max	17	ns
	t_{EHQX}(1)	t_{OH}	Chip Enable High to Output Transition	Min	0	ns
	t_{EHQZ}(1)	t_{HZ}	Chip Enable High to Output Hi-Z	Max	17	ns
	t_{GLQV}(2)	t_{OE}	Output Enable Low to Output Valid	Max	25	ns
	t_{GLQX}(1)	t_{OLZ}	Output Enable Low to Output Transition	Min	0	ns
	t_{GLTV}		Output Enable Low to Wait Valid	Max	17	ns
	t_{GHQX}(1)	t_{OH}	Output Enable High to Output Transition	Min	0	ns
	t_{GHQZ}(1)	t_{DF}	Output Enable High to Output Hi-Z	Max	17	ns
	t_{GHTZ}		Output Enable High to Wait Hi-Z	Max	17	ns
Latch Timings	t_{AVLH}	t_{AVADVH}	Address Valid to Latch Enable High	Min	10	ns
	t_{ELLH}	t_{ELADVH}	Chip Enable Low to Latch Enable High	Min	10	ns
	t_{LHAX}	t_{ADVHAX}	Latch Enable High to Address Transition	Min	9	ns
	t_{LLLH}	$t_{ADVLADVH}$	Latch Enable Pulse Width	Min	10	ns
	t_{LLQV}	t_{ADVLQV}	Latch Enable Low to Output Valid (Random)	Max	85	ns

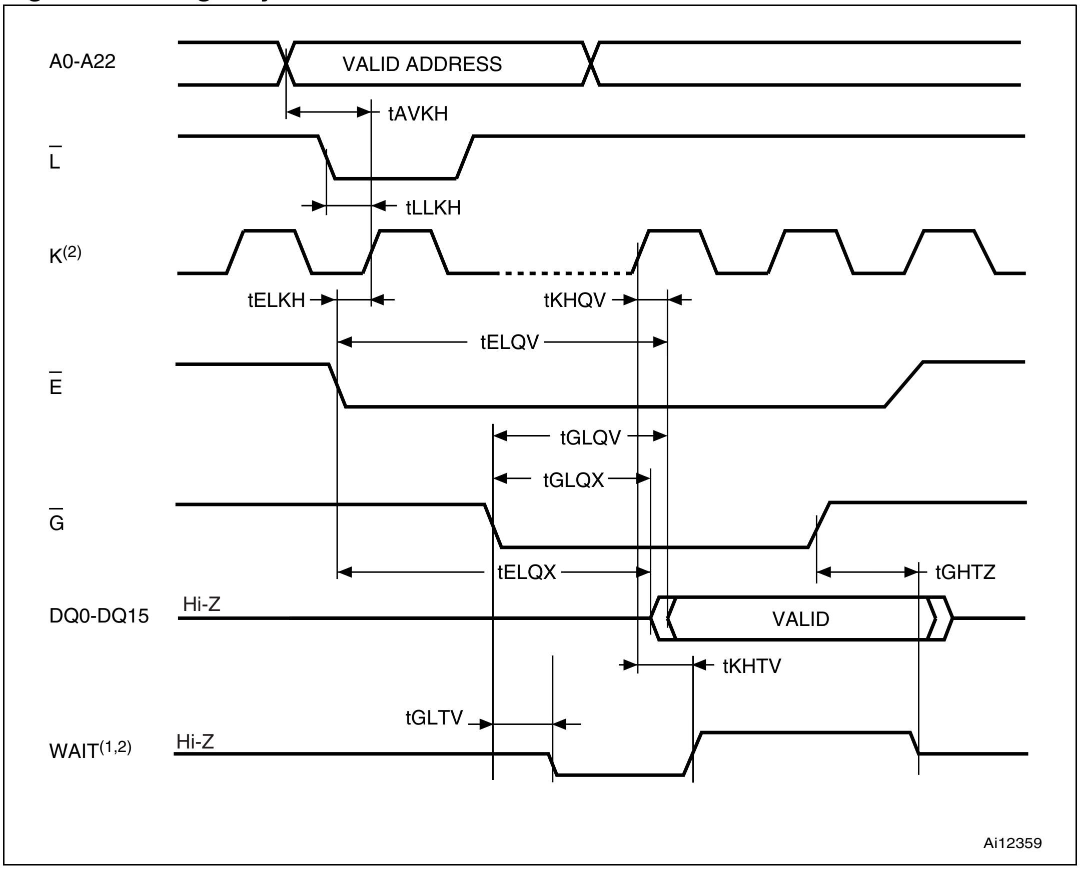
1. Sampled only, not 100% tested.
2. $\overline{G}$ may be delayed by up to t_{ELQV} - t_{GLQV} after the falling edge of $\overline{E}$ without increasing t_{ELQV}.

Figure 11. Synchronous Burst Read AC waveforms



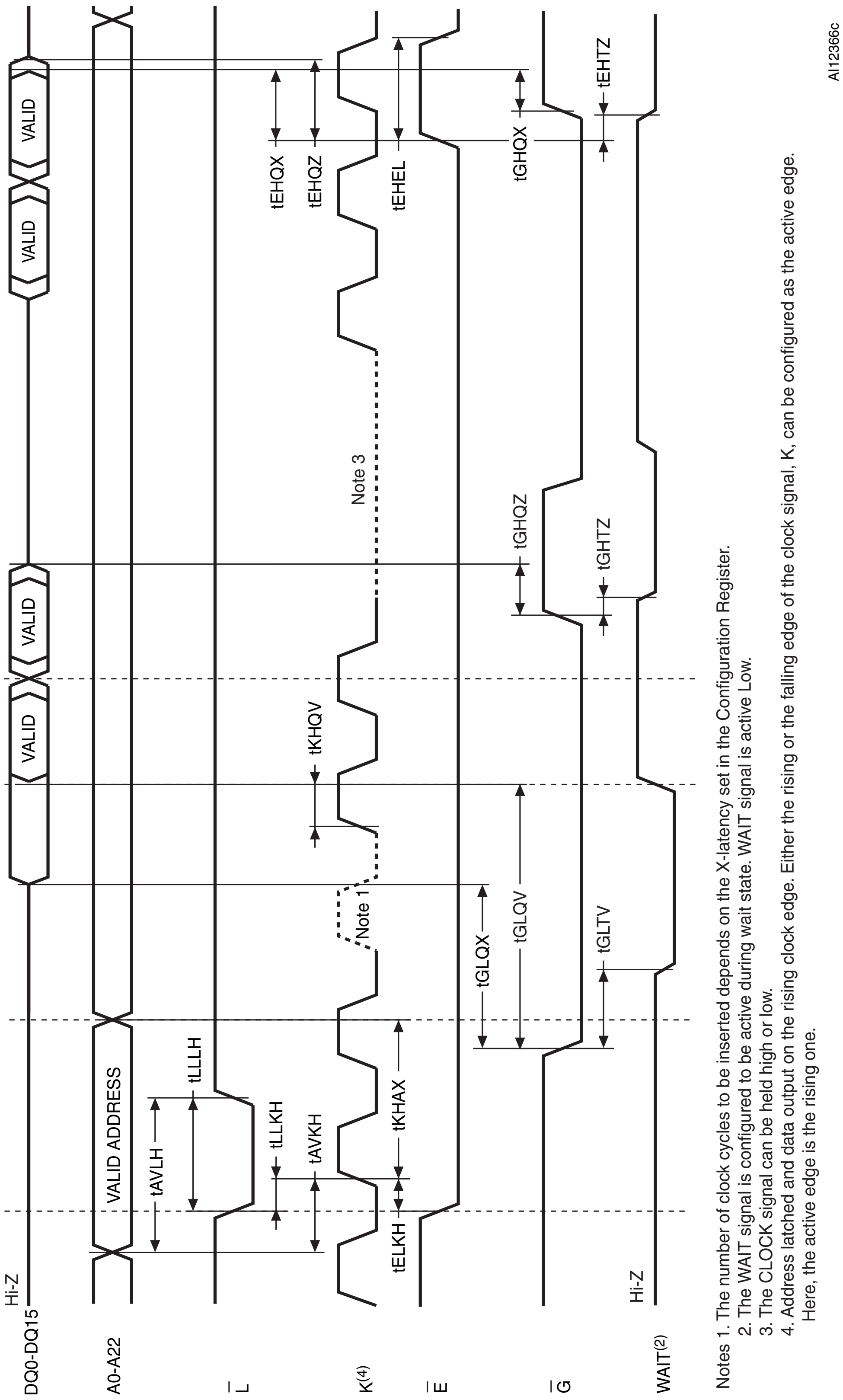
Note 1. The number of clock cycles to be inserted depends on the X-latency set in the Burst Configuration Register.
2. The WAIT signal can be configured to be active during wait state or one cycle before. WAIT signal is active Low.
3. Address latched and data output on the rising clock edge.
4. Either the rising or the falling edge of the clock signal, K, can be configured as the active edge. Here, the active edge of K is the rising one.

Figure 12. Single Synchronous Read AC waveforms



1. The WAIT signal is configured to be active during wait state. WAIT signal is active Low.
2. Address latched and data output on the rising clock edge. Either the rising or the falling edge of the clock signal, K, can be configured as the active edge. Here, the active edge is the rising one.

Figure 13. Synchronous Burst Read Suspend AC waveforms



Notes 1. The number of clock cycles to be inserted depends on the X-latency set in the Configuration Register.
2. The WAIT signal is configured to be active during wait state. WAIT signal is active Low.
3. The CLOCK signal can be held high or low.
4. Address latched and data output on the rising clock edge. Either the rising or the falling edge of the clock signal, K, can be configured as the active edge. Here, the active edge is the rising one.

Figure 14. Clock input AC waveform

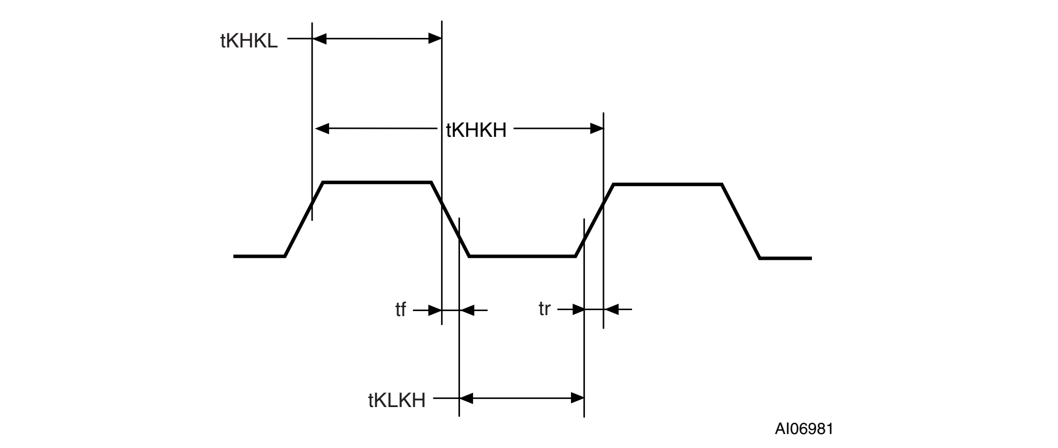


Table 23. Synchronous Read AC characteristics(1) (2)

Symbol		Alt	Parameter		M58LT128HST/B 85	Unit
Synchronous Read Timings	t_{AVKH}	t_{AVCLKH}	Address Valid to Clock High	Min	9	ns
	t_{ELKH}	t_{ELCLKH}	Chip Enable Low to Clock High	Min	9	ns
	t_{EHEL}		Chip Enable Pulse Width (subsequent synchronous reads)	Min	20	ns
	t_{EHTZ}		Chip Enable High to Wait Hi-Z	Max	17	ns
	t_{KHAX}	t_{CLKHAX}	Clock High to Address Transition	Min	10	ns
	t_{KHQV} t_{KHTV}	t_{CLKHQV}	Clock High to Output Valid Clock High to WAIT Valid	Max	17	ns
	t_{KHQX} t_{KHTX}	t_{CLKHQX}	Clock High to Output Transition Clock High to WAIT Transition	Min	3	ns
	t_{LLKH}	$t_{ADVLCLKH}$	Latch Enable Low to Clock High	Min	9	ns
Clock Specifications	t_{KHKH}	t_{CLK}	Clock Period (f=52 MHz)	Min	19	ns
	t_{KHKL} t_{KLKH}		Clock High to Clock Low Clock Low to Clock High	Min	6	ns
	t_f t_r		Clock Fall or Rise Time	Max	2	ns

1. Sampled only, not 100% tested.
2. For other timings please refer to *Table 22: Asynchronous Read AC characteristics*.

Figure 15. Write AC waveforms, Write Enable controlled

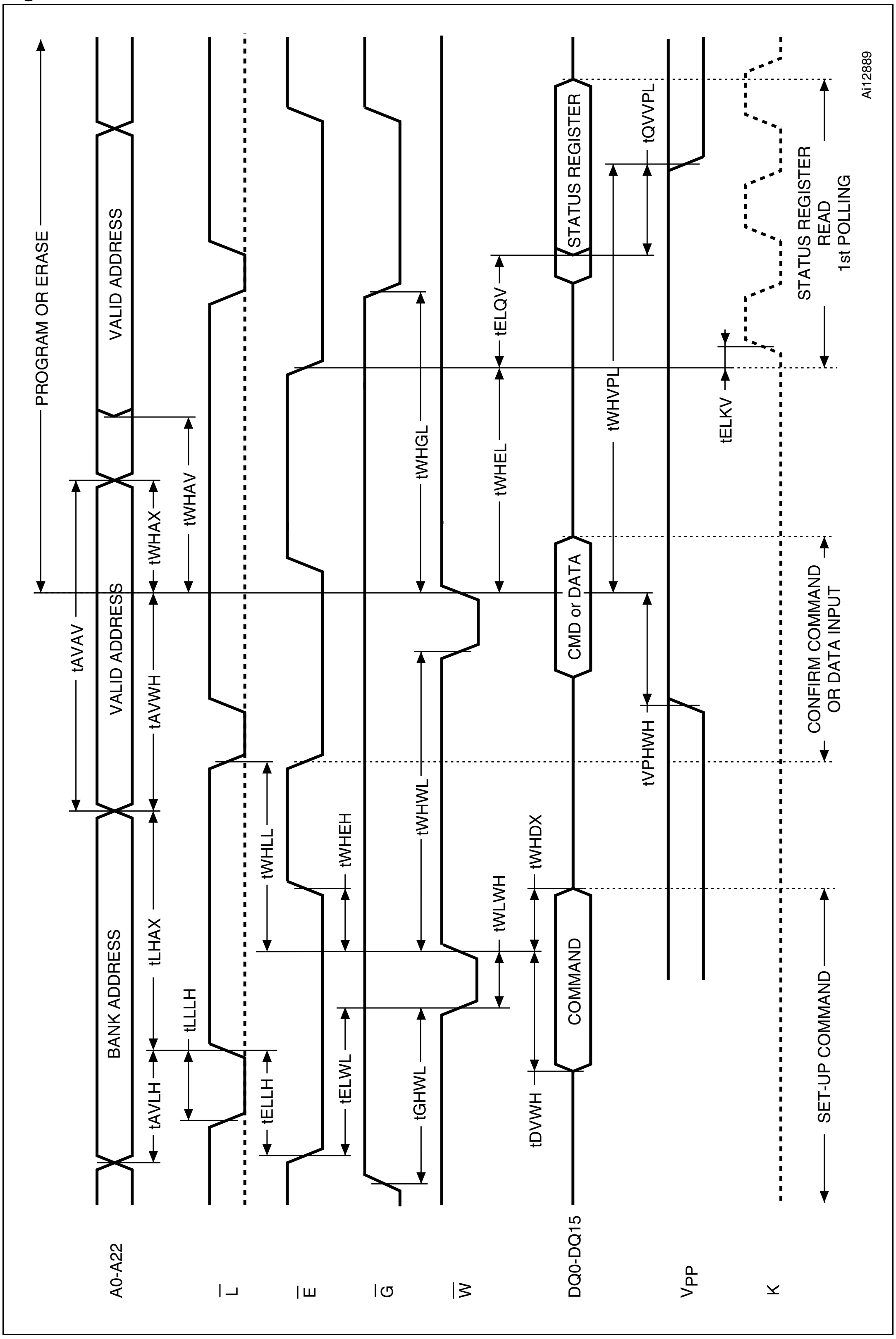


Table 24. Write AC characteristics, Write Enable controlled[(1)]

	Symbol	Alt	Parameter	M58LT128HST/B 85		Unit
Write Enable Controlled Timings	t_{AVAV}	t_{WC}	Address Valid to Next Address Valid	Min	85	ns
	t_{AVLH}		Address Valid to Latch Enable High	Min	10	ns
	t_{AVWH}[(2)]		Address Valid to Write Enable High	Min	50	ns
	t_{DVWH}	t_{DS}	Data Valid to Write Enable High	Min	50	ns
	t_{ELLH}		Chip Enable Low to Latch Enable High	Min	10	ns
	t_{ELWL}	t_{CS}	Chip Enable Low to Write Enable Low	Min	0	ns
	t_{ELQV}		Chip Enable Low to Output Valid	Min	85	ns
	t_{ELKV}		Chip Enable Low to Clock Valid	Min	9	ns
	t_{GHWL}		Output Enable High to Write Enable Low	Min	17	ns
	t_{LHAX}		Latch Enable High to Address Transition	Min	9	ns
	t_{LLLH}		Latch Enable Pulse Width	Min	10	ns
	t_{WHAV}[(2)]		Write Enable High to Address Valid	Min	0	ns
	t_{WHAX}[(2)]	t_{AH}	Write Enable High to Address Transition	Min	0	ns
	t_{WHDX}	t_{DH}	Write Enable High to Input Transition	Min	0	ns
	t_{WHEH}	t_{CH}	Write Enable High to Chip Enable High	Min	0	ns
	t_{WHEL}[(3)]		Write Enable High to Chip Enable Low	Min	25	ns
	t_{WHGL}		Write Enable High to Output Enable Low	Min	0	ns
	t_{WHLL}[(3)]		Write Enable High to Latch Enable Low	Min	25	ns
	t_{WHWL}	t_{WPH}	Write Enable High to Write Enable Low	Min	25	ns
	t_{WLWH}	t_{WP}	Write Enable Low to Write Enable High	Min	50	ns
Protection Timings	t_{QVVPL}		Output (Status Register) Valid to V_{PP} Low	Min	0	ns
	t_{VPHWH}	t_{VPS}	V_{PP} High to Write Enable High	Min	200	ns
	t_{WHVPL}		Write Enable High to V_{PP} Low	Min	200	ns

1. Sampled only, not 100% tested.
2. Meaningful only if $\overline{L}$ is always kept low.
3. t_{WHEL} and t_{WHLL} have this value when reading in the targeted bank or when reading following a Set Configuration Register command. System designers should take this into account and may insert a software No-Op instruction to delay the first read in the same bank after issuing any command and to delay the first read to any address after issuing a Set Configuration Register command. If the first read after the command is a Read Array operation in a different bank and no changes to the Configuration Register have been issued, t_{WHEL} and t_{WHLL} are 0 ns.

Figure 16. Write AC waveforms, Chip Enable controlled

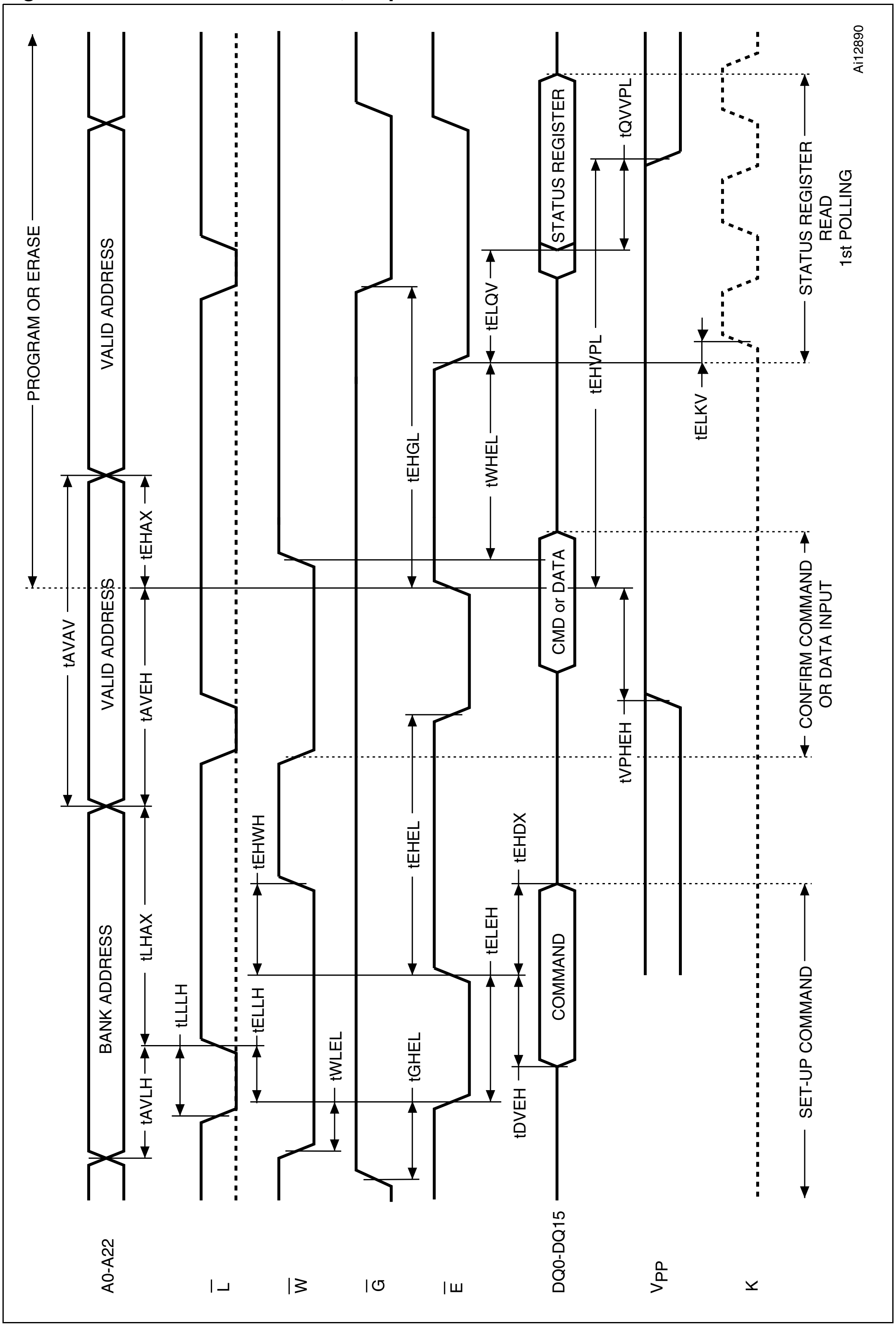


Table 25. Write AC characteristics, Chip Enable controlled(1)

	Symbol	Alt	Parameter	M58LT128HST/B		Unit
				85		
Chip Enable Controlled Timings	t_{AVAV}	t_{WC}	Address Valid to Next Address Valid	Min	85	ns
	t_{AVEH}		Address Valid to Chip Enable High	Min	50	ns
	t_{AVLH}		Address Valid to Latch Enable High	Min	10	ns
	t_{DVEH}	t_{DS}	Data Valid to Chip Enable High	Min	50	ns
	t_{EHAX}	t_{AH}	Chip Enable High to Address Transition	Min	0	ns
	t_{EHDX}	t_{DH}	Chip Enable High to Input Transition	Min	0	ns
	t_{EHEL}	t_{CPH}	Chip Enable High to Chip Enable Low	Min	25	ns
	t_{EHGL}		Chip Enable High to Output Enable Low	Min	0	ns
	t_{EHWH}	t_{CH}	Chip Enable High to Write Enable High	Min	0	ns
	t_{ELKV}		Chip Enable Low to Clock Valid	Min	9	ns
	t_{ELEH}	t_{CP}	Chip Enable Low to Chip Enable High	Min	50	ns
	t_{ELLH}		Chip Enable Low to Latch Enable High	Min	10	ns
	t_{ELQV}		Chip Enable Low to Output Valid	Min	85	ns
	t_{GHEL}		Output Enable High to Chip Enable Low	Min	17	ns
	t_{LHAX}		Latch Enable High to Address Transition	Min	9	ns
	t_{LLLH}		Latch Enable Pulse Width	Min	10	ns
	t_{WHEL}(2)		Write Enable High to Chip Enable Low	Min	25	ns
	t_{WLEL}	t_{CS}	Write Enable Low to Chip Enable Low	Min	0	ns
Protection Timings	t_{EHVPL}		Chip Enable High to V_{PP} Low	Min	200	ns
	t_{QVVPL}		Output (Status Register) Valid to V_{PP} Low	Min	0	ns
	t_{VPHEH}	t_{VPS}	V_{PP} High to Chip Enable High	Min	200	ns

1. Sampled only, not 100% tested.
2. t_{WHEL} has this value when reading in the targeted bank or when reading following a Set Configuration Register command. System designers should take this into account and may insert a software No-Op instruction to delay the first read in the same bank after issuing any command and to delay the first read to any address after issuing a Set Configuration Register command. If the first read after the command is a Read Array operation in a different bank and no changes to the Configuration Register have been issued, t_{WHEL} is 0 ns.

Figure 17. Reset and power-up AC waveforms

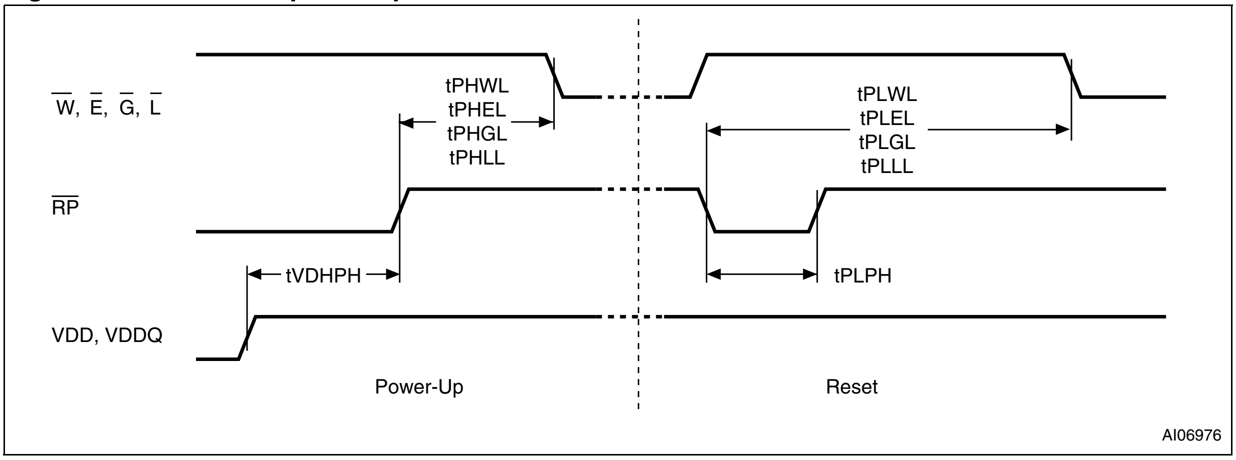


Table 26. Reset and power-up AC characteristics

Symbol	Parameter	Test condition		85	Unit
t_{PLWL} t_{PLEL} t_{PLGL} t_{PLLL}	Reset Low to Write Enable Low, Chip Enable Low, Output Enable Low, Latch Enable Low	During Program	Min	25	µs
		During Erase	Min	25	µs
		Read	Min	80	ns
		Other conditions	Min	20	µs
t_{PHWL} t_{PHEL} t_{PHGL} t_{PHLL}	Reset High to Write Enable Low Chip Enable Low Output Enable Low Latch Enable Low		Min	30	ns
t_{PLPH}(1),(2)	$\overline{RP}$ Pulse Width		Min	50	ns
t_{VDHPH}(3)	Supply Voltages High to Reset High		Min	250	µs

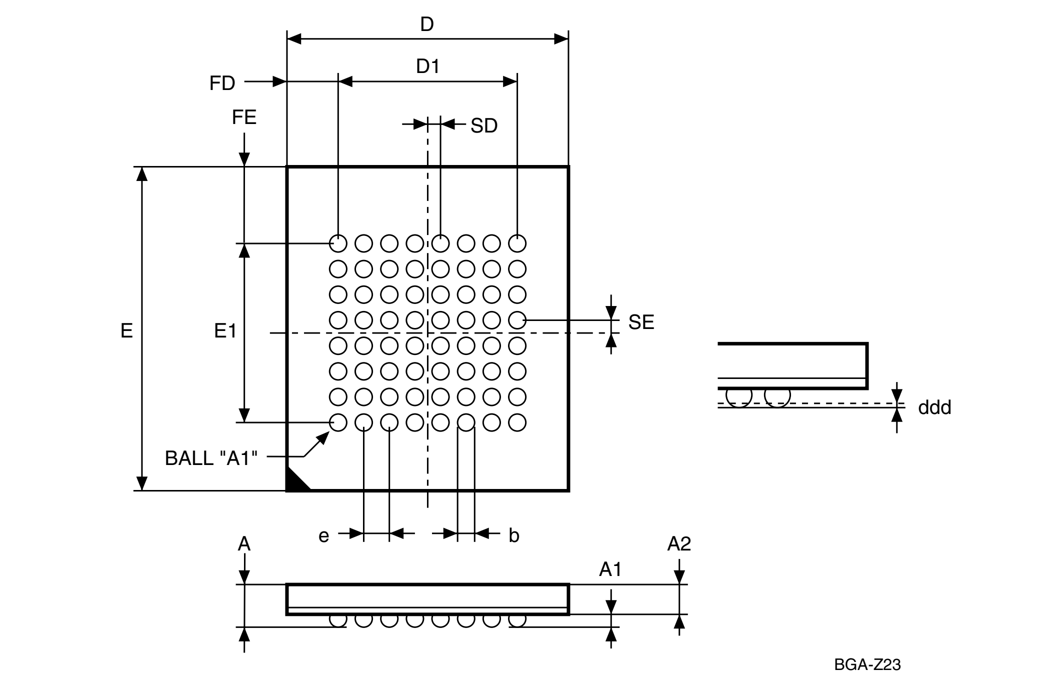
1. The device Reset is possible but not guaranteed if t_{PLPH} < 50ns.
2. Sampled only, not 100% tested.
3. It is important to assert $\overline{RP}$ in order to allow proper CPU initialization during power-up or reset.

13 Package mechanical

To meet environmental requirements, Numonyx offers the M58LT128HST and M58LT128HSB devices in ECOPACK® packages that have a lead-free, second-level interconnect. The category of second-level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97.

The maximum ratings related to soldering conditions are also marked on the inner box label.

Figure 18. TBGA64 10 × 13 mm - 8 × 8 active ball array, 1 mm pitch, bottom view package outline



1. Drawing is not to scale.

Table 27. TBGA64 10 × 13 mm - 8 × 8 active ball array, 1 mm pitch, package mechanical data

Symbol	millimeters			inches		
	Typ	Min	Max	Typ	Min	Max
A			1.200			0.0472
A1	0.300	0.200	0.350	0.0118	0.0079	0.0138
A2	0.800			0.0315		
b		0.350	0.500		0.0138	0.0197
D	10.000	9.900	10.100	0.3937	0.3898	0.3976
D1	7.000	–	–	0.2756	–	–
ddd			0.100			0.0039
e	1.000	–	–	0.0394	–	–
E	13.000	12.900	13.100	0.5118	0.5079	0.5157
E1	7.000	–	–	0.2756	–	–
FD	1.500	–	–	0.0591	–	–
FE	3.000	–	–	0.1181	–	–
SD	0.500	–	–	0.0197	–	–
SE	0.500	–	–	0.0197	–	–

14 Part numbering

Table 28. Ordering information scheme

Example: M58LT128HST 8 ZA 6 E

Device type

M58

Architecture

L = multilevel, multiple bank, burst mode

Operating voltage

T = V_{DD} = 1.7 V to 2.0 V, V_{DDQ} = 2.7 V to 3.6 V

Density

128 = 128 Mbit (x16)

Technology

H = 90nm technology

Security

S = Secure

Parameter location

T = Top boot

B = Bottom boot

Speed

8 = 85 ns

Package

ZA = TBGA64, 10 × 13 mm, 1 mm pitch

Temperature range

6 = –40 to 85°C

Packing option

E = ECOPACK® package, standard packing

F = ECOPACK® package, tape & reel packing

T = tape & reel packing

Blank = standard packing

Devices are shipped from the factory with the memory content bits erased to ’1’.

For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact the Numonyx Sales Office nearest to you.

Appendix A Block address tables

Table 29. Top boot block addresses, M58LT128HST

Bank[(1)]	#	Size (Kword)	Address range
Parameter Bank	0	16	7FC000-7FFFFF
	1	16	7F8000-7FBFFF
	2	16	7F4000-7F7FFF
	3	16	7F0000-7F3FFF
	4	64	7E0000-7EFFFF
	5	64	7D0000-7DFFFF
	6	64	7C0000-7CFFFF
	7	64	7B0000-7BFFFF
	8	64	7A0000-7AFFFF
	9	64	790000-79FFFF
	10	64	780000-78FFFF
Bank 1	11	64	770000-77FFFF
	12	64	760000-76FFFF
	13	64	750000-75FFFF
	14	64	740000-74FFFF
	15	64	730000-73FFFF
	16	64	720000-72FFFF
	17	64	710000-71FFFF
	18	64	700000-70FFFF
Bank 2	19	64	6F0000-6FFFFF
	20	64	6E0000-6EFFFF
	21	64	6D0000-6DFFFF
	22	64	6C0000-6CFFFF
	23	64	6B0000-6BFFFF
	24	64	6A0000-6AFFFF
	25	64	690000-69FFFF
	26	64	680000-68FFFF
Bank 3	27	64	670000-67FFFF
	28	64	660000-66FFFF
	29	64	650000-65FFFF
	30	64	640000-64FFFF
	31	64	630000-63FFFF
	32	64	620000-62FFFF
	33	64	610000-61FFFF
	34	64	600000-60FFFF

Table 29. Top boot block addresses, M58LT128HST (continued)

Bank[1]	#	Size (Kword)	Address range
Bank 4	35	64	5F0000-5FFFFF
	36	64	5E0000-5EFFFF
	37	64	5D0000-5DFFFF
	38	64	5C0000-5CFFFF
	39	64	5B0000-5BFFFF
	40	64	5A0000-5AFFFF
	41	64	590000-59FFFF
	42	64	580000-58FFFF
Bank 5	43	64	570000-57FFFF
	44	64	560000-56FFFF
	45	64	550000-55FFFF
	46	64	540000-54FFFF
	47	64	530000-53FFFF
	48	64	520000-52FFFF
	49	64	510000-51FFFF
	50	64	500000-50FFFF
Bank 6	51	64	4F0000-4FFFFF
	52	64	4E0000-4EFFFF
	53	64	4D0000-4DFFFF
	54	64	4C0000-4CFFFF
	55	64	4B0000-4BFFFF
	56	64	4A0000-4AFFFF
	57	64	490000-49FFFF
	58	64	480000-48FFFF
Bank 7	59	64	470000-47FFFF
	60	64	460000-46FFFF
	61	64	450000-45FFFF
	62	64	440000-44FFFF
	63	64	430000-43FFFF
	64	64	420000-42FFFF
	65	64	410000-41FFFF
	66	64	400000-40FFFF

Table 29. Top boot block addresses, M58LT128HST (continued)

Bank[(1)]	#	Size (Kword)	Address range
Bank 8	67	64	3F0000-3FFFFF
	68	64	3E0000-3EFFFF
	69	64	3D0000-3DFFFF
	70	64	3C0000-3CFFFF
	71	64	3B0000-3BFFFF
	72	64	3A0000-3AFFFF
	73	64	390000-39FFFF
	74	64	380000-38FFFF
Bank 9	75	64	370000-37FFFF
	76	64	360000-36FFFF
	77	64	350000-35FFFF
	78	64	340000-34FFFF
	79	64	330000-33FFFF
	80	64	320000-32FFFF
	81	64	310000-31FFFF
	82	64	300000-30FFFF
Bank 10	83	64	2F0000-2FFFFF
	84	64	2E0000-2EFFFF
	85	64	2D0000-2DFFFF
	86	64	2C0000-2CFFFF
	87	64	2B0000-2BFFFF
	88	64	2A0000-2AFFFF
	89	64	290000-29FFFF
	90	64	280000-28FFFF
Bank 11	91	64	270000-27FFFF
	92	64	260000-26FFFF
	93	64	250000-25FFFF
	94	64	240000-24FFFF
	95	64	230000-23FFFF
	96	64	220000-22FFFF
	97	64	210000-21FFFF
	98	64	200000-20FFFF

Table 29. Top boot block addresses, M58LT128HST (continued)

Bank(1)	#	Size (Kword)	Address range
Bank 12	99	64	1F0000-1FFFFF
	100	64	1E0000-1EFFFF
	101	64	1D0000-1DFFFF
	102	64	1C0000-1CFFFF
	103	64	1B0000-1BFFFF
	104	64	1A0000-1AFFFF
	105	64	190000-19FFFF
	106	64	180000-18FFFF
Bank 13	107	64	170000-17FFFF
	108	64	160000-16FFFF
	109	64	150000-15FFFF
	110	64	140000-14FFFF
	111	64	130000-13FFFF
	112	64	120000-12FFFF
	113	64	110000-11FFFF
	114	64	100000-10FFFF
Bank 14	115	64	0F0000-0FFFFF
	116	64	0E0000-0EFFFF
	117	64	0D0000-0DFFFF
	118	64	0C0000-0CFFFF
	119	64	0B0000-0BFFFF
	120	64	0A0000-0AFFFF
	121	64	090000-09FFFF
	122	64	080000-08FFFF
Bank 15	123	64	070000-07FFFF
	124	64	060000-06FFFF
	125	64	050000-05FFFF
	126	64	040000-04FFFF
	127	64	030000-03FFFF
	128	64	020000-02FFFF
	129	64	010000-01FFFF
	130	64	000000-00FFFF

1. There are two Bank Regions: Bank Region 1 contains all the banks that are made up of main blocks only; Bank Region 2 contains the banks that are made up of the parameter and main blocks (Parameter Bank).

Table 30. Bottom boot block addresses, M58LT128HSB

Bank(1)	#	Size (Kword)	Address range
Bank 15	130	64	7F0000-7FFFFF
	129	64	7E0000-7EFFFF
	128	64	7D0000-7DFFFF
	127	64	7C0000-7CFFFF
	126	64	7B0000-7BFFFF
	125	64	7A0000-7AFFFF
	124	64	790000-79FFFF
	123	64	780000-78FFFF
Bank 14	122	64	770000-77FFFF
	121	64	760000-76FFFF
	120	64	750000-75FFFF
	119	64	740000-74FFFF
	118	64	730000-73FFFF
	117	64	720000-72FFFF
	116	64	710000-71FFFF
	115	64	700000-70FFFF
Bank 13	114	64	6F0000-6FFFFF
	113	64	6E0000-6EFFFF
	112	64	6D0000-6DFFFF
	111	64	6C0000-6CFFFF
	110	64	6B0000-6BFFFF
	109	64	6A0000-6AFFFF
	108	64	690000-69FFFF
	107	64	680000-68FFFF
Bank 12	106	64	670000-67FFFF
	105	64	660000-66FFFF
	104	64	650000-65FFFF
	103	64	640000-64FFFF
	102	64	630000-63FFFF
	101	64	620000-62FFFF
	100	64	610000-61FFFF
	99	64	600000-60FFFF

Table 30. Bottom boot block addresses, M58LT128HSB (continued)

Bank[(1)]	#	Size (Kword)	Address range
Bank 11	98	64	5F0000-5FFFFF
	97	64	5E0000-5EFFFF
	96	64	5D0000-5DFFFF
	95	64	5C0000-5CFFFF
	94	64	5B0000-5BFFFF
	93	64	5A0000-5AFFFF
	92	64	590000-59FFFF
	91	64	580000-58FFFF
Bank 10	90	64	570000-57FFFF
	89	64	560000-56FFFF
	88	64	550000-55FFFF
	87	64	540000-54FFFF
	86	64	530000-53FFFF
	85	64	520000-52FFFF
	84	64	510000-51FFFF
	83	64	500000-50FFFF
Bank 9	82	64	4F0000-4FFFFF
	81	64	4E0000-4EFFFF
	80	64	4D0000-4DFFFF
	79	64	4C0000-4CFFFF
	78	64	4B0000-4BFFFF
	77	64	4A0000-4AFFFF
	76	64	490000-49FFFF
	75	64	480000-48FFFF
Bank 8	74	64	470000-47FFFF
	73	64	460000-46FFFF
	72	64	450000-45FFFF
	71	64	440000-44FFFF
	70	64	430000-43FFFF
	69	64	420000-42FFFF
	68	64	410000-41FFFF
	67	64	400000-40FFFF

Table 30. Bottom boot block addresses, M58LT128HSB (continued)

Bank[(1)]	#	Size (Kword)	Address range
Bank 7	66	64	3F0000-3FFFFF
	65	64	3E0000-3EFFFF
	64	64	3D0000-3DFFFF
	63	64	3C0000-3CFFFF
	62	64	3B0000-3BFFFF
	61	64	3A0000-3AFFFF
	60	64	390000-39FFFF
	59	64	380000-38FFFF
Bank 6	58	64	370000-37FFFF
	57	64	360000-36FFFF
	56	64	350000-35FFFF
	55	64	340000-34FFFF
	54	64	330000-33FFFF
	53	64	320000-32FFFF
	52	64	310000-31FFFF
	51	64	300000-30FFFF
Bank 5	50	64	2F0000-2FFFFF
	49	64	2E0000-2EFFFF
	48	64	2D0000-2DFFFF
	47	64	2C0000-2CFFFF
	46	64	2B0000-2BFFFF
	45	64	2A0000-2AFFFF
	44	64	290000-29FFFF
	43	64	280000-28FFFF
Bank 4	42	64	270000-27FFFF
	41	64	260000-26FFFF
	40	64	250000-25FFFF
	39	64	240000-24FFFF
	38	64	230000-23FFFF
	37	64	220000-22FFFF
	36	64	210000-21FFFF
	35	64	200000-20FFFF

Table 30. Bottom boot block addresses, M58LT128HSB (continued)

Bank(1)	#	Size (Kword)	Address range
Bank 3	34	64	1F0000-1FFFFF
	33	64	1E0000-1EFFFF
	32	64	1D0000-1DFFFF
	31	64	1C0000-1CFFFF
	30	64	1B0000-1BFFFF
	29	64	1A0000-1AFFFF
	28	64	190000-19FFFF
	27	64	180000-18FFFF
Bank 2	26	64	170000-17FFFF
	25	64	160000-16FFFF
	24	64	150000-15FFFF
	23	64	140000-14FFFF
	22	64	130000-13FFFF
	21	64	120000-12FFFF
	20	64	110000-11FFFF
	19	64	1F0000-1FFFFF
Bank 1	18	64	0F0000-0FFFFF
	17	64	0E0000-0EFFFF
	16	64	0D0000-0DFFFF
	15	64	0C0000-0CFFFF
	14	64	0B0000-0BFFFF
	13	64	0A0000-0AFFFF
	12	64	090000-09FFFF
	11	64	080000-08FFFF
Parameter Bank	10	64	070000-07FFFF
	9	64	060000-06FFFF
	8	64	050000-05FFFF
	7	64	040000-04FFFF
	6	64	030000-03FFFF
	5	64	020000-02FFFF
	4	64	010000-01FFFF
	3	16	00C000-00FFFF
	2	16	008000-00BFFF
	1	16	004000-007FFF
	0	16	000000-003FFF

1. There are two bank regions: bank region 2 contains all the banks that are made up of main blocks only; bank region 1 contains the banks that are made up of the parameter and main blocks (parameter bank).

Appendix B Common Flash Interface

The Common Flash Interface is a JEDEC-approved, standardized data structure that can be read from the Flash memory device. It allows a system software to query the device to determine various electrical and timing parameters, density information, and functions supported by the memory. The system can interface easily with the device, enabling the software to upgrade itself when necessary.

When the Read CFI Query Command is issued the device enters CFI Query mode and the data structure is read from the memory. Tables *31*, *32*, *33*, *34*, *35*, *36*, *37*, *38*, *39* and *40* show the addresses used to retrieve the data. The query data is always presented on the lowest order data outputs (DQ0-DQ7), and the other outputs (DQ8-DQ15) are set to 0.

The CFI data structure also contains a security area where a 64-bit unique security number is written (see *Figure 4: Protection Register memory map*). This area can be accessed only in Read mode by the final user. It is impossible to change the security number after it has been written by Numonyx. Issue a Read Array command to return to Read mode.

Table 31. Query structure overview

Offset	Sub-section name	Description
000h	Reserved	Reserved for algorithm-specific information
010h	CFI Query Identification String	Command set ID and algorithm data offset
01Bh	System Interface Information	Device timing and voltage information
027h	Device Geometry Definition	Flash device layout
P	Primary Algorithm-specific Extended Query table	Additional information specific to the primary algorithm (optional)
A	Alternate Algorithm-specific Extended Query table	Additional information specific to the alternate algorithm (optional)
080h	Security Code Area	Lock Protection Register, unique device number and user-programmable OTP

1. The Flash memory displays the CFI data structure when the CFI Query command is issued. This table lists the main sub-sections detailed in Tables *32*, *33*, *34* and *35*. Query data is always presented on the lowest order data outputs.

Table 32. CFI query identification string

Offset	Sub-section name	Description		Value
000h	0020h	Manufacturer code		Numonyx
001h	88D6h 88D7h	Device code	M58LT128HST M58LT128HSB	Top Bottom
002h-00Fh	Reserved	Reserved		
010h 011h 012h	0051h 0052h 0059h	Query unique ASCII string "QRY"		"Q" "R" "Y"
013h 014h	0001h 0000h	Primary Algorithm Command Set and Control Interface ID code 16 bit ID code defining a specific algorithm		
015h 016h	offset = P = 000Ah 0001h	Address for Primary Algorithm extended query table (see *Table 35*)		p = 10Ah
017h 018h	0000h 0000h	Alternate Vendor Command Set and Control Interface ID Code second vendor - specified algorithm supported		NA
019h 01Ah	value = A = 0000h 0000h	Address for Alternate Algorithm extended query table		NA

Table 33. CFI query system interface information

Offset	Data	Description	Value
01Bh	0017h	V_{DD} logic supply minimum Program/Erase or Write voltage bit 7 to 4 BCD value in volts bit 3 to 0 BCD value in 100 millivolts	1.7V
01Ch	0020h	V_{DD} logic supply maximum Program/Erase or Write voltage bit 7 to 4 BCD value in volts bit 3 to 0 BCD value in 100 millivolts	2V
01Dh	0085h	V_{PP} [programming] supply minimum Program/Erase voltage bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 millivolts	8.5V
01Eh	0095h	V_{PP} [programming] supply maximum Program/Erase voltage bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 millivolts	9.5V
01Fh	0004h	Typical timeout per single byte/word program = 2^n µs	16µs
020h	0009h	Typical timeout for Buffer Program = 2^n µs	512µs
021h	000Ah	Typical timeout per individual block erase = 2^n ms	1s
022h	0000h	Typical timeout for full chip erase = 2^n ms	NA
023h	0004h	Maximum timeout for word program = 2^n times typical	256 µs
024h	0004h	Maximum timeout for Buffer Program = 2^n times typical	8192 µs
025h	0002h	Maximum timeout per individual block erase = 2^n times typical	4s
026h	0000h	Maximum timeout for chip erase = 2^n times typical	NA

Table 34. Device geometry definition

	Offset	Data	Description	Value
	027h	0018h	Device size = 2^n in number of bytes	16 Mbytes
	028h 029h	0001h 0000h	Flash Device Interface Code description	x16 Async.
	02Ah 02Bh	0006h 0000h	Maximum number of bytes in multi-byte program or page = 2^n	64 bytes
	02Ch	0002h	Number of identical size erase block regions within the device bit 7 to 0 = x = number of Erase Block regions	2
TOP DEVICES	02Dh 02Eh	007Eh 0000h	Erase Block Region 1 information Number of identical-size erase blocks = 007Eh+1	127
TOP DEVICES	02Fh 030h	0000h 0002h	Erase Block Region 1 information Block size in Region 1 = 0200h * 256 byte	128 Kbyte
TOP DEVICES	031h 032h	0003h 0000h	Erase Block Region 2 information Number of identical-size erase blocks = 0003h+1	4
TOP DEVICES	033h 034h	0080h 0000h	Erase Block Region 2 information Block size in Region 2 = 0080h * 256 byte	32 Kbyte
TOP DEVICES	035h 038h	Reserved	Reserved for future erase block region information	NA
BOTTOM DEVICES	02Dh 02Eh	0003h 0000h	Erase Block Region 1 information Number of identical-size erase block = 0003h+1	4
BOTTOM DEVICES	02Fh 030h	0080h 0000h	Erase Block Region 1 information Block size in Region 1 = 0080h * 256 bytes	32 Kbytes
BOTTOM DEVICES	031h 032h	007Eh 0000h	Erase Block Region 2 information Number of identical-size erase block = 007Eh+1	127
BOTTOM DEVICES	033h 034h	0000h 0002h	Erase Block Region 2 information Block size in Region 2 = 0200h * 256 bytes	128 Kbytes
BOTTOM DEVICES	035h 038h	Reserved	Reserved for future erase block region information	NA

Table 35. Primary algorithm-specific extended query table

Offset	Data	Description	Value
(P)h = 10Ah	0050h 0052h 0049h	Primary Algorithm extended query table unique ASCII string “PRI”	"P" "R" "I"
(P+3)h =10Dh	0031h	Major version number, ASCII	"1"
(P+4)h = 10Eh	0033h	Minor version number, ASCII	"3"
(P+5)h = 10Fh (P+7)h = 111h (P+8)h = 112h	00E6h 0003h 0000h 0000h	Extended query table contents for Primary Algorithm. address (P+5)h contains less significant bytes. bit 0 Chip Erase supported(1 = Yes, 0 = No) bit 1 Erase Suspend supported(1 = Yes, 0 = No) bit 2 Program Suspend supported(1 = Yes, 0 = No) bit 3 Legacy Protect/Unprotect supported(1 = Yes, 0 = No) bit 4 Queued Erase supported(1 = Yes, 0 = No) bit 5 Instant individual block protection supported(1 = Yes, 0 = No) bit 6 Protection bits supported(1 = Yes, 0 = No) bit 7 Page mode read supported(1 = Yes, 0 = No) bit 8 Synchronous read supported(1 = Yes, 0 = No) bit 9 Simultaneous operation supported(1 = Yes, 0 = No) bit 10 to 31 Reserved; undefined bits are ‘0’. If bit 31 is ’1’ then another 31 bit field of optional features follows at the end of the bit-30 field.	No Yes Yes No No Yes Yes Yes Yes Yes
(P+9)h = 113h	0001h	Supported Functions after Suspend Read Array, Read Status Register and CFI Query bit 0 Program supported after Erase Suspend (1 = Yes, 0 = No) bit 7 to 1 Reserved; undefined bits are ‘0’	Yes
(P+A)h = 114h (P+B)h = 115h	0003h 0000h	Block Protect Status Defines which bits in the Block Status Register section of the query are implemented. bit 0 Block protect Status Register Protect/Unprotect bit active (1 = Yes, 0 = No) bit 1 Block Protection Status Register Lock-Down bit active (1 = Yes, 0 = No) bit 15 to 2 Reserved for future use; undefined bits are ‘0’	Yes No
(P+C)h = 116h	0018h	V_{DD} logic supply optimum Program/Erase voltage (highest performance) bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 mV	1.8V
(P+D)h = 117h	0090h	V_{PP} supply optimum Program/Erase voltage bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 mV	9V

Table 36. Protection Register information

Offset	Data	Description	Value
(P+E)h = 118h	0002h	Number of Protection Register fields in JEDEC ID space. 0000h indicates that 256 fields are available.	2
(P+F)h = 119h	0080h	Protection Field 1: Protection description Bits 0-7 Lower byte of Protection Register address Bits 8-15 Upper byte of Protection Register address Bits 16-23 2^n bytes in factory preprogrammed region Bits 24-31 2^n bytes in user-programmable region	80h
(P+10)h = 11Ah	0000h		00h
(P+ 11)h = 11Bh	0003h		8 bytes
(P+12)h = 11Ch	0003h		8 bytes
(P+13)h = 11Dh	0089h	Protection Register 2: protection description Bits 0-31 Protection Register address Bits 32-39 n number of factory programmed regions (lower byte) Bits 40-47 n number of factory programmed regions (upper byte) Bits 48-55 2^n bytes in factory programmable region Bits 56-63 n number of user programmable regions (lower byte) Bits 64-71 n number of user programmable regions (upper byte) Bits 72-79 2^n bytes in user programmable region	89h
(P+14)h = 11Eh	0000h		00h
(P+15)h = 11Fh	0000h		00h
(P+16)h = 120h	0000h		00h
(P+17)h = 121h	0000h		0
(P+18)h = 122h	0000h		0
(P+19)h = 123h	0000h		0
(P+1A)h = 124h	0010h		16
(P+1B)h = 125h	0000h		0
(P+1C)h = 126h	0004h		16

Table 37. Burst Read information

Offset	Data	Description	Value
(P+1D)h = 127h	0003h	Page-mode read capability bits 0-7 n' such that 2^n HEX value represents the number of read-page bytes. See offset 0028h for device word width to determine page-mode data output width.	8 bytes
(P+1E)h = 128h	0004h	Number of Synchronous mode read configuration fields that follow.	4
(P+1F)h = 129h	0001h	Synchronous mode read capability configuration 1 bit 3-7 Reserved bit 0-2 n' such that 2^{n+1} HEX value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width. A value of 07h indicates that the device is capable of continuous linear bursts that output data until the internal burst counter reaches the end of the device's burstable address space. This field's 3-bit value can be written directly to the read Configuration Register bit 0-2 if the device is configured for its maximum word width. See offset 0028h for word width to determine the burst data output width.	4
(P+20)h = 12Ah	0002h	Synchronous mode read capability configuration 2	8
(P-21)h = 12Bh	0003h	Synchronous mode read capability configuration 3	16
(P+22)h = 12Ch	0007h	Synchronous mode read capability configuration 4	Cont.

Table 38. Bank and Erase Block region information[1] [2]

Flash memory (top)		Flash memory (bottom)		Description
Offset	Data	Offset	Data	
(P+23)h = 12Dh	02h	(P+23)h = 12Dh	02h	Number of bank regions within the device

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There are two bank regions, see *Table 29* and *Table 30*.

Table 39. Bank and Erase Block region 1 information

M58LT128HST (top)		M58LT128HSB (bottom)		Description
Offset	Data	Offset	Data	
(P+24)h = 12Eh	0Fh	(P+24)h = 12Eh	01h	Number of identical banks within bank region 1
(P+25)h = 12Fh	00h	(P+25)h = 12Fh	00h	
(P+26)h = 130h	11h	(P+26)h = 130h	11h	Number of program or erase operations allowed in bank region 1: Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+27)h = 131h	00h	(P+27)h = 131h	00h	Number of Program or Erase operations allowed in other banks while a bank in same region is programming Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+28)h = 132h	00h	(P+28)h = 132h	00h	Number of Program or Erase operations allowed in other banks while a bank in this region is erasing Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+29)h = 133h	01h	(P+29)h = 133h	02h	Types of Erase Block regions in bank region 1 n = number of Erase Block regions with contiguous same-size erase blocks. Symmetrically blocked banks have one blocking region[2].
(P+2A)h = 134h	07h	(P+2A)h = 134h	03h	Bank region 1 Erase Block Type 1 Information Bits 0-15: n+1 = number of identical size erase blocks Bits 16-31: n×256 = number of bytes in Erase Block region
(P+2B)h = 135h	00h	(P+2B)h = 135h	00h	
(P+2C)h = 136h	00h	(P+2C)h = 136h	80h	
(P+2D)h = 137h	02h	(P+2D)h = 137h	00h	
(P+2E)h = 138h	64h	(P+2E)h = 138h	64h	Bank region 1 (Erase Block Type 1) Minimum block erase cycles × 1000
(P+2F)h = 139h	00h	(P+2F)h = 139h	00h	

Table 39. Bank and Erase Block region 1 information (continued)

M58LT128HST (top)		M58LT128HSB (bottom)		Description
Offset	Data	Offset	Data	
(P+30)h = 13Ah	01h	(P+30)h = 13Ah	01h	Bank region 1 (Erase Block Type 1): bits per cell, internal ECC Bits 0-3: bits per cell in erase region Bit 4: reserved for "internal ECC used" BIts 5-7: reserved
(P+31)h = 13Bh	03h	(P+31)h = 13Bh	03h	Bank region 1 (Erase Block Type 1): Page mode and Synchronous mode capabilities Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
		(P+32)h = 13Ch	06h	Bank region 1 Erase Block Type 2 Information Bits 0-15: n+1 = number of identical size Erase Blocks Bits 16-31: n×256 = number of bytes in Erase Block region
		(P+33)h = 13Dh	00h	
		(P+34)h = 13Eh	00h	
		(P+35)h = 13Fh	02h	
		(P+36)h = 140h	64h	Bank region 1 (Erase Block Type 2) Minimum Block Erase cycles × 1000
		(P+37)h = 141h	00h	
		(P+38)h = 142h	01h	Bank regions 1 (Erase Block Type 2): BIts per cell, internal ECC Bits 0-3: bits per cell in erase region Bit 4: reserved for "internal ECC used" BIts 5-7: reserved
		(P+39)h = 143h	03h	Bank region 1 (Erase Block Type 2): Page mode and Synchronous mode capabilities Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There are two bank regions, see *Table 29* to *Table 30*.

Table 40. Bank and Erase Block region 2 Information

M58LT128HST (top)		M58LT128HSB (bottom)		Description
Offset	Data	Offset	Data	
(P+32)h = 13Ch	01h	(P+3A)h = 144h	0Fh	Number of identical banks within bank region 2
(P+33)h = 13Dh	00h	(P+3B)h = 145h	00h	
(P+34)h = 13Eh	11h	(P+3C)h = 146h	11h	Number of Program or Erase operations allowed in bank region 2: Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+35)h = 13Fh	00h	(P+3D)h = 147h	00h	Number of Program or Erase operations allowed in other banks while a bank in this region is programming Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+36)h = 140h	00h	(P+3E)h = 148h	00h	Number of Program or Erase operations allowed in other banks while a bank in this region is erasing Bits 0-3: Number of simultaneous Program operations Bits 4-7: Number of simultaneous Erase operations
(P+37)h = 141h	02h	(P+3F)h = 149h	01h	Types of Erase Block regions in Bank Region 2 n = number of Erase Block regions with contiguous same-size erase blocks. Symmetrically blocked banks have one blocking region.(2)
(P+38)h = 142h	06h	(P+40)h = 14Ah	07h	Bank region 2 Erase Block type 1 Information Bits 0-15: n+1 = number of same-size erase blocks Bits 16-31: n×256 = number of bytes in Erase Block region
(P+39)h = 143h	00h	(P+41)h = 14Bh	00h	
(P+3A)h = 144h	00h	(P+42)h = 14Ch	00h	
(P+3B)h = 145h	02h	(P+43)h = 14Dh	02h	
(P+3C)h = 146h	64h	(P+44)h = 14Eh	64h	Bank region 2 (Erase Block type 1) Minimum Block Erase cycles × 1000
(P+3D)h = 147h	00h	(P+45)h = 14Fh	00h	
(P+3E)h = 148h	01h	(P+46)h = 150h	01h	Bank region 2 (Erase Block type 1): BIts per cell, internal ECC Bits 0-3: bits per cell in Erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved

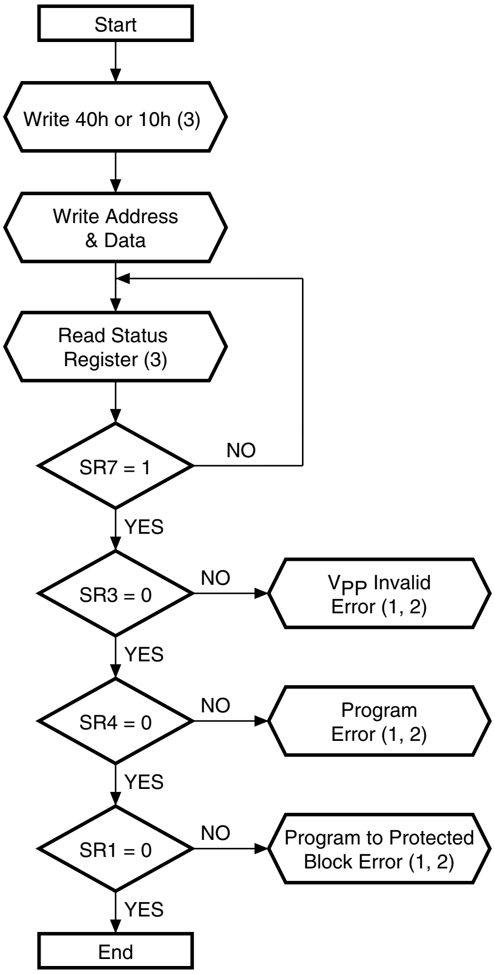
Table 40. Bank and Erase Block region 2 Information (continued)

M58LT128HST (top)		M58LT128HSB (bottom)		Description
Offset	Data	Offset	Data	
(P+3F)h = 149h	03h	(P+47)h = 151h	03h	Bank region 2 (Erase Block type 1):Page mode and Synchronous mode capabilities (defined in *Table 37*) Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
(P+40)h = 14Ah	03h			Bank region 2 Erase Block type 2 Information Bits 0-15: n+1 = number of same-size erase blocks Bits 16-31: n × 256 = number of bytes in Erase Block region
(P+41)h = 14Bh	00h			
(P+42)h = 14Ch	80h			
(P+43)h = 14Dh	00h			
(P+44)h = 14Eh	64h			Bank region 2 (Erase Block type 2) Minimum Block Erase cycles × 1000
(P+45)h = 14Fh	00h			
(P+46)h = 150h	01h			Bank region 2 (Erase Block type 2): BIts per cell, internal ECC Bits 0-3: bits per cell in Erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved
(P+47)h = 151h	03h			Bank region 2 (Erase Block type 2): Page mode and Synchronous mode capabilities (defined in *Table 37*) Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
(P+48)h = 152h		(P+48)h = 152h		Feature Space definitions
(P+49)h = 153h		(P+43)h = 153h		Reserved

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There are two bank regions, see *Table 29* and *Table 30*.

Appendix C Flowcharts and pseudo codes

Figure 19. Program flowchart and pseudo code

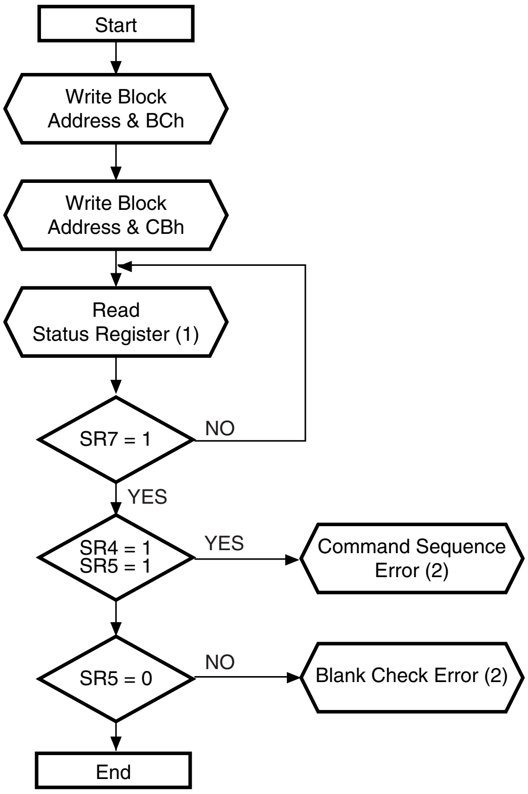



```
program_command (addressToProgram, dataToProgram) {:

        writeToFlash (addressToProgram, 0x40);
        /*writeToFlash (addressToProgram, 0x10);*/
                                        /*see note (3)*/

         writeToFlash (addressToProgram, dataToProgram) ;
         /*Memory enters read status state after
         the Program Command*/

      do {
         status_register=readFlash (addressToProgram);
                                       "see note (3)";
         /* E or G must be toggled*/

         }  while (status_register.SR7== 0) ;

if  (status_register.SR3==1) /*VPP invalid error */
      error_handler ( ) ;

if  (status_register.SR4==1) /*program error */
      error_handler ( ) ;

if  (status_register.SR1==1) /*program to protect block error */
      error_handler ( ) ;

}
```

AI06170b

1. Status check of SR1 (protected block), SR3 (V_{PP} Invalid) and SR4 (program error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase Controller operations.
3. Any address within the bank can equally be used.

Figure 20. Blank Check flowchart and pseudo code

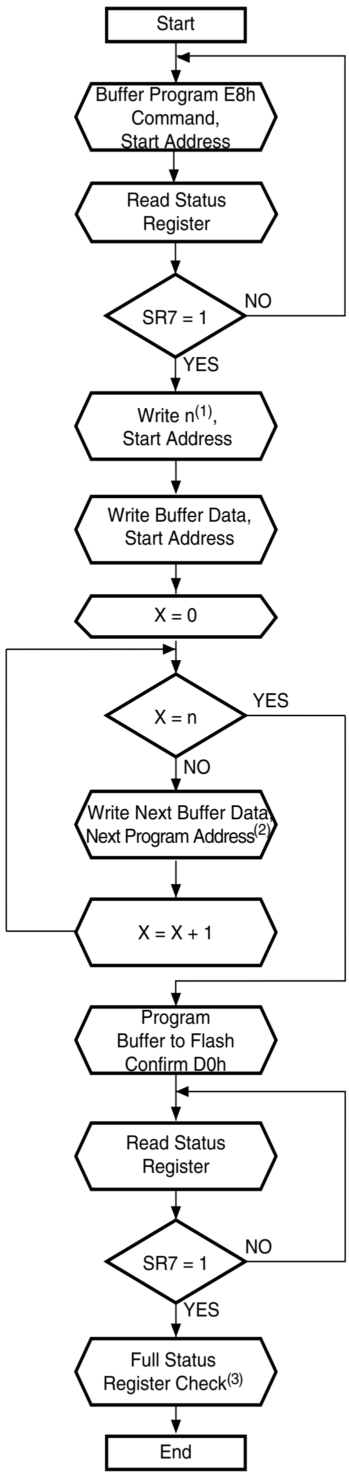



```
blank_check_command (blockToCheck) {

        writeToFlash (blockToCheck, 0xBC);

        writeToFlash (blockToCheck, 0xCB);
        /* Memory enters read status state after
        the Blank Check Command */

        do {
             status_register = readFlash (blockToCheck);
                                /* see note (1) */
             /* E or G must be toggled */

             } while (status_register.SR7==0);

        if (status_register.SR4==1) && (status_register.SR5==1)
        /* command sequence error */
                 error_handler () ;

        if (status_register.SR5==1)
        /* Blank Check error */
                 error_handler () ;

  }
```

ai10520c

1. Any address within the bank can equally be used.
2. If an error is found, the Status Register must be cleared before further Program/Erase operations.

Figure 21. Buffer Program flowchart and pseudo code

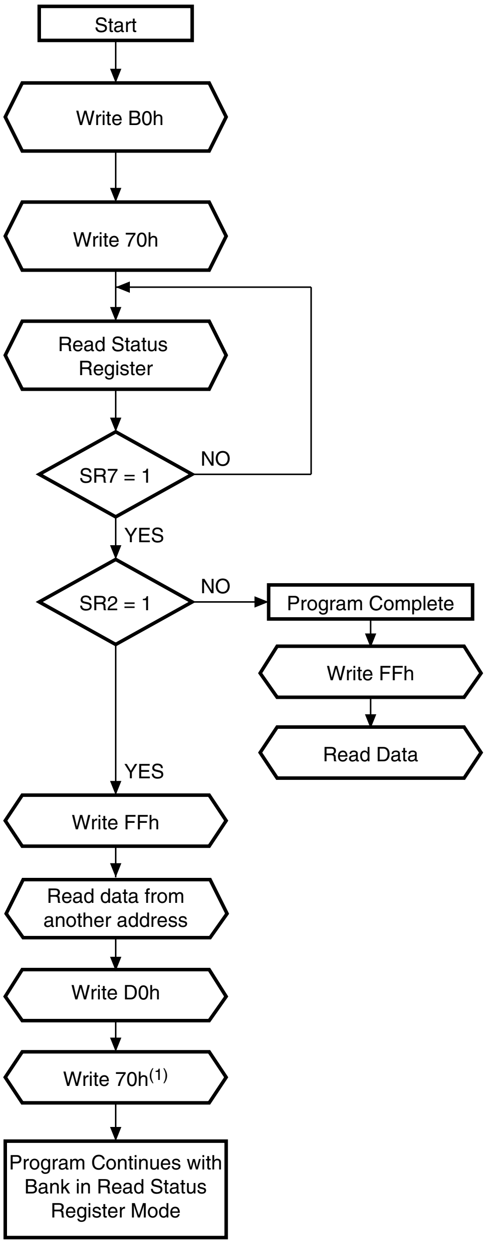



```
Buffer_Program_command (Start_Address, n, buffer_Program[] )
 /* buffer_Program [] is an array structure used to store the address and
data to be programmed to the Flash memory (the address must be within
the segment Start Address and Start Address+n) */
{
 do {writeToFlash (Start_Address, 0xE8) ;

status_register=readFlash (Start_Address);

 } while (status_register.SR7==0);

 writeToFlash (Start_Address, n);

 writeToFlash (buffer_Program[0].address, buffer_Program[0].data);
/*buffer_Program[0].address is the start address*/

 x = 0;

 while (x<n)

  { writeToFlash (buffer_Program[x+1].address, buffer_Program[x+1].data);

        x++;

   }

  writeToFlash (Start_Address, 0xD0);

 do {status_register=readFlash (Start_Address);

        } while (status_register.SR7==0);

 full_status_register_check();

 }
```

AI08913b

1. n + 1 is the number of data being programmed.
2. Next program data is an element belonging to buffer_Program[].data; next program address is an element belonging to buffer_Program[].address
3. Routine for Error Check by reading SR3, SR4 and SR1.

Figure 22. Program Suspend & Resume flowchart and pseudo code

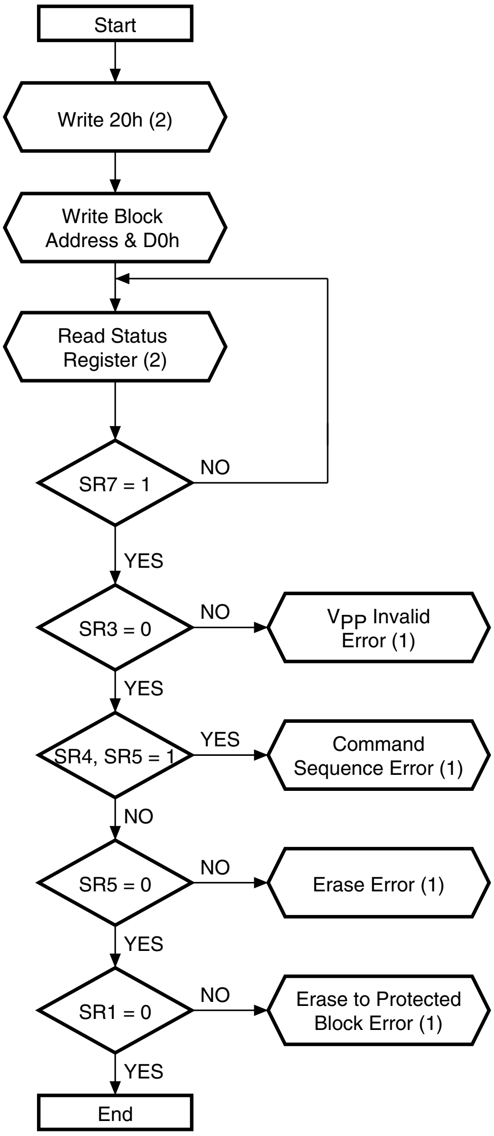



```
program_suspend_command ( ) {
        writeToFlash (any_address, 0xB0) ;

        writeToFlash (bank_address, 0x70) ;
        /* read status register to check if
        program has already completed */

   do {
        status_register=readFlash (bank_address) ;
        /* E̅ or G̅ must be toggled*/

        } while (status_register.SR7== 0) ;

if (status_register.SR2==0) /*program completed */

        { writeToFlash (bank_address, 0xFF) ;
          read_data ( ) ;
          /*The device returns to Read Array
          (as if program/erase suspend was not issued).*/
          }
else
        { writeToFlash (bank_address, 0xFF) ;

          read_data ( ); /*read data from another address*/

          writeToFlash (any_address, 0xD0) ;
          /*write 0xD0 to resume program*/

          writeToFlash (bank_address, 0x70) ;
          /*read status register to check if program has completed */

          }
}
```

AI10117b

1. The Read Status Register command (Write 70h) can be issued just before or just after the Program Resume command.

Figure 23. Block Erase flowchart and pseudo code

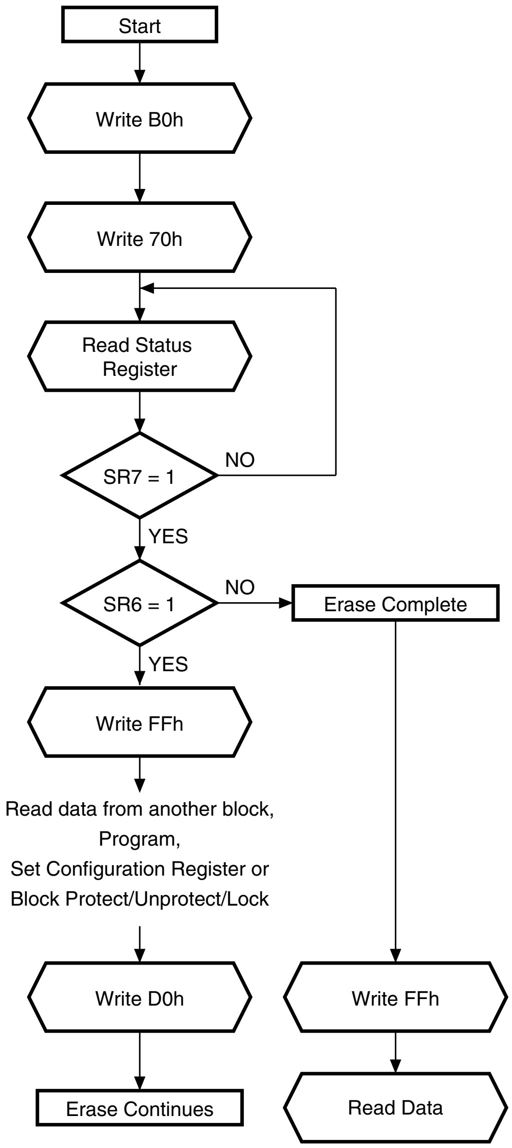



```
erase_command   ( blockToErase ) {
        writeToFlash (blockToErase, 0x20) ;
                                /*see note (2) */

        writeToFlash (blockToErase, 0xD0) ;
        /* Memory enters read status state after
        the Erase Command */

    do {
        status_register=readFlash (blockToErase) ;
                                        /* see note (2) */
        /* E or G must be toggled*/

        } while (status_register.SR7== 0) ;

if  (status_register.SR3==1) /*VPP invalid error */
        error_handler ( ) ;

if  ( (status_register.SR4==1) && (status_register.SR5==1) )
/* command sequence error */
        error_handler ( ) ;

if  ( (status_register.SR5==1) )
/* erase error */
        error_handler ( ) ;

if  (status_register.SR1==1) /*program to protect block error */
        error_handler ( ) ;

}
```

AI10976

1. If an error is found, the Status Register must be cleared before further Program/Erase operations.
2. Any address within the bank can equally be used.

Figure 24. Erase Suspend & Resume flowchart and pseudo code

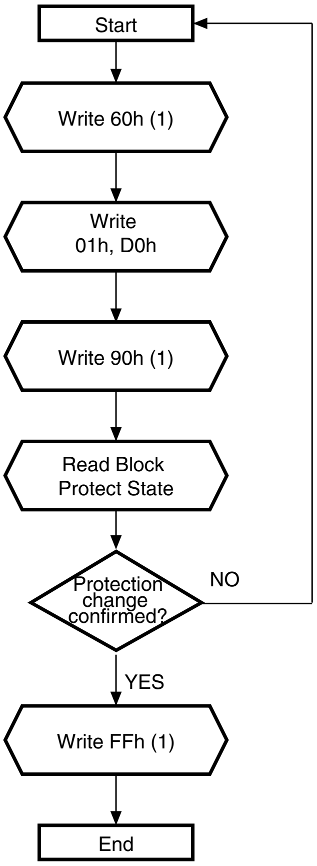



```
erase_suspend_command ( ) {
        writeToFlash (bank_address, 0xB0) ;

        writeToFlash (bank_address, 0x70) ;
        /* read status register to check if
        erase has already completed */

     do {
         status_register=readFlash (bank_address) ;
         /* Ē or Ḡ must be toggled*/

        } while (status_register.SR7== 0) ;

if (status_register.SR6==0) /*erase completed */
        { writeToFlash (bank_address, 0xFF) ;

          read_data ( ) ;
          /*read data from another block*/
          /*The device returns to Read Array
          (as if program/erase suspend was not issued).*/

        }
else
        { writeToFlash (bank_address, 0xFF) ;
          read_program_data ( );
          /*read or program data from another address*/
          writeToFlash (bank_address, 0xD0) ;
          /*write 0xD0 to resume erase*/
        }
}
```

AI13893

1. The Read Status Register command (Write 70h) can be issued just before or just after the Erase Resume command.

Figure 25. Protect/Unprotect operation flowchart and pseudo code

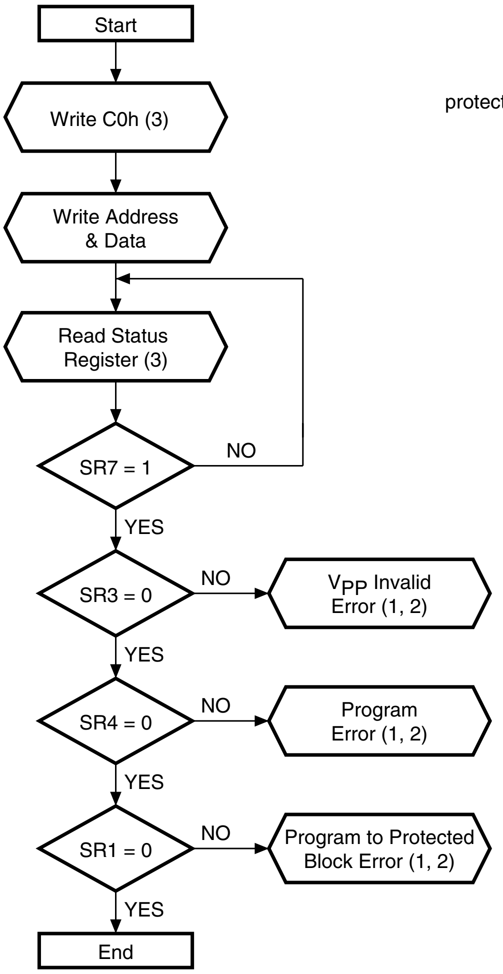



```
protect_operation_command   (address, protect_operation) {
       writeToFlash (address, 0x60) ; /*configuration setup*/
                                   /* see note (1) */

       if  (protect_operation==PROTECT)  /*to protect the block*/
              writeToFlash (address, 0x01)  ;
       else if  (protect_operation==UNPROTECT)  /*to unprotect the block*/
              writeToFlash (address, 0xD0)  ;


              writeToFlash (address, 0x90) ;
                                 /*see note (1) */


       if  (readFlash (address) ! = protection_state_expected)
                 error_handler () ;
              /*Check the protection state (see Read Block Signature table )*/


       writeToFlash (address, 0xFF) ; /*Reset to Read Array mode*/
                          /*see note (1) */

}
```

AI12895

1. Any address within the bank can equally be used.

Figure 26. Protection Register Program flowchart and pseudo code

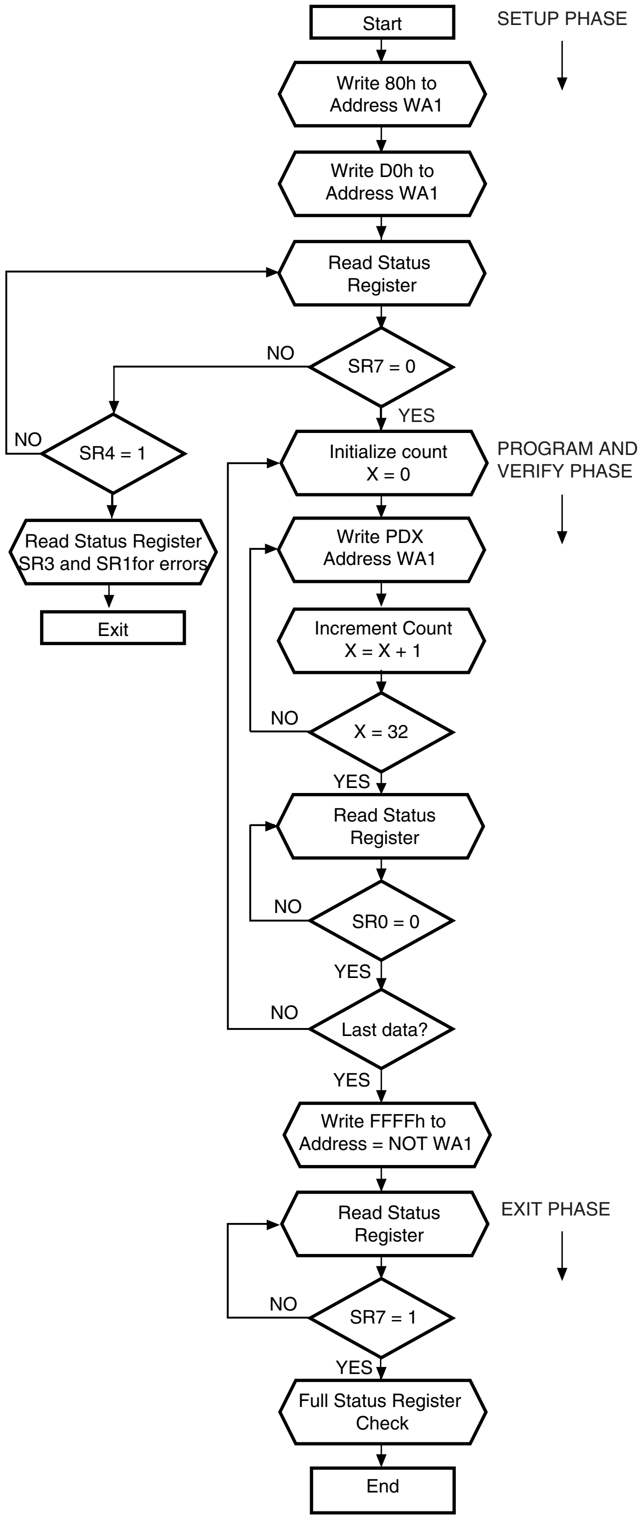



```
protection_register_program_command (addressToProgram, dataToProgram) {:
        writeToFlash (addressToProgram, 0xC0) ;
                /*see note (3) */

        writeToFlash (addressToProgram, dataToProgram) ;
        /*Memory enters read status state after
        the Program Command*/

    do {
        status_register=readFlash (addressToProgram) ;
                /* see note (3) */
        /* E or G must be toggled*/

    } while (status_register.SR7== 0) ;

if (status_register.SR3==1) /*VPP invalid error */
    error_handler ( ) ;

if (status_register.SR4==1) /*program error */
    error_handler ( ) ;

if (status_register.SR1==1) /*program to protect block error */
    error_handler ( ) ;

}
```

AI06177b

1. Status check of SR1 (protected block), SR3 (V_{PP} Invalid) and SR4 (program error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase Controller operations.
3. Any address within the bank can equally be used.

Figure 27. Buffer Enhanced Factory Program flowchart and pseudo code




```
Buffer_Enhanced_Factory_Program_Command
          (start_address, DataFlow[]) {

writeToFlash (start_address, 0x80) ;

writeToFlash (start_address, 0xD0) ;

do {
do {
status_register = readFlash (start_address);

if (status_register.SR4==1) { /*error*/
if (status_register.SR3==1) error_handler ( ) ;/*VPP error */
if (status_register.SR1==1) error_handler ( ) ;/* Protected Block */
}
while (status_register.SR7==1)
x=0; /* initialize count */
do {

writeToFlash (start_address, DataFlow[x]);

x++;

}while (x<32)
do {

status_register = readFlash (start_address);

}while (status_register.SR0==1)

} while (not last data)

writeToFlash (another_block_address, FFFFh)

do {
status_register = readFlash (start_address)

}while (status_register.SR7==0)

full_status_register_check();

 }
```

AI12898

Appendix D Command Interface state tables

Table 41. Command Interface states - modify table, next state[1]

<table>
<tr><th colspan="2" rowspan="2">Current CI State</th><th colspan="12">Command Input</th></tr>
<tr><th>Read Array[2] (FFh)</th><th>Program Setup[3][4] (10/40h)</th><th>Buffer Program [3][4] (E8h)</th><th>Block Erase, Setup[3][4] (20h)</th><th>BEFP Setup (80h)</th><th>Blank Check setup (BCh)</th><th>Erase Confirm P/E Resume, Block Unprotect confirm, BEFP Confirm[3][4] (D0h)</th><th>Blank Check confirm (CBh)</th><th>Buffer Program, Program/ Erase Suspend (B0h)</th><th>Read Status Register (70h)</th><th>Clear Status Register [5] (50h)</th><th>Read Electronic Signature , Read CFI Query (90h, 98h)</th></tr>
<tr><td colspan="2">Ready</td><td>Ready</td><td>Program Setup</td><td>BP Setup</td><td>Erase Setup</td><td>BEFP Setup</td><td>Blank Check setup</td><td colspan="6">Ready</td></tr>
<tr><td colspan="2">Protect/CR Setup</td><td colspan="6">Ready (Protect Error)</td><td>Ready (unprotect block)</td><td colspan="5">Ready (Protect Error)</td></tr>
<tr><td rowspan="3">OTP</td><td>Setup</td><td colspan="12">OTP Busy</td></tr>
<tr><td>Busy</td><td>OTP Busy</td><td>IS in OTP Busy</td><td>OTP busy</td><td colspan="2">IS in OTP Busy</td><td colspan="7">OTP Busy</td></tr>
<tr><td>IS in OTP busy</td><td colspan="12">OTP Busy</td></tr>
<tr><td rowspan="5">Program</td><td>Setup</td><td colspan="12">Program Busy</td></tr>
<tr><td>Busy</td><td>Program Busy</td><td>IS in Program Busy</td><td>Program Busy</td><td colspan="2">IS in Program Busy</td><td colspan="3">Program Busy</td><td>Program Suspend</td><td colspan="3">Program Busy</td></tr>
<tr><td>IS in Program Busy</td><td colspan="12">Program Busy</td></tr>
<tr><td>Suspend</td><td>PS</td><td>IS in PS</td><td>PS</td><td colspan="2">IS in Program Suspend</td><td>PS</td><td>Program Busy</td><td colspan="5">Program Suspend</td></tr>
<tr><td>IS in PS</td><td colspan="12">Program Suspend</td></tr>
<tr><td rowspan="8">Buffer Program</td><td>Setup</td><td colspan="12">Buffer Program Load 1 (give word count load (N-1));</td></tr>
<tr><td>Buffer Load 1</td><td colspan="12">if N=0 go to Buffer Program Confirm. Else (N ≠ 0) go to Buffer Program Load 2 (data load)</td></tr>
<tr><td>Buffer Load 2</td><td colspan="12">Buffer Program Confirm when count =0; Else Buffer Program Load 2
(note: Buffer Program fails at this point if any block address is different from the first address)</td></tr>
<tr><td>Confirm</td><td colspan="6">Ready (error)</td><td>BP Busy</td><td colspan="5">Ready (error)</td></tr>
<tr><td>Busy</td><td>BP Busy</td><td>IS in BP Busy</td><td>BP Busy</td><td colspan="2">IS in BP Busy</td><td colspan="3">BP Busy</td><td>BP Suspend</td><td colspan="3">Buffer Program Busy</td></tr>
<tr><td>IS in BP Busy</td><td colspan="12">Buffer Program Busy</td></tr>
<tr><td>Suspend</td><td>BP Suspend</td><td>IS in BP Suspend</td><td>BP Suspend</td><td colspan="2">IS in BP Suspend</td><td>BP Suspend</td><td>BP busy</td><td colspan="5">Buffer Program Suspend</td></tr>
<tr><td>IS in BP Suspend</td><td colspan="12">Buffer Program Suspend</td></tr>
</table>

Table 41. Command Interface states - modify table, next state[1] (continued)

Current CI State		Read Array[2] (FFh)	Program Setup[3][4] (10/40h)	Buffer Program[3][4] (E8h)	Block Erase, Setup[3][4] (20h)	BEFP Setup (80h)	Blank Check setup (BCh)	Erase Confirm P/E Resume, Block Unprotect confirm, BEFP Confirm[3][4] (D0h)	Blank Check confirm (CBh)	Buffer Program, Program/ Erase Suspend (B0h)	Read Status Register (70h)	Clear Status Register[5] (50h)	Read Electronic Signature, Read CFI Query (90h, 98h)
Erase	Setup	Ready (error)						Erase Busy	Ready (error)				
	Busy	Erase Busy	IS in Erase Busy	Erase Busy	IS in Erase Busy		Erase Busy			Erase Suspend	Erase Busy		
	IS in Erase Busy	Erase Busy											
	Suspend	Erase Suspend	Program in ES	BP in ES	IS in Erase Suspend		ES	Erase Busy	Erase Suspend				
	IS in ES	Erase Suspend											
Program in Erase Suspend	Setup	Program Busy in Erase Suspend											
	Busy	Program Busy in ES	IS in Program Busy in ES	Program Busy in ES	IS in Program Busy in ES		Program Busy in ES			PS in ES	Program Busy in Erase Suspend		
	IS in Program busy in ES	Program busy in Erase Suspend											
	Suspend	PS in ES	IS in PS in ES	PS in ES	IS in Program Suspend in ES		PS in ES	Program Busy in ES	Program Suspend in Erase Suspend				
	IS in PS in ES	Program Suspend in Erase Suspend											
Buffer Program in Erase Suspend	Setup	Buffer Program Load 1 in Erase Suspend (give word count load (N-1)); if N=0 go to Buffer Program confirm. Else (N ≠ 0) go to Buffer Program Load 2											
	Buffer Load 1	Buffer Program Load 2 in Erase Suspend (data load)											
	Buffer Load 2	Buffer Program Confirm in Erase Suspend when count =0; Else Buffer Program Load 2 in Erase Suspend (note: Buffer Program fails at this point if any block address is different from the first address)											
	Confirm	Erase Suspend (sequence error)						BP Busy in ES	Erase Suspend (sequence error)				
	Busy	BP Busy in ES	IS in BP Busy in ES	BP busy in ES	IS in BP busy in ES		BP Busy in ES			BP Suspend in ES	Buffer Program Busy in ES		
	IS in BP busy in ES	Buffer Program Busy in Erase Suspend											
	Suspend	BP Suspend in ES	IS in BP Suspend in ES	BP Suspend in ES	IS in BP Suspend in Erase Suspend		BP Suspend in ES	BP Busy in Erase Suspend	Buffer Program Suspend in Erase Suspend				
	IS in BP Suspend in ES	BP Suspend in Erase Suspend											

Table 41. Command Interface states - modify table, next state(1) (continued)

Current CI State		Command Input											
		Read Array(2) (FFh)	Program Setup(3)(4) (10/40h)	Buffer Program (3)(4) (E8h)	Block Erase, Setup(3)(4) (20h)	BEFP Setup (80h)	Blank Check setup (BCh)	Erase Confirm P/E Resume, Block Unprotect confirm, BEFP Confirm(3)(4) (D0h)	Blank Check confirm (CBh)	Buffer Program, Program/ Erase Suspend (B0h)	Read Status Register (70h)	Clear Status Register (5) (50h)	Read Electronic Signature , Read CFI Query (90h, 98h)
Blank Check	Setup	Ready (error)							Blank Check busy	Ready (error)			
	Busy	Blank Check busy	IS in Blank Check busy	Blank Check busy	IS in Blank Check busy		Blank Check busy						
Protect/CR Setup in Erase Suspend		Erase Suspend (Protect Error)						Erase Suspend	Erase Suspend (Protect Error)				
Buffer EFP	Setup	Ready (error)						BEFP Busy	Ready (error)				
	Busy	BEFP Busy(6)											

1. CI = Command Interface, CR = Configuration register, BEFP = Buffer Enhanced Factory program, P/E C = Program/Erase controller, IS = Illegal State, BP = Buffer Program, ES = Erase Suspend.
2. At power-up, all banks are in Read Array mode. Issuing a Read Array command to a busy bank, results in undetermined data output.
3. The two cycle command should be issued to the same bank address.
4. If the Program/Erase Controller is active, both cycles are ignored.
5. The Clear Status Register command clears the SR error bits except when the Program/Erase Controller is busy or suspended.
6. BEFP is allowed only when Status Register bit SR0 is reset to '0'. BEFP is busy if block address is first BEFP address. Any other commands are treated as data.

Table 42. Command Interface states - modify table, next output state[1] [2]

<table>
<tr><th rowspan="2">Current CI State</th><th colspan="12">Command Input</th></tr>
<tr><th>Read Array [3] (FFh)</th><th>Program Setup[4][5] (10/40h)</th><th>Buffer Program (E8h)</th><th>Block Erase, Setup[4][5] (20h)</th><th>BEFP Setup (80h)</th><th>Blank Check setup (BCh)</th><th>Erase Confirm P/E Resume, Block Unprotect confirm, BEFP Confirm[4][5] (D0h)</th><th>Blank Check confirm (CBh)</th><th>Program/ Erase Suspend (B0h)</th><th>Read Status Register (70h)</th><th>Clear Status Register (50h)</th><th>Read Electronic signature, Read CFI Query (90h, 98h)</th></tr>
<tr><td>Program Setup</td><td colspan="12" rowspan="17">Status Register</td></tr>
<tr><td>Erase Setup</td></tr>
<tr><td>OTP Setup</td></tr>
<tr><td>Program Setup in Erase Suspend</td></tr>
<tr><td>BEFP Setup</td></tr>
<tr><td>BEFP Busy</td></tr>
<tr><td>Buffer Program Setup</td></tr>
<tr><td>Buffer Program Load 1</td></tr>
<tr><td>Buffer Program Load 2</td></tr>
<tr><td>Buffer Program Confirm</td></tr>
<tr><td>Buffer Program Setup in Erase Suspend</td></tr>
<tr><td>Buffer Program Load 1 in Erase Suspend</td></tr>
<tr><td>Buffer Program Load 2 in Erase Suspend</td></tr>
<tr><td>Buffer Program Confirm in Erase Suspend</td></tr>
<tr><td>Blank Check setup</td></tr>
<tr><td>Protect/CR Setup</td></tr>
<tr><td>Protect/CR Setup in Erase Suspend</td></tr>
</table>

Table 42. Command Interface states - modify table, next output state(1) (2) (continued)

Current CI State	Command Input											
	Read Array (3) (FFh)	Program Setup(4)(5) (10/40h)	Buffer Program (E8h)	Block Erase, Setup(4)(5) (20h)	BEFP Setup (80h)	Blank Check setup (BCh)	Erase Confirm P/E Resume, Block Unprotect confirm, BEFP Confirm(4)(5) (D0h)	Blank Check confirm (CBh)	Program/ Erase Suspend (B0h)	Read Status Register (70h)	Clear Status Register (50h)	Read Electronic signature, Read CFI Query (90h, 98h)
OTP Busy	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Status Register
Ready	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Program Busy	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Erase Busy	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Buffer Program Busy	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Program/Erase Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Buffer Program Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Program Busy in Erase Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Buffer Program Busy in Erase Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Program Suspend in Erase Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Buffer Program Suspend in Erase Suspend	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Blank Check busy	Array	Status Register	Status Register	Status Register	Status Register	Status Register	Output Unchanged	Output Unchanged	Output Unchanged	Status Register	Output Unchanged	Electronic Signature/ CFI
Illegal State	Output Unchanged											

1. The output state shows the type of data that appears at the outputs if the bank address is the same as the command address. A bank can be placed in Read Array, Read Status Register, Read Electronic Signature or Read CFI mode, depending on the command issued. Each bank remains in its last output state until a new command is issued to that bank. The next state does not depend on the bank output state.
2. CI = Command Interface, CR = Configuration Register, BEFP = Buffer Enhanced Factory Program, Program/Erase Controller = Program/Erase Controller.
3. At power-up, all banks are in Read Array mode. Issuing a Read Array command to a busy bank, results in undetermined data output.
4. The two cycle command should be issued to the same bank address.
5. If the Program/Erase Controller is active, both cycles are ignored.

Table 43. Command interface states - lock table, next state[1]

Current CI State		Command Input						
		Protect/CR Setup[2] (60h)	OTP Setup[2] (C0h)	Block Protect Confirm (01h)	Set CR Confirm (03h)	Block Address (WA0)[3] (XXXXh)	Illegal Command[4]	P/E C operation completed[5]
Ready		Protect/CR Setup	OTP Setup	Ready				N/A
Protect/CR Setup		Ready (Protect error)		Ready		Ready (Protect error)		N/A
OTP	Setup	OTP Busy						N/A
	Busy	IS in OTP Busy		OTP Busy				Ready
	IS in OTP busy	OTP Busy						IS Ready
Program	Setup	Program Busy						N/A
	Busy	IS in Program Busy		Program Busy				Ready
	IS in Program busy	Program busy						IS Ready
	Suspend	IS in PS		Program Suspend				N/A
	IS in PS	Program Suspend						
Buffer Program	Setup	Buffer Program Load 1 (give word count load (N-1));						N/A
	Buffer Load 1	Buffer Program Load 2[6]				Exit	see note [6]	N/A
	Buffer Load 2	Buffer Program Confirm when count =0; Else Buffer Program Load 2 (note: Buffer Program fails at this point if any block address is different from the first address)						N/A
	Confirm	Ready (error)						N/A
	Busy	IS in BP Busy		Buffer Program Busy				Ready
	IS in Buffer Program busy	Buffer Program Busy						IS Ready
	Suspend	IS in BP Suspend		Buffer Program Suspend				N/A
	IS in BP Suspend	Buffer Program Suspend						
Erase	Setup	Ready (error)						N/A
	Busy	IS in Erase Busy		Erase Busy				Ready
	IS in Erase busy	Erase Busy						IS ready
	Suspend	Protect/CR Setup in ES	IS in ES	Erase Suspend				N/A
	IS in ES	Erase Suspend						

Table 43. Command interface states - lock table, next state[1] (continued)

Current CI State		Command Input: Protect/CR Setup[2] (60h)	OTP Setup[2] (C0h)	Block Protect Confirm (01h)	Set CR Confirm (03h)	Block Address (WA0)[3] (XXXXh)	Illegal Command[4]	P/E C operation completed[5]
Program in Erase Suspend	**Setup**	Program Busy in Erase Suspend						N/A
	Busy	IS in Program busy in ES		Program Busy in Erase Suspend				ES
	IS in Program busy in ES	Program Busy in Erase Suspend						IS in ES
	Suspend	IS in PS in ES		Program Suspend in Erase Suspend				N/A
	IS in PS in ES	Program Suspend in Erase Suspend						
Buffer Program in Erase Suspend	**Setup**	Buffer Program Load 1 in Erase Suspend (give word count load (N-1))						N/A
	Buffer Load 1	Buffer Program Load 2 in Erase Suspend[7]				Exit	see note [7]	
	Buffer Load 2	Buffer Program Confirm in Erase Suspend when count =0; Else Buffer Program Load 2 in Erase Suspend (note: Buffer Program fails at this point if any block address is different from the first address)						
	Confirm	Erase Suspend (sequence error)						
	Busy	IS in BP busy in ES		Buffer Program Busy in Erase Suspend				ES
	IS in BP busy in ES	BP busy in ES						IS in ES
	Suspend	IS in BP suspend in ES		Buffer Program Suspend in Erase Suspend				N/A
	IS in BP Suspend in ES	Buffer Program Suspend in Erase Suspend						
Blank Check	**Setup**	Ready (error)						N/A
	Blank Check busy	IS in Blank Check busy		Blank Check busy				Ready
Protect/CR Setup in ES		Erase Suspend (Protect error)		Erase Suspend		Erase Suspend (Protect error)		N/A
BEFP	**Setup**	Ready (error)						N/A
	Busy	BEFP Busy[8]				Exit	BEFP Busy[8]	N/A

1. CI = Command Interface, CR = Configuration register, BEFP = Buffer Enhanced Factory program, P/E C = Program/Erase controller, IS = Illegal State, BP = Buffer program, ES = Erase suspend, WA0 = Address in a block different from first BEFP address.
2. If the Program/Erase Controller is active, both cycle are ignored.
3. BEFP exit when block address is different from first block address and data are FFFFh.
4. Illegal commands are those not defined in the command set.
5. N/A: not available. In this case the state remains unchanged.
6. If N=0 go to Buffer Program Confirm. Else (not = 0) go to Buffer Program Load 2 (data load)
7. If N=0 go to Buffer Program Confirm in Erase suspend. Else (not =0) go to Buffer Program Load 2 in Erase suspend.
8. BEFP is allowed only when Status Register bit SR0 is set to '0'. BEFP is busy if Block Address is first BEFP Address. Any other commands are treated as data.

Table 44. Command interface states - lock table, next output state (1) (2)

<table>
<tr><th rowspan="2">Current CI State</th><th colspan="9">Command Input</th></tr>
<tr><th>Protect/CR Setup(3) (60h)</th><th>Blank Check setup (BCh)</th><th>OTP Setup(3) (C0h)</th><th>Blank Check confirm (CBh)</th><th>Block Protect Confirm (01h)</th><th>Set CR Confirm (03h)</th><th>BEFP Exit(4) (FFFFh)</th><th>Illegal Command(5)</th><th>P. E./C. Operation Completed</th></tr>
<tr><td>Program Setup</td><td colspan="8" rowspan="15">Status Register</td><td rowspan="17">Output Unchanged</td></tr>
<tr><td>Erase Setup</td></tr>
<tr><td>OTP Setup</td></tr>
<tr><td>Program Setup in Erase Suspend</td></tr>
<tr><td>BEFP Setup</td></tr>
<tr><td>BEFP Busy</td></tr>
<tr><td>Buffer Program Setup</td></tr>
<tr><td>Buffer Program Load 1</td></tr>
<tr><td>Buffer Program Load 2</td></tr>
<tr><td>Buffer Program Confirm</td></tr>
<tr><td>Buffer Program Setup in Erase Suspend</td></tr>
<tr><td>Buffer Program Load 1 in Erase Suspend</td></tr>
<tr><td>Buffer Program Load 2 in Erase Suspend</td></tr>
<tr><td>Buffer Program Confirm in Erase Suspend</td></tr>
<tr><td>Blank Check setup</td></tr>
<tr><td>Protect/CR Setup</td><td colspan="5" rowspan="2">Status Register</td><td rowspan="2">Array</td><td colspan="2" rowspan="2">Status Register</td></tr>
<tr><td>Protect/CR Setup in Erase Suspend</td></tr>
</table>

Table 44. Command interface states - lock table, next output state (continued)[(1) (2)]

<table>
<tr><th rowspan="2">Current CI State</th><th colspan="9">Command Input</th></tr>
<tr><th>Protect/CR Setup[(3)] (60h)</th><th>Blank Check setup (BCh)</th><th>OTP Setup[(3)] (C0h)</th><th>Blank Check confirm (CBh)</th><th>Block Protect Confirm (01h)</th><th>Set CR Confirm (03h)</th><th>BEFP Exit[(4)] (FFFFh)</th><th>Illegal Command[(5)]</th><th>P. E./C. Operation Completed</th></tr>
<tr><td>OTP Busy</td><td colspan="3" rowspan="12">Status Register</td><td colspan="3" rowspan="12">Output Unchanged</td><td rowspan="12">Array</td><td colspan="2" rowspan="12">Output Unchanged</td></tr>
<tr><td>Ready</td></tr>
<tr><td>Program Busy</td></tr>
<tr><td>Erase Busy</td></tr>
<tr><td>Buffer Program Busy</td></tr>
<tr><td>Program/Erase Suspend</td></tr>
<tr><td>Buffer Program Suspend</td></tr>
<tr><td>Program Busy in Erase Suspend</td></tr>
<tr><td>Buffer Program Busy in Erase Suspend</td></tr>
<tr><td>Program Suspend in Erase Suspend</td></tr>
<tr><td>Buffer Program Suspend in Erase Suspend</td></tr>
<tr><td>Blank Check busy</td></tr>
<tr><td>Illegal State</td><td colspan="9">Output Unchanged</td></tr>
</table>

1. The output state shows the type of data that appears at the outputs if the bank address is the same as the command address. A bank can be placed in Read Array, Read Status Register, Read Electronic Signature or Read CFI mode, depending on the command issued. Each bank remains in its last output state until a new command is issued to that bank. The next state does not depend on the bank's output state.
2. CI = Command Interface, CR = Configuration Register, BEFP = Buffer Enhanced Factory Program, P/E. C. = Program/Erase Controller.
3. If the Program/Erase Controller is active, both cycles are ignored.
4. BEFP Exit when block address is different from first block address and data are FFFFh.
5. Illegal commands are those not defined in the command set.

Revision history

Table 45. Document revision history

Date	Revision	Changes
02-Mar-2007	1	Initial release.
27-Jun-2007	2	– Modified WAIT de-assertion criteria *on page 45*. – Removed t_{ELTV} from waveforms in *Figure 11* and *Figure 13*, and from *Table 23*. – Modified the t_{VDHPH} parameter values in *Table 26*. – Added the T packaging option in part numbering information in *Table 28*.
10-Dec-2007	3	Applied Numonyx branding.

Please Read Carefully: