

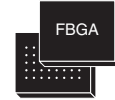


# M58PR256LE M58PR512LE M58PR001LE

256-Mbit, 512-Mbit or 1-Gbit (× 16, multiple bank, multilevel, burst)  
1.8 V supply Flash memories

## 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: 108 MHz, 66 MHz
  - Asynchronous page read mode
  - Random access: 96 ns
- Programming time
  - 4.2  $\mu\text{s}$  typical word program time using Buffer Enhanced Factory Program command
- Memory organization
  - Multiple bank memory array:
    - 32 Mbit banks (256 Mb devices)
    - 64 Mbit banks (512 Mb devices)
    - 128 Mbit banks (1 Gb devices)
  - Four EFA (extended flash array) blocks of 64 Kbits
- 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  $V_{PP} = V_{SS}$



TFBGA105 (ZAD)  
9 x 11 mm

TFBGA107 (ZAC)  
8 x 11 mm

- Security
  - 64 bit unique device number
  - 2112 bit user programmable OTP cells
- CFI (common Flash interface)
- 100 000 program/erase cycles per block
- Electronic signature
  - Manufacturer code: 20h
  - 256 Mbit device: 8818
  - 512 Mbit device: 8819
  - 1 Gbit device: 880F
- ECOPACK® package available.

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Description</b>                     | <b>9</b>  |
| <b>2</b> | <b>Signal descriptions</b>             | <b>17</b> |
| 2.1      | Address inputs (A0-Amax)               | 17        |
| 2.2      | Data inputs/outputs (DQ0-DQ15)         | 17        |
| 2.3      | Chip Enable ( $\overline{E}$ )         | 17        |
| 2.4      | Output Enable ( $\overline{G}$ )       | 17        |
| 2.5      | Write Enable ( $\overline{W}$ )        | 17        |
| 2.6      | Write Protect ( $\overline{WP}$ )      | 17        |
| 2.7      | Reset ( $\overline{RP}$ )              | 18        |
| 2.8      | Deep power-down (DPD)                  | 18        |
| 2.9      | Latch Enable ( $\overline{L}$ )        | 18        |
| 2.10     | Clock (K)                              | 18        |
| 2.11     | Wait (WAIT)                            | 18        |
| 2.12     | V <sub>DD</sub> supply voltage         | 19        |
| 2.13     | V <sub>DDQ</sub> supply voltage        | 19        |
| 2.14     | V <sub>PP</sub> program supply voltage | 19        |
| 2.15     | V <sub>SS</sub> ground                 | 19        |
| 2.16     | V <sub>SSQ</sub> ground                | 19        |
| <b>3</b> | <b>Bus operations</b>                  | <b>20</b> |
| 3.1      | Bus Read                               | 20        |
| 3.2      | Bus Write                              | 20        |
| 3.3      | Address Latch                          | 20        |
| 3.4      | Output Disable                         | 20        |
| 3.5      | Standby                                | 21        |
| 3.6      | Reset                                  | 21        |
| 3.7      | Deep power-down (DPD)                  | 21        |
| <b>4</b> | <b>Command interface</b>               | <b>23</b> |
| 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      | Program command                             | 27        |
| 4.8      | Buffer Program command                      | 28        |
| 4.9      | Buffer Enhanced Factory Program command     | 29        |
| 4.9.1    | Setup phase                                 | 29        |
| 4.9.2    | Program and verify phase                    | 30        |
| 4.9.3    | Exit phase                                  | 30        |
| 4.10     | Program/Erase Suspend command               | 31        |
| 4.11     | Program/Erase Resume command                | 32        |
| 4.12     | Protection Register Program command         | 32        |
| 4.13     | Set Configuration Register command          | 33        |
| 4.14     | Block Lock command                          | 33        |
| 4.15     | Block Unlock command                        | 33        |
| 4.16     | Block Lock-down command                     | 34        |
| 4.17     | Blank Check command                         | 34        |
| 4.18     | Set Enhanced Configuration Register command | 35        |
| 4.19     | Read EFA Block command                      | 35        |
| 4.20     | Program EFA Block command                   | 36        |
| 4.21     | Erase EFA Block command                     | 36        |
| 4.22     | Suspend EFA Block command                   | 37        |
| 4.23     | Resume EFA Block command                    | 38        |
| 4.24     | Lock EFA Block command                      | 38        |
| 4.25     | Unlock EFA Block command                    | 39        |
| 4.26     | Lock-down EFA Block command                 | 39        |
| <b>5</b> | <b>Program operations</b>                   | <b>43</b> |
| 5.1      | Program regions                             | 43        |
| 5.2      | Program modes                               | 44        |
| 5.2.1    | Control program mode                        | 44        |
| 5.2.2    | Object program mode                         | 44        |
| 5.3      | Program methods                             | 45        |

|           |   |           |
|-----------|---|-----------|
| 5.3.1     | Single word program method                        | 46        |
| 5.3.2     | Buffer program method                             | 46        |
| 5.3.3     | Buffer enhanced factory program method            | 47        |
| <b>6</b>  | <b>Status Register</b>                            | <b>48</b> |
| 6.1       | Control program mode status bit (SR9)             | 48        |
| 6.2       | Object program mode status bit (SR8)              | 49        |
| 6.3       | Program/Erase Controller status bit (SR7)         | 49        |
| 6.4       | Erase suspend status bit (SR6)                    | 49        |
| 6.5       | Erase status bit (SR5)                            | 50        |
| 6.6       | Program status bit (SR4)                          | 50        |
| 6.7       | V <sub>PP</sub> status bit (SR3)                  | 50        |
| 6.8       | Program Suspend Status bit (SR2)                  | 51        |
| 6.9       | Block protection status bit (SR1)                 | 51        |
| 6.10      | Bank write/multiple word program status bit (SR0) | 51        |
| <b>7</b>  | <b>Configuration Register</b>                     | <b>53</b> |
| 7.1       | Read select bit (CR15)                            | 53        |
| 7.2       | X latency bits (CR14-CR11)                        | 53        |
| 7.3       | Wait polarity bit (CR10)                          | 54        |
| 7.4       | Wait configuration bit (CR8)                      | 54        |
| 7.5       | Burst length bits (CR2-CR0)                       | 54        |
| <b>8</b>  | <b>Enhanced Configuration Register</b>            | <b>58</b> |
| 8.1       | Deep power-down mode bit (ECR15)                  | 58        |
| 8.2       | Deep power-down polarity bit (ECR14)              | 58        |
| 8.3       | Output driver control bits (ECR2-ECR0)            | 58        |
| <b>9</b>  | <b>Extended Flash array (EFA)</b>                 | <b>60</b> |
| <b>10</b> | <b>Read modes</b>                                 | <b>61</b> |
| 10.1      | Asynchronous read mode                            | 61        |
| 10.2      | Synchronous burst read mode                       | 62        |
| 10.3      | Single synchronous read mode                      | 62        |

|                   |   |            |
|-------------------|---|------------|
| <b>11</b>         | <b>Dual operations and multiple bank architecture</b> | <b>63</b>  |
| <b>12</b>         | <b>Block locking</b>                                  | <b>65</b>  |
| 12.1              | Reading a block's lock status                         | 65         |
| 12.2              | Locked state  | 65         |
| 12.3              | Unlocked state  | 65         |
| 12.4              | Lock-down state                                       | 66         |
| 12.5              | Locking operations during erase suspend               | 66         |
| <b>13</b>         | <b>Program and erase times and endurance cycles</b>   | <b>68</b>  |
| <b>14</b>         | <b>Maximum ratings</b>                                | <b>70</b>  |
| <b>15</b>         | <b>DC and AC parameters</b>                           | <b>71</b>  |
| <b>16</b>         | <b>Package mechanical</b>                             | <b>87</b>  |
| <b>17</b>         | <b>Part numbering</b>                                 | <b>90</b>  |
| <b>Appendix A</b> | <b>Block address tables</b>                           | <b>91</b>  |
| <b>Appendix B</b> | <b>Common Flash interface</b>                         | <b>97</b>  |
| <b>Appendix C</b> | <b>Flowcharts and pseudocodes</b>                     | <b>107</b> |
| <b>Appendix D</b> | <b>Command interface state tables</b>                 | <b>116</b> |
| <b>18</b>         | <b>Revision history</b>                               | <b>123</b> |

## List of tables

|           |   |     |
|-----------|---|-----|
| Table 1.  | Signal names . . . . .  | 11  |
| Table 2.  | M58PR256LE bank architecture . . . . .  | 14  |
| Table 3.  | M58PR512LE bank architecture . . . . .  | 14  |
| Table 4.  | M58PR001LE bank architecture . . . . .  | 14  |
| Table 5.  | EFA memory map . . . . .  | 15  |
| Table 6.  | Bus operations . . . . .  | 22  |
| Table 7.  | Command codes . . . . .   | 23  |
| Table 8.  | Standard commands . . . . .   | 40  |
| Table 9.  | Factory Program command . . . . .   | 41  |
| Table 10. | Electronic signature codes . . . . .  | 41  |
| Table 11. | Protection Register locks . . . . .   | 42  |
| Table 12. | Program methods available with each program mode . . . . .  | 47  |
| Table 13. | Relationships between program methods and program modes . . . . .   | 47  |
| Table 14. | Status Register bits . . . . .  | 52  |
| Table 15. | X latency settings . . . . .  | 54  |
| Table 16. | Configuration Register . . . . .  | 55  |
| Table 17. | Burst type definition . . . . .   | 56  |
| Table 18. | Enhanced Configuration Register . . . . .   | 59  |
| Table 19. | Dual operations allowed in other banks . . . . .  | 63  |
| Table 20. | Dual operations allowed in same bank . . . . .  | 64  |
| Table 21. | Dual operation limitations . . . . .  | 64  |
| Table 22. | Lock status . . . . .   | 67  |
| Table 23. | Program/erase times and endurance cycles . . . . .  | 68  |
| Table 24. | Absolute maximum ratings . . . . .  | 70  |
| Table 25. | Operating and AC measurement conditions . . . . .   | 71  |
| Table 26. | Capacitance . . . . .   | 72  |
| Table 27. | DC characteristics - currents . . . . .   | 73  |
| Table 28. | DC characteristics - voltages . . . . .   | 74  |
| Table 29. | Asynchronous read AC characteristics . . . . .  | 77  |
| Table 30. | Synchronous read AC characteristics . . . . .   | 80  |
| Table 31. | Write AC characteristics, write enable controlled . . . . .   | 82  |
| Table 32. | Write AC characteristics, Chip Enable controlled . . . . .  | 84  |
| Table 33. | Reset and power-up AC characteristics . . . . .   | 85  |
| Table 34. | Deep power-down AC characteristics . . . . .  | 86  |
| Table 35. | TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, mechanical data . . . . .                    | 88  |
| Table 36. | Stacked TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch,<br>package mechanical data . . . . . | 89  |
| Table 37. | Ordering information scheme . . . . .   | 90  |
| Table 38. | M58PR256LE - bank base addresses . . . . .  | 92  |
| Table 39. | M58PR512LE - bank base addresses . . . . .  | 92  |
| Table 40. | M58PR001LE - bank base addresses . . . . .  | 92  |
| Table 41. | M58PR256LE - block addresses . . . . .  | 93  |
| Table 42. | M58PR512LE - block addresses . . . . .  | 94  |
| Table 43. | M58PR001LE - block addresses . . . . .  | 95  |
| Table 44. | Query structure overview . . . . .  | 97  |
| Table 45. | CFI query identification string . . . . .   | 98  |
| Table 46. | CFI query system interface information . . . . .  | 99  |
| Table 47. | Device geometry definition . . . . .  | 100 |

|           |  |     |
|-----------|--|-----|
| Table 48. | Primary algorithm-specific extended query table . . . . .                | 101 |
| Table 49. | Protection Register information . . . . .                                | 102 |
| Table 50. | Burst Read information . . . . .   | 102 |
| Table 51. | Bank and erase block region information . . . . .                        | 103 |
| Table 52. | Bank and erase block region 1 information . . . . .                      | 103 |
| Table 53. | Extended Flash array bank and erase block region information . . . . .   | 104 |
| Table 54. | Extended Flash array bank and erase block region 1 information . . . . . | 105 |
| Table 55. | Command interface states - modify table, next state 1 . . . . .          | 116 |
| Table 56. | Command interface states - modify table, next state 2 . . . . .          | 118 |
| Table 57. | Command interface states - modify table, next output 1 . . . . .         | 120 |
| Table 58. | Command interface states - modify table, next output 2 . . . . .         | 121 |
| Table 59. | Document revision history . . . . .                                      | 123 |

## List of figures

|            |   |     |
|------------|---|-----|
| Figure 1.  | Logic diagram . . . . .   | 10  |
| Figure 2.  | TFBGA105 connections (top view through package) . . . . .                             | 12  |
| Figure 3.  | TFBGA107 connections (top view through package) . . . . .                             | 13  |
| Figure 4.  | Memory map . . . . .  | 15  |
| Figure 5.  | Main array architecture . . . . .   | 16  |
| Figure 6.  | Protection Register memory map . . . . .  | 42  |
| Figure 7.  | Program regions configured in control or object program mode. . . . .                 | 45  |
| Figure 8.  | X latency and data output configuration example. . . . .                              | 56  |
| Figure 9.  | Wait configuration example . . . . .  | 57  |
| Figure 10. | AC measurement I/O waveform . . . . .   | 71  |
| Figure 11. | AC measurement load circuit . . . . .   | 72  |
| Figure 12. | Asynchronous random access read AC waveforms . . . . .                                | 75  |
| Figure 13. | Asynchronous page read AC waveforms . . . . .   | 76  |
| Figure 14. | Synchronous burst read AC waveforms . . . . .   | 78  |
| Figure 15. | Single synchronous read AC waveforms . . . . .  | 79  |
| Figure 16. | Clock input AC waveform . . . . .   | 79  |
| Figure 17. | Write AC waveforms, write enable controlled . . . . .                                 | 81  |
| Figure 18. | Write AC waveforms, Chip Enable controlled . . . . .                                  | 83  |
| Figure 19. | Reset and power-up AC waveforms . . . . .   | 85  |
| Figure 20. | Deep power-down AC waveforms . . . . .  | 86  |
| Figure 21. | Reset during deep power-down AC waveforms . . . . .                                   | 86  |
| Figure 22. | TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline. . . . . | 87  |
| Figure 23. | TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline. . . . . | 88  |
| Figure 24. | Program and EFA block program flowchart and pseudocode . . . . .                      | 107 |
| Figure 25. | Buffer program flowchart and pseudocode . . . . .                                     | 108 |
| Figure 26. | Program suspend and resume flowchart and pseudocode . . . . .                         | 109 |
| Figure 27. | Block erase and EFA block erase flowchart and pseudocode . . . . .                    | 110 |
| Figure 28. | Erase suspend and resume flowchart and pseudocode . . . . .                           | 111 |
| Figure 29. | Main array and EFA locking operations flowchart and pseudocode. . . . .               | 112 |
| Figure 30. | Blank check flowchart and pseudocode . . . . .  | 113 |
| Figure 31. | Protection Register program flowchart and pseudocode . . . . .                        | 114 |
| Figure 32. | Buffer enhanced factory program flowchart and pseudocode . . . . .                    | 115 |

# 1 Description

The M58PR256LE, M58PR512LE and M58PR001LE are 256 Mbit (16 Mbit x 16), 512 Mbit (32 Mbit x 16) and 1 Gbit (64 Mbit x 16) non-volatile Flash memories. They are collectively referred to as the M58PRxxxLE in the rest of the document, unless otherwise specified.

The M58PRxxxLE may be erased electrically at block level and programmed in-system on a word-by-word basis using a 1.7 V to 2.0 V  $V_{DD}$  supply for the circuitry and a 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 speed up factory programming.

The M58PRxxxLE has a uniform block architecture and is based on a multilevel cell technology:

- The M58PR256LE has an array of 128 blocks, and is divided into 32 Mbit banks. There are 8 banks each containing 16 blocks of 128 KWords.
- The M58PR512LE has an array of 256 blocks, and is divided into 64 Mbit banks. There are 8 banks each containing 32 blocks of 128 KWords.
- The M58PR001LE has an array of 512 blocks, and is divided into 128 Mbit banks. There are 8 banks each containing 64 blocks of 128 KWords.

Each block contains 256 program regions of 1 Kbyte each, that are divided into 32 segments of 16 words. Each segment is split into two halves (A and B), according by the value on address input A3.

The memory map is illustrated in [Figure 4](#) and the main array architecture in [Figure 5](#).

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 maps are shown in [Figure 4](#).

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 takes care of 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 108 MHz.

The device features an Automatic standby mode and deep power-down mode. When the bus is inactive during asynchronous read operations, the device automatically switches to automatic standby mode. In this state the power consumption is reduced to the standby value and the outputs are still driven.

The DPD (deep power-down) mode starts when the device is properly configured (ECR bit 15 is set) and the DPD signal is asserted. In DPD mode the device has the lowest power consumption.

The M58PRxxxLE 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.

In addition to the main memory array, the M58PRxxxLE features an extended flash array (EFA) divided into 4 blocks of 64 Kbits each.

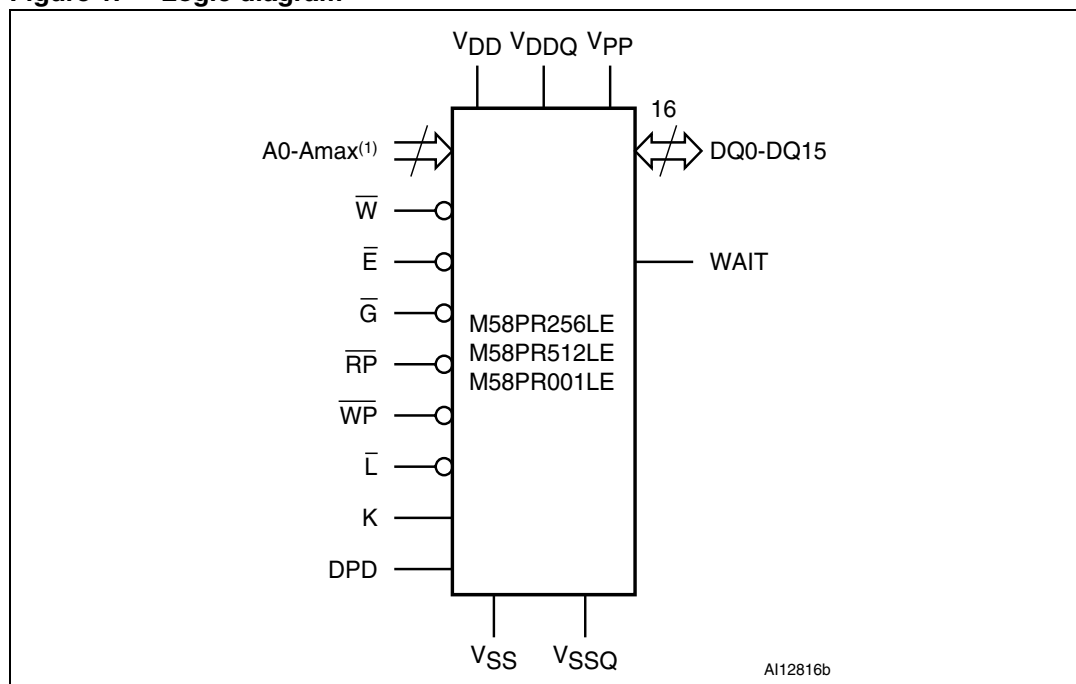
The EFA blocks are accessed through a separate set of commands. The operations available in the EFA blocks are asynchronous read (in non-page mode), single word program, erase and block locking. See [Section 4: Command interface](#) for details of the EFA commands set.

See [Table 5](#) for an extended flash array memory map. [Table 19](#) and [Table 20](#) describe the simultaneous operations allowed in the EFA blocks and the main memory array.

The device includes 17 protection registers and 2 protection register locks, one for the first protection register and the other for the 16 OTP (one-time-programmable) protection registers of 128 bits each. The first protection register is divided into two areas: a 64-bit area containing a unique device number written by ST, and a 64-bit area one-time-programmable by the user. The user programmable area can be permanently protected. [Figure 6](#), shows the Protection Register memory map.

The memory is available in TFBGA105 or TFBGA107 packages, and is supplied with all the bits erased (set to '1').

**Figure 1. Logic diagram**



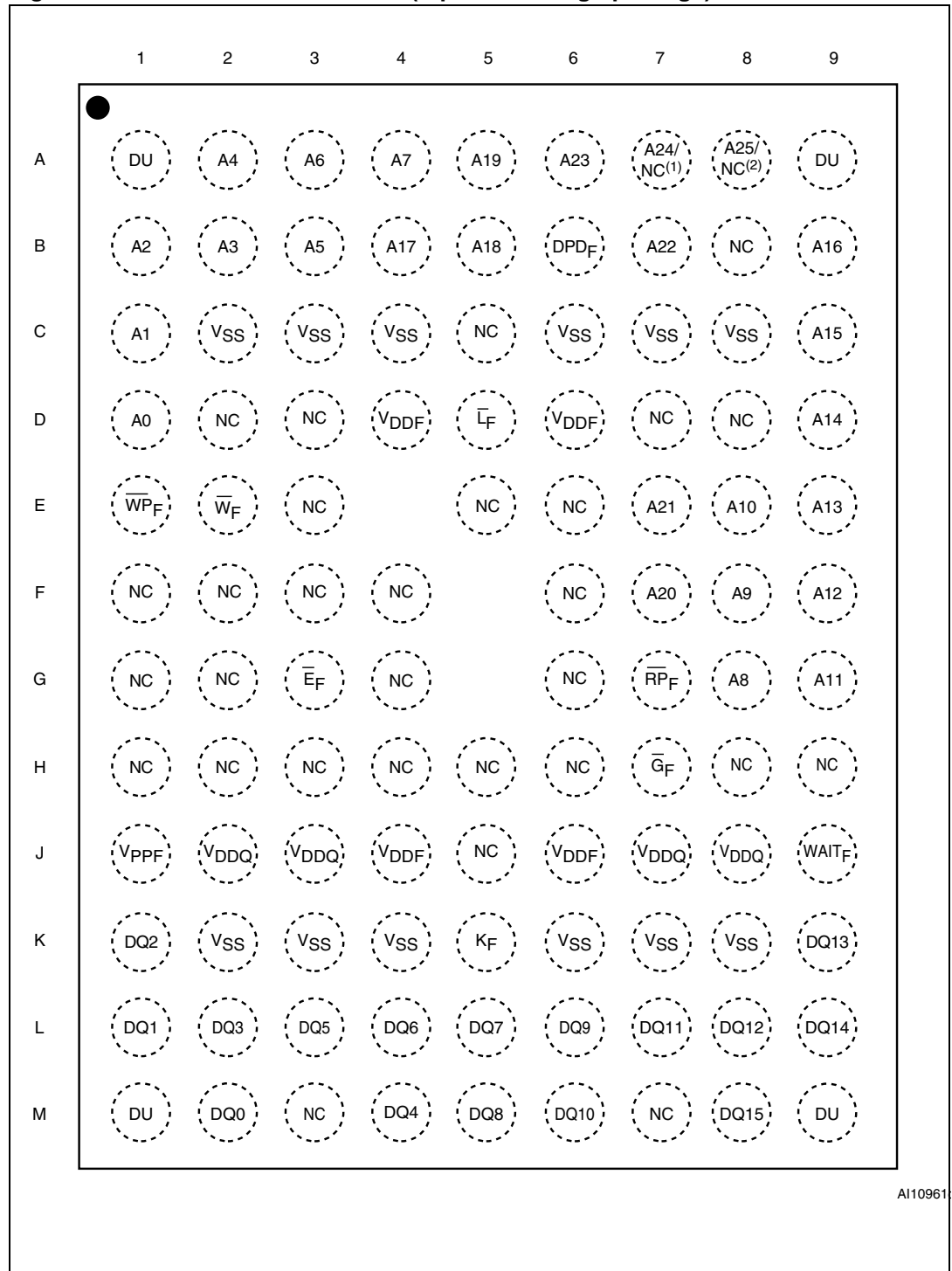
1. Amax is equal to A23 in the M58PR256LE, to A24 in the M58PR512LE, and to A25 in the M58PR001LE.

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    |
| DPD                    | Deep power-down                                    | Input     |
| 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 and erase |           |
| V <sub>SS</sub>        | Ground   |           |
| V <sub>SSQ</sub>       | Ground input/output supply                         |           |
| DU                     | Do not use   |           |
| NC                     | Not connected                                      |           |

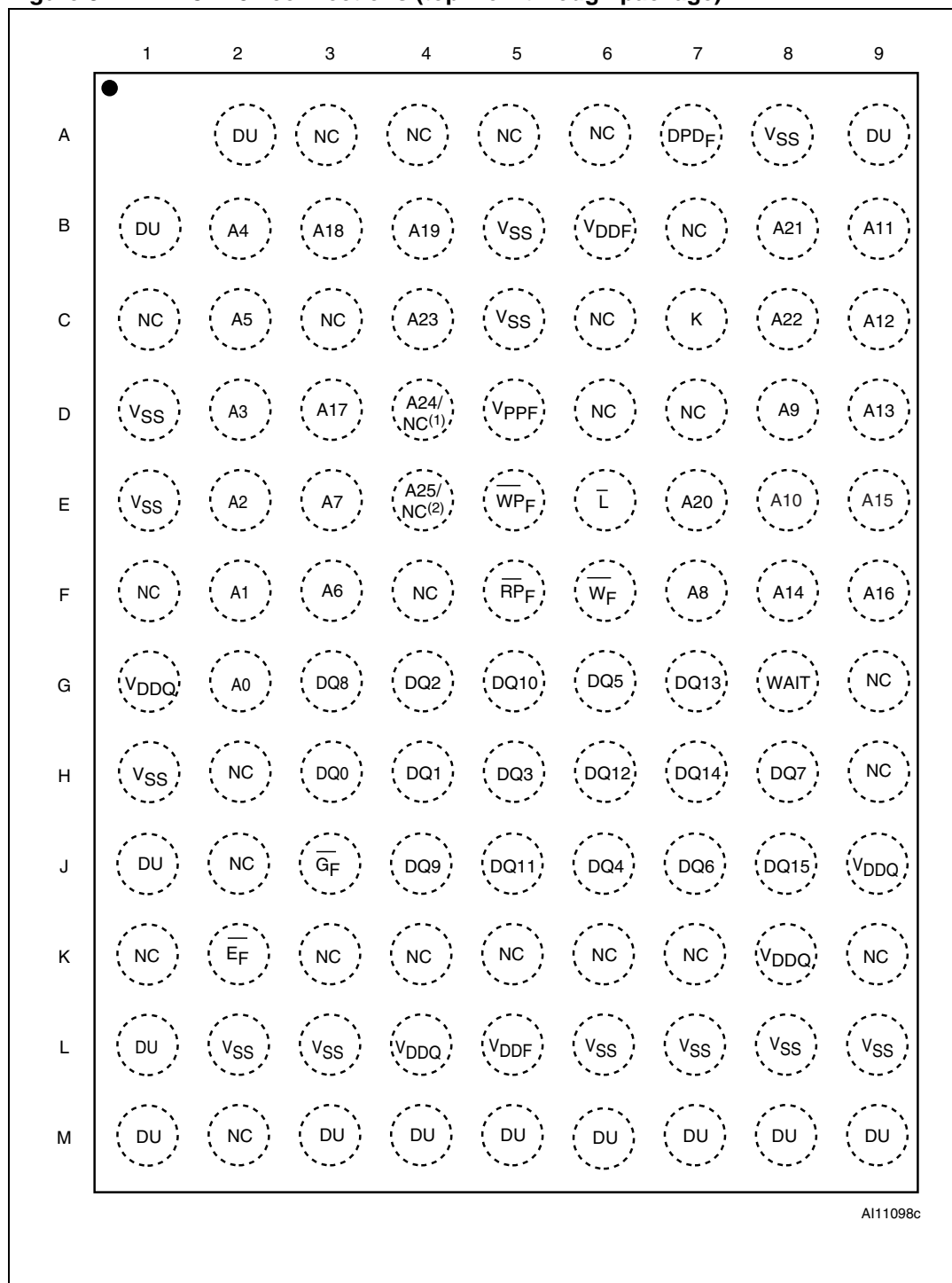
1. Amax is equal to A23 in the M58PR256LE, to A24 in the M58PR512LE, and to A25 in the M58PR001LE.

Figure 2. TFBGA105 connections (top view through package)



1. Ball A7 is A24 in the M58PR512LE and M58PR0001LE, and is Not Connected internally (NC) in the M58PR256LE.
2. Ball A8 is A25 in the M58PR0001LE, and it is Not Connected internally (NC) in the M58PR512LE and M58PR256LE.

Figure 3. TFBGA107 connections (top view through package)



1. Ball D4 is A24 in the M58PR512LE and M58PR0001LE, and is Not Connected internally (NC) in the M58PR256LE.
2. Ball E4 is A25 in the M58PR0001LE, and is Not Connected internally (NC) in the M58PR512LE and M58PR256LE.

**Table 2. M58PR256LE bank architecture**

| Number | Bank size | Blocks                  |
|--------|-----------|-------------------------|
| Bank 0 | 32 Mbits  | 16 blocks of 128 KWords |
| Bank 1 | 32 Mbits  | 16 blocks of 128 KWords |
| Bank 2 | 32 Mbits  | 16 blocks of 128 KWords |
| ⋮      | ⋮         | ⋮                       |
| Bank 7 | 32 Mbits  | 16 blocks of 128 KWords |

**Table 3. M58PR512LE bank architecture**

| Number | Bank size | Blocks                  |
|--------|-----------|-------------------------|
| Bank 0 | 64 Mbits  | 32 blocks of 128 KWords |
| Bank 1 | 64 Mbits  | 32 blocks of 128 KWords |
| Bank 2 | 64 Mbits  | 32 blocks of 128 KWords |
| ⋮      | ⋮         | ⋮                       |
| Bank 7 | 64 Mbits  | 32 blocks of 128 KWords |

**Table 4. M58PR001LE bank architecture**

| Number | Bank size | Blocks                  |
|--------|-----------|-------------------------|
| Bank 0 | 128 Mbits | 64 blocks of 128 KWords |
| Bank 1 | 128 Mbits | 64 blocks of 128 KWords |
| Bank 2 | 128 Mbits | 64 blocks of 128 KWords |
| ⋮      | ⋮         | ⋮                       |
| Bank 7 | 128 Mbits | 64 blocks of 128 KWords |

Figure 4. Memory map

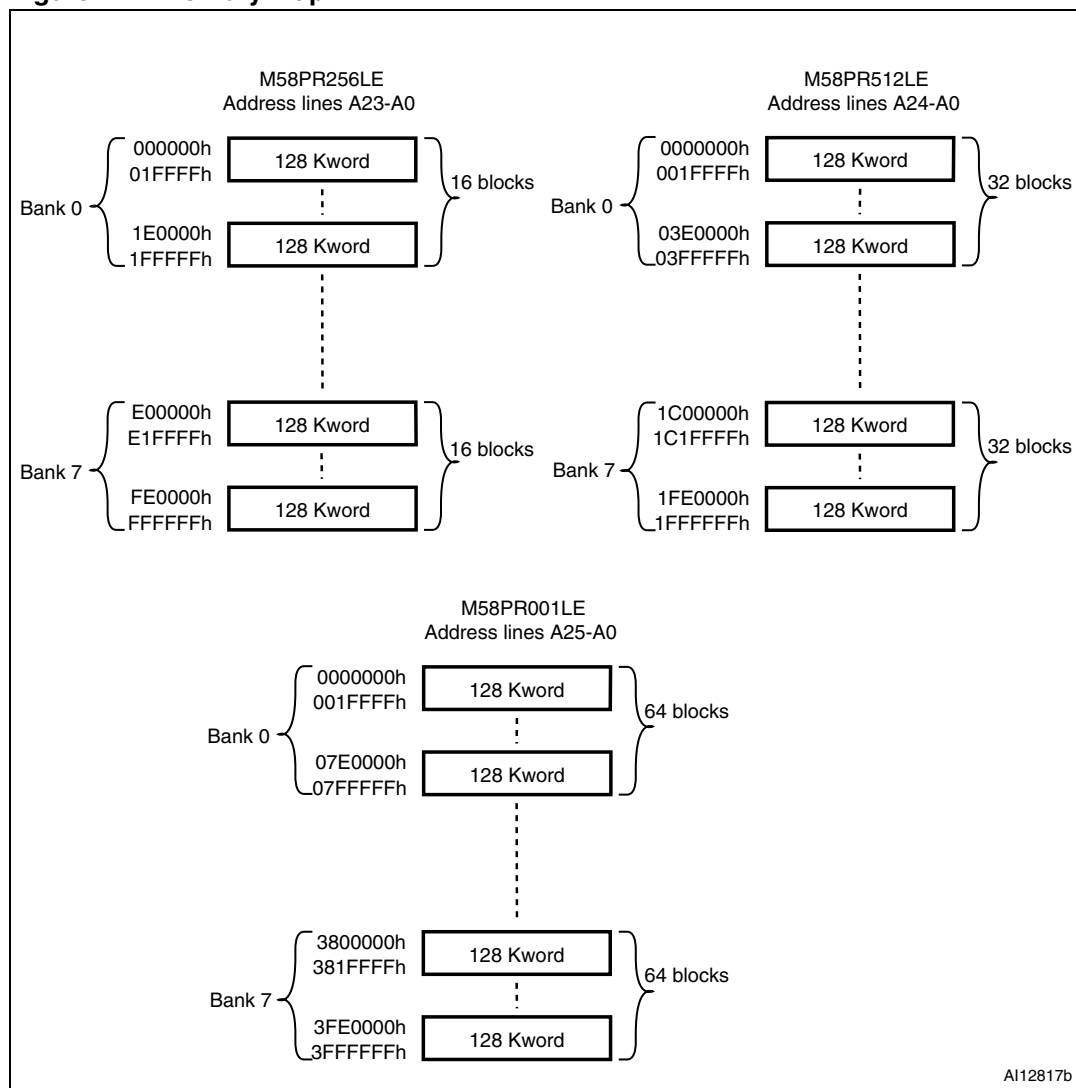
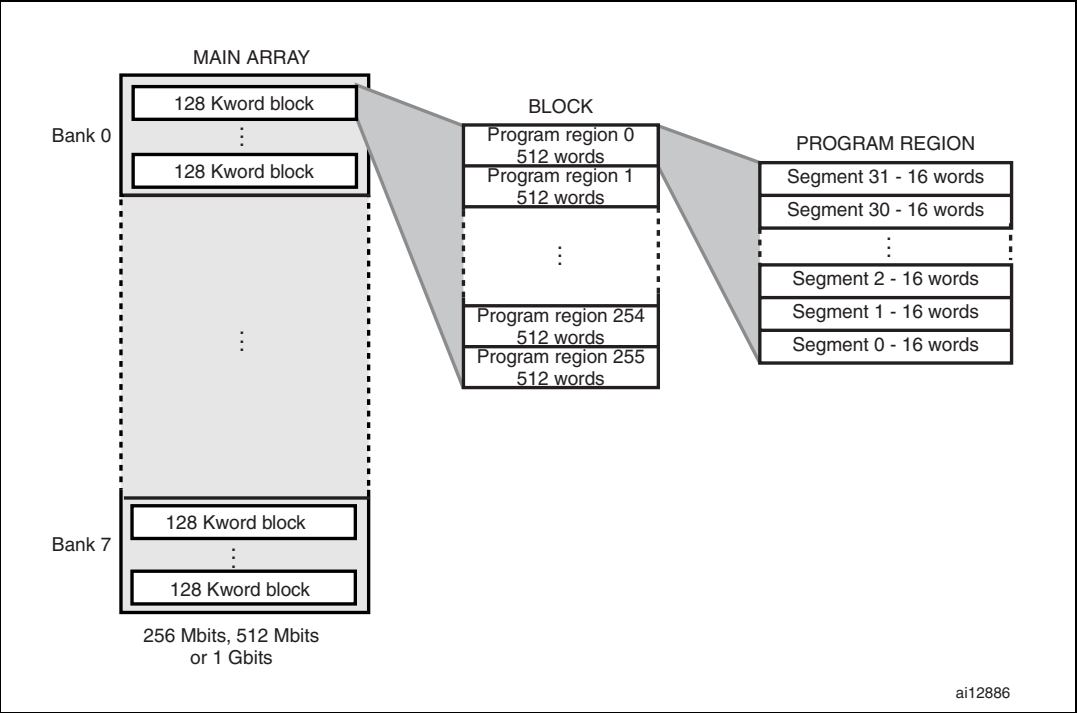


Table 5. EFA memory map

| EFA block | Size                | Address range     |
|-----------|---------------------|-------------------|
| 3         | 4 KWords (64 Kbits) | 0003000 - 0003FFF |
| 2         | 4 KWords (64 Kbits) | 0002000 - 0002FFF |
| 1         | 4 KWords (64 Kbits) | 0001000 - 0001FFF |
| 0         | 4 KWords (64 Kbits) | 0000000 - 0000FFF |

Figure 5. Main array architecture



## 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. Amax is A23 in the M58PR256LE, A24 in the M58PR512LE, and A25 in the M58PR001LE. 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 22: 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, this means the outputs are high impedance and the current consumption is reduced to the reset supply current  $I_{\text{DD2}}$ . Refer to [Table 27: 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, but a negative transition of Chip Enable or Latch Enable is required to ensure valid data outputs.

The Reset pin can be interfaced with 3 V logic without any additional circuitry, and can be tied to  $V_{\text{RPH}}$  (refer to [Table 28: DC characteristics - voltages](#)).

## 2.8 Deep power-down (DPD)

The deep power-down input is used to put the device in deep power-down mode.

When the device is in Standby mode and the Enhanced Configuration Register bit ECR15 is set, asserting the deep power-down input will cause the memory to enter the deep power-down mode.

When the device is in the deep power-down mode, the memory cannot be modified and the data is protected.

The polarity of the DPD pin is determined by ECR14. The deep power-down input is active Low by default.

## 2.9 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  $V_{\text{IH}}$ . Latch Enable can be kept Low (also at board level) when the Latch Enable function is not required or supported.

## 2.10 Clock (K)

The clock input synchronizes the memory to the microcontroller during synchronous read operations; the address is latched on a Clock edge when Latch Enable is at  $V_{\text{IL}}$ . Clock is ignored during asynchronous read and in write operations.

## 2.11 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 data cycle in advance.

## 2.12 $V_{DD}$ supply voltage

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

## 2.13 $V_{DDQ}$ supply voltage

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

$V_{DDQ}$  is sampled at the beginning of program/erase operations. If  $V_{DDQ}$  is lower than  $V_{LKOQ}$ , the device is reset.

## 2.14 $V_{PP}$ 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 (0 V to  $V_{DDQ}$ )  $V_{PP}$  is seen as a control input. In this case a voltage lower than  $V_{PPLK}$  gives an absolute protection against program or erase, while  $V_{PP} > V_{PP1}$  enables these functions (see Tables 27 and 28, 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.

## 2.15 $V_{SS}$ ground

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

## 2.16 $V_{SSQ}$ 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 11: 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 seven standard bus operations that control the device. These are Bus Read, Bus Write, Address Latch, Output Disable, Standby, Reset and deep power-down. See [Table 6: 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  $V_{IL}$  in order to perform a read operation. The Chip Enable input should be used to enable the device. Output Enable should be used to gate data onto the output. The data read depends on the previous command written to the memory (see [Section 4: Command interface](#)). See figures [12](#), [13](#), [14](#) and [15](#), Read AC waveforms, and tables [29](#) and [30](#), 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 can also be latched prior to the Write operation by toggling Latch Enable. In this case the Latch Enable should be tied to  $V_{IH}$  during the Bus Write operation.

See figures [17](#) and [18](#), Write AC waveforms, and tables [31](#) and [32](#), 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.

### 3.7 Deep power-down (DPD)

The memory enters the deep power-down mode from the Standby mode ( $\overline{RP}$  and  $\overline{E}$  are de-asserted,  $V_{IH}$ ) by setting ECR15 High (set to '1') and asserting the DPD pin (these two events can be done in any order).

The DPD pin polarity is determined by the value of ECR14 when:

- ECR14 is cleared ('0') the DPD pin is active Low. The DPD pin is active Low by default.
- ECR14 is set ('1'), the DPD pin is active High.

While in DPD mode, the:

- Values of the Configuration Register, Enhanced Configuration Register, block lock bits, and bank states are preserved.
- Status Register is reset to 80h.

If the Status Register contains errors before entering the DPD mode, the error bits are lost after exiting DPD mode.

The device should not be put in deep power-down mode while a program, erase or suspend operation is in progress, otherwise the operation aborts, and the memory contents are no longer valid.

The deep power-down mode is exited  $t_{DPHEL}$  after de-asserting the DPD pin. Upon exiting the deep power-down mode, the memory reverts to standby mode.

If the  $\overline{RP}$  pin is asserted while in DPD mode, the device exits DPD mode after  $t_{PHEL}$  and ECR15 is reset to 0.

**Table 6. Bus operations<sup>(1)</sup>**

| Operation       | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{L}$ | $\overline{RP}$ | DPD <sup>(2)</sup>         | WAIT <sup>(3)</sup> | DQ15-DQ0                           |
|-----------------|----------------|----------------|----------------|----------------|-----------------|----------------------------|---------------------|------------------------------------|
| Bus read        | $V_{IL}$       | $V_{IL}$       | $V_{IH}$       | $V_{IL}^{(4)}$ | $V_{IH}$        | De-asserted <sup>(5)</sup> |                     | Data output                        |
| Bus write       | $V_{IL}$       | $V_{IH}$       | $V_{IL}$       | $V_{IL}^{(4)}$ | $V_{IH}$        | De-asserted <sup>(5)</sup> |                     | Data input                         |
| Address latch   | $V_{IL}$       | X              | $V_{IH}$       | $V_{IL}$       | $V_{IH}$        | De-asserted <sup>(5)</sup> |                     | Data output or Hi-Z <sup>(6)</sup> |
| Output disable  | $V_{IL}$       | $V_{IH}$       | $V_{IH}$       | X              | $V_{IH}$        | De-asserted <sup>(5)</sup> | Hi-Z                | Hi-Z                               |
| Standby         | $V_{IH}$       | X              | X              | X              | $V_{IH}$        | De-asserted <sup>(5)</sup> | Hi-Z                | Hi-Z                               |
| Reset           | X              | X              | X              | X              | $V_{IL}$        | De-asserted <sup>(5)</sup> | Hi-Z                | Hi-Z                               |
| Deep power-down | $V_{IH}$       | X              | X              | X              | $V_{IH}$        | Asserted <sup>(7)</sup>    | Hi-Z                | Hi-Z                               |

1. X = Don't care.
2. The DPD signal polarity depends on the value of the ECR14 bit.
3. WAIT signal polarity is configured using the Set Configuration Register command.
4.  $\overline{L}$  can be tied to  $V_{IH}$  if the valid address has been previously latched.
5. If ECR15 is set to '0', the device cannot enter the deep power-down mode, even if DPD is asserted.
6. Depends on  $\overline{G}$ .
7. ECR15 has to be set to '1' for the device to enter deep power-down.

## 4 Command interface

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

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 precisely. Any invalid combination of commands are ignored.

Refer to [Table 7: Command codes](#), [Table 8: Standard commands](#), [Table 9: Factory Program command](#) for a summary of the command interface.

**Table 7. Command codes**

| Hex code | Command  |
|----------|--|
| 01h      | Block Lock Confirm and EFA Block Lock Confirm  |
| 03h      | Set Configuration Register Confirm   |
| 04h      | Set Enhanced Configuration Register Confirm  |
| 20h      | Block Erase Setup  |
| 24h      | EFA Block Erase Setup  |
| 2Fh      | Block Lock-down Confirm and EFA Block Lock-down Confirm  |
| 41h      | Program Setup  |
| 44h      | EFA Program Setup  |
| 50h      | Clear Status Register  |
| 60h      | Block Lock Setup, Block Unlock Setup, Block Lock-down Setup, Set Configuration Register Setup and Enhanced Configuration Register Setup  |
| 64h      | EFA Block Lock, EFA Block Lock-down, EFA Block Unlock  |
| 70h      | Read Status Register   |
| 80h      | Buffer Enhanced Factory Program  |
| 90h      | Read Electronic Signature  |
| 94h      | Read EFA   |
| 98h      | Read CFI query   |
| B0h      | Program/Erase Suspend  |
| BCh      | Blank Check Setup  |
| C0h      | Protection Register Program  |
| D0h      | Program/Erase Resume, Block Erase Confirm, Block Unlock Confirm or Buffer Program Confirm, Buffer Enhanced Factory Program Confirm, Blank Check Confirm, Unlock EFA Block Confirm, EFA Block Erase Confirm |
| E9h      | 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 output 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 14](#) 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, the Configuration Register, and the Enhanced 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, the Configuration Register, or the Enhanced Configuration Register (see [Table 10](#)).

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 EFA and the electronic signature locations are not allowed (see [Table 21: 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 continues to program or erase in the background.

The Read Electronic Signature command only changes the read mode of the addressed bank. The read modes of other banks are not affected. Only asynchronous read and single synchronous read operations should be used to read the electronic signature. A Read Array command is required to return the bank to read array mode.

### 4.4 Read CFI Query command

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

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 output the contents of the CFI.

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 EFA and the CFI memory space are not allowed (see [Table 21: Dual operation limitations](#) for details).

## 4.5 Clear Status Register command

The Clear Status Register command can be used to reset (set to '0') all error bits (SR1, SR3, SR4, SR5, SR8 and SR9) 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', and all previous data in the block is lost.

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

Two bus write cycles are required to issue the command.

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

If the second bus cycle is not the block erase confirm code, Status Register bits SR4 and SR5 are set and the command is aborted.

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

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

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

Refer to [Chapter 11](#) for detailed information about simultaneous operations allowed in banks not being erased.

Typical erase times are provided in [Table 23: Program/erase times and endurance cycles](#).

See [Appendix C, Figure 27: Block erase and EFA block erase flowchart and pseudocode](#) for a suggested flowchart for using the Block Erase command.

## 4.7 Program command

The Program command programs a single word to the memory array. It is supported only by program regions configured in control program mode. If a Program command is issued to a program region configured in object program mode, the program operation is aborted and the SR4 and SR8 Status Register bits are set (see [Section 5: Program operations](#)).

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 P/EC (Program/Erase Controller).

The Program command has to be written to the 'A' segment halves (address bit A3 = 0) in the 1 Kbyte program region, whereas the data to be programmed is written to the specific address of the bank to be programmed.

Once the programming has started, read operations in the bank being programmed output the Status Register contents. Programming can be performed in one bank at a time, meanwhile the other banks must be in Read or Erase Suspend mode.

The Status Register P/EC bit, SR7, indicates the progress of the program operation. It should be read to check whether the operation has completed or not.

After completion of the program operation (SR7 = 1), one of the error bits (SR4, SR3 and SR1) going High means that an error was detected. Either a failure occurred during programming,  $V_{PP}$  is outside the allowed voltage range, or an attempt to program a locked block was made. See [Section 6: Status Register](#) for detailed information.

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 command; all other commands are ignored. A Read Array command is required to return the bank to read array mode.

Refer to [Chapter 11](#) for detailed information about simultaneous operations allowed in banks not being programmed.

Typical program times are given in [Table 23: 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 and EFA block program flowchart and pseudocode](#) for the flowchart for using the Program command.

## 4.8 Buffer Program command

The Buffer Program command uses the device's 1 Kbyte write buffer to speed up programming. Up to 1 Kbyte can be loaded into the write buffer and programmed into the specified 1 Kb aligned location in the main array. The Buffer Program command dramatically reduces in-system programming time compared to the standard non-buffered program command.

The Buffer Program command is supported in both object program mode and control program mode.

When using the Buffer Program command in a region configured in object mode, the start programming address must be aligned to the 1 Kb buffer.

When using the Buffer Program command in a region configured in control program mode, the programmed address must be within the 'A' segment halves of the program region (addresses with A3 = 0) and the 'B' segment halves of the program region (addresses with A3 = 1) must be filled only with FFFFh data.

Before issuing the Buffer Program Setup command, the Status Register bit SR7 at the bank address should be read to ensure that the buffer is available (SR7=1).

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.
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. The maximum buffer count is 1FF (512 words).
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. The start address must be aligned to a 1 Kb 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.

The buffer program operation does not change the read status of the banks until the Buffer Program Confirm command is issued. The Buffer Program Confirm command changes the read status of the bank to read Status Register, therefore, after the Buffer Program Confirm command, read operations in the bank output the contents of the Status Register.

Invalid address combinations or failure 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 block being programmed is protected, an error is set in the Status Register and the operation aborts without affecting the data in the memory array.

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

Refer to [Chapter 11](#) for detailed information about simultaneous operations allowed in banks not being programmed.

See [Appendix C, Figure 25: Buffer program flowchart and pseudocode](#) for a suggested flowchart on using the Buffer Program command.

## 4.9 Buffer Enhanced Factory Program command

The Buffer Enhanced Factory Program command has been specially developed to speed up programming in manufacturing environments where programming time is critical.

This command programs one or more write buffer(s) of 1 Kb to an aligned 1 Kb program region. 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 main array block. Only one block can be programmed at a time.

When programming a program region configured in control program mode with the Buffer Enhanced Factory Program command, the 'B' half segment addresses ( $A3 = 1$ ) should not contain '0' values. When writing to a program region configured in object program mode, the B half may contain some '0' values.

If the number of bytes to program is less than 1 Kbyte, the remaining addresses must be filled with FFFFh.

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  $30\text{ }^{\circ}\text{C} \pm 10\text{ }^{\circ}\text{C}$
- The targeted block must be unlocked
- The start address must be aligned with the start of a 1 Kb 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 command programs one block at a time. All data to be programmed must be contained in a single block. If the internal address counter is incremented beyond the highest block address, addressing wraps around to the beginning of the block.

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 9: Factory Program command](#) for detailed information).

### 4.9.1 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 [Chapter 6: Status Register](#) for details on the error.

### 4.9.2 Program and verify phase

The program and verify phase requires 512 cycles to program the 512 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 (512 words). To program less than 512 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 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 512 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 FFFFh to any address outside the block containing the start address to terminate the 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.

### 4.9.3 Exit phase

When Status Register P/EC bit SR7 is set to '1' this indicates that the device has exited the buffer enhanced factory program operation.

Upon exiting the buffered enhanced factory program algorithm by writing FFFFh to an address outside the block containing the start address, the Read mode of the programmed and addressed banks remains unchanged.

A full Status Register check should be done to ensure that the block has been successfully programmed. See [Chapter 6: 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 provided in [Table 23](#).

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

## 4.10 Program/Erase Suspend command

The Program/Erase Suspend command pauses a program or block erase operation. The command can be addressed to any bank and is required to restart a 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
- Read EFA
- Clear Status Register

Additionally, if the suspended operation is 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
- Program EFA

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 EFA, Read Status Register, Read Electronic Signature or Read CFI Query mode, the bank remains in that mode and outputs the corresponding data.

Refer to [Chapter 11](#) 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,  $\overline{RP}$ , goes to  $V_{IL}$ .

See [Appendix C, Figure 26: Program suspend and resume flowchart and pseudocode](#) for flowcharts for using the Program/Erase Suspend command.

## 4.11 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 is complete.

See [Appendix C, Figure 26: Program suspend and resume flowchart and pseudocode](#) and [Figure 28: Erase suspend and resume flowchart and pseudocode](#) for flowcharts for using the Program/Erase Resume command.

## 4.12 Protection Register Program command

The Protection Register Program command programs the user OTP area of the Protection Register and the two Protection Register Locks.

The device features 16 OTP areas of 128 bits and one OTP area of 64 bits, as shown in [Figure 6: Protection Register memory map](#).

The areas are programmed one word at a time. When shipped, all bits in the area are set to '1'. Only the user can 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 EFA and the Protection Register memory space are not allowed (see [Table 21: Dual operation limitations](#) for details).

The two Protection Register locks are used to protect the OTP areas from further modification. The protection of the OTP areas is not reversible. Refer to [Figure 6: Protection Register memory map](#) 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.

## 4.13 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, such as A0 = CR0, A1 = CR1, ..., A15 = CR15. Addresses A16-Amax 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.

## 4.14 Block Lock command

The Block Lock command is used to lock 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 22](#) 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 12: Block locking](#) for a detailed explanation. See [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Lock command.

## 4.15 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 22](#) shows the protection status after issuing a Block Unlock command.

Refer to [Section 12: Block locking](#) for a detailed explanation and [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Block Unlock command.

## 4.16 Block Lock-down command

The Block Lock-down command locks 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 22](#) shows the Lock Status after issuing a Block Lock-down command.

Refer to [Section 12: Block locking](#) for a detailed explanation and [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Lock-down command.

## 4.17 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. Two bus cycles are required to issue the Blank Check command:

- The first bus cycle writes the Blank Check command to any address in the block to be checked.
- The second bus cycle writes the Blank Check Confirm command (D0h) 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. SR7 = '1' means that the operation is complete.

The SR5 Status Register bit goes High (SR5 = '1') to indicate that a 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 30: Blank check flowchart and pseudocode](#) for a suggested flowchart for using the Blank Check command.

Typical Blank Check times are provided in [Table 23: Program/erase times and endurance cycles](#).

## 4.18 Set Enhanced Configuration Register command

The Set Enhanced Configuration Register command is used to write a new value to the Enhanced Configuration Register. Two bus write cycles are required to issue the Set Enhanced Configuration Register command.

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

The Enhanced Configuration Register data must be written as an address during the bus write cycle, such as A0 = ECR0, A1 = ECR1, ..., A15 = ECR15.

If the Set Enhanced Configuration Register setup write cycle is not followed by the Set Enhanced Configuration Register Confirm command (04h), Status Register bits SR4 and SR5 are set.

After successfully executing this command, the bank addressed returns to read array state.

## 4.19 Read EFA Block command

The Read EFA Block command places the addressed bank in the read EFA mode, where all addresses in the addressed bank are remapped to EFA block addresses.

When the device is in read EFA mode, the main array blocks in the addressed bank can no longer be accessed until a Read Array command is issued to the bank.

One bus write cycle is required to issue the Read EFA Block command. Once a bank is in read EFA mode, subsequent read operations from any address within the EFA block output the EFA data from the EFA block. See [Table 5: EFA memory map](#) for details.

EFA blocks can be read through asynchronous or single synchronous read operations only. The Asynchronous page read mode cannot be used to read the EFA blocks.

If a Read EFA command is issued in a bank that is programming or erasing, the read mode of the bank changes to Read EFA mode.

## 4.20 Program EFA Block command

The Program EFA Block command programs a single word to an EFA block.

Two bus write cycles are required to issue the Program EFA Block command.

- The first bus cycle sets up the Program EFA Block command.
- The second cycle 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 contents.

Issuing the Program EFA Block command to an address outside the EFA block address range generates a Program Error in the Status Register (SR4=1).

A Read EFA Block command is required to return the bank to Read EFA mode. Refer to [Section 11](#) for detailed information about simultaneous operations allowed in the banks not being programmed. Typical EFA Program times are given in [Table 23: Program/erase times and endurance cycles](#).

The program operation aborts if Reset,  $\overline{RP}$ , is at  $V_{IL}$ . As data integrity cannot be guaranteed when a program EFA block operation is aborted, the word must be reprogrammed. See [Appendix C, Figure 24: Program and EFA block program flowchart and pseudocode](#) for the flowchart for using the Program EFA Block command.

## 4.21 Erase EFA Block command

The Erase EFA Block command erases an EFA block. It sets all the bits within the selected block to '1', and all previous data in the block is lost. If the EFA block is protected, then the erase operation aborts, the data in the EFA block is not changed, and the Status Register outputs the error.

Two bus write cycles are required to issue the command.

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

The first cycle brings the EFA plane to the foreground and latches the address of the EFA block to be erased. Reading from the bank when the EFA plane is in the foreground returns the Status Register.

Once the Erase operation has started, read operations in the bank being erased output the Status Register contents.

If the Erase EFA Block Confirm command code is not issued in the second bus cycle, Status Register bits SR4 and SR5 are set, the command is aborted, and the addressed bank remains in the Read Status Register mode.

Issuing the Erase EFA Block command outside the EFA block address range generates an error in the Status Register (SR5=1).

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

Refer to [Section 11: Dual operations and multiple bank architecture](#) section for detailed information about simultaneous operations allowed with array and non-array blocks. Typical erase times are provided in [Table 23: Program/erase times and endurance cycles](#).

See [Appendix C, Figure 27: Block erase and EFA block erase flowchart and pseudocode](#) for a suggested flowchart for using the Erase EFA Block command.

## 4.22 Suspend EFA Block command

The Suspend EFA Block command pauses a program or erase EFA block operation. The command can be addressed to any bank. The Resume EFA Block command is required to restart the suspended operation.

One bus write cycle is required to issue the Suspend EFA Block 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 Suspend EFA Block:

- Resume EFA Block
- Read Array
- Read EFA Block (data from erase-suspended blocks or program-suspended words is not valid)
- Read Status Register
- Read Electronic Signature
- Read CFI Query.
- Clear Status Register

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

- Set Configuration Register
- Program EFA Block (except in the erase-suspended block)
- Program and Buffer Program in the main array
- Block Lock
- Block Lock-down
- Block Unlock
- EFA Block Lock
- EFA Block Lock-down
- EFA Block Unlock

During Suspend EFA Block the EFA block being erased can be protected by issuing the EFA Block Lock or EFA Block Lock-down commands. When the Resume EFA Block command is issued, the operation is resumed and completes. The suspend EFA block operation can be repeated.

For example, it is possible to suspend an erase EFA block operation, to start a program EFA block operation, to suspend the program operation, and then read EFA locations.

If a Program EFA Block command is issued during a suspend EFA block operation, the erase EFA block operation cannot be resumed until the program operation has completed.

The state of the bank where the command was issued do not change.

Refer to [Section 11](#) for detailed information about simultaneous operations allowed during a suspend EFA block operation.

During a suspend EFA block operation, the device can be placed in standby mode by taking Chip Enable to  $V_{IH}$ . Program/erase is aborted if Reset,  $\overline{RP}$ , is  $V_{IL}$ .

See [Appendix C, Figure 26: Program suspend and resume flowchart and pseudocode](#) and [Figure 28: Erase suspend and resume flowchart and pseudocode](#) for flowcharts for using the Suspend EFA Block command.

## 4.23 Resume EFA Block command

The Resume EFA Block command restarts the program or erase EFA block operation suspended by the Suspend EFA Block command. One bus write cycle is required to issue the command. The command can be issued to any address.

The Resume EFA Block command does not change the read mode of the banks.

If a Program EFA Block command is issued while an erase EFA block operation has been suspended, then the erase operation cannot be resumed until the program operation has completed.

See [Appendix C, Figure 26: Program suspend and resume flowchart and pseudocode](#) and [Figure 28: Erase suspend and resume flowchart and pseudocode](#) for flowcharts for using the Resume EFA Block command.

## 4.24 Lock EFA Block command

The Lock EFA Block command is used to lock an EFA block and prevent program or erase operations from changing the data in it. All EFA blocks are locked after power-up or reset.

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

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

The lock status can be monitored for each EFA block using the [Read Electronic Signature command](#).

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

Program or erase operations to a locked EFA block generates an error in the Status Register (SR1=1).

Refer to [Section 12: Block locking](#) for a detailed explanation. See [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Lock EFA Block command.

## 4.25 Unlock EFA Block command

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

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

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

The lock status can be monitored for each EFA block using the [Read Electronic Signature command](#).

Refer to [Section 12: Block locking](#) for a detailed explanation and to [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Unlock EFA Block command.

## 4.26 Lock-down EFA Block command

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

A locked-down EFA block cannot be programmed or erased. The lock status of a locked-down EFA 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 EFA blocks can be individually unlocked by issuing the Unlock EFA Block command.

Two bus write cycles are required to issue the Lock-down EFA Block command.

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

The lock status can be monitored for each EFA block using the [Read Electronic Signature command](#).

Locked-down EFA blocks revert to the locked (and not locked-down) state when the device is reset on power-down. [Table 22](#) shows the lock status after issuing a Lock-down EFA Block command.

Refer to [Section 12: Block locking](#) for a detailed explanation and to [Appendix C, Figure 29: Main array and EFA locking operations flowchart and pseudocode](#) for a flowchart for using the Lock-down EFA Block command.

Table 8. Standard commands

| Commands                            | Cycles             | Bus operations <sup>(1)</sup> |                          |                   |           |                    |                 |
|-------------------------------------|--------------------|-------------------------------|--------------------------|-------------------|-----------|--------------------|-----------------|
|                                     |                    | 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> | 41h               | Write     | WA                 | PD              |
| Buffer Program                      | n+4 <sup>(4)</sup> | Write                         | BA                       | E9h               | 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             |
| Set Enhanced Configuration Register | 2                  | Write                         | ECRD                     | 60h               | Write     | ECRD               | 04h             |
| 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             |
| Blank Check                         | 2                  | Write                         | BA                       | BCh               | Write     | BA                 | D0h             |
| Read EFA Block                      | 1+                 | Write                         | BKA                      | 94h               | Read      | WA                 | RD              |
| Program EFA Block                   | 2                  | Write                         | BKA or WA <sup>(3)</sup> | 44h               | Write     | WA                 | PD              |
| Erase EFA Block                     | 2                  | Write                         | BKA or BA <sup>(3)</sup> | 24h               | Write     | BA                 | D0h             |
| Suspend EFA Block                   | 1                  | Write                         | X                        | B0h               |           |                    |                 |
| Resume EFA Block                    | 1                  | Write                         | X                        | D0h               |           |                    |                 |
| Lock EFA Block                      | 2                  | Write                         | BKA or BA <sup>(3)</sup> | 64h               | Write     | BA                 | 01h             |
| Unlock EFA Block                    | 2                  | Write                         | BKA or BA <sup>(3)</sup> | 64h               | Write     | BA                 | D0h             |
| Lock-down EFA Block                 | 2                  | Write                         | BA                       | 64h               | 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, PA = Program Address, PD = Program Data, PRA = Protection Register Address, PRD = Protection Register Data, CRD = Configuration Register Data, ECRD = Enhanced Configuration Register Data.

2. Must be same bank as in the first cycle. The signature addresses are listed in [Table 10](#).

3. Any address within the bank can be used.

4. n+1 is the number of words to be programmed.

**Table 9. Factory Program command<sup>(1)</sup>**

| Command                                  | Phase                          | Cycles | Bus write operations                  |                 |                 |                 |                 |                 |                 |                   |                 |                   |
|--|--------------------------------|--------|---------------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-------------------|-----------------|-------------------|
|  |                                |        | 1st                                   |                 | 2nd             |                 | 3rd             |                 | Final -1        |                   | Final           |                   |
|  |                                |        | Add                                   | Data            | Add             | Data            | Add             | Data            | Add             | Data              | Add             | Data              |
| Buffer<br>Enhanced<br>Factory<br>Program | Setup                          | 2      | BKA or<br>WA <sup>(2)</sup>           | 80h             | WA <sub>1</sub> | D0h             |                 |                 |                 |                   |                 |                   |
|  | Program/ Verify <sup>(3)</sup> | ≥512   | 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>511</sub> | WA <sub>1</sub> | PD <sub>512</sub> |
|  | Exit                           | 1      | NOT<br>BA <sub>1</sub> <sup>(4)</sup> | FFFFh           |                 |                 |                 |                 |                 |                   |                 |                   |

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 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 10. Electronic signature codes**

| Code                                      |                             |                          | Address (h)        | Data (h)             |
|---|-----------------------------|--------------------------|--------------------|----------------------|
| Manufacturer code                         |                             |                          | Bank address + 00  | 0020                 |
| Device code                               |                             | 256 Mbit                 | Bank address + 01  | 8818                 |
|   |                             | 512 Mbit                 |                    | 8819                 |
|   |                             | 1 Gbit                   |                    | 880F                 |
| Block protection                          | Main block                  | Locked                   | Block address + 02 | DQ1, DQ0 = 01        |
|   |                             | Unlocked                 |                    | DQ1, DQ0 = 00        |
|   |                             | Locked and Locked-down   |                    | DQ1, DQ0 = 11        |
|   |                             | Unlocked and Locked-down |                    | DQ1, DQ0 = 10        |
|   | EFA block                   | Locked                   |                    | DQ5, DQ4 = 01        |
|   |                             | Unlocked                 |                    | DQ5, DQ4 = 00        |
|   |                             | Locked and Locked-down   |                    | DQ5, DQ4 = 11        |
|   |                             | Unlocked and Locked-down |                    | DQ5, DQ4 = 10        |
| Configuration Register                    |                             |                          | Bank address + 05  | CR <sup>(1)</sup>    |
| Enhanced Configuration Register           |                             |                          | Bank address + 06  | ECR <sup>(1)</sup>   |
| Protection Register PR0 Lock              | ST factory default          |                          | Bank address + 80  | 0002                 |
|   | OTP area permanently locked |                          |                    | 0000                 |
| Protection Register PR0                   |                             |                          | Bank address + 81  | Unique device number |
|   |                             |                          | Bank address + 84  |                      |
|   |                             |                          | Bank address + 85  | OTP area             |
|   |                             |                          | Bank address + 88  |                      |
| Protection Register PR1 through PR16 Lock |                             |                          | Bank address + 89  | PRLD <sup>(1)</sup>  |
| Protection Registers PR1-PR16             |                             |                          | Bank address + 8A  | OTP area             |
|   |                             |                          | Bank address + 109 |                      |

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

Figure 6. Protection Register memory map

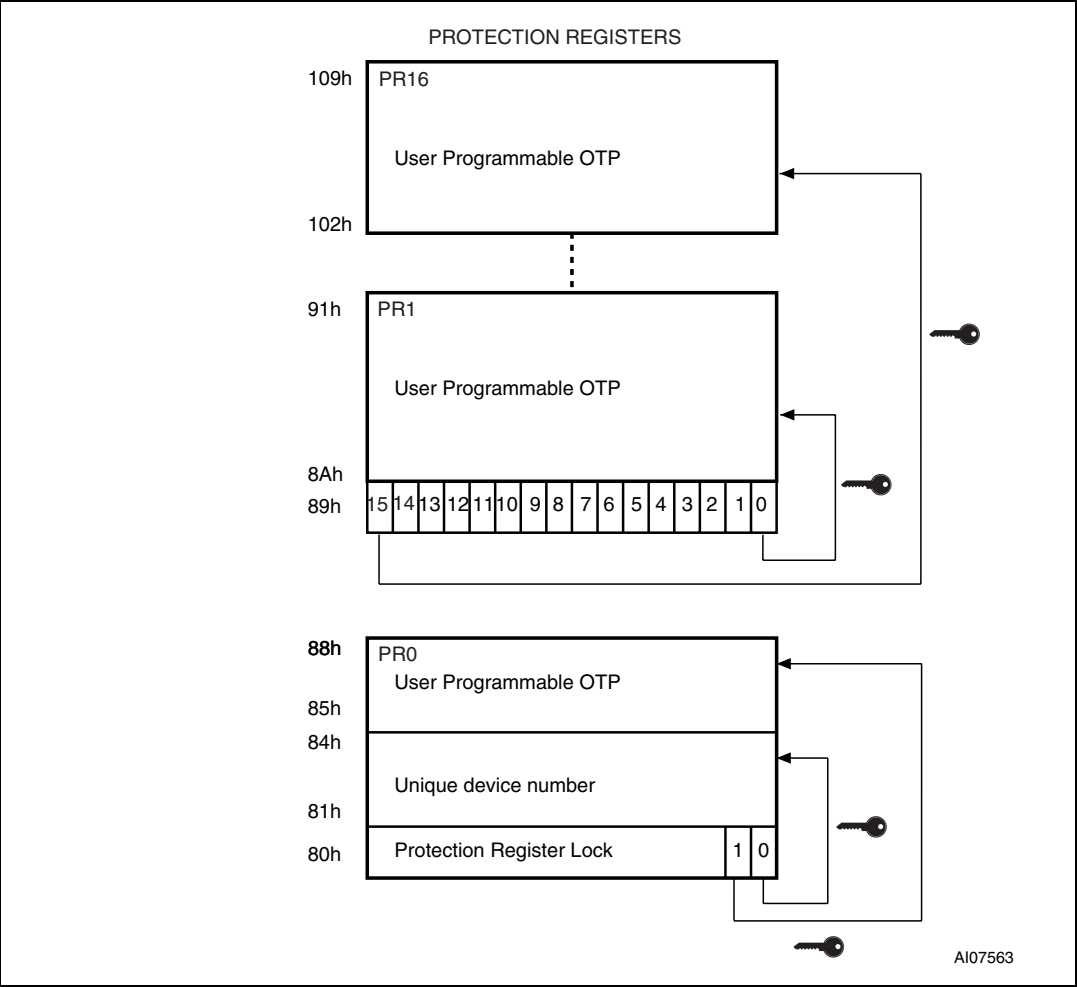


Table 11. Protection Register locks

| 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 area, address 85h to 88h in PR0                  |
|        |         | Bits 2 to 15 | Reserved   |
| Lock 2 | 89h     | Bit 0        | Protects 128 bits of OTP area PR1  |
|        |         | Bit 1        | Protects 128 bits of OTP area PR2  |
|        |         | Bit 2        | Protects 128 bits of OTP area PR3  |
|        |         | ⋮            | ⋮  |
|        |         | Bit 13       | Protects 128 bits of OTP area PR14                                       |
|        |         | Bit 14       | Protects 128 bits of OTP area PR15                                       |
|        |         | Bit 15       | Protects 128 bits of OTP area PR16                                       |

## 5 Program operations

The M58PR256LE, M58PR512LE and M58PR001LE have innovative features specially developed to improve the storage flexibility and efficiency of NOR Flash memory arrays.

Data and code can be stored more efficiently by using the right combination of program methods and program modes.

There are two types of program methods that use commands that consist of one or more sequential bus write operations interpreted by the command interface:

- Single word program method, which uses the Program command.
- Buffered program method, which uses either the Buffer Program command or the Buffer Enhanced Factory Program command.

There are two program modes:

- Control program mode
- Object program mode.

The control program mode supports the two program methods, whereas the object program mode only supports the buffered program method.

This new logical organization of program operations is made possible by the device architecture, and, in particular, by the new concept of program regions.

### 5.1 Program regions

Each Flash memory block is divided into 256 program regions (see [Figure 7: Program regions configured in control or object program mode](#)). Erase operations have a block granularity, whereas program operations have a program region granularity.

The user can configure each program region to be programmed either in the control program mode or in the object program mode. A given block can contain program regions configured in the control program mode and others configured in the object program mode.

Special care should be taken when selecting the programming mode for the program regions because once the program regions are configured, their program mode cannot be changed until the entire block is erased.

Each program region is split into 32 segments of 32 bytes and each segment is subdivided into two halves, 'A' and 'B'. Address bit A3 determines whether a bit belongs to the 'A' half (A3 = 0) or to the 'B' half (A3 = 1).

## 5.2 Program modes

There are two program modes, which allow the Flash memory to store different types of data: control program mode and object program mode.

### 5.2.1 Control program mode

The control program mode is best suited to the storage of small, dynamic information. Typically such data is contained within one program region and it is frequently updated and/or new data is added to it.

Program regions are configured in the control program mode by programming data only to the 'A' halves (bit A3= 0) of the segments they contain. The 'B' halves of the segments must remain erased, meaning that they should not contain any zeros (see [Figure 7: Program regions configured in control or object program mode](#)).

In a program region of 1 Kbyte configured in the control program mode, only 512 bytes of data can be stored.

When the program regions are configured in the control program mode, any program method can be used: the single word or the buffered program methods.

Once a program region has been configured in the control program mode, if a zero is written to a 'B' half of one of its segments, the program operation is terminated and an error is generated. The Status Register bits SR4 and SR9 are set to '1'. (Refer to [Status Register](#) and to [Table 13: Relationships between program methods and program modes](#) for details.)

The program mode of a program region configured in the control program mode can only be changed by first erasing the block that contains the program region.

### 5.2.2 Object program mode

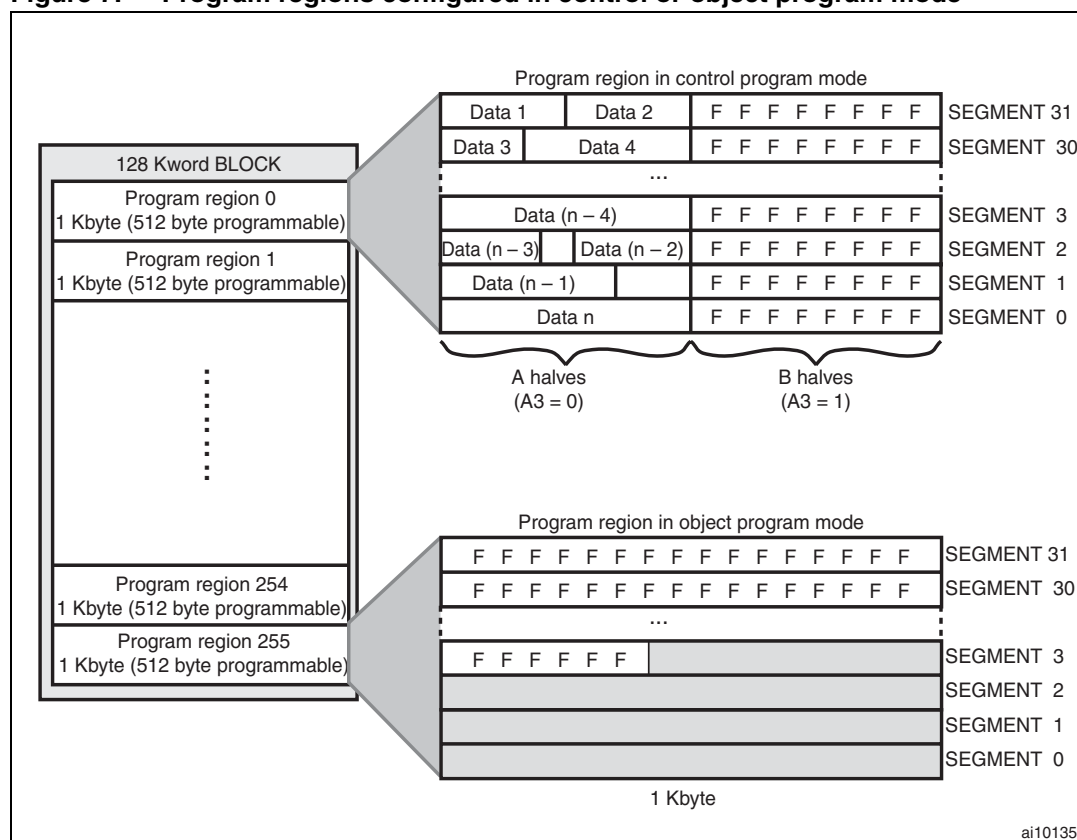
The object program mode is best suited to the storage of large amounts of static information. In a program region of 1 Kbyte configured in the object program mode, 1 Kbyte of data can be stored.

When a program region is configured in the object program mode, it cannot be re-programmed or have new data added without first erasing the entire block that contains the program region.

Program regions are configured in the object program mode simply by programming at least one bit in the 'B' half (A3 = 1) of one of the segments they contain.

If the programmed data is smaller than 1 Kbyte, the unused space remains in the erased state (all the bits set to FFFFh), but can no longer be used to program data. See [Figure 7: Program regions configured in control or object program mode](#).

When the program regions are configured in the object program mode, only the buffered program methods can be used. If an attempt is made to use the single word program method, the program operation is aborted and Status Register Error bits SR4 and SR8 are set to '1'. (Refer to [Status Register](#) and to [Table 13: Relationships between program methods and program modes](#), for details.)

**Figure 7. Program regions configured in control or object program mode**

ai10135

## 5.3 Program methods

The device supports two types of program methods:

- Single word program method, which is used to program a single word to a specific address of the memory array.
- Buffered program methods, which can be split into two different methods:
  - Buffer program method, which uses the device's write buffer to speed up programming. The data is written into the write buffer and then programmed to the specified block address.
  - Buffer enhanced factory program method, which is developed to speed up programming in manufacturing environments where the programming time is critical. The data is written in the write buffer and then programmed to the specified block.

The following sections describe the relationship between program commands and program methods in detail. See [Table 13: Relationships between program methods and program modes](#) and [Table 12: Program methods available with each program mode](#).

### 5.3.1 Single word program method

The single word program method is based on the Program command. It is only supported by program regions configured in the control program mode.

If the single word program method is attempted in a program region configured in the object program mode, the program operation is aborted and Status Register bits SR4 and SR8 are set.

See [Section 4: Command interface](#) for a detailed description of the Program command.

In program regions configured in the control program mode, the Program command can be issued several times.

Using the single word program method to program one or more bits to '0' in the 'B' halves of the segments ( $A3 = 1$ ) of an erased or already programmed program region generates an error:

- In the case of an erased program region, this is considered an illegal operation that sets Status Register bits SR4 and SR9.
- In the case of an already programmed program region, an error is always generated because:
  - To be able to write to a program region configured in the object program mode, the entire block that contains the program region must be erased first.
  - It is not allowed to write to the 'B' halves of the segments of a program region configured in the control program mode.

### 5.3.2 Buffer program method

The buffer program method is based on the Buffer Program command and uses a 1 Kbyte write buffer to speed up programming. The data is written to the write buffer and then programmed to the specified main array location.

The buffer program method is supported regardless of the program mode of the program regions. When using the buffer program method in a program region configured in the object program mode, the start address must be aligned to the 1 Kbyte write buffer.

When using the buffer program method in a program region configured in the control program mode, the address to be programmed must be located inside the 'A' halves of the program region's segments (addresses with  $A3 = 0$ ) and the 'B' halves of the segments (addresses with  $A3 = 1$ ) must be filled only with FFFFh data.

The Buffer Program command can be issued several times to program regions configured in the control program mode.

The Buffer Program command can only be issued once in program regions configured in the object program mode. Attempts to program the same program regions by re-issuing the Buffer Program command leads to data corruption.

See [Section 4: Command interface](#) for a detailed description of the Buffer Program command.

### 5.3.3 Buffer enhanced factory program method

The buffer enhanced factory program method is based on the Buffer Enhanced Factory Program command.

The buffer enhanced factory program method is supported by the program regions, regardless of the program mode in which they are configured.

In this program method, the program region (1 Kbyte) must be completely filled, regardless of the program mode used. If the size of the data to be written is less than 1 Kbyte, the remaining addresses in the program region must be filled with FFFFh.

When using the buffer enhanced factory program method in a program region configured in the control program mode, the addresses to be programmed must be located in the 'A' half of the program regions' segments (A3 = 0) and the 'B' half of the segments (A3 = 1) must be filled only with FFFFh.

See [Section 4: Command interface](#) for a detailed description of the Buffer Enhanced Factory Program command.

**Table 12. Program methods available with each program mode**

| Program mode         | Program methods <sup>(1)</sup> |                  |                                 |
|----------------------|--------------------------------|------------------|---------------------------------|
|                      | Single word program            | Buffered program |                                 |
|                      |                                | Buffer program   | Buffer enhanced factory Program |
| Control program mode | X                              | X                | X                               |
| Object program mode  |                                | X                | X                               |

1. X means available.

**Table 13. Relationships between program methods and program modes**

| Program region status | Address bit A3 value | Program method  |                                 |   |
|-----------------------|----------------------|---|---------------------------------|---|
|                       |                      | Buffered program  |                                 | Single word program   |
|                       |                      | Buffer program  | Buffer enhanced factory program |   |
| Erased                | A3 = 0 ('A' half)    | Program region configured in control program mode                           |                                 | Program region configured in control program mode                           |
|                       | A3 = 1 ('B' half)    | Program region configured in object program mode                            |                                 | Not allowed.<br>Program aborted, Status Register Error bits SR4 and SR9 set |
| Control program mode  | A3 = 0 ('A' half)    | Program operation successful  |                                 |   |
|                       | A3 = 1 ('B' half)    | Not allowed.<br>Program aborted, Status Register Error bits SR4 and SR9 set |                                 |   |
| Object program mode   | A3 = 0 ('A' half)    | Subsequent program not allowed.   |                                 |   |
|                       | A3 = 1 ('B' half)    | Program aborted, Status Register Error bits SR4 and SR8 set                 |                                 |   |

## 6 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](#) for more details on the command itself).

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 SR9, SR8, 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 14: Status Register bits](#). Refer to [Table 14](#) in conjunction with the following text descriptions.

### 6.1 Control program mode status bit (SR9)

The control program mode status bit, SR9, indicates whether an error occurred while writing to a program region that is configured in control program mode.

The SR9 bit should be read once the Program/Erase Controller status bit SR7 is set to '1' (Program/Erase Controller inactive).

SR9 is set to 1 when the user attempts to program object data in a control mode region.

When:

- SR9 = 0, the program operation completed successfully.
- SR9 = 1, the program operation failed.

Once set to '1', SR9 can only be cleared by issuing a Clear Status Register command or through a hardware reset.

SR9 should be cleared before a new program command is issued, otherwise the new command appears to fail.

## 6.2 Object program mode status bit (SR8)

The object program mode status bit, SR8, indicates whether an error occurred while writing to a program region that was configured in the object program mode.

The SR8 bit should be read once the Program/Erase Controller status bit SR7 is set to '1' (Program/Erase Controller inactive).

SR8 is set to 1 when the user attempts to rewrite an object mode region.

When:

- SR8 = 0, the program operation completed successfully.
- SR8 = 1, the program operation failed.

Once set, SR8 can only be cleared by issuing a Clear Status Register command or through a hardware reset.

SR8 should be cleared before a new program command is issued, otherwise the new command appears to fail.

## 6.3 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:

- SR7 = 0, the Program/Erase Controller is active.
- SR7 = 1, the Program/Erase Controller is inactive, and the device is ready to process a new command.

The Program/Erase Controller status bit is set to '0' immediately after a Program/Erase Suspend command is issued until the Program/Erase Controller pauses. After the Program/Erase Controller pauses, the bit is set to '1'.

## 6.4 Erase suspend status bit (SR6)

The erase suspend status bit indicates that an erase operation has been suspended in the addressed block.

When:

- SR6 = 0, no Program/Erase Suspend command has been issued.
- SR6 = 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 set to '1' (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 suspend mode.

When a Program/Erase Resume command is issued, the erase suspend status bit is reset to '0'.

## 6.5 Erase status bit (SR5)

The erase status bit identifies if there is an error during a block erase operation.

When:

- SR5 = 0, no error occurred.
- SR5 = 1, the Program/Erase Controller applied the maximum number of pulses to the block or bank and still failed to verify that it has erased correctly.

The erase status bit should be read once the Program/Erase Controller Status bit is set to '1' (Program/Erase Controller inactive). Once set, the erase status bit must be cleared by a Clear Status Register command or a hardware reset. This must be done before a new erase command is issued, otherwise the new command appears to fail.

## 6.6 Program status bit (SR4)

The program status bit identifies if there is an error during a program operation.

The program status bit should be read once the Program/Erase Controller status bit is set to '1' (Program/Erase Controller inactive).

When:

- SR4 = 0, no error occurred.
- SR4 = 1, the Program/Erase Controller applied the maximum number of pulses to the word and still failed to verify that it programmed correctly.

Attempting to program a '1' to an already programmed bit while  $V_{PP} = V_{PPH}$  also sets the program status bit to '1'. If  $V_{PP}$  is different from  $V_{PPH}$ , SR4 remains set to '0' and the attempt is not shown.

Once set to '1', the program status bit must be cleared by a Clear Status Register command or a hardware reset. This must be done before a new program command is issued, otherwise the new command appears to fail.

## 6.7 $V_{PP}$ 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:

- SR3 = 0, the voltage on the  $V_{PP}$  pin is sampled at a valid voltage.
- SR3 = 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 to '1', the  $V_{PP}$  status bit must be cleared by a Clear Status Register command or a hardware reset. This must be done before a new program or erase command is issued, otherwise the new command appears to fail.

## 6.8 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 is only considered valid when the Program/Erase Controller status bit is set to '1' (Program/Erase Controller inactive).

When:

- SR2 = 0, no Program/Erase Suspend command has been issued.
- SR2 = 1, a Program/Erase Suspend command has been issued and the memory is waiting for a Program/Erase Resume command.

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

When a Program/Erase Resume command is issued, the program suspend status bit is reset to '0'.

## 6.9 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 block.

When:

- SR1 = 0, no program or erase operation has been attempted on a locked block.
- SR1 = 1, a program or erase operation has been attempted on a locked block.

Once set to '1', the block protection status bit must be cleared by a Clear Status Register command or a hardware reset. This must be done before a new program or erase command is issued, otherwise the new command appears to fail.

## 6.10 Bank write/multiple word program status bit (SR0)

The bank write status bit indicates if the addressed bank is programming or erasing.

The bank write status bit is only considered valid when the Program/Erase Controller status bit SR7 is set to '0'.

When:

- SR0 = 0 and SR7 = 0, the addressed bank is executing a program or erase operation.
- SR0 = 1 and SR7 = 0, a program or erase operation is being executed in a bank other than the one being addressed.

During buffer enhanced factory program operations the multiple word program bit, SR0, shows if the device is ready to accept a new word to be programmed to the memory array.

When:

- SR0 = 0, the device is ready for the next word.
- SR0 = 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 14. Status Register bits

| Bit       | Name  | Type   | Logic level <sup>(1)</sup> | Definition  |
|-----------|---|--------|----------------------------|---|
| SR15-SR10 | Reserved <sup>(2)</sup>   |        |                            |   |
| SR9       | Control program mode status   | Error  | 1                          | Program error in program region configured in control program mode                |
|           |   |        | 0                          | Program successful  |
| SR8       | Object program mode status  | Error  | 1                          | Program error in program region configured in object program mode                 |
|           |   |        | 0                          | Program successful  |
| 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 status  | Error  | 1                          | Erase error   |
|           |   |        | 0                          | Erase 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 word or is going to exit BEFP mode |
|           |   |        | 0                          | SR7 = '1' The device is exiting from BEFP   |
|           |   |        |                            | SR7 = '0' The device is ready for the next word                                   |

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

2. Reserved bits should always be reset to '0'.

## 7 Configuration Register

The Configuration Register configures the type of bus access that the memory performs. Refer to [Section 10: Read modes](#) for details on read operations.

The Configuration Register is set through the command interface using the Set Configuration Register command. The Configuration Register is volatile: after a reset or a power-down/power-up sequence, the register is set for asynchronous read (CR15=1) and all bits return to their default value. The Configuration Register bits are described in [Table 16](#). They specify the selection of the burst length, burst X latency and the read operation. Refer to [Figure 8](#) and [Figure 9](#) for examples of synchronous burst configurations.

### 7.1 Read select bit (CR15)

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

When:

- CR15 = 0:
  - Read operations in the main array are performed in synchronous burst mode,
  - Operations to read the Status Register, electronic signature, CFI and EFA are performed in single synchronous mode (See [Section 10.3: Single synchronous read mode](#) for details).
- CR15 = 1:
  - Read operations in the main array are performed in Asynchronous page mode,
  - Operations to read the Status Register, electronic signature, CFI and EFA are performed in asynchronous random access mode.

Synchronous burst read can be performed across banks.

On reset or power-up the read select bit is set to '1' for asynchronous access.

### 7.2 X latency bits (CR14-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.

For correct operation the X latency bits can only assume the values in [Table 16: Configuration Register](#).

[Table 15](#) shows how to set the X latency parameter, taking into account the frequency used to read the Flash memory in synchronous mode.

Refer to [Figure 8: X latency and data output configuration example](#) for an example waveform.

**Table 15. X latency settings**

| $f_{\max}$ | $t_{kmin}$ | X latency |
|------------|------------|-----------|
| 40 MHz     | 25 ns      | 4         |
| 54 MHz     | 19 ns      | 5         |
| 66 MHz     | 15 ns      | 6         |
| 108 MHz    | 9 ns       | 10        |

### 7.3 Wait polarity bit (CR10)

The wait polarity bit sets the polarity of the Wait signal used in synchronous burst read mode.

When:

- CR10 = 0, the Wait signal is active Low.
- CR10 = 1, the Wait signal is active High.

During synchronous burst read mode the Wait signal indicates whether the data output is valid or a Wait state must be inserted.

### 7.4 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:

- CR8 = 0, the Wait output pin is asserted during the wait state.
- CR8 = 1, the Wait output pin is asserted one data cycle before the wait state.

When WAIT is asserted, data is not valid and when WAIT is de-asserted, data is valid.

### 7.5 Burst length bits (CR2-CR0)

The burst length bits set the number of words to be output during a synchronous burst read operation as result of a single address latch cycle.

They can be set for 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, the device asserts the Wait signal to indicate that a delay is necessary before the data is output.

In continuous burst mode, if the starting address is not aligned to the 16-word boundary, Wait is asserted 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.

In the worst case scenario, the number of wait states is one clock cycle less than the latency setting. Wait is asserted only once during a continuous burst access. See also [Table 17: Burst type definition](#).

**CR9, CR7, CR6, CR5, CR4** and **CR3** are reserved for future use.

**Table 16. Configuration Register**

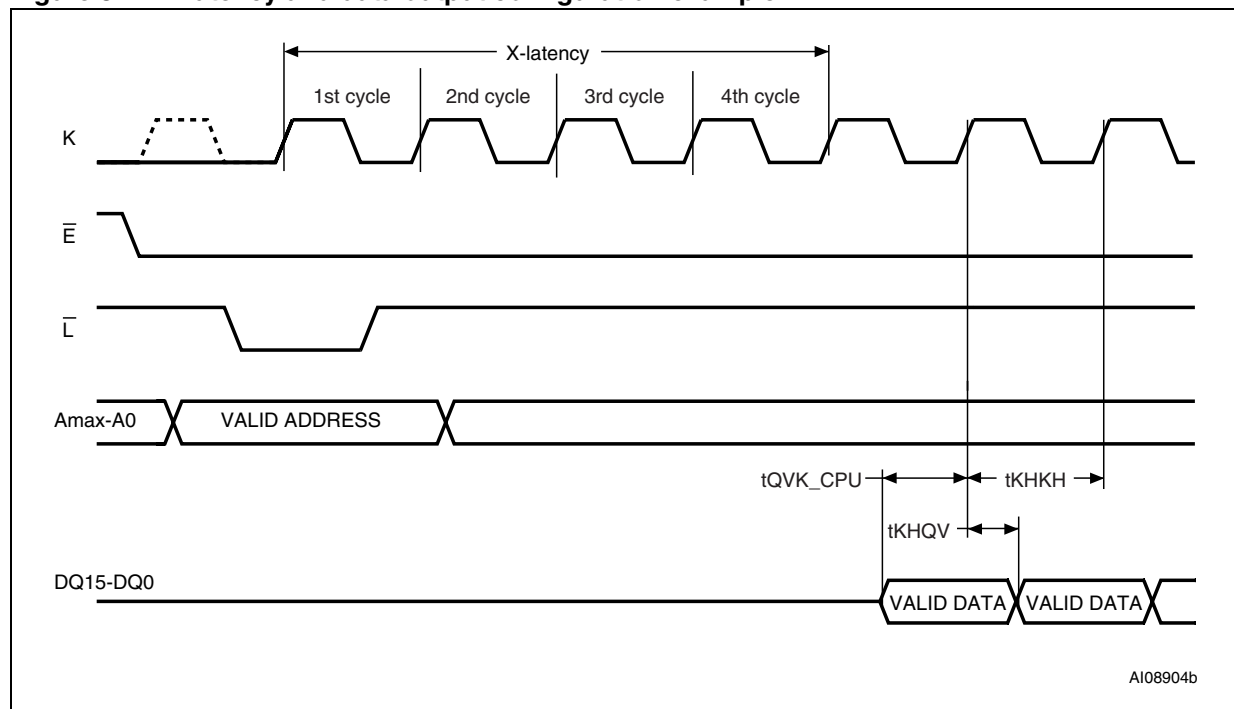
| Bit       | Description             | Value                         | Description  |
|-----------|-------------------------|-------------------------------|--|
| CR15      | Read Select             | 0                             | Synchronous read   |
|           |                         | 1                             | Asynchronous read (default at power-on)                                |
| CR14-CR11 | X latency               | 0011                          | 3 Clock latency  |
|           |                         | 0100                          | 4 Clock latency  |
|           |                         | 0101                          | 5 Clock latency  |
|           |                         | 0110                          | 6 Clock latency  |
|           |                         | 0111                          | 7 Clock latency  |
|           |                         | 1000                          | 8 Clock latency  |
|           |                         | 1001                          | 9 Clock latency  |
|           |                         | 1010                          | 10 Clock latency   |
|           |                         | 1011                          | 11 Clock latency   |
|           |                         | 1100                          | 12 Clock latency   |
|           |                         | 1101                          | 13 Clock latency   |
|           |                         | 1110                          | 14 Clock latency   |
|           |                         | 1111                          | 15 Clock latency (default)   |
|           |                         | Other configurations reserved |  |
| CR10      | Wait polarity           | 0                             | WAIT is active Low (default)   |
|           |                         | 1                             | WAIT is active High  |
| CR9       | Reserved <sup>(1)</sup> |                               |  |
| CR8       | Wait configuration      | 0                             | WAIT is active during wait state                                       |
|           |                         | 1                             | WAIT is active one data cycle before wait state (default) <sup>1</sup> |
| CR7-CR3   | Reserved <sup>(1)</sup> |                               |  |
| CR2-CR0   | Burst length            | 010                           | 8 words (wrap only)  |
|           |                         | 011                           | 16 words (wrap only)   |
|           |                         | 111                           | Continuous (default, no wrap only)                                     |

1. Reserved bits should be cleared to '0'.

Table 17. Burst type definition

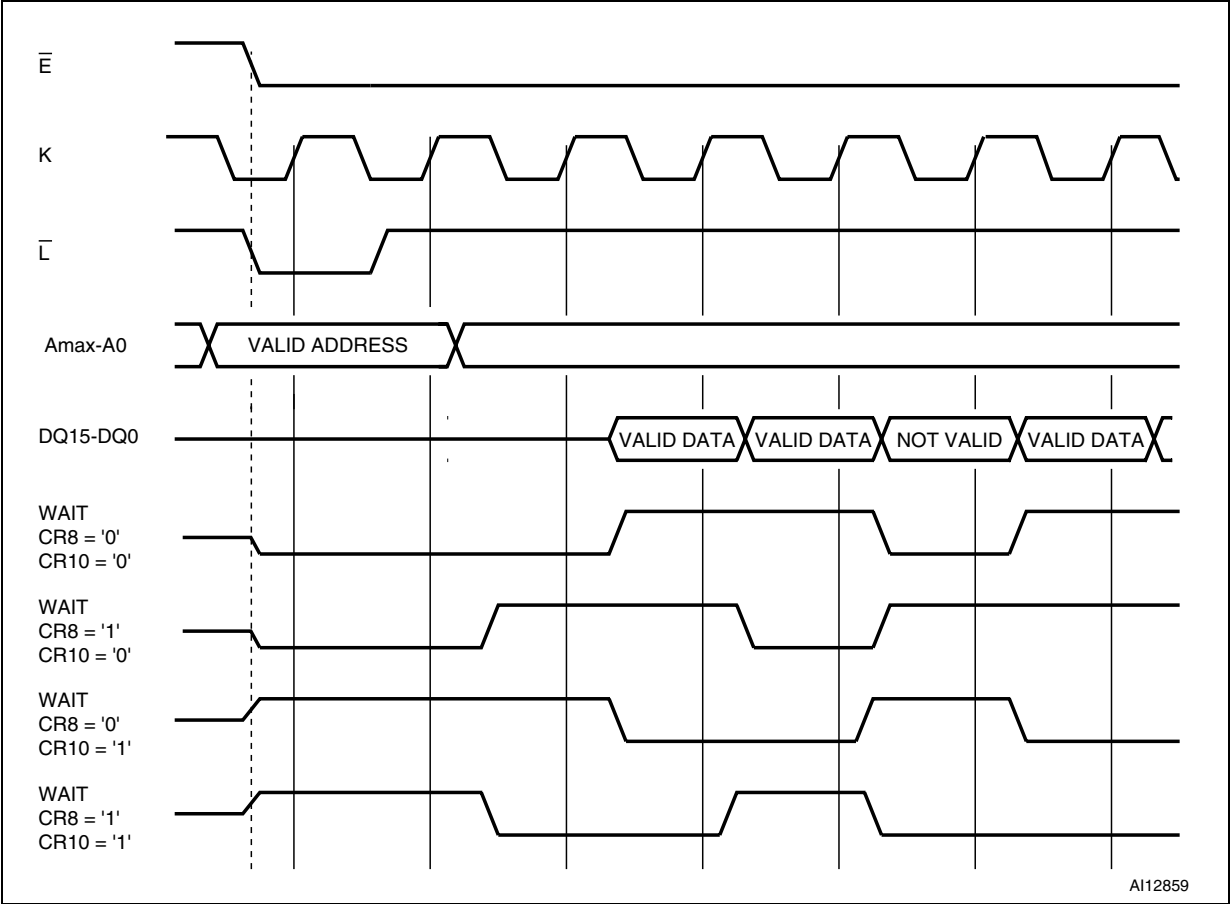
| Start address | 8 words               | 16 words                              | Continuous burst     |
|---------------|-----------------------|---------------------------------------|----------------------|
| 0             | 0-1-2-3-4-5-6-7       | 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 | 0-1-2-3-4-5-6...     |
| 1             | 1-2-3-4-5-6-7-0       | 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-0 | 1-2-3-4-5-6-7...     |
| 2             | 2-3-4-5-6-7-0-1       | 2-3-4-5-6-7-8-9-10-11-12-13-14-15-0-1 | 2-3-4-5-6-7-8...     |
| 3             | 3-4-5-6-7-0-1-2       | 3-4-5-6-7-8-9-10-11-12-13-14-15-0-1-2 | 3-4-5-6-7-8-9...     |
| ...           |                       |                                       |                      |
| 7             | 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... |
| ...           |                       |                                       |                      |
| 12            | 12-13-14-15-8-9-10-11 | 12-13-14-15-0-1-2-3-4-5-6-7-8-9-10-11 | 12-13-14-15-16-17... |
| 13            | 13-14-15-8-9-10-11-12 | 13-14-15-0-1-2-3-4-5-6-7-8-9-10-11-12 | 13-14-15-16-17-18... |
| 14            | 14-15-8-9-10-11-12-13 | 14-15-0-1-2-3-4-5-6-7-8-9-10-11-12-13 | 14-15-16-17-18-19... |
| 15            | 15-8-9-10-11-12-13-14 | 15-0-1-2-3-4-5-6-7-8-9-10-11-12-13-14 | 15-16-17-18-19-20... |

Figure 8. X latency and data output configuration example



1. The settings shown are X latency = 4.  $t_{QVK\_CPU}$  is the data setup time required by the system CPU.

Figure 9. Wait configuration example



## 8 Enhanced Configuration Register

The ECR (Enhanced Configuration Register) enables the deep power-down mode and configures the output driver strength. It is set through the command interface using the Set Enhanced Configuration Register command.

The contents of the ECR can be read by issuing the Read Electronic Signature command, and then by reading from the bank base address + 06h.

The ECR is volatile: after a reset or a power-down/power-up sequence the register is set to the default value. The Configuration Register bits are described in [Table 18](#).

### 8.1 Deep power-down mode bit (ECR15)

The deep power-down mode bit, ECR15, enable the deep power-down mode.

The device can only enter the deep power-down mode from standby, by, in any order, asserting the DPD pin and setting ECR15 to '1'.

When the device is in the deep power-down mode, de-asserting the DPD pin and/or resetting ECR15 causes the device to revert to standby mode.

### 8.2 Deep power-down polarity bit (ECR14)

The deep power-down polarity bit sets the polarity of the DPD signal.

When:

- ECR14 = 0, the DPD signal is active Low (default).
- ECR14 = 1, the DPD signal is active High.

### 8.3 Output driver control bits (ECR2-ECR0)

The output driver control bits, ECR0, ECR1 and ECR2, selects the output driver impedance best suited to the system requirements.

After reset or power-up the output driver control bits are set to the Enhanced Configuration Register default value (ECR2-ECR0 = 100, that is 30  $\Omega$  (30 pF) (default)). Optimum performance is only achieved if the output driver impedance is properly configured.

Once a configuration has been selected, all data and wait output drivers are set to the same setting.

[Table 18](#) lists the output driver impedances at  $V_{DDQ}/2$  and the loads that correspond for each ECR2-ECR0 bit configuration.

**Table 18. Enhanced Configuration Register**

| Bit        | Description             | Value                         | Description                   |
|------------|-------------------------|-------------------------------|-------------------------------|
| ECR15      | Deep power-down mode    | 0                             | DPD mode disabled (default)   |
|            |                         | 1                             | DPD mode enabled              |
| ECR14      | DPD polarity            | 0                             | DPD is active Low (default)   |
|            |                         | 1                             | DPD is active High            |
| ECR13-ECR3 | Reserved <sup>(1)</sup> |                               |                               |
| ECR2-ECR0  | Output driver Impedance | 001                           | 90 $\Omega$ (10 pF)           |
|            |                         | 010                           | 60 $\Omega$ (15 pF)           |
|            |                         | 011                           | 45 $\Omega$ (20 pF)           |
|            |                         | 100                           | 30 $\Omega$ (30 pF) (default) |
|            |                         | 101                           | 20 $\Omega$ (35 pF)           |
|            |                         | 110                           | 15 $\Omega$ (40 pF)           |
|            |                         | Other configurations reserved |                               |

1. Reserved bits should be cleared to '0'.

## 9 Extended Flash array (EFA)

In addition to its main array, the M58PRxxxLE features an EFA that is divided into 4 blocks of 4 KWords each. See [Table 5: EFA memory map](#).

The EFA blocks are accessed through a separate set of commands (see [Section 4: Command interface](#) for details).

The operations available on the EFA blocks are asynchronous random access read, single synchronous read, (single word) program, erase, block lock, block unlock, and block lock-down.

The EFA blocks support program/erase suspend and dual operations with the main array. Dual operations between the EFA and the OTP area are not supported. See [Table 21: Dual operation limitations](#) for details.

## 10 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 7: Configuration Register](#) for details). All banks support both asynchronous and synchronous read operations.

### 10.1 Asynchronous read mode

In asynchronous read operations the clock signal is 'don't care'. The device outputs the data corresponding to the address latched, such as 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, therefore, 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 16 words and is addressed by address inputs A0, A1, A2 and A3. During the page access, Amax-A4 and  $\bar{L}$  must remain stable.

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.

Read operations to read non-array data (Status Register, electronic signature, CFI) should be performed in asynchronous single word mode. If the asynchronous page mode is used to read non-array data, only the first output data is valid and all subsequent data is not accurately determined.

The asynchronous page read mode is not available in the EFA.

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 state the power consumption is reduced to the standby value and the outputs are still driven.

In asynchronous read mode, the Wait signal is always de-asserted.

See [Table 29: Asynchronous read AC characteristics](#), [Figure 12: Asynchronous random access read AC waveforms](#) and [Figure 13: Asynchronous page read AC waveforms](#) for details.

## 10.2 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, read electronic signature and read EFA, 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 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 CR14-CR11 of the Configuration Register.

The number of words to be output during a synchronous burst read operation can be configured as 8 words, 16 words or continuous (burst length bits CR2-CR0).

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 an 8- and 16-word burst. It is only de-asserted when output data is valid.

In continuous burst read mode, a wait state occurs when crossing the first 16-word boundary if the start address is not 16-word aligned.

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

See [Table 30: Synchronous read AC characteristics](#) and [Figure 14: Synchronous burst read AC waveforms](#) for details.

## 10.3 Single synchronous read mode

Single synchronous read operations are similar to synchronous burst read operations except that the memory outputs the same data until the burst length requirements are satisfied (according to Configuration Register bits CR2-CR0).

Single synchronous read operations are used to read the EFA, electronic signature, Status Register, CFI, block protection status, Configuration Register status or Protection Register.

When the addressed bank is in read CFI, read Status Register or read electronic signature mode, the Wait signal is asserted during the X latency, the wait state and at the end of a 4-, 8- and 16-word burst. It is only de-asserted when the output data is valid.

See [Table 30: Synchronous read AC characteristics](#) and [Figure 14: Synchronous burst read AC waveforms](#) for details.

## 11 Dual operations and multiple bank architecture

The multiple bank architecture of the M58PRxxxLE gives greater flexibility for software developers to split the code and data spaces within the memory array. The dual operations feature simplifies the software management of the device by allowing code to be executed from one bank while another bank is being programmed or erased.

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

If a read operation is required in a bank, which is programming or erasing, the program or erase operation can be suspended.

Also, if the suspended operation was erase then a Program command can be issued to another block. This means the device can have one block in erase suspend mode, one programming and other banks in read mode.

Bus read operations are allowed in another bank between setup and confirm cycles of program or erase operations.

By using a combination of these features, read operations are possible at any moment in the M58PRxxxLE device.

Dual operations between the EFA and either of the CFI, the OTP, or the electronic signature memory space are not allowed. [Table 21](#) shows which dual operations are allowed or not between the CFI, the OTP, the electronic signature locations and the memory array.

[Table 19](#) and [Table 20](#) show the dual operations possible in other banks and in the same bank.

**Table 19. 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 | Read EFA | Program, Buffer Program | Block Erase | Program/ Erase Suspend | Program/ Erase Resume |
| Idle              | Yes                              | Yes                  | Yes            | Yes                       | Yes      | Yes                     | Yes         | Yes                    | Yes                   |
| Programming       | Yes                              | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | Yes                    | —                     |
| Erasing           | Yes                              | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | Yes                    | —                     |
| Program suspended | Yes                              | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | —                      | Yes                   |
| Erase suspended   | Yes                              | Yes                  | Yes            | Yes                       | Yes      | Yes                     | —           | —                      | Yes                   |

**Table 20. 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 | Read EFA | Program, Buffer Program | Block Erase | Program/ Erase Suspend | Program/ Erase Resume |
| Idle              | Yes                           | Yes                  | Yes            | Yes                       | Yes      | Yes                     | Yes         | Yes                    | Yes                   |
| Programming       | — <sup>(1)</sup>              | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | Yes                    | —                     |
| Erasing           | — <sup>(1)</sup>              | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | Yes                    | —                     |
| Program suspended | Yes <sup>(2)</sup>            | Yes                  | Yes            | Yes                       | Yes      | —                       | —           | —                      | Yes                   |
| Erase suspended   | Yes <sup>(2)</sup>            | Yes                  | 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 program region that is being programmed.

**Table 21. Dual operation limitations**

| OTP, EFA or CFI data | Main array bank | Comments   |
|----------------------|-----------------|--|
| Read                 | Program/erase   | While programming or erasing in a main array bank, the OTP, CFI data and EFA blocks may be read from any other bank.   |
| Program              | Read            | While programming to the OTP area, read operations are only allowed in the other main array banks. Access to EFA data or CFI data is not allowed.  |
| Program/erase        | Read            | While programming or erasing an EFA block, it is not allowed to read OTP or CFI data. Read operations to the banks whose addresses are not being used to address the EFA, are supported. |

## 12 Block locking

The M58PRxxxLE 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 a 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 22](#) defines all of the possible protection states ( $\overline{WP}$ , DQ1, DQ0), and [Appendix C](#), [Figure 29](#) shows a flowchart for the locking operations.

### 12.1 Reading a block's lock status

The lock status of every block can be read in the read electronic signature mode of the device. To enter this mode issue the Read Electronic Signature command. Subsequent reads at the address specified in [Table 10](#) 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, only by a hardware reset or power-down.

The following sections explain the operation of the locking system.

### 12.2 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 returns 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.

### 12.3 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.

## 12.4 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.

## 12.5 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 are changed immediately, but when the erase is resumed, the erase operation completes. Locking operations cannot be performed during a program suspend.

Table 22. 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 A1 = V<sub>IH</sub> and A0 = V<sub>IL</sub>.
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<sub>IH</sub> on a locked block will restore the previous DQ0 value, giving a 111 or 110.

## 13 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 23](#). Exact erase times may change depending on the memory array condition. The best scenario is when all the bits in the block are at '0' (pre-programmed). The worst scenario 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 M58PRxxxLE the maximum number of program/erase cycles depends on the  $V_{PP}$  voltage supply used.

**Table 23. Program/erase times and endurance cycles<sup>(1) (2)</sup>**

| Parameter         |                                  |   | Condition                   | Min     | Typ  | Typical after 100 kW/E cycles | Max  | Unit    |
|-------------------|----------------------------------|---|-----------------------------|---------|------|-------------------------------|------|---------|
| $V_{PP} = V_{DD}$ | Erase                            | EFA block (4 KWord)                           |                             |         | 0.7  |                               | 2.5  | s       |
|                   |                                  | Main array block (128 KWord)                  |                             |         | 0.9  |                               | 4    | s       |
|                   | Program <sup>(3)</sup>           | Single cell <sup>(4)</sup>                    | Word Program <sup>(5)</sup> |         | 50   |                               | 230  | $\mu$ s |
|                   |                                  |   | Buffer Program              |         | 250  |                               | 500  | $\mu$ s |
|                   |                                  | Single word <sup>(4)</sup>                    | Word Program <sup>(5)</sup> |         | 50   |                               | 230  | $\mu$ s |
|                   |                                  |   | Buffer Program              |         | 250  |                               | 500  | $\mu$ s |
|                   |                                  | Buffer (512 words) (Buffer Program)           |                             |         | 2.15 |                               | 4.3  | ms      |
|                   |                                  | EFA block (4 KWord)                           |                             |         | 0.2  |                               | 0.94 | s       |
|                   |                                  | Main array block (128 KWord) (Buffer Program) |                             |         | 0.55 |                               | 1.1  | s       |
|                   | Suspend Latency                  | Program                                       |                             |         | 20   |                               | 30   | $\mu$ s |
|                   |                                  | Erase   |                             |         | 20   |                               | 30   | $\mu$ s |
|                   | Program/Erase cycles (per block) | Main array block                              |                             | 100 000 |      |                               |      | cycles  |
|                   |                                  | EFA block                                     |                             | 100 000 |      |                               |      |         |
|                   | Blank Check                      | Main array block                              |                             |         | 3.2  |                               |      | ms      |

**Table 23. Program/erase times and endurance cycles<sup>(1) (2)</sup> (continued)**

| Parameter                          |                                     |                               | Condition  | Min     | Typ  | Typical after<br>100 kW/E<br>cycles | Max  | Unit   |
|------------------------------------|-------------------------------------|-------------------------------|--|---------|------|-------------------------------------|------|--------|
| V <sub>PP</sub> = V <sub>PPH</sub> | Erase                               | EFA block (4 KWord)           |  |         | 0.7  |                                     | 2.5  | s      |
|                                    |                                     | Main array block (128 KWord)  |  |         | 0.9  |                                     | 4    | s      |
|                                    | Program <sup>(3)</sup>              | Single cell <sup>(4)</sup>    | Word Program <sup>(5)</sup>                          |         | 50   |                                     | 230  | μs     |
|                                    |                                     | Single word <sup>(4)</sup>    | Word Program <sup>(5)</sup>                          |         | 50   |                                     | 230  | μs     |
|                                    |                                     |                               | Buffer Enhanced<br>Factory<br>Program <sup>(4)</sup> |         | 4.2  |                                     |      | μs     |
|                                    |                                     | Buffer (512<br>words)         | Buffer Program                                       |         | 2.15 |                                     | 4.3  | ms     |
|                                    |                                     |                               | Buffer Enhanced<br>Factory Program                   |         | 2.15 |                                     |      | ms     |
|                                    |                                     | Main block<br>(128 KWords)    | Buffer Program                                       |         | 0.55 |                                     | 1.1  | s      |
|                                    |                                     |                               | Buffer Enhanced<br>Factory Program                   |         | 0.55 |                                     |      | s      |
|                                    |                                     | EFA block (4 KWords)          |  |         | 0.2  |                                     | 0.94 |        |
|                                    | Program/Erase<br>cycles (per block) | Main array block (128 KWords) |  | 100 000 |      |                                     |      | cycles |
|                                    |                                     | EFA block (4 KWords)          |  | 100 000 |      |                                     |      |        |
|                                    | Blank Check                         | Main array block              |  |         | 3.2  |                                     |      | ms     |

1. T<sub>A</sub> = -30 to 85 °C; V<sub>DD</sub> = 1.7 V to 2 V; V<sub>DDQ</sub> = 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.

5. The first Word Program in a program region will take 115 μs, the subsequent words will take 50 μs to program.

## 14 Maximum ratings

Stressing the device above the ratings listed in [Table 24: 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 STMicroelectronics SURE Program and other relevant quality documents.

**Table 24. 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        | -1    | 3   | V     |
| $V_{DD}$   | Supply voltage                 | -1    | 3   | V     |
| $V_{DDQ}$  | Input/output supply voltage    | -1    | 3   | V     |
| $V_{PP}$   | Program voltage                | -1    | 10  | V     |
| $I_O$      | Output short circuit current   |       | 100 | mA    |
| $t_{VPPH}$ | Time for $V_{PP}$ at $V_{PPH}$ |       | 100 | hours |

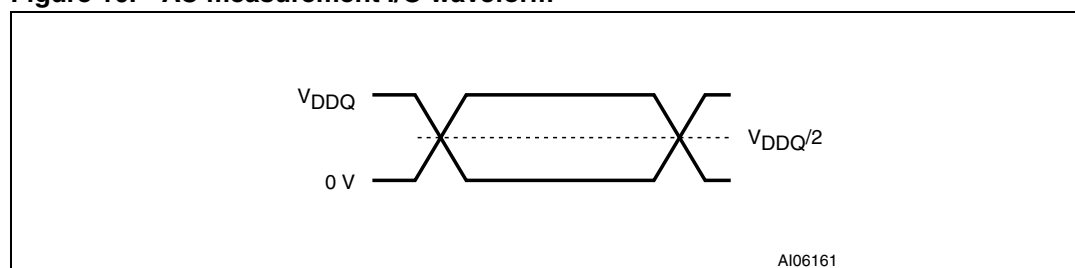
## 15 DC and AC parameters

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 25: 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 25. Operating and AC measurement conditions**

| Parameter   | Min            | Max | Units |
|---|----------------|-----|-------|
| $V_{DD}$ supply voltage                           | 1.7            | 2.0 | V     |
| $V_{DDQ}$ supply voltage                          | 1.7            | 2.0 | V     |
| $V_{PP}$ supply voltage (factory environment)     | 8.5            | 9.5 | V     |
| $V_{PP}$ supply voltage (application environment) | 0.9            | 2.0 | V     |
| Ambient operating temperature                     | -30            | 85  | °C    |
| Load capacitance ( $C_L$ )                        | 30             |     | pF    |
| Input rise and fall times                         |                | 3   | ns    |
| Input pulse voltages                              | 0 to $V_{DDQ}$ |     | V     |
| Input and output timing ref. voltages             | $V_{DDQ}/2$    |     | V     |

**Figure 10. AC measurement I/O waveform**



CL includes JIG capacitance

AI06162

| Symbol           | Parameter          | Test condition         | Min | Max | Unit |
|------------------|--------------------|------------------------|-----|-----|------|
| C <sub>IN</sub>  | Input capacitance  | V <sub>IN</sub> = 0 V  | 2   | 8   | pF   |
| C <sub>OUT</sub> | Output capacitance | V <sub>OUT</sub> = 0 V | 4   | 8   | pF   |

72/124

Table 27. DC characteristics - currents

| Symbol             | Parameter   | Test condition <sup>(1)</sup>   |          | 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)     | $\bar{E} = V_{IL}, \bar{G} = V_{IH}$  |          | 25  | 35      | mA            |
|                    | Supply current<br>Page read (f=13 MHz)            |   |          | 11  | 15      | mA            |
|                    | Supply current<br>Synchronous read (f=66 MHz)     | 8 word  |          | 22  | 32      | mA            |
|                    |   | 16 word   |          | 19  | 26      | mA            |
|                    |   | Continuous  |          | 25  | 34      | mA            |
|                    | Supply current<br>Synchronous read (f = 108 MHz)  | 8 word  |          | 26  | 36      | mA            |
|                    |   | 16 word   |          | 23  | 30      | mA            |
|                    |   | Continuous  |          | 30  | 42      | mA            |
| $I_{DD2}$          | Supply current (reset)                            | $\overline{RP} = V_{SS} \pm 0.2\text{ V}$   | 256 Mbit | 35  | 95      | $\mu\text{A}$ |
|                    |   |   | 512 Mbit | 50  | 120     |               |
|                    |   |   | 1 Gbit   | 70  | 224     |               |
| $I_{DD3}$          | Supply current (standby)                          | $\bar{E} = V_{DDQ} \pm 0.2\text{ V}$<br>$K = V_{SS}$                                  | 256 Mbit | 35  | 95      | $\mu\text{A}$ |
|                    |   |   | 512 Mbit | 50  | 120     |               |
|                    |   |   | 1 Gbit   | 70  | 224     |               |
| $I_{DD4}^{(1)}$    | Supply current (automatic standby)                | $\bar{E} = V_{IL}, \bar{G} = V_{IH},$<br>$\overline{RP} = V_{IH}$                     | 256 Mbit | 35  | 95      | $\mu\text{A}$ |
|                    |   |   | 512 Mbit | 50  | 120     |               |
|                    |   |   | 1 Gbit   | 70  | 224     |               |
| $I_{DD5}^{(2)}$    | Supply current (deep power-down)                  |   |          | 2   | 45      | $\mu\text{A}$ |
| $I_{DD6}^{(3)}$    | Supply current (program)                          | $V_{PP} = V_{PPH}, V_{PP} = V_{DD}$   |          | 35  | 50      | mA            |
|                    | Supply current (erase)                            | $V_{PP} = V_{PPH}, V_{PP} = V_{DD}$   |          | 35  | 50      | mA            |
|                    | Supply current (blank check)                      | $V_{PP} = V_{PPH}, V_{PP} = V_{DD}$   |          | 35  | 50      | mA            |
| $I_{DD7}^{(3)(4)}$ | Supply current (dual operations)                  | Program/erase in one bank, asynchronous read in another bank                          |          | 60  | 85      | mA            |
|                    |   | Program/erase in one bank, synchronous read (continuous, f = 108 MHz) in another bank |          | 65  | 92      | mA            |
| $I_{DD8}^{(3)}$    | Supply current program/ erase suspended (standby) | $\bar{E} = V_{DDQ} \pm 0.2\text{ V}$<br>$K = V_{SS}$                                  | 256 Mbit | 35  | 95      | $\mu\text{A}$ |
|                    |   |   | 512 Mbit | 50  | 120     |               |
|                    |   |   | 1 Gbit   | 70  | 224     |               |

**Table 27. DC characteristics - currents (continued)**

| Symbol          | Parameter  | Test condition <sup>(1)</sup> | Typ  | Max | Unit    |
|-----------------|--|-------------------------------|------|-----|---------|
| $I_{PP1}^{(3)}$ | $V_{PP}$ supply current (program)                        | $V_{PP} = V_{PPH}$            | 8    | 22  | mA      |
|                 |  | $V_{PP} = V_{DD}$             | 0.05 | 0.1 | mA      |
|                 | $V_{PP}$ supply current (erase)                          | $V_{PP} = V_{PPH}$            | 8    | 22  | mA      |
|                 |  | $V_{PP} = V_{DD}$             | 0.05 | 0.1 | mA      |
| $I_{PP2}$       | $V_{PP}$ supply current (read)                           | $V_{PP} \leq V_{DD}$          | 2    | 15  | $\mu$ A |
| $I_{PP3}^{(3)}$ | $V_{PP}$ supply current (standby, program/erase suspend) | $V_{PP} \leq V_{DD}$          | 0.2  | 5   | $\mu$ A |
| $I_{PP4}$       | $V_{PP}$ supply current (blank check)                    | $V_{PP} = V_{PPH}$            | 0.05 | 0.1 | mA      |
|                 |  | $V_{PP} = V_{PP1}$            | 0.05 | 0.1 | mA      |

1. All inputs stable.

2. The DPD current is measured 40  $\mu$ s after entering the deep power-down mode.

3. Sampled only, not 100% tested.

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

**Table 28. 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        | 1.1             | 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    |
| $V_{RPH}$  | $\overline{RP}$ pin extended high voltage |                       |                 |     | 3.3             | V    |
| $V_{LKOQ}$ | $V_{DDQ}$ lock voltage                    |                       | 0.9             |     |                 | V    |

Figure 12. Asynchronous random access read AC waveforms

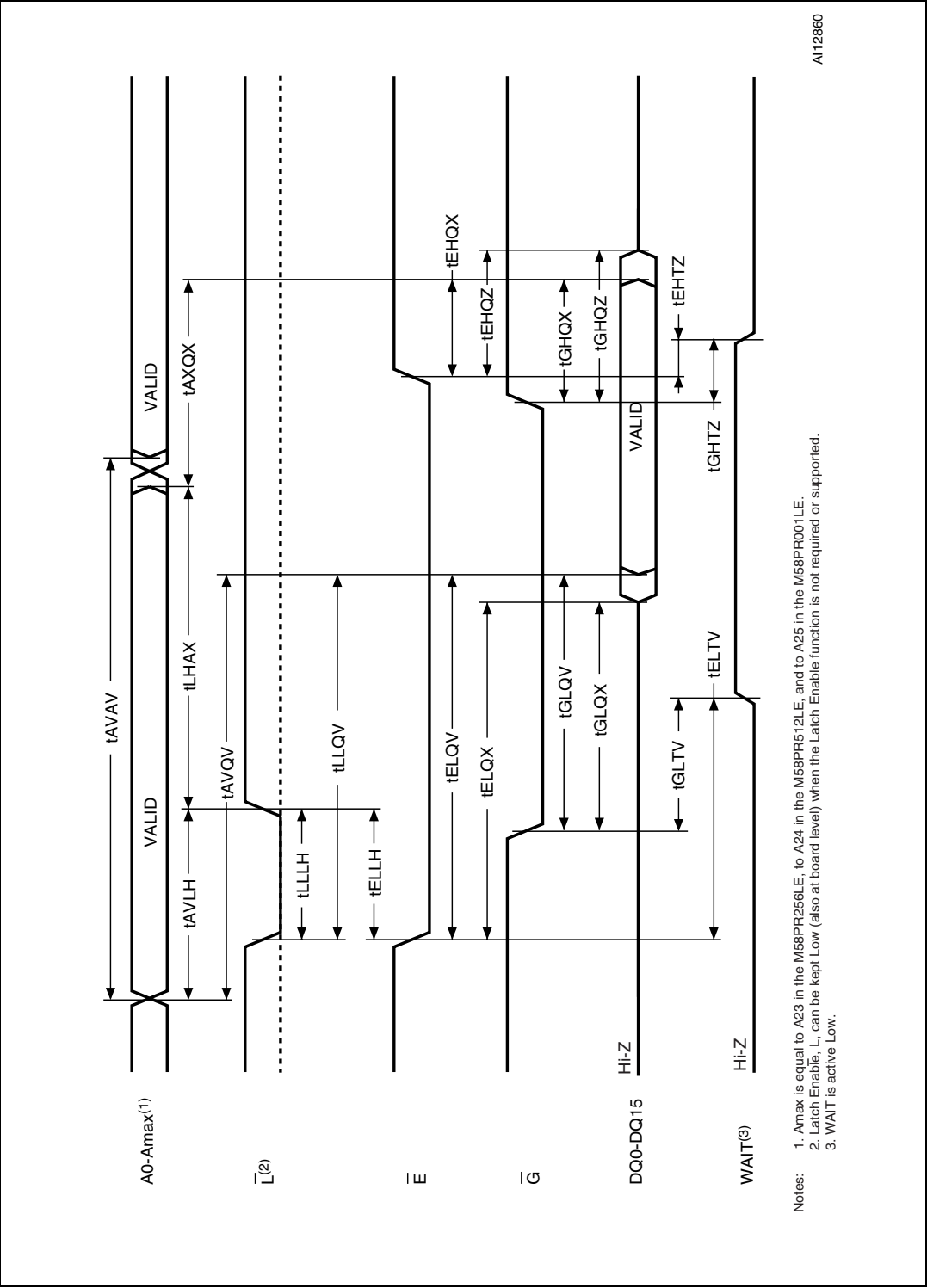


Figure 13. Asynchronous page read AC waveforms

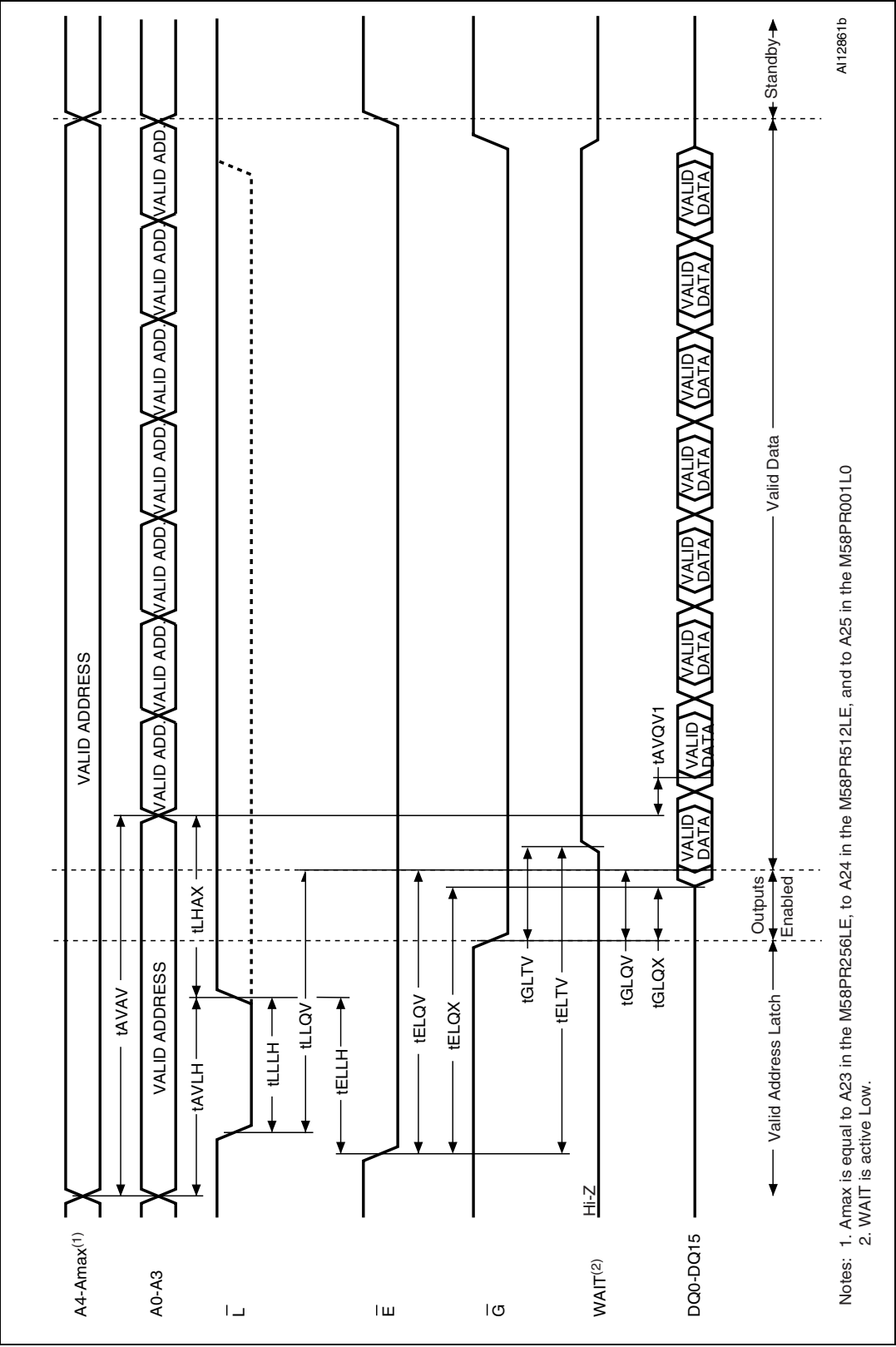


Table 29. Asynchronous read AC characteristics

| Symbol        |                                  | Alt                   | Parameter                                 |     | 108 MHz | 66 MHz | Unit |
|---------------|----------------------------------|-----------------------|---|-----|---------|--------|------|
| Read timings  | t <sub>AVAV</sub>                | t <sub>RC</sub>       | Address Valid to Next Address Valid       | Min | 96      | 96     | ns   |
|               | t <sub>AVQV</sub>                | t <sub>ACC</sub>      | Address Valid to Output Valid (random)    | Max | 96      | 96     | ns   |
|               | t <sub>AVQV1</sub>               | t <sub>PAGE</sub>     | Address Valid to Output Valid (page)      | Max | 20      | 25     | ns   |
|               | t <sub>AXQX</sub> <sup>(1)</sup> | t <sub>OH</sub>       | Address Transition to Output transition   | Min | 0       | 0      | ns   |
|               | t <sub>ELTV</sub>                |                       | Chip Enable Low to Wait Valid             | Max | 14      | 14     | ns   |
|               | t <sub>ELQV</sub> <sup>(2)</sup> | t <sub>CE</sub>       | Chip Enable Low to Output Valid           | Max | 96      | 96     | ns   |
|               | t <sub>ELQX</sub> <sup>(1)</sup> | t <sub>LZ</sub>       | Chip Enable Low to Output transition      | Min | 0       | 0      | ns   |
|               | t <sub>EHTZ</sub>                |                       | Chip Enable High to Wait Hi-Z             | Max | 9       | 14     | ns   |
|               | t <sub>EHQX</sub> <sup>(1)</sup> | t <sub>OH</sub>       | Chip Enable High to Output transition     | Min | 0       | 0      | ns   |
|               | t <sub>EHQZ</sub> <sup>(1)</sup> | t <sub>HZ</sub>       | Chip Enable High to Output Hi-Z           | Max | 9       | 14     | ns   |
|               | t <sub>GLQV</sub> <sup>(2)</sup> | t <sub>OE</sub>       | Output Enable Low to Output Valid         | Max | 20      | 20     | ns   |
|               | t <sub>GLQX</sub> <sup>(1)</sup> | t <sub>OLZ</sub>      | Output Enable Low to Output transition    | Min | 0       | 0      | ns   |
|               | t <sub>GLTV</sub>                |                       | Output Enable Low to Wait Valid           | Max | 7       | 11     | ns   |
|               | t <sub>GHQX</sub> <sup>(1)</sup> | t <sub>OH</sub>       | Output Enable High to Output transition   | Min | 0       | 0      | ns   |
|               | t <sub>GHQZ</sub> <sup>(1)</sup> | t <sub>DF</sub>       | Output Enable High to Output Hi-Z         | Max | 9       | 14     | ns   |
|               | t <sub>GHTZ</sub>                |                       | Output Enable High to Wait Hi-Z           | Max | 9       | 17     | ns   |
| Latch timings | t <sub>AVLH</sub>                | t <sub>AVADVH</sub>   | Address Valid to Latch Enable High        | Min | 5       | 5      | ns   |
|               | t <sub>ELLH</sub>                | t <sub>ELADVH</sub>   | Chip Enable Low to Latch Enable High      | Min | 9       | 10     | ns   |
|               | t <sub>LHAX</sub>                | t <sub>ADVHAX</sub>   | Latch Enable High to Address transition   | Min | 5       | 5      | ns   |
|               | t <sub>LLLH</sub>                | t <sub>ADVLADVH</sub> | Latch Enable Pulse width                  | Min | 7       | 7      | ns   |
|               | t <sub>LLQV</sub>                | t <sub>ADVLQV</sub>   | Latch Enable Low to Output Valid (random) | Max | 96      | 96     | ns   |

1. Sampled only, not 100% tested.

2.  $\bar{Q}$  may be delayed by up to t<sub>ELQV</sub> - t<sub>GLQV</sub> after the falling edge of  $\bar{E}$  without increasing t<sub>ELQV</sub>.

Timing diagram for M58PR001LE showing signals DQ0-DQ15, A0-Amax, L, K, E, G, and WAIT. The diagram illustrates the relationship between address, data, and control signals during a burst configuration. Key timing parameters include tAVLH, tLLH, tLLKH, tAVKH, tELKH, tKHAX, tGLQX, tGLTV, tKHQV, tKHQX, tEHQX, tEHQZ, tGHQX, tGHQZ, tKHTV, tKHTX, tEHTZ, tGLQX, tGLTV, tKHQV, tKHQX, tEHQX, tEHQZ, tGHQX, tGHQZ, tKHTV, tKHTX, tEHTZ. The diagram also shows the Valid Data Flow, Boundary Crossing, Valid Data, and Standby periods.

Figure 15. Single synchronous read AC waveforms

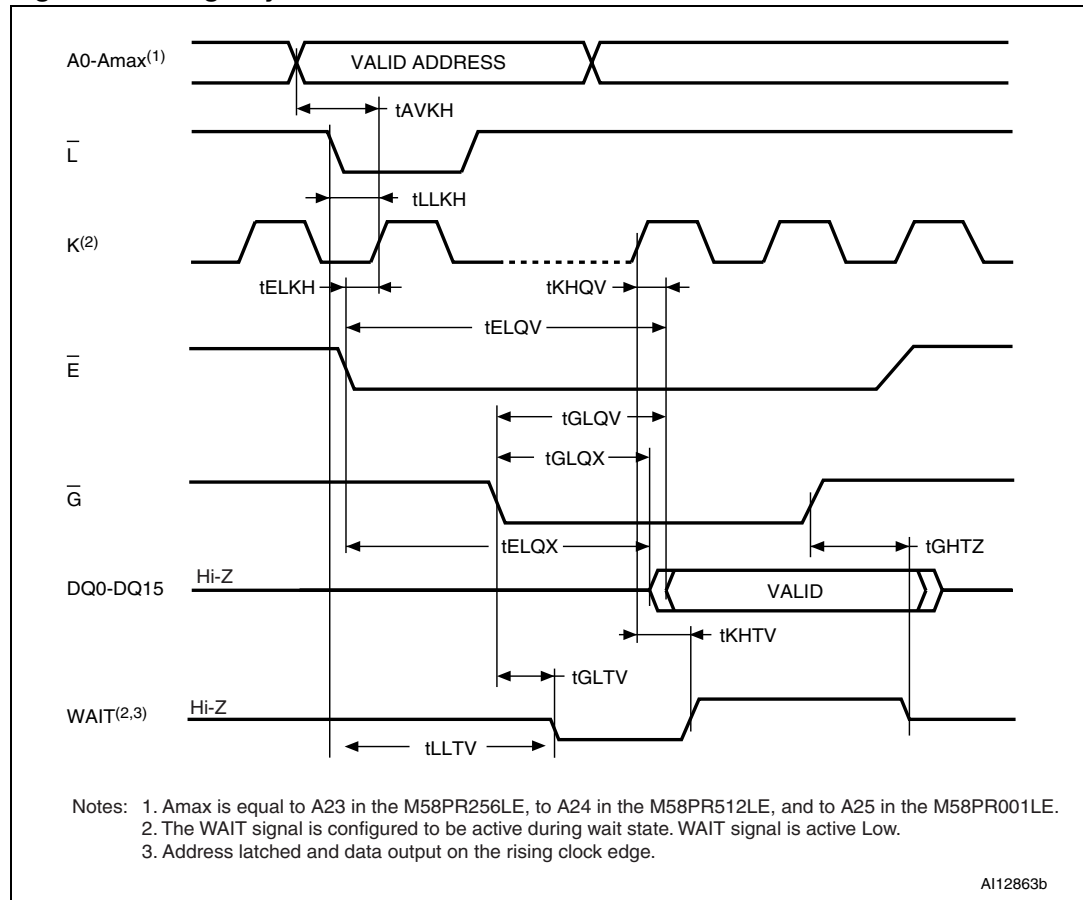
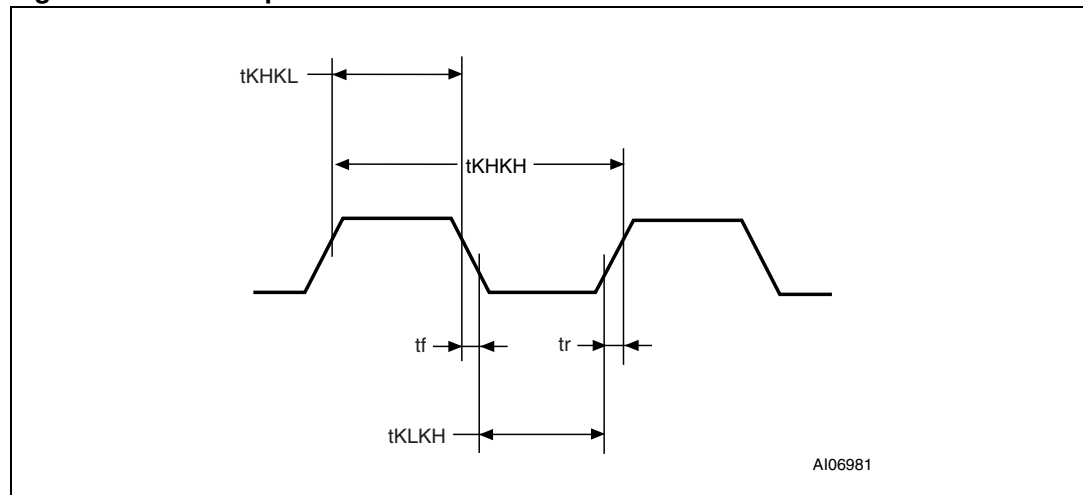


Figure 16. Clock input AC waveform



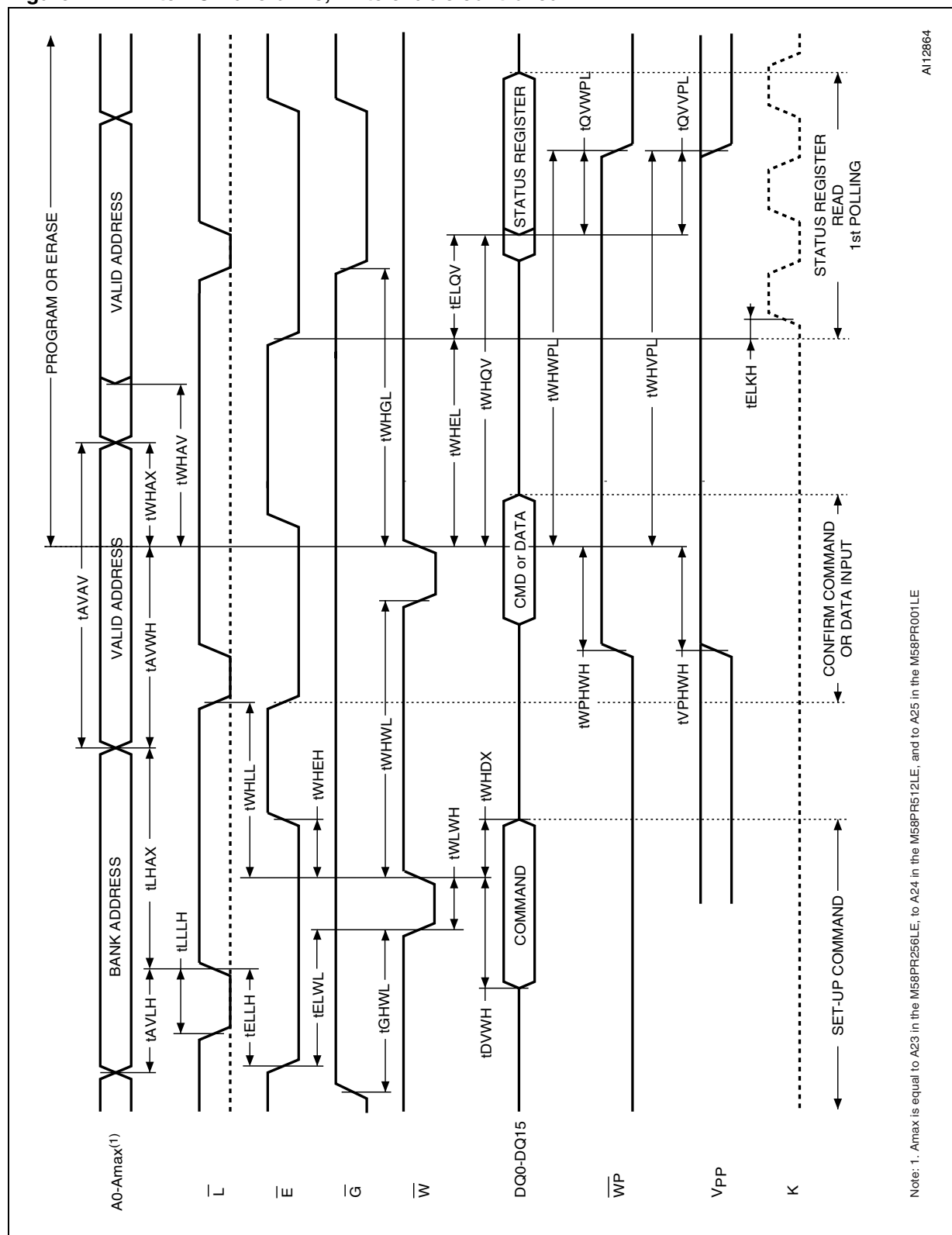
**Table 30. Synchronous read AC characteristics**

| Symbol                   |  | Alt                               | Parameter  |     | 108 MHz | 66 MHz | Unit |
|--------------------------|--|-----------------------------------|--|-----|---------|--------|------|
| Synchronous Read timings | t <sub>AVKH</sub>                      | t <sub>AVCLKH</sub>               | Address Valid to Clock High                                      | Min | 5       | 5      | ns   |
|                          | t <sub>ELKH</sub>                      | t <sub>ELCLKH</sub>               | Chip Enable Low to Clock High                                    | Min | 5       | 5      | ns   |
|                          | t <sub>EHEL</sub>                      |                                   | Chip Enable Pulse width (subsequent synchronous reads)           | Min | 9       | 11     | ns   |
|                          | t <sub>EHTZ</sub>                      |                                   | Chip Enable High to Wait Hi-Z                                    | Max | 9       | 11     | ns   |
|                          | t <sub>KHAX</sub>                      | t <sub>CLKHAX</sub>               | Clock High to Address transition                                 | Min | 5       | 5      | ns   |
|                          | t <sub>KHQV</sub><br>t <sub>KHTV</sub> | t <sub>CLKHQV</sub>               | Clock High to Output Valid<br>Clock High to WAIT Valid           | Max | 7       | 11     | ns   |
|                          | t <sub>KHQX</sub><br>t <sub>KHTX</sub> | t <sub>CLKHQX</sub>               | Clock High to Output transition<br>Clock High to WAIT transition | Min | 2       | 3      | ns   |
|                          | t <sub>LLKH</sub>                      | t <sub>ADVLCCLK<sub>H</sub></sub> | Latch Enable Low to Clock High                                   | Min | 5       | 5      | ns   |
|                          | t <sub>LLTV</sub>                      |                                   | Latch Enable Low to WAIT Valid                                   | Max | 14      | 14     | ns   |
| Clock specifications     | t <sub>KHKH</sub>                      | t <sub>CLK</sub>                  | Clock period (f=66 MHz)  | Min |         | 15     | ns   |
|                          |  |                                   | Clock period (f=108 MHz)   | Min | 9       |        | ns   |
|                          | t <sub>KHKL</sub><br>t <sub>KLKH</sub> |                                   | Clock High to Clock Low<br>Clock Low to Clock High               | Min | 2.5     | 3.5    | ns   |
|                          | t <sub>f</sub><br>t <sub>r</sub>       |                                   | Clock fall or rise time  | Min | 0.3     | -      | ns   |
|                          |  |                                   |  | Max | 2       | 3      | ns   |

1. Sampled only, not 100% tested.

2. For other timings please refer to [Table 29: Asynchronous read AC characteristics](#).

**Figure 17. Write AC waveforms, write enable controlled**



**Table 31. Write AC characteristics, write enable controlled<sup>(1)</sup>**

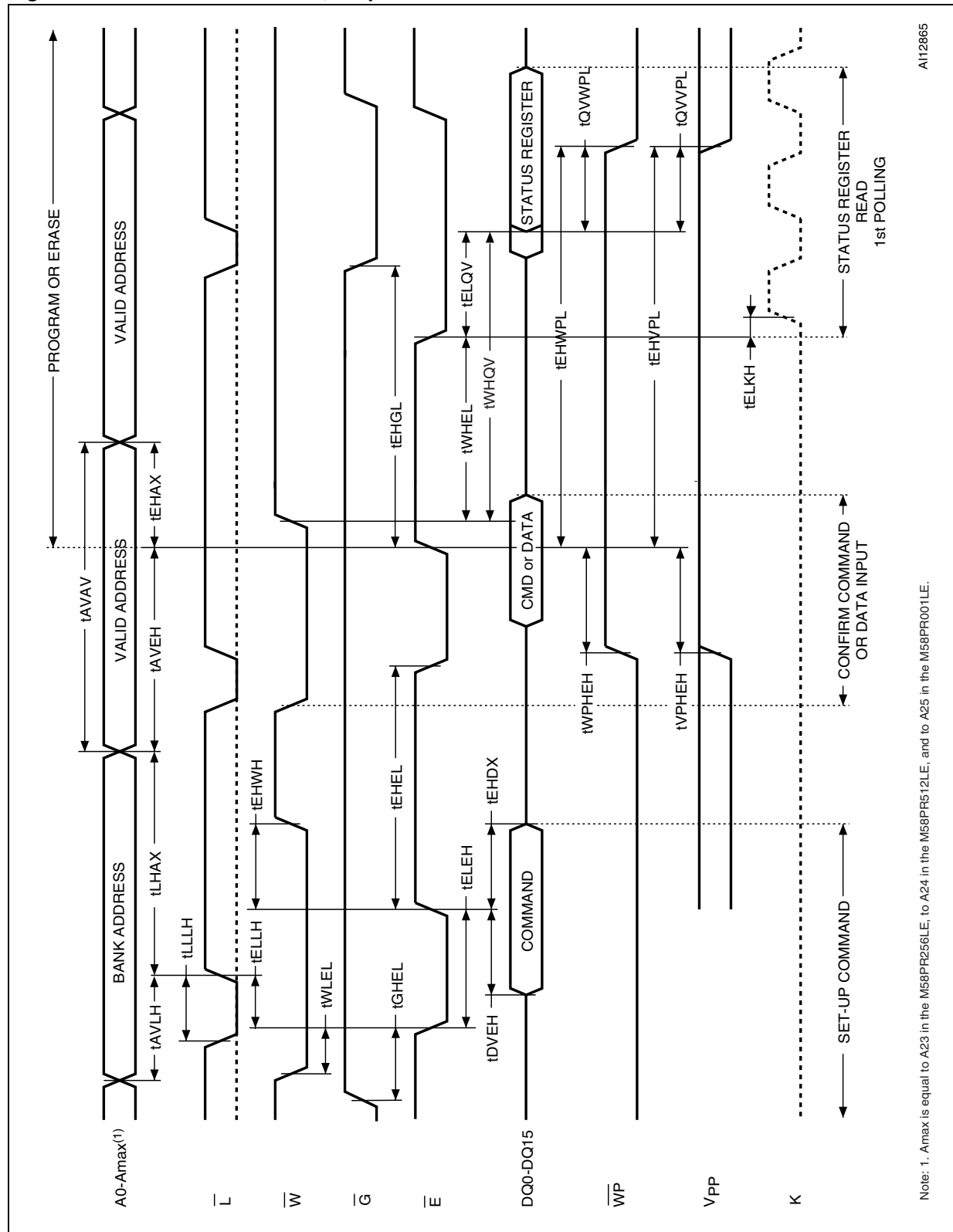
| Symbol                          |                                  | Alt                                   | Parameter   |     | 108 MHz | 66 MHz | Unit |
|---------------------------------|----------------------------------|---------------------------------------|---|-----|---------|--------|------|
| Write Enable Controlled timings | t <sub>AVAV</sub>                | t <sub>WC</sub>                       | Address Valid to Next Address Valid                   | Min | 96      | 96     | ns   |
|                                 | t <sub>AVLH</sub>                |                                       | Address Valid to Latch Enable High                    | Min | 5       | 5      | ns   |
|                                 | t <sub>AVWH</sub> <sup>(2)</sup> |                                       | Address Valid to Write Enable High                    | Min | 40      | 40     | ns   |
|                                 | t <sub>DVWH</sub>                | t <sub>DS</sub>                       | Data Valid to Write Enable High                       | Min | 40      | 40     | ns   |
|                                 | t <sub>ELLH</sub>                |                                       | Chip Enable Low to Latch Enable High                  | Min | 9       | 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 | 96      | 96     | ns   |
|                                 | t <sub>ELKH</sub>                |                                       | Chip Enable Low to Clock High                         | Min | 5       | 5      | ns   |
|                                 | t <sub>GHWL</sub>                |                                       | Output Enable High to Write Enable Low                | Min | 14      | 17     | ns   |
|                                 | t <sub>LHAX</sub>                |                                       | Latch Enable High to Address transition               | Min | 5       | 5      | ns   |
|                                 | t <sub>LLLH</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 | 20      | 20     | 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 | 20      | 20     | ns   |
|                                 | t <sub>WHWL</sub>                | t <sub>WPH</sub>                      | Write Enable High to Write Enable Low                 | Min | 20      | 20     | ns   |
|                                 | t <sub>WHQV</sub>                |                                       | Write Enable High to Output Valid                     | Min | 116     | 116    | ns   |
| t <sub>WLWH</sub>               | t <sub>WP</sub>                  | Write Enable Low to Write Enable High | Min   | 40  | 40      | ns     |      |
| Protection timings              | 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>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.  $t_{WHEL}$  and  $t_{WHLL}$  have this value when reading in the targeted bank or when reading after a Set Configuration Register command. System designers should take this into account and may insert a software No-Op instruction to delay the first read in the same bank after issuing any command and to delay the first read to any address after issuing a Set Configuration Register command. If the first read after the command is a Read Array operation in a different bank and no changes to the Configuration Register have been issued,  $t_{WHEL}$  is 0 ns.

**Figure 18. Write AC waveforms, Chip Enable controlled**



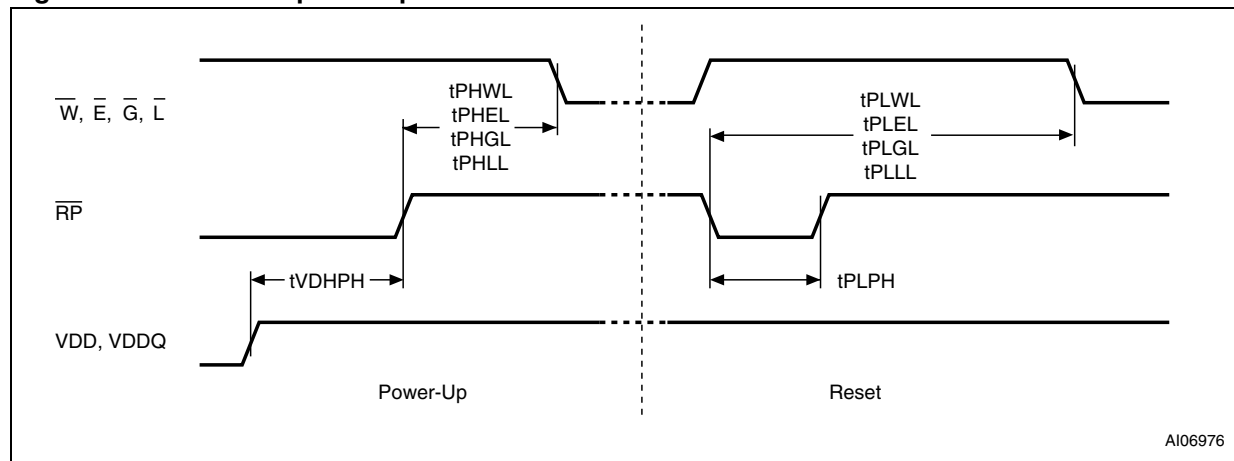
**Table 32. Write AC characteristics, Chip Enable controlled<sup>(1)</sup>**

| Symbol                         |                                  | Alt                | Parameter   |   | 108 MHz | 66 MHz | Unit |
|--------------------------------|----------------------------------|--------------------|---|---|---------|--------|------|
| Chip Enable Controlled timings | t <sub>AVAV</sub>                | t <sub>WC</sub>    | Address Valid to Next Address Valid                   | Min                                     | 96      | 96     | ns   |
|                                | t <sub>AVEH</sub>                | t <sub>WC</sub>    | Address Valid to Chip Enable High                     | Min                                     | 40      | 45     | ns   |
|                                | t <sub>AVLH</sub>                |                    | Address Valid to Latch Enable High                    | Min                                     | 5       | 5      | ns   |
|                                | t <sub>DVEH</sub>                | t <sub>DS</sub>    | Data Valid to Chip Enable High                        | Min                                     | 40      | 40     | 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>EHEL</sub>                | t <sub>CPH</sub>   | Chip Enable High to Chip Enable Low                   | Min                                     | 20      | 20     | ns   |
|                                | t <sub>EHGL</sub>                |                    | Chip Enable High to Output Enable Low                 | Min                                     | 0       | 0      | ns   |
|                                | t <sub>EHHW</sub>                | t <sub>CH</sub>    | Chip Enable High to Write Enable High                 | Min                                     | 0       | 0      | ns   |
|                                | t <sub>ELKH</sub>                |                    | Chip Enable Low to Clock High                         | Min                                     | 5       | 5      | ns   |
|                                | t <sub>ELEH</sub>                | t <sub>CP</sub>    | Chip Enable Low to Chip Enable High                   | Min                                     | 40      | 45     | ns   |
|                                | t <sub>ELLH</sub>                |                    | Chip Enable Low to Latch Enable High                  | Min                                     | 9       | 10     | ns   |
|                                | t <sub>ELQV</sub>                |                    | Chip Enable Low to Output Valid                       | Min                                     | 96      | 96     | ns   |
|                                | t <sub>GHEL</sub>                |                    | Output Enable High to Chip Enable Low                 | Min                                     | 14      | 17     | ns   |
|                                | t <sub>LHAX</sub>                |                    | Latch Enable High to Address transition               | Min                                     | 5       | 5      | ns   |
|                                | t <sub>LLLH</sub>                |                    | Latch Enable Pulse width                              | Min                                     | 7       | 7      | ns   |
|                                | t <sub>WHEL</sub> <sup>(2)</sup> |                    | Write Enable High to Chip Enable Low                  | Min                                     | 20      | 20     | ns   |
|                                | t <sub>WHQV</sub>                |                    | Write Enable High to Output Valid                     | Min                                     | 116     | 116    | ns   |
|                                | t <sub>WLEL</sub>                | t <sub>CS</sub>    | Write Enable Low to Chip Enable Low                   | Min                                     | 0       | 0      | ns   |
|                                | Protection timings               | t <sub>EHVPL</sub> |   | Chip Enable High to V <sub>PP</sub> Low | Min     | 200    | 200  |
| 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.  $t_{WHEL}$  has this value when reading in the targeted bank or when reading after a Set Configuration Register command. System designers should take this into account and may insert a software No-Op instruction to delay the first read in the same bank after issuing any command and to delay the first read to any address after issuing a Set Configuration Register command. If the first read after the command is a Read Array operation in a different bank and no changes to the Configuration Register have been issued,  $t_{WHEL}$  is 0 ns.

Figure 19. Reset and power-up AC waveforms



AI06976

Table 33. Reset and power-up AC characteristics

| Symbol   | Parameter  | Test condition   |     | 108 MHz / 66 MHz | Unit |
|--|--|------------------|-----|------------------|------|
| t <sub>PLWL</sub><br>t <sub>PLEL</sub><br>t <sub>PLGL</sub><br>t <sub>PLLL</sub> | Reset Low to Write Enable Low,   | During program   | Min | 25               | μs   |
|  | Reset Low to Chip Enable Low,  | During erase     | Min | 30               | μs   |
|  | Reset Low to Output Enable Low,<br>Reset Low to Latch Enable Low   | Other conditions | Min | 80               | ns   |
| t <sub>PHWL</sub><br>t <sub>PHEL</sub><br>t <sub>PHGL</sub><br>t <sub>PHLL</sub> | Reset High to Write Enable Low<br>Reset High to Chip Enable Low<br>Reset High to Output Enable Low<br>Reset High to Latch Enable Low |                  | Min | 30               | ns   |
| t <sub>PLPH</sub> <sup>(1),(2)</sup>   | RP Pulse width   |                  | Min | 50               | ns   |
| t <sub>VDHPH</sub> <sup>(3)</sup>  | Supply voltages High to Reset High   |                  | Min | 300              | μs   |

1. The device reset is possible but not guaranteed if t<sub>PLPH</sub> < 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.

Figure 20. Deep power-down AC waveforms

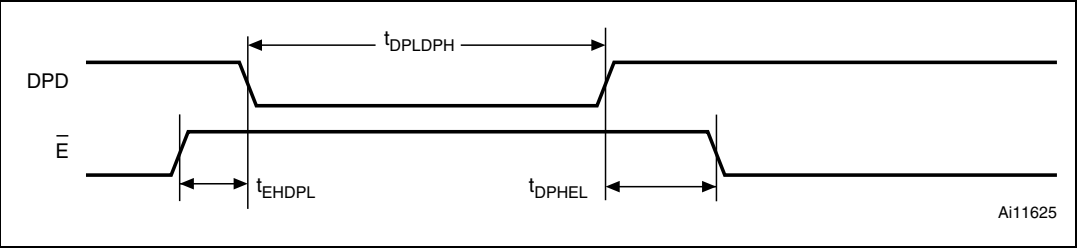


Figure 21. Reset during deep power-down AC waveforms

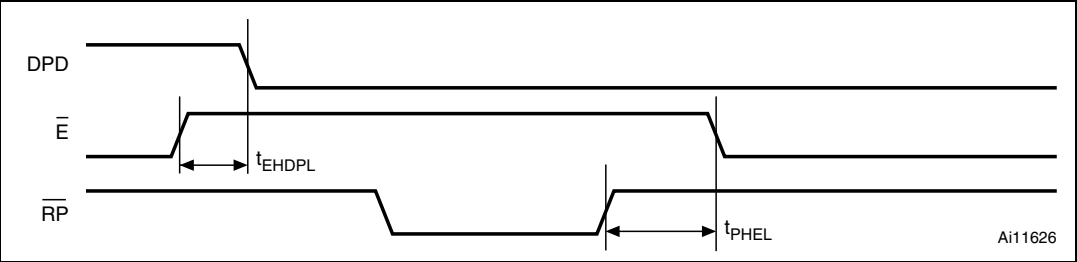


Table 34. Deep power-down AC characteristics

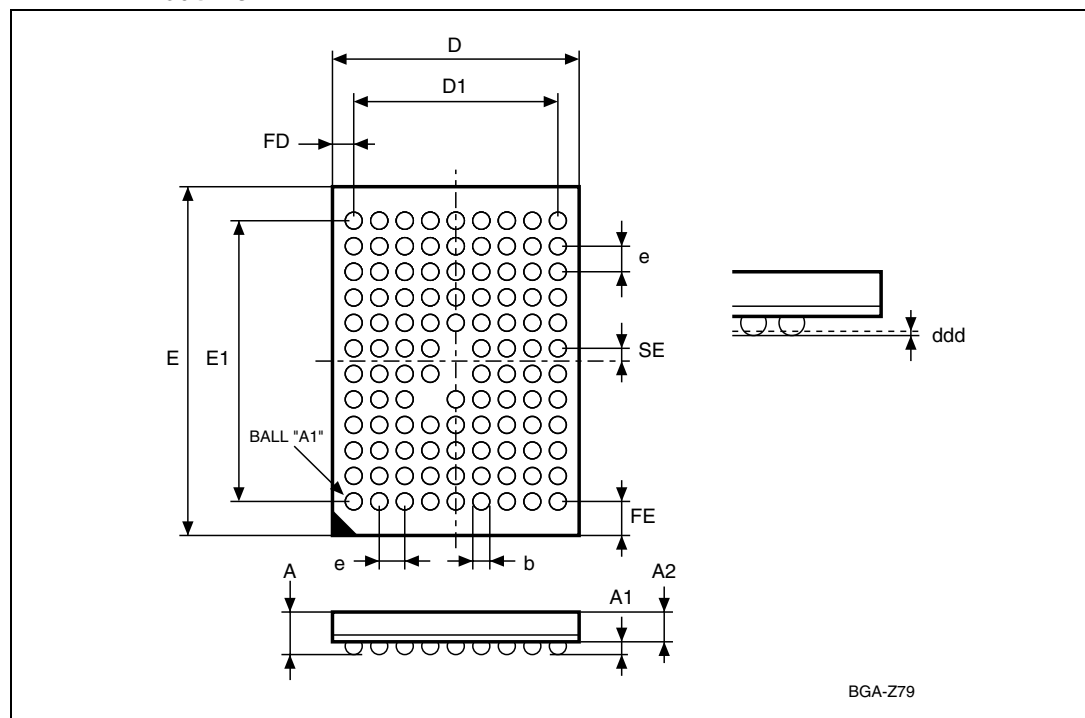
| Symbol       | Parameter   | Test condition         |     | 108 MHz/66 MHz | Unit    |
|--------------|---|------------------------|-----|----------------|---------|
| $t_{DPLDPH}$ | Deep power-down asserted to Deep power-down de-asserted |                        | Min | 50             | ns      |
| $t_{EHDPL}$  | Chip Enable Low to deep power-down asserted             |                        | Min | 0              | $\mu$ s |
| $t_{DPHEL}$  | Deep power-down de-asserted to Chip Enable Low          |                        | Min | 75             | $\mu$ s |
| $t_{PHEL}$   | Reset High to Chip Enable Low                           | During deep power-down | Min | 75             | $\mu$ s |

## 16 Package mechanical

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

The maximum ratings related to soldering conditions are also marked on the inner box label. ECOPACK is an ST trademark. ECOPACK specifications are available at: [www.st.com](http://www.st.com).

**Figure 22. TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline**

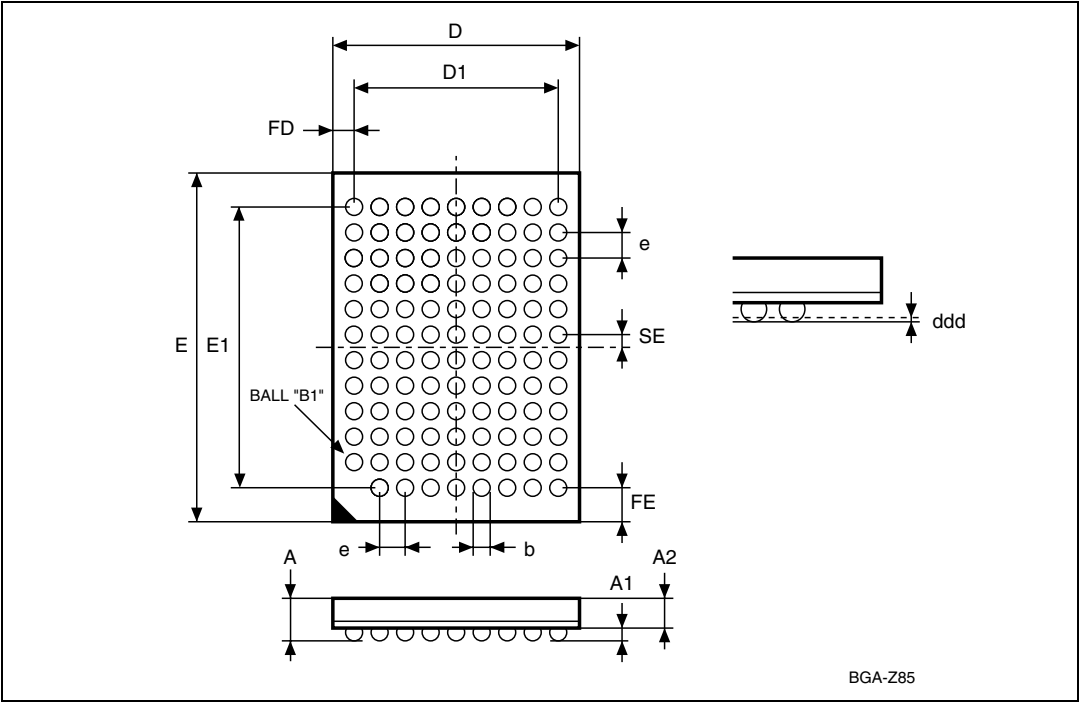


1. Drawing is not to scale.

**Table 35. TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, mechanical data**

| Symbol | millimeters |       |       | inches |       |       |
|--------|-------------|-------|-------|--------|-------|-------|
|        | Typ         | Min   | Max   | Typ    | Min   | Max   |
| A      |             |       | 1.20  |        |       | 0.047 |
| A1     |             | 0.20  |       |        | 0.008 |       |
| A2     | 0.80        |       |       | 0.031  |       |       |
| b      | 0.35        | 0.30  | 0.40  | 0.014  | 0.012 | 0.016 |
| D      | 9.00        | 8.90  | 9.10  | 0.354  | 0.350 | 0.358 |
| D1     | 6.40        |       |       | 0.252  |       |       |
| ddd    |             |       | 0.10  |        |       | 0.004 |
| E      | 11.00       | 10.90 | 11.10 | 0.433  | 0.429 | 0.437 |
| E1     | 8.80        |       |       | 0.346  |       |       |
| e      | 0.80        | –     | –     | 0.031  | –     | –     |
| FD     | 1.30        |       |       | 0.051  |       |       |
| FE     | 1.10        |       |       | 0.043  |       |       |
| SE     | 0.40        |       |       | 0.016  |       |       |

**Figure 23. TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline**



1. Drawing is not to scale.

**Table 36. Stacked TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package mechanical data**

| Symbol | millimeters |       |       | inches |       |       |
|--------|-------------|-------|-------|--------|-------|-------|
|        | Typ         | Min   | Max   | Typ    | Min   | Max   |
| A      |             |       | 1.20  |        |       | 0.047 |
| A1     |             | 0.20  |       |        | 0.008 |       |
| A2     | 0.85        |       |       | 0.033  |       |       |
| b      | 0.35        | 0.30  | 0.40  | 0.014  | 0.012 | 0.016 |
| D      | 8.00        | 7.90  | 8.10  | 0.315  | 0.311 | 0.319 |
| D1     | 6.40        |       |       | 0.252  |       |       |
| ddd    |             |       | 0.10  |        |       | 0.004 |
| E      | 11.00       | 10.90 | 11.10 | 0.433  | 0.429 | 0.437 |
| E1     | 8.80        |       |       | 0.346  |       |       |
| e      | 0.80        |       |       | 0.031  |       |       |
| FD     | 0.80        |       |       | 0.031  |       |       |
| FE     | 1.10        |       |       | 0.043  |       |       |
| SE     | 0.40        |       |       | 0.016  |       |       |

# 17 Part numbering

**Table 37. Ordering information scheme**

|   |           |   |    |     |   |
|---|-----------|---|----|-----|---|
| Example:  | M58PR256L | E | 96 | ZAD | 5 |
| <b>Device type</b>  |           |   |    |     |   |
| M58   |           |   |    |     |   |
| <b>Architecture</b>   |           |   |    |     |   |
| P = multilevel, multiple bank, large buffer   |           |   |    |     |   |
| <b>Operating voltage</b>  |           |   |    |     |   |
| R = V <sub>DD</sub> = 1.7 V to 2.0 V, V <sub>DDQ</sub> = 1.7 V to 2.0 V                   |           |   |    |     |   |
| <b>Density</b>  |           |   |    |     |   |
| 256 = 256 Mbit  |           |   |    |     |   |
| 512 = 512 Mbit  |           |   |    |     |   |
| 001 = 1 Gbit  |           |   |    |     |   |
| <b>Technology</b>   |           |   |    |     |   |
| L = 65 nm technology multilevel design  |           |   |    |     |   |
| <b>Memory organization</b>  |           |   |    |     |   |
| E = uniform blocks  |           |   |    |     |   |
| <b>Speed</b>  |           |   |    |     |   |
| 96 = 96 ns  |           |   |    |     |   |
| <b>Package</b>  |           |   |    |     |   |
| ZAD = stacked TFBGA105 D stacked footprint.<br>ZAC= stacked TFBGA107 C stacked footprint. |           |   |    |     |   |
| <b>Temperature range</b>  |           |   |    |     |   |
| 5 = -30 to 85 °C  |           |   |    |     |   |

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 ST 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 using the information contained in tables [38](#), [39](#), [40](#), [41](#), [42](#) and [43](#).

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:

For the M58PR256LE:

$$\text{Bank\_Number} = \text{Block\_Number} / 16$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 16)$$

For the M58PR512LE:

$$\text{Bank\_Number} = \text{Block\_Number} / 32$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 32)$$

For the M58PR001LE:

$$\text{Bank\_Number} = \text{Block\_Number} / 64$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 64)$$

The Block Base Address is calculated using the formula below:

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

To calculate the Bank number and the Block Number from the Block Base Address:

The Block Number, Bank Number and Block Number Offset can be calculated using the formulas below:

For the M58PR256LE:

$$\text{Block\_Number} = \text{address} / 2^{16}$$

$$\text{Bank\_Number} = \text{Block\_Number} / 16$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 16)$$

For the M58PR512LE:

$$\text{Block\_Number} = \text{address} / 2^{32}$$

$$\text{Bank\_Number} = \text{Block\_Number} / 32$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 32)$$

For the M58PR001LE:

$$\text{Block\_Number} = \text{address} / 2^{64}$$

$$\text{Bank\_Number} = \text{Block\_Number} / 64$$

$$\text{Block\_Number\_Offset} = \text{Block\_Number} - (\text{Bank\_Number} \times 64)$$

**Table 38. M58PR256LE - bank base addresses**

| Bank number | Block numbers | Bank base address |
|-------------|---------------|-------------------|
| 0           | 0 - 15        | 000000            |
| 1           | 16 - 31       | 200000            |
| 2           | 32 - 47       | 400000            |
| 3           | 48 - 63       | 600000            |
| 4           | 64 - 79       | 800000            |
| 5           | 80 - 95       | A00000            |
| 6           | 96 - 111      | C00000            |
| 7           | 112 - 127     | E00000            |

**Table 39. M58PR512LE - bank base addresses**

| Bank number | Block numbers | Bank base address |
|-------------|---------------|-------------------|
| 0           | 0 - 31        | 0000000           |
| 1           | 32 - 63       | 0400000           |
| 2           | 64 - 95       | 0800000           |
| 3           | 96 - 127      | 0C00000           |
| 4           | 128 - 159     | 1000000           |
| 5           | 160 - 191     | 1400000           |
| 6           | 192 - 223     | 1800000           |
| 7           | 224 - 255     | 1C00000           |

**Table 40. M58PR001LE - bank base addresses**

| Bank number | Block numbers | Bank base address |
|-------------|---------------|-------------------|
| 0           | 0 - 63        | 0                 |
| 1           | 64 - 127      | 800000            |
| 2           | 128 - 191     | 1000000           |
| 3           | 192 - 255     | 1800000           |
| 4           | 256 - 319     | 2000000           |
| 5           | 320 - 383     | 2800000           |
| 6           | 384 - 447     | 3000000           |
| 7           | 448 - 511     | 3800000           |

Table 41. M58PR256LE - block addresses

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 0                   | 0000000                   |
| 1                   | 0020000                   |
| 2                   | 0040000                   |
| 3                   | 0060000                   |
| 4                   | 0080000                   |
| 5                   | 00A0000                   |
| 6                   | 00C0000                   |
| 7                   | 00E0000                   |
| 8                   | 0100000                   |
| 9                   | 0120000                   |
| 10                  | 0140000                   |
| 11                  | 0160000                   |
| 12                  | 0180000                   |
| 13                  | 01A0000                   |
| 14                  | 01C0000                   |
| 15                  | 01E0000                   |

Table 42. M58PR512LE - block addresses

| Block number offset | Block base address offset |
|---------------------|---------------------------|
| 0                   | 0000000                   |
| 1                   | 0020000                   |
| 2                   | 0040000                   |
| 3                   | 0060000                   |
| 4                   | 0080000                   |
| 5                   | 00A0000                   |
| 6                   | 00C0000                   |
| 7                   | 00E0000                   |
| 8                   | 0100000                   |
| 9                   | 0120000                   |
| 10                  | 0140000                   |
| 11                  | 0160000                   |
| 12                  | 0180000                   |
| 13                  | 01A0000                   |
| 14                  | 01C0000                   |
| 15                  | 01E0000                   |
| 16                  | 0200000                   |
| 17                  | 0220000                   |
| 18                  | 0240000                   |
| 19                  | 0260000                   |
| 20                  | 0280000                   |
| 21                  | 02A0000                   |
| 22                  | 02C0000                   |
| 23                  | 02E0000                   |
| 24                  | 0300000                   |
| 25                  | 0320000                   |
| 26                  | 0340000                   |
| 27                  | 0360000                   |
| 28                  | 0380000                   |
| 29                  | 03A0000                   |
| 30                  | 03C0000                   |
| 31                  | 03E0000                   |

Table 43. M58PR001LE - block addresses

| Block number | Block base address offset |
|--------------|---------------------------|
| 0            | 0000000                   |
| 1            | 0020000                   |
| 2            | 0040000                   |
| 3            | 0060000                   |
| 4            | 0080000                   |
| 5            | 00A0000                   |
| 6            | 00C0000                   |
| 7            | 00E0000                   |
| 8            | 0100000                   |
| 9            | 0120000                   |
| 10           | 0140000                   |
| 11           | 0160000                   |
| 12           | 0180000                   |
| 13           | 01A0000                   |
| 14           | 01C0000                   |
| 15           | 01E0000                   |
| 16           | 0200000                   |
| 17           | 0220000                   |
| 18           | 0240000                   |
| 19           | 0260000                   |
| 20           | 0280000                   |
| 21           | 02A0000                   |
| 22           | 02C0000                   |
| 23           | 02E0000                   |
| 24           | 0300000                   |
| 25           | 0320000                   |
| 26           | 0340000                   |
| 27           | 0360000                   |
| 28           | 0380000                   |
| 29           | 03A0000                   |
| 30           | 03C0000                   |
| 31           | 03E0000                   |
| 32           | 0400000                   |
| 33           | 0420000                   |
| 34           | 0440000                   |

**Table 43. M58PR001LE - block addresses (continued)**

| Block number | Block base address offset |
|--------------|---------------------------|
| 35           | 0460000                   |
| 36           | 0480000                   |
| 37           | 04A0000                   |
| 38           | 04C0000                   |
| 39           | 04E0000                   |
| 40           | 0500000                   |
| 41           | 0520000                   |
| 42           | 0540000                   |
| 43           | 0560000                   |
| 44           | 0580000                   |
| 45           | 05A0000                   |
| 46           | 05C0000                   |
| 47           | 05E0000                   |
| 48           | 0600000                   |
| 49           | 0620000                   |
| 50           | 0640000                   |
| 51           | 0660000                   |
| 52           | 0680000                   |
| 53           | 06A0000                   |
| 54           | 06C0000                   |
| 55           | 06E0000                   |
| 56           | 0700000                   |
| 57           | 0720000                   |
| 58           | 0740000                   |
| 59           | 0760000                   |
| 60           | 0780000                   |
| 61           | 07A0000                   |
| 62           | 07C0000                   |
| 63           | 07E0000                   |

## 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 [44](#), [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), and the other outputs (DQ8-DQ15) are set to 0.

The CFI data structure also contains a security area where a 64-bit unique security number is written (see [Figure 6: Protection Register memory map](#)). This area can only be accessed in read mode by the final user. It is impossible to change the security number after it has been written by ST. Issue a Read Array command to return to read mode.

**Table 44. Query structure overview<sup>(1)</sup>**

| 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 [45](#), [46](#), [47](#) and [48](#). Query data is always presented on the lowest order data outputs.

**Table 45. CFI query identification string**

| Offset    | Sub-section name        | Description   |  | Value                            |
|-----------|-------------------------|---|--|----------------------------------|
| 000h      | 0020h                   | Manufacturer code   |  | ST                               |
| 001h      | 8818h<br>8819h<br>880Fh | Device code   | M58PR256LE<br>M58PR512LE<br>M58PR001LE | 256 Mbits<br>512 Mbits<br>1 Gbit |
| 002h-00Fh | Reserved                | Reserved  |  |                                  |
| 010h      | 0051h                   | Query unique ASCII string "QRY"   |  | "Q"                              |
| 011h      | 0052h                   |   |  | "R"                              |
| 012h      | 0059h                   |   |  | "Y"                              |
| 013h      | 0000h                   | Primary algorithm command set and control interface<br>ID code 16 bit ID code defining a specific algorithm |  |                                  |
| 014h      | 0002h                   |   |  |                                  |
| 015h      | offset = P = 000Ah      | Address for primary algorithm extended query table<br>(see <a href="#">Table 48</a> )                       |  | P = 10Ah                         |
| 016h      | 0001h                   |   |  |                                  |
| 017h      | 0000h                   | Alternate vendor command set and control interface<br>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 46. 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 | 1.7 V   |
| 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 | 2 V     |
| 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  | 8.5 V   |
| 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  | 9.5 V   |
| 01Fh   | 0006h | Typical timeout per single byte/word program = 2 <sup>n</sup> μs   | 64 μs   |
| 020h   | 000Bh | Typical timeout for buffer program = 2 <sup>n</sup> μs   | 2048 μs |
| 021h   | 000Ah | Typical timeout per individual block erase = 2 <sup>n</sup> ms   | 1 s     |
| 022h   | 0000h | Typical timeout for full chip erase = 2 <sup>n</sup> ms  | NA      |
| 023h   | 0002h | Maximum timeout for word program = 2 <sup>n</sup> times typical  | 256 μs  |
| 024h   | 0002h | Maximum timeout for buffer program = 2 <sup>n</sup> times typical  | 8192 μs |
| 025h   | 0002h | Maximum timeout per individual block erase = 2 <sup>n</sup> times typical  | 4 s     |
| 026h   | 0000h | Maximum timeout for chip erase = 2 <sup>n</sup> times typical  | NA      |

**Table 47. Device geometry definition**

| Offset       | Data           | Description   | Value         |
|--------------|----------------|---|---------------|
| 027h         | 0019h          | M58PR256LE device size = $2^n$ in number of bytes   | 32 Mbytes     |
|              | 001Ah          | M58PR512LE device size = $2^n$ in number of bytes   | 64 Mbytes     |
|              | 001Bh          | M58PR001LE device size = $2^n$ in number of bytes   | 128 Mbytes    |
| 028h<br>029h | 0001h<br>0000h | Flash device interface code description   | x16<br>Async. |
| 02Ah<br>02Bh | 000Ah<br>0000h | Maximum number of bytes in multi-byte program or page = $2^n$   | 1024 bytes    |
| 02Ch         | 0001h          | Number of identical sized erase block regions within the device<br>bit 7 to 0 = x = number of Erase block regions | 1             |
| 02Dh<br>02Eh | 007Fh<br>0000h | M58PR256LE erase block region 1 information<br>Number of identical-size erase blocks = 007Fh+1                    | 127           |
|              | 00FFh<br>0000h | M58PR512LE erase block region 1 information<br>Number of identical-size erase blocks = 00FFh+1                    | 255           |
|              | 01FFh<br>0000h | M58PR001LE erase block region 1 information<br>Number of identical-size erase blocks = 01FFh+1                    | 511           |
| 02Fh<br>030h | 0000h<br>0004h | Erase block region 1 information<br>Block size in region 1 = 0400h * 256 byte                                     | 256 Kbyte     |
| 031h<br>038h | Reserved       | Reserved for future erase block region information  | NA            |

Table 48. 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 | 0034h                   | Minor version number, ASCII  | "4"   |
| (P+5)h = 10Fh | 00E6h                   | Extended query table contents for primary algorithm. Address (P+5)h contains less significant byte (1 = Yes, 0 = No)   |   |
| (P+6)h = 110h | 0007h                   |  |   |
| (P+7)h = 111h | 0000h                   |  |   |
| (P+8)h = 112h | 0000h                   |  |   |
|               |                         | bit 0 Chip Erase supported<br>bit 1 Erase Suspend supported<br>bit 2 Program Suspend supported<br>bit 3 Legacy Lock/Unlock supported<br>bit 4 Queued Erase supported<br>bit 5 Instant individual block locking supported<br>bit 6 Protection bits supported<br>bit 7 Page mode read supported<br>bit 8 Synchronous read supported<br>bit 9 Simultaneous operation supported<br>bit 10 Extended Flash Array Blocks supported<br>bit 11 to 29 reserved; undefined bits are '0'.<br>bit 30 CFI links to follow<br>bit 31 Optional features. 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<br>Yes<br>Yes<br>No<br>No |
| (P+9)h = 113h | 0001h                   | Supported functions after suspend<br>Read Array, Read Status Register and CFI query<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 | 0033h                   | Block protect status   |   |
|               |                         | Defines which bits in the Block Status Register section of the query are implemented.  |   |
| (P+B)h = 115h | 0000h                   | bit 0 Block protect Status Register lock/unlock<br>bit active (1 = Yes, 0 = No)<br>bit 1 Block Lock Status Register Lock-Down bit active (1 = Yes, 0 = No)<br>bit 4 EFA Block protect Status Register Lock/Unlock bit active (1=yes, 2=No)<br>bit 5 EFA Block Lock Status Register Lock-Down bit active (1=yes, 2=No)<br>bit 15 to 6 and 3 to 2 Reserved for future use; undefined bits are '0'  |   |
|               |                         |  |   |
| (P+C)h = 116h | 0018h                   | $V_{DD}$ logic supply optimum program/erase voltage (highest performance)<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 mV   | 1.8 V   |
| (P+D)h = 117h | 0090h                   | $V_{PP}$ supply optimum program/erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 mV   | 9 V   |

**Table 49. 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  | 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 |  | 16      |
| (P+1C)h = 126h | 0004h |  |         |

**Table 50. Burst Read information**

| Offset         | Data  | Description  | Value    |
|----------------|-------|--|----------|
| (P+1D)h = 127h | 0005h | 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.  | 32 bytes |
| (P+1E)h = 128h | 0003h | Number of synchronous mode read configuration fields that follow.  | 3        |
| (P+1F)h = 129h | 0002h | Synchronous mode read capability configuration 1<br>bit 3-7 Reserved<br>bit 0-2 'n' such that 2 <sup>n+1</sup> 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. | 8        |
| (P+20)h = 12Ah | 0003h | Synchronous mode read capability configuration 2   | 16       |
| (P+21)h = 12Bh | 0007h | Synchronous mode read capability configuration 3   | Cont.    |

**Table 51. Bank and erase block region information**

| Offset <sup>(1)</sup> | Data | Description   |
|-----------------------|------|---|
| (P+22)h = 12Ch        | 01h  | Number of bank regions within the device <sup>(2)</sup> |

1. The variable P is a pointer which is defined at CFI offset 015h.

2. Bank regions. There is one bank region, see tables 38, 39, 40, 41, 42 and 43 in [Appendix A](#).

**Table 52. Bank and erase block region 1 information<sup>(1)</sup>**

| Offset         | Data   | Description  |
|----------------|--|--|
| (P+23)h = 12Dh | 16h  | Data size of this bank region information section (addressable locations including this one)   |
| (P+24)h = 12Eh | 00h  |  |
| (P+25)h = 12Fh | 08h  | Number of identical banks within bank region 1   |
| (P+26)h = 130h | 00h  |  |
| (P+27)h = 131h | 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+28)h = 132h | 00h  | Number of program or erase operations allowed in other banks while a bank in this region is being erased<br>Bits 0-3: number of simultaneous program operations<br>Bits 4-7: number of simultaneous erase operations |
| (P+29)h = 133h | 00h  | Number of program or erase operations allowed in other banks while a bank in this region is being erased<br>Bits 0-3: number of simultaneous program operations<br>Bits 4-7: number of simultaneous erase operations |
| (P+2A)h = 134h | 01h  | 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+2B)h = 135h | 0Fh <sup>(3)</sup><br>1Fh <sup>(4)</sup><br>3Fh <sup>(5)</sup> | Bank region 1 Erase block type 1 information<br>Bits 0-15: n+1 = number of identical-sized erase blocks in each bank<br>Bits 16-31: n×256 = number of bytes in erase block region                                    |
| (P+2C)h = 136h | 00h  |  |
| (P+2D)h = 137h | 00h  |  |
| (P+2E)h = 138h | 04h  |  |
| (P+2F)h = 139h | 64h  | Bank region 1 (Erase block type 1)   |
| (P+30)h = 13Ah | 00h  | Minimum block erase cycles × 1000  |
| (P+31)h = 13Bh | 12h  | 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  |

**Table 52. Bank and erase block region 1 information<sup>(1)</sup> (continued)**

| Offset         | Data | Description   |
|----------------|------|---|
| (P+32)h = 13Ch | 03h  | Bank region 1 (Erase block type 1): 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  |
| (P+33)h = 13Dh | 0Ah  | Bank region 1 (Erase block type 1) programming region information<br>Bit 0-7: aligned size of programming region in bytes<br>Bit 8-14: reserved<br>Bit 15: Legacy Flash operation (ignore bit 0-7)<br>Bit 16-23: Control mode valid size in bytes<br>Bit 24-31: reserved<br>Bit 32-39: Control mode invalid size in bytes<br>Bit 40-46: reserved<br>Bit 47: Legacy Flash operation (ignore bit 16-23 and 32-39) |
| (P+34)h = 13Eh | 00h  |   |
| (P+35)h = 13Fh | 10h  |   |
| (P+36)h = 140h | 00h  |   |
| (P+37)h = 141h | 10h  |   |
| (P+38)h = 142h | 00h  |   |

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There is one bank region, see tables 38, 39, 40, 41, 42 and 43 in [Appendix A](#).
3. Applies to M58PR256LE.
4. Applies to M58PR512LE.
5. Applies to M58PR001LE.

**Table 53. Extended Flash array bank and erase block region information**

| Offset <sup>(1)</sup> | Data | Description   |
|-----------------------|------|---|
| (P+39)h = 143h        | 01h  | Number of bank regions within the device <sup>(2)</sup> |

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There is one EFA bank region.

**Table 54. Extended Flash array bank and erase block region 1 information**

| Offset <sup>(1)</sup> | Data | Description  |
|-----------------------|------|--|
| (P+3A)h = 144h        | 16h  | Data size of this bank region information section (addressable locations including this one)   |
| (P+3B)h = 145h        | 00h  |  |
| (P+3C)h = 146h        | 01h  | Number of identical banks within bank region 1   |
| (P+3D)h = 147h        | 00h  |  |
| (P+3E)h = 148h        | 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+3F)h = 149h        | 00h  | Number of program or erase operations allowed in other banks while a bank in this region is being erased<br>Bits 0-3: number of simultaneous program operations<br>Bits 4-7: number of simultaneous erase operations   |
| (P+40)h = 14Ah        | 00h  | Number of program or erase operations allowed in other banks while a bank in this region is being erased<br>Bits 0-3: number of simultaneous program operations<br>Bits 4-7: number of simultaneous erase operations   |
| (P+41)h = 14Bh        | 01h  | 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+42)h = 14Ch        | 03h  | Bank region 1 Erase block type 1 information<br>Bits 0-15: n+1 = number of identical-sized erase blocks in each bank<br>Bits 16-31: n×256 = number of bytes in erase block region                                      |
| (P+43)h = 14Dh        | 00h  |  |
| (P+44)h = 14Eh        | 20h  |  |
| (P+45)h = 14Fh        | 00h  |  |
| (P+46)h = 150h        | 64h  | Bank region 1 (Erase block type 1)   |
| (P+47)h = 151h        | 00h  | Minimum block erase cycles × 1000  |
| (P+48)h = 152h        | 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+49)h = 153h        | 03h  | Bank region 1 (Erase block type 1): 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 |

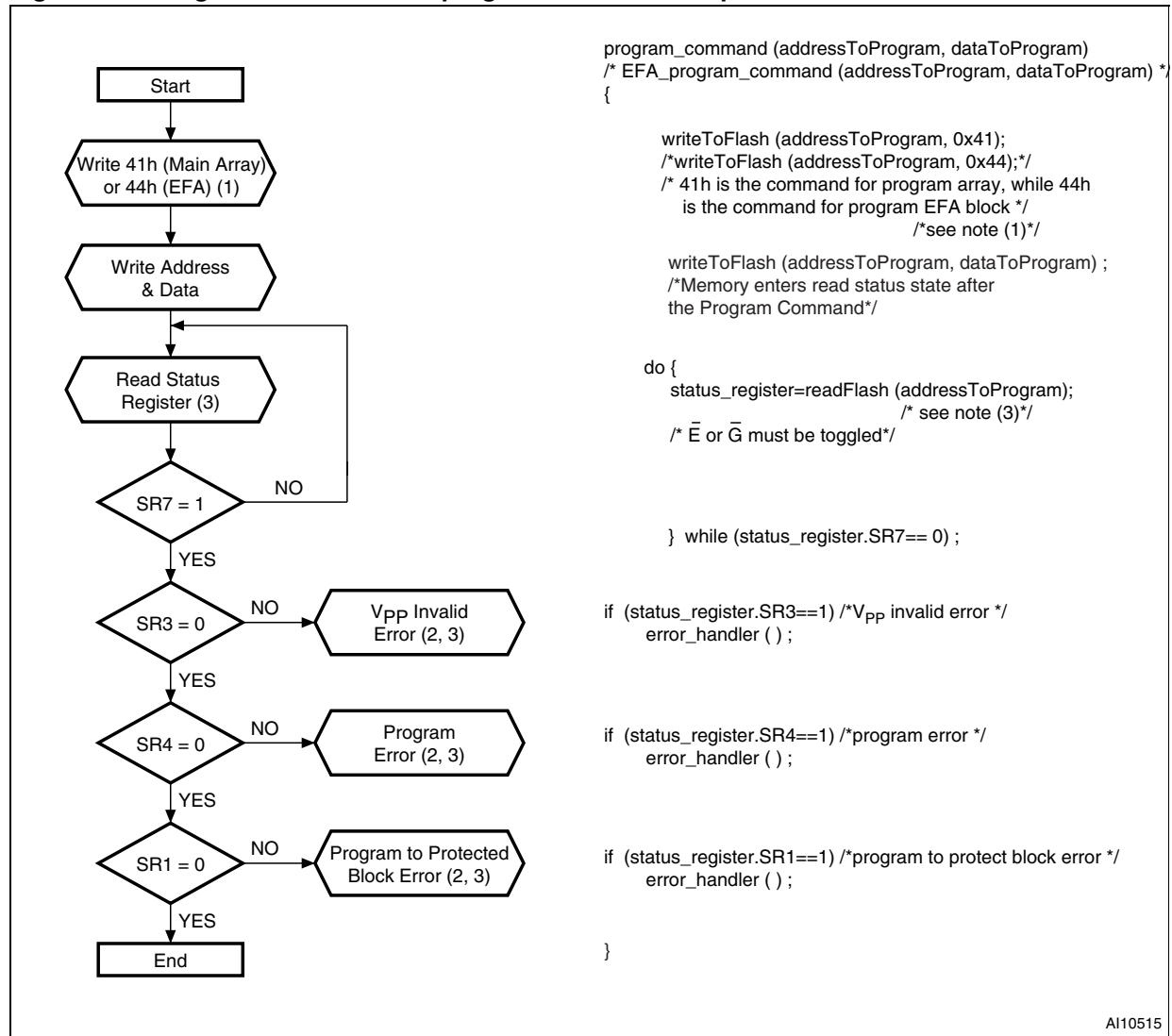
**Table 54. Extended Flash array bank and erase block region 1 information**

| Offset <sup>(1)</sup> | Data | Description   |
|-----------------------|------|---|
| (P+4A)h = 154h        | 00h  | Bank region 1 (Erase block type 1) programming region information |
| (P+4B)h = 155h        | 80h  | Bit 0-7: aligned size of programming region in bytes              |
| (P+4C)h = 156h        | 00h  | Bit 8-14: reserved  |
| (P+4D)h = 157h        | 00h  | Bit 15: legacy Flash operation (ignore bit 0-7)                   |
| (P+4E)h = 158h        | 00h  | Bit 16-23: control mode valid size in bytes                       |
| (P+4F)h = 159h        | 80h  | Bit 24-31: reserved   |
|                       |      | Bit 32-39: control mode invalid size in bytes                     |
|                       |      | Bit 40-46: reserved   |
|                       |      | Bit 47: legacy Flash operation (ignore bit 16-23 and 32-39)       |

1. The variable P is a pointer which is defined at CFI offset 015h.
2. Bank regions. There is one EFA bank region.

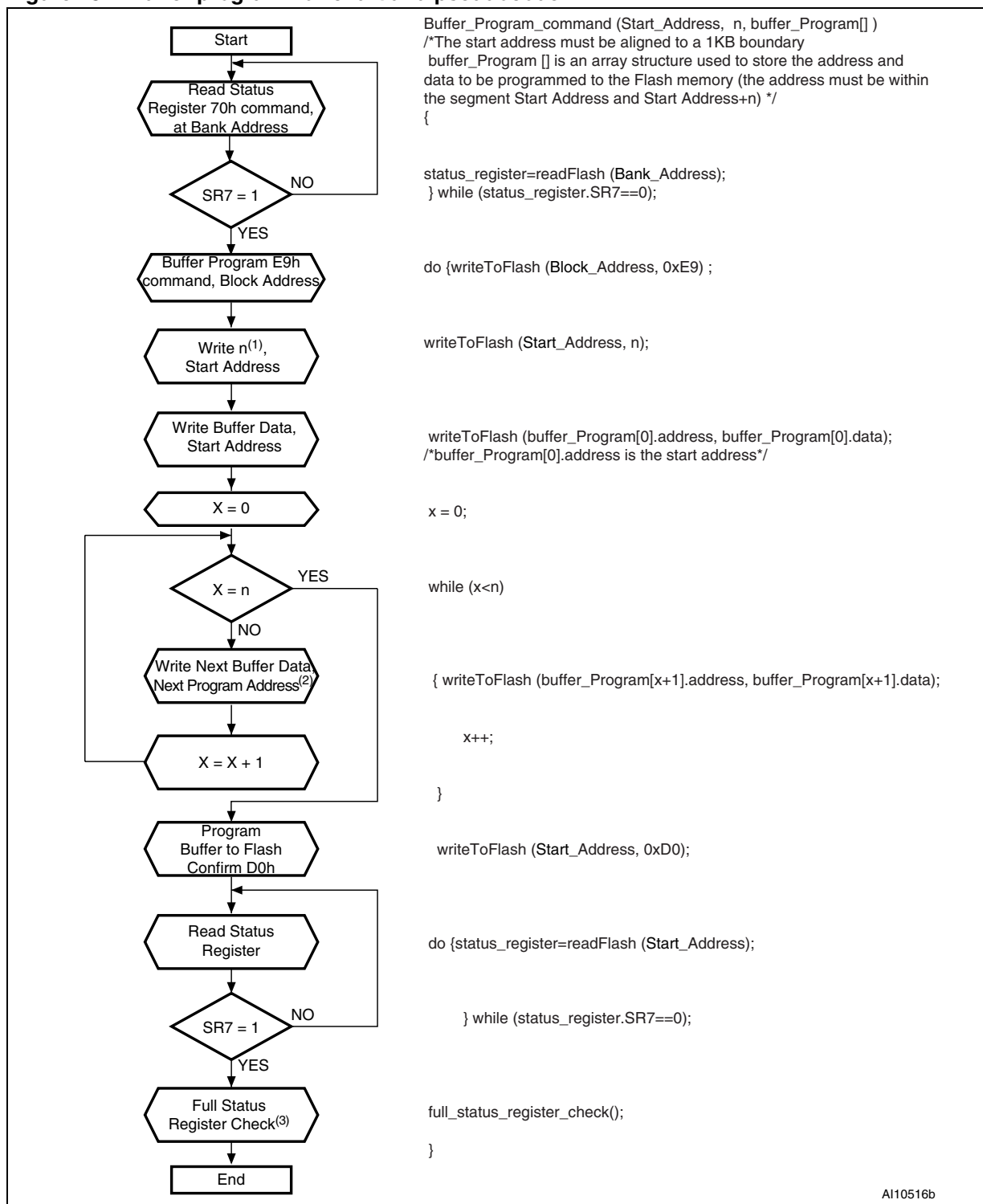
## Appendix C Flowcharts and pseudocodes

Figure 24. Program and EFA block program flowchart and pseudocode



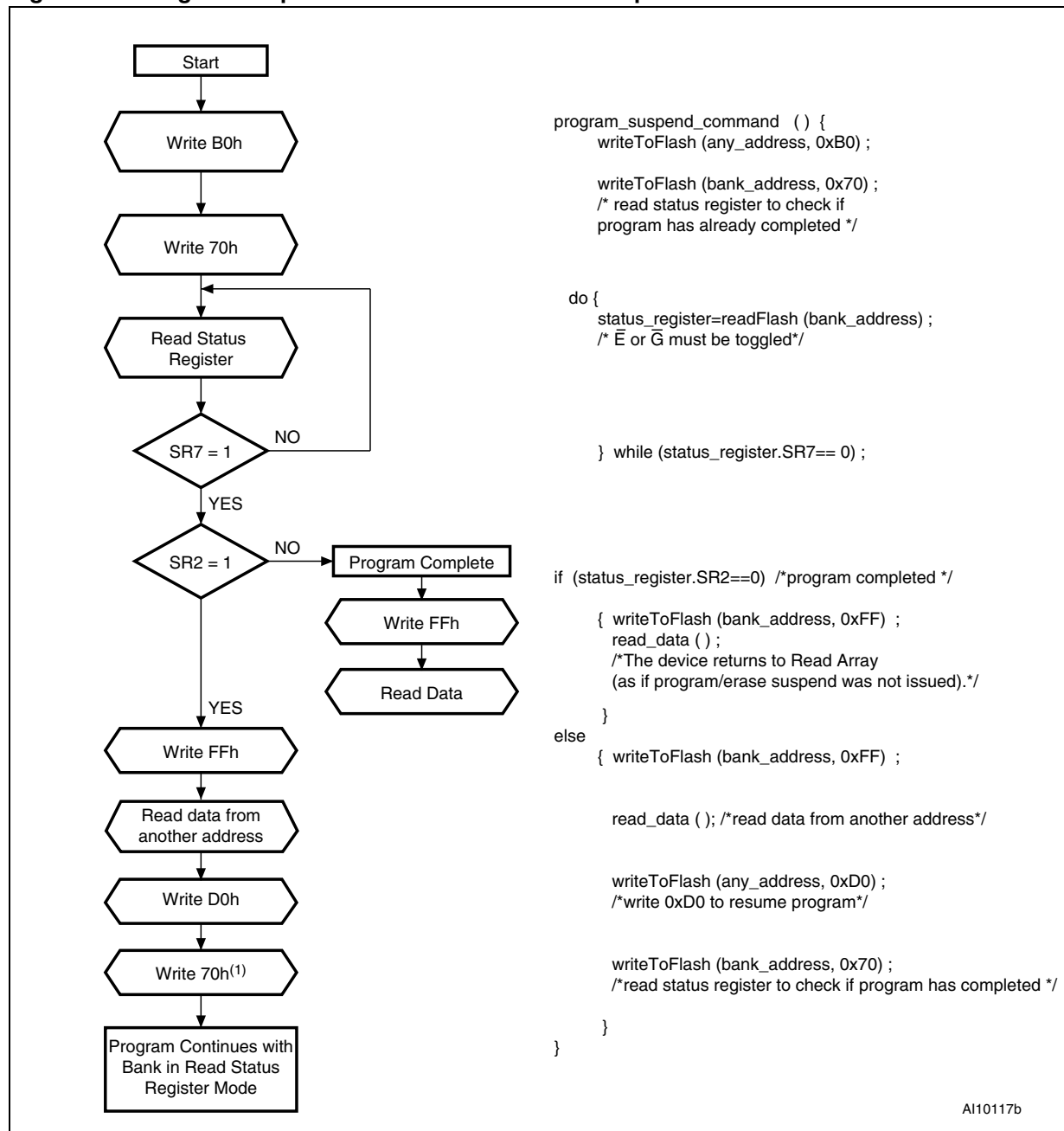
- Any address within the 'A' segment halves (A3=0) in a 1 Kbyte program region configured in control program mode. If a Program command is issued to a program region configured in the Object Program mode, SR4 and SR8 are set.
- Status check of SR1 (protected block), SR3 (V<sub>PP</sub> invalid) and SR4 (program error) can be made after each program operation or after a sequence.
- If an error is found, the Status Register must be cleared before further Program/Erase controller operations.

Figure 25. Buffer program flowchart and pseudocode



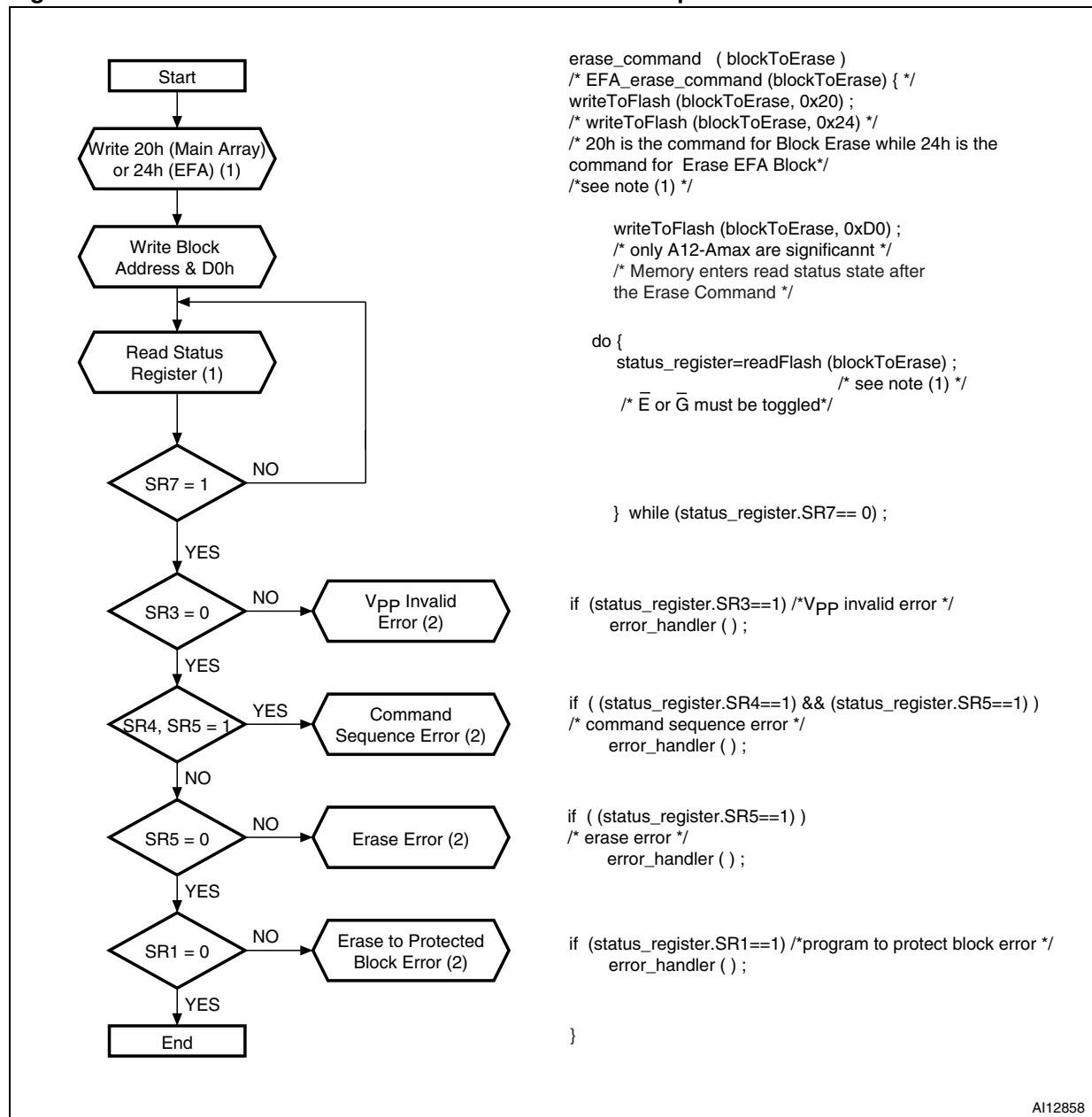
1.  $n + 1$  is the number of data being programmed. The maximum buffer count is 1FF (512 words).
2. Next Program data is an element belonging to `buffer_Program[].data`; Next Program address is an element belonging to `buffer_Program[].address`. In a program region configured in Control Program mode `buffer_Program[].data = FFFFh` if `A3 = 1`.
3. Routine for error check by reading SR3, SR4 and SR1.

Figure 26. Program suspend and resume flowchart and pseudocode



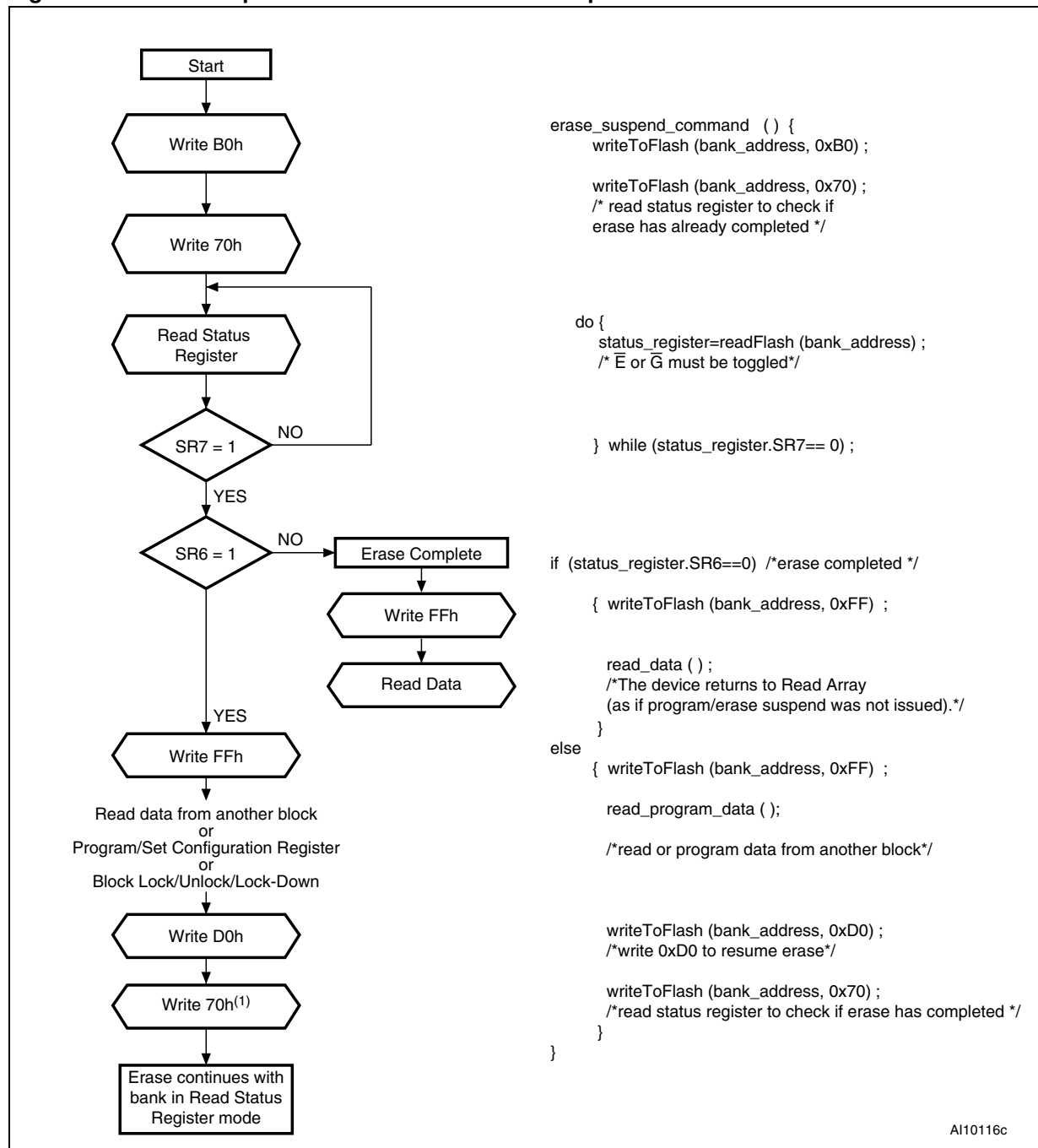
1. The Read Status Register command (Write 70h) can be issued just before or just after the Program Resume command.

Figure 27. Block erase and EFA block erase flowchart and pseudocode



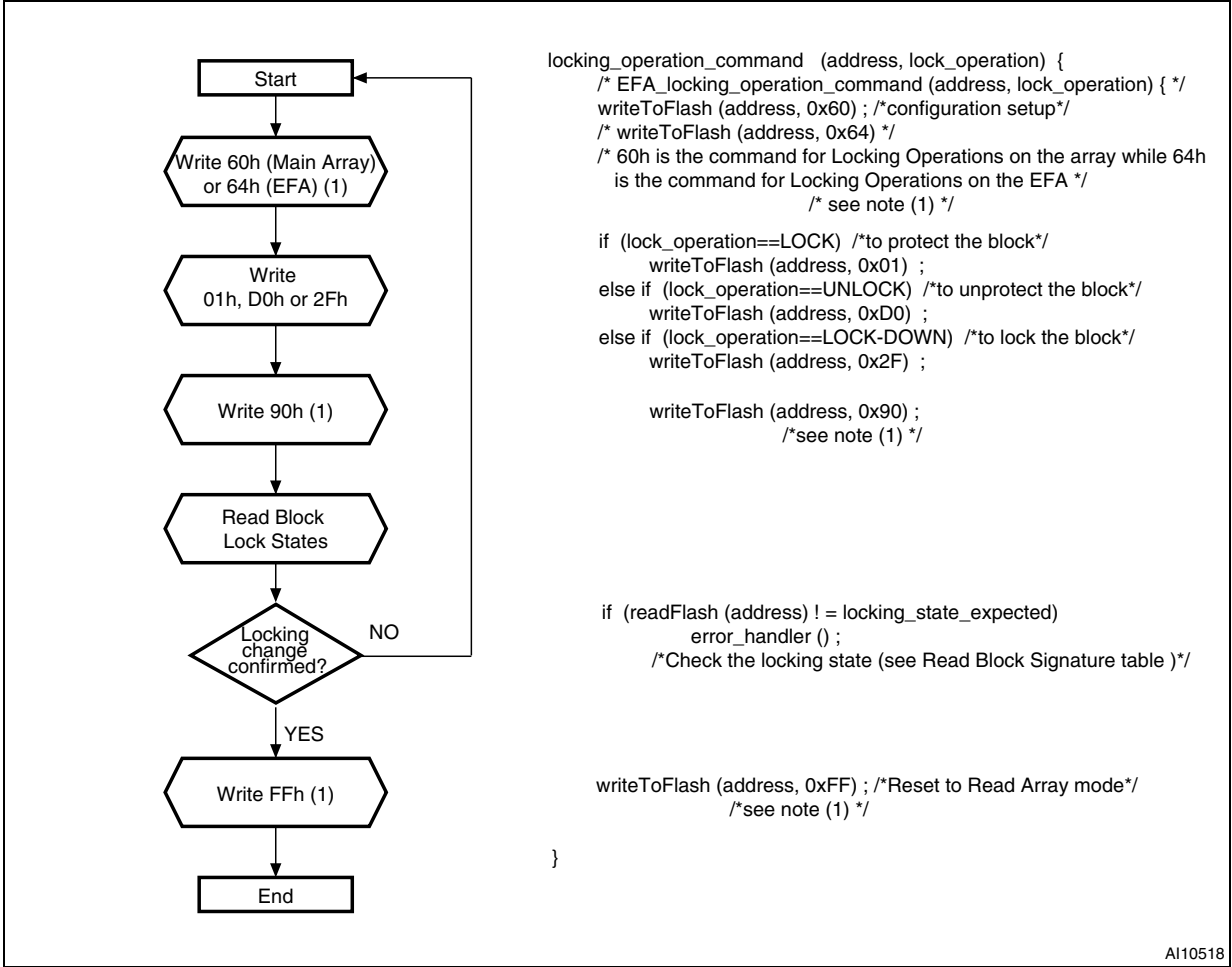
1. Any address within the bank can equally be used.
2. If an error is found, the Status Register must be cleared before further program/erase operations.

Figure 28. Erase suspend and resume flowchart and pseudocode



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

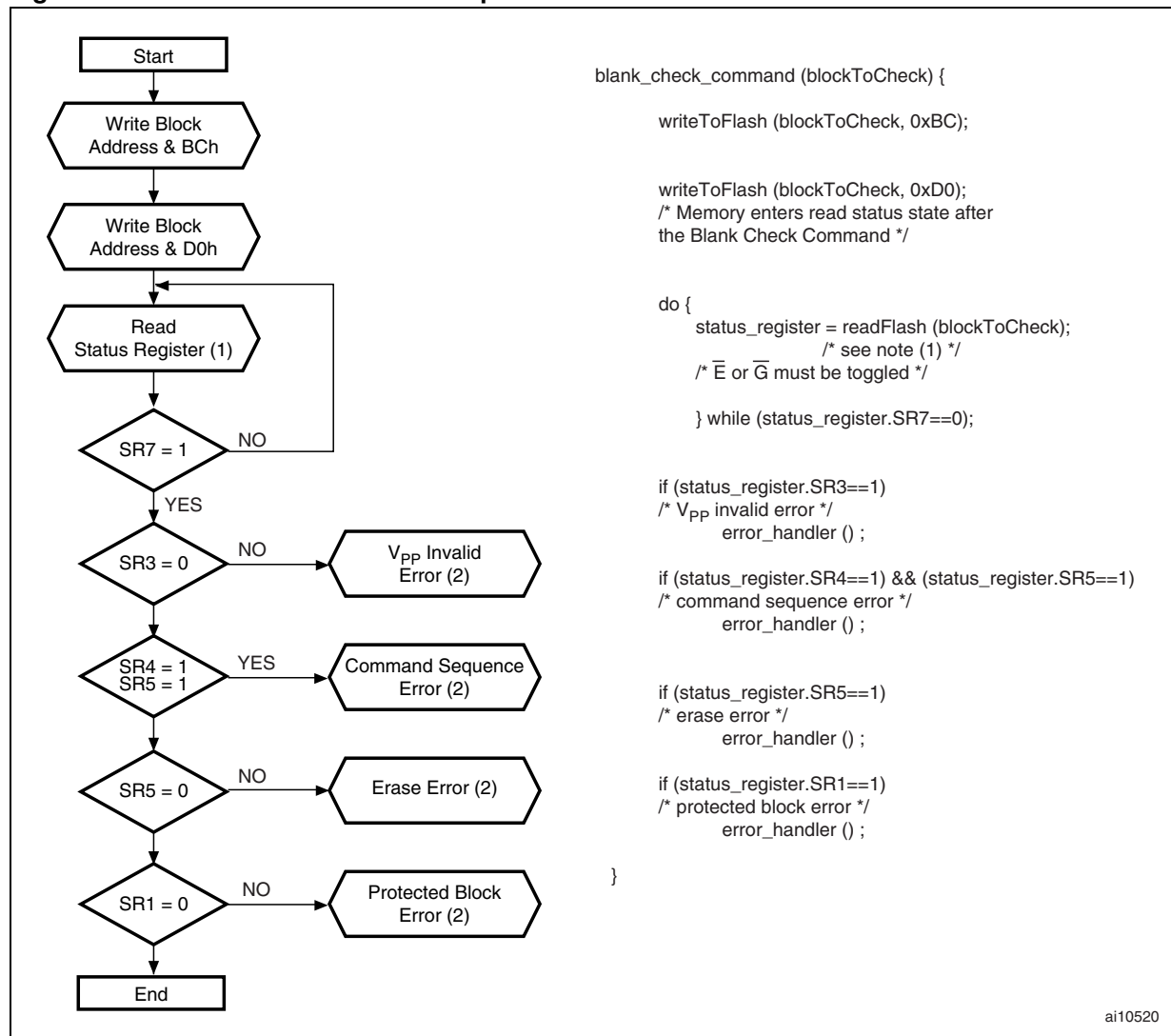
Figure 29. Main array and EFA locking operations flowchart and pseudocode



AI10518

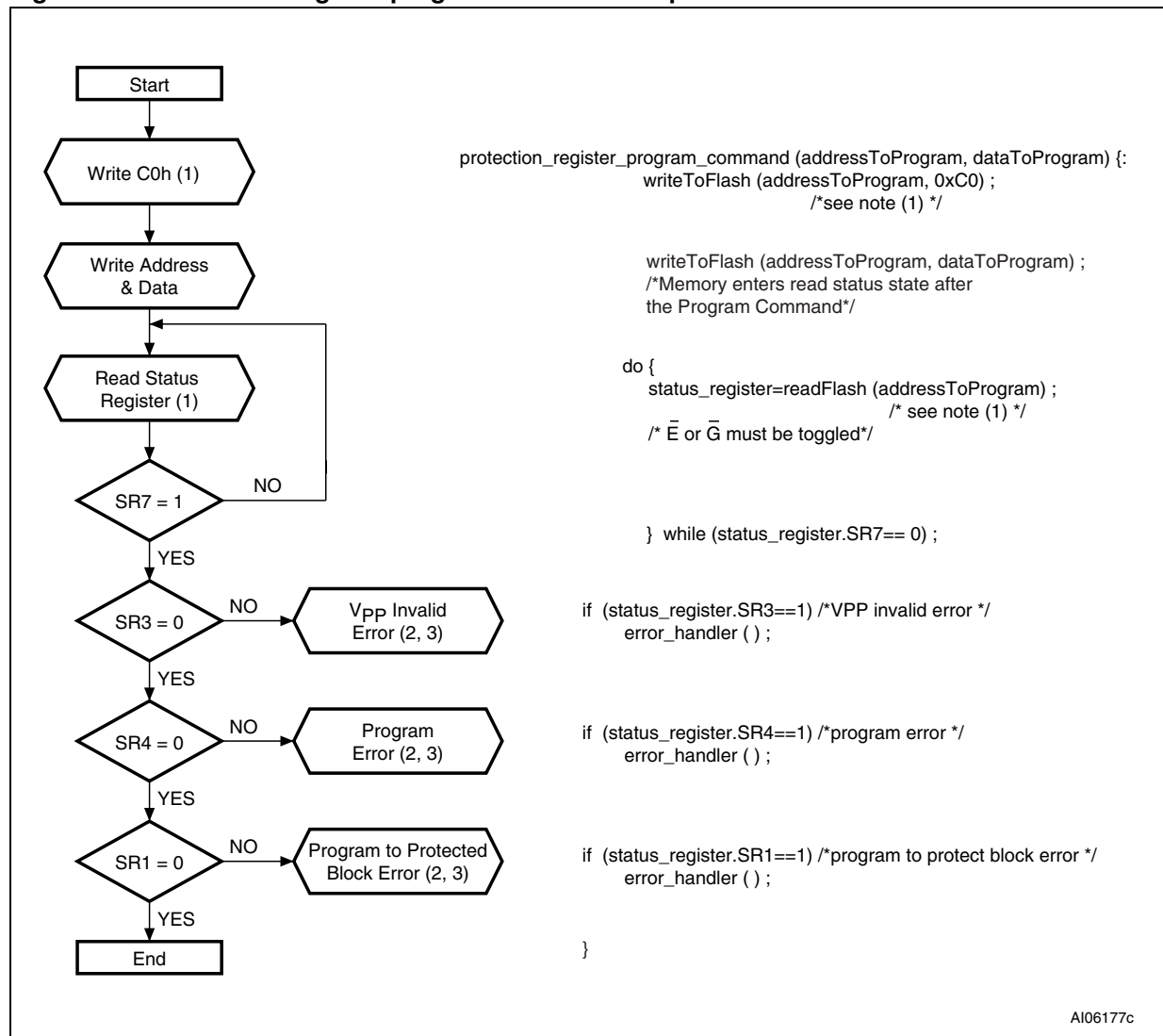
1. Any address within the bank can equally be used.

Figure 30. Blank check flowchart and pseudocode



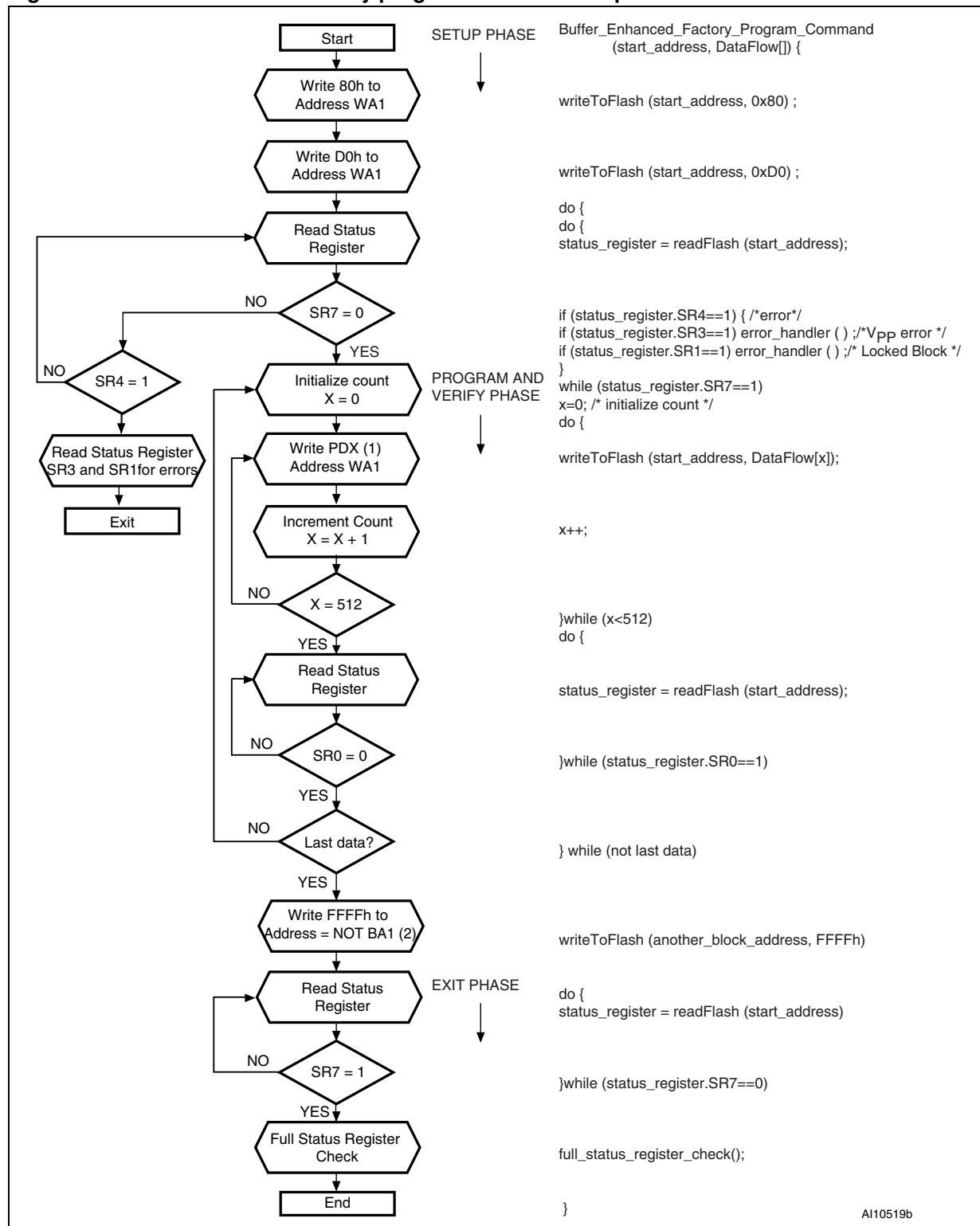
1. Any address within the bank can equally be used.
2. If an error is found, the Status Register must be cleared before further program/erase operations.

Figure 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.

Figure 32. Buffer enhanced factory program flowchart and pseudocode



1. When programming a program region configured in Control Program mode, 'B' half segment addresses (A3 = 1) should not contain '0' values.
2. BA1 = block containing start address WA1.

## Appendix D Command interface state tables

Table 55. Command interface states - modify table, next state 1

| command input to chip and resulting chip next state <sup>(7)</sup> |                           |   |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|--|---------------------------|---|----------|--------------------|----------------------|-------------------------|--------------------|-----------------------|------------|------------------------|------------------------|-------------|
| Current chip state   |                           | Read Array  | Read EFA | Program Setup      | EFA Program Setup    | BP                      | Block Erase Setup  | EFA Block Erase Setup | BEFP       | Confirm, Resume        | Program/ Erase Suspend | Read Status |
|  |                           | FFh   | 94h      | 41h                | 44h                  | E9h <sup>(8,9,10)</sup> | 20h                | 24h                   | 80h        | D0h                    | B0h                    | 70h         |
| ready  |                           | ready   |          | pgrm setup         | EFA block pgrm setup | BP setup                | erase setup        | EFA block erase setup | BEFP setup | ready                  |                        |             |
| Lock/CR/ECR Setup  |                           | ready (sequence error)  |          |                    |                      |                         |                    |                       |            | ready (unlock)         | ready (sequence error) |             |
| Lock EFA Block Setup   |                           |   |          |                    |                      |                         |                    |                       |            |                        |                        |             |
| 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  |          |                    |                      |                         |                    |                       |            |                        |                        |             |
| WP or EFA Block WP   | setup                     | WP busy   |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | busy                      | program busy  |          | IS in program busy |                      | pgrm busy               | IS in program busy |                       |            | program busy           | WP suspend             | WP busy     |
|  | IS in program busy        | WP busy   |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | suspend                   | PS  |          | IS in PS           |                      | pgrm suspend            | IS in PS           |                       |            | program busy           | WP suspend             |             |
|  | IS in PS                  | WP suspend  |          |                    |                      |                         |                    |                       |            |                        |                        |             |
| BP   | setup <sup>(1), (2)</sup> | BP load 1 (give word count load (N-1))                                  |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | BP load 1 <sup>(2)</sup>  | if N=0 go to BP confirm else go to BP load 2 (data load) <sup>(4)</sup> |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | BP load 2 <sup>(2)</sup>  | BP confirm when count =0 else BP load 2                                 |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | BP confirm                | ready (sequence error)  |          |                    |                      |                         |                    |                       |            | BP busy <sup>(3)</sup> | ready (sequence error) |             |
|  | BP busy                   | BP busy   |          | IS in BP busy      |                      | BP busy                 | IS in BP busy      |                       |            | BP busy                | BP suspend             | BP busy     |
|  | IS in BP busy             | BP busy   |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | BP suspend                | BP suspend  |          | IS in BP suspend   |                      | BP suspend              | IS in BP suspend   |                       |            | BP busy                | BP suspend             |             |
|  | IS in BP suspend          | BP suspend  |          |                    |                      |                         |                    |                       |            |                        |                        |             |
| Erase or EFA Block Erase   | setup                     | ready (sequence error)  |          |                    |                      |                         |                    |                       |            | erase busy             | ready (sequence error) |             |
|  | busy                      | erase busy  |          | IS in erase busy   |                      | erase busy              | Is in erase busy   |                       |            | erase busy             | ES                     | erase busy  |
|  | IS in erase busy          | erase busy  |          |                    |                      |                         |                    |                       |            |                        |                        |             |
|  | suspend                   | ES  |          | WP setup in ES     | EFA block WP in ES   | BP setup in ES          | IS in ES           |                       |            | erase busy             | ES                     |             |
|  | IS in ES                  | ES  |          |                    |                      |                         |                    |                       |            |                        |                        |             |

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

| command input to chip and resulting chip next state <sup>(7)</sup> |                              |   |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|--|------------------------------|---|--------------------------------|--------------------------|-------------------|-------------------------|--------------------------|-----------------------|---------------|------------------------------|------------------------|---------------|
| Current chip state   |                              | Read Array  | Read EFA                       | Program Setup            | EFA Program Setup | BP                      | Block Erase Setup        | EFA Block Erase Setup | BEFP          | Confirm, Resume              | Program/ Erase Suspend | Read Status   |
|  |                              | FFh   | 94h                            | 41h                      | 44h               | E9h <sup>(8,9,10)</sup> | 20h                      | 24h                   | 80h           | D0h                          | B0h                    | 70h           |
| WP in ES or EFA Block WP in ES                                     | setup                        | WB busy in ES   |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | busy                         | WP busy in ES   |                                | IS in program busy in ES |                   | WP busy in ES           | IS in program busy in ES |                       |               | WP busy in ES                | WP suspend is ES       | WP busy in ES |
|  | IS in program busy in ES     | WP busy in ES   |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | suspend                      | WP suspend in ES  |                                | IS in PS in ES           |                   | WP suspend in ES        | IS in PS in ES           |                       |               | WP busy in ES                | WP suspend is ES       |               |
|  | IS in PS in ES               | WP suspend in ES  |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
| BP in ES   | setup <sup>(1) (2)</sup>     | BP load 1 in ES (give word count load (N-1))  |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | BP load 1 <sup>(2)</sup>     | if N=0 go to BP confirm in ES else go to BP load 2 in ES (data load) <sup>(4)</sup>                           |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | BP load 2 <sup>(2)</sup>     | BP confirm in ES when count = 0 else BP load 2 in ES  |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | BP confirm                   | ES (sequence error)   |                                |                          |                   |                         |                          |                       |               | BP busy in ES <sup>(3)</sup> | ES (sequence error)    |               |
|  | BP 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             | BP busy in ES          |               |
|  | IS in BP busy in ES          | BP busy in ES   |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
|  | BP suspend                   | BP suspend in ES  | IS in BP suspend in ES suspend |                          | BP suspend in ES  | IS in BP suspend in ES  |                          |                       | BP busy in ES | BP suspend in ES             |                        |               |
|  | IS in BP suspend in ES       | BP suspend in ES  |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
| Lock/CR/ECR/Lock EFA block setup in ES                             |                              | ES (sequence error)   |                                |                          |                   |                         |                          |                       |               | ES (unlock block)            | ES (sequence error)    |               |
| Blank Check  | setup                        | ready (sequence error)  |                                |                          |                   |                         |                          |                       |               | blank check busy             | ready (sequence error) |               |
|  | blank check busy             | BC busy   |                                | IS in BC busy            |                   | BC busy                 | IS in BC busy            |                       |               | BC busy                      |                        |               |
|  | IS in blank check busy       | BC busy   |                                |                          |                   |                         |                          |                       |               |                              |                        |               |
| BEFP mode  | setup                        | ready (sequence error)  |                                |                          |                   |                         |                          |                       |               | BEFP loading data            | ready (sequence error) |               |
|  | BEFP Busy <sup>(5) (6)</sup> | BEFP program and verify busy (in block address given matches on BEFP setup command). Commands treated as data |                                |                          |                   |                         |                          |                       |               |                              |                        |               |

Table 56. Command interface states - modify table, next state 2

| Current chip state                |                          | command input to chip and resulting chip next state      |               |                                |                                     |                         |              |                          |                          |                              |  | WSM<br>Operation<br>Complete |
|-----------------------------------|--------------------------|--|---------------|--------------------------------|-------------------------------------|-------------------------|--------------|--------------------------|--------------------------|------------------------------|--|------------------------------|
|                                   |                          | Clear SR   | Read ID       | Lock, CR,<br>ECR<br>Setup      | Lock EFA<br>Setup                   | Blank<br>Check<br>Setup | OTP<br>Setup | Block<br>Lock<br>Confirm | Lock-<br>down<br>Confirm | Write<br>CR/ECR<br>Confirm   | Illegal<br>Cmd or<br>BEFP<br>Data                    |                              |
|                                   |                          | 50h  | 90h 98h       | 60h                            | 64h                                 | BCh                     | C0h          | 01h                      | 2Fh                      | 03h 04h                      | others   |                              |
| ready                             |                          | ready  |               | lock/CR/E<br>CR setup          | lock EFA<br>block<br>setup          | blank<br>check<br>setup | OTP<br>setup | ready                    |                          |                              | ready  | N/A                          |
| Lock/CR/ECR Setup                 |                          | ready (sequence error)                                   |               |                                |                                     |                         |              | ready<br>(lock)          | ready<br>(lock<br>down)  | ready (set<br>CR)            | ready<br>(sequence<br>error)                         |                              |
| Lock EFA Block Setup              |                          |  |               |                                |                                     |                         |              | ready<br>(EFA lock)      |                          | ready<br>(sequence<br>error) |  |                              |
| OTP                               | setup                    | OTP busy   |               |                                |                                     |                         |              |                          |                          |                              |  | IS ready                     |
|                                   | busy                     | OTP busy   |               | IS in OTP busy                 |                                     |                         | OTP busy     |                          |                          |                              |  |                              |
|                                   | IS in OTP<br>busy        | OTP busy   |               |                                |                                     |                         |              |                          |                          |                              |  |                              |
| WP or<br>EFA<br>Block<br>WP       | setup                    | WP busy  |               |                                |                                     |                         |              |                          |                          |                              | WP busy  | N/A                          |
|                                   | busy                     | WP busy  |               | IS in program busy             |                                     |                         | IS           | WP busy                  |                          |                              |  | ready                        |
|                                   | IS in<br>program<br>busy | WP busy  |               |                                |                                     |                         |              |                          |                          |                              |  | IS ready                     |
|                                   | suspend                  | WP<br>suspend<br>(error bits<br>cleared)                 | WP<br>suspend | IS in PS                       |                                     |                         | WP suspend   |                          |                          |                              | N/A  |                              |
|                                   | IS in PS                 | WP suspend   |               |                                |                                     |                         |              |                          |                          |                              |  |                              |
| BP                                | setup                    | BP load 1 (give word count load (N-1))                   |               |                                |                                     |                         |              |                          |                          |                              |  | N/A                          |
|                                   | BP load 1                | if N=0 go to BP confirm else go to BP load 2 (data load) |               |                                |                                     |                         |              |                          |                          |                              |  |                              |
|                                   | BP load 2                | BP confirm when count =0 else BP load 2                  |               |                                |                                     |                         |              |                          |                          |                              | BP confirm<br>when<br>count = 0<br>else BP<br>load 2 |                              |
|                                   | BP confirm               | ready (sequence error)                                   |               |                                |                                     |                         |              |                          |                          |                              |  |                              |
|                                   | BP busy                  | BP busy  |               | IS in BP busy                  |                                     |                         | IS           | BP busy                  |                          |                              |  | ready                        |
|                                   | IS in BP<br>busy         | BP busy  |               |                                |                                     |                         |              |                          |                          |                              |  | IS ready                     |
|                                   | BP suspend               | BP<br>suspend<br>(error bits<br>cleared)                 | BP<br>suspend | IS in BP suspend               |                                     |                         | BP suspend   |                          |                          |                              | N/A  |                              |
|                                   | IS in BP<br>suspend      | BP suspend   |               |                                |                                     |                         |              |                          |                          |                              |  |                              |
| Erase or<br>EFA<br>Block<br>Erase | setup                    | ready (sequence error)                                   |               |                                |                                     |                         |              |                          |                          |                              |  | N/A                          |
|                                   | busy                     | erase busy   |               | IS in erase busy               |                                     |                         | IS           | erase busy               |                          |                              |  | ready                        |
|                                   | IS in erase<br>busy      | erase busy   |               |                                |                                     |                         |              |                          |                          |                              |  | IS non-<br>ready             |
|                                   | suspend                  | ES (error<br>bits<br>cleared)                            | ES            | Lock/CR/E<br>CR setup<br>in ES | lock EFA<br>block<br>setup in<br>ES | IS                      | ES           |                          |                          |                              | N/A  |                              |
|                                   | IS in ES                 | ES   |               |                                |                                     |                         |              |                          |                          |                              |  |                              |

Table 56. Command interface states - modify table, next state 2 (continued)

| Current chip state             |                          | command input to chip and resulting chip next state   |                  |                        |                |                   |                  |                    |                     |                      |  | WSM Operation Complete |
|--------------------------------|--------------------------|---|------------------|------------------------|----------------|-------------------|------------------|--------------------|---------------------|----------------------|--|------------------------|
|                                |                          | Clear SR  | Read ID          | Lock, CR, ECR Setup    | Lock EFA Setup | Blank Check Setup | OTP Setup        | Block Lock Confirm | Lock-down Confirm   | Write CR/ECR Confirm | Illegal Cmd or BEFP Data                 |                        |
|                                |                          | 50h   | 90h 98h          | 60h                    | 64h            | BCh               | C0h              | 01h                | 2Fh                 | 03h 04h              | others                                   |                        |
| WP in ES or EFA Block WP in ES | setup                    | WP busy in ES   |                  |                        |                |                   |                  |                    |                     |                      |  | N/A                    |
|                                | busy                     | WP busy in ES   |                  | IS                     |                |                   | WP busy in ES    |                    |                     |                      | ES                                       |                        |
|                                | IS in program busy in ES | WP busy in ES   |                  |                        |                |                   |                  |                    |                     |                      | WP busy in ES                            | IS in ES               |
|                                | suspend                  | WP suspend is ES (error bits cleared)   | WP suspend in ES | IS in WP suspend in ES |                |                   | WP suspend in ES |                    |                     |                      | N/A                                      |                        |
|                                | IS in PS in ES           | WP suspend in ES  |                  |                        |                |                   |                  |                    |                     |                      |  | WP suspend in ES       |
| BP in ES                       | setup                    | BP load 1 in ES (give word count load (N-1))  |                  |                        |                |                   |                  |                    |                     |                      |  | N/A                    |
|                                | BP load 1                | if N=0 go to BP confirm in ES else go to BP load 2 in ES (data load)  |                  |                        |                |                   |                  |                    |                     |                      |  |                        |
|                                | BP load 2                | BP confirm in ES when count = 0 else BP load 2 in ES  |                  |                        |                |                   |                  |                    |                     |                      | BP confirm when count = 0 else BP load 2 |                        |
|                                | BP confirm               | ready (sequence error) in ES  |                  |                        |                |                   |                  |                    |                     |                      |  |                        |
|                                | BP busy                  | BP busy in ES   |                  | IS in BP busy in ES    |                |                   | IS               | BP busy in ES      |                     |                      | ES                                       |                        |
|                                | IS in BP busy in ES      | BP busy in ES   |                  |                        |                |                   |                  |                    |                     |                      |  | IS in ES               |
|                                | BP suspend               | BP suspend in ES (error bits cleared)   | BP suspend is ES | IS in BP suspend in ES |                |                   | BP suspend in ES |                    |                     |                      | N/A                                      |                        |
|                                | IS in BP suspend in ES   | BP suspend in ES  |                  |                        |                |                   |                  |                    |                     |                      |  | BP suspend in ES       |
| Lock/CR/ECR setup in ES        |                          | ES (sequence error)   |                  |                        |                |                   | ES (lock block)  | ES (lock down)     | ES                  | ES (lock error)      | N/A                                      |                        |
| Lock EFA block setup in ES     |                          |   |                  |                        |                |                   |                  |                    | ES (sequence error) |                      |  |                        |
| Blank Check                    | setup                    | ready (sequence error)  |                  |                        |                |                   |                  |                    |                     |                      |  | ready (error)          |
|                                | blank check busy         | BC busy   |                  | IS in BC busy          |                |                   | IS               | BC busy            |                     |                      | ready                                    |                        |
|                                | IS in blank check busy   | BC busy   |                  |                        |                |                   |                  |                    |                     |                      | BC busy                                  | IS ready               |
| BEFP Mode                      | setup                    | ready (sequence error)  |                  |                        |                |                   |                  |                    |                     |                      |  | N/A                    |
|                                | BEFP Busy                | BEFP program and verify busy (in block address given matches on BEFP setup command). Commands treated as data |                  |                        |                |                   |                  |                    |                     |                      | BEFP busy                                | ready                  |

Table 57. Command interface states - modify table, next output 1

| Current chip state   | command input to chip and resulting chip next state |                 |             |                       |                              |             |                       |             |                              |                        |             |
|--|---|-----------------|-------------|-----------------------|------------------------------|-------------|-----------------------|-------------|------------------------------|------------------------|-------------|
|  | Read Array  | Read EFA Block  | WP          | EFA WP                | BP                           | Erase Setup | EFA Block Erase Setup | BEFP Setup  | Confirm, Resume              | Program/ Erase Suspend | Read Status |
|  | FFh   | 94h             | 41h         | 44h                   | E9h                          | 20h         | 24h                   | 80h         | D0h                          | B0h                    | 70h         |
| BEFP Setup, BEFP program and verify busy, Erase Setup, Erase EFA setup<br>OTP Setup, BP Confirm, WP setup, WP setup in ES, BP confirm in ES, blank check setup | status read   |                 |             |                       |                              |             |                       |             |                              |                        |             |
| EFA block WP setup, EFA block program setup in ES  | EFA block status read                               |                 |             |                       |                              |             |                       |             |                              |                        |             |
| Lock/CR/ECR setup, Lock/CR/ECR setup in ES   | status read   |                 |             |                       |                              |             |                       |             |                              |                        |             |
| EFA block lock setup, EFA block lock setup in ES   | EFA block status read                               |                 |             |                       |                              |             |                       |             |                              |                        |             |
| OTP busy   | read array  | read EFA blocks | status read | EFA block status read | output state does not change | status read | EFA block status read | status read | output state does not change | status read            |             |
| ready, ES, BP suspend, WP busy, erase busy, BP busy, BP busy in ES, WP suspend, WP busy in ES, PS in ES, BP suspend in ES, Blank Check busy                    |   |                 |             |                       |                              |             |                       |             |                              |                        |             |
| BP setup, BP load 1, BP load 2, IS   | output state does not change                        |                 |             |                       |                              |             |                       |             |                              |                        |             |

Table 58. Command interface states - modify table, next output 2

| Current Chip State  | command input to chip and resulting chip next state |         |                     |                       |             |                              |              |                   |                              |                          |
|---|---|---------|---------------------|-----------------------|-------------|------------------------------|--------------|-------------------|------------------------------|--------------------------|
|   | Clear SR  | Read ID | Lock, CR, ECR Setup | Lock EFA Setup        | Blank Check | OTP Setup                    | Lock Confirm | Lock-down Confirm | Write CR/ ECR Confirm        | Illegal cmd or BEFP Data |
|   | 50h   | 90h 98h | 60h                 | 64h                   | BCh         | C0h                          | 01h          | 2Fh               | 03h 04h                      | others                   |
| BEFP Setup, BEFP program and verify busy, Erase Setup, Erase EFA setup OTP Setup, BP Confirm, WP setup, WP setup in ES, BP confirm in ES, blank check setup | status read   |         |                     |                       |             |                              |              |                   |                              |                          |
| EFA block WP setup, EFA block program setup in ES   | EFA block status read                               |         |                     |                       |             |                              |              |                   |                              |                          |
| Lock/CR/ECR setup, Lock/CR/ECR setup in ES  | status read   |         |                     |                       |             |                              |              |                   | array read                   | status read              |
| EFA block lock setup, EFA block lock setup in ES  | EFA block status read                               |         |                     |                       |             |                              |              |                   | array read                   | EFA block status read    |
| OTP busy  | output state does not change                        | ID read | status read         | EFA block status read | status read | output state does not change |              |                   | output state does not change |                          |
| ready, ES, BP suspend, WP busy, erase busy, BP busy, blank check busy, BP busy in ES, WP suspend, WP busy in ES, PS in ES, BP suspend in ES                 |   |         |                     |                       |             |                              |              |                   |                              |                          |
| BP setup, BP load 1, BP load 2, IS  | output state does not change                        |         |                     |                       |             |                              |              |                   |                              |                          |

- Note: 1 WP = Word Program, BP = Buffer program, cmd = command, SR = Status Register, pgm = program, IS = Illegal state, PS = Program suspend, ES = Erase suspend, CI = Command Interface, CR = Configuration Register, BEFP = Buffer Enhanced Factory Program, P/E. C. = Program/Erase Controller, WA0 = Address in a block different from first BEFP address, ECR = Enhanced Configuration Register.
- 2 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.
- 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 Clear Status Register command clears the Status Register error bits except when the P/EC is busy or suspended.
- 5 BEFP is allowed only when Status Register bit SR0 is reset to '0'. BEFP is busy if the block address is the first BEFP address. Any other commands are treated as data.

- 6 *BEFP aborts when the block address is different from the first block address and data are FFFFh.*
- 7 *BEFP Exit when block address is different from first block address and data are FFFFh.*
- 8 *During BP setup, while entering the number of words to be programmed and filling the buffer, the read status of the partition does not change.*
- 9 *The BP confirm command changes the read status of the partition to Status Read.*
- 10 *Illegal commands are commands not defined in the command set.*

## 18 Revision history

**Table 59. Document revision history**

| Date        | Revision | Changes   |
|-------------|----------|---|
| 28-Apr-2006 | 0.1      | Initial release.  |
| 15-May-2006 | 0.2      | 1Gb density (M58PR001LE part number) added.<br>V <sub>PP</sub> range for application environment changed in <a href="#">Table 25: Operating and AC measurement conditions</a> .<br>I <sub>PP1</sub> unit changed in <a href="#">Table 27: DC characteristics - currents</a> .   |
| 14-Nov-2006 | 1        | Document status promoted from Target Specification to Preliminary Data.<br>Address lines modified in <a href="#">Figure 13: Asynchronous page read AC waveforms</a> . V <sub>PP</sub> max value modified in <a href="#">Table 24: Absolute maximum ratings</a> . Small text changes.  |
| 06-Sep-2007 | 2        | Modified <a href="#">Section 4.18: Set Enhanced Configuration Register command</a> and <a href="#">Section 4.22: Suspend EFA Block command</a> .<br>Updated <a href="#">Table 23: Program/erase times and endurance cycles</a> , <a href="#">Table 24: Absolute maximum ratings</a> , <a href="#">Table 27: DC characteristics - currents</a> , <a href="#">Table 29: Asynchronous read AC characteristics</a> , and <a href="#">Table 31: Write AC characteristics, write enable controlled</a> .<br>Added t <sub>LLTV</sub> timing in <a href="#">Table 30: Synchronous read AC characteristics</a> and <a href="#">Figure 15: Single synchronous read AC waveforms</a> .<br>Modified <a href="#">Figure 28: Erase suspend and resume flowchart and pseudocode</a> .<br>Document status promoted from Preliminary Data to Datasheet.    |
| 19-Nov-2007 | 3        | Added the TFBGA105 (ZAD) and TFBGA107 (ZAC) packages to the document, most specifically in <a href="#">Figure 2: TFBGA105 connections (top view through package)</a> , <a href="#">Figure 3: TFBGA107 connections (top view through package)</a> , <a href="#">Figure 22: TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline</a> , <a href="#">Figure 23: TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package outline</a> , <a href="#">Table 35: TFBGA105 9 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, mechanical data</a> , and <a href="#">Table 36: Stacked TFBGA107 8 × 11 mm - 9 × 12 active ball array, 0.8 mm pitch, package mechanical data</a> . Changed the maximum I <sub>DD7</sub> value for "program/erase in one bank, asynchronous read in another bank" from 80 to 85. |

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2007 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)