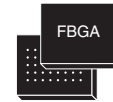


## Features

- Supply voltage
  - $V_{DD} = 1.7\text{ V to }2.0\text{ V}$  for program, erase and read
  - $V_{DDQ} = 1.7\text{ V to }2.0\text{ V}$  for I/O buffers
  - $V_{PP} = 9\text{ V}$  for fast program
- Synchronous/asynchronous read
  - Synchronous burst read mode: 54 MHz, 66 MHz
  - Asynchronous page read mode
  - Random access: 70 ns, 85 ns
- Synchronous burst read suspend
- Programming time
  - 2.5  $\mu\text{s}$  typical word program time using Buffer Enhanced Factory Program command
- Memory organization
  - Multiple bank memory array:  
8 Mbit banks for the M58LR128KT/B  
16 Mbit banks for the M58LR256KT/B
  - 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 locking
  - All blocks locked at power-up
  - Any combination of blocks can be locked with zero latency
  - $\overline{WP}$  for block lock-down
  - Absolute write protection with  $V_{PP} = V_{SS}$



VFPGA56 (ZB) 7.7 x 9 mm  
VFPGA79 (ZC) 9 x 11 mm  
TFPGA88 (ZQ) 8 x 10 mm

- Security
  - 64 bit unique device number
  - 21 12 bit user programmable OTP cells
- Common flash interface (CFI)
- 100,000 program/erase cycles per block
- Electronic signature
  - Manufacturer code: 20h
  - Top device codes:  
M58LR128KT: 88C4h  
M58LR256KT: 880Dh
  - Bottom device codes  
M58LR128KB: 88C5h  
M58LR256KB: 880Eh
- The M58LR128KT/B is available in the ECOPACK-compliant VFPGA56 package.
- The M58LR256KT/B is available in the ECOPACK-compliant VFPGA79 and TFPGA88 packages.

# Contents

- 1 Description . . . . . 9**
- 2 Signal descriptions . . . . . 18**
  - 2.1 Address inputs (A0-Amax) . . . . . 18
  - 2.2 Data inputs/outputs (DQ0-DQ15) . . . . . 18
  - 2.3 Chip Enable ( $\overline{E}$ ) . . . . . 18
  - 2.4 Output Enable ( $\overline{G}$ ) . . . . . 18
  - 2.5 Write Enable ( $\overline{W}$ ) . . . . . 18
  - 2.6 Write Protect ( $\overline{WP}$ ) . . . . . 18
  - 2.7 Reset ( $\overline{RP}$ ) . . . . . 19
  - 2.8 Latch Enable ( $\overline{L}$ ) . . . . . 19
  - 2.9 Clock (K) . . . . . 19
  - 2.10 Wait (WAIT) . . . . . 19
  - 2.11  $V_{DD}$  supply voltage . . . . . 19
  - 2.12  $V_{DDQ}$  supply voltage . . . . . 19
  - 2.13  $V_{PP}$  program supply voltage . . . . . 20
  - 2.14  $V_{SS}$  ground . . . . . 20
  - 2.15  $V_{SSQ}$  ground . . . . . 20
- 3 Bus operations . . . . . 21**
  - 3.1 Bus read . . . . . 21
  - 3.2 Bus write . . . . . 21
  - 3.3 Address Latch . . . . . 21
  - 3.4 Output disable . . . . . 21
  - 3.5 Standby . . . . . 22
  - 3.6 Reset . . . . . 22
- 4 Command interface . . . . . 23**
  - 4.1 Read Array command . . . . . 24
  - 4.2 Read Status Register command . . . . . 24
  - 4.3 Read Electronic Signature command . . . . . 25
  - 4.4 Read CFI Query command . . . . . 25

|          |   |           |
|----------|---|-----------|
| 4.5      | Clear Status Register command                     | 26        |
| 4.6      | Block Erase command                               | 26        |
| 4.7      | Blank Check command                               | 27        |
| 4.8      | Program command                                   | 28        |
| 4.9      | Buffer Program command                            | 28        |
| 4.10     | Buffer Enhanced Factory Program command           | 29        |
| 4.10.1   | Setup phase                                       | 30        |
| 4.10.2   | Program and verify phase                          | 30        |
| 4.10.3   | Exit phase  | 31        |
| 4.11     | Program/Erase Suspend command                     | 31        |
| 4.12     | Program/Erase Resume command                      | 32        |
| 4.13     | Protection Register Program command               | 32        |
| 4.14     | Set Configuration Register command                | 33        |
| 4.15     | Block Lock command                                | 33        |
| 4.16     | Block Unlock command                              | 34        |
| 4.17     | Block Lock-Down command                           | 34        |
| <b>5</b> | <b>Status Register</b>                            | <b>39</b> |
| 5.1      | Program/Erase Controller status bit (SR7)         | 39        |
| 5.2      | Erase suspend status bit (SR6)                    | 39        |
| 5.3      | Erase/blank check status bit (SR5)                | 40        |
| 5.4      | Program status bit (SR4)                          | 40        |
| 5.5      | V <sub>PP</sub> status bit (SR3)                  | 40        |
| 5.6      | Program suspend status bit (SR2)                  | 41        |
| 5.7      | Block protection status bit (SR1)                 | 41        |
| 5.8      | Bank write/multiple word program status bit (SR0) | 41        |
| <b>6</b> | <b>Configuration Register</b>                     | <b>43</b> |
| 6.1      | Read select bit (CR15)                            | 43        |
| 6.2      | X latency bits (CR13-CR11)                        | 43        |
| 6.3      | Wait polarity bit (CR10)                          | 44        |
| 6.4      | Data output configuration bit (CR9)               | 44        |
| 6.5      | Wait configuration bit (CR8)                      | 44        |
| 6.6      | Burst type bit (CR7)                              | 44        |

|                   |   |            |
|-------------------|---|------------|
| 6.7               | Valid Clock edge bit (CR6) .....                            | 45         |
| 6.8               | Wrap burst bit (CR3) .....                                  | 45         |
| 6.9               | Burst length bits (CR2-CR0) .....                           | 45         |
| <b>7</b>          | <b>Read modes .....</b>                                     | <b>51</b>  |
| 7.1               | Asynchronous read mode .....                                | 51         |
| 7.2               | Synchronous burst read mode .....                           | 52         |
| 7.2.1             | Synchronous burst read suspend .....                        | 53         |
| 7.3               | Single synchronous read mode .....                          | 53         |
| <b>8</b>          | <b>Dual operations and multiple bank architecture .....</b> | <b>54</b>  |
| <b>9</b>          | <b>Block locking .....</b>                                  | <b>56</b>  |
| 9.1               | Reading block lock status .....                             | 56         |
| 9.2               | Locked state .....  | 56         |
| 9.3               | Unlocked state .....  | 56         |
| 9.4               | Lock-down state .....                                       | 57         |
| 9.5               | Locking operations during erase suspend .....               | 57         |
| <b>10</b>         | <b>Program and erase times and endurance cycles .....</b>   | <b>59</b>  |
| <b>11</b>         | <b>Maximum ratings .....</b>                                | <b>61</b>  |
| <b>12</b>         | <b>DC and AC parameters .....</b>                           | <b>62</b>  |
| <b>13</b>         | <b>Package mechanical .....</b>                             | <b>78</b>  |
| <b>14</b>         | <b>Part numbering .....</b>                                 | <b>82</b>  |
| <b>Appendix A</b> | <b>Block address tables .....</b>                           | <b>84</b>  |
| <b>Appendix B</b> | <b>Common Flash interface .....</b>                         | <b>91</b>  |
| <b>Appendix C</b> | <b>Flowcharts and pseudocodes .....</b>                     | <b>101</b> |
| <b>Appendix D</b> | <b>Command interface state tables .....</b>                 | <b>110</b> |

**15**      **Revision history** ..... **119**

## List of tables

|           |   |    |
|-----------|---|----|
| Table 1.  | Signal names . . . . .  | 12 |
| Table 2.  | M58LR128KT/B bank architecture . . . . .  | 16 |
| Table 3.  | M58LR256KT/B bank architecture . . . . .  | 17 |
| Table 4.  | Bus operations . . . . .  | 22 |
| Table 5.  | Command codes . . . . .   | 23 |
| Table 6.  | Standard commands . . . . .   | 35 |
| Table 7.  | Factory commands . . . . .  | 36 |
| Table 8.  | Electronic signature codes . . . . .  | 36 |
| Table 9.  | Protection Register locks . . . . .   | 38 |
| Table 10. | Status Register bits . . . . .  | 42 |
| Table 11. | X latency settings . . . . .  | 43 |
| Table 12. | Configuration Register . . . . .  | 46 |
| Table 13. | Burst type definition . . . . .   | 47 |
| Table 14. | Dual operations allowed in other banks . . . . .  | 54 |
| Table 15. | Dual operations allowed in same bank . . . . .  | 55 |
| Table 16. | Dual operation limitations . . . . .  | 55 |
| Table 17. | Lock status . . . . .   | 58 |
| Table 18. | Program/erase times and endurance cycles . . . . .                                      | 59 |
| Table 19. | Absolute maximum ratings . . . . .  | 61 |
| Table 20. | Operating and AC measurement conditions . . . . .                                       | 62 |
| Table 21. | Capacitance . . . . .   | 63 |
| Table 22. | DC characteristics - currents . . . . .   | 64 |
| Table 23. | DC characteristics - voltages . . . . .   | 65 |
| Table 24. | Asynchronous read AC characteristics . . . . .  | 68 |
| Table 25. | Synchronous read AC characteristics . . . . .   | 72 |
| Table 26. | Write AC characteristics, Write Enable controlled . . . . .                             | 74 |
| Table 27. | Write AC characteristics, Chip Enable controlled . . . . .                              | 76 |
| Table 28. | Reset and power-up AC characteristics . . . . .   | 77 |
| Table 29. | VFBGA56 7.7 x 9 mm - 8 x 7 ball array, 0.75 mm pitch, package mechanical data . . . . . | 79 |
| Table 30. | TFBGA88 8 x 10 mm - 8 x 10 ball array, 0.8 mm pitch, package mechanical data . . . . .  | 80 |
| Table 31. | VFBGA79 9x11mm - 0.75 mm pitch, package mechanical data . . . . .                       | 81 |
| Table 32. | Ordering information scheme . . . . .   | 82 |
| Table 33. | M58LR128KT - Parameter bank block addresses . . . . .                                   | 85 |
| Table 34. | M58LR128KT - main bank base addresses . . . . .   | 85 |
| Table 35. | M58LR128KT - block addresses in main banks . . . . .                                    | 86 |
| Table 36. | M58LR256KT - parameter bank block addresses . . . . .                                   | 86 |
| Table 37. | M58LR256KT - main bank base addresses . . . . .   | 87 |
| Table 38. | M58LR256KT - block addresses in main banks . . . . .                                    | 87 |
| Table 39. | M58LR128KB - parameter bank block addresses . . . . .                                   | 88 |
| Table 40. | M58LR128KB - main bank base addresses . . . . .   | 88 |
| Table 41. | M58LR128KB - block addresses in main banks . . . . .                                    | 89 |
| Table 42. | M58LR256KB - parameter bank block addresses . . . . .                                   | 89 |
| Table 43. | M58LR256KB - main bank base addresses . . . . .   | 90 |
| Table 44. | M58LR256KB - block addresses in main banks . . . . .                                    | 90 |
| Table 45. | Query structure overview . . . . .  | 91 |
| Table 46. | CFI query identification string . . . . .   | 92 |
| Table 47. | CFI query system interface information . . . . .  | 93 |
| Table 48. | Device geometry definition . . . . .  | 94 |

---

|           |  |     |
|-----------|--|-----|
| Table 49. | Primary algorithm-specific extended query table            | 95  |
| Table 50. | Protection Register information                            | 96  |
| Table 51. | Burst read information                                     | 96  |
| Table 52. | Bank and erase block region information                    | 97  |
| Table 53. | Bank and erase block region 1 information                  | 97  |
| Table 54. | Bank and erase block region 2 information                  | 99  |
| Table 55. | Command interface states - modify table, next state        | 110 |
| Table 56. | Command interface states - modify table, next output state | 113 |
| Table 57. | Command interface states - lock table, next state          | 115 |
| Table 58. | Command interface states - lock table, next output state   | 117 |
| Table 59. | Document revision history                                  | 119 |

## List of figures

|            |  |     |
|------------|--|-----|
| Figure 1.  | Logic diagram . . . . .  | 11  |
| Figure 2.  | VFBGA56 package connections (top view through package) . . . . .                                     | 13  |
| Figure 3.  | TFBGA88 connections (top view through package) . . . . .   | 14  |
| Figure 4.  | M58LR128KT/B memory map . . . . .  | 16  |
| Figure 5.  | M58LR256KT/B memory map . . . . .  | 17  |
| Figure 6.  | Protection Register memory map . . . . .   | 37  |
| Figure 7.  | X latency and data output configuration example . . . . .  | 49  |
| Figure 8.  | Wait configuration example . . . . .   | 50  |
| Figure 9.  | AC measurement I/O waveform . . . . .  | 62  |
| Figure 10. | AC measurement load circuit . . . . .  | 63  |
| Figure 11. | Asynchronous random access read AC waveforms . . . . .   | 66  |
| Figure 12. | Asynchronous page read AC waveforms . . . . .  | 67  |
| Figure 13. | Synchronous burst read AC waveforms . . . . .  | 69  |
| Figure 14. | Single synchronous read AC waveforms . . . . .   | 70  |
| Figure 15. | Synchronous burst read suspend AC waveforms. . . . .   | 71  |
| Figure 16. | Clock input AC waveform . . . . .  | 72  |
| Figure 17. | Write AC waveforms, Write Enable controlled . . . . .  | 73  |
| Figure 18. | Write AC waveforms, Chip Enable controlled. . . . .  | 75  |
| Figure 19. | Reset and power-up AC waveforms . . . . .  | 77  |
| Figure 20. | VFBGA56 7.7 x 9 mm - 8 x 7 active ball array, 0.75 mm pitch, bottom view<br>package outline. . . . . | 78  |
| Figure 21. | TFBGA88 8 x 10 mm - 8 x 10 ball array, 0.8 mm pitch, bottom view package outline . . . . .           | 79  |
| Figure 22. | Drawing is not to scale. . . . .   | 79  |
| Figure 23. | VFBGA79 9x11mm - 0.75 pitch -package outline . . . . .   | 81  |
| Figure 24. | Program flowchart and pseudocode . . . . .   | 101 |
| Figure 25. | Blank check flowchart and pseudocode . . . . .   | 102 |
| Figure 26. | Buffer program flowchart and pseudocode . . . . .  | 103 |
| Figure 27. | Program suspend and resume flowchart and pseudocode . . . . .  | 104 |
| Figure 28. | Block erase flowchart and pseudocode . . . . .   | 105 |
| Figure 29. | Erase suspend and resume flowchart and pseudocode . . . . .  | 106 |
| Figure 30. | Locking operations flowchart and pseudocode. . . . .   | 107 |
| Figure 31. | Protection Register program flowchart and pseudocode . . . . .                                       | 108 |
| Figure 32. | Buffer enhanced factory program flowchart and pseudocode . . . . .                                   | 109 |



# 1 Description

The M58LR128KT/B and M58LR256KT/B are 128 Mbit (8 Mbit x16) and 256 Mbit (16 Mbit x 16) non-volatile Flash memories, respectively. They can 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 1.7 V to 2.0 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:

- The M58LR128KT/B have an array of 131 blocks, and are 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 M58LR256KT/B have an array of 259 blocks, and are divided into 16 Mbit banks. There are 15 banks each containing 16 main blocks of 64 Kwords, and one parameter bank containing 4 parameter blocks of 16 Kwords and 15 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 4](#). The parameter blocks are located at the top of the memory address space for the M58LR128KT and M58LR256KT, and at the bottom for the M58LR128KB and M58LR256KB.

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 speed up 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 66 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 automatic standby mode. In this condition the power consumption is reduced to the standby value and the outputs are still driven.

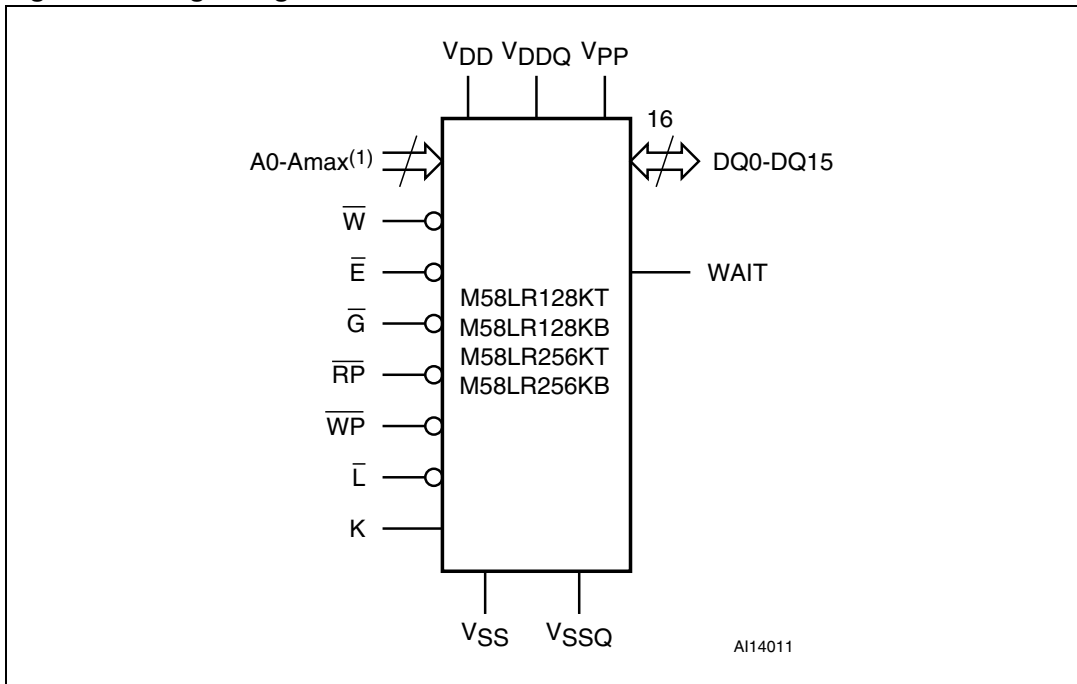
The M58LRxxxKT/B features an instant, individual block locking scheme that allows any block to be locked or unlocked with no latency, enabling instant code and data protection. All blocks have three levels of protection. They can be locked and locked-down individually preventing any accidental programming 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 locked 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 6](#), shows the Protection Register memory map.

The M58LR128KT/B is offered in a VFBGA56 7.7 x 9 mm, 0.75 mm package, and the M58LR256KT/B is offered in a VFBGA79 9 x 11 mm, 0.75 mm package and TFGBA88 8 x 10mm, 0.8 mm package.

All devices are supplied with all the bits erased (set to '1').

Figure 1. Logic diagram



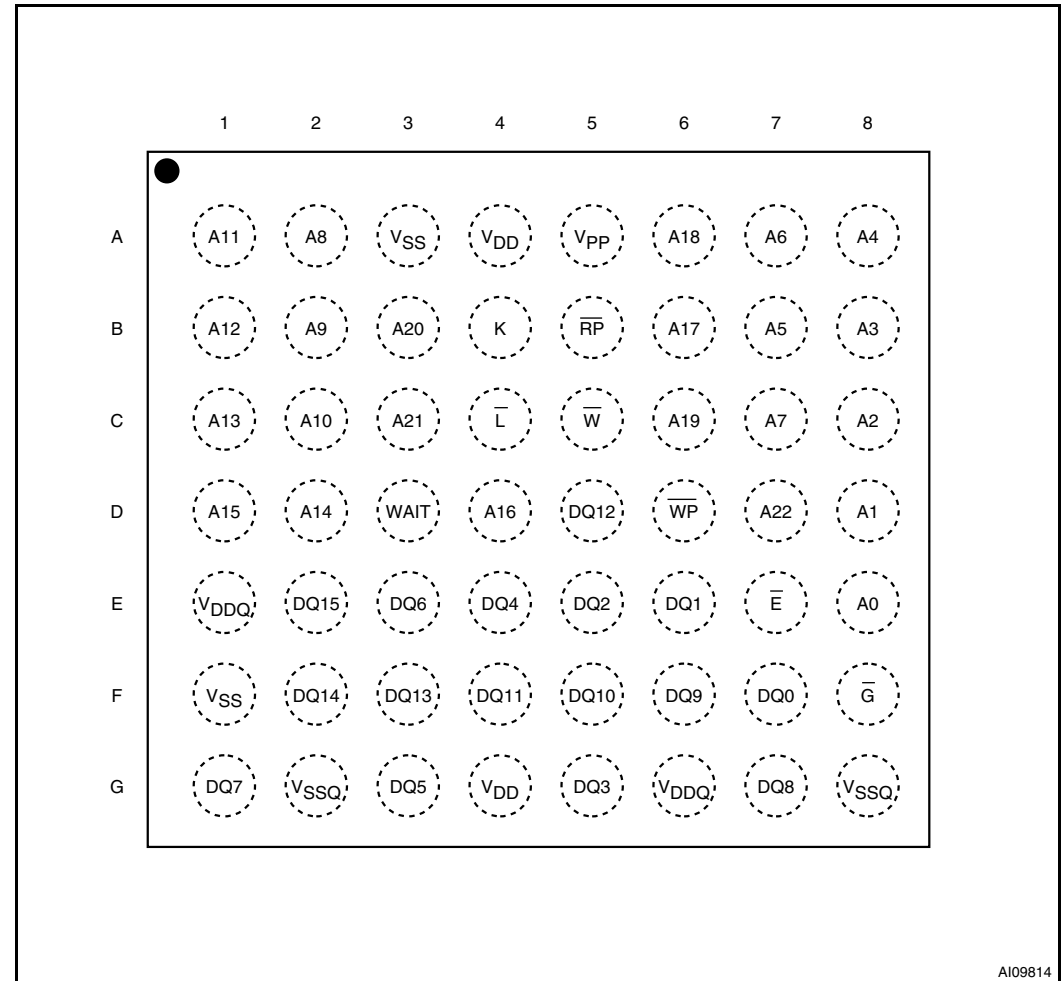
1. Amax is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B.

Table 1. Signal names

| Signal name            | Function   | Direction |
|------------------------|--|-----------|
| A0-Amax <sup>(1)</sup> | 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     |
| $\overline{WP}$        | Write Protect                                    | Input     |
| K                      | Clock  | Input     |
| $\overline{L}$         | Latch Enable                                     | Input     |
| WAIT                   | Wait   | Output    |
| V <sub>DD</sub>        | Supply voltage                                   |           |
| V <sub>DDQ</sub>       | Supply voltage for input/output buffers          |           |
| V <sub>PP</sub>        | Optional supply voltage for fast program & erase |           |
| V <sub>SS</sub>        | Ground   |           |
| V <sub>SSQ</sub>       | Ground input/output supply                       |           |

1. Amax is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B.

Figure 2. VFBGA56 package connections (top view through package)



1. This package is available only for the M58LR128KT/B devices.

Figure 3. TFBGA88 connections (top view through package)

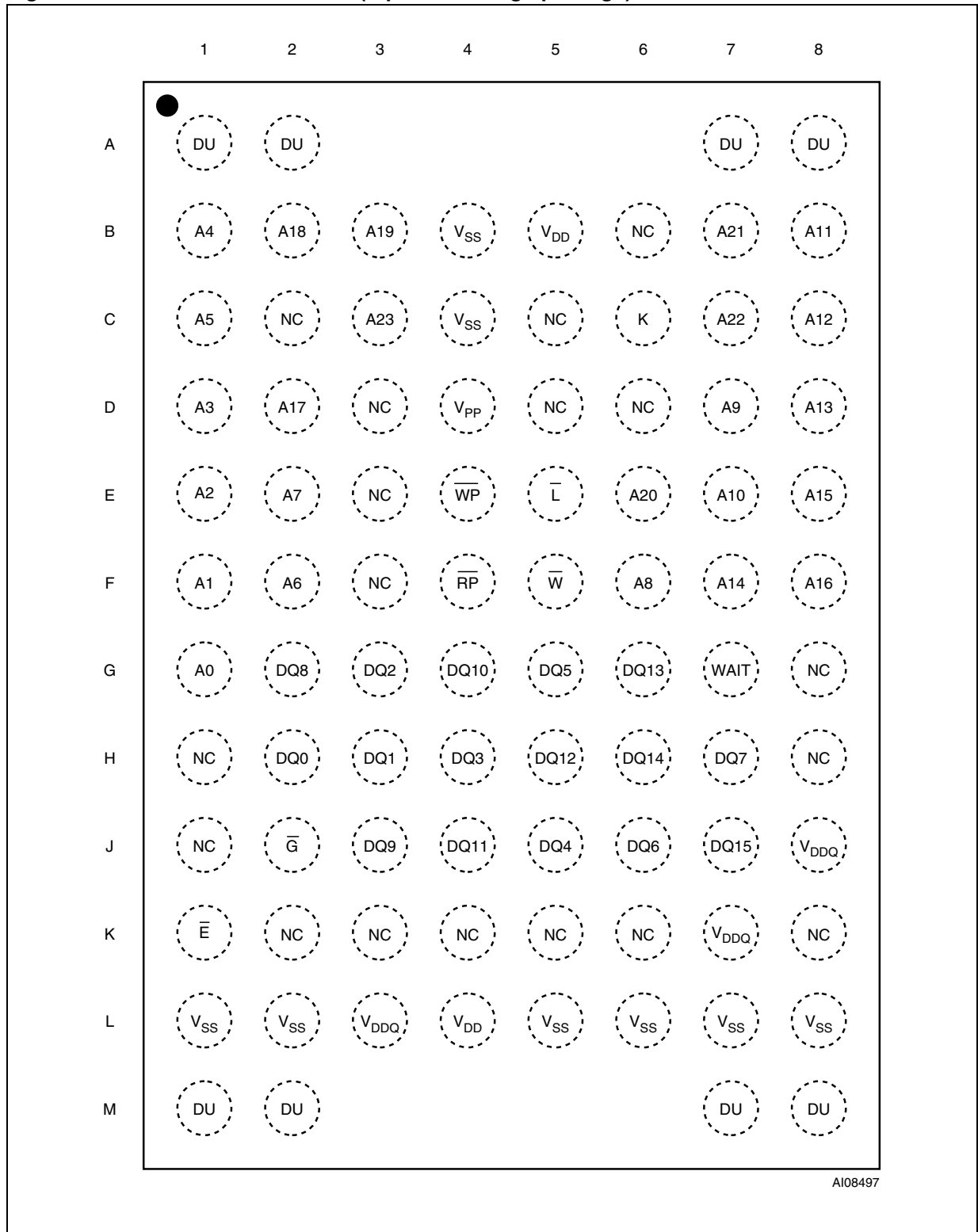
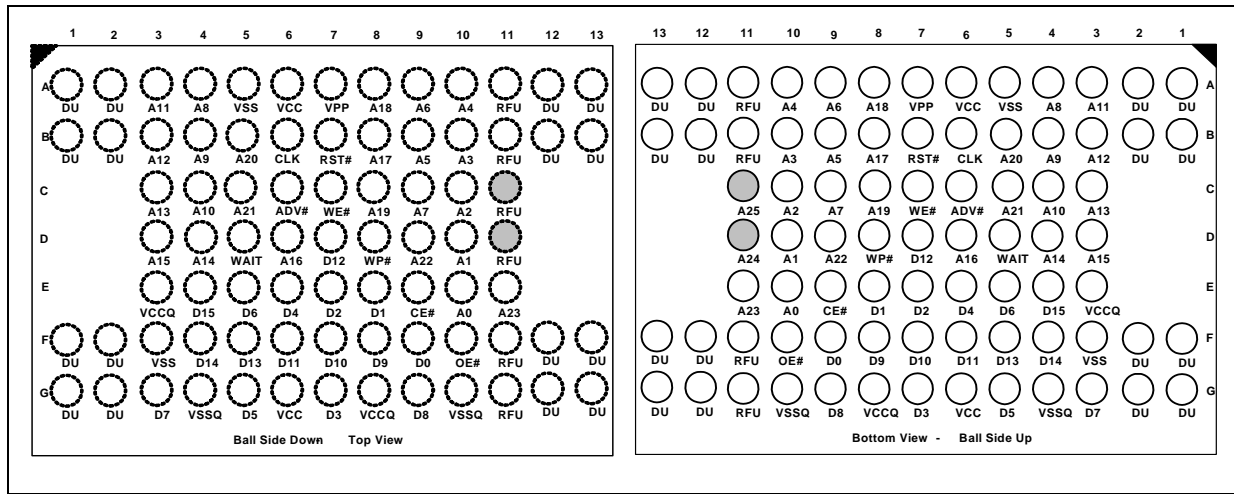


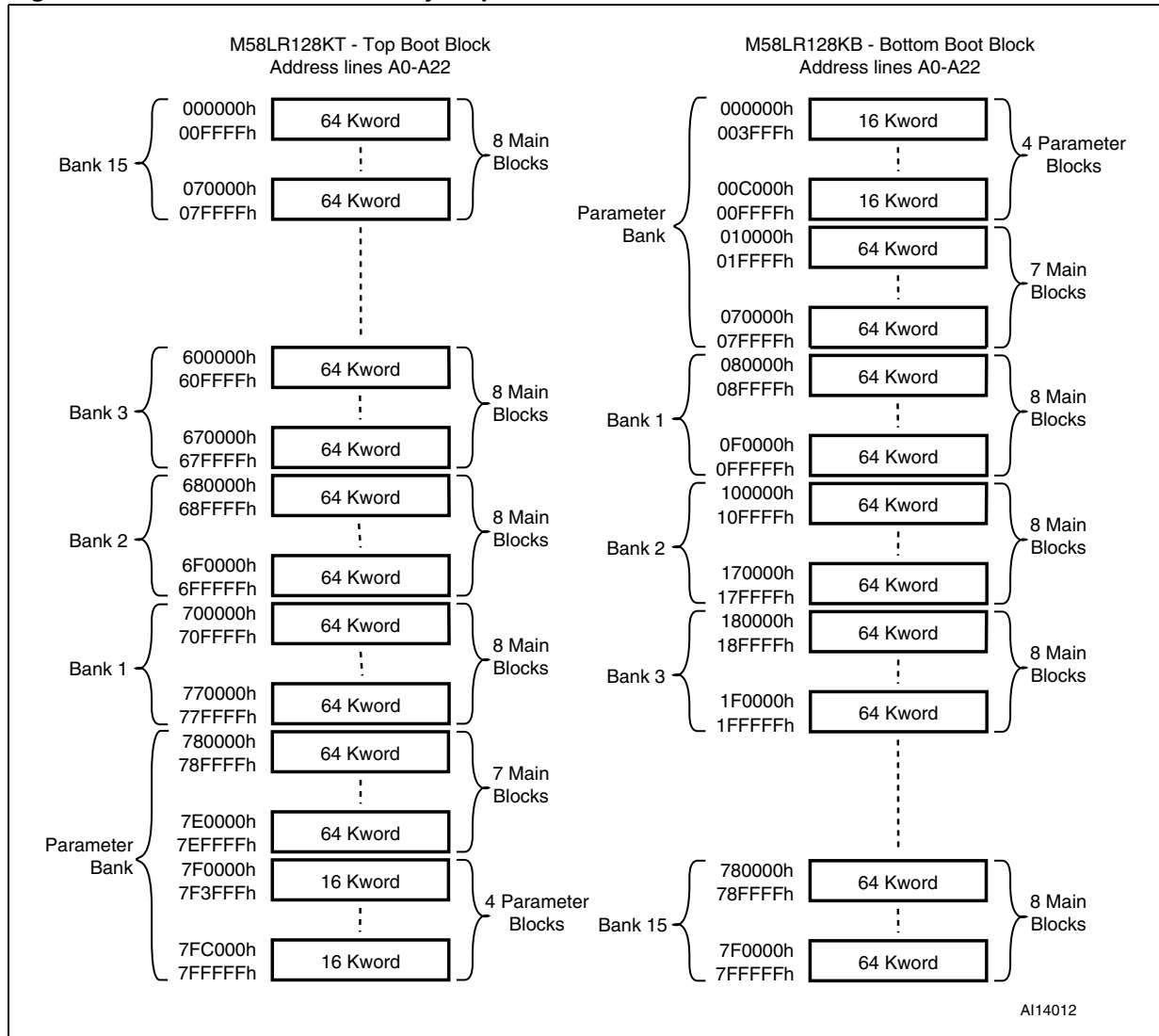
Figure 1: 7x9 Active-Ball Matrix for 256-Mbit Density in VF BGA Package



**Table 2. M58LR128KT/B 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 4. M58LR128KT/B memory map**

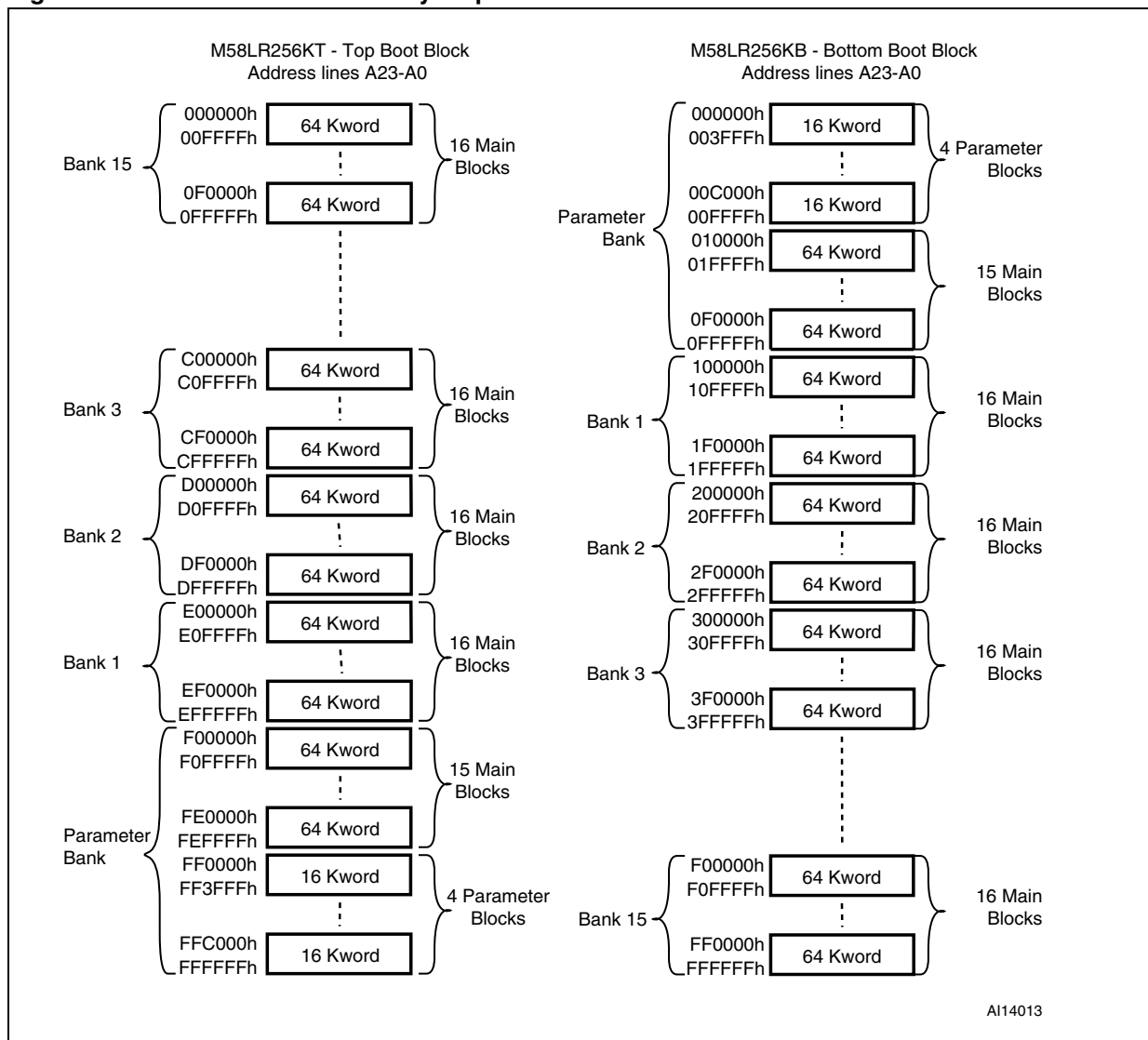




**Table 3. M58LR256KT/B bank architecture**

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

**Figure 5. M58LR256KT/B 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-Amax)

Amax is the highest order address input. It is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B. 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 Write Protect ( $\overline{WP}$ )

Write Protect is an input that gives an additional hardware protection for each block. When Write Protect is at  $V_{IL}$ , the lock-down is enabled and the protection status of the locked-down blocks cannot be changed. When Write Protect is at  $V_{IH}$ , the lock-down is disabled and the locked-down blocks can be locked or unlocked. (refer to [Table 17: Lock status](#)).

## 2.7 Reset ( $\overline{\text{RP}}$ )

The Reset input provides a hardware reset of the memory. When Reset is at  $V_{\text{IL}}$ , the memory is in reset mode: the outputs are high impedance and the current consumption is reduced to the Reset supply current  $I_{\text{DD2}}$ . Refer to [Table 22: DC characteristics - currents](#) for the value of  $I_{\text{DD2}}$ . After Reset all blocks are in the locked state and the Configuration Register is reset. When Reset is at  $V_{\text{IH}}$ , the device is in normal operation. Exiting reset mode the device enters asynchronous read mode, and a negative transition of Chip Enable or Latch Enable is required to ensure valid data outputs.

## 2.8 Latch Enable ( $\overline{\text{L}}$ )

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

## 2.9 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  $V_{\text{IL}}$ . Clock is ignored during asynchronous read and in write operations.

## 2.10 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  $V_{\text{IH}}$ , Output Enable is at  $V_{\text{IH}}$  or Reset is at  $V_{\text{IL}}$ . It can be configured to be active during the wait cycle or one clock cycle in advance.

## 2.11 $V_{\text{DD}}$ supply voltage

$V_{\text{DD}}$  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).

## 2.12 $V_{\text{DDQ}}$ supply voltage

$V_{\text{DDQ}}$  provides the power supply to the I/O pins and enables all outputs to be powered independently from  $V_{\text{DD}}$ .  $V_{\text{DDQ}}$  can be tied to  $V_{\text{DD}}$  or can use a separate supply.

## 2.13 $V_{PP}$ 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  $V_{PPLK}$  provides absolute protection against program or erase, while  $V_{PP}$  in the  $V_{PP1}$  range enables these functions (see Tables 22 and 23, 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.

## 2.14 $V_{SS}$ ground

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

## 2.15 $V_{SSQ}$ 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 0.1  $\mu$ F ceramic capacitor close to the pin (high frequency, inherently low inductance capacitors should be as close as possible to the package). See [Figure 10: 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 4: 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 should be used to enable the device, and Output Enable should be 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 11, 12 and 13](#) read AC waveforms, and [Tables 24 and 25](#) 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 17 and 18](#), write AC waveforms, and [Tables 26 and 27](#), 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 4. Bus operations<sup>(1)</sup>**

| Operation      | $\bar{E}$ | $\bar{G}$ | $\bar{W}$ | $\bar{L}$      | $\bar{RP}$ | WAIT <sup>(2)</sup> | 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 <sup>(4)</sup> |
| 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.  $\bar{L}$  can be tied to  $V_{IH}$  if the valid address has been previously latched.
4. Depends on  $\bar{G}$ .

## 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 manages 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.

The command interface is reset to read mode when power is first applied, when exiting from Reset or whenever  $V_{DD}$  is lower than  $V_{LKO}$ . Command sequences must be followed exactly. Any invalid combination of commands are ignored.

Refer to [Table 5: Command codes](#), [Table 6: Standard commands](#), [Table 7: Factory commands](#) and [Appendix D: Command interface state tables](#) for a summary of the command interface.

**Table 5. Command codes**

| Hex Code | Command  |
|----------|--|
| 01h      | Block Lock Confirm   |
| 03h      | Set Configuration Register Confirm   |
| 10h      | Alternative Program Setup  |
| 20h      | Block Erase Setup  |
| 2Fh      | Block Lock-Down Confirm  |
| 40h      | Program Setup  |
| 50h      | Clear Status Register  |
| 60h      | Block Lock Setup, Block Unlock Setup, Block Lock Down 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 Unlock 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 has finished. The read modes of other banks are not affected.

## 4.2 Read Status Register command

The device contains a Status Register that monitors program or erase operations.

The Read Status Register command reads 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 10](#) for the description of the Status Register bits.



### 4.3 Read Electronic Signature command

The Read Electronic Signature command reads the manufacturer and device codes, the lock 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 lock status of the addressed bank, the Protection Register, or the Configuration Register (see [Table 9](#)).

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 16: 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 goes into read electronic signature mode. Subsequent bus read cycles output the electronic signature data and the Program/Erase Controller continue 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 reads 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 16: Dual operation limitations](#) for details).

See [Appendix B: Common Flash interface](#) and Tables [45](#), [46](#), [47](#), [48](#), [49](#), [50](#), [51](#), [52](#), [53](#) and [54](#) for details on the information contained in the common Flash interface memory area.

## 4.5 Clear Status Register command

The Clear Status Register command resets (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 erases 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 does not change and the Status Register outputs the error.

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 commands; 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 [Section 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 18: Program/erase times and endurance cycles](#).

See [Appendix C, Figure 28: Block erase flowchart and pseudocode](#) for a suggested flowchart for using the Block Erase command.

## 4.7 Blank Check command

The Blank Check command checks whether a main array 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.

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 within the bank output the Status Register contents.

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 that 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 25: Blank check flowchart and pseudocode](#) for a suggested flowchart for using the Blank Check command.

Typical blank check times are given in [Table 18: Program/erase times and endurance cycles](#).

## 4.8 Program command

The Program command programs a single word to the memory array.

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

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 [Section 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 18: 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 24: Program flowchart and pseudocode](#) for the flowchart for using the Program command.

## 4.9 Buffer Program command

The Buffer Program Command uses the device's 32-word write buffer to speed up programming. Up to 32 words can be loaded into the write buffer. The Buffer Program command dramatically reduces 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 failing to follow the correct 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 Status Register bits SR4 and SR5 are set to '1', the Buffer Program command is not accepted. Clear the Status Register before re-issuing the command.

If the block being programmed is protected an error sets 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 commands; all other commands are ignored.

Refer to [Section 8: Dual operations and multiple bank architecture](#) for detailed information about simultaneous operations allowed in banks not being programmed.

See [Appendix C, Figure 26: Buffer program flowchart and pseudocode](#) 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 speed up 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 does not change, 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  $30^{\circ}\text{C} \pm 10^{\circ}\text{C}$
- The targeted block must be unlocked
- 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. Please refer to [Table 7: Factory commands](#) for detailed information.

#### 4.10.1 Setup phase

The Buffer Enhanced Factory Program command requires 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 or it is interpreted as data to program.

The Status Register P/EC bit SR7 should be read to check that the P/EC 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 [Section 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.

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 P/EC 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 P/EC 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 P/EC 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 have 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 P/EC 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 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 18](#).

See [Appendix C, Figure 32: Buffer enhanced factory program flowchart and pseudocode](#) for a suggested flowchart on using the Buffer Enhanced Factory Program command.

## 4.11 Program/Erase Suspend command

The Program/Erase Suspend command pauses 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 block or program-suspended word is not valid)
- Read Status Register
- Read Electronic Signature
- Read CFI Query
- Clear Status Register

Additionally, if the suspended operation was a block erase then the following commands are also accepted:

- Set Configuration Register
- Program (except in erase-suspended block)
- Buffer Program (except in erase suspended blocks)
- Block Lock
- Block Lock-Down
- Block Unlock.

During an erase suspend the block being erased can be protected by issuing the Block Lock or Block Lock-Down commands. When the Program/Erase Resume command is issued the operation completes.

It is possible to accumulate multiple suspend operations. For example, it is possible to 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 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 taking Chip Enable to  $V_{IH}$ . Program/erase is aborted if Reset, RP, goes to  $V_{IL}$ .

See [Appendix C, Figure 27: Program suspend and resume flowchart and pseudocode](#), and [Figure 29: Erase suspend and resume flowchart and pseudocode](#) for flowcharts for using the Program/Erase Suspend command.

## 4.12 Program/Erase Resume command

The Program/Erase Resume command restarts 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 27: Program suspend and resume flowchart and pseudocode](#), and [Figure 29: Erase suspend and resume flowchart and pseudocode](#) for flowcharts for using the Program/Erase Resume command.

## 4.13 Protection Register Program command

The Protection Register Program command programs the user OTP 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 6: 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'.

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 16: Dual operation limitations](#) for details)

The two Protection Register locks protect the OTP segments from further modification. The protection of the OTP segments is not reversible. Refer to [Figure 6: Protection Register memory map](#) and [Table 9: Protection Register locks](#) for details on the lock bits.

See [Appendix C, Figure 31: Protection Register program flowchart and pseudocode](#) for a flowchart for using the Protection Register Program command.

## 4.14 Set Configuration Register command

The Set Configuration Register command writes a new value to the Configuration Register.

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, that is  $A0 = CR0$ ,  $A1 = CR1$ , ...,  $A15 = CR15$ . Addresses  $A16$ - $A_{max}$  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 Lock command

The Block Lock command locks a block and prevent program or erase operations from changing the data in it. All blocks are locked after power-up or reset.

Two bus write cycles are required to issue the Block Lock command:

- The first bus cycle sets up the Block Lock command.
- The second bus write cycle latches the block address and locks the block.

The lock status can be monitored for each block using the Read Electronic Signature command. [Table 17](#) shows the lock status after issuing a Block Lock command.

Once set, the block lock bits remain set even after a hardware reset or power-down/power-up. They are cleared by a Block Unlock command.

Refer to [Section 9: Block locking](#) for a detailed explanation. See [Appendix C, Figure 30: Locking operations flowchart and pseudocode](#) for a flowchart for using the Lock command.

## 4.16 Block Unlock command

The Block Unlock command unlocks a block, allowing the block to be programmed or erased.

Two bus write cycles are required to issue the Block Unlock command:

- The first bus cycle sets up the Block Unlock command.
- The second bus write cycle latches the block address and unlocks the block.

The lock status can be monitored for each block using the Read Electronic Signature command. [Table 17](#) shows the protection status after issuing a Block Unlock command.

Refer to [Section 9: Block locking](#) for a detailed explanation and [Appendix C, Figure 30: Locking operations flowchart and pseudocode](#) for a flowchart for using the Block Unlock command.

## 4.17 Block Lock-Down command

The Block Lock-Down command is used to lock down a locked or unlocked block.

A locked-down block cannot be programmed or erased. The lock status of a locked-down block cannot be changed when  $\overline{WP}$  is low,  $V_{IL}$ . When  $\overline{WP}$  is high,  $V_{IH}$ , the lock-down function is disabled and the locked blocks can be individually unlocked by the Block Unlock command.

Two bus write cycles are required to issue the Block Lock-Down command:

- The first bus cycle sets up the Block Lock-Down command.
- The second bus write cycle latches the block address and locks down the block.

The lock status can be monitored for each block using the Read Electronic Signature command.

Locked-down blocks revert to the locked (and not locked-down) state when the device is reset on power-down. [Table 17](#) shows the lock status after issuing a Block Lock-Down command.

Refer to [Section 9: Block locking](#) for a detailed explanation and [Appendix C, Figure 30: Locking operations flowchart and pseudocode](#) for a flowchart for using the Lock-Down command.

Table 6. Standard commands<sup>(1)</sup>

| 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 <sup>(2)</sup> | SRD             |
| Read Electronic Signature     | 1+     | Write          | BKA                      | 90h               | Read      | BKA <sup>(2)</sup> | ESD             |
| Read CFI Query                | 1+     | Write          | BKA                      | 98h               | Read      | BKA <sup>(2)</sup> | QD              |
| Clear Status Register         | 1      | Write          | X                        | 50h               |           |                    |                 |
| Block Erase                   | 2      | Write          | BKA or BA <sup>(3)</sup> | 20h               | Write     | BA                 | D0h             |
| Program                       | 2      | Write          | BKA or WA <sup>(3)</sup> | 40h or 10h        | Write     | WA                 | PD              |
| Buffer Program <sup>(4)</sup> | n+4    | Write          | BA                       | E8h               | Write     | BA                 | n               |
|                               |        | Write          | PA <sub>1</sub>          | PD <sub>1</sub>   | Write     | PA <sub>2</sub>    | PD <sub>2</sub> |
|                               |        | Write          | PA <sub>n+1</sub>        | PD <sub>n+1</sub> | 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 Lock                    | 2      | Write          | BKA or BA <sup>(3)</sup> | 60h               | Write     | BA                 | 01h             |
| Block Unlock                  | 2      | Write          | BKA or BA <sup>(3)</sup> | 60h               | Write     | BA                 | D0h             |
| Block Lock-Down               | 2      | Write          | BKA or BA <sup>(3)</sup> | 60h               | Write     | BA                 | 2Fh             |

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 8](#).
3. Any address within the bank can be used.
4. n+1 is the number of words to be programmed.

**Table 7. Factory commands**

| Command                         | Phase                         | Cycles | Bus write operations <sup>(1)</sup> |                 |                 |                 |                 |                 |                 |                  |                 |                  |  |
|---------------------------------|-------------------------------|--------|-------------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|------------------|-----------------|------------------|--|
|                                 |                               |        | 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 <sup>(2)</sup>            | 80h             | WA <sub>1</sub> | D0h             |                 |                 |                 |                  |                 |                  |  |
|                                 | Program/verify <sup>(3)</sup> | ≥32    | WA <sub>1</sub>                     | PD <sub>1</sub> | WA <sub>1</sub> | PD <sub>2</sub> | WA <sub>1</sub> | PD <sub>3</sub> | WA <sub>1</sub> | PD <sub>31</sub> | WA <sub>1</sub> | PD <sub>32</sub> |  |
|                                 | Exit                          | 1      | NOT BA <sub>1</sub> <sup>(4)</sup>  | X               |                 |                 |                 |                 |                 |                  |                 |                  |  |

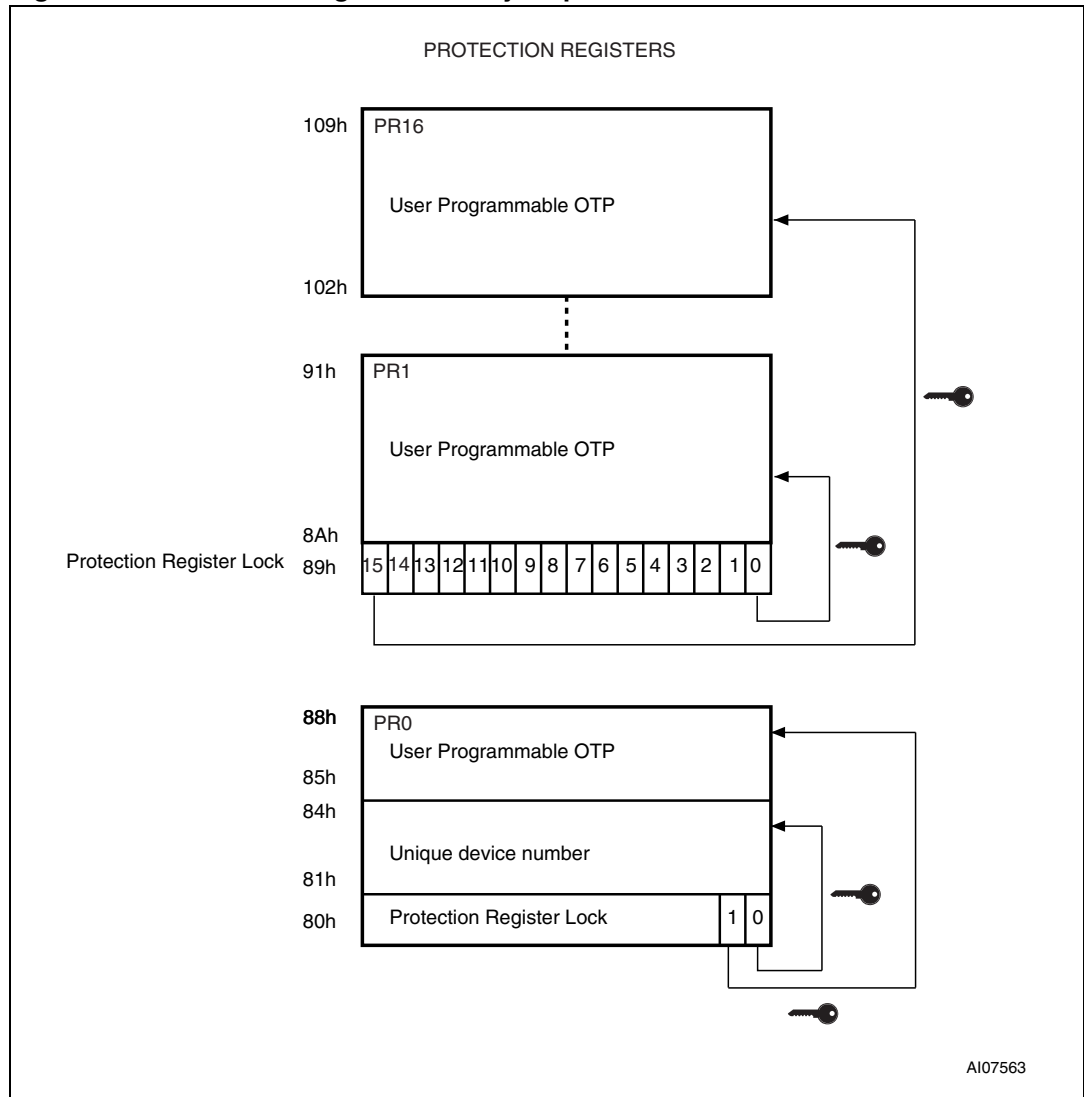
1. WA = Word Address in targeted bank, BKA = Bank Address, PD =Program Data, BA = Block Address, X = 'don't care'.
2. Any address within the bank can be used.
3. The program/verify phase can be executed any number of times as long as the data is to be programmed to the same block.
4. WA<sub>1</sub> is the start address, NOT BA<sub>1</sub> = Not Block Address of WA<sub>1</sub>.

**Table 8. Electronic signature codes**

| Code                                      |                             | Address (h)                             | Data (h)                               |
|---|-----------------------------|---|--|
| Manufacturer code                         |                             | Bank address + 00                       | 0020                                   |
| Device code                               | Top                         | Bank address + 01                       | 88C4 (M58LR128KT)<br>880D (M58LR256KT) |
|   | Bottom                      | Bank address + 01                       | 88C5 (M58LR128KB)<br>880E (M58LR256KB) |
| Block protection                          | Locked                      | Block address + 02                      | 0001                                   |
|   | Unlocked                    |   | 0000                                   |
|   | Locked and locked-down      |   | 0003                                   |
|   | Unlocked and locked-down    |   | 0002                                   |
| Configuration Register                    |                             | Bank address + 05                       | CR <sup>(1)</sup>                      |
| Protection Register PR0 lock              | Numonyx factory default     | Bank address + 80                       | 0002                                   |
|   | OTP area permanently locked |   | 0000                                   |
| Protection Register PR0                   |                             | Bank address + 81<br>Bank address + 84  | Unique device number                   |
|   |                             | Bank address + 85<br>Bank address + 88  | OTP Area                               |
| Protection Register PR1 through PR16 Lock |                             | Bank address + 89                       | PRLD <sup>(1)</sup>                    |
| Protection Registers PR1-PR16             |                             | Bank address + 8A<br>Bank address + 109 | OTP area                               |

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

Figure 6. Protection Register memory map



**Table 9. 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. Issue a Read Status Register command to read 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. Bus read operations from any address within the bank always read the Status Register during program and erase operations if no Read Array command has been issued.

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 on 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 10: Status Register bits](#). Refer to [Table 10](#) 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 in the addressed block. 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 identifies 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 identifies if there was an error during a program operation. It 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 identifies 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,  $V_{PPLK}$ , 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 in the addressed block. 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 the 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 identifies if a program or block erase operation has tried to modify the contents of a locked or locked-down block.

When the block protection status bit is High (set to '1'), a program or erase operation has been attempted on a locked or locked-down 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 bit 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 10. Status Register bits**

| Bit | Name  | Type   | Logic level <sup>(1)</sup> | Definition                              |   |
|-----|---|--------|----------------------------|---|---|
| SR7 | P/EC status   | Status | '1'                        | Ready                                   |   |
|     |   |        | '0'                        | Busy                                    |   |
| SR6 | Erase suspend status  | Status | '1'                        | Erase suspended                         |   |
|     |   |        | '0'                        | Erase In progress or completed          |   |
| SR5 | Erase/blank check status  | Error  | '1'                        | Erase/blank check error                 |   |
|     |   |        | '0'                        | Erase/blank check success               |   |
| SR4 | Program status  | Error  | '1'                        | Program error                           |   |
|     |   |        | '0'                        | Program success                         |   |
| SR3 | V <sub>PP</sub> status  | Error  | '1'                        | V <sub>PP</sub> invalid, abort          |   |
|     |   |        | '0'                        | V <sub>PP</sub> OK                      |   |
| SR2 | Program suspend status  | Status | '1'                        | Program suspended                       |   |
|     |   |        | '0'                        | Program in progress or completed        |   |
| SR1 | Block protection status   | Error  | '1'                        | Program/erase on protected block, abort |   |
|     |   |        | '0'                        | No operation to protected blocks        |   |
| SR0 | Bank write status   | Status | '1'                        | SR7 = '1'                               | Not allowed   |
|     |   |        |                            | SR7 = '0'                               | Program or erase operation in a bank other than the addressed bank                    |
|     |   |        | '0'                        | SR7 = '1'                               | No program or erase operation in the device   |
|     |   |        |                            | SR7 = '0'                               | Program or erase operation in addressed bank  |
|     | Multiple word program status (buffer enhanced factory program mode) | Status | '1'                        | SR7 = '1'                               | Not allowed   |
|     |   |        |                            | SR7 = '0'                               | The device is not ready for the next buffer loading or is going to exit the BEFP mode |
|     |   |        | '0'                        | SR7 = '1'                               | The device has exited the BEFP mode   |
|     |   |        |                            | SR7 = '0'                               | The device is ready for the next buffer loading                                       |

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

## 6 Configuration Register

The Configuration Register configures the type of bus access that the memory performs. Refer to [Section 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 12](#). They specify the selection of the burst length, burst type, burst X latency and the read operation. Refer to [Figures 7 and 8](#) for examples of synchronous burst configurations.

### 6.1 Read select bit (CR15)

The read select bit, CR15, switches between asynchronous and synchronous read operations.

When the read select bit is set to '1', read operations are asynchronous, and 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 7: X latency and data output configuration example](#).

For correct operation the X latency bits can only assume the values in [Table 12: Configuration Register](#).

[Table 11](#) 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 11. X latency settings**

| fmax   | t <sub>k</sub> min | X latency min |
|--------|--------------------|---------------|
| 30 MHz | 33 ns              | 3             |
| 40 MHz | 25 ns              | 4             |
| 54 MHz | 19 ns              | 5             |
| 66 MHz | 15 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 are 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 configures 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, and 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 7: 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'), as the memory outputs from sequential addresses only.

See [Table 13: 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, configures 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, selects 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 sets 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 asserted only once during a continuous burst access. See also [Table 13: Burst type definition](#).

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

**Table 12. 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 <sup>(1)</sup>   |
|           |                           | 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) <sup>(1)</sup>                  |
| CR8       | Wait configuration        | 0                             | WAIT is active during WAIT state   |
|           |                           | 1                             | WAIT is active one data cycle before WAIT state <sup>(1)</sup> (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 13. Burst type definition**

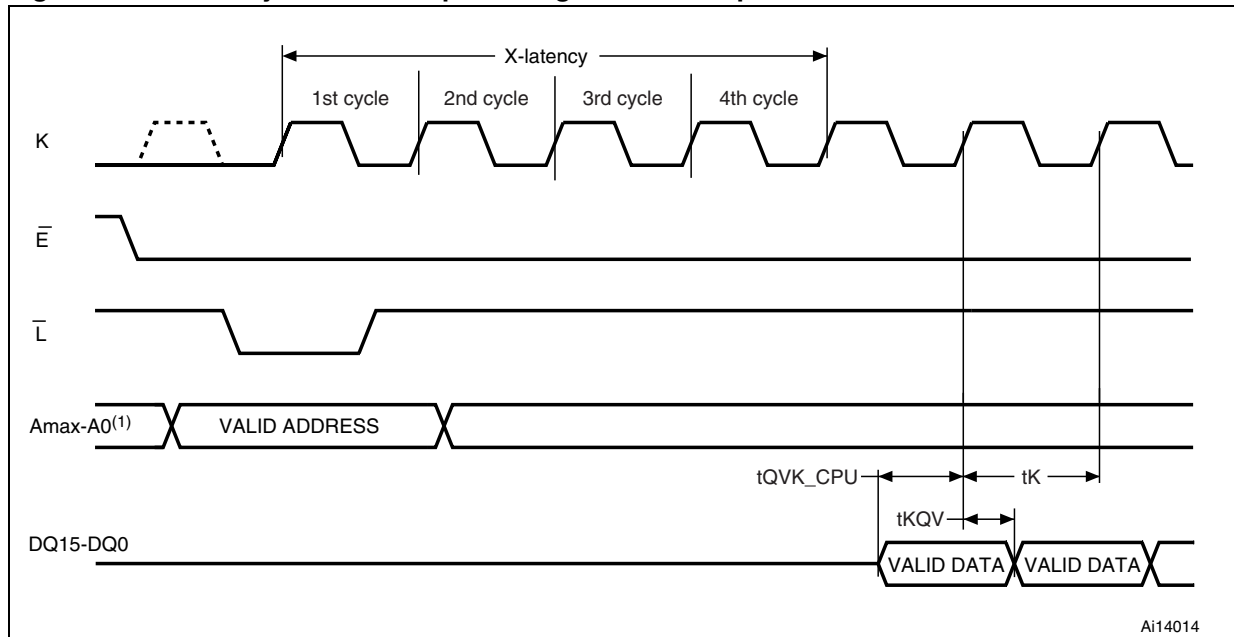
| Mode | Start addr | Sequential |                 |                                       | Continuous burst                                |
|------|------------|------------|-----------------|---------------------------------------|---|
|      |            | 4 words    | 8 words         | 16 words                              |   |
| 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 13. Burst type definition (continued)**

| Mode    | Start addr                 | 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<br>(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 |   |   |

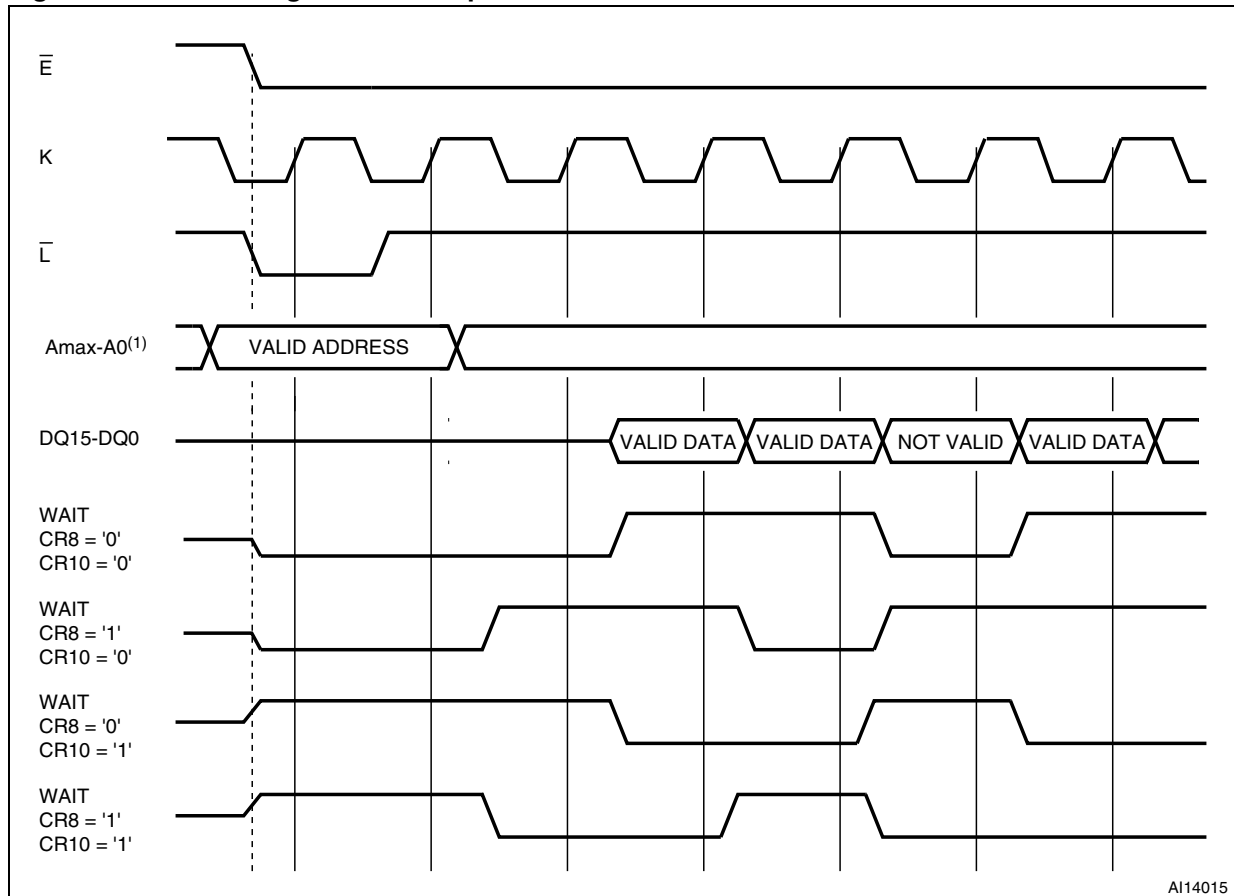


Figure 7. X latency and data output configuration example



1. Amax is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B.
2. The settings shown are X latency = 4, data output held for one clock cycle.

Figure 8. Wait configuration example



AI14015

1. Amax is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B.

## 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 asynchronous 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 automatic standby mode. In this condition the power consumption is reduced to the standby value and the outputs are still driven.

In asynchronous read mode, the WAIT signal is always deasserted.

See [Table 24: Asynchronous read AC characteristics](#), [Figure 11: Asynchronous random access read AC waveforms](#) and [Figure 12: 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, 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 will occur. 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 are 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 25: Synchronous read AC characteristics](#) and [Figure 13: 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 is resumed exactly where it stopped.

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

See [Table 25: Synchronous read AC characteristics](#) and [Figure 15: 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 the 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 25: Synchronous read AC characteristics](#) and [Figure 14: Single synchronous read AC waveforms](#) for details.

## 8 Dual operations and multiple bank architecture

The multiple bank architecture of the M58LRxxxKT/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.

In addition, if the suspended operation is 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 M58LRxxxKT/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 16](#) shows which dual operations are allowed or not between the CFI, the OTP, the electronic signature locations and the memory array.

Tables [14](#) and [15](#) show the dual operations possible in other banks and in the same bank.

**Table 14. 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 15. 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       | – <sup>(1)</sup>              | Yes                  | Yes            | Yes                       | –                       | –           | Yes                    | –                     |
| Erasing           | – <sup>(1)</sup>              | Yes                  | Yes            | Yes                       | –                       | –           | Yes                    | –                     |
| Program suspended | Yes <sup>(2)</sup>            | Yes                  | Yes            | Yes                       | –                       | –           | –                      | Yes                   |
| Erase suspended   | Yes <sup>(2)</sup>            | Yes                  | Yes            | Yes                       | Yes <sup>(1)</sup>      | –           | –                      | Yes                   |

1. The Read Array command is accepted but the data output is not guaranteed until the program or erase has completed.

2. Not allowed in the block that is being erased or in the word that is being programmed.

**Table 16. 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 locking

The M58LRxxxKT/B features an instant, individual block locking scheme that allows any block to be locked or unlocked with no latency. This locking scheme has three levels of protection.

- Lock/unlock - this first level allows software only control of block locking.
- Lock-down - this second level requires hardware interaction before locking can be changed.
- $V_{PP} \leq V_{PPLK}$  - the third level offers complete hardware protection against program and erase on all blocks.

The protection status of each block can be set to locked, unlocked, and locked-down. [Table 17](#), defines all of the possible protection states (WP, DQ1, DQ0), and [Appendix C](#), [Figure 30](#) shows a flowchart for the locking operations.

### 9.1 Reading block lock status

The lock status of every block can be read in 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 8](#) output the protection status of that block.

The lock status is represented by DQ0 and DQ1. DQ0 indicates the block lock/unlock status and is set by the Lock command and cleared by the Unlock command. DQ0 is automatically set when entering lock-down. DQ1 indicates the lock-down status and is set by the Lock-Down command. DQ1 cannot be cleared by software, except a hardware reset or power-down.

The following sections explain the operation of the locking system.

### 9.2 Locked state

The default status of all blocks on power-up or after a hardware reset is Locked (states (0,0,1) or (1,0,1)). Locked blocks are fully protected from program or erase operations. Any program or erase operations attempted on a locked block will return an error in the Status Register. The status of a locked block can be changed to unlocked or locked-down using the appropriate software commands. An unlocked block can be locked by issuing the Lock command.

### 9.3 Unlocked state

Unlocked blocks (states (0,0,0), (1,0,0) (1,1,0)), can be programmed or erased. All unlocked blocks return to the locked state after a hardware reset or when the device is powered-down. The status of an unlocked block can be changed to locked or locked-down using the appropriate software commands. A locked block can be unlocked by issuing the Unlock command.



## 9.4 Lock-down state

Blocks that are locked-down (state (0,1,x)) are protected from program and erase operations (as for locked blocks) but their protection status cannot be changed using software commands alone. A locked or unlocked block can be locked down by issuing the Lock-Down command. Locked-down blocks revert to the locked state when the device is reset or powered-down.

The lock-down function is dependent on the Write Protect,  $\overline{WP}$ , input pin.

When  $\overline{WP}=0$  ( $V_{IL}$ ), the blocks in the lock-down state (0,1,x) are protected from program, erase and protection status changes.

When  $\overline{WP}=1$  ( $V_{IH}$ ) the lock-down function is disabled (1,1,x) and locked-down blocks can be individually unlocked to the (1,1,0) state by issuing the software command, where they can be erased and programmed.

When the lock-down function is disabled ( $\overline{WP}=1$ ) blocks can be locked (1,1,1) and unlocked (1,1,0) as desired. When  $\overline{WP}=0$  blocks that were previously locked-down return to the lock-down state (0,1,x) regardless of any changes that were made while  $\overline{WP}=1$ .

Device reset or power-down resets all blocks, including those in lock-down, to the locked state.

## 9.5 Locking operations during erase suspend

Changes to block lock status can be performed during an erase suspend by using the standard locking command sequences to unlock, lock or lock-down a block. This is useful in the case when another block needs to be updated while an erase operation is in progress.

To change block locking 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 Lock command sequence to a block and the lock status is changed. After completing any desired lock, read, or program operations, resume the erase operation with the Erase Resume command.

If a block is locked or locked down during an erase suspend of the same block, the locking status bits is changed immediately, but when the erase is resumed, the erase operation completes. Locking operations cannot be performed during a program suspend.

Table 17. Lock status

| Current protection status <sup>(1)</sup><br>( $\overline{WP}$ , DQ1, DQ0) |                       | Next protection status <sup>(1)</sup><br>( $\overline{WP}$ , DQ1, DQ0) |                            |                               |                                  |
|---|-----------------------|--|----------------------------|-------------------------------|----------------------------------|
| Current state   | Program/erase allowed | After Block Lock command   | After Block Unlock command | After Block Lock-Down command | After $\overline{WP}$ transition |
| 1,0,0   | yes                   | 1,0,1  | 1,0,0                      | 1,1,1                         | 0,0,0                            |
| 1,0,1 <sup>(2)</sup>  | no                    | 1,0,1  | 1,0,0                      | 1,1,1                         | 0,0,1                            |
| 1,1,0   | yes                   | 1,1,1  | 1,1,0                      | 1,1,1                         | 0,1,1                            |
| 1,1,1   | no                    | 1,1,1  | 1,1,0                      | 1,1,1                         | 0,1,1                            |
| 0,0,0   | yes                   | 0,0,1  | 0,0,0                      | 0,1,1                         | 1,0,0                            |
| 0,0,1 <sup>(2)</sup>  | no                    | 0,0,1  | 0,0,0                      | 0,1,1                         | 1,0,1                            |
| 0,1,1   | no                    | 0,1,1  | 0,1,1                      | 0,1,1                         | 1,1,1 or 1,1,0 <sup>(3)</sup>    |

1. The lock status is defined by the write protect pin and by DQ1 ('1' for a locked-down block) and DQ0 ('1' for a locked block) as read in the Read Electronic Signature command with  $DQ1 = V_{IH}$  and  $DQ0 = V_{IL}$ .
2. All blocks are locked at power-up, so the default configuration is 001 or 101 according to  $\overline{WP}$  status.
3. A  $\overline{WP}$  transition to  $V_{IH}$  on a locked block will restore the previous DQ0 value, giving a 111 or 110.

## 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 18](#). 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' (pre-programmed). 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 M58LRxxxKT/B the maximum number of program/erase cycles depends on the  $V_{PP}$  voltage supply used.

**Table 18. Program/erase times and endurance cycles<sup>(1) (2)</sup>**

| Parameter         |                                  | Condition                          | Min               | Typ     | Typical after 100kW/E cycles | Max    | Unit    |         |
|-------------------|----------------------------------|------------------------------------|-------------------|---------|------------------------------|--------|---------|---------|
| $V_{PP} = V_{DD}$ | Erase                            | Parameter block (16 Kword)         |                   | 0.6     | 1                            | 2.5    | s       |         |
|                   |                                  | Main block (64 Kword)              | Preprogrammed     |         | 1.2                          | 3      | 4       | s       |
|                   |                                  |                                    | Not preprogrammed |         | 1.5                          |        | 4       | s       |
|                   | Program <sup>(3)</sup>           | Single Word                        | Word program      |         | 12                           |        | 180     | $\mu$ s |
|                   |                                  |                                    | Buffer program    |         | 12                           |        | 180     | $\mu$ s |
|                   |                                  | Buffer (32 words) (buffer program) |                   |         | 384                          |        |         | $\mu$ s |
|                   |                                  | Main block (64 Kword)              |                   |         | 768                          |        |         | ms      |
|                   | Suspend latency                  | Program                            |                   | 20      |                              | 25     | $\mu$ s |         |
|                   |                                  | Erase                              |                   | 20      |                              | 25     | $\mu$ s |         |
|                   | Program/erase cycles (per block) | Main blocks                        |                   | 100 000 |                              |        |         | cycles  |
| Parameter blocks  |                                  | 100 000                            |                   |         |                              | cycles |         |         |

**Table 18. Program/erase times and endurance cycles<sup>(1) (2)</sup> (continued)**

| Parameter                        | Condition                  |  | Min | Typ  | Typical after 100kW/E cycles | Max  | Unit   |
|----------------------------------|----------------------------|--|-----|------|------------------------------|------|--------|
| Erase                            | Parameter block (16 Kword) |  |     | 0.6  |                              | 2.5  | s      |
|                                  | Main block (64 Kword)      |  |     | 1    |                              | 4    | s      |
| Program <sup>(3)</sup>           | Single word                | Word program                                   |     | 10   |                              | 170  | μs     |
|                                  |                            | Buffer enhanced factory program <sup>(4)</sup> |     | 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.  $T_A = -30$  to  $85$  °C;  $V_{DD} = 1.7$  V to  $2$  V;  $V_{DDQ} = 1.7$  V to  $2$  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 ratings

Stressing the device above the rating listed in the [Table 19: Absolute maximum ratings](#) 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 also to the Numonyx SURE program and other relevant quality documents.

**Table 19. Absolute maximum ratings**

| Symbol     | Parameter                      | Value |                 | Unit  |
|------------|--------------------------------|-------|-----------------|-------|
|            |                                | Min   | Max             |       |
| $T_A$      | Ambient operating temperature  | -30   | 85              | °C    |
| $T_{BIAS}$ | Temperature under bias         | -30   | 85              | °C    |
| $T_{STG}$  | Storage temperature            | -65   | 125             | °C    |
| $V_{IO}$   | Input or output voltage        | -0.5  | $V_{DDQ} + 0.6$ | V     |
| $V_{DD}$   | Supply voltage                 | -0.2  | 2.5             | V     |
| $V_{DDQ}$  | Input/output supply voltage    | -0.2  | 2.5             | 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 in this section are derived from tests performed under the measurement conditions summarized in [Table 20: 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 20. Operating and AC measurement conditions**

| Parameter  | M58LRxxxKT/B          |                       |                       |                       | Unit |
|--|-----------------------|-----------------------|-----------------------|-----------------------|------|
|  | 70 ns                 |                       | 85 ns                 |                       |      |
|  | Min                   | Max                   | Min                   | Max                   |      |
| V <sub>DD</sub> supply voltage                           | 1.7                   | 2.0                   | 1.7                   | 2.0                   | V    |
| V <sub>DDQ</sub> supply voltage                          | 1.7                   | 2.0                   | 1.7                   | 2.0                   | V    |
| V <sub>PP</sub> supply voltage (factory environment)     | 8.5                   | 9.5                   | 8.5                   | 9.5                   | V    |
| V <sub>PP</sub> supply voltage (application environment) | -0.4                  | V <sub>DDQ</sub> +0.4 | -0.4                  | V <sub>DDQ</sub> +0.4 | V    |
| Ambient operating temperature                            | -30                   | 85                    | -30                   | 85                    | °C   |
| Load capacitance (C <sub>L</sub> )                       | 30                    |                       | 30                    |                       | pF   |
| Input rise and fall times                                |                       | 5                     |                       | 5                     | ns   |
| Input pulse voltages                                     | 0 to V <sub>DDQ</sub> |                       | 0 to V <sub>DDQ</sub> |                       | V    |
| Input and output timing ref. voltages                    | V <sub>DDQ</sub> /2   |                       | V <sub>DDQ</sub> /2   |                       | V    |

**Figure 9. AC measurement I/O waveform**

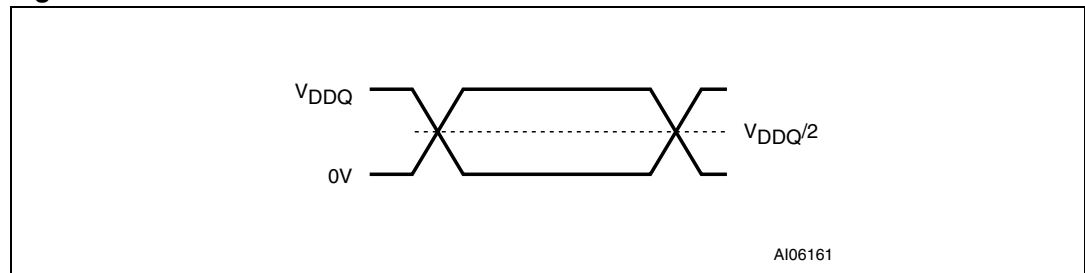


Figure 10. AC measurement load circuit

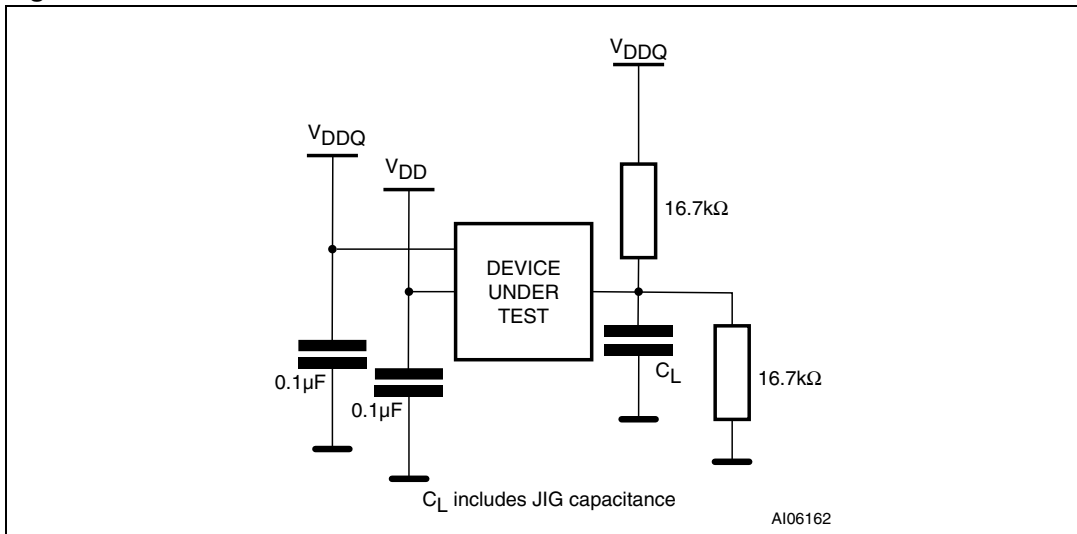


Table 21. Capacitance<sup>(1)</sup>

| Symbol    | Parameter          | Test condition         | Min | Max | Unit |
|-----------|--------------------|------------------------|-----|-----|------|
| $C_{IN}$  | Input capacitance  | $V_{IN} = 0\text{ V}$  | 6   | 8   | pF   |
| $C_{OUT}$ | Output capacitance | $V_{OUT} = 0\text{ V}$ | 8   | 12  | pF   |

1. Sampled only, not 100% tested.

Table 22. DC characteristics - currents

| Symbol              | Parameter  |          | Test condition   | Typ | Max     | Unit          |
|---------------------|--|----------|--|-----|---------|---------------|
| $I_{LI}$            | Input leakage current                                  |          | $0\text{ V} \leq V_{IN} \leq V_{DDQ}$  |     | $\pm 1$ | $\mu\text{A}$ |
| $I_{LO}$            | Output leakage current                                 |          | $0\text{ V} \leq V_{OUT} \leq V_{DDQ}$   |     | $\pm 1$ | $\mu\text{A}$ |
| $I_{DD1}$           | Supply current<br>Asynchronous read (f = 5 MHz)        |          | $\bar{E} = V_{IL}, \bar{G} = V_{IH}$   | 13  | 15      | mA            |
|                     | Supply current<br>Synchronous read (f = 54 MHz)        |          | 4 word   | 18  | 20      | mA            |
|                     |  |          | 8 word   | 20  | 22      | mA            |
|                     |  |          | 16 word  | 25  | 27      | mA            |
|                     |  |          | Continuous   | 28  | 30      | mA            |
|                     | Supply current<br>Synchronous read (f = 66 MHz)        |          | 4 word   | 20  | 22      | mA            |
|                     |  |          | 8 word   | 22  | 24      | mA            |
|                     |  |          | 16 word  | 27  | 29      | mA            |
| Continuous          |  |          | 30   | 32  | mA      |               |
| $I_{DD2}$           | Supply current (reset)                                 | 128 Mbit | $\overline{RP} = V_{SS} \pm 0.2\text{ V}$  | 22  | 70      | $\mu\text{A}$ |
|                     |  | 256 Mbit |  | 70  | 110     |               |
| $I_{DD3}$           | Supply current<br>(standby)                            | 128 Mbit | $\bar{E} = V_{DD} \pm 0.2\text{ V}$<br>$K = V_{SS}$  | 22  | 70      | $\mu\text{A}$ |
|                     |  | 256 Mbit |  | 70  | 110     |               |
| $I_{DD4}$           | Supply current (automatic standby)                     |          | $\bar{E} = V_{IL}, \bar{G} = V_{IH}$   | 22  | 50      | $\mu\text{A}$ |
| $I_{DD5}^{(1)}$     | Supply current (program)                               |          | $V_{PP} = V_{PPH}$   | 10  | 30      | mA            |
|                     |  |          | $V_{PP} = V_{DD}$  | 20  | 35      | mA            |
|                     | Supply current (erase)                                 |          | $V_{PP} = V_{PPH}$   | 10  | 30      | mA            |
|                     |  |          | $V_{PP} = V_{DD}$  | 20  | 35      | mA            |
| $I_{DD6}^{(1),(2)}$ | Supply current<br>(dual operations)                    |          | Program/erase in one bank,<br>asynchronous read in<br>another bank                           | 33  | 50      | mA            |
|                     |  |          | Program/erase in one bank,<br>synchronous read<br>(continuous f = 66 MHz) in<br>another bank | 50  | 67      | mA            |
| $I_{DD7}^{(1)}$     | Supply current<br>program/erase<br>suspended (standby) | 128 Mbit | $\bar{E} = V_{DD} \pm 0.2\text{ V}$<br>$K = V_{SS}$  | 22  | 70      | $\mu\text{A}$ |
|                     |  | 256 Mbit |  | 70  | 110     |               |
| $I_{PP1}^{(1)}$     | $V_{PP}$ supply current (program)                      |          | $V_{PP} = V_{PPH}$   | 2   | 5       | mA            |
|                     |  |          | $V_{PP} = V_{DD}$  | 0.2 | 5       | $\mu\text{A}$ |
|                     | $V_{PP}$ supply current (erase)                        |          | $V_{PP} = V_{PPH}$   | 2   | 5       | mA            |
|                     |  |          | $V_{PP} = V_{DD}$  | 0.2 | 5       | $\mu\text{A}$ |



**Table 22. DC characteristics - currents**

| Symbol          | Parameter                         | Test condition       | Typ | Max | Unit    |
|-----------------|-----------------------------------|----------------------|-----|-----|---------|
| $I_{PP2}$       | $V_{PP}$ supply current (read)    | $V_{PP} \leq V_{DD}$ | 0.2 | 5   | $\mu A$ |
| $I_{PP3}^{(1)}$ | $V_{PP}$ supply current (standby) | $V_{PP} \leq V_{DD}$ | 0.2 | 5   | $\mu A$ |

1. Sampled only, not 100% tested.

2.  $V_{DD}$  dual operation current is the sum of read and program or erase currents.

**Table 23. 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               | $I_{OL} = 100 \mu A$  |                 |     | 0.1             | V    |
| $V_{OH}$   | Output High voltage              | $I_{OH} = -100 \mu A$ | $V_{DDQ} - 0.1$ |     |                 | V    |
| $V_{PP1}$  | $V_{PP}$ Program voltage-logic   | Program, erase        | 1.3             | 1.8 | 3.3             | 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 11. Asynchronous random access read AC waveforms

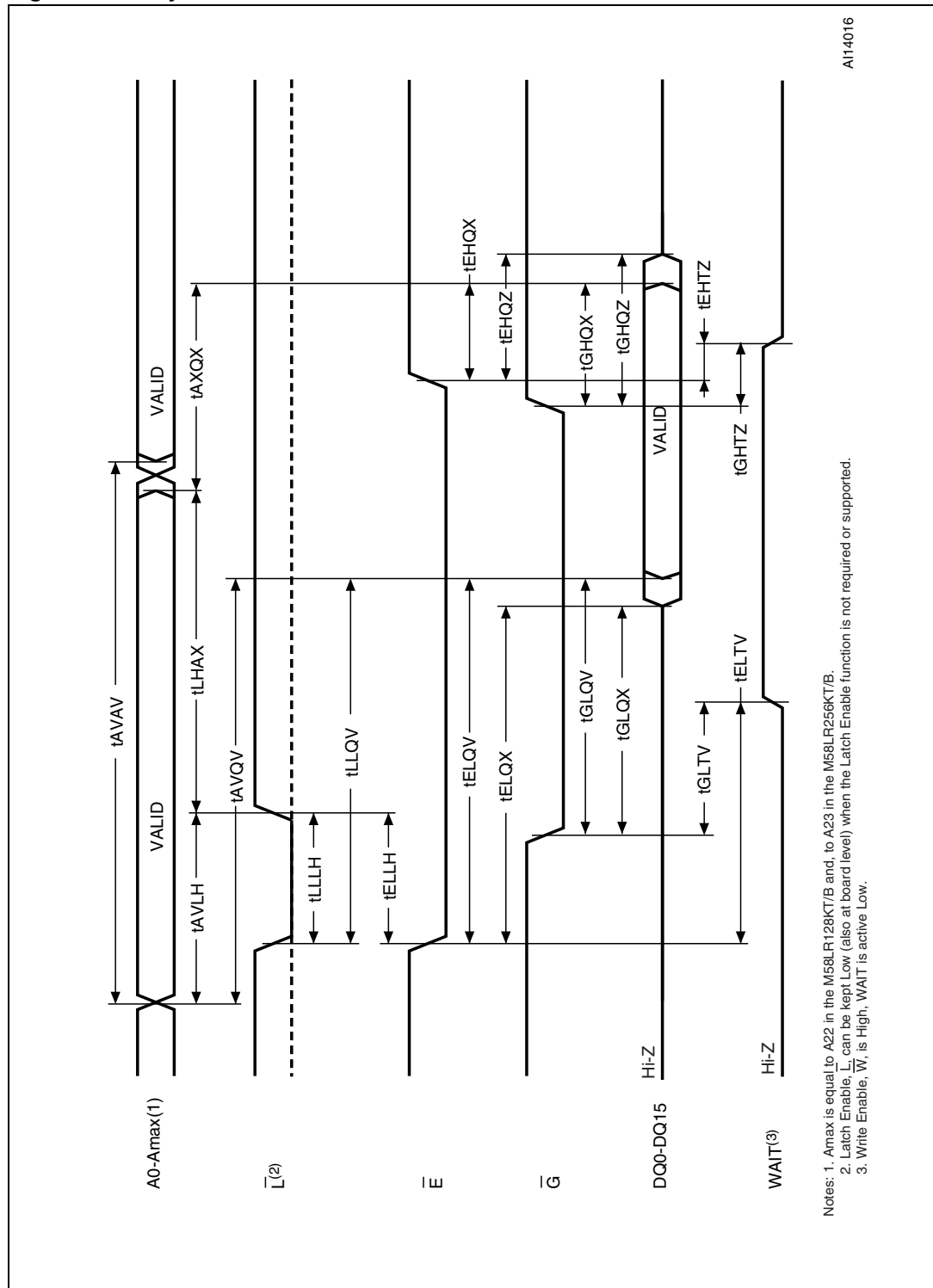


Figure 12. Asynchronous page read AC waveforms

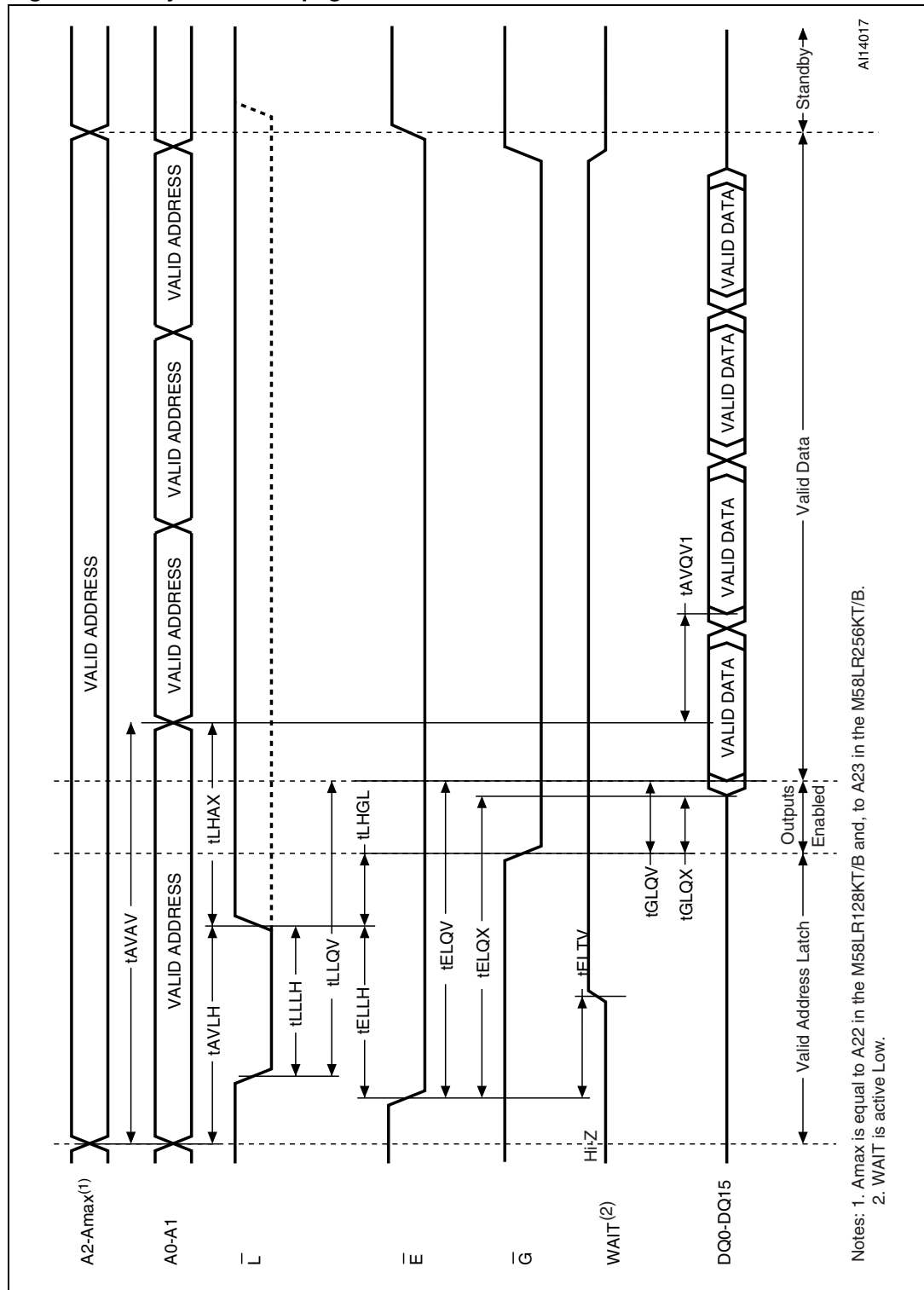


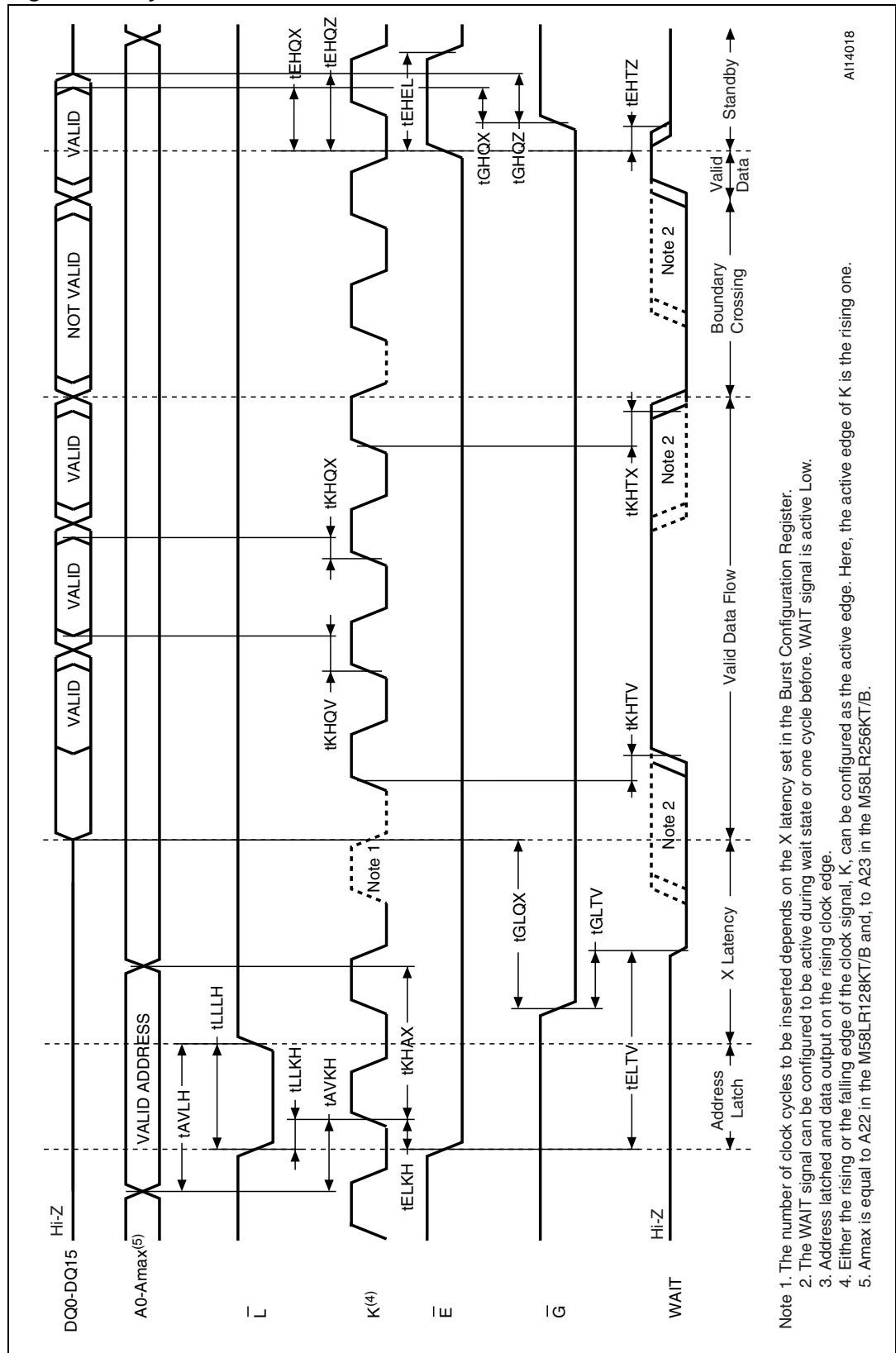
Table 24. Asynchronous read AC characteristics

| Symbol        | Alt              | Parameter                       |   | M58LRxxxKT/B |    | Unit |    |
|---------------|------------------|---------------------------------|---|--------------|----|------|----|
|               |                  |                                 |   | 70           | 85 |      |    |
| Read Timings  | $t_{AVAV}$       | $t_{RC}$                        | Address Valid to Next Address Valid       | Min          | 70 | 85   | ns |
|               | $t_{AVQV}$       | $t_{ACC}$                       | Address Valid to Output Valid (Random)    | Max          | 70 | 85   | ns |
|               | $t_{AVQV1}$      | $t_{PAGE}$                      | Address Valid to Output Valid (page)      | Max          | 20 | 25   | ns |
|               | $t_{AXQX}^{(1)}$ | $t_{OH}$                        | Address Transition to Output Transition   | Min          | 0  | 0    | ns |
|               | $t_{ELTV}$       |                                 | Chip Enable Low to Wait Valid             | Max          | 11 | 14   | ns |
|               | $t_{ELQV}^{(2)}$ | $t_{CE}$                        | Chip Enable Low to Output Valid           | Max          | 70 | 85   | ns |
|               | $t_{ELQX}^{(1)}$ | $t_{LZ}$                        | Chip Enable Low to Output Transition      | Min          | 0  | 0    | ns |
|               | $t_{EHTZ}$       |                                 | Chip Enable High to Wait Hi-Z             | Max          | 11 | 14   | ns |
|               | $t_{EHQX}^{(1)}$ | $t_{OH}$                        | Chip Enable High to Output Transition     | Min          | 2  | 2    | ns |
|               | $t_{EHQZ}^{(1)}$ | $t_{HZ}$                        | Chip Enable High to Output Hi-Z           | Max          | 14 | 14   | ns |
|               | $t_{GLQV}^{(2)}$ | $t_{OE}$                        | Output Enable Low to Output Valid         | Max          | 20 | 20   | ns |
|               | $t_{GLQX}^{(1)}$ | $t_{OLZ}$                       | Output Enable Low to Output Transition    | Min          | 0  | 0    | ns |
|               | $t_{GLTV}$       |                                 | Output Enable Low to Wait Valid           | Max          | 14 | 14   | ns |
|               | $t_{GHQX}^{(1)}$ | $t_{OH}$                        | Output Enable High to Output Transition   | Min          | 2  | 2    | ns |
|               | $t_{GHQZ}^{(1)}$ | $t_{DF}$                        | Output Enable High to Output Hi-Z         | Max          | 14 | 14   | ns |
| $t_{GHTZ}$    |                  | Output Enable High to Wait Hi-Z | Max                                       | 14           | 14 | ns   |    |
| Latch Timings | $t_{AVLH}$       | $t_{AVADVH}$                    | Address Valid to Latch Enable High        | Min          | 7  | 7    | ns |
|               | $t_{ELLH}$       | $t_{ELADVH}$                    | Chip Enable Low to Latch Enable High      | Min          | 10 | 10   | ns |
|               | $t_{LHAX}$       | $t_{ADVHAX}$                    | Latch Enable High to Address Transition   | Min          | 7  | 7    | ns |
|               | $t_{LLLH}$       | $t_{ADVLADVH}$                  | Latch Enable Pulse Width                  | Min          | 7  | 7    | ns |
|               | $t_{LLQV}$       | $t_{ADVLQV}$                    | Latch Enable Low to Output Valid (Random) | Max          | 70 | 85   | ns |

1. Sampled only, not 100% tested.

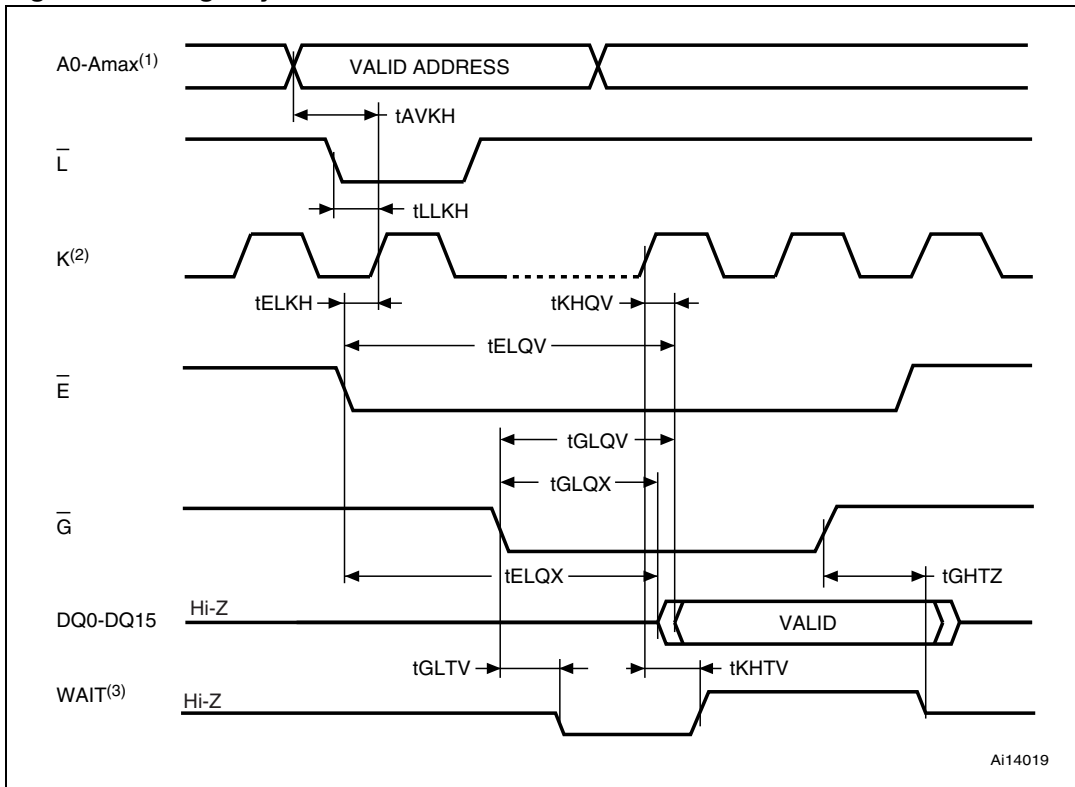
2.  $\bar{G}$  may be delayed by up to  $t_{ELQV} - t_{GLQV}$  after the falling edge of  $\bar{E}$  without increasing  $t_{ELQV}$ .

Figure 13. Synchronous burst read AC waveforms



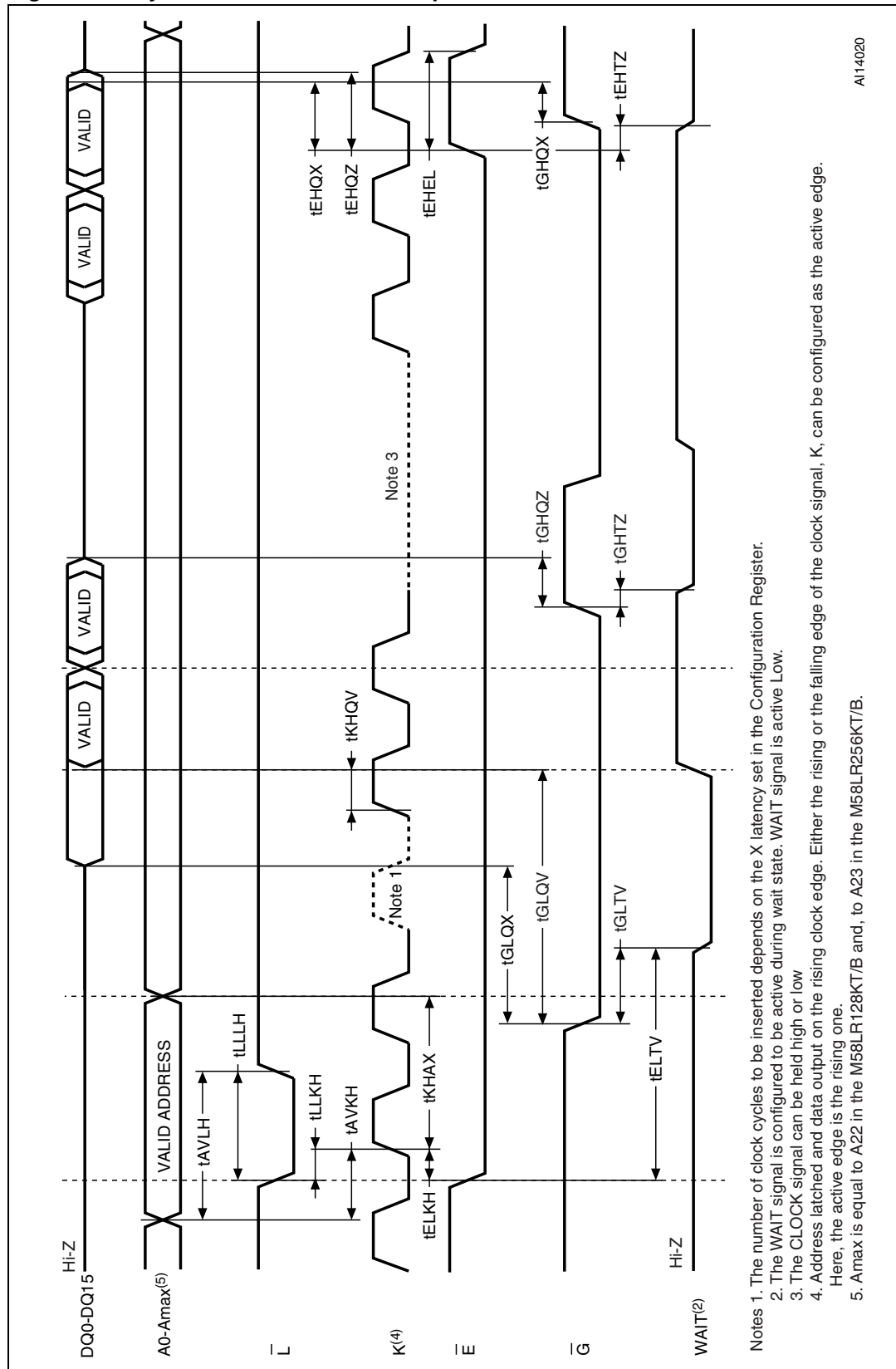
A114018

Figure 14. Single synchronous read AC waveforms



1. Amax is equal to A22 in the M58LR128KT/B and, to A23 in the M58LR256KT/B.
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.
3. The WAIT signal is configured to be active during wait state. WAIT signal is active Low.

Figure 15. Synchronous burst read suspend AC waveforms



AI14020

Figure 16. Clock input AC waveform

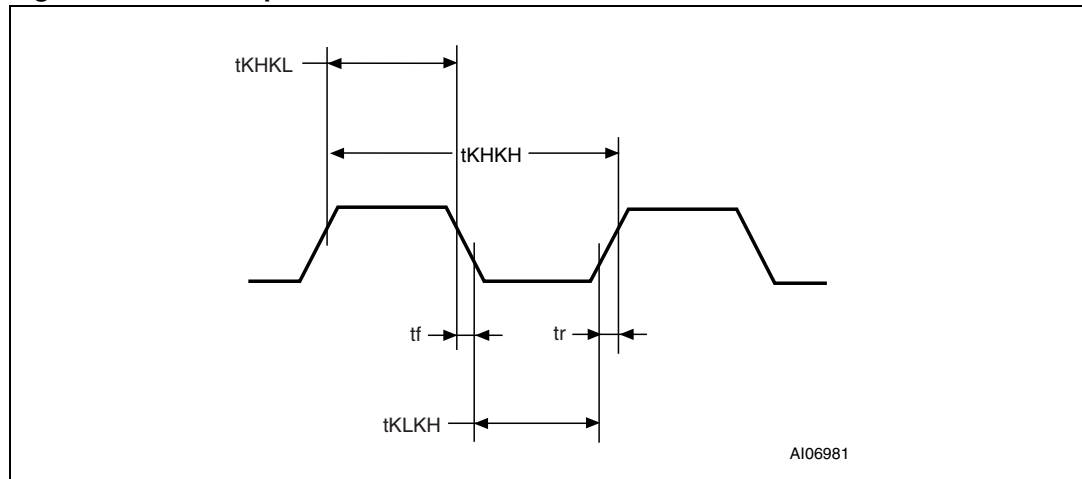


Table 25. Synchronous read AC characteristics<sup>(1) (2)</sup>

| Symbol                    | Alt                      | Parameter      |  | M58LRxxxKT/B              |     | Unit |      |
|---------------------------|--------------------------|----------------|--|---------------------------|-----|------|------|
|                           |                          |                |  | 70                        | 85  |      |      |
| Synchronous Read Timings  | $t_{AVKH}$               | $t_{AVCLKH}$   | Address Valid to Clock High                                      | Min                       | 7   | 7    | ns   |
|                           | $t_{ELKH}$               | $t_{ELCLKH}$   | Chip Enable Low to Clock High                                    | Min                       | 7   | 7    | ns   |
|                           | $t_{ELTV}$               |                | Chip Enable Low to Wait Valid                                    | Max                       | 14  | 14   | ns   |
|                           | $t_{EHEL}$               |                | Chip Enable Pulse Width (subsequent synchronous reads)           | Min                       | 14  | 14   | ns   |
|                           | $t_{EHTZ}$               |                | Chip Enable High to Wait Hi-Z                                    | Max                       | 14  | 14   | ns   |
|                           | $t_{KHAX}$               | $t_{CLKHAX}$   | Clock High to Address Transition                                 | Min                       | 7   | 7    | ns   |
|                           | $t_{KHQV}$<br>$t_{KHTV}$ | $t_{CLKHQV}$   | Clock High to Output Valid<br>Clock High to WAIT Valid           | Max                       | 11  | 14   | ns   |
|                           | $t_{KHQX}$<br>$t_{KHTX}$ | $t_{CLKHQX}$   | Clock High to Output Transition<br>Clock High to WAIT Transition | Min                       | 3   | 3    | ns   |
|                           | $t_{LLKH}$               | $t_{ADVLCLKH}$ | Latch Enable Low to Clock High                                   | Min                       | 7   | 7    | ns   |
|                           | Clock Specifications     | $t_{KHKH}$     | $t_{CLK}$  | Clock Period (f = 54 MHz) | Min |      | 18.5 |
| Clock Period (f = 66 MHz) |                          |                |  | 15                        |     | ns   |      |
| $t_{KHKL}$<br>$t_{KLKH}$  |                          |                | Clock High to Clock Low<br>Clock Low to Clock High               | Min                       | 4.5 | 4.5  | ns   |
| $t_f$<br>$t_r$            |                          |                | Clock Fall or Rise Time  | Max                       | 3   | 3    | ns   |

1. Sampled only, not 100% tested.
2. For other timings please refer to [Table 24: Asynchronous read AC characteristics](#).



Figure 17. Write AC waveforms, Write Enable controlled

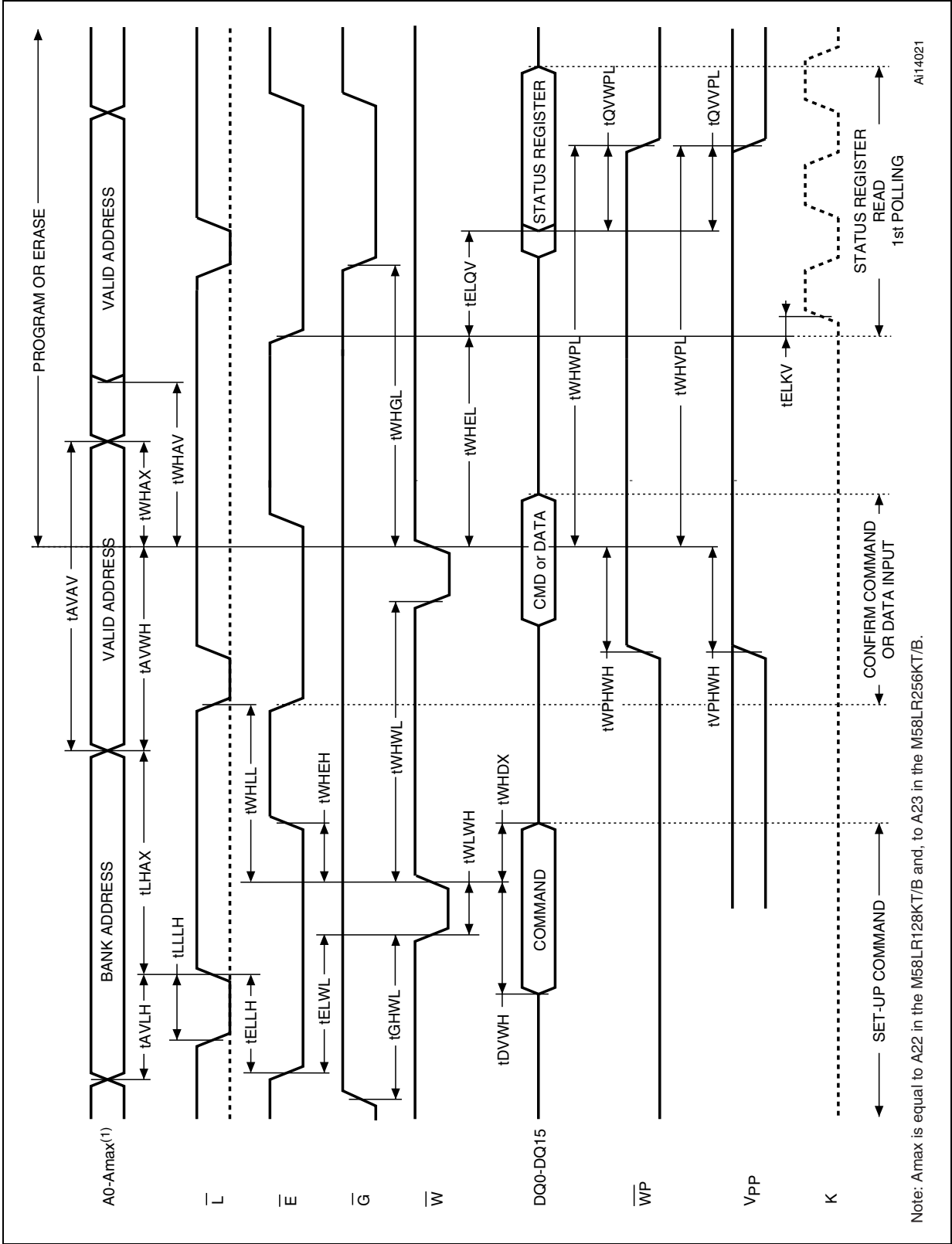


Table 26. Write AC characteristics, Write Enable controlled<sup>(1)</sup>

| Symbol                          | Alt                              | Parameter                             |   | M58LRxxxKT/B |     | Unit |    |
|---------------------------------|----------------------------------|---------------------------------------|---|--------------|-----|------|----|
|                                 |                                  |                                       |   | 70           | 85  |      |    |
| Write Enable Controlled Timings | t <sub>AVAV</sub>                | t <sub>WC</sub>                       | Address Valid to Next Address Valid                   | Min          | 70  | 85   | ns |
|                                 | t <sub>AVLH</sub>                |                                       | Address Valid to Latch Enable High                    | Min          | 7   | 7    | ns |
|                                 | t <sub>AVWH</sub> <sup>(2)</sup> |                                       | Address Valid to Write Enable High                    | Min          | 45  | 45   | ns |
|                                 | t <sub>DVWH</sub>                | t <sub>DS</sub>                       | Data Valid to Write Enable High                       | Min          | 45  | 45   | ns |
|                                 | t <sub>ELLH</sub>                |                                       | Chip Enable Low to Latch Enable High                  | Min          | 10  | 10   | ns |
|                                 | t <sub>ELWL</sub>                | t <sub>CS</sub>                       | Chip Enable Low to Write Enable Low                   | Min          | 0   | 0    | ns |
|                                 | t <sub>ELQV</sub>                |                                       | Chip Enable Low to Output Valid                       | Min          | 70  | 85   | ns |
|                                 | t <sub>ELKV</sub>                |                                       | Chip Enable Low to Clock Valid                        | Min          | 7   | 7    | ns |
|                                 | t <sub>GHWL</sub>                |                                       | Output Enable High to Write Enable Low                | Min          | 17  | 17   | ns |
|                                 | t <sub>LHAX</sub>                |                                       | Latch Enable High to Address Transition               | Min          | 7   | 7    | ns |
|                                 | t <sub>LLH</sub>                 |                                       | Latch Enable Pulse Width                              | Min          | 7   | 7    | ns |
|                                 | t <sub>WHAV</sub> <sup>(2)</sup> |                                       | Write Enable High to Address Valid                    | Min          | 0   | 0    | ns |
|                                 | t <sub>WHAX</sub> <sup>(2)</sup> | t <sub>AH</sub>                       | Write Enable High to Address Transition               | Min          | 0   | 0    | ns |
|                                 | t <sub>WHDX</sub>                | t <sub>DH</sub>                       | Write Enable High to Input Transition                 | Min          | 0   | 0    | ns |
|                                 | t <sub>WHEH</sub>                | t <sub>CH</sub>                       | Write Enable High to Chip Enable High                 | Min          | 0   | 0    | ns |
|                                 | t <sub>WHEL</sub> <sup>(3)</sup> |                                       | Write Enable High to Chip Enable Low                  | Min          | 25  | 25   | ns |
|                                 | t <sub>WHGL</sub>                |                                       | Write Enable High to Output Enable Low                | Min          | 0   | 0    | ns |
|                                 | t <sub>WHLL</sub> <sup>(3)</sup> |                                       | Write Enable High to Latch Enable Low                 | Min          | 25  | 25   | ns |
|                                 | t <sub>WHWL</sub>                | t <sub>WPH</sub>                      | Write Enable High to Write Enable Low                 | Min          | 25  | 25   | ns |
| t <sub>WLWH</sub>               | t <sub>WP</sub>                  | Write Enable Low to Write Enable High | Min   | 45           | 45  | ns   |    |
| Protection Timings              | t <sub>QVPL</sub>                |                                       | Output (Status Register) Valid to V <sub>PP</sub> Low | Min          | 0   | 0    | ns |
|                                 | t <sub>QVWPL</sub>               |                                       | Output (Status Register) Valid to Write Protect Low   | Min          | 0   | 0    | ns |
|                                 | t <sub>VPHWH</sub>               | t <sub>VPS</sub>                      | V <sub>PP</sub> High to Write Enable High             | Min          | 200 | 200  | ns |
|                                 | t <sub>WHVPL</sub>               |                                       | Write Enable High to V <sub>PP</sub> Low              | Min          | 200 | 200  | ns |
|                                 | t <sub>WHWPL</sub>               |                                       | Write Enable High to Write Protect Low                | Min          | 200 | 200  | ns |
|                                 | t <sub>WPHWH</sub>               |                                       | Write Protect High to Write Enable High               | Min          | 200 | 200  | ns |

1. Sampled only, not 100% tested.
2. Meaningful only if  $\bar{L}$  is always kept low.
3. t<sub>WHEL</sub> and t<sub>WHLL</sub> 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<sub>WHEL</sub> and t<sub>WHLL</sub> is 0 ns.

Figure 18. Write AC waveforms, Chip Enable controlled

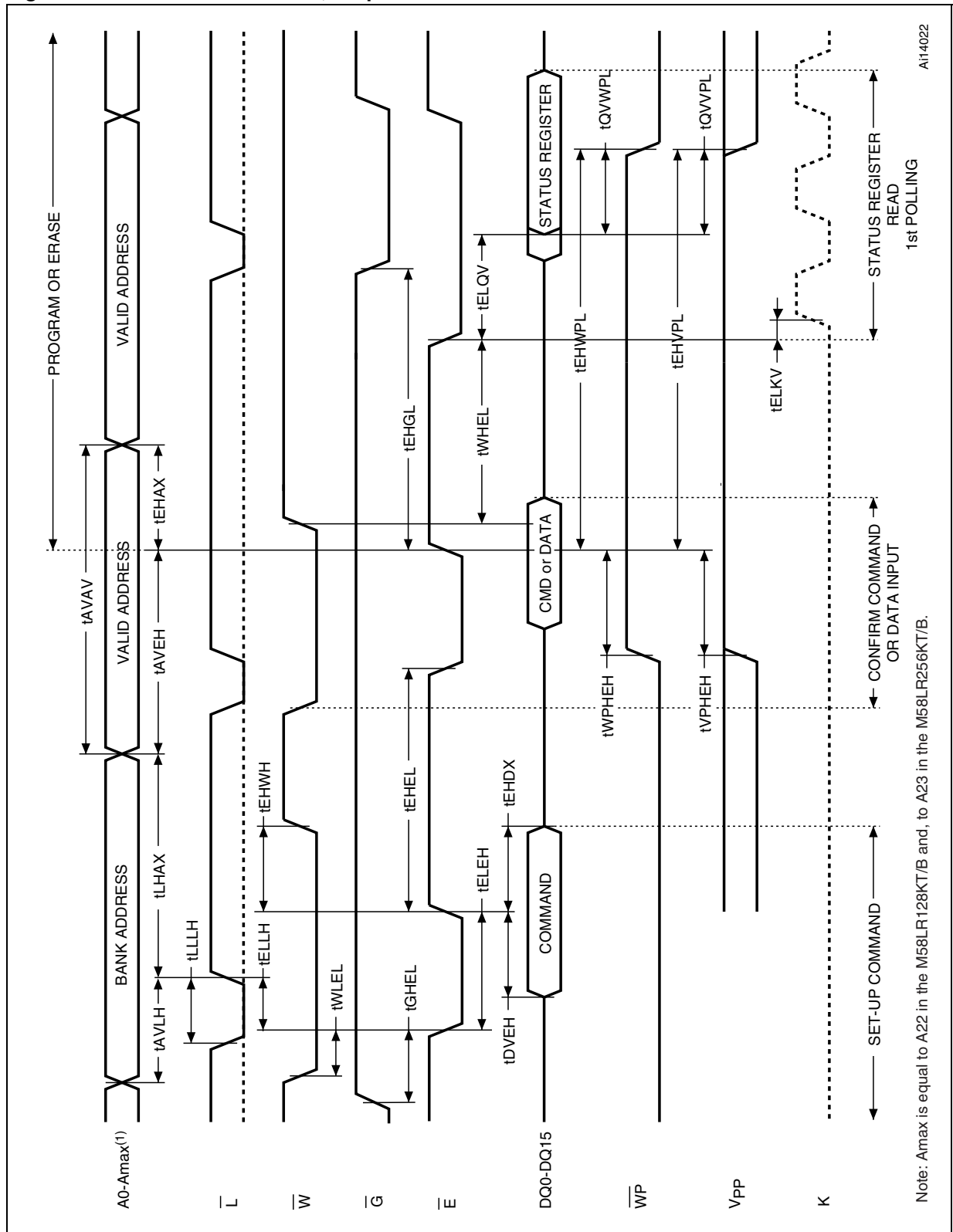


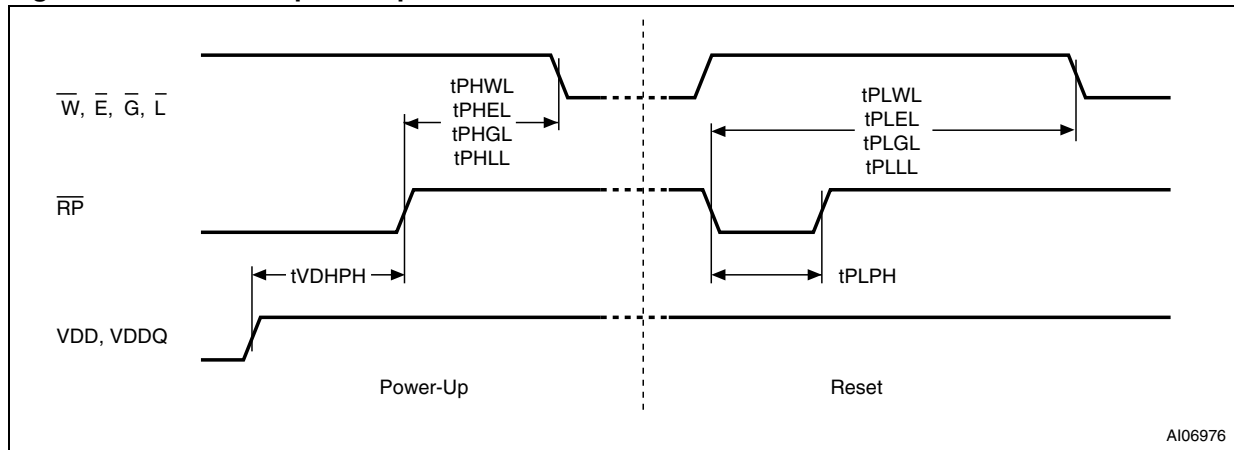
Table 27. Write AC characteristics, Chip Enable controlled<sup>(1)</sup>

| Symbol                         | Alt                | Parameter                                |   | M58LRxxxKT/B                         |     | Unit |    |
|--------------------------------|--------------------|--|---|--------------------------------------|-----|------|----|
|                                |                    |  |   | 70                                   | 85  |      |    |
| Chip Enable Controlled Timings | t <sub>AVAV</sub>  | t <sub>WC</sub>                          | Address Valid to Next Address Valid                   | Min                                  | 70  | 85   | ns |
|                                | t <sub>AVEH</sub>  |  | Address Valid to Chip Enable High                     | Min                                  | 45  | 45   | ns |
|                                | t <sub>AVLH</sub>  |  | Address Valid to Latch Enable High                    | Min                                  | 7   | 7    | ns |
|                                | t <sub>DVEH</sub>  | t <sub>DS</sub>                          | Data Valid to Chip Enable High                        | Min                                  | 45  | 45   | ns |
|                                | t <sub>EHAX</sub>  | t <sub>AH</sub>                          | Chip Enable High to Address Transition                | Min                                  | 0   | 0    | ns |
|                                | t <sub>EHDX</sub>  | t <sub>DH</sub>                          | Chip Enable High to Input Transition                  | Min                                  | 0   | 0    | ns |
|                                | t <sub>EHXL</sub>  | t <sub>CPH</sub>                         | Chip Enable High to Chip Enable Low                   | Min                                  | 25  | 25   | ns |
|                                | t <sub>EHGL</sub>  |  | Chip Enable High to Output Enable Low                 | Min                                  | 0   | 0    | ns |
|                                | t <sub>EHWH</sub>  | t <sub>CH</sub>                          | Chip Enable High to Write Enable High                 | Min                                  | 0   | 0    | ns |
|                                | t <sub>ELKV</sub>  |  | Chip Enable Low to Clock Valid                        | Min                                  | 7   | 7    | ns |
|                                | t <sub>ELEH</sub>  | t <sub>CP</sub>                          | Chip Enable Low to Chip Enable High                   | Min                                  | 45  | 45   | ns |
|                                | t <sub>ELLH</sub>  |  | Chip Enable Low to Latch Enable High                  | Min                                  | 10  | 10   | ns |
|                                | t <sub>ELQV</sub>  |  | Chip Enable Low to Output Valid                       | Min                                  | 70  | 85   | ns |
|                                | t <sub>GHEL</sub>  |  | Output Enable High to Chip Enable Low                 | Min                                  | 17  | 17   | ns |
|                                | t <sub>LHAX</sub>  |  | Latch Enable High to Address Transition               | Min                                  | 7   | 7    | ns |
|                                | t <sub>LLLH</sub>  |  | Latch Enable Pulse Width                              | Min                                  | 7   | 7    | ns |
|                                | Protection Timings | t <sub>WHEL</sub> <sup>(2)</sup>         |   | Write Enable High to Chip Enable Low | Min | 25   | 25 |
| t <sub>WLEL</sub>              |                    | t <sub>CS</sub>                          | Write Enable Low to Chip Enable Low                   | Min                                  | 0   | 0    | ns |
| t <sub>EHVPL</sub>             |                    |  | Chip Enable High to V <sub>PP</sub> Low               | Min                                  | 200 | 200  | ns |
| t <sub>EHWPL</sub>             |                    |  | Chip Enable High to Write Protect Low                 | Min                                  | 200 | 200  | ns |
| t <sub>QVVPL</sub>             |                    |  | Output (Status Register) Valid to V <sub>PP</sub> Low | Min                                  | 0   | 0    | ns |
| t <sub>QVWPL</sub>             |                    |  | Output (Status Register) Valid to Write Protect Low   | Min                                  | 0   | 0    | ns |
| t <sub>VPHEH</sub>             | t <sub>VPS</sub>   | V <sub>PP</sub> High to Chip Enable High | Min   | 200                                  | 200 | ns   |    |
| t <sub>WPHEH</sub>             |                    | Write Protect High to Chip Enable High   | Min   | 200                                  | 200 | ns   |    |

1. Sampled only, not 100% tested.

2. t<sub>WHEL</sub> 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<sub>WHEL</sub> is 0ns.

Figure 19. Reset and power-up AC waveforms



AI06976

Table 28. Reset and power-up AC characteristics

| Symbol               | Parameter                          | Test condition   |     | M58LRxxxKT/B | Unit    |
|----------------------|------------------------------------|------------------|-----|--------------|---------|
| $t_{PLWL}$           | Reset Low to Write Enable Low      | During program   | Min | 25           | $\mu s$ |
| $t_{PLEL}$           | Reset Low to Chip Enable Low,      | During erase     | Min | 25           | $\mu s$ |
| $t_{PLGL}$           | Reset Low to Output Enable Low,    | Other conditions | Min | 80           | ns      |
| $t_{PLLL}$           | Reset Low to Latch Enable Low      |                  |     |              |         |
| $t_{PHWL}$           | Reset High to Write Enable Low     |                  | Min | 30           | ns      |
| $t_{PHEL}$           | Reset High to Chip Enable Low      |                  |     |              |         |
| $t_{PHGL}$           | Reset High to Output Enable Low    |                  |     |              |         |
| $t_{PHLL}$           | Reset High to Latch Enable Low     |                  |     |              |         |
| $t_{PLPH}^{(1),(2)}$ | $\overline{RP}$ pulse width        |                  | Min | 50           | ns      |
| $t_{VDHPH}^{(3)}$    | Supply voltages High to Reset High |                  | Min | 300          | $\mu s$ |

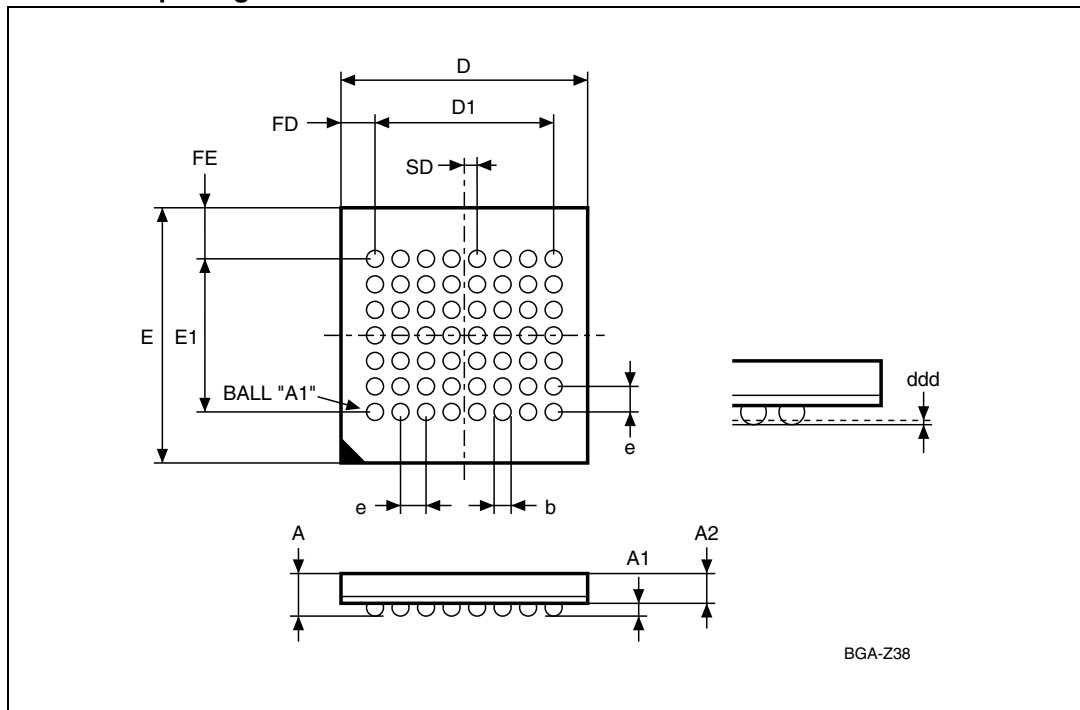
1. The device Reset is possible but not guaranteed if  $t_{PLPH} < 50$  ns.
2. Sampled only, not 100% tested.
3. It is important to assert  $\overline{RP}$  to allow proper CPU initialization during power-up or Reset.

### 13 Package mechanical

To meet environmental requirements, Numonyx offers the M58LR128KT/B in ECOPACK® packages. These packages 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. ECOPACK is an Numonyx trademark.

**Figure 20. VFBGA56 7.7 x 9 mm - 8 x 7 active ball array, 0.75 mm pitch, bottom view package outline**



1. Drawing is not to scale.

**Table 29. VFPGA56 7.7 x 9 mm - 8 x 7 ball array, 0.75 mm pitch, package mechanical data**

| Symbol | Millimeters |       |       | Inches |        |        |
|--------|-------------|-------|-------|--------|--------|--------|
|        | Typ         | Min   | Max   | Typ    | Min    | Max    |
| A      |             |       | 1.000 |        |        | 0.0394 |
| A1     |             | 0.200 |       |        | 0.0079 |        |
| A2     | 0.660       |       |       | 0.0260 |        |        |
| b      | 0.350       | 0.300 | 0.400 | 0.0138 | 0.0118 | 0.0157 |
| D      | 7.700       | 7.600 | 7.800 | 0.3031 | 0.2992 | 0.3071 |
| D1     | 5.250       | –     | –     | 0.2067 | –      | –      |
| ddd    |             |       | 0.080 |        |        | 0.0031 |
| e      | 0.750       | –     | –     | 0.0295 | –      | –      |
| E      | 9.000       | 8.900 | 9.100 | 0.3543 | 0.3504 | 0.3583 |
| E1     | 4.500       | –     | –     | 0.1772 | –      | –      |
| FD     | 1.225       | –     | –     | 0.0482 | –      | –      |
| FE     | 2.250       | –     | –     | 0.0886 | –      | –      |
| SD     | 0.375       | –     | –     | 0.0148 | –      | –      |

**Figure 21. TFBGA88 8 x 10 mm - 8 x 10 ball array, 0.8 mm pitch, bottom view package outline**

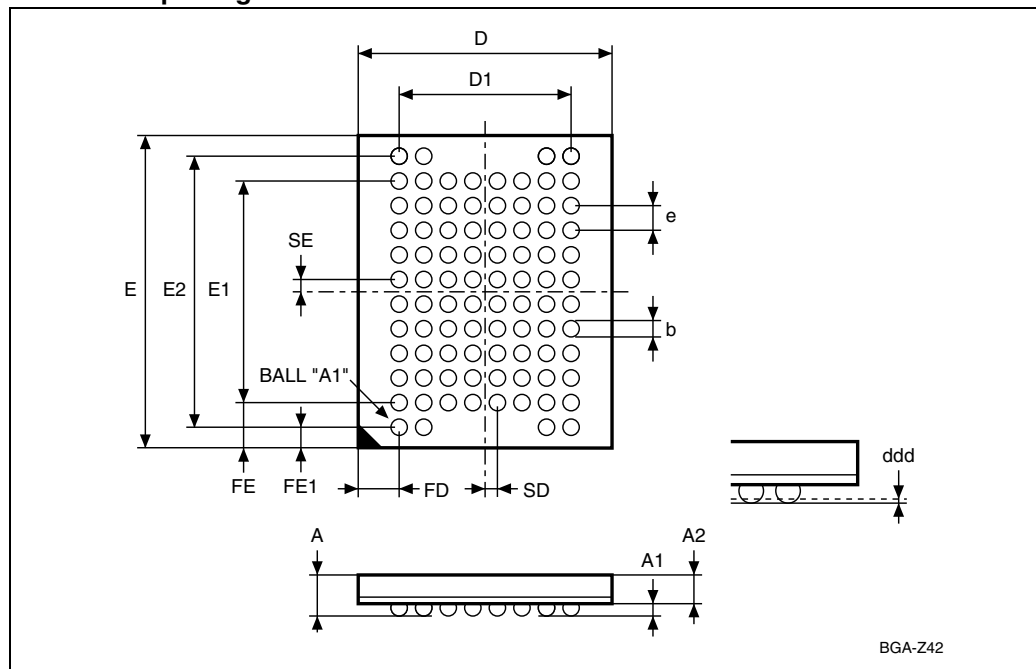


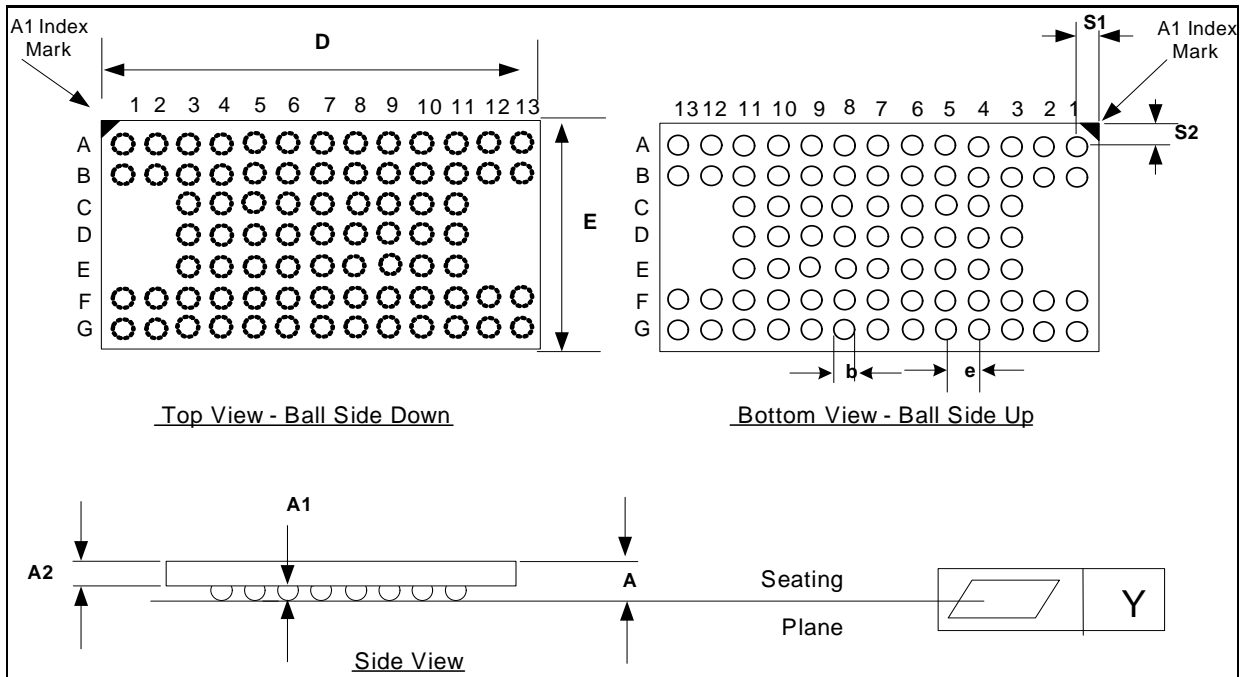
Figure 22. Drawing is not to scale.

**Table 30. TFBGA88 8 x 10 mm - 8 x 10 ball array, 0.8 mm pitch, package mechanical data**

| Symbol | Millimeters |       |        | Inches |        |        |
|--------|-------------|-------|--------|--------|--------|--------|
|        | Typ         | Min   | Max    | Typ    | Min    | Max    |
| A      |             |       | 1.200  |        |        | 0.0472 |
| A1     |             | 0.200 |        |        | 0.0079 |        |
| A2     | 0.850       |       |        | 0.0335 |        |        |
| b      | 0.350       | 0.300 | 0.400  | 0.0138 | 0.0118 | 0.0157 |
| D      | 8.000       | 7.900 | 8.100  | 0.3150 | 0.3110 | 0.3189 |
| D1     | 5.600       |       |        | 0.2205 |        |        |
| ddd    |             |       | 0.100  |        |        | 0.0039 |
| E      | 10.000      | 9.900 | 10.100 | 0.3937 | 0.3898 | 0.3976 |
| E1     | 7.200       |       |        | 0.2835 |        |        |
| E2     | 8.800       |       |        | 0.3465 |        |        |
| e      | 0.800       | -     | -      | 0.0315 | -      | -      |
| FD     | 1.200       |       |        | 0.0472 |        |        |
| FE     | 1.400       |       |        | 0.0551 |        |        |
| FE1    | 0.600       |       |        | 0.0236 |        |        |
| SD     | 0.400       |       |        | 0.0157 |        |        |
| SE     | 0.400       |       |        | 0.0157 |        |        |



Figure 23. VFBGA79 9x11mm - 0.75 pitch -package outline (1)



1. Drawing not to scale

Table 31. VFBGA79 9x11mm - 0.75 mm pitch, package mechanical data

| Symbol | Millimeters |        |        | Inches |        |        |
|--------|-------------|--------|--------|--------|--------|--------|
|        | Min         | Typ    | Max    | Min    | Typ    | Max    |
| A      |             |        | 1.000  |        |        | 0.0394 |
| A1     | 0.150       |        |        | 0.0059 |        |        |
| A2     |             | 0.665  |        |        | 0.0262 |        |
| b      | 0.325       | 0.375  | 0.425  | 0.0128 | 0.0148 | 0.0167 |
| D      | 10.900      | 11.000 | 11.100 | 0.4291 | 0.4331 | 0.4370 |
| E      | 8.900       | 9.000  | 9.100  | 0.3504 | 0.3543 | 0.3583 |
| e      |             | 0.750  |        |        | 0.0295 |        |
| N      |             | 79     |        |        | 79     |        |
| Y      |             |        | 0.100  |        |        | 0.0039 |
| S1     | 0.900       | 1.000  | 1.100  | 0.0354 | 0.0394 | 0.0433 |
| S2     | 2.150       | 2.250  | 2.350  | 0.0846 | 0.0886 | 0.0925 |

# 14 Part numbering

**Table 32. Ordering information scheme**

|  |            |    |    |   |   |
|--|------------|----|----|---|---|
| Example:   | M58LR128KT | 70 | ZB | 5 | E |
| <b>Device type</b><br>M58  |            |    |    |   |   |
| <b>Architecture</b><br>L = multilevel, multiple bank, burst mode   |            |    |    |   |   |
| <b>Operating voltage</b><br>R = $V_{DD} = 1.7\text{ V to }2.0\text{ V}$ , $V_{DDQ} = 1.7\text{ V to }2.0\text{ V}$   |            |    |    |   |   |
| <b>Density</b><br>128 = 128 Mbit (x16)<br>256 = 256 Mbit (x16)   |            |    |    |   |   |
| <b>Technology</b><br>K = 65 nm technology  |            |    |    |   |   |
| <b>Parameter location</b><br>T = top boot<br>B = bottom boot   |            |    |    |   |   |
| <b>Speed</b><br>70 = 70 ns<br>85 = 85 ns   |            |    |    |   |   |
| <b>Package</b><br>ZB = VFPGA56, 7.7 x 9 mm, 0.75 mm pitch <sup>(1)</sup><br>ZC = VFPGA79, 9 x 11 mm, 0.75 mm pitch<br>ZQ = TFBGA88, 8 x 10mm, 0.8 mm pitch |            |    |    |   |   |
| <b>Temperature range</b><br>5 = -30 to 85 °C   |            |    |    |   |   |
| <b>Packing option</b><br>E = ECOPACK® package, standard packing<br>F = ECOPACK® package, tape and reel packing   |            |    |    |   |   |

1. The M58LR128KT/B are available in the ZB package, while the M58LR256KT/B are available in the ZC and ZQ package.

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

The following set of equations can be used to calculate a complete set of block addresses for the M58LRxxxKT/B using the information contained in Tables 36 to 44.

### To calculate the block base address from the block number:

First it is necessary to calculate the bank number and the block number offset. This can be achieved using the following formulas:

$$\text{Bank\_Number} = (\text{Block\_Number} - 3) / 8$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - 3 - (\text{Bank\_Number} \times 8),$$

If  $\text{Bank\_Number} = 0$ , the block base address can be directly read from Tables 36 and 42 (parameter bank block addresses) in the address range column, in the row that corresponds to the given block number.

Otherwise:

$$\text{Block\_Base\_Address} = \text{Bank\_Base\_Address} + \text{Block\_Base\_Address\_Offset}$$

### To calculate the bank number and the block number from the block base address:

If the address is in the range of the parameter bank, the Bank Number is 0 and the Block Number can be directly read from Tables 36 and 42 (parameter bank Block Addresses), in the Block Number column, in the row that corresponds to the address given. Otherwise, the Block Number can be calculated using the formulas below:

For the top configuration (M58LR256KT and M58LR128KT):

$$\text{Block\_Number} = ((\text{NOT address}) / 2^{16}) + 3$$

For the bottom configuration (M58LR256KB and M58LR128KB):

$$\text{Block\_Number} = (\text{address} / 2^{16}) + 3$$

For both configurations the Bank Number and the Block Number Offset can be calculated using the following formulas:

$$\text{Bank\_Number} = (\text{Block\_Number} - 3) / 8$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - 3 - (\text{Bank\_Number} \times 8)$$

**Table 33. M58LR128KT - Parameter bank block addresses**

| Block number | Size (Kwords) | Address range  |
|--------------|---------------|----------------|
| 0            | 16            | 7FC000-7FFFFFF |
| 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  |

**Table 34. M58LR128KT - main bank base addresses**

| Bank number <sup>(1)</sup> | Block numbers | Bank base address |
|----------------------------|---------------|-------------------|
| 1                          | 11-18         | 70 0000           |
| 2                          | 19-26         | 68 0000           |
| 3                          | 27-34         | 60 0000           |
| 4                          | 35-42         | 58 0000           |
| 5                          | 43-50         | 50 0000           |
| 6                          | 51-58         | 48 0000           |
| 7                          | 59-66         | 40 0000           |
| 8                          | 67-74         | 38 0000           |
| 9                          | 75-82         | 30 0000           |
| 10                         | 83-90         | 28 0000           |
| 11                         | 91-98         | 20 0000           |
| 12                         | 99-106        | 18 0000           |
| 13                         | 107-114       | 10 0000           |
| 14                         | 115-122       | 08 0000           |
| 15                         | 123-130       | 00 0000           |

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 35. M58LR128KT - block addresses in main banks**

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 0                   | 07 0000                   |
| 1                   | 06 0000                   |
| 2                   | 05 0000                   |
| 3                   | 04 0000                   |
| 4                   | 03 0000                   |
| 5                   | 02 0000                   |
| 6                   | 01 0000                   |
| 7                   | 00 0000                   |

**Table 36. M58LR256KT - parameter bank block addresses**

| Block number | Size (Kwords) | Address range |
|--------------|---------------|---------------|
| 0            | 16            | FFC000-FFFFFF |
| 1            | 16            | FF8000-FFBFFF |
| 2            | 16            | FF4000-FF7FFF |
| 3            | 16            | FF0000-FF3FFF |
| 4            | 64            | FE0000-FEFFFF |
| 5            | 64            | FD0000-FDFFFF |
| 6            | 64            | FC0000-FCFFFF |
| 7            | 64            | FB0000-FBFFFF |
| 8            | 64            | FA0000-FAFFFF |
| 9            | 64            | F90000-F9FFFF |
| 10           | 64            | F80000-F8FFFF |
| 11           | 64            | F70000-F7FFFF |
| 12           | 64            | F60000-F6FFFF |
| 13           | 64            | F50000-F5FFFF |
| 14           | 64            | F40000-F4FFFF |
| 15           | 64            | F30000-F3FFFF |
| 16           | 64            | F20000-F2FFFF |
| 17           | 64            | F10000-F1FFFF |
| 18           | 64            | F00000-F0FFFF |

**Table 37. M58LR256KT - main bank base addresses**

| Bank number <sup>(1)</sup> | Block numbers | Bank base address |
|----------------------------|---------------|-------------------|
| 1                          | 19-34         | E00000            |
| 2                          | 35-50         | D00000            |
| 3                          | 51-66         | C00000            |
| 4                          | 67-82         | B00000            |
| 5                          | 83-98         | A00000            |
| 6                          | 99-114        | 900000            |
| 7                          | 115-130       | 800000            |
| 8                          | 131-146       | 700000            |
| 9                          | 147-162       | 600000            |
| 10                         | 163-178       | 500000            |
| 11                         | 179-194       | 400000            |
| 12                         | 195-210       | 300000            |
| 13                         | 211-226       | 200000            |
| 14                         | 227-242       | 100000            |
| 15                         | 243-258       | 000000            |

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 38. M58LR256KT - block addresses in main banks**

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 0                   | 0F0000                    |
| 1                   | 0E0000                    |
| 2                   | 0D0000                    |
| 3                   | 0C0000                    |
| 4                   | 0B0000                    |
| 5                   | 0A0000                    |
| 6                   | 090000                    |
| 7                   | 080000                    |
| 8                   | 070000                    |
| 9                   | 060000                    |
| 10                  | 050000                    |
| 11                  | 040000                    |
| 12                  | 030000                    |
| 13                  | 020000                    |
| 14                  | 010000                    |
| 15                  | 000000                    |

**Table 39. M58LR128KB - parameter bank block addresses**

| Block number | Size (Kwords) | Address range |
|--------------|---------------|---------------|
| 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 |

**Table 40. M58LR128KB - main bank base addresses**

| Bank number <sup>(1)</sup> | Block numbers | Bank base address |
|----------------------------|---------------|-------------------|
| 15                         | 123-130       | 78 0000           |
| 14                         | 115-122       | 70 0000           |
| 13                         | 107-114       | 68 0000           |
| 12                         | 99-106        | 60 0000           |
| 11                         | 91-98         | 58 0000           |
| 10                         | 83-90         | 50 0000           |
| 9                          | 75-82         | 48 0000           |
| 8                          | 67-74         | 40 0000           |
| 7                          | 59-66         | 38 0000           |
| 6                          | 51-58         | 30 0000           |
| 5                          | 43-50         | 28 0000           |
| 4                          | 35-42         | 20 0000           |
| 3                          | 27-34         | 18 0000           |
| 2                          | 19-26         | 10 0000           |
| 1                          | 11-18         | 08 0000           |

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).



Table 41. M58LR128KB - block addresses in main banks

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 7                   | 070000                    |
| 6                   | 060000                    |
| 5                   | 050000                    |
| 4                   | 040000                    |
| 3                   | 030000                    |
| 2                   | 020000                    |
| 1                   | 010000                    |
| 0                   | 000000                    |

Table 42. M58LR256KB - parameter bank block addresses

| Block number | Size (Kwords) | Address range   |
|--------------|---------------|-----------------|
| 18           | 64            | 0F0000-0FFFFFFF |
| 17           | 64            | 0E0000-0EFFFFFF |
| 16           | 64            | 0D0000-0DFFFFFF |
| 15           | 64            | 0C0000-0CFFFFFF |
| 14           | 64            | 0B0000-0BFFFFFF |
| 13           | 64            | 0A0000-0AFFFFFF |
| 12           | 64            | 090000-09FFFFFF |
| 11           | 64            | 080000-08FFFFFF |
| 10           | 64            | 070000-07FFFFFF |
| 9            | 64            | 060000-06FFFFFF |
| 8            | 64            | 050000-05FFFFFF |
| 7            | 64            | 040000-04FFFFFF |
| 6            | 64            | 030000-03FFFFFF |
| 5            | 64            | 020000-02FFFFFF |
| 4            | 64            | 010000-01FFFFFF |
| 3            | 16            | 00C000-00FFFFFF |
| 2            | 16            | 008000-00BFFF   |
| 1            | 16            | 004000-007FFF   |
| 0            | 16            | 000000-003FFF   |

**Table 43. M58LR256KB - main bank base addresses**

| Bank number | Block numbers | Bank base address |
|-------------|---------------|-------------------|
| 15          | 243-258       | F00000            |
| 14          | 227-242       | E00000            |
| 13          | 211-226       | D00000            |
| 12          | 195-210       | C00000            |
| 11          | 179-194       | B00000            |
| 10          | 163-178       | A00000            |
| 9           | 147-162       | 900000            |
| 8           | 131-146       | 800000            |
| 7           | 115-130       | 700000            |
| 6           | 99-114        | 600000            |
| 5           | 83-98         | 500000            |
| 4           | 67-82         | 400000            |
| 3           | 51-66         | 300000            |
| 2           | 35-50         | 200000            |
| 1           | 19-34         | 100000            |

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).

**Table 44. M58LR256KB - block addresses in main banks**

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 15                  | 0F0000                    |
| 14                  | 0E0000                    |
| 13                  | 0D0000                    |
| 12                  | 0C0000                    |
| 11                  | 0B0000                    |
| 10                  | 0A0000                    |
| 9                   | 090000                    |
| 8                   | 080000                    |
| 7                   | 070000                    |
| 6                   | 060000                    |
| 5                   | 050000                    |
| 4                   | 040000                    |
| 3                   | 030000                    |
| 2                   | 020000                    |
| 1                   | 010000                    |
| 0                   | 000000                    |

## 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 [45](#), [46](#), [47](#), [48](#), [49](#), [50](#), [51](#), [52](#), [53](#) and [54](#) show the addresses used to retrieve the data. The query data is always presented on the lowest order data outputs (DQ0-DQ7), 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 6: 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 45. 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<br>Unique device number and<br>User programmable OTP |

1. The Flash memory display the CFI data structure when CFI Query command is issued. In this table are listed the main sub-sections detailed in Tables [46](#), [47](#), [48](#) and [49](#). Query data is always presented on the lowest order data outputs.

**Table 46. CFI query identification string**

| Offset    | Sub-section Name   | Description  |            | Value    |
|-----------|--------------------|--|------------|----------|
| 000h      | 0020h              | Manufacturer code  |            | Numonyx  |
| 001h      | 88C4h              | Device code  | M58LR128KT | Top      |
|           | 88C5h              |  | M58LR128KB | Bottom   |
|           | 880Dh              |  | M58LR256KT | Top      |
|           | 880Eh              |  | M58LR256KB | Bottom   |
| 002h-00Fh | Reserved           | Reserved   |            |          |
| 010h      | 0051h              | Query Unique ASCII String "QRY"  |            | "Q"      |
| 011h      | 0052h              |  |            | "R"      |
| 012h      | 0059h              |  |            | "Y"      |
| 013h      | 0001h              | Primary Algorithm Command Set and Control Interface ID code 16 bit ID code defining a specific algorithm |            |          |
| 014h      | 0000h              |  |            |          |
| 015h      | offset = P = 000Ah | Address for Primary Algorithm extended Query table (see <a href="#">Table 49</a> )                       |            | p = 10Ah |
| 016h      | 0001h              |  |            |          |
| 017h      | 0000h              | Alternate Vendor Command Set and Control Interface ID Code second vendor - specified algorithm supported |            | NA       |
| 018h      | 0000h              |  |            |          |
| 019h      | value = A = 0000h  | Address for Alternate Algorithm extended Query table   |            | NA       |
| 01Ah      | 0000h              |  |            |          |

Table 47. CFI query system interface information

| Offset | Data  | Description  | Value  |
|--------|-------|--|--------|
| 01Bh   | 0017h | V <sub>DD</sub> Logic Supply Minimum Program/Erase or Write voltage<br>bit 7 to 4 BCD value in volts<br>bit 3 to 0 BCD value in 100 millivolts | 1.7V   |
| 01Ch   | 0020h | V <sub>DD</sub> Logic Supply Maximum Program/Erase or Write voltage<br>bit 7 to 4 BCD value in volts<br>bit 3 to 0 BCD value in 100 millivolts | 2V     |
| 01Dh   | 0085h | V <sub>PP</sub> [Programming] Supply Minimum Program/Erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 millivolts  | 8.5V   |
| 01Eh   | 0095h | V <sub>PP</sub> [Programming] Supply Maximum Program/Erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 millivolts  | 9.5V   |
| 01Fh   | 0004h | Typical time-out per single byte/word program = 2 <sup>n</sup> μs  | 16μs   |
| 020h   | 0009h | Typical time-out for Buffer Program = 2 <sup>n</sup> μs  | 512μs  |
| 021h   | 000Ah | Typical time-out per individual block erase = 2 <sup>n</sup> ms  | 1s     |
| 022h   | 0000h | Typical time-out for full chip erase = 2 <sup>n</sup> ms   | NA     |
| 023h   | 0004h | Maximum time-out for word program = 2 <sup>n</sup> times typical   | 256μs  |
| 024h   | 0004h | Maximum time-out for Buffer Program = 2 <sup>n</sup> times typical   | 8192μs |
| 025h   | 0002h | Maximum time-out per individual block erase = 2 <sup>n</sup> times typical   | 4s     |
| 026h   | 0000h | Maximum time-out for chip erase = 2 <sup>n</sup> times typical   | NA     |

Table 48. Device geometry definition

| Offset       | Data           | Description   | Value  |  |
|--------------|----------------|---|--|--|
| 027h         | 0018h          | M58LR128KT/B device size = $2^n$ in number of bytes   | 16 Mbytes  |  |
|              | 0019h          | M58LR256KT/B device size = $2^n$ in number of bytes   | 32 Mbytes  |  |
| 028h<br>029h | 0001h<br>0000h | Flash Device Interface Code description   | x16<br>Async.  |  |
| 02Ah<br>02Bh | 0006h<br>0000h | Maximum number of bytes in multi-byte program or page = $2^n$   | 64 bytes   |  |
| 02Ch         | 0002h          | Number of identical sized erase block regions within the device<br>bit 7 to 0 = x = number of Erase Block Regions | 2  |  |
| TOP DEVICES  | 02Dh<br>02Eh   | 007Eh<br>0000h  | M58LR128KT/B Erase Block Region 1 Information<br>Number of identical-size erase blocks = 007Eh+1 | 127  |
|              |                | 00FEh<br>0000h  | M58LR256KT/B Erase Block Region 1 Information<br>Number of identical-size erase blocks = 00FEh+1 | 255  |
|              | 02Fh<br>030h   | 0000h<br>0002h  | Erase Block Region 1 Information<br>Block size in Region 1 = 0200h * 256 byte                    | 128 Kbyte  |
|              | 031h<br>032h   | 0003h<br>0000h  | Erase Block Region 2 Information<br>Number of identical-size erase blocks = 0003h+1              | 4  |
|              | 033h<br>034h   | 0080h<br>0000h  | Erase Block Region 2 Information<br>Block size in Region 2 = 0080h * 256 byte                    | 32 Kbyte   |
|              | 035h<br>038h   | Reserved  | Reserved for future erase block region information   | NA   |
|              | BOTTOM DEVICES | 02Dh<br>02Eh  | 0003h<br>0000h   | Erase Block Region 1 Information<br>Number of identical-size erase block = 0003h+1 |
| 02Fh<br>030h |                | 0080h<br>0000h  | Erase Block Region 1 Information<br>Block size in Region 1 = 0080h * 256 bytes                   | 32 Kbytes  |
| 031h<br>032h |                | 007Eh<br>0000h  | M58LR128KT/B Erase Block Region 2 Information<br>Number of identical-size erase block = 007Eh+1  | 127  |
|              |                | 00FEh<br>0000h  | M58LR256KT/B Erase Block Region 2 Information<br>Number of identical-size erase block = 00FEh+1  | 255  |
| 033h<br>034h |                | 0000h<br>0002h  | Erase Block Region 2 Information<br>Block size in Region 2 = 0200h * 256 bytes                   | 128 Kbytes   |
| 035h<br>038h |                | Reserved  | Reserved for future erase block region information   | NA   |

Table 49. Primary algorithm-specific extended query table

| Offset  | Data                                 | Description  | Value   |
|---|--------------------------------------|--|---|
| (P)h = 10Ah   | 0050h<br>0052h<br>0049h              | Primary Algorithm extended Query table unique ASCII string "PRI"   | "P"<br>"R"<br>"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<br><br>(P+7)h = 111h<br><br>(P+8)h = 112h | 00E6h<br>0003h<br>0000h<br><br>0000h | Extended Query table contents for Primary Algorithm. Address (P+5)h contains less significant byte.<br><br>bit 0 Chip Erase supported(1 = Yes, 0 = No)<br>bit 1 Erase Suspend supported(1 = Yes, 0 = No)<br>bit 2 Program Suspend supported(1 = Yes, 0 = No)<br>bit 3 Legacy Lock/Unlock supported(1 = Yes, 0 = No)<br>bit 4 Queued Erase supported(1 = Yes, 0 = No)<br>bit 5 Instant individual block locking supported(1 = Yes, 0 = No)<br>bit 6 Protection bits supported(1 = Yes, 0 = No)<br>bit 7 Page mode read supported(1 = Yes, 0 = No)<br>bit 8 Synchronous read supported(1 = Yes, 0 = No)<br>bit 9 Simultaneous operation supported(1 = Yes, 0 = No)<br>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<br>Yes<br>Yes<br>No<br>No<br>Yes<br>Yes<br>Yes<br>Yes<br>Yes |
| (P+9)h = 113h   | 0001h                                | Supported Functions after Suspend<br>Read Array, Read Status Register and CFI Query<br><br>bit 0 Program supported after Erase Suspend (1 = Yes, 0 = No)<br>bit 7 to 1 Reserved; undefined bits are '0'  | Yes   |
| (P+A)h = 114h<br><br>(P+B)h = 115h                      | 0003h<br><br>0000h                   | Block Protect Status<br>Defines which bits in the Block Status Register section of the Query are implemented.<br><br>bit 0 Block protect Status Register Lock/Unlock bit active (1 = Yes, 0 = No)<br>bit 1 Block Lock Status Register Lock-Down bit active (1 = Yes, 0 = No)<br>bit 15 to 2 Reserved for future use; undefined bits are '0'  | Yes<br>Yes  |

**Table 49. Primary algorithm-specific extended query table**

| Offset        | Data  | Description   | Value |
|---------------|-------|---|-------|
| (P+C)h = 116h | 0018h | V <sub>DD</sub> Logic Supply Optimum Program/Erase voltage (highest performance)<br><br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 mV | 1.8V  |
| (P+D)h = 117h | 0090h | V <sub>PP</sub> Supply Optimum Program/Erase voltage<br><br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 mV                             | 9V    |

**Table 50. 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   | 80h  |     |
| (P+10)h = 11Ah | 0000h | Bits 0-7 Lower byte of protection register address   | 00h  |     |
| (P+11)h = 11Bh | 0003h | Bits 8-15 Upper byte of protection register address  | 8 bytes  |     |
| (P+12)h = 11Ch | 0003h | Bits 16-23 2 <sup>n</sup> bytes in factory pre-programmed region                                       | 8 bytes  |     |
| (P+13)h = 11Dh | 0089h | Protection Register 2: Protection Description<br>Bits 0-31 protection register address                 | 89h  |     |
| (P+14)h = 11Eh | 0000h |  | Bits 0-31 protection register address                          | 00h |
| (P+15)h = 11Fh | 0000h |  | Bits 32-39 n number of factory programmed regions (lower byte) | 00h |
| (P+16)h = 120h | 0000h |  | Bits 40-47 n number of factory programmed regions (upper byte) | 00h |
| (P+17)h = 121h | 0000h |  | Bits 48-55 2 <sup>n</sup> bytes in factory programmable region | 0   |
| (P+18)h = 122h | 0000h |  | Bits 56-63 n number of user programmable regions (lower byte)  | 0   |
| (P+19)h = 123h | 0000h |  | Bits 64-71 n number of user programmable regions (upper byte)  | 16  |
| (P+1A)h = 124h | 0010h |  | Bits 72-79 2 <sup>n</sup> bytes in user programmable region    | 0   |
| (P+1B)h = 125h | 0000h |  |  | 0   |
| (P+1C)h = 126h | 0004h |  |  | 16  |

**Table 51. Burst read information**

| Offset         | Data  | Description  | Value   |
|----------------|-------|--|---------|
| (P+1D)h = 127h | 0003h | Page-mode read capability<br>bits 0-7 n' such that 2 <sup>n</sup> 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       |



**Table 51. Burst read information**

| Offset         | Data  | Description  | Value |
|----------------|-------|--|-------|
| (P+1F)h = 129h | 0001h | Synchronous mode read capability configuration 1<br>bit 3-7 Reserved<br>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 will 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 52. Bank and erase block region information<sup>(1) (2)</sup>**

| 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 that is defined at CFI offset 015h.

2. Bank Regions. There are two Bank Regions, see [Table 34](#), [Table 37](#), [Table 40](#) and [Table 43](#).

**Table 53. Bank and erase block region 1 information**

| Flash memory (top)    |      | Flash memory (bottom) |      | Description   |
|-----------------------|------|-----------------------|------|---|
| Offset <sup>(1)</sup> | Data | Offset <sup>(1)</sup> | 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:<br>Bits 0-3: Number of simultaneous program operations<br>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<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |

Table 53. Bank and erase block region 1 information (continued)

| Flash memory (top)    |  | Flash memory (bottom) |  | Description   |
|-----------------------|--|-----------------------|--|---|
| Offset <sup>(1)</sup> | Data                                     | Offset <sup>(1)</sup> | Data                                     |   |
| (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<br>Bits 0-3: Number of simultaneous program operations<br>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<br>n = number of erase block regions with contiguous same-size erase blocks.<br>Symmetrically blocked banks have one blocking region <sup>(2)</sup>                                     |
| (P+2A)h = 134h        | 07h <sup>(3)</sup><br>0Fh <sup>(4)</sup> | (P+2A)h = 134h        | 03h                                      | Bank Region 1 Erase Block Type 1 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n x 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)  |
| (P+2F)h = 139h        | 00h                                      | (P+2F)h = 139h        | 00h                                      | Minimum block erase cycles x 1000   |
| (P+30)h = 13Ah        | 01h                                      | (P+30)h = 13Ah        | 01h                                      | Bank Region 1 (Erase Block Type 1): Bits per cell, internal ECC<br>Bits 0-3: bits per cell in erase region<br>Bit 4: reserved for "internal ECC used"<br>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<br>Bit 0: Page-mode reads permitted <sup>(5)</sup><br>Bit 1: Synchronous reads permitted<br>Bit 2: Synchronous writes permitted<br>Bits 3-7: reserved |
|                       |  | (P+32)h = 13Ch        | 06h <sup>(3)</sup><br>0Eh <sup>(4)</sup> | Bank Region 1 Erase Block Type 2 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n x 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)  |
|                       |  | (P+37)h = 141h        | 00h                                      | Minimum block erase cycles x 1000   |

**Table 53. Bank and erase block region 1 information (continued)**

| Flash memory (top)    |      | Flash memory (bottom) |      | Description  |
|-----------------------|------|-----------------------|------|--|
| Offset <sup>(1)</sup> | Data | Offset <sup>(1)</sup> | Data |  |
|                       |      | (P+38)h = 142h        | 01h  | Bank Regions 1 (Erase Block Type 2): Blts per cell, internal ECC<br>Bits 0-3: bits per cell in erase region<br>Bit 4: reserved for "internal ECC used"<br>Bits 5-7: reserved   |
|                       |      | (P+39)h = 143h        | 03h  | Bank Region 1 (Erase Block Type 2): Page mode and Synchronous mode capabilities<br>Bit 0: Page-mode reads permitted<br>Bit 1: Synchronous reads permitted<br>Bit 2: Synchronous writes permitted<br>Bits 3-7: reserved |

1. The variable P is a pointer that is defined at CFI offset 015h.
2. Bank Regions. There are two Bank Regions, see [Table 34](#), [Table 37](#), [Table 40](#) and [Table 43](#).
3. Applies to M58LR128KT/B only.
4. Applies to M58LR256KT/B only.
5. Although the device supports Page Read mode, this is not described in the datasheet as its use is not advantageous in a multiplexed device.

**Table 54. Bank and erase block region 2 information**

| Flash memory (top)    |      | Flash memory (bottom) |      | Description   |
|-----------------------|------|-----------------------|------|---|
| Offset <sup>(1)</sup> | Data | Offset <sup>(1)</sup> | 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:<br>Bits 0-3: Number of simultaneous program operations<br>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<br>Bits 0-3: Number of simultaneous program operations<br>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<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations     |

Table 54. Bank and erase block region 2 information (continued)

| Flash memory (top)    |  | Flash memory (bottom) |  | Description   |
|-----------------------|--|-----------------------|--|---|
| Offset <sup>(1)</sup> | Data                                     | Offset <sup>(1)</sup> | Data                                     |   |
| (P+37)h = 141h        | 02h                                      | (P+3F)h = 149h        | 01h                                      | Types of erase block regions in Bank Region 2<br>n = number of erase block regions with contiguous same-size erase blocks.<br>Symmetrically blocked banks have one blocking region. <sup>(2)</sup>  |
| (P+38)h = 142h        | 06h <sup>(3)</sup><br>0Eh <sup>(4)</sup> | (P+40)h = 14Ah        | 07h <sup>(3)</sup><br>0Fh <sup>(4)</sup> | Bank Region 2 Erase Block Type 1 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n x 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                                      |   |
| (P+3D)h = 147h        | 00h                                      | (P+45)h = 14Fh        | 00h                                      | Bank Region 2 (Erase Block Type 1)<br>Minimum block erase cycles x 1000   |
| (P+3E)h = 148h        | 01h                                      | (P+46)h = 150h        | 01h                                      | Bank Region 2 (Erase Block Type 1): Blts per cell, internal ECC<br>Bits 0-3: bits per cell in erase region<br>Bit 4: reserved for "internal ECC used"<br>Blts 5-7: reserved   |
| (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 <a href="#">Table 51</a> )<br>Bit 0: Page-mode reads permitted <sup>(5)</sup><br>Bit 1: Synchronous reads permitted<br>Bit 2: Synchronous writes permitted<br>Bits 3-7: reserved |
| (P+40)h = 14Ah        | 03h                                      |                       |  | Bank Region 2 Erase Block Type 2 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n x 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)<br>Minimum block erase cycles x 1000   |
| (P+45)h = 14Fh        | 00h                                      |                       |  |   |
| (P+46)h = 150h        | 01h                                      |                       |  | Bank Region 2 (Erase Block Type 2): Blts per cell, internal ECC<br>Bits 0-3: bits per cell in erase region<br>Bit 4: reserved for "internal ECC used"<br>Blts 5-7: reserved   |

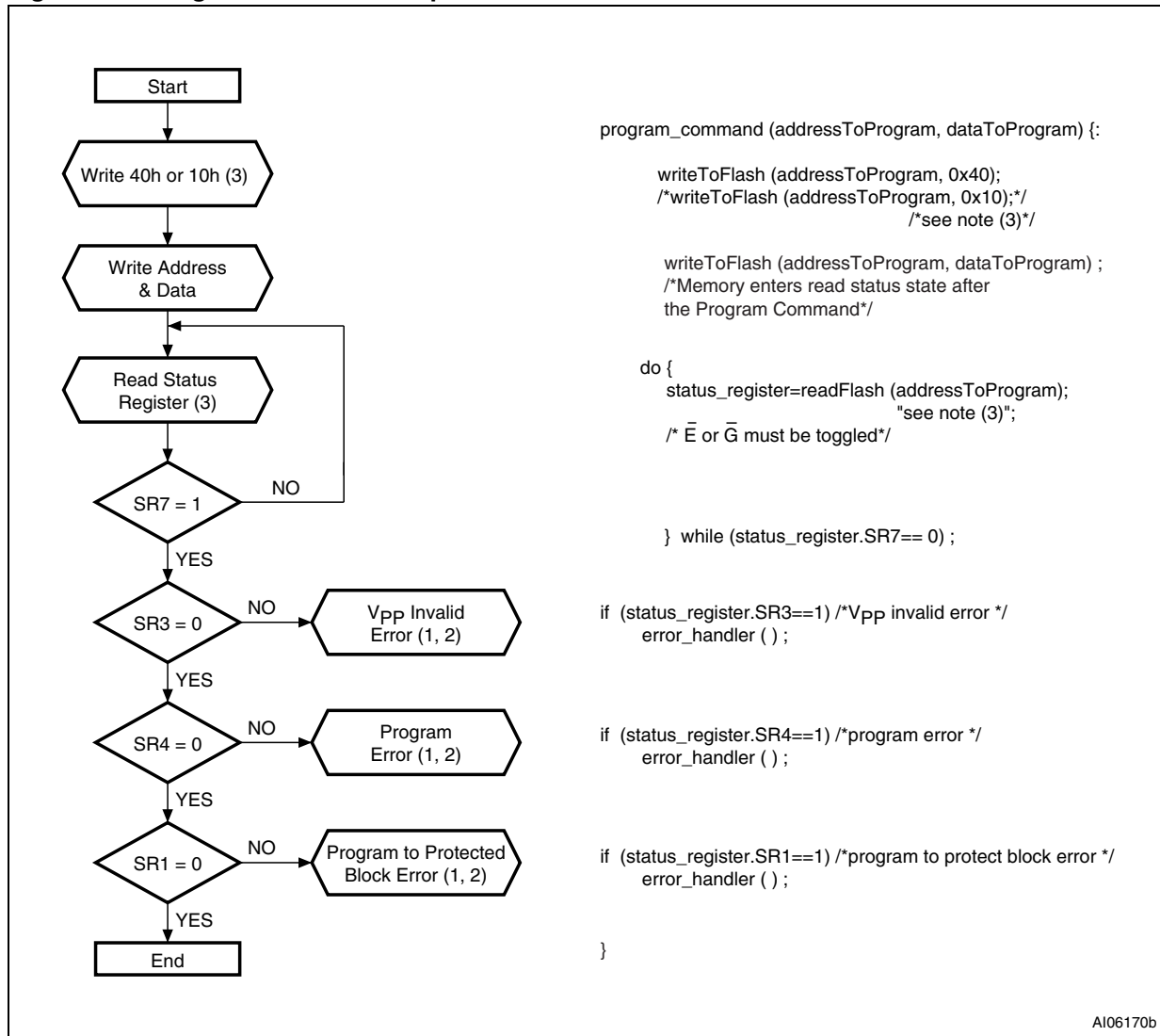
Table 54. Bank and erase block region 2 information (continued)

| Flash memory (top)    |      | Flash memory (bottom) |      | Description   |
|-----------------------|------|-----------------------|------|---|
| Offset <sup>(1)</sup> | Data | Offset <sup>(1)</sup> | Data |   |
| (P+47)h = 151h        | 03h  |                       |      | Bank Region 2 (Erase Block Type 2): Page mode and Synchronous mode capabilities (defined in <a href="#">Table 51</a> )<br>Bit 0: Page-mode reads permitted<br>Bit 1: Synchronous reads permitted<br>Bit 2: Synchronous writes permitted<br>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 34](#), [Table 37](#), [Table 40](#) and [Table 43](#).
3. Applies to M58LR128KT/B only.
4. Applies to M58LR256KT/B only.
5. Although the device supports Page Read mode, this is not described in the datasheet as its use is not advantageous in a multiplexed device.

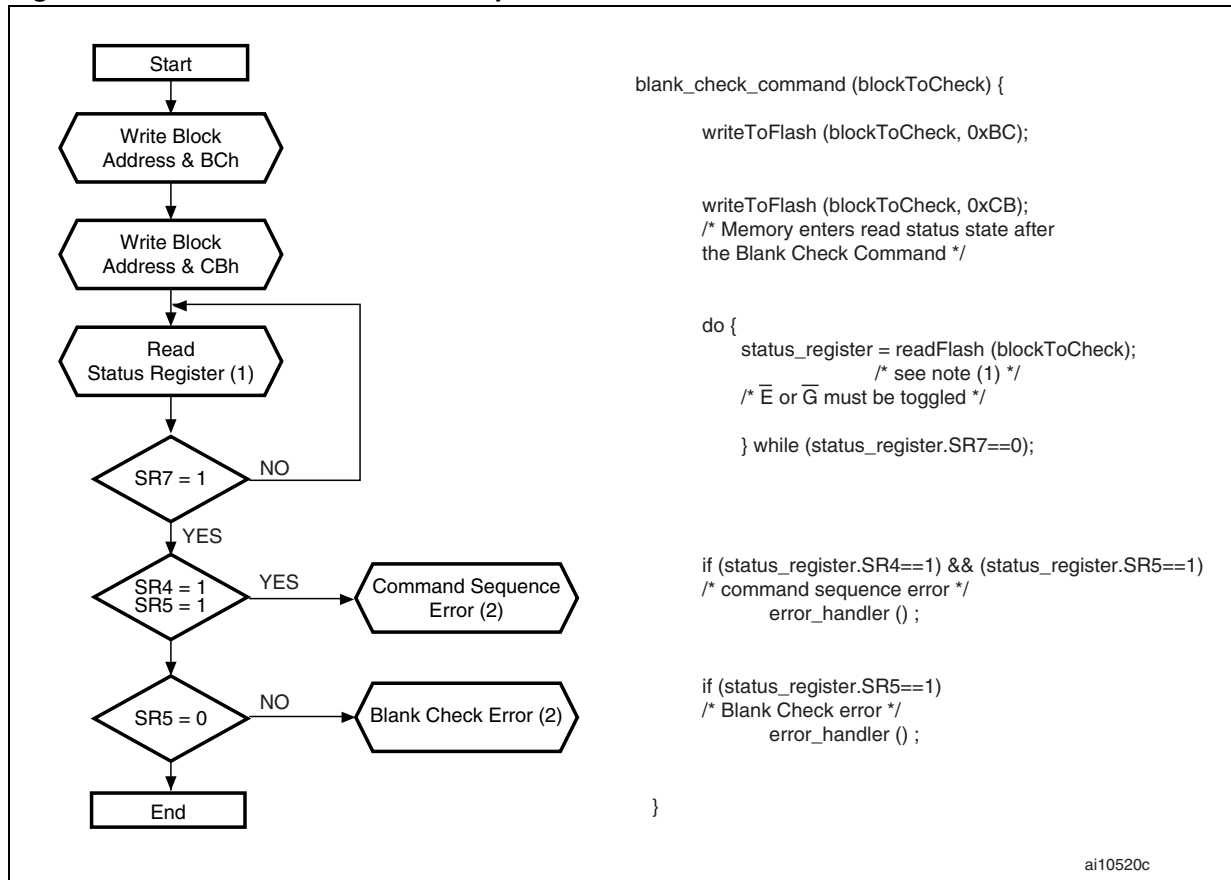
## Appendix C Flowcharts and pseudocodes

Figure 24. Program flowchart and pseudocode



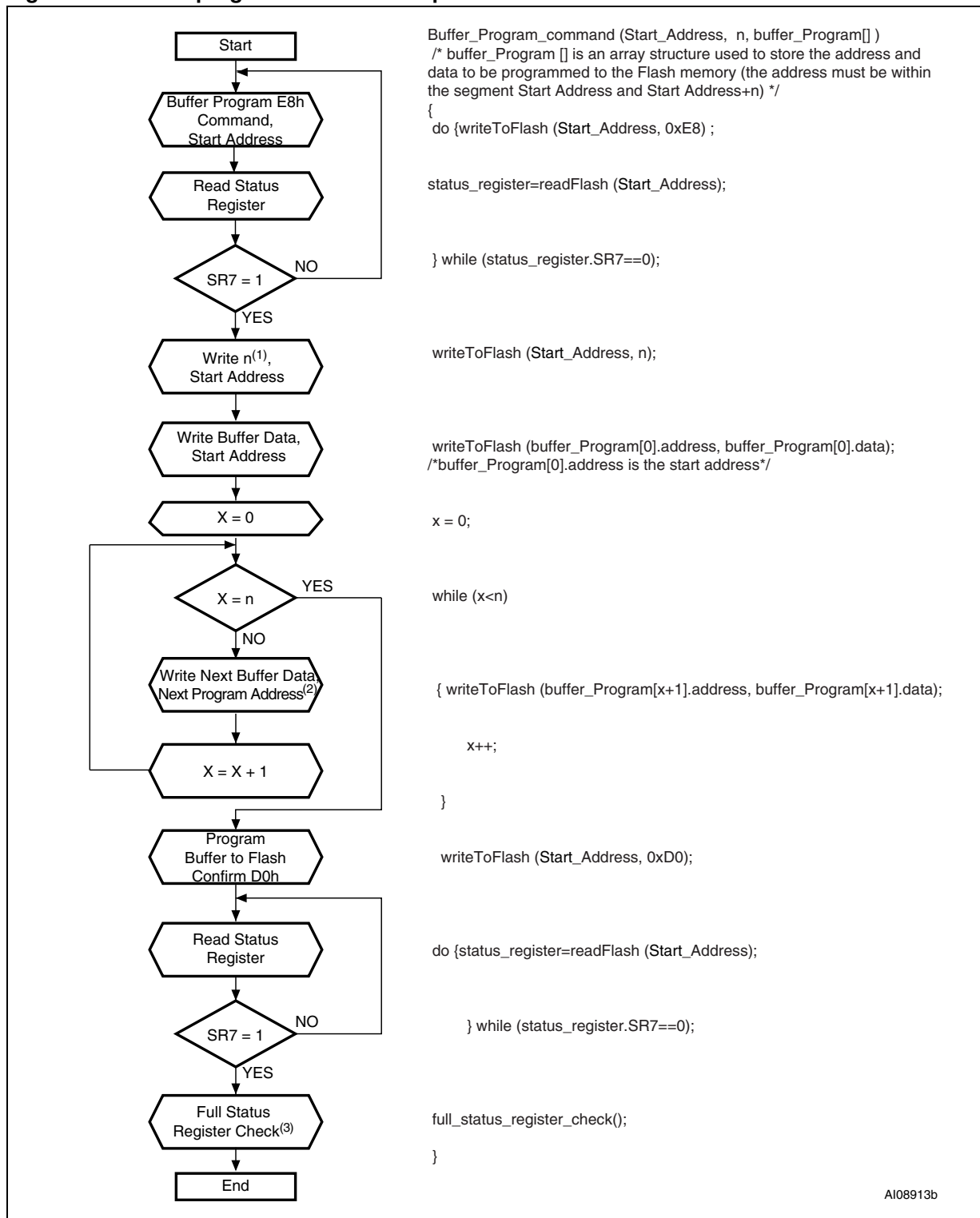
1. Status check of SR1 (Protected Block), SR3 (V<sub>PP</sub> Invalid) and SR4 (Program Error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase Controller operations.
3. Any address within the bank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

Figure 25. Blank check flowchart and pseudocode



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.

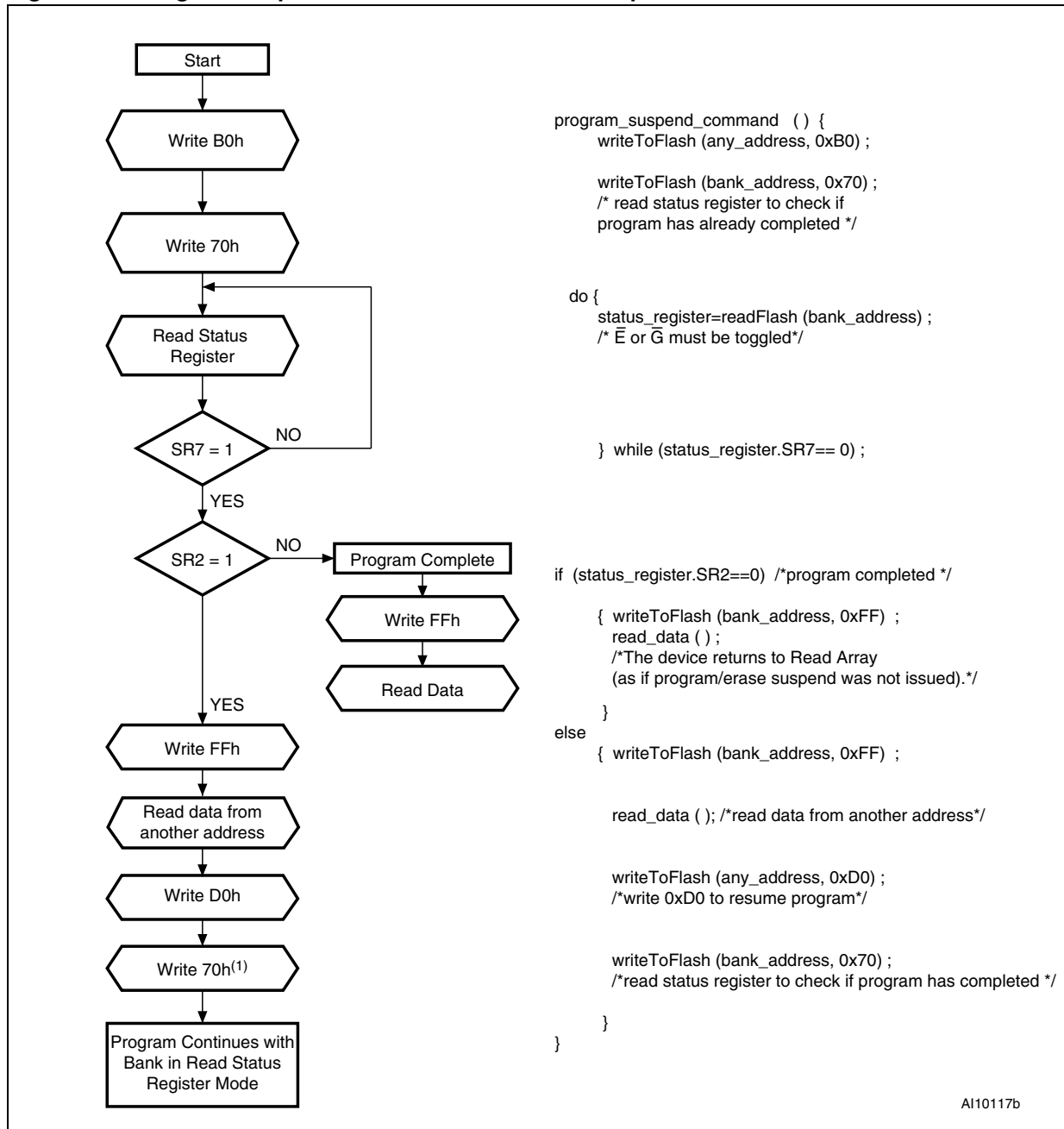
Figure 26. Buffer program flowchart and pseudocode



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.

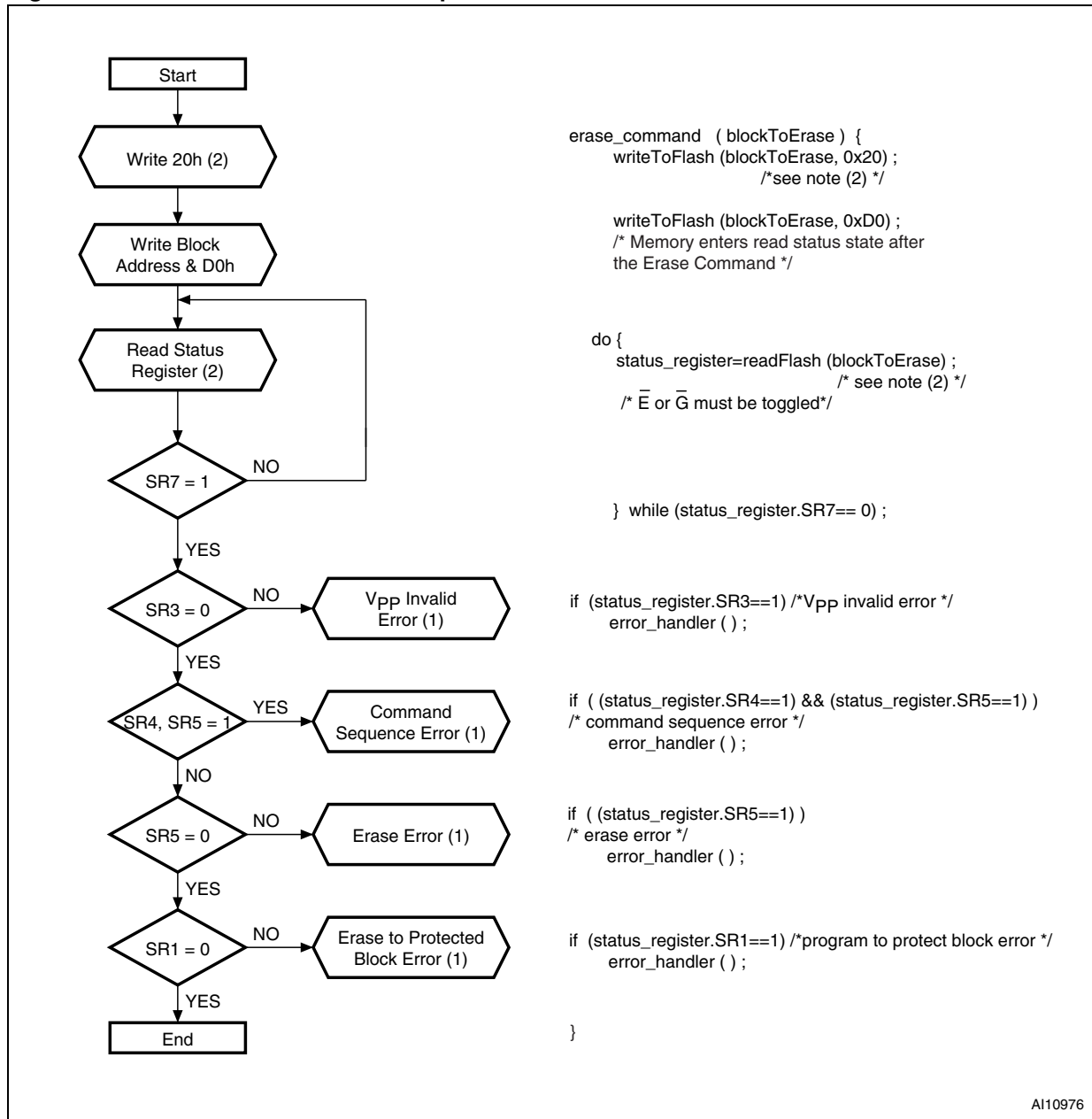


Figure 27. Program suspend and resume flowchart and pseudocode



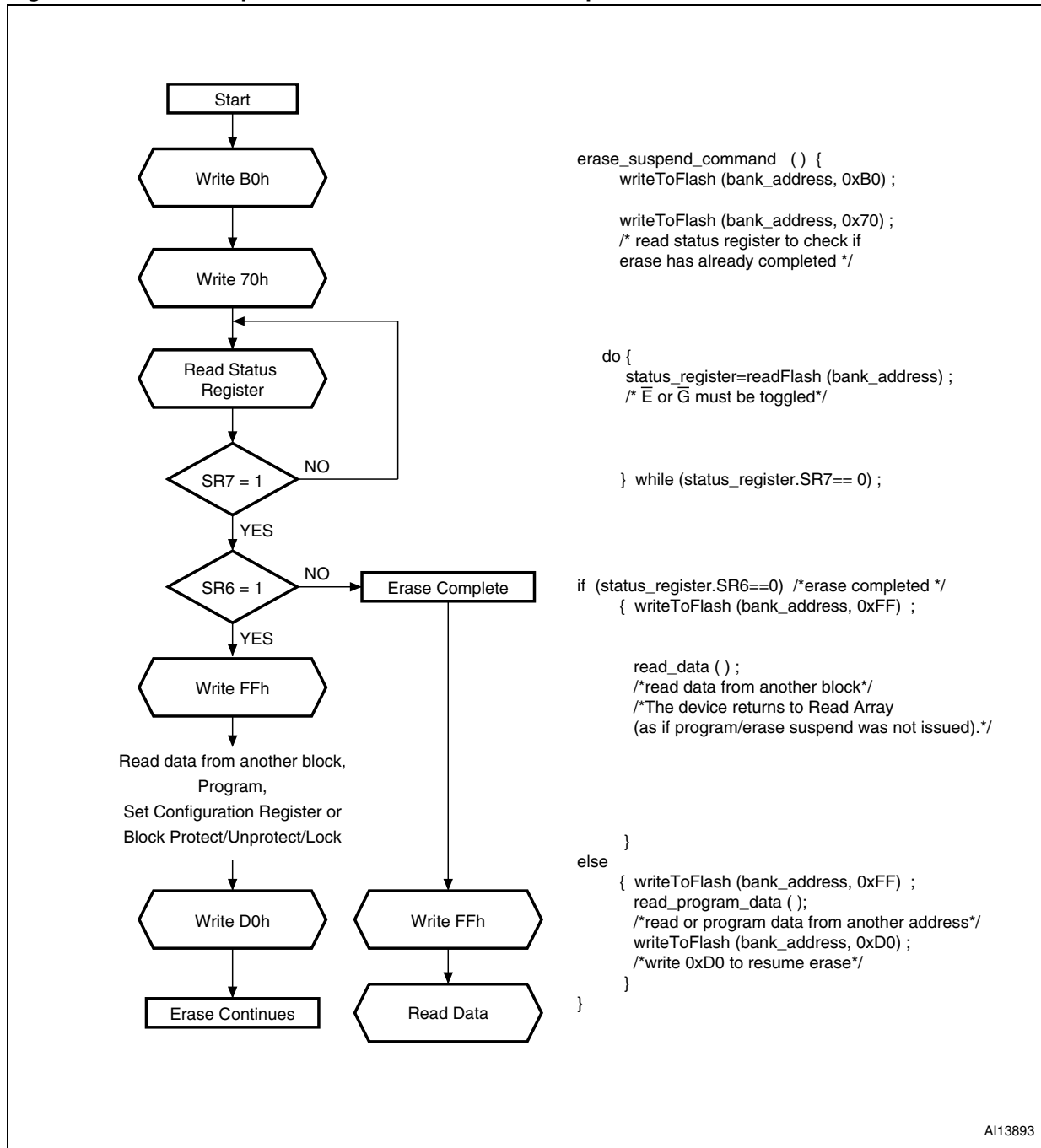
1. The Read Status Register command (Write 70h) can be issued just before or just after the Program Resume command.

Figure 28. Block erase flowchart and pseudocode



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.

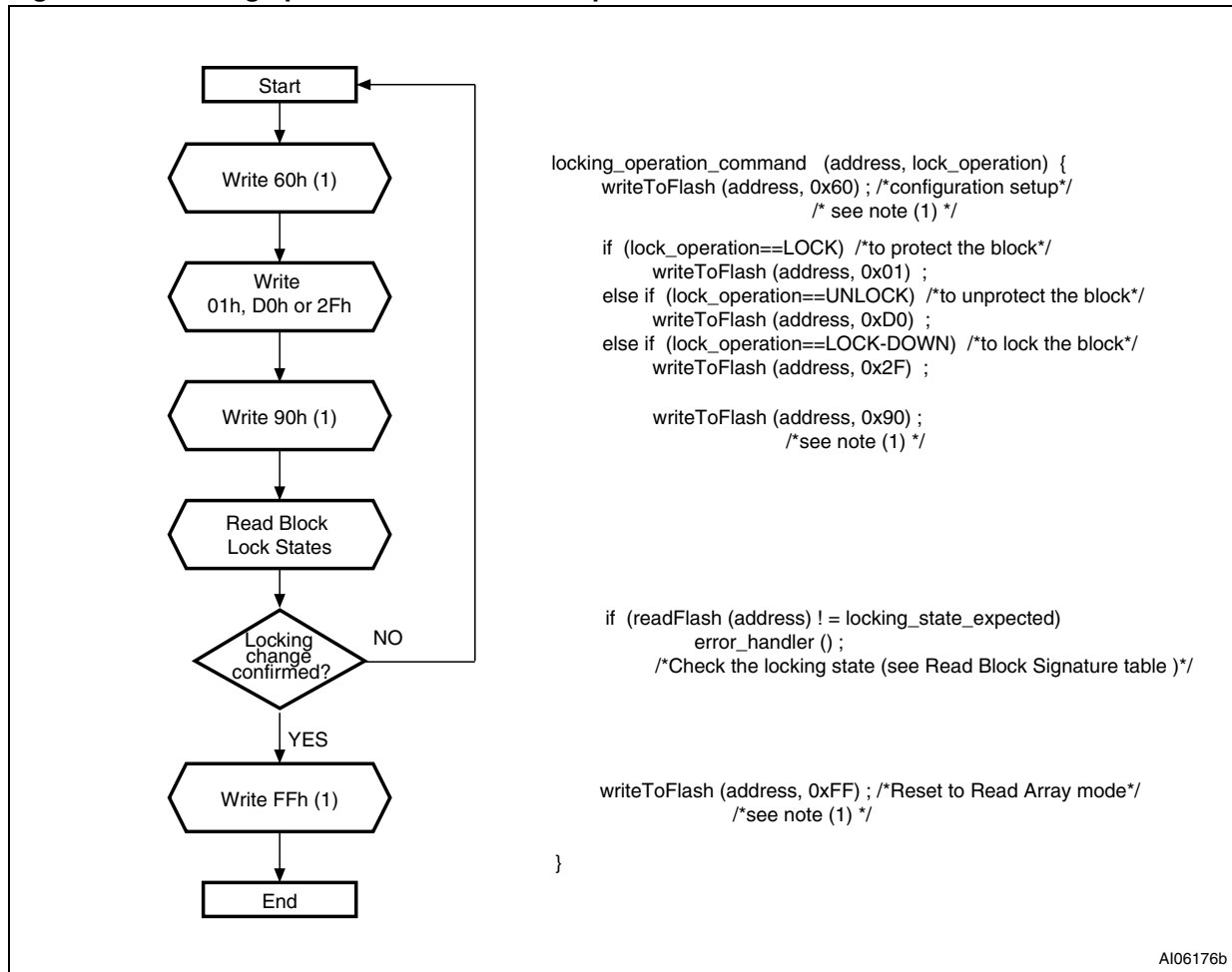
Figure 29. Erase suspend and resume flowchart and pseudocode



AI13893

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

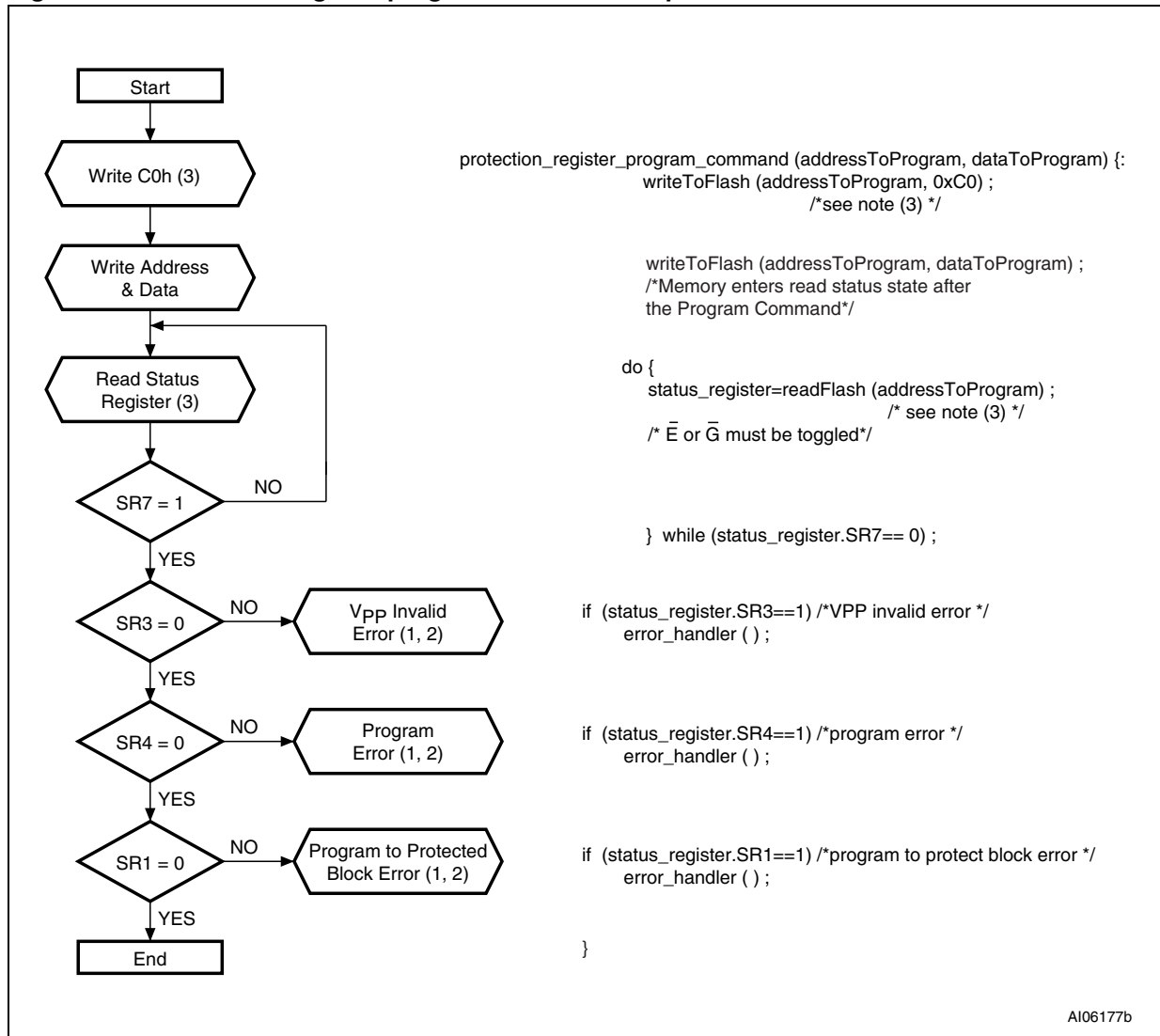
Figure 30. Locking operations flowchart and pseudocode



AI06176b

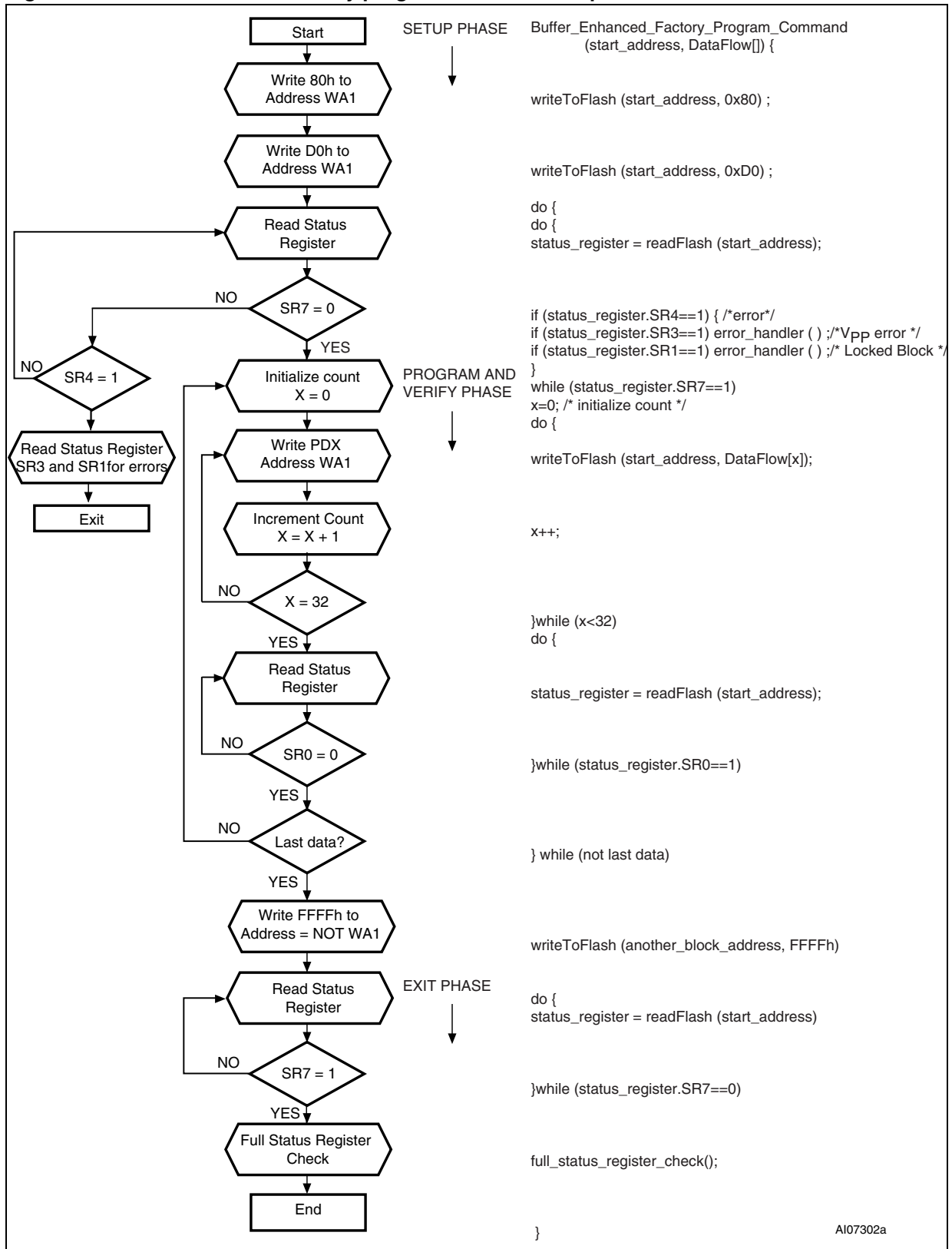
1. Any address within the bank can equally be used.

Figure 31. Protection Register program flowchart and pseudocode



1. Status check of SR1 (Protected Block), SR3 (V<sub>PP</sub> 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 32. Buffer enhanced factory program flowchart and pseudocode



## Appendix D Command interface state tables

Table 55. Command interface states - modify table, next state<sup>(1)</sup>

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

Table 55. Command interface states - modify table, next state<sup>(1)</sup> (continued)

| Current CI State                |                          | Command Input  |  |  |  |                    |                          |  |                               |  |                            |  |
|---------------------------------|--------------------------|--|--|--|--|--------------------|--------------------------|--|-------------------------------|--|----------------------------|--|
|                                 |                          | Read Array <sup>(2)</sup> (FFh)  | Program Setup <sup>(3)(4)</sup> (10/40h) | Buffer Program <sup>(3)(4)</sup> (E8h) | Block Erase, Setup <sup>(3)(4)</sup> (20h) | BEFP Setup (80h)   | Blank Check setup (BCh)  | Erase Confirm P/E Resume, Block Unlock confirm, BEFP Confirm <sup>(3)(4)</sup> (D0h) | Blank Check confirm (CBh)     | Buffer Program, Program/ Erase Suspend (B0h) | Read Status Register (70h) | Clear Status Register <sup>(5)</sup> (50h) |
| 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 will fail 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 55. Command interface states - modify table, next state<sup>(1)</sup> (continued)

| Current CI State               |       | Command Input                   |  |  |  |                  |                         |  |                           |  |                            |  |
|--------------------------------|-------|---------------------------------|--|--|--|------------------|-------------------------|--|---------------------------|--|----------------------------|--|
|                                |       | Read Array <sup>(2)</sup> (FFh) | Program Setup <sup>(3)(4)</sup> (10/40h) | Buffer Program <sup>(3)(4)</sup> (E8h) | Block Erase, Setup <sup>(3)(4)</sup> (20h) | BEFP Setup (80h) | Blank Check setup (BCh) | Erase Confirm P/E Resume, Block Unlock confirm, BEFP Confirm <sup>(3)(4)</sup> (D0h) | Blank Check confirm (CBh) | Buffer Program, Program/ Erase Suspend (B0h) | Read Status Register (70h) | Clear Status Register <sup>(5)</sup> (50h) |
| 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 |                         |  |                           |  |                            |  |
| Lock/CR Setup in Erase Suspend |       | Erase Suspend (Lock Error)      |  |  |  |                  | Erase Suspend           | Erase Suspend (Lock Error)   |                           |  |                            |  |
| Buffer EFP                     | Setup | Ready (error)                   |  |  |  |                  | BEFP Busy               | Ready (error)  |                           |  |                            |  |
|                                | Busy  | BEFP Busy <sup>(6)</sup>        |  |  |  |                  |                         |  |                           |  |                            |  |

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 P/E C is active, both cycles are ignored.
5. The Clear Status Register command clears the SR error bits except when the P/E C. 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 56. Command interface states - modify table, next output state<sup>(1) (2)</sup>

| Current CI State                        | Command Input                      |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
|---|------------------------------------|---|-------------------------|---|---------------------|----------------------------|---|------------------------------|---------------------------------|-------------------------------|--------------------------------|---|
|   | Read Array <sup>(3)</sup><br>(FFh) | Program Setup <sup>(4)(5)</sup><br>(10/40h) | Buffer Program<br>(E8h) | Block Erase, Setup <sup>(4)(5)</sup><br>(20h) | BEFP Setup<br>(80h) | Blank Check setup<br>(BCh) | Erase Confirm P/E Resume, Block Unlock confirm, BEFP Confirm <sup>(4)(5)</sup><br>(D0h) | Blank Check confirm<br>(CBh) | Program/ Erase Suspend<br>(B0h) | Read Status Register<br>(70h) | Clear Status Register<br>(50h) | Read Electronic signature, Read CFI Query<br>(90h, 98h) |
| Program Setup                           | Status Register                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Erase Setup                             |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| OTP Setup                               |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Program Setup in Erase Suspend          |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| BEFP Setup                              |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| BEFP Busy                               |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Setup                    |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Load 1                   |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Load 2                   |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Confirm                  |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Setup in Erase Suspend   |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Load 1 in Erase Suspend  |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Load 2 in Erase Suspend  |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Buffer Program Confirm in Erase Suspend |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Blank Check setup                       |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Lock/CR Setup                           |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |
| Lock/CR Setup in Erase Suspend          |                                    |   |                         |   |                     |                            |   |                              |                                 |                               |                                |   |

Table 56. Command interface states - modify table, next output state<sup>(1)</sup> <sup>(2)</sup> (continued)

| Current CI State                        | Command Input                      |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
|---|------------------------------------|---|-------------------------|---|---------------------|----------------------------|---|------------------------------|---------------------------------------|-------------------------------|--------------------------------|--|
|   | Read Array <sup>(3)</sup><br>(FFh) | Program Setup <sup>(4)</sup> <sup>(5)</sup><br>(10/40h) | Buffer Program<br>(E8h) | Block Erase, Setup <sup>(4)</sup> <sup>(5)</sup><br>(20h) | BEFP Setup<br>(80h) | Blank Check setup<br>(BCh) | Erase Confirm<br>P/E Resume,<br>Block Unlock<br>confirm, BEFP<br>Confirm <sup>(4)</sup> <sup>(5)</sup><br>(D0h) | Blank Check confirm<br>(CBh) | Program/<br>Erase<br>Suspend<br>(B0h) | Read Status Register<br>(70h) | Clear Status Register<br>(50h) | Read Electronic signature,<br>Read CFI Query<br>(90h, 98h) |
| OTP Busy                                | Array                              | Status Register   | Status Register         | Output Unchanged  | Output Unchanged    | Status Register            | Output Unchanged  | Electronic Signature/<br>CFI | Status Register                       | Electronic Signature/<br>CFI  | Electronic Signature/<br>CFI   | Electronic Signature/<br>CFI                               |
| Ready                                   |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Program Busy                            |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Erase Busy                              |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Buffer Program Busy                     |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Program/Erase Suspend                   |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Buffer Program Suspend                  |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Program Busy in Erase Suspend           |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Buffer Program Busy in Erase Suspend    |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Program Suspend in Erase Suspend        |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Buffer Program Suspend in Erase Suspend |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| Blank Check busy                        |                                    |   |                         |   |                     |                            |   |                              |                                       |                               |                                |  |
| 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, P/E. C. = 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 P/EC is active, both cycles are ignored.

Table 57. Command interface states - lock table, next state<sup>(1)</sup>

| Current CI State |                           | Command Input  |                                   |                             |                                  |                         |   |                                | P/E C operation completed <sup>(5)</sup> |
|------------------|---------------------------|--|-----------------------------------|-----------------------------|----------------------------------|-------------------------|---|--------------------------------|--|
|                  |                           | Lock/CR Setup <sup>(2)</sup><br>(60h)  | OTP Setup <sup>(2)</sup><br>(C0h) | Block Lock Confirm<br>(01h) | Block Lock-Down Confirm<br>(2Fh) | Set CR Confirm<br>(03h) | Block Address (WA0) <sup>(3)</sup><br>(XXXXh) | Illegal Command <sup>(4)</sup> |  |
| Ready            |                           | Lock/CR Setup  | OTP Setup                         | Ready                       |                                  |                         |   | N/A                            |  |
| Lock/CR Setup    |                           | Ready (Lock error)   |                                   | Ready                       |                                  | Ready (Lock 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 <sup>(6)</sup>   |                                   |                             |                                  | Exit                    | see note <sup>(6)</sup>                       | N/A                            |  |
|                  | Buffer Load 2             | Buffer Program Confirm when count =0; Else Buffer Program Load 2 (note: Buffer Program will fail 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                   | Lock/CR Setup in ES  | IS in ES                          | Erase Suspend               |                                  |                         |   | N/A                            |  |
|                  | IS in ES                  | Erase Suspend  |                                   |                             |                                  |                         |   |                                |  |

Table 57. Command interface states - lock table, next state<sup>(1)</sup> (continued)

| Current CI State                |                          | Command Input  |   |                             |                                  |                            |   |                                | P/E C operation completed <sup>(5)</sup> |
|---------------------------------|--------------------------|--|---|-----------------------------|----------------------------------|----------------------------|---|--------------------------------|--|
|                                 |                          | Lock/CR Setup <sup>(2)</sup><br>(60h)  | OTP Setup <sup>(2)</sup><br>(C0h)       | Block Lock Confirm<br>(01h) | Block Lock-Down Confirm<br>(2Fh) | Set CR Confirm<br>(03h)    | Block Address (WA0) <sup>(3)</sup><br>(XXXXh) | Illegal Command <sup>(4)</sup> |  |
| 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 <sup>(7)</sup>  |   |                             |                                  | Exit                       | see note <sup>(7)</sup>                       |                                |  |
|                                 | Buffer Load 2            | Buffer Program Confirm in Erase Suspend when count =0; Else Buffer Program Load 2 in Erase Suspend (note: Buffer Program will fail 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                                    |
| Lock/CR Setup in ES             |                          | Erase Suspend (Lock error)   | Erase Suspend                           |                             |                                  | Erase Suspend (Lock error) |   | N/A                            |  |
| BEFP                            | Setup                    | Ready (error)  |   |                             |                                  |                            |   |                                | N/A                                      |
|                                 | Busy                     | BEFP Busy <sup>(8)</sup>   |   |                             |                                  | Exit                       | BEFP Busy <sup>(8)</sup>                      |                                | 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 P/E C 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 58. Command interface states - lock table, next output state (1) (2)**

| Current CI State                        | Command Input                      |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
|---|------------------------------------|-------------------------|--------------------------------|---------------------------|--------------------------|-------------------------------|----------------------|---------------------------------|--------------------------------|------------------------------|
|   | Lock/CR Setup <sup>(3)</sup> (60h) | Blank Check setup (BCh) | OTP Setup <sup>(3)</sup> (C0h) | Blank Check confirm (CBh) | Block Lock Confirm (01h) | Block Lock-Down Confirm (2Fh) | Set CR Confirm (03h) | BEFP Exit <sup>(4)</sup> (FFFh) | Illegal Command <sup>(5)</sup> | P. E./C. Operation Completed |
| Program Setup                           | Status Register                    |                         |                                |                           |                          |                               |                      |                                 |                                | Output Unchanged             |
| Erase Setup                             |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| OTP Setup                               |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Program Setup in Erase Suspend          |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| BEFP Setup                              |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| BEFP Busy                               |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Setup                    |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Load 1                   |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Load 2                   |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Confirm                  |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Setup in Erase Suspend   |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Load 1 in Erase Suspend  |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Load 2 in Erase Suspend  |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Confirm in Erase Suspend |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Blank Check setup                       | Status Register                    |                         |                                |                           |                          |                               |                      |                                 |                                | Output Unchanged             |
| Lock/CR Setup                           |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Lock/CR Setup in Erase Suspend          | Status Register                    |                         |                                |                           |                          |                               |                      |                                 |                                | Output Unchanged             |

**Table 58. Command interface states - lock table, next output state (continued)<sup>(1) (2)</sup>**

| Current CI State                        | Command Input                      |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
|---|------------------------------------|-------------------------|--------------------------------|---------------------------|--------------------------|-------------------------------|----------------------|---------------------------------|--------------------------------|------------------------------|
|   | Lock/CR Setup <sup>(3)</sup> (60h) | Blank Check setup (BCh) | OTP Setup <sup>(3)</sup> (C0h) | Blank Check confirm (CBh) | Block Lock Confirm (01h) | Block Lock-Down Confirm (2Fh) | Set CR Confirm (03h) | BEFP Exit <sup>(4)</sup> (FFFh) | Illegal Command <sup>(5)</sup> | P. E./C. Operation Completed |
| OTP Busy                                | Status Register                    |                         |                                | Output Unchanged          |                          |                               |                      | Array                           | Output Unchanged               |                              |
| Ready                                   |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Program Busy                            |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Erase Busy                              |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Busy                     |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Program/Erase Suspend                   |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Suspend                  |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Program Busy in Erase Suspend           |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Busy in Erase Suspend    |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Program Suspend in Erase Suspend        |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Buffer Program Suspend in Erase Suspend |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| Blank Check busy                        |                                    |                         |                                |                           |                          |                               |                      |                                 |                                |                              |
| 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's output state.
2. CI = Command Interface, CR = Configuration Register, BEFP = Buffer Enhanced Factory Program, P/E. C. = Program/Erase Controller.
3. If the P/EC 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.

## 15 Revision history

**Table 59. Document revision history**

| Date        | Revision | Changes  |
|-------------|----------|--|
| 23-Mar-2007 | 1        | Initial release.   |
| 06-Dec-2007 | 2        | <p>In <a href="#">Table 18: Program/erase times and endurance cycles</a>, changed the suspend latency program and erase values to 5, 10 and 5, 20 respectively, to 20 and 25 for both.</p> <p>In <a href="#">Table 19: Absolute maximum ratings</a>, changed the <math>V_{IO}</math> maximum value from 3.8 to <math>V_{DDQ} + 0.6</math> V.</p> <p>In <a href="#">Table 22: DC characteristics - currents</a>:</p> <ul style="list-style-type: none"> <li>– Added the 256 Mbit values for <math>I_{DD2}</math>, <math>I_{DD3}</math>, and <math>I_{DD7}</math>.</li> <li>– For <math>I_{DD5}</math>, changed the supply current (both program and erase) <math>V_{PPH}</math> values to 10 and 30, and changed the <math>V_{DD}</math> and values to 20 and 35, respectively.</li> <li>– For <math>I_{DD6}</math>, changed the “asynchronous read in another bank” values to 33 and 50, respectively; changed the “synchronous read in another bank” values to 50 and 67, respectively.</li> </ul> <p>In <a href="#">Table 28: Reset and power-up AC characteristics</a>, changed the <math>t_{VDHPH}</math> value from 250 to 300.</p> |
| 27-Mar-2008 | 3        | Applied Numonyx branding.  |
| 01-Jul-2008 | 4        | Added <a href="#">Figure 3: TFBGA88 connections (top view through package)</a> , <a href="#">Figure 21: TFBGA88 8 x 10 mm - 8 x 10 ball array, 0.8 mm pitch, bottom view package outline</a> , <a href="#">Table 23: VFBGA79 9x11mm - 0.75 pitch -package outline</a> , and amended <a href="#">Table 32: Ordering information scheme</a> to provide the ZB package information, which is available for the M58LR128KT/B devices.  |
| 11-Sep-2008 | 5        | Add 79B vFBGA 9x11 package for 256Mbit. Add 88b package 8x10.<br>Change from T=-25°C to T=-30°C<br>Change Terase for Boot sectors (0.4s to 0.6s)   |
| 23-Oct-2008 | 6        | Change the package reference from ZA to ZQ.  |



**Please Read Carefully:**

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH NUMONYX™ PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN NUMONYX'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, NUMONYX ASSUMES NO LIABILITY WHATSOEVER, AND NUMONYX DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF NUMONYX PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Numonyx products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Numonyx may make changes to specifications and product descriptions at any time, without notice.

Numonyx, B.V. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Numonyx reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Contact your local Numonyx sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Numonyx literature may be obtained by visiting Numonyx's website at <http://www.numonyx.com>.

Numonyx StrataFlash is a trademark or registered trademark of Numonyx or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 11/5/7, Numonyx, B.V., All Rights Reserved.