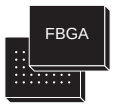
# M58WR064HT
# M58WR064HB

## 64 Mbit (4 Mb x16, multiple bank, burst) 1.8 V supply Flash memories

## Features

- Supply voltage
  - $V_{DD}$ = 1.7 V to 2 V for program, erase and read
  - $V_{DDQ}$ = 1.7 V to 2.24 V for I/O buffers
  - $V_{PP}$ = 12 V for fast program (optional)
- Synchronous/asynchronous read
  - Synchronous burst read mode: 66 MHz
  - Asynchronous/synchronous page read mode
  - Random access: 60 ns, 70 ns
- Synchronous burst read suspend
- Programming time
  - 8 µs by word typical for fast factory program
  - Double/quadruple word program option
  - Enhanced factory program options
- Memory blocks
  - Multiple bank memory array: 4 Mbit banks
  - Parameter blocks (top or bottom location)
- Dual operations
  - Program erase in one bank while read in others
  - No delay between read and write operations
- Block locking
  - All blocks locked at power-up
  - Any combination of blocks can be locked
  - $\overline{WP}$ for block lock-down
- Security
  - 128 bit user programmable OTP cells
  - 64 bit unique device number
- Common Flash interface (CFI)
- 100 000 program/erase cycles per block




VFBGA56 (ZB)
7.7 x 9 mm

- Electronic signature
  - Manufacturer code: 20h
  - Device codes:
    M58WR064HT (top): 8810h
    M58WR064HB (bottom): 8811h
- Package
  - ECOPACK®

# Contents

# List of tables

# List of figures

# 1 Description

The M58WR064HT and M58WR064HB are 64 Mbit (4 Mbit x16) non-volatile Flash memories, and are collectively referred to in this document as the M58WR064HT/B. It can be erased electrically at block level and programmed in-system on a word-by-word basis using a 1.7 V to 2 V $V_{DD}$ supply for the circuitry and a 1.7 V to 2.24 V $V_{DDQ}$ supply for the Input/output pins. An optional 12 V $V_{PP}$ power supply is provided to speed up customer programming.

The device features an asymmetrical block architecture.

The M58WR064HT/B has an array of 135 blocks, and is divided into 4 Mbit banks. There are 15 banks each containing 8 main blocks of 32 KWords, and one parameter bank containing 8 parameter blocks of 4 KWords and 7 main blocks of 32 KWords.

The multiple bank architecture allows dual operations. While programming or erasing in one bank, read operations are possible in other banks. Only one bank at a time is allowed to be in program or erase mode. It is possible to perform burst reads that cross bank boundaries. The bank architecture is summarized in *Table 2* and the memory map is shown in *Figure 3*. The parameter blocks are located at the top of the memory address space for the M58WR064HT, and at the bottom for the M58WR064HB.

Each block can be erased separately. Erase can be suspended to program in any other block, and then resumed. Program can be suspended to read data in any other block, and then resumed. Each block can be programmed and erased over 100 000 cycles using the supply voltage $V_{DD}$. There are two enhanced factory programming commands available to speed up programming.

Program and erase commands are written to the command interface of the memory. An internal Program/Erase Controller manages the timings necessary for program and erase operations. The end of a program or erase operation can be detected and any error conditions are 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 mode, data is output on each clock cycle at frequencies of up to 66 MHz. The synchronous burst read operation can be suspended and resumed.

The device features an automatic standby mode. When the bus is inactive during asynchronous read operations, the device automatically switches to the automatic standby mode. In this condition the power consumption is reduced to the standby value $I_{DD4}$ and the outputs are still driven.

The M58WR064HT/B features an instant, individual block locking scheme that allows any block to be locked or unlocked with no latency, enabling instant code and data protection. All blocks have three levels of protection. They can be locked and locked-down individually preventing any accidental programming or erasure. There is additional hardware protection against program and erase. When $V_{PP} \leq V_{PPLK}$ all blocks are protected against program or erase. All blocks are locked at power- up.

The device includes a Protection Register to increase the protection of a system's design. The Protection Register is divided into two segments: a 64 bit segment containing a unique device number written by ST, and a 128 bit segment one-time-programmable (OTP) by the user. The user programmable segment can be permanently protected. *Figure 4* shows the Protection Register memory map.

The memory is offered in a VFBGA56, 7.7 x 9 mm, 8 x 7 active ball array, 0.75 mm pitch package. It is supplied with all the bits erased (set to '1').

**Figure 1. Logic diagram**

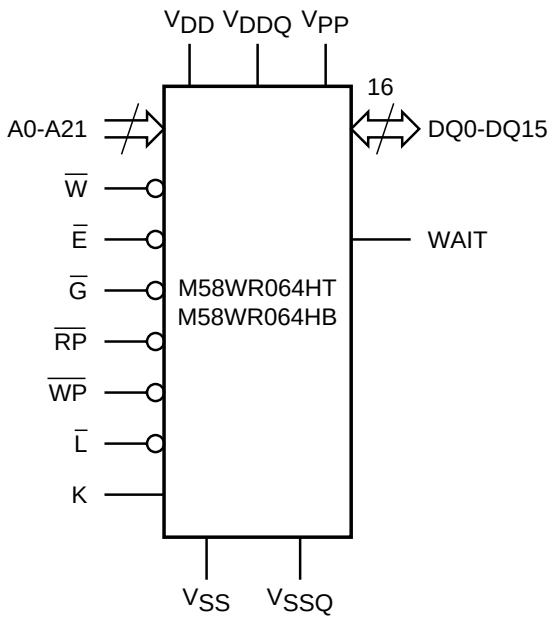



**Table 1. Signal names**

| Name | Function |
|---|---|
| A0-A21 | Address inputs |
| DQ0-DQ15 | Data input/outputs, command inputs |
| $\overline{E}$ | Chip Enable |
| $\overline{G}$ | Output Enable |
| $\overline{W}$ | Write Enable |
| $\overline{RP}$ | Reset |
| $\overline{WP}$ | Write Protect |
| K | Clock |
| $\overline{L}$ | Latch Enable |
| WAIT | Wait |
| $V_{DD}$ | Supply voltage |
| $V_{DDQ}$ | Supply voltage for input/output buffers |
| $V_{PP}$ | Optional supply voltage for fast program and erase |
| $V_{SS}$ | Ground |
| $V_{SSQ}$ | Ground input/output supply |
| NC | Not connected internally |

**Figure 2. VFBGA package connections (top view through package)**

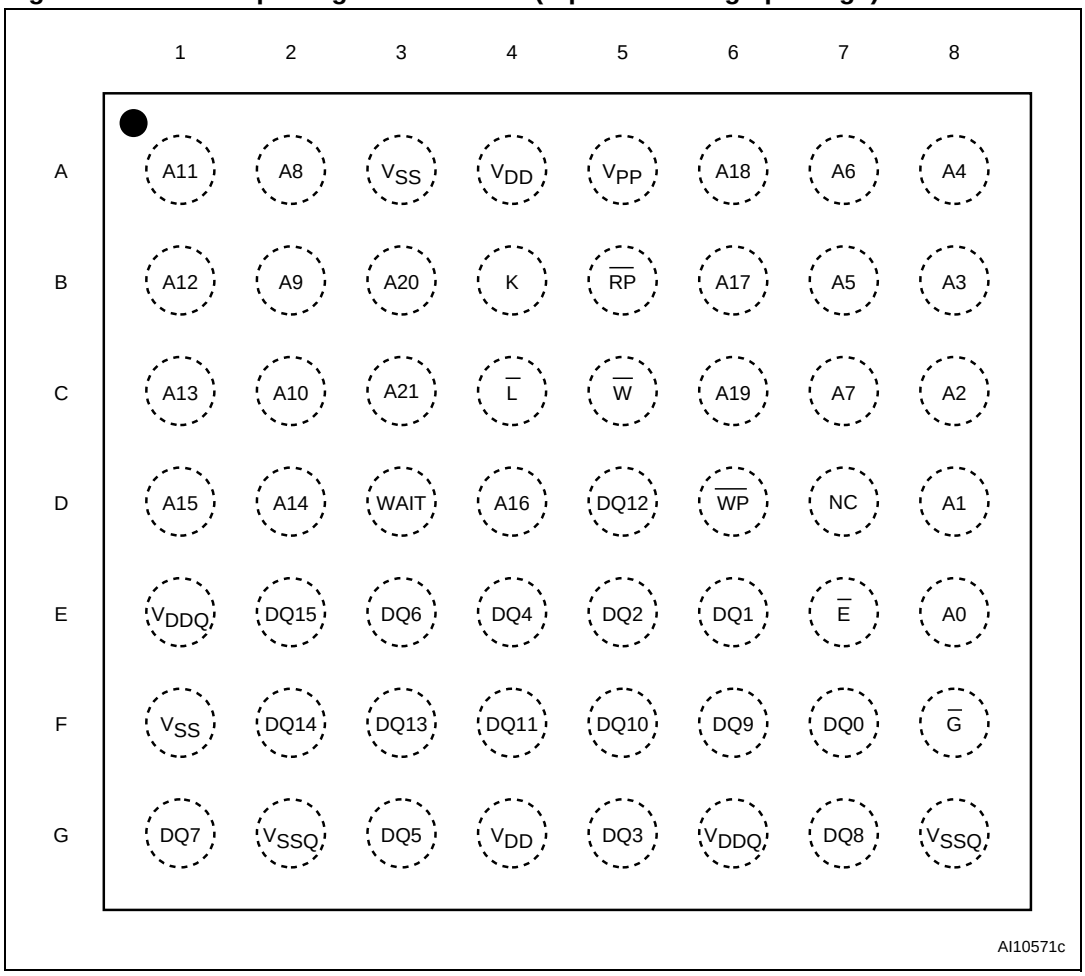
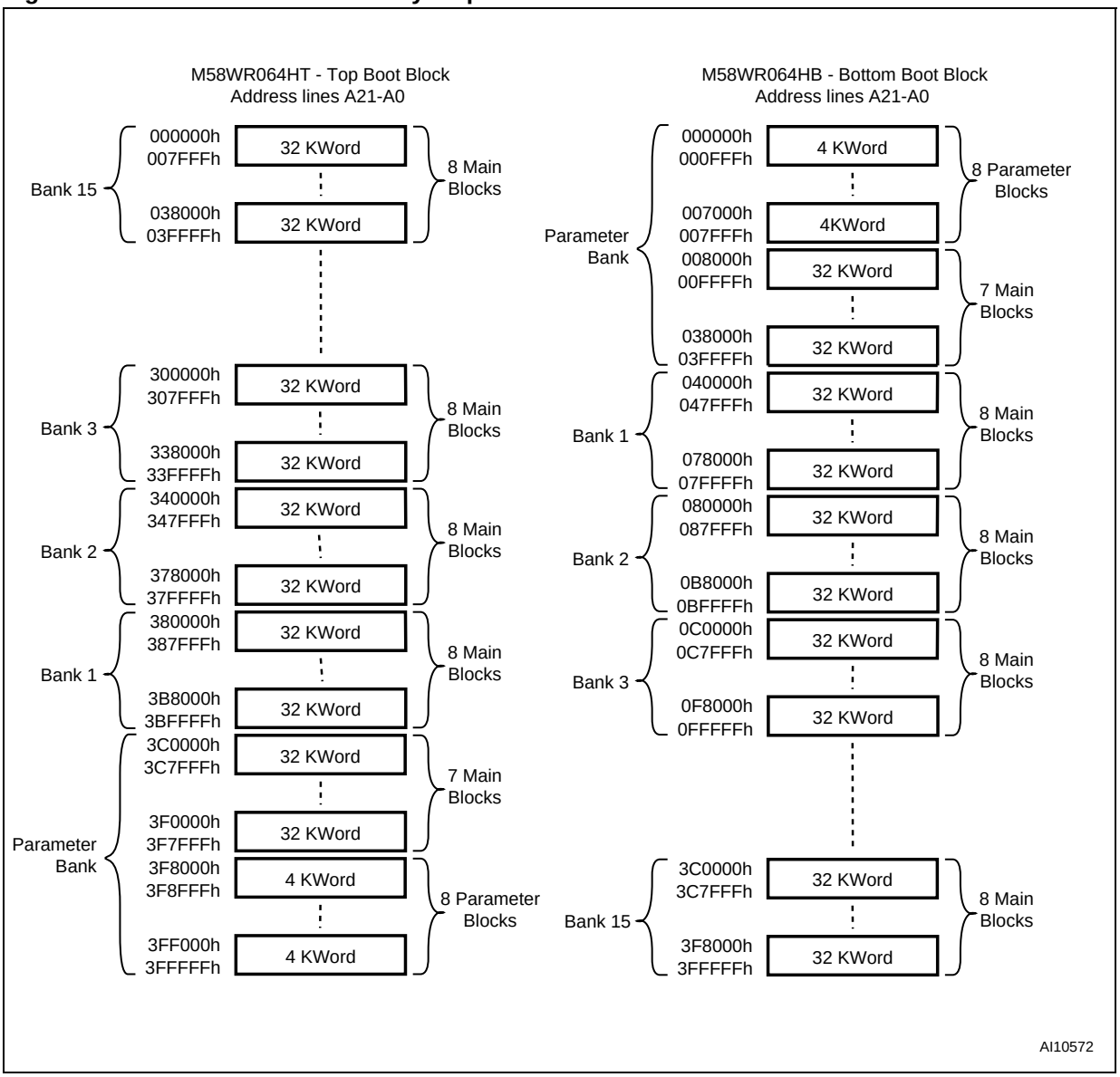



**Table 2. M58WR064HT/B bank architecture**

| Number | Bank size | Parameter blocks | Main blocks |
|---|---|---|---|
| Parameter bank | 4 Mbits | 8 blocks of 4 KWords | 7 blocks of 32 KWords |
| Bank 1 | 4 Mbits | - | 8 blocks of 32 KWords |
| Bank 2 | 4 Mbits | - | 8 blocks of 32 KWords |
| Bank 3 | 4 Mbits | - | 8 blocks of 32 KWords |
| ⋮ | ⋮ | ⋮ | ⋮ |
| Bank 14 | 4 Mbits | - | 8 blocks of 32 KWords |
| Bank 15 | 4 Mbits | - | 8 blocks of 32 KWords |

**Figure 3. M58WR064HT/B memory map**



# 2 Signal descriptions

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

## 2.1 Address inputs (A0-A21)

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 input/output (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 provides 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 15: Lock status*).

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

The Reset input provides a hardware reset of the memory. When Reset is at $V_{IL}$, the memory is in reset mode: the outputs are high impedance and the current consumption is reduced to the Reset supply current $I_{DD2}$. Refer to *Table 20: DC characteristics - currents* for the value of $I_{DD2}$. After Reset all blocks are in the locked state and the Configuration Register is reset. When Reset is at $V_{IH}$, the device is in normal operation. Upon 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 $V_{RPH}$ (refer to *Table 21: DC characteristics - voltages*).

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

Latch Enable latches the address bits on its rising edge. The address latch is transparent when Latch Enable is at $V_{IL}$ and it is inhibited when Latch Enable is at $V_{IH}$. Latch Enable can be kept Low (also at board level) when the Latch Enable function is not required or supported.

## 2.9 Clock (K)

The clock input synchronizes the memory to the microcontroller during synchronous read operations; the address is latched on a Clock edge (rising or falling, according to the configuration settings) when Latch Enable is at $V_{IL}$. Clock is ‘don't care’ during asynchronous read and in write operations.

## 2.10 Wait (WAIT)

Wait is an output signal used during synchronous read to indicate whether the data on the output bus is valid. This output is high impedance when Chip Enable is at $V_{IH}$ or Reset is at $V_{IL}$. It can be configured to be active during the wait cycle or one clock cycle in advance. The WAIT signal is not gated by Output Enable.

## 2.11 $V_{DD}$ 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).

## 2.12 $V_{DDQ}$ 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.

## 2.13 $V_{PP}$ program supply voltage

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

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

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

## 2.14 $V_{SS}$ ground

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

## 2.15 $V_{SSQ}$ ground

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

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

# 3 Bus operations

There are six standard bus operations that control the device. These are bus read, bus write, address latch, output disable, standby and reset. See *Table 3: Bus operations* for a summary.

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

## 3.1 Bus read

Bus read operations output the contents of the memory array, the electronic signature, the Status Register and the common Flash interface. Both Chip Enable and Output Enable must be at $V_{IL}$ to perform a read operation. The Chip Enable input should be used to enable the device. Output Enable should be used to gate data onto the output. The data read depends on the previous command written to the memory (see command interface section). See Figures *9*, *10*, *11* and *12* Read AC Waveforms, and Tables *22* and *23* Read AC Characteristics for details of when the output becomes valid.

## 3.2 Bus write

Bus write operations write commands to the memory or latch input data to be programmed. A bus write operation is initiated when Chip Enable and Write Enable are at $V_{IL}$ with Output Enable at $V_{IH}$. Commands, input data and addresses are latched on the rising edge of Write Enable or Chip Enable, whichever occurs first. The addresses 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 *15* and *16*, Write AC Waveforms, and Tables *24* and *25*, Write AC Characteristics for details of the timing requirements.

## 3.3 Address latch

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

## 3.4 Output disable

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

## 3.5 Standby

Standby disables most of the internal circuitry allowing a substantial reduction of the current consumption. The memory is in stand-by when Chip Enable and Reset are at $V_{IH}$. The power consumption is reduced to the stand-by level 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 standby level, independently from the Chip Enable, Output Enable, or Write Enable inputs. If Reset is pulled to $V_{SS}$ during a program or erase, this operation is aborted and the memory content is no longer valid.

**Table 3. Bus operations(1)**

| Operation | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{L}$ | $\overline{RP}$ | WAIT(2) | DQ15-DQ0 |
|---|---|---|---|---|---|---|---|
| Bus read | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IL}$(3) | $V_{IH}$ | | Data output |
| Bus write | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | $V_{IL}$(3) | $V_{IH}$ | | Data input |
| Address latch | $V_{IL}$ | X | $V_{IH}$ | $V_{IL}$ | $V_{IH}$ | | Data output or Hi-Z (4) |
| Output disable | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | X | $V_{IH}$ | | Hi-Z |
| Standby | $V_{IH}$ | X | X | X | $V_{IH}$ | Hi-Z | Hi-Z |
| Reset | X | X | X | X | $V_{IL}$ | Hi-Z | Hi-Z |

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

# 4 Command interface

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

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

Refer to *Table 4: Command codes* and *Appendix D*, Tables *41*, *42*, *43* and *44*, command interface States - Modify and Lock Tables for a summary of the command interface.

The command interface is split into two types of commands: standard commands and factory program commands. The following sections explain in detail how to perform each command.

**Table 4. Command codes**

| Hex Code | Command |
|---|---|
| 01h | Block Lock Confirm |
| 03h | Set Configuration Register Confirm |
| 10h | Alternative Program Setup |
| 20h | Block Erase Setup |
| 2Fh | Block Lock-Down Confirm |
| 30h | Enhanced Factory Program Setup |
| 35h | Double Word Program Setup |
| 40h | Program Setup |
| 50h | Clear Status Register |
| 56h | Quadruple Word Program Setup |
| 60h | Block Lock Setup, Block Unlock Setup, Block Lock Down Setup and Set Configuration Register Setup |
| 70h | Read Status Register |
| 75h | Quadruple Enhanced Factory Program Setup |
| 80h | Bank Erase Setup |
| 90h | Read Electronic Signature |
| 98h | Read CFI Query |
| B0h | Program/Erase Suspend |
| C0h | Protection Register Program |
| D0h | Program/Erase Resume, Block Erase Confirm, Bank Erase Confirm, Block Unlock Confirm or Enhanced Factory Program Confirm |
| FFh | Read Array |

# 5 Command interface - standard commands

The following commands are the basic commands used to read, write to and configure the device. Refer to *Table 5: Standard commands* in conjunction with the descriptions in the following sections.

## 5.1 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 and return the addressed bank to read array mode. Subsequent read operations read the addressed location and output the data. A Read Array command can be issued in one bank while programming or erasing in another bank. However, if a Read Array command is issued to a bank currently executing a program or erase operation, the command executes but the output data is not guaranteed.

## 5.2 Read Status Register command

The Status Register indicates when a program or erase operation is complete and the success or failure of operation itself. Issue a Read Status Register command to read the Status Register content. The Read Status Register command can be issued at any time, even during program or erase operations.

The following read operations output the content of the Status Register of the addressed bank. The Status Register is latched on the falling edge of $\overline{E}$ or $\overline{G}$ signals, and can be read until $\overline{E}$ or $\overline{G}$ returns to $V_{IH}$. Either $\overline{E}$ or $\overline{G}$ must be toggled to update the latched data. See *Table 8* for the description of the Status Register bits. This mode supports asynchronous or single synchronous reads only.

## 5.3 Read Electronic Signature command

The Read Electronic Signature command reads the manufacturer and device codes, the block locking status, the Protection Register, and the Configuration Register.

The Read Electronic Signature command consists of one write cycle to an address within one of the banks. A subsequent read operation in the same bank outputs the manufacturer code, the device code, the protection status of the blocks in the targeted bank, the Protection Register, or the Configuration Register (see *Table 6*).

Dual operations between the parameter bank and the electronic signature locations are not allowed (see *Table 14: Dual operation limitations*).

If a Read Electronic Signature command is issued in 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. This mode supports asynchronous or single synchronous reads only; it does not support page mode or synchronous burst reads.

## 5.4 Read CFI Query command

The Read CFI Query command reads data from the common Flash interface (CFI). The Read CFI Query command consists of one bus write cycle, to an address within one of the banks. Once the command is issued subsequent bus read operations in the same bank read from the common Flash interface.

If a Read CFI Query command is issued in 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. This mode supports asynchronous or single synchronous reads only; it does not support page mode or synchronous burst reads.

The status of the other banks is not affected by the command (see *Table 12*). After issuing a Read CFI Query command, a Read Array command should be issued to the addressed bank to return the bank to read array mode.

Dual operations between the parameter bank and the CFI memory space are not allowed (see *Table 14: Dual operation limitations* for details).

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

## 5.5 Clear Status Register command

The Clear Status Register command resets (set to ‘0’) error bits SR1, SR3, SR4 and SR5 in the Status Register. One bus write cycle is required to issue the Clear Status Register command. The Clear Status Register command does not change 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.

## 5.6 Block Erase command

The Block Erase command erases a block. It sets all the bits within the selected block to ’1’. All previous data in the block is lost. If the block is protected then the erase operation aborts, the data in the block does not change, and the Status Register outputs the error. The Block Erase command can be issued at any moment, regardless of whether the block has been programmed or not.

Two bus write cycles are required to issue the command:

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

If the second bus cycle is not Write Erase Confirm (D0h), Status Register bits SR4 and SR5 are set, and the command aborts. The erase aborts if Reset turns to $V_{IL}$. As data integrity cannot be guaranteed when the erase operation is aborted, the block must be erased again.

Once the command is issued the device outputs the Status Register data when any address within the bank is read. At the end of the operation the bank remains in read Status Register mode until a Read Array, Read CFI Query or Read Electronic Signature command is issued.

During erase operations the bank containing the block being erased only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and the Program/Erase Suspend commands; all other commands are ignored. Refer to*Section 10* for detailed information about simultaneous operations allowed in banks not being erased. Typical erase times are given in *Table 16: Program/erase times and endurance cycles*.

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

## 5.7 Program command

The memory array can be programmed word-by-word. Only one word in one bank can be programmed at any one time. If the block is protected, the program operation aborts, the data in the block does not change, and the Status Register outputs the error.

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

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

After programming has started, read operations in the bank being programmed output the Status Register content.

During 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. Refer to *Section 10* for detailed information about simultaneous operations allowed in banks not being programmed. Typical program times are given in *Table 16: Program/erase times and endurance cycles*. Programming aborts if Reset goes to $V_{IL}$. As data integrity cannot be guaranteed when the program operation is aborted, the memory location must be reprogrammed.

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

## 5.8 Program/Erase Suspend command

The Program/Erase Suspend command pauses a program or block erase operation. A bank erase operation cannot be suspended.

One bus write cycle is required to issue the Program/Erase command. Once the Program/Erase Controller has paused, bits SR7, SR6 and/ or SR2 of the Status Register are set to ‘1’. The command can be addressed to any bank.

During program/erase suspend the command interface accepts the Program/Erase Resume, Read Array (cannot read the erase-suspended block or the program-suspended word), Read Status Register, Read Electronic Signature and Read CFI Query commands. In addition, if the suspend operation is erase then the Clear Status Register, Program, Block Lock, Block Lock-Down or Block Unlock commands are also accepted.

The block being erased may be protected by issuing the Block Lock, Block Lock-Down or Protection Register Program commands. Only the blocks not being erased may be read or programmed correctly. When the Program/Erase Resume command is issued the operation completes. Refer to*Section 10* 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 turns to $V_{IL}$.

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

## 5.9 Program/Erase Resume command

The Program/Erase Resume command restarts the Program/Erase Controller after a Program/Erase Suspend command has paused it. One bus write cycle is required to issue the command. The command can be written 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 the bank was in read array mode, subsequent read operations output invalid data.

If a program command is issued during a block erase suspend, then the erase cannot be resumed until the programming operation has completed. It is possible to accumulate suspend operations. For example, it is possible to suspend an erase operation, start a programming operation, suspend the programming operation, and then read the array. See *Appendix C*, *Figure 22: Program suspend and resume flowchart and pseudo code* and *Figure 24: Erase suspend and resume flowchart and pseudo code* for flowcharts for using the Program/Erase Resume command.

## 5.10 Protection Register Program command

The Protection Register Program command programs the 128 bit user OTP segment of the Protection Register and the Protection Register lock. The segment is programmed 16 bits at a time. When shipped, all bits in the segment are set to ‘1’. The user can only program the bits to ‘0’.

Two 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 the data to be written to the Protection Register and starts the Program/Erase Controller.

Read operations output the Status Register content after the programming has started.

The segment can be protected by programming bit 1 of the Protection Lock Register (see *Figure 4: Protection Register memory map*). Attempting to program a previously protected Protection Register results in a Status Register error. The protection of the Protection Register is not reversible. The Protection Register program cannot be suspended. Dual operations between the parameter bank and the Protection Register memory space are not allowed (see *Table 14: Dual operation limitations*)

## 5.11 Set Configuration Register command

The Set Configuration Register command is used to write a new value to the Configuration Register, which defines the burst length, type, X latency, synchronous/asynchronous read mode, and the valid Clock edge configuration.

Two bus write cycles are required to issue the Set Configuration Register command.

- The first cycle writes the setup command and the address corresponding to the Configuration Register content.
- The second cycle writes the Configuration Register data and the confirm command.

Read operations output the memory array content after the Set Configuration Register command is issued.

The value for the Configuration Register is always presented on A0-A15. CR0 is on A0, CR1 on A1, and so on. The other address bits are ignored.

## 5.12 Block Lock command

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

The lock status can be monitored for each block using the Read Electronic Signature command. *Table 15* shows the lock status after issuing a Block Lock command.

The block lock bits are volatile; once set they remain set until a hardware reset or power-down/power-up. They are cleared by a Block Unlock command. Refer to *Section 11: Block locking* for a detailed explanation. See *Appendix C*, *Figure 25: Locking operations flowchart and pseudo code* for a flowchart for using the Lock command.

## 5.13 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.

The lock status can be monitored for each block using the Read Electronic Signature command. *Table 15* shows the protection status after issuing a Block Unlock command. Refer to *Section 11: Block locking* for a detailed explanation and *Appendix C*, *Figure 25: Locking operations flowchart and pseudo code* for a flowchart for using the Unlock command.

## 5.14 Block Lock-Down command

A locked or unlocked block can be locked down by issuing the Block Lock-Down command. A locked-down block cannot be programmed or erased, or have its protection status 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 command.
- The second bus write cycle latches the block address.

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 15* shows the lock status after issuing a Block Lock-Down command. Refer to *Section 11: Block locking* for a detailed explanation and *Appendix C*, *Figure 25: Locking operations flowchart and pseudo code* for a flowchart for using the Lock-Down command.

**Table 5. Standard commands**

| Commands | Cycles | Bus operations(1) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1st cycle | | | 2nd cycle | | |
| | | Operation | Address | Data | Op. | Add | Data |
| Read Array | 1+ | Write | BKA | FFh | Read | WA | RD |
| Read Status Register | 1+ | Write | BKA | 70h | Read | BKA(2) | SRD |
| Read Electronic Signature | 1+ | Write | BKA | 90h | Read | BKA(2) | ESD |
| Read CFI Query | 1+ | Write | BKA | 98h | Read | BKA(2) | QD |
| Clear Status Register | 1 | Write | X | 50h | | | |
| Block Erase | 2 | Write | BKA or BA(3) | 20h | Write | BA | D0h |
| Program | 2 | Write | BKA or WA(3) | 40h or 10h | Write | WA | PD |
| 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(3) | 60h | Write | BA | 01h |
| Block Unlock | 2 | Write | BKA or BA(3) | 60h | Write | BA | D0h |
| Block Lock-Down | 2 | Write | BKA or BA(3) | 60h | Write | BA | 2Fh |

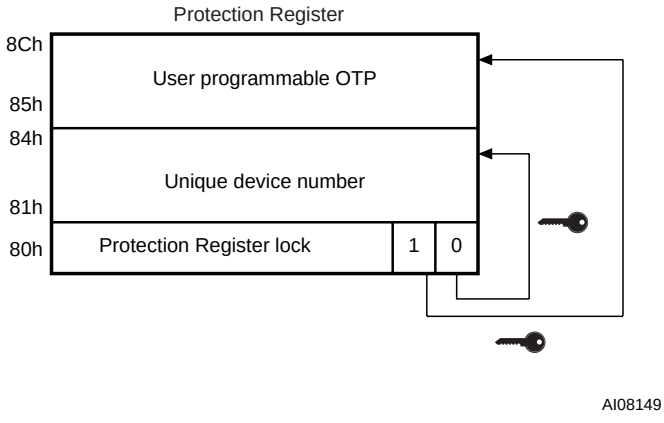
1. X = Don't Care, WA = Word Address in targeted bank, RD = Read Data, SRD = Status Register Data, ESD = Electronic Signature Data, QD = Query Data, BA = Block Address, BKA = Bank Address, PD = Program Data, PRA = Protection Register Address, PRD = Protection Register Data, CRD = Configuration Register Data.
2. Must be same bank as in the first cycle. The signature addresses are listed in *Table 6*.
3. Any address within the bank can be used.

**Table 6. Electronic signature codes**

| Code | | Address (h) | Data (h) |
|---|---|---|---|
| Manufacturer code | | Bank address + 00 | 0020 |
| Device code | Top (M58WR064HT) | Bank address + 01 | 8810 |
| | Bottom (M58WR064HB) | Bank address + 01 | 8811 |
| Block protection | Locked | Block address + 02 | 0001 |
| | Unlocked | | 0000 |
| | Locked and locked-down | | 0003 |
| | Unlocked and locked-down | | 0002 |
| Reserved | | Bank address + 03 | Reserved |
| Configuration Register | | Bank address + 05 | CR(1) |
| Protection Register lock | ST factory default | Bank address + 80 | 0002 |
| | OTP area permanently locked | | 0000 |
| Protection Register | | Bank address + 81<br>Bank address + 84 | Unique Device Number |
| | | Bank address + 85<br>Bank address + 8C | OTP Area |

1. CR = Configuration Register.

**Figure 4. Protection Register memory map**



# 6 Command interface - factory program commands

The factory program commands are used to speed up programming. They require $V_{PP}$ to be at $V_{PPH}$ except for the Bank Erase command, which also operates at $V_{PP} = V_{DD}$. Refer to *Table 7: Factory program commands* in conjunction with the descriptions in the following subsections.

The use of factory program commands requires certain operating conditions.

- $V_{PP}$ must be set to $V_{PPH}$ (except for Bank Erase command).
- $V_{DD}$ must be within operating range.
- Ambient temperature, $T_A$ must be 25°C ± 5°C.
- The targeted block must be unlocked.

## 6.1 Bank Erase command

The Bank Erase command erases a bank. It sets all the bits within the selected bank to '1'. All previous data in the bank is lost. The Bank Erase command ignores any protected blocks within the bank. If all blocks in the bank are protected then the Bank Erase operation aborts and the data in the bank does not change. The Status Register does not output any error.

Bank erase operations can be performed at both $V_{PP} = V_{PPH}$ and $V_{PP} = V_{DD}$.

Two bus write cycles are required to issue the command.

- The first bus cycle sets up the Bank Erase command.
- The second latches the bank address in the Program/Erase Controller and starts it.

If the second bus cycle is not Write Bank Erase Confirm (D0h), Status Register bits SR4 and SR5 are set, and the command aborts. Erase aborts if Reset turns to $V_{IL}$. As data integrity cannot be guaranteed when the erase operation is aborted, the bank must be erased again.

Once the command is issued, the device outputs the Status Register data when any address within the bank is read. At the end of the operation the bank remains in Read Status Register mode until a Read Array, Read CFI Query or Read Electronic Signature command is issued.

During bank erase operations the bank being erased only accepts the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query commands; all other commands are ignored.

For optimum performance, Bank Erase commands should be limited to a maximum of 100 program/erase cycles per block. After 100 program/erase cycles the internal algorithm still operates properly but some degradation in performance may occur.

Dual operations are not supported during bank erase operations and the command cannot be suspended.

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

## 6.2 Double Word Program command

The Double Word Program command improves the programming throughput by writing a page of two adjacent words in parallel. The two words must differ only for the address A0.

If the block is protected, then the Double Word Program operation aborts, the data in the block does not be change, and the Status Register outputs the error.

$V_{PP}$ must be set to $V_{PPH}$ during Double Word Program, otherwise the command is ignored, and the Status Register does not output any error.

Three bus write cycles are necessary to issue the Double Word Program command.

- The first bus cycle sets up the Double Word Program command.
- The second bus cycle latches the address and the data of the first word to be written.
- The third bus cycle latches the address and the data of the second word to be written and starts the Program/Erase Controller.

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

During Double Word Program operations the bank being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query commands; all other commands are ignored. Dual operations are not supported during Double word program operations and the command cannot be suspended. Typical program times are given in *Table 16: Program/erase times and endurance cycles*.

Programming aborts if Reset goes to $V_{IL}$. As data integrity cannot be guaranteed when the program operation is aborted, the memory locations must be reprogrammed.

See *Appendix C*, *Figure 20: Double word program flowchart and pseudo code* for the flowchart for using the Double Word Program command.

## 6.3 Quadruple Word Program command

The Quadruple Word Program command improves the programming throughput by writing a page of four adjacent words in parallel. The four words must only differ for the addresses A0 and A1.

$V_{PP}$ must be set to $V_{PPH}$ during Quadruple Word Program, otherwise the command is ignored and the Status Register does not output any error.

If the block is protected, then the Quadruple Word Program operation aborts, the data in the block does not change, and the Status Register outputs the error.

Five bus write cycles are necessary to issue the Quadruple Word Program command.

- The first bus cycle sets up the Double Word Program command.
- The second bus cycle latches the address and the data of the first word to be written.
- The third bus cycle latches the address and the data of the second word to be written.
- The fourth bus cycle latches the address and the data of the third word to be written.
- The fifth bus cycle latches the address and the data of the fourth word to be written and starts the Program/Erase Controller (P/EC).

Read operations to the bank being programmed output the Status Register content after the programming has started.

Programming aborts if Reset goes to $V_{IL}$. As data integrity cannot be guaranteed when the program operation is aborted, the memory locations must be reprogrammed.

During quadruple word program operations the bank being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query commands; all other commands are ignored.

Dual operations are not supported during quadruple word program operations and the command cannot be suspended. Typical program times are given in *Table 16: Program/erase times and endurance cycles*.

See *Appendix C*, *Figure 21: Quadruple word program flowchart and pseudo code* for the flowchart for using the Quadruple Word Program command.

## 6.4 Enhanced Factory Program command

The Enhanced Factory Program command programs large streams of data within any one block. It greatly reduces the total programming time when a large number of words are written to a block at any one time.

Dual operations are not supported during the enhanced factory program operation and the command cannot be suspended.

For optimum performance the Enhanced Factory Program commands should be limited to a maximum of 10 program/erase cycles per block. If this limit is exceeded the internal algorithm continues to work properly but some degradation in performance is possible. Typical program times are given in *Table 16*

If the block is protected then the Enhanced Factory Program operation aborts, the data in the block does note change, and the Status Register outputs the error.

The Enhanced Factory Program command has four phases: the setup phase, the program phase to program the data to the memory, the verify phase to check that the data has been correctly programmed and reprogram if necessary and the exit phase. Refer to *Table 7: Factory program commands*, and *Figure 27: Enhanced factory program flowchart*.

### 6.4.1 Setup phase

The Enhanced Factory Program command requires two bus write operations to initiate the command.

- The first bus cycle sets up the Enhanced Factory Program command.
- The second bus cycle confirms the command.

The Status Register P/EC bit SR7 should be read to check that the P/EC is ready. After the confirm command is issued, read operations output the Status Register data. The read Status Register command must not be issued, otherwise it is interpreted as data to program.

If the second bus cycle is not EFP confirm (D0h), Status Register bits SR4 and SR5 are set and the command aborts.

$V_{PP}$ value must be in the $V_{PPH}$ range during the confirm command, otherwise SR4 and SR3 are set and command is aborted.

### 6.4.2 Program phase

The program phase requires n+1 cycles, where n is the number of words (refer to *Table 7: Factory program commands*, and *Figure 27: Enhanced factory program flowchart*).

Three successive steps are required to issue and execute the program phase of the command:

1. Use one bus write operation to latch the start address and the first word to be programmed, where the start address is the location of the first data 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 can either remain the start address, in which case the P/EC increments the address location, or the address can be incremented, in which case the P/EC jumps to the new address. If any address is given with data FFFFh that is not in the same block as the start address, the program phase terminates and the verify phase begins. The Status Register bit SR0 should be read between each bus write cycle to check that the P/EC is ready for the next word.
3. Finally, after all words have been programmed, write one bus write operation with data FFFFh to any address outside the block containing the start address, to terminate the programming phase. If the data is not FFFFh, the command is ignored.

The memory is now set to enter the verify phase.

### 6.4.3 Verify phase

The verify phase is similar to the program phase in that all words must be resent to the memory for them to be checked against the programmed data. The Program/Erase Controller checks the stream of data with the data that was programmed in the program phase and reprograms the memory location, if necessary.

Three successive steps are required to execute the verify phase of the command.

1. Use one bus write operation to latch the start address and the first word, to be verified. The Status Register bit SR0 should be read to check that the Program/Erase Controller is ready for the next word.
2. Each subsequent word to be verified is latched with a new bus write operation. The words must be written in the same order as in the program phase. The address can remain the start address or be incremented. If any address is given with data FFFFh that is not in the same block as the start address, the verify phase terminates. Status Register bit SR0 should be read to check that the P/EC is ready for the next word.
3. Finally, after all words have been verified, write one bus write operation with data FFFFh to any address outside the block containing the start address, to terminate the verify phase.

If the verify phase is successfully completed the memory remains in Read Status Register mode. If the Program/Erase Controller fails to reprogram a given location, the error is signaled in the Status Register.

### 6.4.4 Exit phase

Status Register P/EC bit SR7 set to '1' indicates that the device has returned to read mode. A full Status Register check should be done to ensure that the block has been successfully programmed. See *Section 7: Status Register* for more details.

## 6.5 Quadruple Enhanced Factory Program command

The Quadruple Enhanced Factory Program command programs one or more pages of four adjacent words in parallel. The four words must only differ for the addresses A0 and A1.

$V_{PP}$ must be set to $V_{PPH}$ during the Quadruple Enhanced Factory Program, otherwise the command is ignored and the Status Register does not output any error.

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

If the block is protected then the Quadruple Enhanced Factory Program operation aborts, the data in the block does not change and the Status Register outputs the error.

The Quadruple Enhanced Factory Program command has four phases: the setup phase, the load phase where the data is loaded into the buffer, the combined program and verify phase where the loaded data is programmed to the memory and then automatically checked and reprogrammed if necessary, and the Exit Phase. Unlike the Enhanced Factory Program it is not necessary to resubmit the data for the verify phase. The load phase and the program and verify phase can be repeated to program any number of pages within the block.

### 6.5.1 Setup phase

The Quadruple Enhanced Factory Program command requires one bus write operation to initiate the load phase. After the setup command is issued, read operations output the Status Register data. The Read Status Register command must not be issued, otherwise is is interpreted as data to program.

### 6.5.2 Load phase

The load phase requires 4 cycles to load the data (refer to *Table 7: Factory program commands* and *Figure 28: Quadruple enhanced factory program flowchart*). Once the first word of each page is written it is impossible to exit the load phase until all four words have been written.

Two successive steps are required to issue and execute the load phase of the Quadruple Enhanced Factory Program command:

1. Use one bus write operation to latch the start address and the first word of the first page to be programmed, where the start address is the location of the first data to be programmed. For subsequent pages the first word address can remain the start address (in which case the next page is programmed) or can be any address in the same block. If any address with data FFFFh is given that is not in the same block as the start address, the device enters the exit phase. For the first load phase Status Register bit SR7 should be read after the first word has been issued to check that the command has been accepted (bit SR7 set to ‘0’). This check is not required for subsequent load phases.
2. Each subsequent word to be programmed is latched with a new bus write operation. The address is only checked for the first word of each page as the order of the words to be programmed is fixed.

The memory is now set to enter the program and verify phase.

### 6.5.3 Program and verify phase

In the program and verify phase the four words that were loaded in the load phase are programmed in the memory array and then verified by the Program/Erase Controller. If any errors are found the Program/Erase Controller reprograms the location. During this phase the Status Register shows that the Program/Erase Controller is busy, Status Register bit SR7 is set to ‘0’, and that the device is not waiting for new data, Status Register bit SR0 set to ‘1’. When Status Register bit SR0 is set to ‘0’ the program and verify phase has terminated.

Once the verify phase has successfully completed, subsequent pages in the same block can be loaded and programmed. The device returns to the beginning of the load phase by issuing one bus write operation to latch the address and the first of the four new words to be programmed.

### 6.5.4 Exit phase

Finally, after all the pages have been programmed, write one bus write operation with data FFFFh to any address outside the block containing the start address, to terminate the load and program and verify phases.

Status Register bit SR7 set to '1' and bit SR0 set to '0' indicate that the Quadruple Enhanced Factory Program command has terminated. A full Status Register check should be done to ensure that the block has been successfully programmed. See *Section 7: Status Register* for more details.

If the program and verify phase has successfully completed, the memory returns to read mode. If the P/EC fails to program and reprogram a given location, the error is signaled in the Status Register.

**Table 7. Factory program commands**

| Command | Phase | Cycles | Bus write operations(1) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1st | | 2nd | | 3rd | | Final -1 | | Final | |
| | | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Bank Erase | | 2 | BKA | 80h | BKA | D0h | | | | | | |
| Double Word Program(2) | | 3 | BKA or WA1(3) | 35h | WA1 | PD1 | WA2 | PD2 | | | | |
| Quadruple Word Program(4) | | 5 | BKA or WA1(3) | 56h | WA1 | PD1 | WA2 | PD2 | WA3 | PD3 | WA4 | PD4 |
| Enhanced Factory Program (5) | Setup, program | 2+ n+1 | BKA or WA1(3) | 30h | BA or WA1(6) | D0h | WA1(7) | PD1 | WAn(8) | PAn | NOT WA1(7) | FFFFh |
| | Verify, exit | n+1 | WA1(7) | PD1 | WA2(8) | PD2 | WA3(8) | PD3 | WAn(8) | PAn | NOT WA1(7) | FFFFh |
| Quadruple Enhanced Factory Program (4),(5) | Setup, first load | 5 | BKA or WA1(3) | 75h | WA1(7) | PD1 | WA2(9) | PD2 | WA3(9) | PD3 | WA4(9) | PD4 |
| | First program & verify | | Automatic | | | | | | | | | |
| | Subsequent loads | 4 | WA1i (7) | PD1i | WA2i(9) | PD2i | WA3i(9) | PD3i | | | WA4i(9) | PD4i |
| | Subsequent program & verify | | Automatic | | | | | | | | | |
| | Exit | 1 | NOT WA1(7) | FFFFh | | | | | | | | |

1. WA = Word Address in targeted bank, BKA = Bank Address, PD = Program Data, BA = Block Address.
2. Word addresses 1 and 2 must be consecutive addresses differing only for A0.
3. Any address within the bank can be used.
4. Word addresses 1,2,3 and 4 must be consecutive addresses differing only for A0 and A1.
5. A bus read must be done between each write cycle where the data is programmed or verified to read the Status Register and check that the memory is ready to accept the next data. n = number of words, i = number of pages to be programmed.
6. Any address within the block can be used.
7. WA1 is the start address. NOT WA1 is any address that is not in the same block as WA1.
8. Address can remain starting address WA1 or be incremented.
9. Address is only checked for the first word of each page as the order to program the words in each page is fixed so subsequent words in each page can be written to any address.

# 7 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 5.2: Read Status Register command* for more details). To output the contents, the Status Register is latched and updated on the falling edge of the Chip Enable or Output Enable signals and can be read until Chip Enable or Output Enable returns to $V_{IH}$. The Status Register can only be read using single asynchronous or single synchronous reads. Bus read operations from any address within the bank always read the Status Register during program and erase operations as long as 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 provide information on the status of the device and are set and reset by the device. Bits SR5, SR4, SR3 and SR1 give information on errors. They are set by the device but must be reset by issuing a Clear Status Register command or a hardware reset. If an error bit is set to ‘1’ the Status Register should be reset before issuing another command. SR7 to SR1 refer to the status of the device, while SR0 refers to the status of the addressed bank.

The bits in the Status Register are summarized in *Table 8: Status Register bits*. Refer to *Table 8* in conjunction with the following sections.

## 7.1 Program/Erase Controller status bit (SR7)

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

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

During program and erase operations the Program/Erase Controller status bit can be polled to find the end of the operation. Other bits in the Status Register should not be tested until the Program/Erase Controller completes the operation and the bit is High.

After the Program/Erase Controller completes its operation the erase status, program status, $V_{PP}$ status and block lock status bits should be tested for errors.

## 7.2 Erase suspend status bit (SR6)

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

The erase suspend status should only be considered valid when the Program/Erase Controller status bit is High (Program/Erase Controller inactive). SR7 is set within the erase suspend latency time of the Program/Erase Suspend command being issued, therefore, the memory may still complete the operation instead of entering the suspend mode.

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

## 7.3 Erase status bit (SR5)

The erase status bit identifies if the memory has failed to verify that the block has erased correctly. When the erase status bit is High (set to ‘1’), the Program/Erase Controller has applied the maximum number of pulses to the block and still failed to verify that it has erased correctly. The erase status bit should be read once the Program/Erase Controller status bit is High (Program/Erase Controller inactive).

Once set High, the erase status bit can only be reset Low by a Clear Status Register command or a hardware reset. If set High it should be reset before a new program or erase command is issued, otherwise the new command appears to fail.

## 7.4 Program status bit (SR4)

The program status bit is used to identify a program failure, or an attempt to program a ‘1’ to an already programmed bit when $V_{PP} = V_{PPH}$.

When the program status bit is High (set to ‘1’), the Program/Erase Controller has applied the maximum number of pulses to the byte and still failed to verify that it has programmed correctly.

After an attempt to program a '1' to an already-programmed bit, the program status bit SR4 only goes High (set to '1') if $V_{PP} = V_{PPH}$ (if $V_{PP}$ is different from $V_{PPH}$, SR4 remains Low (set to '0'), and the attempt is not shown).

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

Once set High, the program status bit can only be reset Low by a Clear Status Register command or a hardware reset. If set High, it should be reset before a new command is issued, otherwise the new command appears to fail.

## 7.5 $V_{PP}$ status bit (SR3)

The $V_{PP}$ status bit identifies an invalid voltage on the $V_{PP}$ pin during program and erase operations. The $V_{PP}$ pin is only sampled at the beginning of a program or erase operation. Indeterminate results can occur if $V_{PP}$ becomes invalid during an operation.

When the $V_{PP}$ status bit is Low (set to '0'), the voltage on the $V_{PP}$ pin was sampled at a valid voltage. When the $V_{PP}$ status bit is High (set to '1'), the $V_{PP}$ pin has a voltage that is below the $V_{PP}$ lockout voltage, $V_{PPLK}$, the memory is protected, and Program and Erase operations cannot be performed.

Once set High, the $V_{PP}$ status bit can only be reset Low by a Clear Status Register command or a hardware reset. If set High it should be reset before a new program or erase command is issued, otherwise the new command appears to fail.

## 7.6 Program suspend status bit (SR2)

The program suspend status bit indicates that a program operation has been suspended in the addressed block. When the program suspend status bit is High (set to '1'), a Program/Erase Suspend command has been issued and the memory is waiting for a Program/Erase Resume command. The program suspend status should only be considered valid when the Program/Erase Controller status bit is High (Program/Erase Controller inactive). SR2 is set within the program suspend latency time of the Program/Erase Suspend command being issued, therefore, the memory may still complete the operation rather than entering the suspend mode.

When a Program/Erase Resume command is issued the program suspend status bit returns Low.

## 7.7 Block protection status bit (SR1)

The block protection status bit identifies if a program or block erase operation has tried to modify the contents of a locked or locked-down block.

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

Once set High, the block protection status bit can only be reset Low by a Clear Status Register command or a hardware reset. If set High it should be reset before a new command is issued, otherwise the new command appears to fail.

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

The bank write status bit indicates whether the addressed bank is programming or erasing. In enhanced factory program mode the multiple word program bit shows if a word has finished programming or verifying depending on the phase. The bank write status bit should only be considered valid when the Program/Erase Controller status bit SR7 is Low (set to ‘0’).

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

In enhanced factory program mode if multiple word program status bit is Low (set to ‘0’), the device is ready for the next word. If the multiple word program status bit is High (set to ‘1’), the device is not ready for the next word.

Refer to *Appendix C: Flowcharts and pseudo codes* for using the Status Register.

**Table 8. Status Register bits**

| Bit | Name | Type | Logic Level(1) | Definition | |
|---|---|---|---|---|---|
| SR7 | P/EC status | Status | '1' | Ready | |
| | | | '0' | Busy | |
| SR6 | Erase suspend status | Status | '1' | Erase suspended | |
| | | | '0' | Erase in progress or completed | |
| SR5 | Erase status | Error | '1' | Erase error | |
| | | | '0' | Erase success | |
| SR4 | Program status | Error | '1' | Program error | |
| | | | '0' | Program success | |
| SR3 | $V_{PP}$ status | Error | '1' | $V_{PP}$ invalid, abort | |
| | | | '0' | $V_{PP}$ 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 (enhanced factory program mode) | Status | '1' | SR7 = ‘1’ | Not allowed |
| | | | | SR7 = ‘0’ | The device is not ready for the next word |
| | | | '0' | SR7 = ‘1’ | The device is exiting EFP |
| | | | | SR7 = ‘0’ | The device is ready for the next word |

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

# 8 Configuration Register

The Configuration Register configures the type of bus access that the memory performs. Refer to *Section 9* for details on read operations.

The Configuration Register is set through the command interface. After a reset or power-up the device is configured for asynchronous page read (CR15 = 1). The Configuration Register bits are described in *Table 10*. They specify the selection of the burst length, burst type, burst X latency, and the read operation. Refer to Figures *5* and *6* for examples of synchronous burst configurations.

## 8.1 Read select bit (CR15)

The read select bit, CR15, switches between asynchronous and synchronous bus read operations. When the read select bit is set to ’1’, read operations are asynchronous, and when it is set to ’0’, read operations are synchronous. Synchronous burst read is supported in both parameter and main blocks and can be performed across banks.

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

## 8.2 X latency bits (CR13-CR11)

The X latency bits are used during Synchronous read operations to set the number of clock cycles between the address being latched and the first data becoming available. For correct operation the X latency bits can only assume the values in *Table 10: Configuration Register*.

*Table 9* shows how to set the X latency parameter, taking into account the speed class of the device and the frequency used to read the Flash memory in synchronous mode.

**Table 9. Latency settings**

| fmax | tKmin | X latency (Min) | |
|---|---|---|---|
| | | Speed 60 ns | Speed 70 ns |
| 30 MHz | 33 ns | 2 | 2 |
| 40 MHz | 25 ns | 2 | 3 |
| 54 MHz | 19 ns | 3 | 3 |
| 66 MHz | 15 ns | 4 | 4 |

## 8.3 Wait polarity bit (CR10)

In synchronous burst mode the Wait signal indicates whether the output data is valid or a WAIT state must be inserted. The Wait polarity bit is used to set the polarity of the Wait signal. When the Wait polarity bit is set to ‘0’ the Wait signal is active Low. When the Wait polarity bit is set to ‘1’ the Wait signal is active High.

## 8.4 Data output configuration bit (CR9)

The data output configuration bit determines whether the output remains valid for one or two clock cycles. When the data output configuration bit is ’0’ the output data is valid for one clock cycle, and when it is ’1’ the output data is valid for two clock cycles.

The data output configuration depends on the condition:

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

where $t_K$ is the clock period, $t_{QVK\_CPU}$ is the data setup time required by the system CPU and $t_{KQV}$ is the clock to data valid time. If this condition is not satisfied, the data output configuration bit should be set to ‘1’ (two clock cycles). Refer to *Figure 5: X latency and data output configuration example*.

## 8.5 Wait configuration bit (CR8)

In burst mode the Wait bit controls the timing of the Wait output pin, WAIT. When WAIT is asserted, data is not valid and when WAIT is de-asserted, data is valid. When the Wait bit is ’0’ the Wait output pin is asserted during the wait state. When the Wait bit is ’1’ the Wait output pin is asserted one clock cycle before the wait state.

## 8.6 Burst type bit (CR7)

The burst type bit configures the sequence of addresses read as sequential or interleaved. When the burst type bit is ’0’ the memory outputs from interleaved addresses. When the burst type bit is ’1’ the memory outputs from sequential addresses. See *Table 11: Burst type definition* for the sequence of addresses output from a given starting address in each mode.

## 8.7 Valid Clock edge bit (CR6)

The valid Clock edge bit, CR6, configures the active edge of the Clock, K, during synchronous burst read operations. When the valid Clock edge bit is ’0’ the falling edge of the Clock is the active edge, and when the valid Clock edge bit is ’1’ the rising edge of the Clock is active.

## 8.8 Wrap burst bit (CR3)

The burst reads can be confined inside the 4 or 8 word boundary (wrap) or can overcome the boundary (no wrap). The wrap burst bit selects either wrap and no wrap. When the wrap burst bit is set to ‘0’ the burst read wraps, and when it is set to ‘1’ the burst read does not wrap.

## 8.9 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 4 words, 8 words, 16 words, or continuous burst, where all the words are read sequentially.

In continuous burst mode the burst sequence can cross bank boundaries.

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

If the starting address is aligned to a 4 word boundary no wait states are needed and the WAIT output is not asserted.

If the starting address is shifted by 1, 2 or 3 positions from the four word boundary, WAIT is asserted for 1, 2 or 3 clock cycles when the burst sequence crosses the first 16 word boundary. This indicates that the device needs an internal delay to read the successive words in the array. WAIT is only asserted once during a continuous burst access. See also *Table 11: Burst type definition*.

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

**Table 10. Configuration Register**

| Bit | Description | Value | Description |
|---|---|---|---|
| CR15 | Read select | 0 | Synchronous read |
| | | 1 | Asynchronous read (default at power-on) |
| CR14 | Reserved | | |
| CR13-CR11 | X latency | 010 | 2 clock latency |
| | | 011 | 3 clock latency |
| | | 100 | 4 clock latency |
| | | 101 | 5 clock latency |
| | | 111 | Reserved (default) |
| | | Other configurations reserved | |
| CR10 | Wait polarity | 0 | WAIT is active Low |
| | | 1 | WAIT is active High (default) |
| CR9 | Data output configuration | 0 | Data held for one clock cycle |
| | | 1 | Data held for two clock cycles (default) |
| CR8 | Wait configuration | 0 | WAIT is active during wait state |
| | | 1 | WAIT is active one data cycle before wait state (default) |
| CR7 | Burst type | 0 | Interleaved |
| | | 1 | Sequential (default) |
| CR6 | Valid Clock edge | 0 | Falling Clock edge |
| | | 1 | Rising Clock edge (default) |
| CR5-CR4 | Reserved | | |
| CR3 | Wrap burst | 0 | Wrap |
| | | 1 | No wrap (default) |
| CR2-CR0 | Burst length | 001 | 4 words |
| | | 010 | 8 words |
| | | 011 | 16 words |
| | | 111 | Continuous (CR7 must be set to ‘1’) (default) |

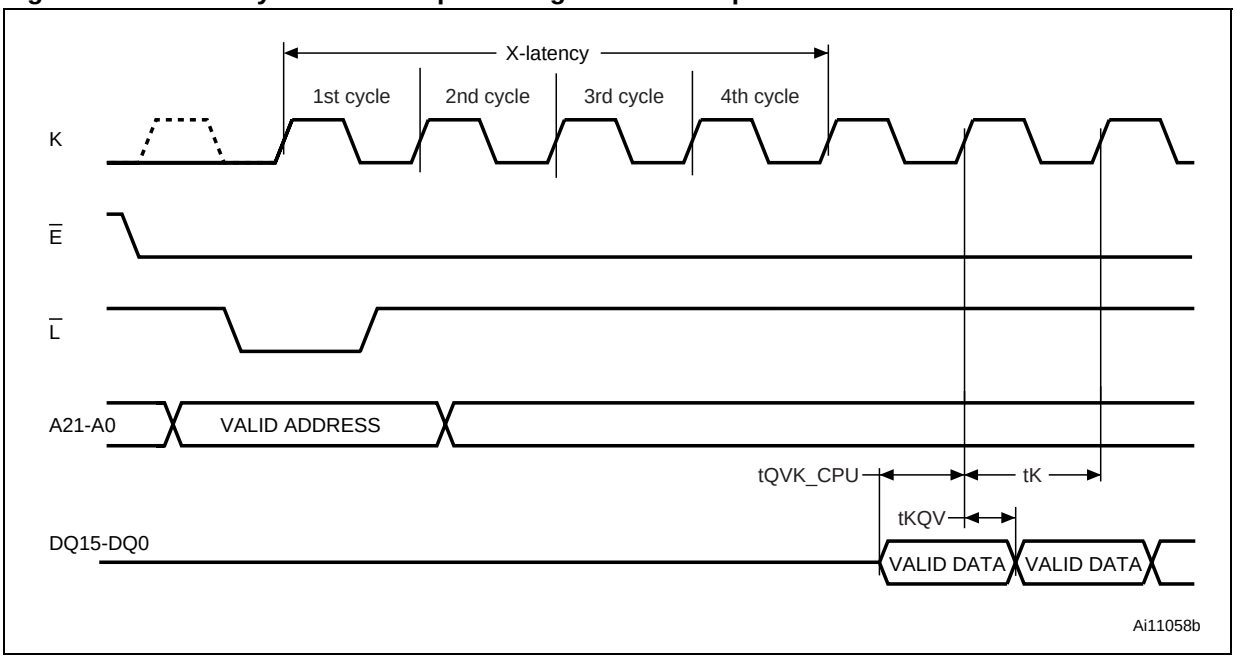
**Table 11. Burst type definition**

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

**Table 11. Burst type definition (continued)**

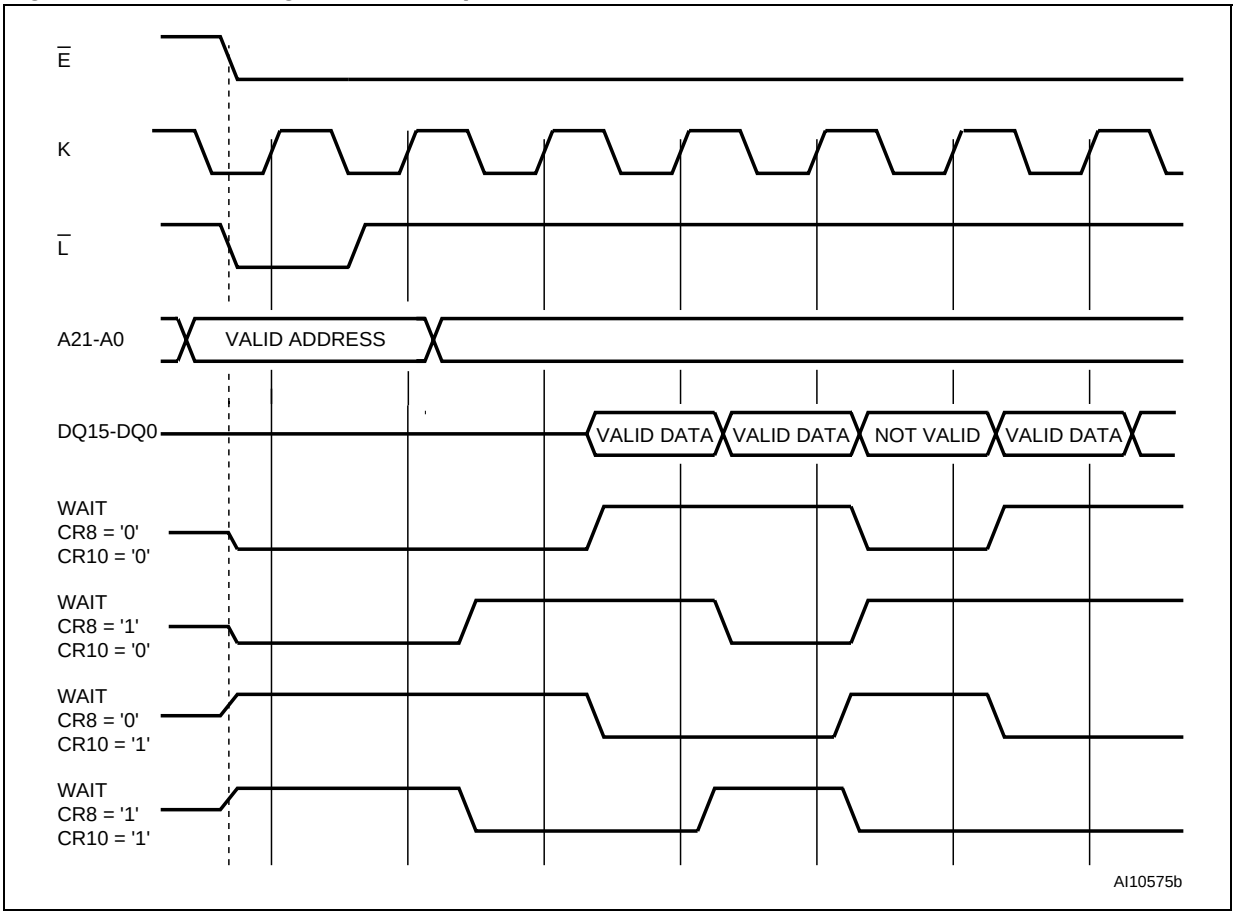
| Mode | Start addr | 4 words | | 8 words | | 16 words | | Continuous burst |
|---|---|---|---|---|---|---|---|---|
| | | Sequential | Interleaved | Sequential | Interleaved | Sequential | Interleaved | |
| No-wrap | 0 | 0-1-2-3 | | 0-1-2-3-4-5-6-7 | | 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 | | Same as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

**Figure 5. X latency and data output configuration example**




1. Settings shown: X-latency = 4, Data Output held for one clock cycle.

**Figure 6. Wait configuration example**



# 9 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 data output format are determined by the Configuration Register (see *Section 8: Configuration Register* for details). All banks supports both asynchronous and synchronous read operations. The multiple bank architecture allows read operations in one bank, while write operations are being executed in another (see Tables *12* and *13*).

## 9.1 Asynchronous read mode

In asynchronous read operations the Clock signal is ‘don’t care’. The device outputs the data corresponding to the address latched, which is the memory array, Status Register, common Flash interface or the Electronic Signature, depending on the command issued. CR15 in the Configuration Register must be set to ‘1’ for asynchronous operations.

In asynchronous read mode a page of data is internally read and stored in a page buffer. The page has a size of 4 words and is addressed by A0 and A1 address inputs. The address inputs A0 and A1 are not gated by Latch Enable in asynchronous read mode.

The first read operation within the page has a longer access time ($T_{acc}$, random access time), while subsequent reads within the same page have much shorter access times. If the page changes then the normal, longer timings apply again.

Asynchronous read operations can be performed in two different ways: asynchronous random access read and asynchronous page read. Only asynchronous page read takes full advantage of the internal page storage so different timings are applied.

During asynchronous read operations, after a bus inactivity of 150 ns, the device automatically switches to automatic standby mode. In this condition the power consumption is reduced to the standby value and the outputs are still driven.

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

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

## 9.2 Synchronous burst read mode

In synchronous burst read mode the data is output in bursts synchronized with the clock. It is possible to perform burst reads across bank boundaries.

Synchronous burst read mode can only be used to read the memory array. For other read operations, such as read Status Register, read CFI and read electronic signature, single synchronous read, or asynchronous random access read must be used.

In synchronous burst read mode the flow of the data output depends on parameters that are configured in the Configuration Register.

A burst sequence is started at the first clock edge (rising or falling depending on valid Clock edge bit CR6 in the Configuration Register) after the falling edge of Latch Enable or Chip Enable, whichever occurs last. Addresses are internally incremented and after a delay of 2 to 5 clock cycles (X latency bits CR13-CR11) the corresponding data is output on each clock cycle.

The number of words to be output during a synchronous burst read operation can be configured as 4, 8, 16 words, or continuous (burst length bits CR2-CR0). The data can be configured to remain valid for one or two clock cycles (data output configuration bit CR9).

The order of the data output can be modified through the burst type and the wrap burst bits in the Configuration Register. The burst sequence may be configured to be sequential or interleaved (CR7). The burst reads can be confined inside the 4, 8 or 16 word boundary (wrap) or overcome the boundary (no wrap). If the starting address is aligned to the burst length (4, 8 or 16 words) the wrapped configuration has no impact on the output sequence. Interleaved mode is not allowed in continuous burst read mode or with no wrap sequences.

A 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. The worst case delay occurs when the sequence is crossing a 16 word boundary and the starting address was at the end of a 4 word boundary.

WAIT is asserted during X latency, the Wait state and at the end of 4-, 8- or 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 burst starting address is aligned to a 4 word page, the Wait state does not occur.

The WAIT signal can be configured to be active Low or active High by setting CR10 in the Configuration Register. The WAIT signal is meaningful only in synchronous burst read mode. In other modes, WAIT is always asserted (except for read array mode).

See *Table 23: Synchronous read AC characteristics* and *Figure 11: Synchronous burst read AC waveforms* for details.

## 9.3 Synchronous burst read suspend

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

A synchronous burst read operation is suspended when $\overline{E}$ is low and the current address has been latched (on a Latch Enable rising edge or on a valid clock edge). The clock signal is then halted at $V_{IH}$ or at $V_{IL}$, and $\overline{G}$ goes high.

When $\overline{G}$ becomes low again and the clock signal restarts, the synchronous burst read operation is resumed exactly where it stopped.

WAIT being gated by $\overline{E}$ remains active and does not revert to high-impedance when $\overline{G}$ goes high. Therefore, if two or more devices are connected to the system's READY signal, to prevent bus contention the WAIT signal of the Flash memory should not be directly connected to the system's READY signal.

See *Table 23: Synchronous read AC characteristics* and *Figure 13: Synchronous burst read suspend AC waveforms* for details.

## 9.4 Single synchronous read mode

Single synchronous read operations are similar to synchronous burst read operations, except that only the first data output after the X latency is valid. Synchronous single reads are used to read the electronic signature, Status Register, CFI, block protection status, Configuration Register status or Protection Register. When the addressed bank is in read CFI, read Status Register or read electronic signature mode, the WAIT signal is always asserted.

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

# 10 Dual operations and multiple bank architecture

The multiple bank architecture of the M58WR064HT/B provides flexibility for software developers by allowing code and data to be split with 4 Mbit granularity. The dual operations feature simplifies the software management of the device and allows 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. Therefore, 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. The combination of these features means that read operations are possible at any moment.

Dual operations between the parameter bank and either the CFI, OTP or the electronic signature memory space are not allowed. *Table 14*, however, shows dual operations, which are allowed between the CFI, OTP, electronic signature locations, and the memory array.

Tables *12* and *13* show the dual operations possible in other banks and in the same bank. For a complete list of possible commands refer to *Appendix D: Command interface state tables*.

**Table 12. Dual operations allowed in other banks**

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

**Table 13. Dual operations allowed in same bank**

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

1. Not allowed in the block or word that is being erased or programmed.
2. The Read Array command is accepted but the data output is no guaranteed until the program or erase has completed.

**Table 14. Dual operation limitations**

| Current status | | Commands allowed | | | |
|---|---|---|---|---|---|
| | | Read CFI/OTP/ Electronic Signature | Read Parameter Blocks | Read main blocks | |
| | | | | Located in parameter bank | Not located in parameter bank |
| Programming/erasing parameter blocks | | No | No | No | Yes |
| Programming/ erasing main blocks | Located in parameter bank | Yes | No | No | Yes |
| | Not located in parameter bank | Yes | Yes | Yes | In different bank only |
| Programming OTP | | No | No | No | No |

# 11 Block locking

The M58WR064HT/B features an instant, individual block locking scheme that allows any block to be locked or unlocked with no latency. This locking scheme has three levels of protection:

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

The protection status of each block can be set to locked, unlocked, and lock-down. *Table 15* defines all of the possible protection states ($\overline{WP}$, DQ1, DQ0), and *Appendix C*, *Figure 25*, shows a flowchart for the locking operations.

## 11.1 Reading the block lock status

The lock status of every block can be read in the read electronic signature mode of the device. To enter this mode write 90 h to the device. Subsequent reads at the address specified in *Table 6* 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. It is also automatically set when entering lock-down. DQ1 indicates the lock-down status and is set by the lock-down command. It cannot be cleared by software, only by a hardware reset or power-down.

The following sections explain the operation of the locking system.

## 11.2 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 any program or erase. 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 lock-down using the appropriate software commands. An unlocked block can be locked by issuing the lock command.

## 11.3 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.

## 11.4 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 $\overline{WP}$ 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. These blocks can then be re-locked (1,1,1) and unlocked (1,1,0) as desired while $\overline{WP}$ remains high. When $\overline{WP}$ is Low, blocks that were previously locked-down return to the lock-down state (0,1,x) regardless of any changes made while $\overline{WP}$ was High. Device reset or power-down resets all blocks, including those in lock-down, to the locked state.

## 11.5 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 changes. 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 change immediately, but when the erase is resumed, the erase operation completes. Locking operations cannot be performed during a program suspend. Refer to *Appendix D: Command interface state tables* for detailed information on which commands are valid during erase suspend.

**Table 15. Lock status**

| Current protection status(1) ($\overline{WP}$, DQ1, DQ0) | | Next protection status(1) ($\overline{WP}$, 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(2) | 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(2) | 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(3) |

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 = $V_{IH}$ and A0 = $V_{IL}$.
2. All blocks are locked at power-up, so the default configuration is 001 or 101 according to $\overline{WP}$ status.
3. A $\overline{WP}$ transition to $V_{IH}$ on a locked block will restore the previous DQ0 value, giving a 111 or 110.

# 12 Program and erase times and endurance cycles

The program and erase times and the number of program/erase cycles per block are shown in *Table 16*. Exact erase times may change depending on the memory array condition. The best case is when all the bits in the block or bank are at '0' (preprogrammed). The worst case is when all the bits in the block or bank are at '1' (not preprogrammed). Usually, the system overhead is negligible with respect to the erase time.

In the M58WR064HT/B the maximum number of program/ erase cycles depends on the $V_{PP}$ voltage supply used.

**Table 16. Program/erase times and endurance cycles**

| Parameter | | | Condition(1) | Min | Typ | Typical after 100k W/E cycles | Max | Unit |
|---|---|---|---|---|---|---|---|---|
| $V_{PP} = V_{DD}$ | Erase | Parameter block (4 KWord)(2) | | | 0.3 | 1 | 2.5 | s |
| | | Main block (32 KWord) | Preprogrammed | | 0.8 | 3 | 4 | s |
| | | | Not preprogrammed | | 1 | | 4 | s |
| | | Bank (4Mbit) | Preprogrammed | | 4.5 | | | s |
| | | | Not preprogrammed | | 6 | | | s |
| | Program(3) | Word program | | | 10 | 10 | 100 | µs |
| | | Parameter block (4 KWord) | | | 32 | | | ms |
| | | Main block (32 KWord) | | | 256 | | | ms |
| | Suspend latency | Program | | | 5 | | 10 | µs |
| | | Erase | | | 5 | | 20 | µs |
| | Program/erase cycles (per block) | Main blocks | | 100 000 | | | | cycles |
| | | Parameter blocks | | 100 000 | | | | cycles |
| $V_{PP} = V_{PPH}$ | Erase | Parameter block (4 KWord) | | | 0.25 | | 2.5 | s |
| | | Main block (32 KWord) | | | 0.8 | | 4 | s |
| | | Bank (4Mbit) | | | 6 | | | s |
| | Program(3) | Word/ double word/ quadruple word (4) | | | 8 | | 100 | µs |
| | | Parameter block (4 KWord) | Quadruple word(4) | | 8 | | | ms |
| | | | Word | | 32 | | | ms |
| | | Main block (32 KWord) | Quadruple word(4) | | 64 | | | ms |
| | | | Word | | 256 | | | ms |
| | | Bank (4 Mbit) | Quad-enhanced factory program(4) | | 0.7 | | | s |
| | | | Quadruple word(4) | | 0.5 | | | s |
| | Program/erase cycles (per block) | Main blocks | | | | | 1000 | cycles |
| | | Parameter blocks | | | | | 2500 | cycles |

1. $T_A$ = –40 to 85°C; $V_{DD}$ = 1.7V to 2V; $V_{DDQ}$ = 1.7 V to 2.24 V.
2. The difference between preprogrammed and not preprogrammed is not significant (‹30 ms).
3. Excludes the time needed to execute the command sequence.
4. Measurements performed at 25°C. $T_A$ = 25°C ±5°C for quadruple word, double word and quadruple enhanced factory program.

# 13 Maximum ratings

Stressing the device above the rating listed in *Table 17: 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 17. Absolute maximum ratings**

| Symbol | Parameter | Value | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| $T_A$ | Ambient operating temperature | –40 | 85 | °C |
| $T_{BIAS}$ | Temperature under bias | –40 | 125 | °C |
| $T_{STG}$ | Storage temperature | –65 | 155 | °C |
| $V_{IO}$ | Input or output voltage | –0.5 | $V_{DDQ}$+0.6 | V |
| $V_{DD}$ | Supply voltage | –0.2 | 2.45 | V |
| $V_{DDQ}$ | Input/output supply voltage | –0.2 | 2.45 | V |
| $V_{PP}$ | Program voltage | –0.2 | 14 | V |
| $I_O$ | Output short circuit current | | 100 | mA |
| $t_{VPPH}$ | Time for $V_{PP}$ at $V_{PPH}$ | | 100 | hours |

# 14 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 18: Operating and AC measurement conditions*. Designers should check that the operating conditions in their circuit match the operating conditions when relying on the quoted parameters.

**Table 18. Operating and AC measurement conditions**

| Parameter | M58WR064HT/B | | | | |
|---|---|---|---|---|---|
| | 60 | | 70 | | Unit |
| | Min | Max | Min | Max | |
| $V_{DD}$ supply voltage | 1.7 | 2 | 1.7 | 2 | V |
| $V_{DDQ}$ supply voltage | 1.7 | 2.24 | 1.7 | 2.24 | V |
| $V_{PP}$ supply voltage (factory environment) | 11.4 | 12.6 | 11.4 | 12.6 | V |
| $V_{PP}$ supply voltage (application environment) | –0.4 | $V_{DDQ}$+0.4 | –0.4 | $V_{DDQ}$+0.4 | V |
| Ambient operating temperature | –40 | 85 | –40 | 85 | °C |
| Load capacitance ($C_L$) | 30 | | 30 | | pF |
| Input rise and fall times | 5 | | 5 | | ns |
| Input pulse voltages | 0 to $V_{DDQ}$ | | 0 to $V_{DDQ}$ | | V |
| Input and output timing ref. voltages | $V_{DDQ}/2$ | | $V_{DDQ}/2$ | | V |

**Figure 7. AC measurement I/O waveform**

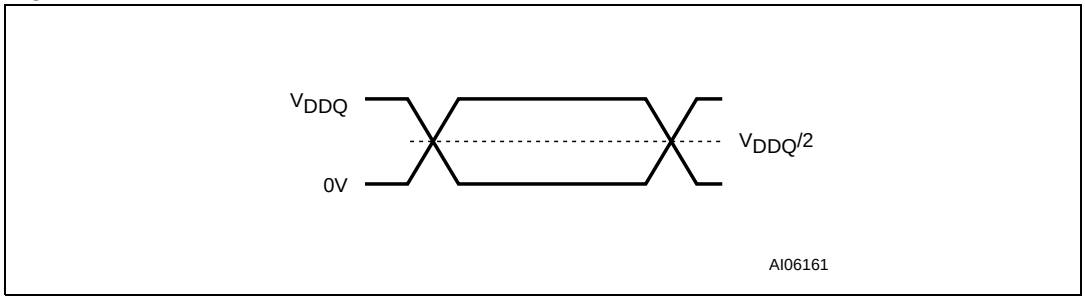


**Figure 8. AC measurement load circuit**

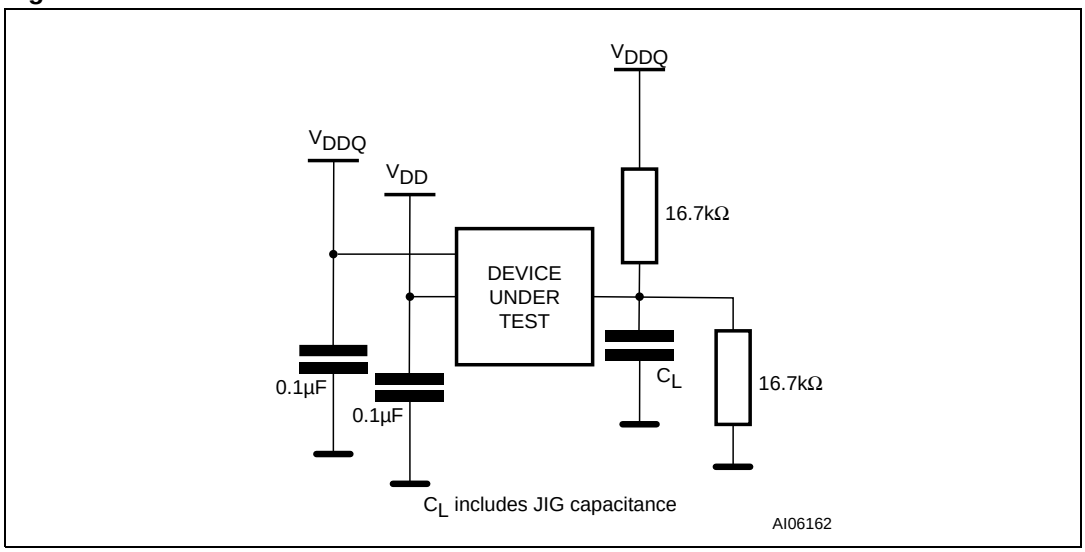



**Table 19. Capacitance(1)**

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

1. Sampled only, not 100% tested.

**Table 20. DC characteristics - currents**

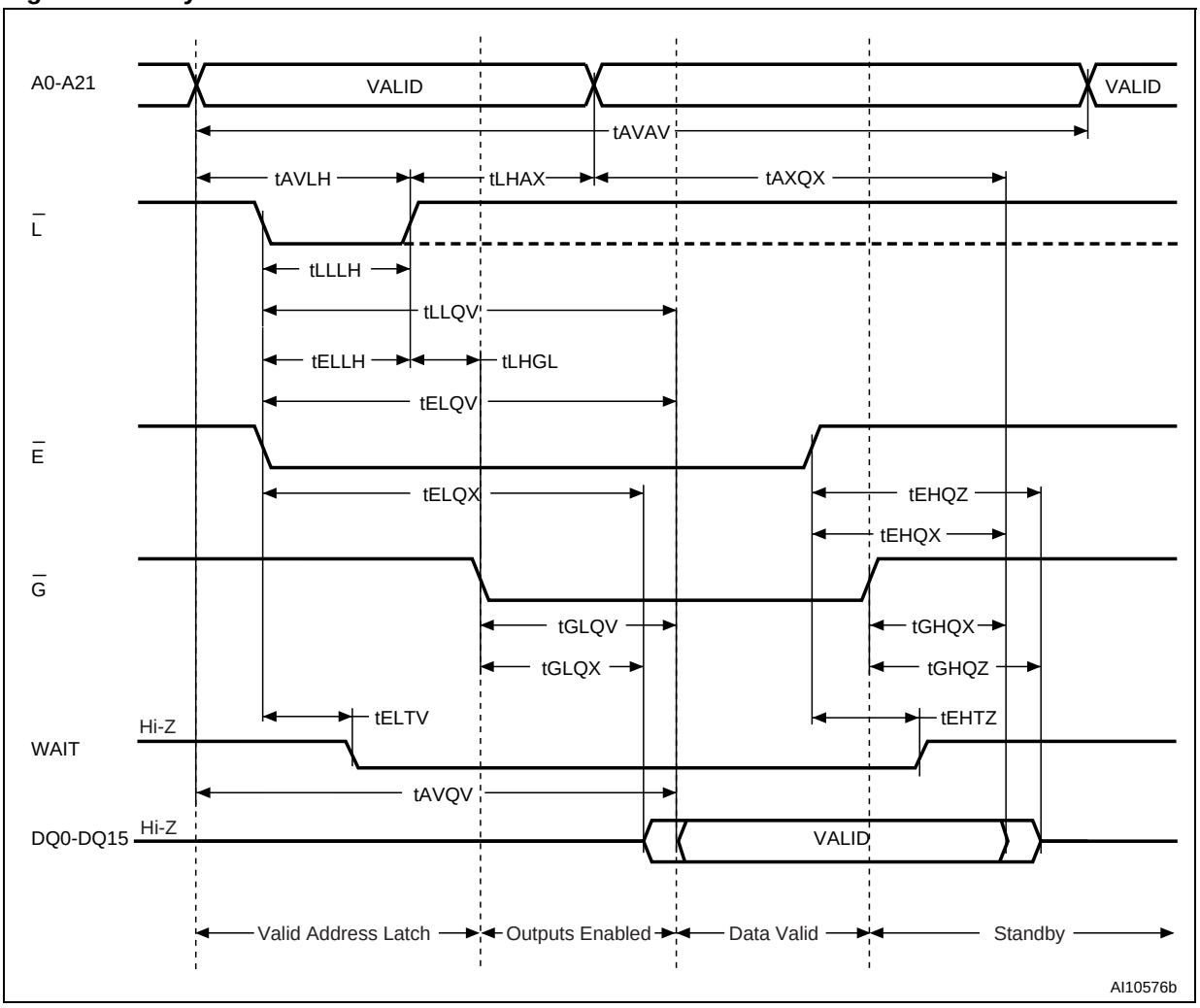
| Symbol | Parameter | Test condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{LI}$ | Input leakage current | $0\ V \le V_{IN} \le V_{DDQ}$ | | | ±1 | µA |
| $I_{LO}$ | Output leakage current | $0\ V \le V_{OUT} \le V_{DDQ}$ | | | ±1 | µA |
| $I_{DD1}$ | Supply current Asynchronous read (f=6 MHz) | $\overline{E} = V_{IL}$, $\overline{G} = V_{IH}$ | | 3 | 6 | mA |
| | Supply current Synchronous read (f=54 MHz) | 4 word | | 7 | 16 | mA |
| | | 8 word | | 10 | 18 | mA |
| | | 16 word | | 12 | 22 | mA |
| | | Continuous | | 13 | 25 | mA |
| | Supply current Synchronous read (f=66MHz) | 4 word | | 8 | 17 | mA |
| | | 8 word | | 11 | 20 | mA |
| | | 16 word | | 14 | 25 | mA |
| | | Continuous | | 16 | 30 | mA |
| $I_{DD2}$ | Supply current (reset) | $\overline{RP} = V_{SS} \pm 0.2\ V$ | | 10 | 50 | µA |
| $I_{DD3}$ | Supply current (standby) | $\overline{E} = V_{DDQ} \pm 0.2\ V$ $K=V_{SS}$ | | 10 | 50 | µA |
| $I_{DD4}$ | Supply current (automatic standby) | $\overline{E} = V_{IL}$, $\overline{G} = V_{IH}$ | | 10 | 50 | µA |
| $I_{DD5}$(1) | Supply current (program) | $V_{PP} = V_{PPH}$ | | 8 | 20 | mA |
| | | $V_{PP} = V_{DD}$ | | 10 | 25 | mA |
| | Supply current (erase) | $V_{PP} = V_{PPH}$ | | 8 | 20 | mA |
| | | $V_{PP} = V_{DD}$ | | 10 | 25 | mA |
| $I_{DD6}$(1),(2) | Supply current (dual operations) | Program/erase in one bank, asynchronous read in another bank | | 13 | 31 | mA |
| | | Program/erase in one bank, synchronous read in another bank | | 23 | 50 | mA |
| $I_{DD7}$(1) | Supply current program/ erase suspended (standby) | $\overline{E} = V_{DDQ} \pm 0.2V$ $K=V_{SS}$ | | 10 | 50 | µA |
| $I_{PP1}$(1) | $V_{PP}$ supply current (program) | $V_{PP} = V_{PPH}$ | | 2 | 5 | mA |
| | | $V_{PP} = V_{DD}$ | | 0.2 | 5 | µA |
| | $V_{PP}$ supply current (erase) | $V_{PP} = V_{PPH}$ | | 2 | 5 | mA |
| | | $V_{PP} = V_{DD}$ | | 0.2 | 5 | µA |
| $I_{PP2}$ | $V_{PP}$ supply current (read) | $V_{PP} \le V_{DD}$ | | 0.2 | 5 | µA |
| $I_{PP3}$(1) | $V_{PP}$ supply current (standby) | $V_{PP} \le V_{DD}$ | | 0.2 | 5 | µA |

1. Sampled only, not 100% tested.
2. $V_{DD}$ dual operation current is the sum of read and program or erase currents.

**Table 21. DC characteristics - voltages**

| Symbol | Parameter | Test condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input low voltage | | –0.5 | | 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 µA | | | 0.1 | V |
| $V_{OH}$ | Output high voltage | $I_{OH}$ = –100 µA | $V_{DDQ}$ –0.1 | | | V |
| $V_{PP1}$ | $V_{PP}$ program voltage-logic | Program, erase | 1.3 | 1.8 | 3.3 | V |
| $V_{PPH}$ | $V_{PP}$ program voltage factory | Program, erase | 11.4 | 12 | 12.6 | 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 |

**Figure 9. Asynchronous random access read AC waveforms**




1. Write Enable, $\overline{W}$, is High, WAIT is active Low.

**Figure 10. Asynchronous page read AC waveforms**

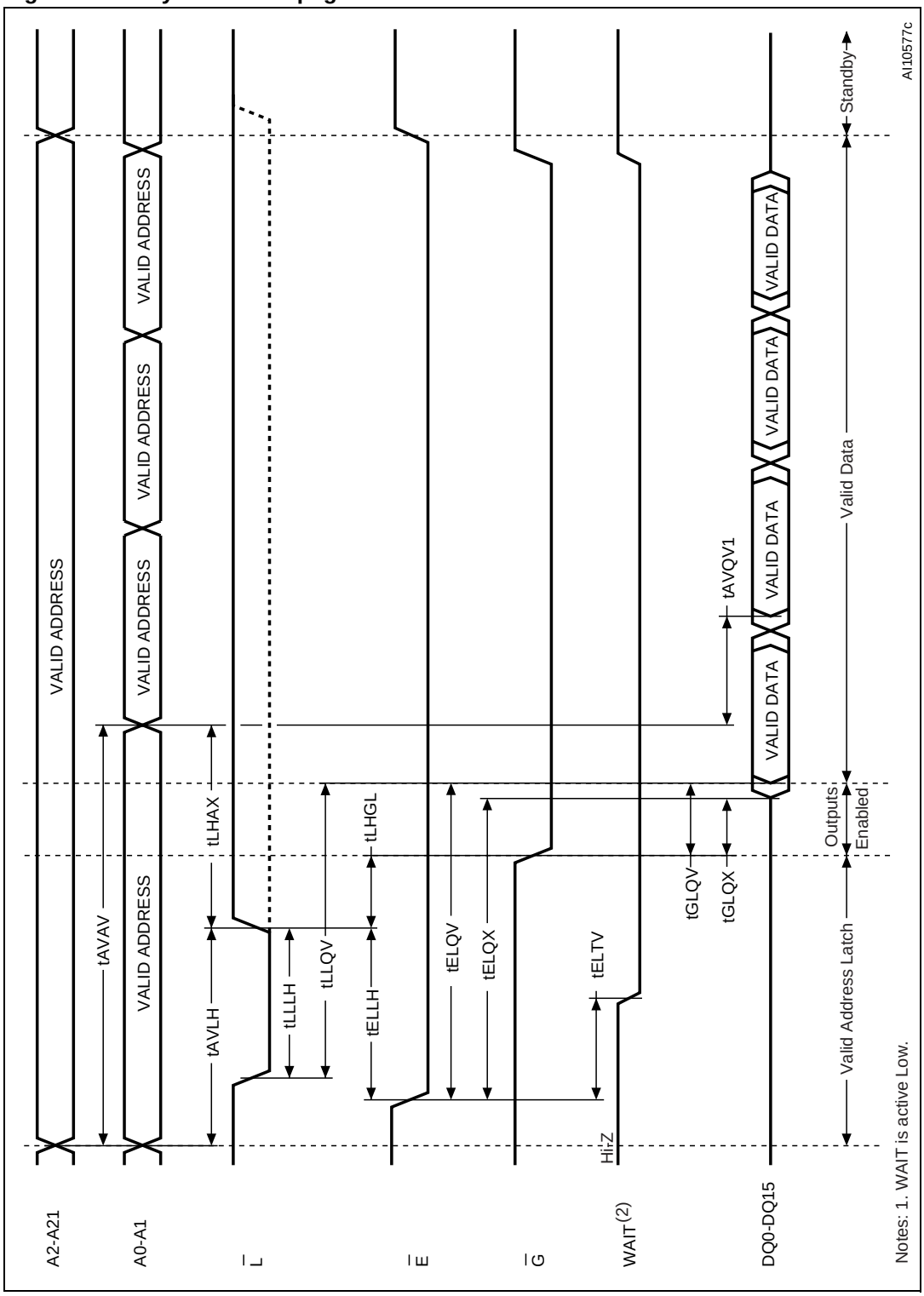


**Table 22. Asynchronous read AC characteristics**

| Symbol | | Alt | Parameter | | M58WR064HT/B | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | | 60 | 70 | |
| Read Timings | $t_{AVAV}$ | $t_{RC}$ | Address Valid to Next Address Valid | Min | 60 | 70 | ns |
| | $t_{AVQV}$ | $t_{ACC}$ | Address Valid to Output Valid (Random) | Max | 60 | 70 | ns |
| | $t_{AVQV1}$ | $t_{PAGE}$ | Address Valid to Output Valid (page) | Max | 20 | 20 | ns |
| | $t_{AXQX}$(1) | $t_{OH}$ | Address Transition to Output Transition | Min | 0 | 0 | ns |
| | $t_{ELTV}$ | | Chip Enable Low to Wait Valid | Max | 11 | 14 | ns |
| | $t_{ELQV}$(2) | $t_{CE}$ | Chip Enable Low to Output Valid | Max | 60 | 70 | ns |
| | $t_{ELQX}$(1) | $t_{LZ}$ | Chip Enable Low to Output Transition | Min | 0 | 0 | ns |
| | $t_{EHTZ}$ | | Chip Enable High to Wait Hi-Z | Max | 14 | 17 | ns |
| | $t_{EHQX}$(1) | $t_{OH}$ | Chip Enable High to Output Transition | Min | 0 | 0 | ns |
| | $t_{EHQZ}$(1) | $t_{HZ}$ | Chip Enable High to Output Hi-Z | Max | 14 | 17 | ns |
| | $t_{GLQV}$(2) | $t_{OE}$ | Output Enable Low to Output Valid | Max | 20 | 20 | ns |
| | $t_{GLQX}$(1) | $t_{OLZ}$ | Output Enable Low to Output Transition | Min | 0 | 0 | ns |
| | $t_{GHQX}$(1) | $t_{OH}$ | Output Enable High to Output Transition | Min | 0 | 0 | ns |
| | $t_{GHQZ}$(1) | $t_{DF}$ | Output Enable High to Output Hi-Z | Max | 14 | 14 | ns |
| Latch Timings | $t_{AVLH}$ | $t_{AVADVH}$ | Address Valid to Latch Enable High | Min | 7 | 9 | ns |
| | $t_{ELLH}$ | $t_{ELADVH}$ | Chip Enable Low to Latch Enable High | Min | 10 | 10 | ns |
| | $t_{LHAX}$ | $t_{ADVHAX}$ | Latch Enable High to Address Transition | Min | 7 | 9 | ns |
| | $t_{LLLH}$ | $t_{ADVLADVH}$ | Latch Enable Pulse Width | Min | 7 | 9 | ns |
| | $t_{LLQV}$ | $t_{ADVLQV}$ | Latch Enable Low to Output Valid (Random) | Max | 60 | 70 | ns |
| | $t_{LHGL}$ | $t_{ADVHGL}$ | Latch Enable High to Output Enable Low | Min | 0 | 0 | ns |

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

**Figure 11. Synchronous burst read AC waveforms**

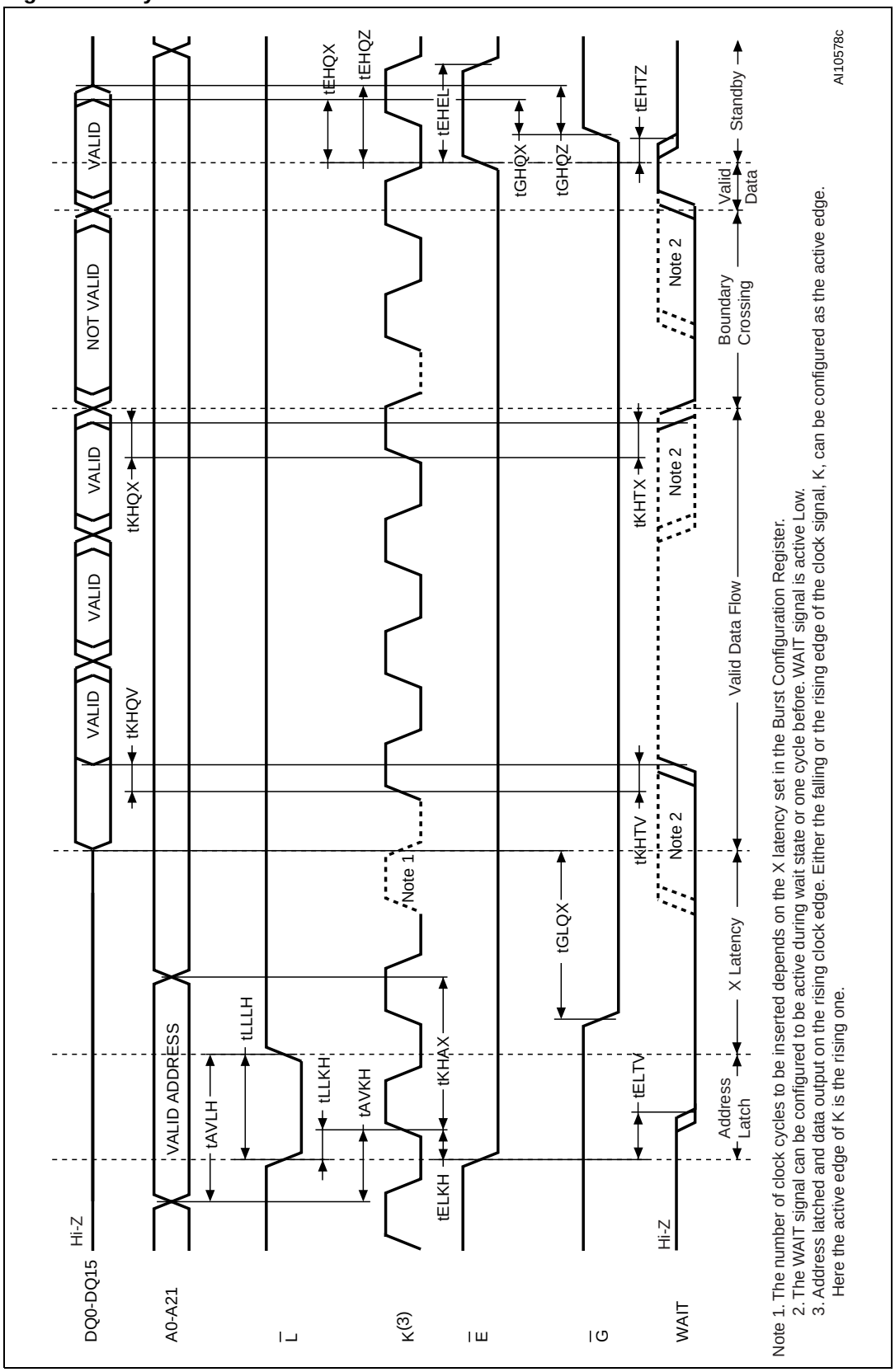
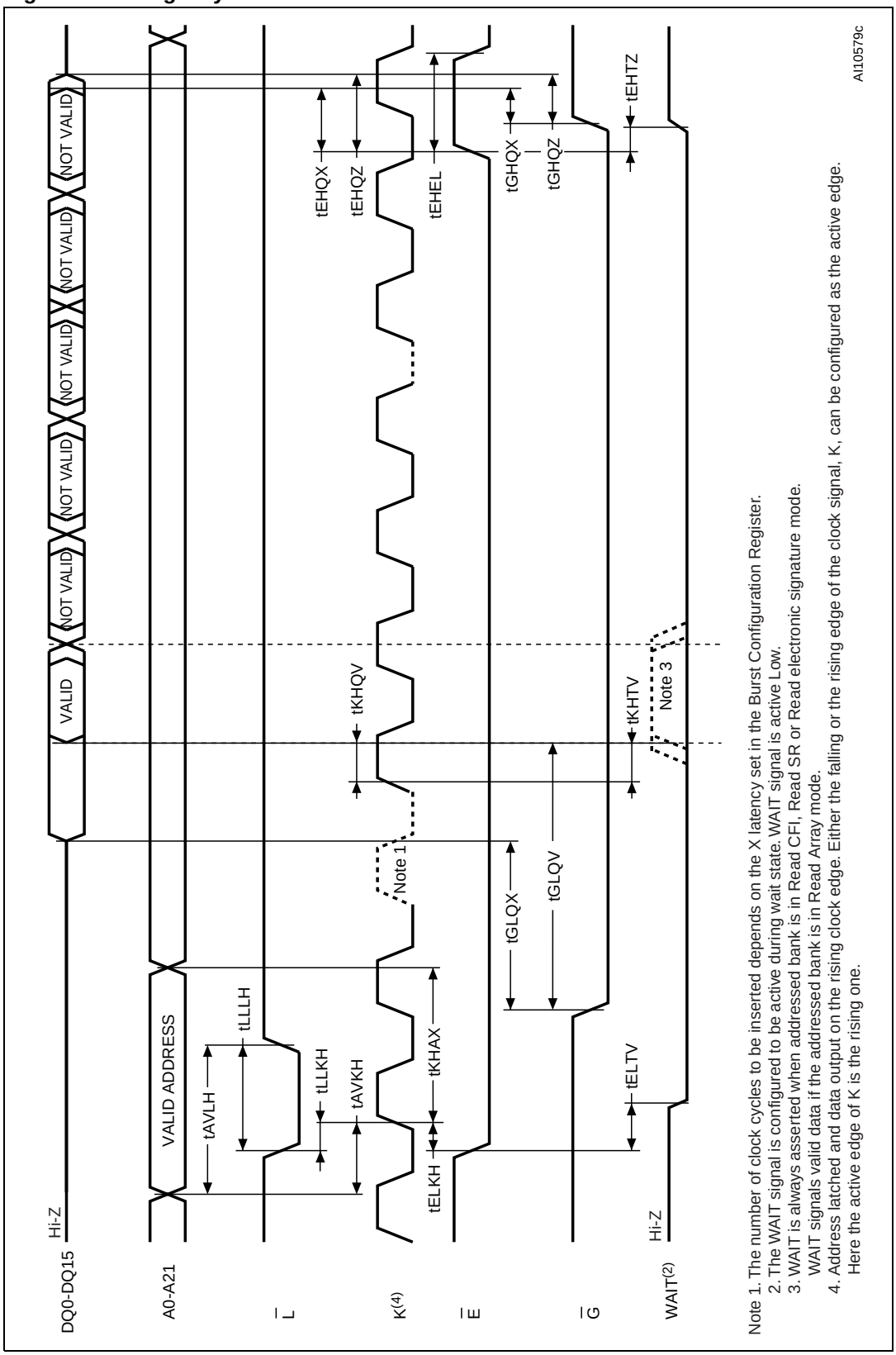
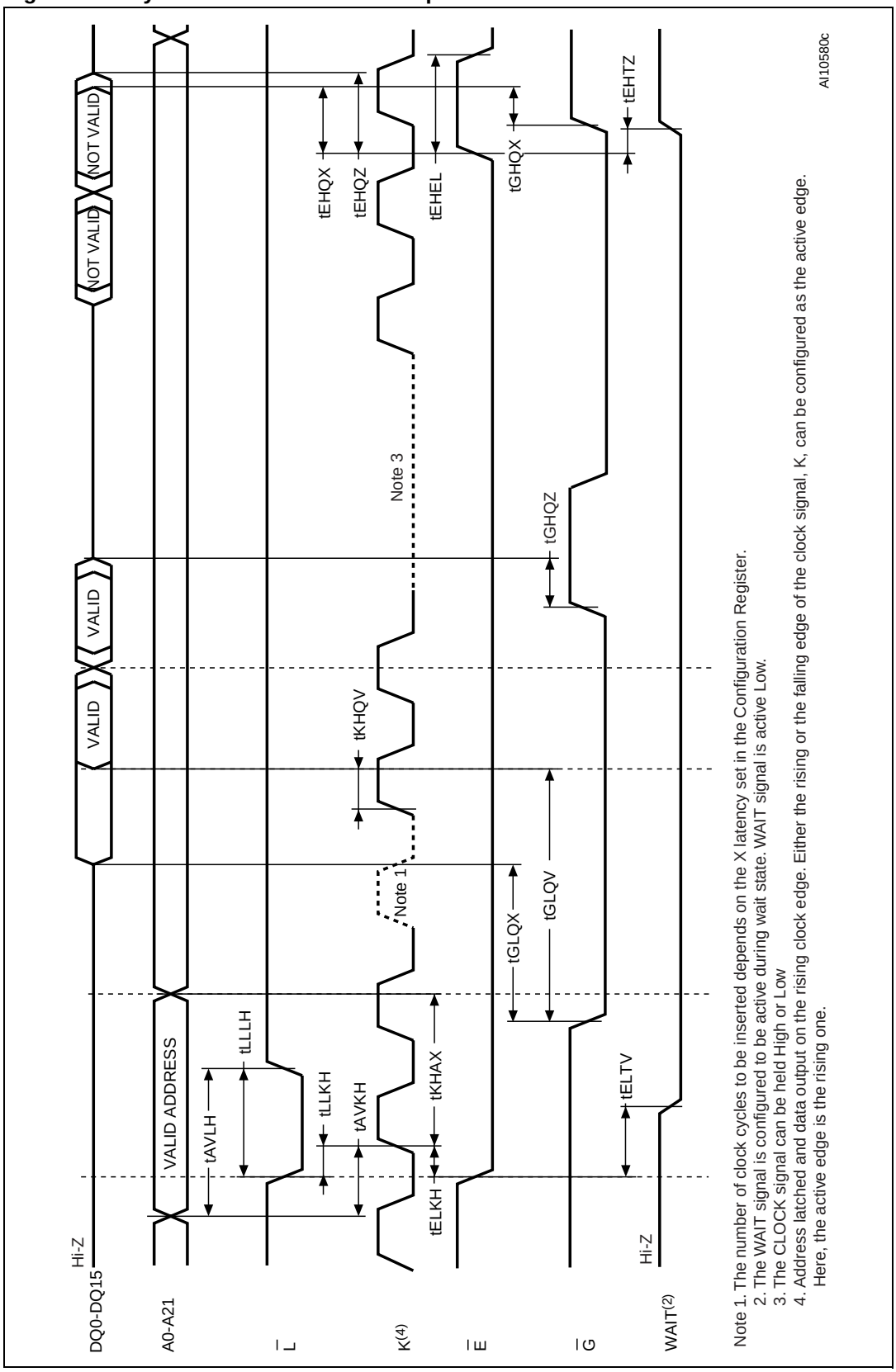


**Figure 12. Single synchronous read AC waveforms**




Note 1. The number of clock cycles to be inserted depends on the X latency set in the Burst Configuration Register.
2. The WAIT signal is configured to be active during wait state. WAIT signal is active Low.
3. WAIT is always asserted when addressed bank is in Read CFI, Read SR or Read electronic signature mode. WAIT signals valid data if the addressed bank is in Read Array mode.
4. Address latched and data output on the rising clock edge. Either the falling or the rising edge of the clock signal, K, can be configured as the active edge. Here the active edge of K is the rising one.

**Figure 13. Synchronous burst read suspend AC waveforms**




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

**Figure 14. Clock input AC waveform**

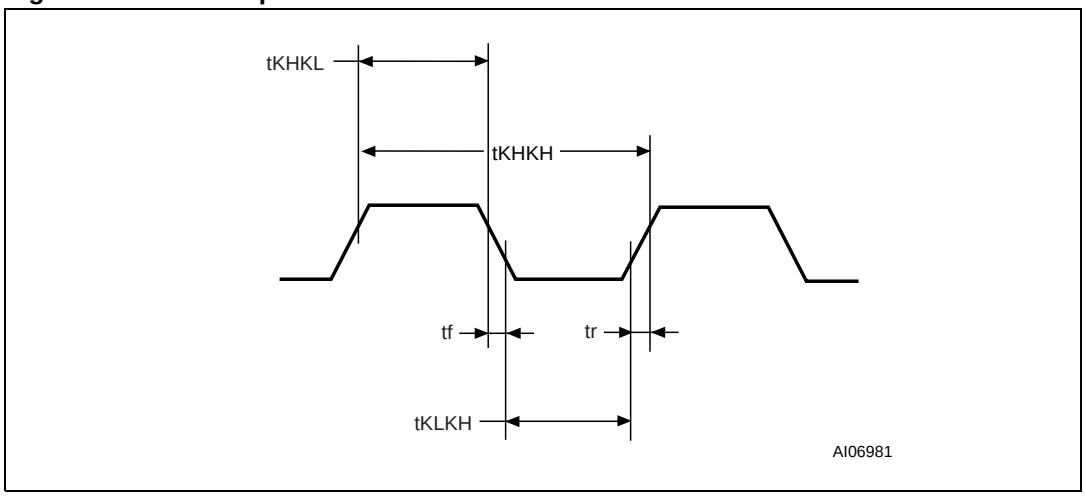



**Table 23. Synchronous read AC characteristics(1) (2)**

| Symbol | | Alt | Parameter | | M58WR064HT/B | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | | 60 | 70 | |
| Synchronous Read Timings | $t_{AVKH}$ | $t_{AVCLKH}$ | Address Valid to Clock High | Min | 7 | 9 | ns |
| | $t_{ELKH}$ | $t_{ELCLKH}$ | Chip Enable Low to Clock High | Min | 7 | 9 | ns |
| | $t_{ELTV}$ | | Chip Enable Low to Wait Valid | Max | 11 | 14 | ns |
| | $t_{EHEL}$ | | Chip Enable Pulse Width (subsequent synchronous reads) | Min | 14 | 14 | ns |
| | $t_{EHTZ}$ | | Chip Enable High to Wait Hi-Z | Max | 11 | 14 | ns |
| | $t_{KHAX}$ | $t_{CLKHAX}$ | Clock High to Address Transition | Min | 7 | 9 | ns |
| | $t_{KHQV}$<br>$t_{KHTV}$ | $t_{CLKHQV}$ | Clock High to Output Valid<br>Clock High to WAIT Valid | Max | 11 | 14 | ns |
| | $t_{KHQX}$<br>$t_{KHTX}$ | $t_{CLKHQX}$ | Clock High to Output Transition<br>Clock High to WAIT Transition | Min | 3 | 4 | ns |
| | $t_{LLKH}$ | $t_{ADVLCLKH}$ | Latch Enable Low to Clock High | Min | 7 | 9 | ns |
| Clock Specifications | $t_{KHKH}$ | $t_{CLK}$ | Clock Period (f=54MHz) | Min | | 18.5 | ns |
| | | | Clock Period (f=66MHz) | Min | 15 | | ns |
| | $t_{KHKL}$<br>$t_{KLKH}$ | | Clock High to Clock Low<br>Clock Low to Clock High | Min | 3.5 | 4.5 | ns |
| | $t_f$<br>$t_r$ | | Clock Fall or Rise Time | Max | 3 | 3 | ns |

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

**Figure 15. Write AC waveforms, Write Enable controlled**

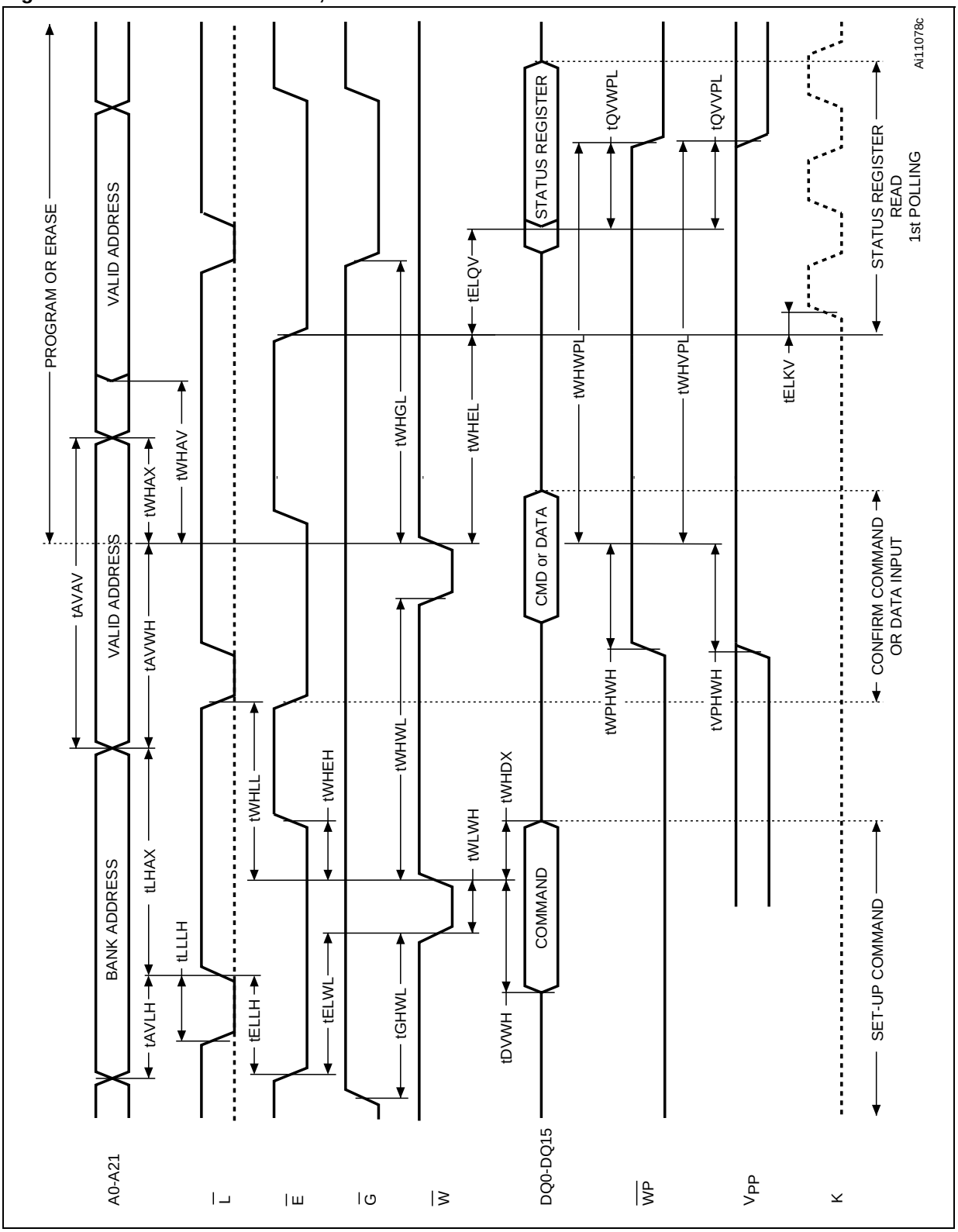


**Table 24. Write AC characteristics, Write Enable controlled(1)**

| | Symbol | Alt | Parameter | | M58WR064HT/B | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | | 60 | 70 | |
| Write Enable Controlled Timings | $t_{AVAV}$ | $t_{WC}$ | Address Valid to Next Address Valid | Min | 60 | 70 | ns |
| | $t_{AVLH}$ | | Address Valid to Latch Enable High | Min | 7 | 9 | ns |
| | $t_{AVWH}$(2) | | Address Valid to Write Enable High | Min | 40 | 45 | ns |
| | $t_{DVWH}$ | $t_{DS}$ | Data Valid to Write Enable High | Min | 40 | 45 | ns |
| | $t_{ELLH}$ | | Chip Enable Low to Latch Enable High | Min | 10 | 10 | ns |
| | $t_{ELWL}$ | $t_{CS}$ | Chip Enable Low to Write Enable Low | Min | 0 | 0 | ns |
| | $t_{ELQV}$ | | Chip Enable Low to Output Valid | Min | 60 | 70 | ns |
| | $t_{ELKV}$ | | Chip Enable Low to Clock Valid | Min | 7 | 9 | ns |
| | $t_{GHWL}$ | | Output Enable High to Write Enable Low | Min | 5 | 5 | ns |
| | $t_{LHAX}$ | | Latch Enable High to Address Transition | Min | 7 | 9 | ns |
| | $t_{LLLH}$ | | Latch Enable Pulse Width | Min | 7 | 9 | ns |
| | $t_{WHAV}$(2) | | Write Enable High to Address Valid | Min | 0 | 0 | ns |
| | $t_{WHAX}$(2) | $t_{AH}$ | Write Enable High to Address Transition | Min | 0 | 0 | ns |
| | $t_{WHDX}$ | $t_{DH}$ | Write Enable High to Input Transition | Min | 0 | 0 | ns |
| | $t_{WHEH}$ | $t_{CH}$ | Write Enable High to Chip Enable High | Min | 0 | 0 | ns |
| | $t_{WHEL}$(3) | | Write Enable High to Chip Enable Low | Min | 20 | 25 | ns |
| | $t_{WHGL}$ | | Write Enable High to Output Enable Low | Min | 0 | 0 | ns |
| | $t_{WHLL}$ | | Write Enable High to Latch Enable Low | Min | 0 | 0 | ns |
| | $t_{WHWL}$ | $t_{WPH}$ | Write Enable High to Write Enable Low | Min | 20 | 25 | ns |
| | $t_{WLWH}$ | $t_{WP}$ | Write Enable Low to Write Enable High | Min | 40 | 45 | ns |
| Protection Timings | $t_{QVVPL}$ | | Output (Status Register) Valid to $V_{PP}$ Low | Min | 0 | 0 | ns |
| | $t_{QVWPL}$ | | Output (Status Register) Valid to Write Protect Low | Min | 0 | 0 | ns |
| | $t_{VPHWH}$ | $t_{VPS}$ | $V_{PP}$ High to Write Enable High | Min | 200 | 200 | ns |
| | $t_{WHVPL}$ | | Write Enable High to $V_{PP}$ Low | Min | 200 | 200 | ns |
| | $t_{WHWPL}$ | | Write Enable High to Write Protect Low | Min | 200 | 200 | ns |
| | $t_{WPHWH}$ | | Write Protect High to Write Enable High | Min | 200 | 200 | ns |

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

**Figure 16. Write AC waveforms, Chip Enable controlled**

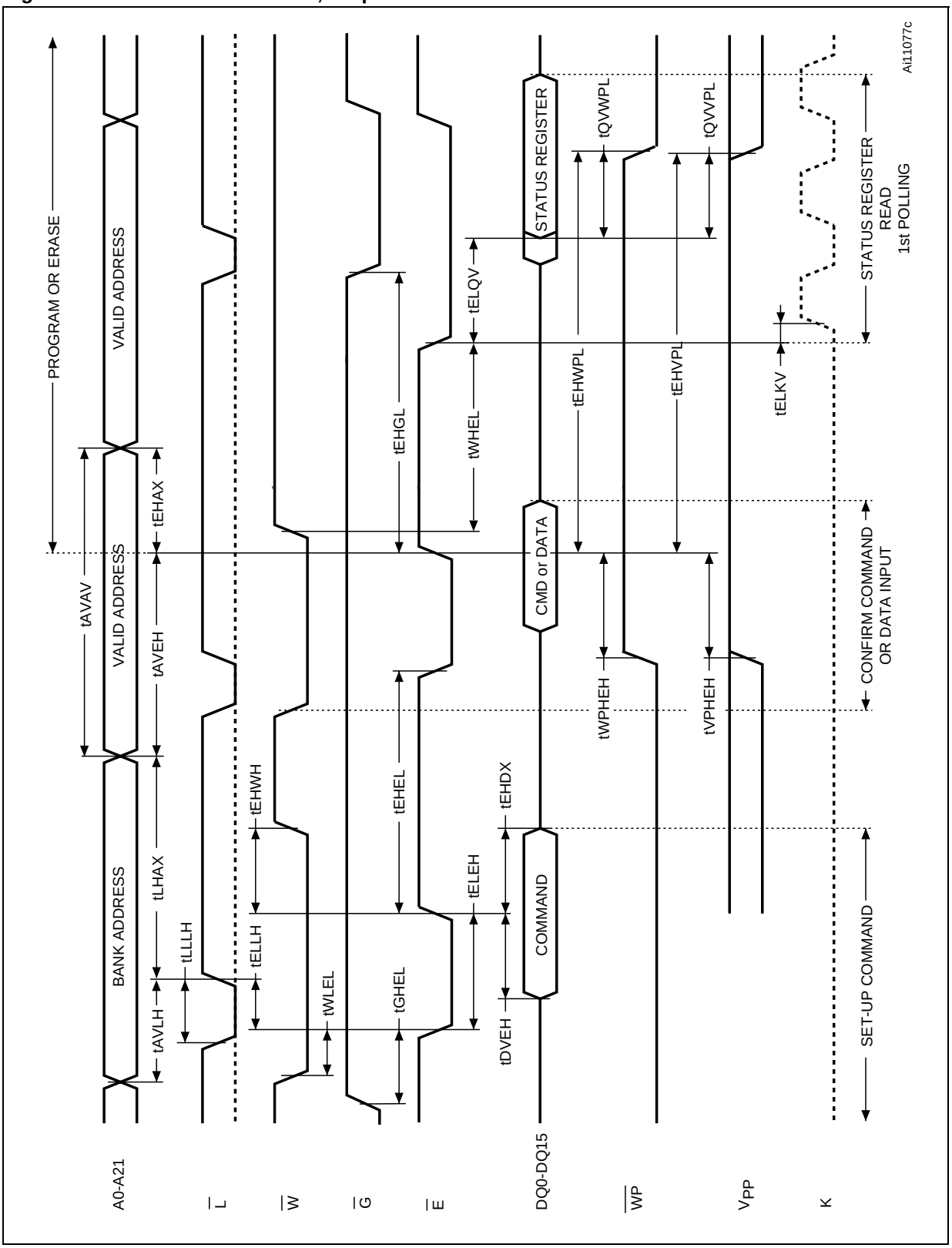


**Table 25. Write AC characteristics, Chip Enable controlled(1)**

| Symbol | | Alt | Parameter | | M58WR064HT/B | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | | 60 | 70 | |
| Chip Enable Controlled Timings | $t_{AVAV}$ | $t_{WC}$ | Address Valid to Next Address Valid | Min | 60 | 70 | ns |
| | $t_{AVEH}$ | | Address Valid to Chip Enable High | Min | 40 | 45 | ns |
| | $t_{AVLH}$ | | Address Valid to Latch Enable High | Min | 7 | 9 | ns |
| | $t_{DVEH}$ | $t_{DS}$ | Data Valid to Chip Enable High | Min | 40 | 45 | ns |
| | $t_{EHAX}$ | $t_{AH}$ | Chip Enable High to Address Transition | Min | 0 | 0 | ns |
| | $t_{EHDX}$ | $t_{DH}$ | Chip Enable High to Input Transition | Min | 0 | 0 | ns |
| | $t_{EHEL}$ | $t_{CPH}$ | Chip Enable High to Chip Enable Low | Min | 20 | 25 | ns |
| | $t_{EHGL}$ | | Chip Enable High to Output Enable Low | Min | 0 | 0 | ns |
| | $t_{EHWH}$ | $t_{CH}$ | Chip Enable High to Write Enable High | Min | 0 | 0 | ns |
| | $t_{ELKV}$ | | Chip Enable Low to Clock Valid | Min | 7 | 9 | ns |
| | $t_{ELEH}$ | $t_{CP}$ | Chip Enable Low to Chip Enable High | Min | 40 | 45 | ns |
| | $t_{ELLH}$ | | Chip Enable Low to Latch Enable High | Min | 10 | 10 | ns |
| | $t_{ELQV}$ | | Chip Enable Low to Output Valid | Min | 60 | 70 | ns |
| | $t_{GHEL}$ | | Output Enable High to Chip Enable Low | Min | 14 | 17 | ns |
| | $t_{LHAX}$ | | Latch Enable High to Address Transition | Min | 7 | 9 | ns |
| | $t_{LLLH}$ | | Latch Enable Pulse Width | Min | 7 | 9 | ns |
| | $t_{WHEL}$(2) | | Write Enable High to Chip Enable Low | Min | 20 | 25 | ns |
| | $t_{WLEL}$ | $t_{CS}$ | Write Enable Low to Chip Enable Low | Min | 0 | 0 | ns |
| Protection Timings | $t_{EHVPL}$ | | Chip Enable High to $V_{PP}$ Low | Min | 200 | 200 | ns |
| | $t_{EHWPL}$ | | Chip Enable High to Write Protect Low | Min | 200 | 200 | ns |
| | $t_{QVVPL}$ | | Output (Status Register) Valid to $V_{PP}$ Low | Min | 0 | 0 | ns |
| | $t_{QVWPL}$ | | Output (Status Register) Valid to Write Protect Low | Min | 0 | 0 | ns |
| | $t_{VPHEH}$ | $t_{VPS}$ | $V_{PP}$ High to Chip Enable High | Min | 200 | 200 | ns |
| | $t_{WPHEH}$ | | 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 following a Set Configuration Register command. System designers should take this into account and may insert a software No-Op instruction to delay the first read in the same bank after issuing any command and to delay the first read to any address after issuing a Set Configuration Register command. If the first read after the command is a Read Array operation in a different bank and no changes to the Configuration Register have been issued, $t_{WHEL}$ is 0ns.

**Figure 17. Reset and power-up AC waveforms**

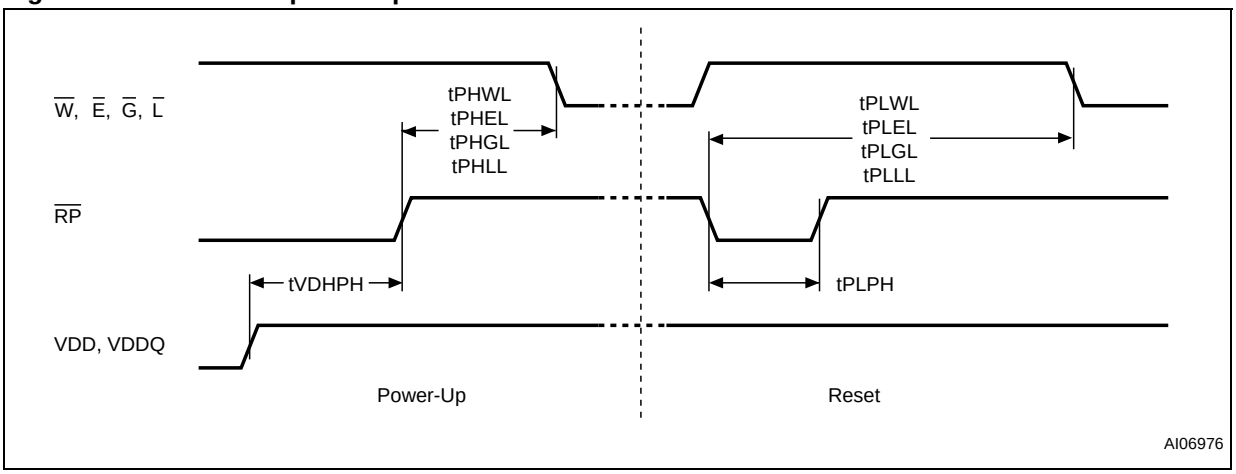



**Table 26. Reset and power-up AC characteristics**

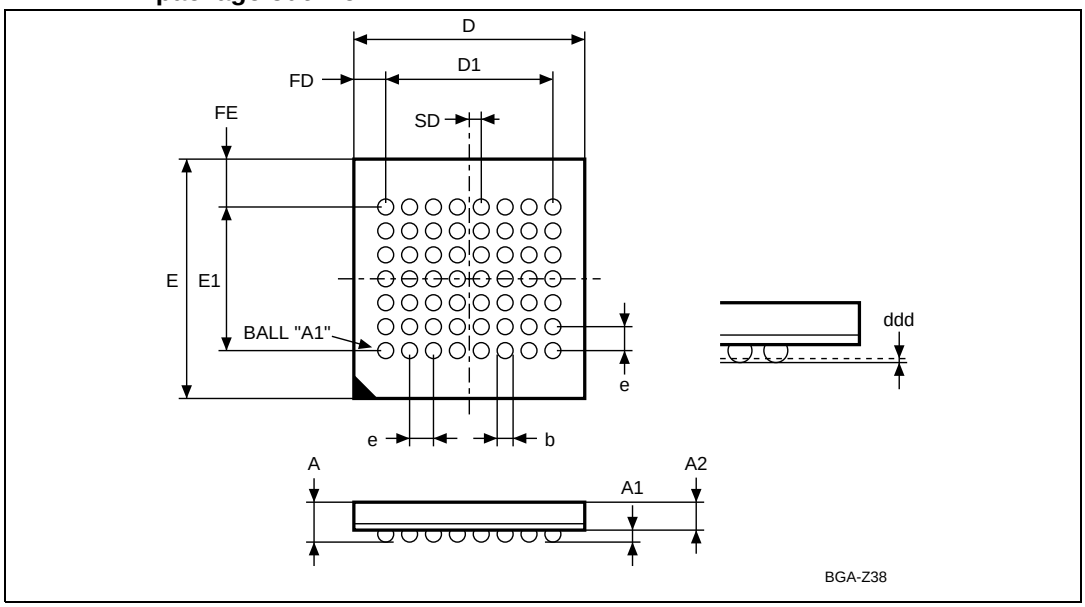
| Symbol | Parameter | Test condition | | 60 | 70 | Unit |
|---|---|---|---|---|---|---|
| $t_{PLWL}$ $t_{PLEL}$ $t_{PLGL}$ $t_{PLLL}$ | Reset Low to Write Enable Low, Chip Enable Low, Output Enable Low, Latch Enable Low | During Program | Min | 10 | 10 | µs |
| | | During Erase | Min | 20 | 20 | µs |
| | | Other Conditions | Min | 80 | 80 | ns |
| $t_{PHWL}$ $t_{PHEL}$ $t_{PHGL}$ $t_{PHLL}$ | Reset High to Write Enable Low Chip Enable Low Output Enable Low Latch Enable Low | | Min | 30 | 30 | ns |
| $t_{PLPH}$(1),(2) | $\overline{RP}$ Pulse Width | | Min | 50 | 50 | ns |
| $t_{VDHPH}$(3) | Supply Voltages High to Reset High | | Min | 50 | 50 | µs |

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

# 15 Package mechanical

To meet environmental requirements, ST offers the M58WR064HB and M58WR064HT in a VFBGA ECOPACK® package. ECOPACK® packages have a lead-free second-level interconnect. The category of second-level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label. ECOPACK is an ST trademark. ECOPACK specifications are available at: *www.st.com*.

**Figure 18. VFBGA56 - 7.7 x 9 mm, 8 x 7 ball array, 0.75 mm pitch, bottom view package outline**




1. Drawing is not to scale.

**Table 27. VFBGA56 - 7.7 x 9 mm, 8 x 7 ball array, 0.75 mm pitch, package mechanical data**

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

# 16 Part numbering

**Table 28. Ordering information scheme**

Example: M58WR064HT 70 ZB 6 F

**Device type**

M58

**Architecture**

W = multiple bank, burst mode

**Operating voltage**

R = $V_{DD}$ = 1.7 V to 2 V, $V_{DDQ}$ = 1.7 V to 2.24 V

**Density**

064 = 64 Mbit

**Technology**

H = 90 nm technology

**Parameter bank location**

T = top boot

B = bottom boot

**Speed**

60 = 60 ns

70 = 70 ns

**Package**

ZB = VFBGA56: 7.7 x 9 mm, 8 x 7 active ball array, 0.75 mm pitch

**Temperature range**

6 = –40 to 85°C

**Option**

E = ECOPACK® package, standard packing

F = ECOPACK® package, tape and reel packing

# Appendix A Block address tables

**Table 29. Top boot block addresses, M58WR064HT**

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Parameter bank | 0 | 4 | 3FF000-3FFFFF |
| | 1 | 4 | 3FE000-3FEFFF |
| | 2 | 4 | 3FD000-3FDFFF |
| | 3 | 4 | 3FC000-3FCFFF |
| | 4 | 4 | 3FB000-3FBFFF |
| | 5 | 4 | 3FA000-3FAFFF |
| | 6 | 4 | 3F9000-3F9FFF |
| | 7 | 4 | 3F8000-3F8FFF |
| | 8 | 32 | 3F0000-3F7FFF |
| | 9 | 32 | 3E8000-3EFFFF |
| | 10 | 32 | 3E0000-3E7FFF |
| | 11 | 32 | 3D8000-3DFFFF |
| | 12 | 32 | 3D0000-3D7FFF |
| | 13 | 32 | 3C8000-3CFFFF |
| | 14 | 32 | 3C0000-3C7FFF |
| Bank 1 | 15 | 32 | 3B8000-3BFFFF |
| | 16 | 32 | 3B0000-3B7FFF |
| | 17 | 32 | 3A8000-3AFFFF |
| | 18 | 32 | 3A0000-3A7FFF |
| | 19 | 32 | 398000-39FFFF |
| | 20 | 32 | 390000-397FFF |
| | 21 | 32 | 388000-38FFFF |
| | 22 | 32 | 380000-387FFF |
| Bank 2 | 23 | 32 | 378000-37FFFF |
| | 24 | 32 | 370000-377FFF |
| | 25 | 32 | 368000-36FFFF |
| | 26 | 32 | 360000-367FFF |
| | 27 | 32 | 358000-35FFFF |
| | 28 | 32 | 350000-357FFF |
| | 29 | 32 | 348000-34FFFF |
| | 30 | 32 | 340000-347FFF |

Table 29. Top boot block addresses, M58WR064HT (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 3 | 31 | 32 | 338000-33FFFF |
| | 32 | 32 | 330000-337FFF |
| | 33 | 32 | 328000-32FFFF |
| | 34 | 32 | 320000-327FFF |
| | 35 | 32 | 318000-31FFFF |
| | 36 | 32 | 310000-317FFF |
| | 37 | 32 | 308000-30FFFF |
| | 38 | 32 | 300000-307FFF |
| Bank 4 | 39 | 32 | 2F8000-2FFFFF |
| | 40 | 32 | 2F0000-2F7FFF |
| | 41 | 32 | 2E8000-2EFFFF |
| | 42 | 32 | 2E0000-2E7FFF |
| | 43 | 32 | 2D8000-2DFFFF |
| | 44 | 32 | 2D0000-2D7FFF |
| | 45 | 32 | 2C8000-2CFFFF |
| | 46 | 32 | 2C0000-2C7FFF |
| Bank 5 | 47 | 32 | 2B8000-2BFFFF |
| | 48 | 32 | 2B0000-2B7FFF |
| | 49 | 32 | 2A8000-2AFFFF |
| | 50 | 32 | 2A0000-2A7FFF |
| | 51 | 32 | 298000-29FFFF |
| | 52 | 32 | 290000-297FFF |
| | 53 | 32 | 288000-28FFFF |
| | 54 | 32 | 280000-287FFF |
| Bank 6 | 55 | 32 | 278000-27FFFF |
| | 56 | 32 | 270000-277FFF |
| | 57 | 32 | 268000-26FFFF |
| | 58 | 32 | 260000-267FFF |
| | 59 | 32 | 258000-25FFFF |
| | 60 | 32 | 250000-257FFF |
| | 61 | 32 | 248000-24FFFF |
| | 62 | 32 | 240000-247FFF |

Table 29. Top boot block addresses, M58WR064HT (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 7 | 63 | 32 | 238000-23FFFF |
| | 64 | 32 | 230000-237FFF |
| | 65 | 32 | 228000-22FFFF |
| | 66 | 32 | 220000-227FFF |
| | 67 | 32 | 218000-21FFFF |
| | 68 | 32 | 210000-217FFF |
| | 69 | 32 | 208000-20FFFF |
| | 70 | 32 | 200000-207FFF |
| Bank 8 | 71 | 32 | 1F8000-1FFFFF |
| | 72 | 32 | 1F0000-1F7FFF |
| | 73 | 32 | 1E8000-1EFFFF |
| | 74 | 32 | 1E0000-1E7FFF |
| | 75 | 32 | 1D8000-1DFFFF |
| | 76 | 32 | 1D0000-1D7FFF |
| | 77 | 32 | 1C8000-1CFFFF |
| | 78 | 32 | 1C0000-1C7FFF |
| Bank 9 | 79 | 32 | 1B8000-1BFFFF |
| | 80 | 32 | 1B0000-1B7FFF |
| | 81 | 32 | 1A8000-1AFFFF |
| | 82 | 32 | 1A0000-1A7FFF |
| | 83 | 32 | 198000-19FFFF |
| | 84 | 32 | 190000-197FFF |
| | 85 | 32 | 188000-18FFFF |
| | 86 | 32 | 180000-187FFF |
| Bank 10 | 87 | 32 | 178000-17FFFF |
| | 88 | 32 | 170000-177FFF |
| | 89 | 32 | 168000-16FFFF |
| | 90 | 32 | 160000-167FFF |
| | 91 | 32 | 158000-15FFFF |
| | 92 | 32 | 150000-157FFF |
| | 93 | 32 | 148000-14FFFF |
| | 94 | 32 | 140000-147FFF |

Table 29. Top boot block addresses, M58WR064HT (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 11 | 95 | 32 | 138000-13FFFF |
| | 96 | 32 | 130000-137FFF |
| | 97 | 32 | 128000-12FFFF |
| | 98 | 32 | 120000-127FFF |
| | 99 | 32 | 118000-11FFFF |
| | 100 | 32 | 110000-117FFF |
| | 101 | 32 | 108000-10FFFF |
| | 102 | 32 | 100000-107FFF |
| Bank 12 | 103 | 32 | 0F8000-0FFFFF |
| | 104 | 32 | 0F0000-0F7FFF |
| | 105 | 32 | 0E8000-0EFFFF |
| | 106 | 32 | 0E0000-0E7FFF |
| | 107 | 32 | 0D8000-0DFFFF |
| | 108 | 32 | 0D0000-0D7FFF |
| | 109 | 32 | 0C8000-0CFFFF |
| | 110 | 32 | 0C0000-0C7FFF |
| Bank 13 | 111 | 32 | 0B8000-0BFFFF |
| | 112 | 32 | 0B0000-0B7FFF |
| | 113 | 32 | 0A8000-0AFFFF |
| | 114 | 32 | 0A0000-0A7FFF |
| | 115 | 32 | 098000-09FFFF |
| | 116 | 32 | 090000-097FFF |
| | 117 | 32 | 088000-08FFFF |
| | 118 | 32 | 080000-087FFF |
| Bank 14 | 119 | 32 | 078000-07FFFF |
| | 120 | 32 | 070000-077FFF |
| | 121 | 32 | 068000-06FFFF |
| | 122 | 32 | 060000-067FFF |
| | 123 | 32 | 058000-05FFFF |
| | 124 | 32 | 050000-057FFF |
| | 125 | 32 | 048000-04FFFF |
| | 126 | 32 | 040000-047FFF |

**Table 29. Top boot block addresses, M58WR064HT (continued)**

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 15 | 127 | 32 | 038000-03FFFF |
| | 128 | 32 | 030000-037FFF |
| | 129 | 32 | 028000-02FFFF |
| | 130 | 32 | 020000-027FFF |
| | 131 | 32 | 018000-01FFFF |
| | 132 | 32 | 010000-017FFF |
| | 133 | 32 | 008000-00FFFF |
| | 134 | 32 | 000000-007FFF |

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

**Table 30. Bottom boot block addresses, M58WR064HB**

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 15 | 134 | 32 | 3F8000-3FFFFF |
| | 133 | 32 | 3F0000-3F7FFF |
| | 132 | 32 | 3E8000-3EFFFF |
| | 131 | 32 | 3E0000-3E7FFF |
| | 130 | 32 | 3D8000-3DFFFF |
| | 129 | 32 | 3D0000-3D7FFF |
| | 128 | 32 | 3C8000-3CFFFF |
| | 127 | 32 | 3C0000-3C7FFF |
| Bank 14 | 126 | 32 | 3B8000-3BFFFF |
| | 125 | 32 | 3B0000-3B7FFF |
| | 124 | 32 | 3A8000-3AFFFF |
| | 123 | 32 | 3A0000-3A7FFF |
| | 122 | 32 | 398000-39FFFF |
| | 121 | 32 | 390000-397FFF |
| | 120 | 32 | 388000-38FFFF |
| | 119 | 32 | 380000-387FFF |

Table 30. Bottom boot block addresses, M58WR064HB (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 13 | 118 | 32 | 378000-37FFFF |
| | 117 | 32 | 370000-377FFF |
| | 116 | 32 | 368000-36FFFF |
| | 115 | 32 | 360000-367FFF |
| | 114 | 32 | 358000-35FFFF |
| | 113 | 32 | 350000-357FFF |
| | 112 | 32 | 348000-34FFFF |
| | 111 | 32 | 340000-347FFF |
| Bank 12 | 110 | 32 | 338000-33FFFF |
| | 109 | 32 | 330000-337FFF |
| | 108 | 32 | 328000-32FFFF |
| | 107 | 32 | 320000-327FFF |
| | 106 | 32 | 318000-31FFFF |
| | 105 | 32 | 310000-317FFF |
| | 104 | 32 | 308000-30FFFF |
| | 103 | 32 | 300000-307FFF |
| Bank 11 | 102 | 32 | 2F8000-2FFFFF |
| | 101 | 32 | 2F0000-2F7FFF |
| | 100 | 32 | 2E8000-2EFFFF |
| | 99 | 32 | 2E0000-2E7FFF |
| | 98 | 32 | 2D8000-2DFFFF |
| | 97 | 32 | 2D0000-2D7FFF |
| | 96 | 32 | 2C8000-2CFFFF |
| | 95 | 32 | 2C0000-2C7FFF |
| Bank 10 | 94 | 32 | 2B8000-2BFFFF |
| | 93 | 32 | 2B0000-2B7FFF |
| | 92 | 32 | 2A8000-2AFFFF |
| | 91 | 32 | 2A0000-2A7FFF |
| | 90 | 32 | 298000-29FFFF |
| | 89 | 32 | 290000-297FFF |
| | 88 | 32 | 288000-28FFFF |
| | 87 | 32 | 280000-287FFF |

Table 30. Bottom boot block addresses, M58WR064HB (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 9 | 86 | 32 | 278000-27FFFF |
| | 85 | 32 | 270000-277FFF |
| | 84 | 32 | 268000-26FFFF |
| | 83 | 32 | 260000-267FFF |
| | 82 | 32 | 258000-25FFFF |
| | 81 | 32 | 250000-257FFF |
| | 80 | 32 | 248000-24FFFF |
| | 79 | 32 | 240000-247FFF |
| Bank 8 | 78 | 32 | 238000-23FFFF |
| | 77 | 32 | 230000-237FFF |
| | 76 | 32 | 228000-22FFFF |
| | 75 | 32 | 220000-227FFF |
| | 74 | 32 | 218000-21FFFF |
| | 73 | 32 | 210000-217FFF |
| | 72 | 32 | 208000-20FFFF |
| | 71 | 32 | 200000-207FFF |
| Bank 7 | 70 | 32 | 1F8000-1FFFFF |
| | 69 | 32 | 1F0000-1F7FFF |
| | 68 | 32 | 1E8000-1EFFFF |
| | 67 | 32 | 1E0000-1E7FFF |
| | 66 | 32 | 1D8000-1DFFFF |
| | 65 | 32 | 1D0000-1D7FFF |
| | 64 | 32 | 1C8000-1CFFFF |
| | 63 | 32 | 1C0000-1C7FFF |
| Bank 6 | 62 | 32 | 1B8000-1BFFFF |
| | 61 | 32 | 1B0000-1B7FFF |
| | 60 | 32 | 1A8000-1AFFFF |
| | 59 | 32 | 1A0000-1A7FFF |
| | 58 | 32 | 198000-19FFFF |
| | 57 | 32 | 190000-197FFF |
| | 56 | 32 | 188000-18FFFF |
| | 55 | 32 | 180000-187FFF |

Table 30. Bottom boot block addresses, M58WR064HB (continued)

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 5 | 54 | 32 | 178000-17FFFF |
| | 53 | 32 | 170000-177FFF |
| | 52 | 32 | 168000-16FFFF |
| | 51 | 32 | 160000-167FFF |
| | 50 | 32 | 158000-15FFFF |
| | 49 | 32 | 150000-157FFF |
| | 48 | 32 | 148000-14FFFF |
| | 47 | 32 | 140000-147FFF |
| Bank 4 | 46 | 32 | 138000-13FFFF |
| | 45 | 32 | 130000-137FFF |
| | 44 | 32 | 128000-12FFFF |
| | 43 | 32 | 120000-127FFF |
| | 42 | 32 | 118000-11FFFF |
| | 41 | 32 | 110000-117FFF |
| | 40 | 32 | 108000-10FFFF |
| | 39 | 32 | 100000-107FFF |
| Bank 3 | 38 | 32 | 0F8000-0FFFFF |
| | 37 | 32 | 0F0000-0F7FFF |
| | 36 | 32 | 0E8000-0EFFFF |
| | 35 | 32 | 0E0000-0E7FFF |
| | 34 | 32 | 0D8000-0DFFFF |
| | 33 | 32 | 0D0000-0D7FFF |
| | 32 | 32 | 0C8000-0CFFFF |
| | 31 | 32 | 0C0000-0C7FFF |
| Bank 2 | 30 | 32 | 0B8000-0BFFFF |
| | 29 | 32 | 0B0000-0B7FFF |
| | 28 | 32 | 0A8000-0AFFFF |
| | 27 | 32 | 0A0000-0A7FFF |
| | 26 | 32 | 098000-09FFFF |
| | 25 | 32 | 090000-097FFF |
| | 24 | 32 | 088000-08FFFF |
| | 23 | 32 | 080000-087FFF |

**Table 30. Bottom boot block addresses, M58WR064HB (continued)**

| Bank[1] | # | Size (KWord) | Address range |
|---|---|---|---|
| Bank 1 | 22 | 32 | 078000-07FFFF |
| | 21 | 32 | 070000-077FFF |
| | 20 | 32 | 068000-06FFFF |
| | 19 | 32 | 060000-067FFF |
| | 18 | 32 | 058000-05FFFF |
| | 17 | 32 | 050000-057FFF |
| | 16 | 32 | 048000-04FFFF |
| | 15 | 32 | 040000-047FFF |
| Parameter Bank | 14 | 32 | 038000-03FFFF |
| | 13 | 32 | 030000-037FFF |
| | 12 | 32 | 028000-02FFFF |
| | 11 | 32 | 020000-027FFF |
| | 10 | 32 | 018000-01FFFF |
| | 9 | 32 | 010000-017FFF |
| | 8 | 32 | 008000-00FFFF |
| | 7 | 4 | 007000-007FFF |
| | 6 | 4 | 006000-006FFF |
| | 5 | 4 | 005000-005FFF |
| | 4 | 4 | 004000-004FFF |
| | 3 | 4 | 003000-003FFF |
| | 2 | 4 | 002000-002FFF |
| | 1 | 4 | 001000-001FFF |
| | 0 | 4 | 000000-000FFF |

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

# Appendix B Common Flash interface

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

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

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

**Table 31. Query structure overview(1)**

| Offset | Sub-section name | Description |
|---|---|---|
| 00h | Reserved | Reserved for algorithm-specific information |
| 10h | CFI query identification string | Command set ID and algorithm data offset |
| 1Bh | System interface information | Device timing and voltage information |
| 27h | 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) |
| 80h | 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 *32*, *33*, *34* and *35*. Query data is always presented on the lowest order data outputs.

**Table 32. CFI query identification string**

| Offset | Sub-section name | Description | Value |
|---|---|---|---|
| 00h | 0020h | Manufacturer code | ST |
| 01h | 8810h<br>8811h | Device code | Top (M58WR064HT)<br>Bottom (M58WR064HB) |
| 02h | reserved | Reserved | |
| 03h | reserved | Reserved | |
| 04h-0Fh | reserved | Reserved | |
| 10h | 0051h | Query Unique ASCII String "QRY" | "Q" |
| 11h | 0052h | | "R" |
| 12h | 0059h | | "Y" |
| 13h | 0003h | Primary Algorithm Command Set and Control Interface ID code 16 bit ID code defining a specific algorithm | |
| 14h | 0000h | | |
| 15h | offset = P = 0039h | Address for Primary Algorithm extended Query table (see *Table 35*) | p = 39h |
| 16h | 0000h | | |
| 17h | 0000h | Alternate Vendor Command Set and Control Interface ID Code second vendor - specified algorithm supported | NA |
| 18h | 0000h | | |
| 19h | value = A = 0000h | Address for Alternate Algorithm extended Query table | NA |
| 1Ah | 0000h | | |

**Table 33. CFI query system interface information**

| Offset | Data | Description | Value |
|---|---|---|---|
| 1Bh | 0017h | $V_{DD}$ 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 |
| 1Ch | 0020h | $V_{DD}$ 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 |
| 1Dh | 00B4h | $V_{PP}$ [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 | 11.4 V |
| 1Eh | 00C6h | $V_{PP}$ [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 | 12.6 V |
| 1Fh | 0004h | Typical timeout per single byte/word program = $2^n$ µs | 16 µs |
| 20h | 0000h | Typical timeout for multi-byte programming = $2^n$ µs | NA |
| 21h | 000Ah | Typical timeout per individual block erase = $2^n$ ms | 1s |
| 22h | 0000h | Typical timeout for full chip erase = $2^n$ ms | NA |
| 23h | 0003h | Maximum timeout for word program = $2^n$ times typical | 128 µs |
| 24h | 0000h | Maximum timeout for multi-byte programming = $2^n$ times typical | NA |
| 25h | 0002h | Maximum timeout per individual block erase = $2^n$ times typical | 4 s |
| 26h | 0000h | Maximum timeout for chip erase = $2^n$ times typical | NA |

**Table 34. Device geometry definition**

| Offset word Mode | | Data | Description | Value |
|---|---|---|---|---|
| 27h | | 0017h | Device Size = $2^n$ in number of bytes | 8 MByte |
| 28h<br>29h | | 0001h<br>0000h | Flash Device Interface Code description | x16 async. |
| 2Ah<br>2Bh | | 0000h<br>0000h | Maximum number of bytes in multi-byte program or page = $2^n$ | NA |
| 2Ch | | 0002h | Number of identical sized erase block regions within the device bit 7 to 0 = x = number of Erase Block Regions | 2 |
| M58WR064HT | 2Dh<br>2Eh | 007Eh<br>0000h | Region 1 Information<br>Number of identical-size erase blocks = 007Eh+1 | 127 |
| | 2Fh<br>30h | 0000h<br>0001h | Region 1 Information<br>Block size in Region 1 = 0100h * 256 byte | 64 KByte |
| | 31h<br>32h | 0007h<br>0000h | Region 2 Information<br>Number of identical-size erase blocks = 0007h+1 | 8 |
| | 33h<br>34h | 0020h<br>0000h | Region 2 Information<br>Block size in Region 2 = 0020h * 256 byte | 8 KByte |
| | 35h<br>38h | reserved | Reserved for future erase block region information | NA |
| M58WR064HB | 2Dh<br>2Eh | 0007h<br>0000h | Region 1 Information<br>Number of identical-size erase block = 0007h+1 | 8 |
| | 2Fh<br>30h | 0020h<br>0000h | Region 1 Information<br>Block size in Region 1 = 0020h * 256 byte | 8 KByte |
| | 31h<br>32h | 007Eh<br>0000h | Region 2 Information<br>Number of identical-size erase block = 007Eh+1 | 127 |
| | 33h<br>34h | 0000h<br>0001h | Region 2 Information<br>Block size in Region 2 = 0100h * 256 byte | 64 KByte |
| | 35h<br>38h | reserved | Reserved for future erase block region information | NA |

**Table 35. Primary algorithm-specific extended query table**

| Offset | Data | Description | Value |
|---|---|---|---|
| (P)h = 39h | 0050h | Primary Algorithm extended Query table unique ASCII string “PRI” | "P" |
| | 0052h | | "R" |
| | 0049h | | "I" |
| (P+3)h = 3Ch | 0031h | Major version number, ASCII | "1" |
| (P+4)h = 3Dh | 0033h | Minor version number, ASCII | "3" |
| (P+5)h = 3Eh | 00E6h | Extended Query table contents for Primary Algorithm. Address (P+5)h contains less significant byte. | |
| | 0003h | | |
| (P+7)h = 40h | 0000h | bit 0Chip Erase supported(1 = Yes, 0 = No) | No |
| (P+8)h = 41h | 0000h | bit 1Erase Suspend supported(1 = Yes, 0 = No)<br>bit 2Program Suspend supported(1 = Yes, 0 = No)<br>bit 3Legacy Lock/Unlock supported(1 = Yes, 0 = No)<br>bit 4Queued Erase supported(1 = Yes, 0 = No)<br>bit 5Instant individual block locking supported(1 = Yes, 0 = No)<br>bit 6Protection bits supported(1 = Yes, 0 = No)<br>bit 7Page mode read supported(1 = Yes, 0 = No)<br>bit 8Synchronous read supported(1 = Yes, 0 = No)<br>bit 9Simultaneous operation supported(1 = Yes, 0 = No)<br>bit 10 to 31Reserved; undefined bits are ‘0’. If bit 31 is ’1’ then another 31 bit field of optional features follows at the end of the bit-30 field. | Yes<br>Yes<br>No<br>No<br>Yes<br>Yes<br>Yes<br>Yes<br>Yes |
| (P+9)h = 42h | 0001h | Supported Functions after Suspend<br>Read Array, Read Status Register and CFI Query<br>bit 0Program supported after Erase Suspend (1 = Yes, 0 = No)<br>bit 7 to 1Reserved; undefined bits are ‘0’ | Yes |
| (P+A)h = 43h | 0003h | Block Protect Status | |
| (P+B)h = 44h | 0000h | Defines which bits in the Block Status Register section of the Query are implemented.<br>bit 0Block protect Status Register Lock/Unlock bit active(1 = Yes, 0 = No)<br>bit 1Block Lock Status Register Lock-Down bit active (1 = Yes, 0 = No)<br>bit 15 to 2Reserved for future use; undefined bits are ‘0’ | Yes<br>Yes |
| (P+C)h = 45h | 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 | 1.8V |
| (P+D)h = 46h | 00C0h | $V_{PP}$ Supply Optimum Program/Erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100 mV | 12V |

**Table 36. Protection Register information**

| Offset | Data | Description | Value |
|---|---|---|---|
| (P+E)h = 47h | 0001h | Number of protection register fields in JEDEC ID space. 0000h indicates that 256 fields are available. | 1 |
| (P+F)h = 48h | 0080h | Protection Field 1: Protection Description | 0080h |
| (P+10)h = 49h | 0000h | Bits 0-7 Lower byte of protection register address | |
| (P+11)h = 4Ah | 0003h | Bits 8-15 Upper byte of protection register address<br>Bits 16-23 $2^n$ bytes in factory pre-programmed region | 8 Bytes |
| (P+12)h= 4Bh | 0004h | Bits 24-31 $2^n$ bytes in user programmable region | 16 Bytes |

**Table 37. Burst read information**

| Offset | Data | Description | Value |
|---|---|---|---|
| (P+13)h = 4Ch | 0003h | Page-mode read capability<br>bits 0-7'n' such that $2^n$ HEX value represents the number of read-page bytes. See offset 28h for device word width to determine page-mode data output width. | 8 Bytes |
| (P+14)h = 4Dh | 0004h | Number of synchronous mode read configuration fields that follow. | 4 |
| (P+15)h = 4Eh | 0001h | Synchronous mode read capability configuration 1<br>bit 3-7Reserved<br>bit 0-2'n' such that $2^{n+1}$ HEX value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width. A value of 07h indicates that the device is capable of continuous linear bursts that will output data until the internal burst counter reaches the end of the device's burstable address space. This field's 3-bit value can be written directly to the read configuration register bit 0-2 if the device is configured for its maximum word width. See offset 28h for word width to determine the burst data output width. | 4 |
| (P+16)h = 4Fh | 0002h | Synchronous mode read capability configuration 2 | 8 |
| (P+17)h = 50h | 0003h | Synchronous mode read capability configuration 3 | 16 |
| (P+18)h = 51h | 0007h | Synchronous mode read capability configuration 4 | Cont. |

**Table 38. Bank and erase block region information(1) (2)**

| Top Devices | | Bottom Devices | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| (P+19)h = 52h | 02h | (P+19)h = 52h | 02h | Number of Bank Regions within the device |

1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, see *Appendix A*, *Table 29* and *Table 30*.

**Table 39. Bank and erase block region 1 information(1)**

| M58WR064HT | | M58WR064HB | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| (P+1A)h = 53h | 0Fh | (P+1A)h = 53h | 01h | Number of identical banks within Bank Region 1 |
| (P+1B)h = 54h | 00h | (P+1B)h = 54h | 00h | |
| (P+1C)h = 55h | 11h | (P+1C)h = 55h | 11h | Number of program or erase operations allowed in region 1:<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |
| (P+1D)h = 56h | 00h | (P+1D)h = 56h | 00h | Number of program or erase operations allowed in other banks while a bank in same region is programming<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |
| (P+1E)h = 57h | 00h | (P+1E)h = 57h | 00h | Number of program or erase operations allowed in other banks while a bank in this region is erasing<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |
| (P+1F)h = 58h | 01h | (P+1F)h = 58h | 02h | Types of erase block regions in region 1<br>n = number of erase block regions with contiguous same-size erase blocks.<br>Symmetrically blocked banks have one blocking region.(2) |
| (P+20)h = 59h | 07h | (P+20)h = 59h | 07h | Bank Region 1 Erase Block Type 1 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n×256 = number of bytes in erase block region |
| (P+21)h = 5Ah | 00h | (P+21)h = 5Ah | 00h | |
| (P+22)h = 5Bh | 00h | (P+22)h = 5Bh | 20h | |
| (P+23)h = 5Ch | 01h | (P+23)h = 5Ch | 00h | |
| (P+24)h = 5Dh | 64h | (P+24)h = 5Dh | 64h | Bank Region 1 (Erase Block Type 1)<br>Minimum block erase cycles × 1000 |
| (P+25)h = 5Eh | 00h | (P+25)h = 5Eh | 00h | |
| (P+26)h = 5Fh | 01h | (P+26)h = 5Fh | 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 5Eh 01 5Eh 01 |
| (P+27)h = 60h | 03h | (P+27)h = 60h | 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 39. Bank and erase block region 1 information(1) (continued)**

| M58WR064HT | | M58WR064HB | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| | | (P+28)h = 61h | 06h | Bank Region 1 Erase Block Type 2 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n×256 = number of bytes in erase block region |
| | | (P+29)h = 62h | 00h | |
| | | (P+2A)h = 63h | 00h | |
| | | (P+2B)h = 64h | 01h | |
| | | (P+2C)h = 65h | 64h | Bank Region 1 (Erase Block Type 2)<br>Minimum block erase cycles × 1000 |
| | | (P+2D)h = 66h | 00h | |
| | | (P+2E)h = 67h | 01h | Bank Regions 1 (Erase Block Type 2): 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+2F)h = 68h | 03h | Bank Region 1 (Erase Block Type 2): Page mode and synchronous mode capabilities<br>Bit 0: Page-mode reads permitted<br>Bit 1: Synchronous reads permitted<br>Bit 2: Synchronous writes permitted<br>Bits 3-7: reserved |

1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, see *Appendix A*, *Table 29* and *Table 30*.

**Table 40. Bank and erase block region 2 information(1)**

| M58WR064HT | | M58WR064HB | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| (P+28)h = 61h | 01h | (P+30)h = 69h | 0Fh | Number of identical banks within bank region 2 |
| (P+29)h = 62h | 00h | (P+31)h = 6Ah | 00h | |
| (P+2A)h = 63h | 11h | (P+32)h = 6Bh | 11h | Number of program or erase operations allowed in bank region 2:<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |
| (P+2B)h = 64h | 00h | (P+33)h = 6Ch | 00h | Number of program or erase operations allowed in other banks while a bank in this region is programming<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |

Table 40. Bank and erase block region 2 information(1) (continued)

| M58WR064HT | | M58WR064HB | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| (P+2C)h = 65h | 00h | (P+34)h = 6Dh | 00h | Number of program or erase operations allowed in other banks while a bank in this region is erasing<br>Bits 0-3: Number of simultaneous program operations<br>Bits 4-7: Number of simultaneous erase operations |
| (P+2D)h = 66h | 02h | (P+35)h = 6Eh | 01h | Types of erase block regions in region 2<br>n = number of erase block regions with contiguous same-size erase blocks.<br>Symmetrically blocked banks have one blocking region.(2) |
| (P+2E)h = 67h | 06h | (P+36)h = 6Fh | 07h | Bank Region 2 Erase Block Type 1 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n×256 = number of bytes in erase block region |
| (P+2F)h = 68h | 00h | (P+37)h = 70h | 00h | |
| (P+30)h = 69h | 00h | (P+38)h = 71h | 00h | |
| (P+31)h = 6Ah | 01h | (P+39)h = 72h | 01h | |
| (P+32)h = 6Bh | 64h | (P+3A)h = 73h | 64h | Bank Region 2 (Erase Block Type 1)<br>Minimum block erase cycles × 1000 |
| (P+33)h = 6Ch | 00h | (P+3B)h = 74h | 00h | |
| (P+34)h = 6Dh | 01h | (P+3C)h = 75h | 01h | Bank Region 2 (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+35)h = 6Eh | 03h | (P+3D)h = 76h | 03h | Bank Region 2 (Erase Block Type 1): Page mode and synchronous mode capabilities (defined in *Table 37*)<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+36)h = 6Fh | 07h | | | Bank Region 2 Erase Block Type 2 Information<br>Bits 0-15: n+1 = number of identical-sized erase blocks<br>Bits 16-31: n×256 = number of bytes in erase block region |
| (P+37)h = 70h | 00h | | | |
| (P+38)h = 71h | 20h | | | |
| (P+39)h = 72h | 00h | | | |
| (P+3A)h = 73h | 64h | | | Bank Region 2 (Erase Block Type 2)<br>Minimum block erase cycles × 1000 |
| (P+3B)h = 74h | 00h | | | |
| (P+3C)h = 75h | 01h | | | Bank Region 2 (Erase Block Type 2): 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 |

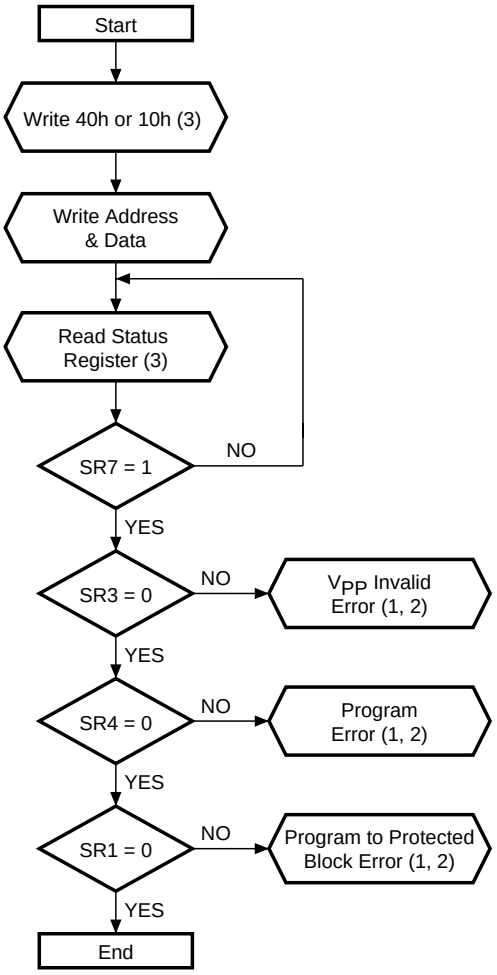
**Table 40. Bank and erase block region 2 information(1) (continued)**

| M58WR064HT | | M58WR064HB | | Description |
|---|---|---|---|---|
| Offset | Data | Offset | Data | |
| (P+3D)h = 76h | 03h | | | Bank Region 2 (Erase Block Type 2): Page mode and synchronous<br>mode capabilities (defined in *Table 37*)<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+3E)h = 77h | | (P+3E)h = 77h | | Feature Space definitions |
| (P+3F)h = 78h | | (P+3F)h = 78h | | Reserved |

1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, see *Appendix A*, *Table 29* and *Table 30*.

# Appendix C Flowcharts and pseudo codes

**Figure 19. Program flowchart and pseudo code**

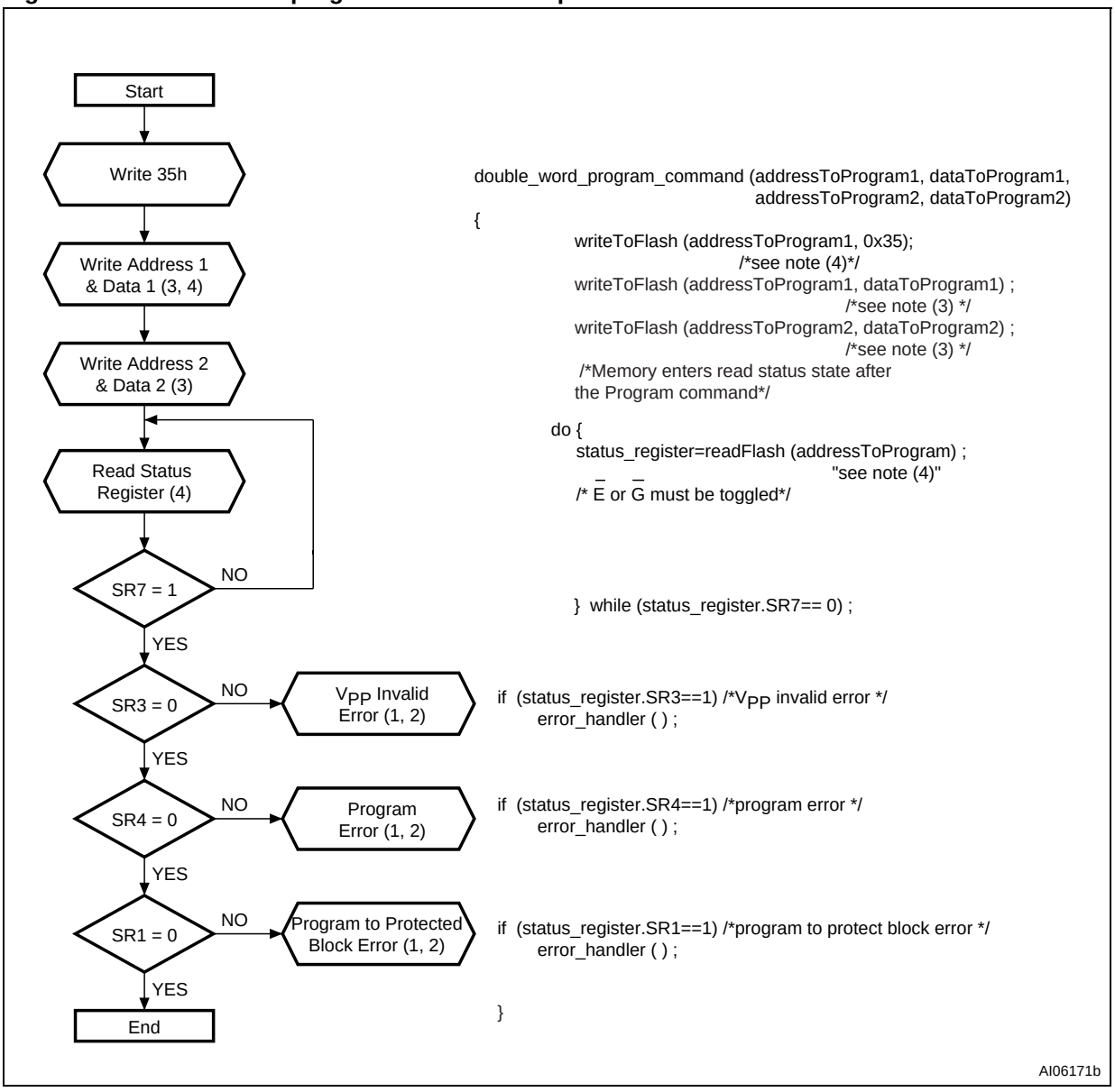



```
program_command (addressToProgram, dataToProgram) {:
        "
        writeToFlash (addressToProgram, 0x40);
        /*writeToFlash (addressToProgram, 0x10);*/
                                        /*see note (3)*/
        "
        writeToFlash (addressToProgram, dataToProgram) ;
        /*Memory enters read status state after
        the Program Command*/

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

        } while (status_register.SR7== 0) ;

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

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

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

}
```

AI06170b

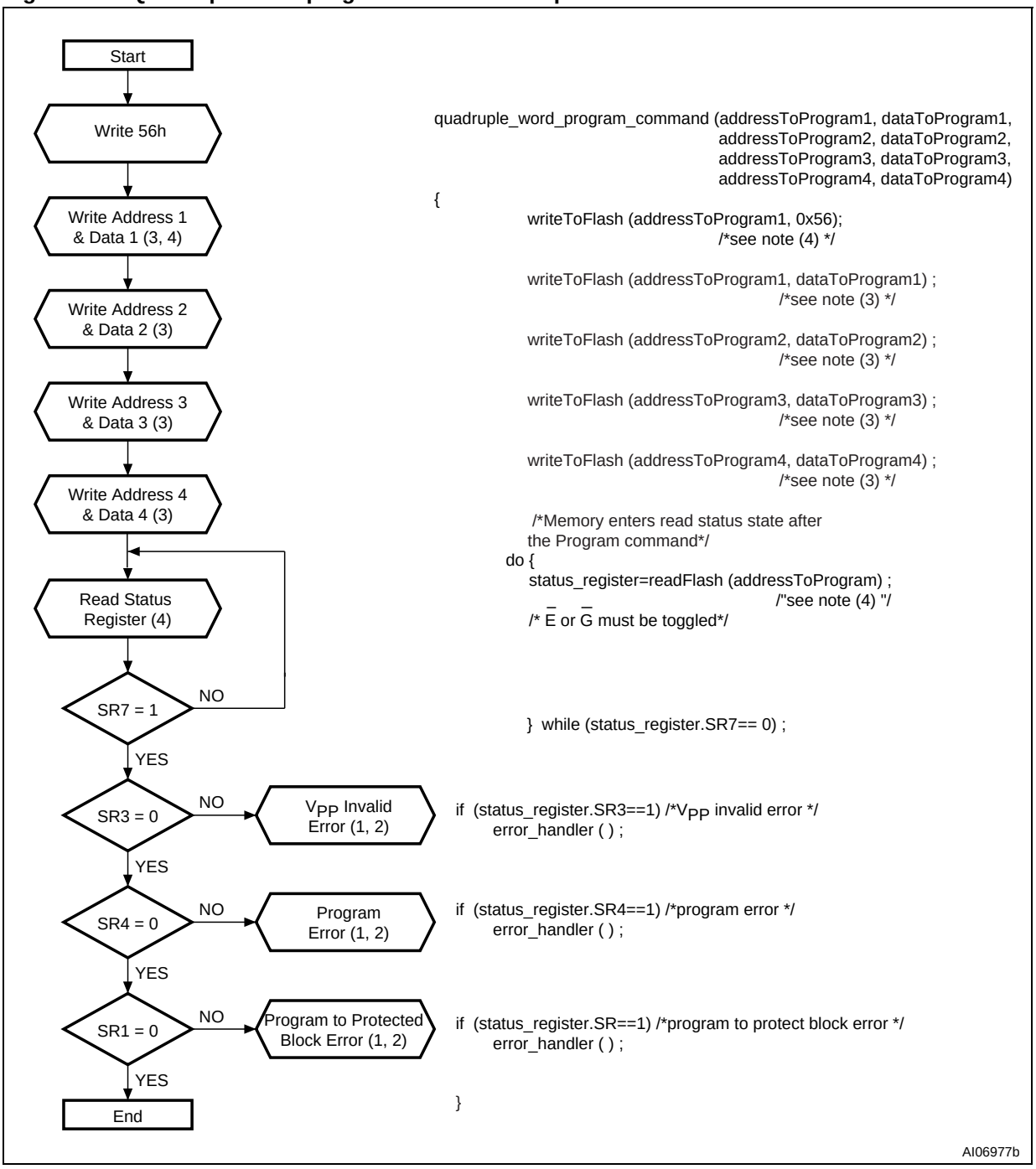
1. Status check of SR1 (Protected Block), SR3 ($V_{PP}$ Invalid) and SR4 (Program Error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase Controller operations.
3. Any address within the bank can equally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

**Figure 20. Double word program flowchart and pseudo code**



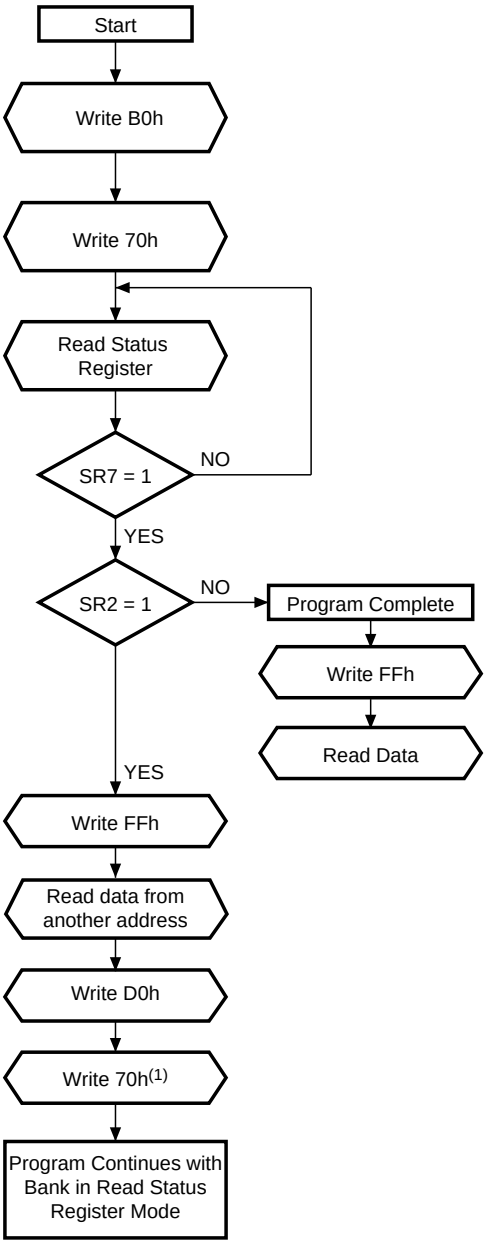

1. Status check of SR1 (Protected Block), SR3 ($V_{PP}$ Invalid) and SR4 (Program Error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase operations.
3. Address 1 and Address 2 must be consecutive addresses differing only for bit A0.
4. Any address within the bank can equally be used.

**Figure 21. Quadruple word program flowchart and pseudo code**




1. Status check of SR1 (Protected Block), SR3 ($V_{PP}$ Invalid) and SR4 (Program Error) can be made after each program operation or after a sequence.
2. If an error is found, the Status Register must be cleared before further Program/Erase operations.
3. Address 1 to Address 4 must be consecutive addresses differing only for bits A0 and A1.
4. Any address within the bank can equally be used.

**Figure 22. Program suspend and resume flowchart and pseudo code**

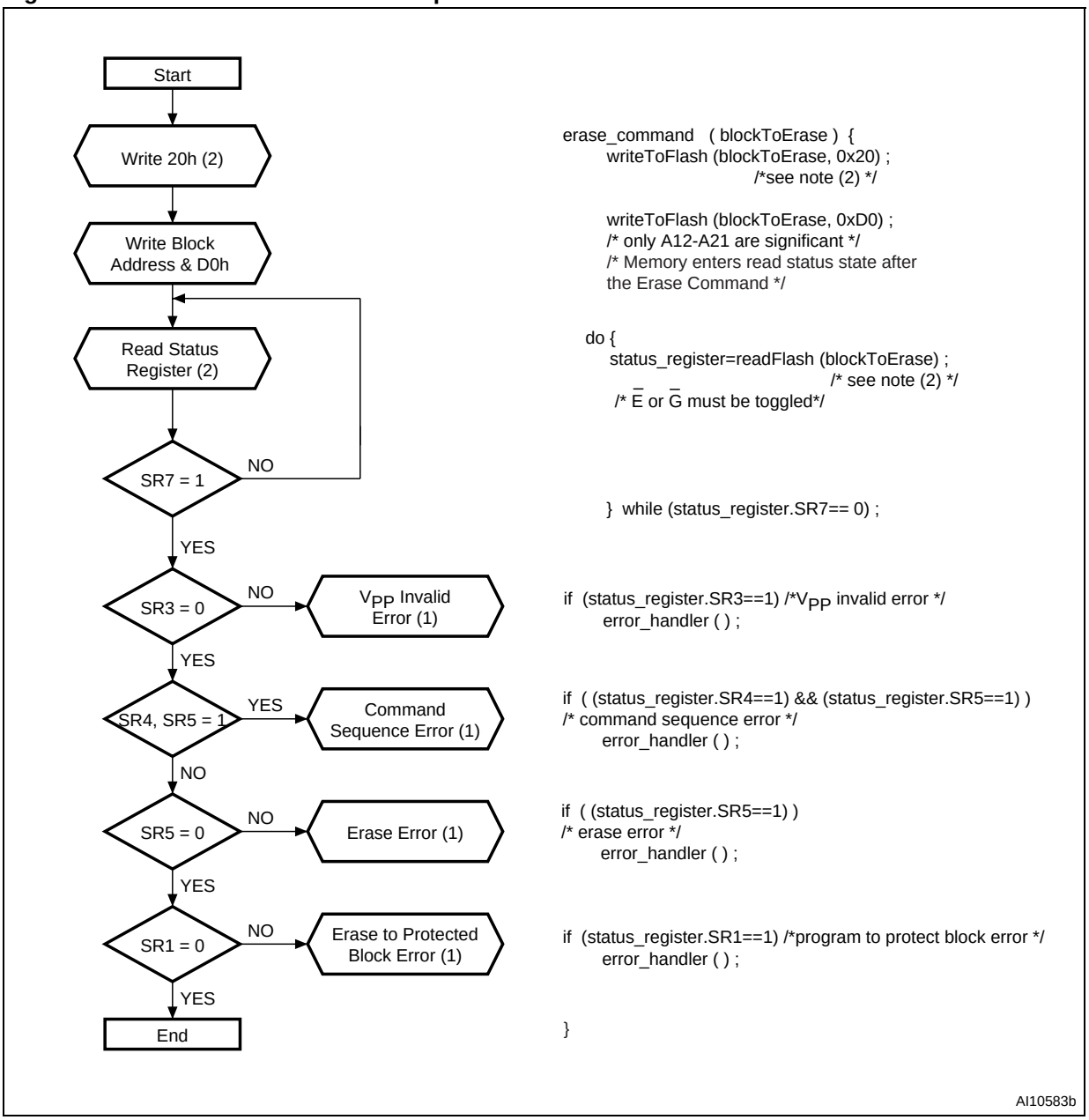



```
program_suspend_command   ( ) {
      writeToFlash (any_address, 0xB0) ;

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


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



      } while (status_register.SR7== 0) ;



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

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


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


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


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

       }
}
```

AI10117b

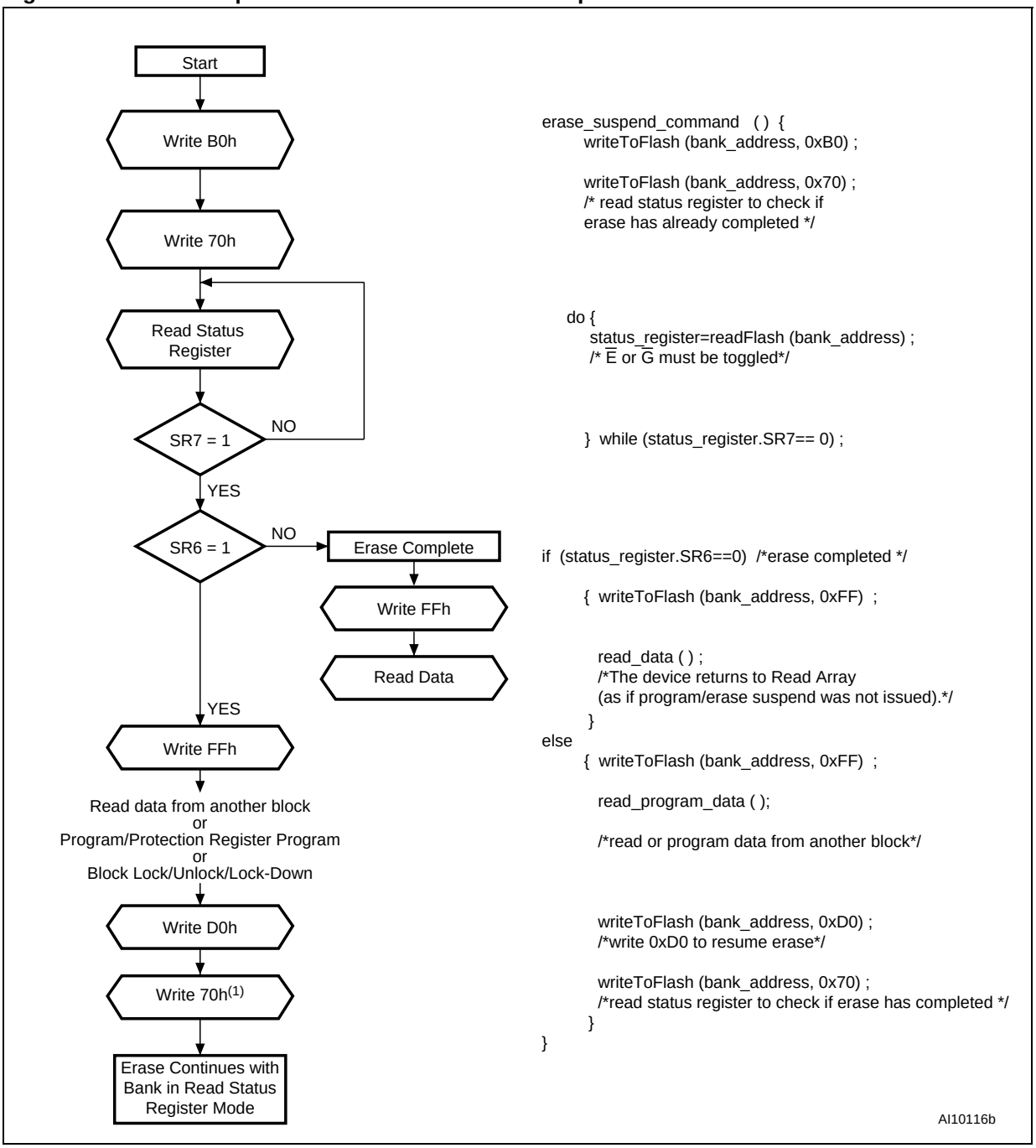
1. The Read Status Register command (Write 70h) can be issued just before or just after the Program Resume command.

**Figure 23. Block erase flowchart and pseudo code**



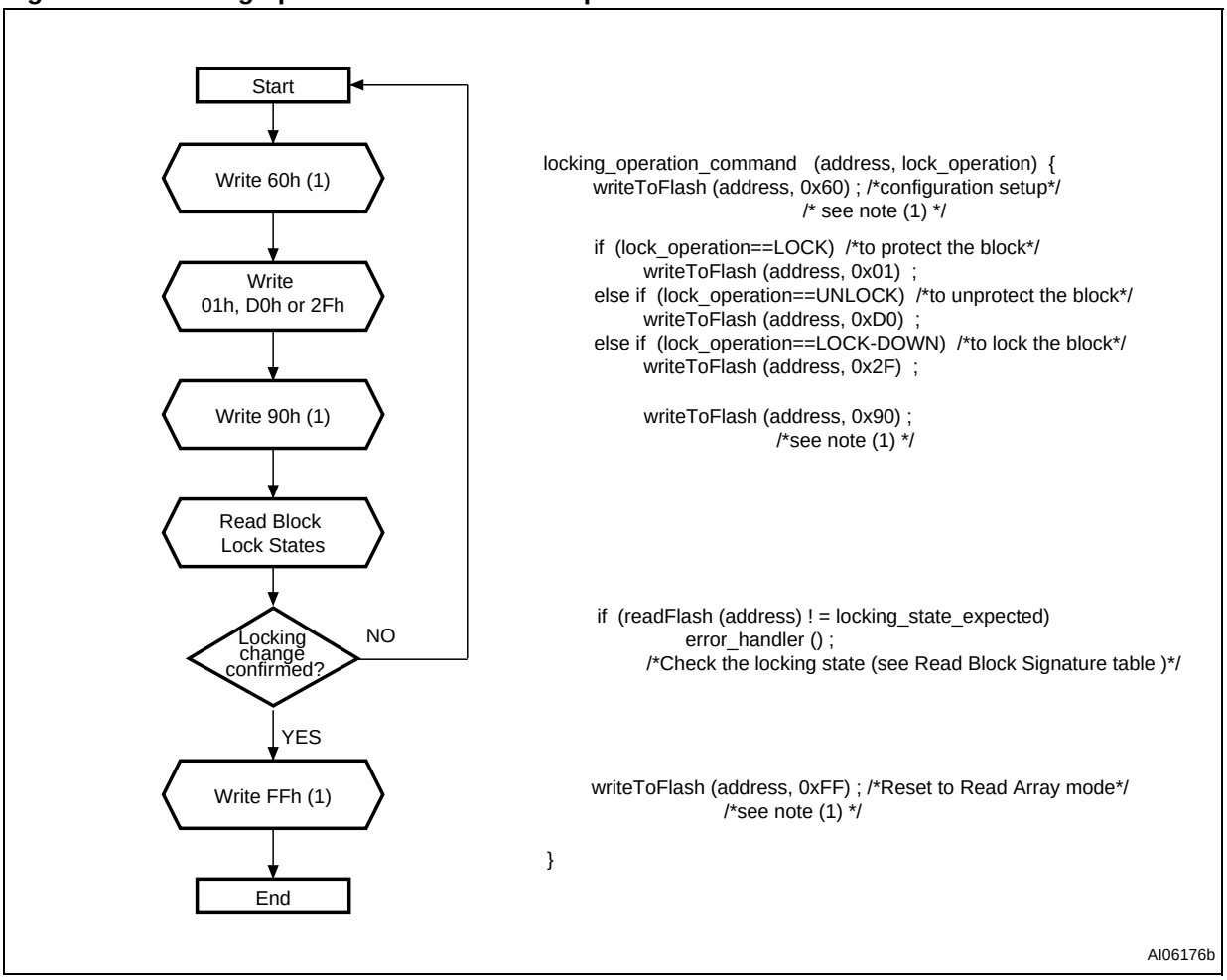

1. If an error is found, the Status Register must be cleared before further Program/Erase operations.
2. Any address within the bank can be used also.

**Figure 24. Erase suspend and resume flowchart and pseudo code**



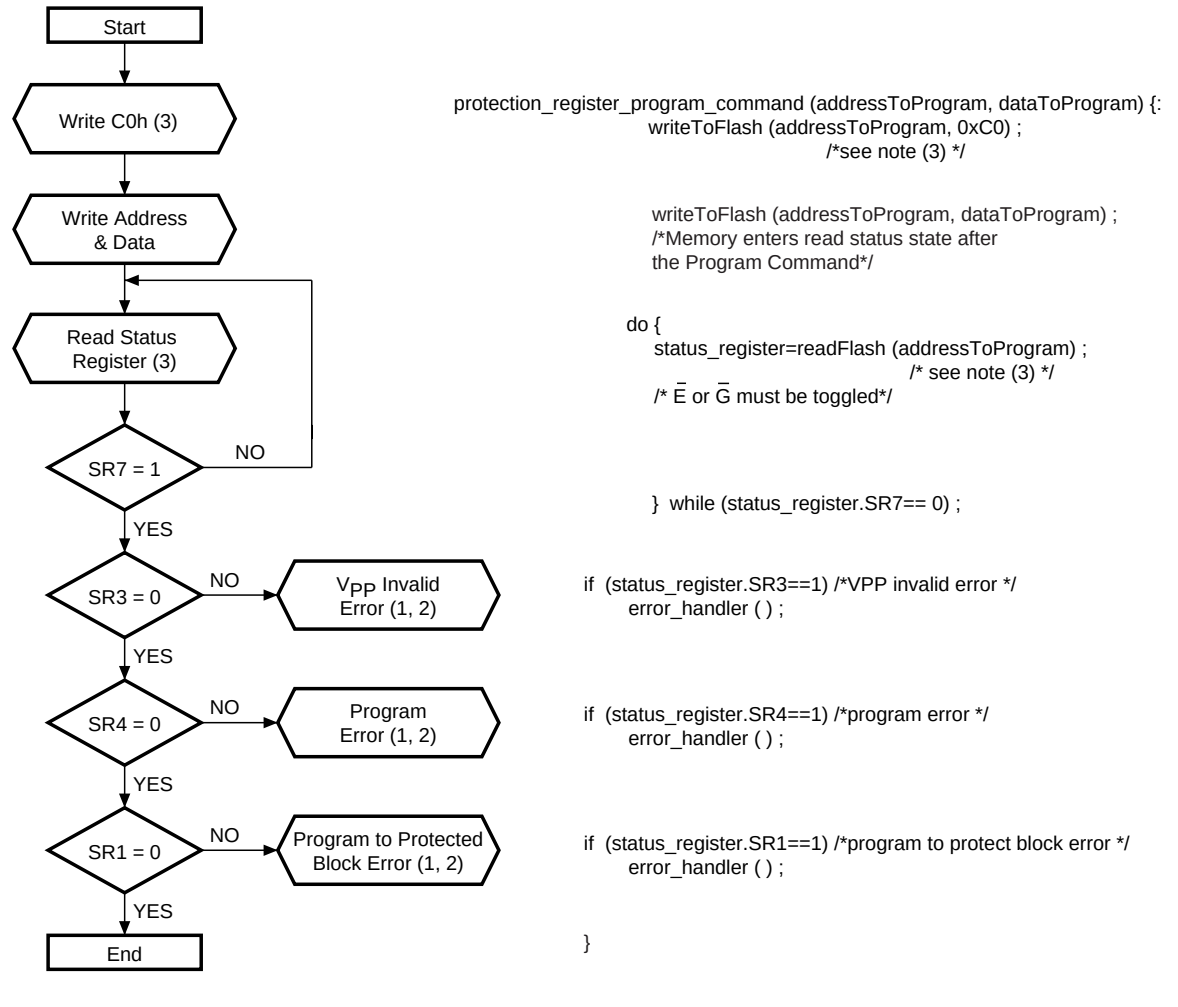

1. The Read Status Register command (Write 70h) can be issued just before or just after the Erase Resume command.

**Figure 25. Locking operations flowchart and pseudo code**



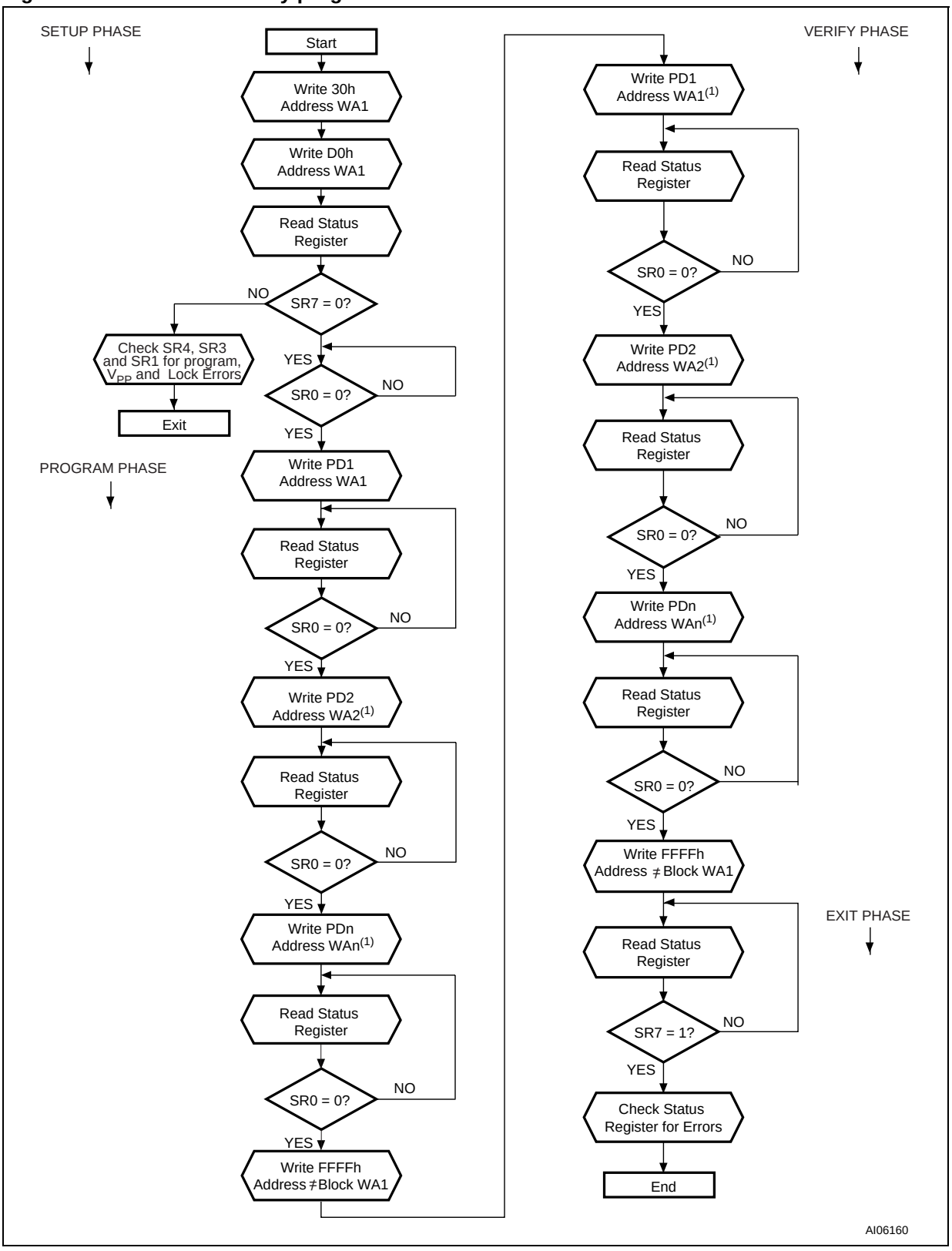

1. Any address within the bank can equally be used.

**Figure 26. Protection Register program flowchart and pseudo code**




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

**Figure 27. Enhanced factory program flowchart**

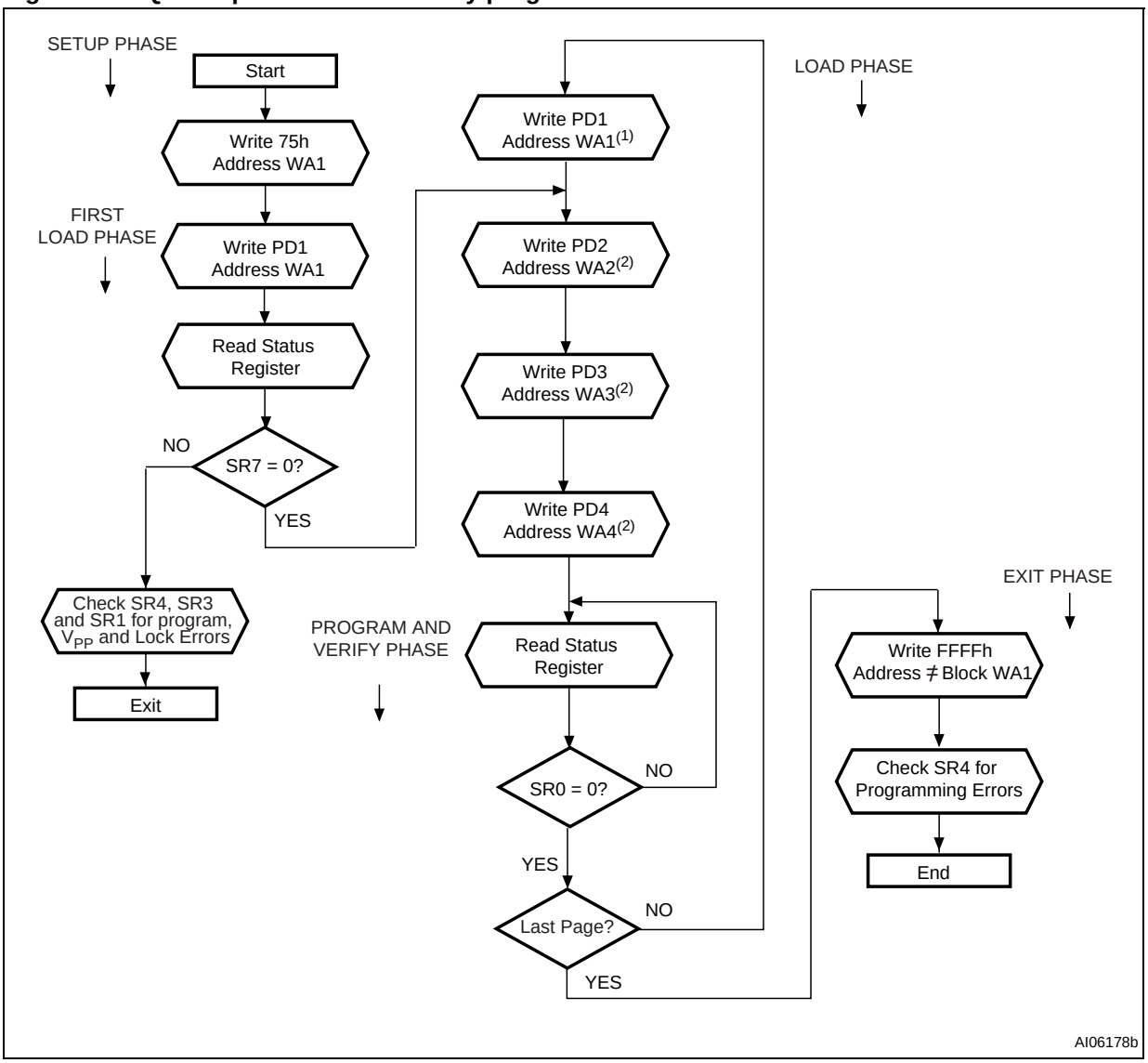



1. Address can remain Starting Address WA1 or be incremented.

# Enhanced factory program pseudo code

```
efp_command(addressFlow,dataFlow,n)
/* n is the number of data to be programmed */
{
   /* setup phase */
   writeToFlash(addressFlow[0],0x30);
   writeToFlash(addressFlow[0],0xD0);
   status_register=readFlash(any_address);
   if (status_register.SR7==1){
      /*EFP aborted for an error*/
      if (status_register.SR4==1) /*program error*/
         error_handler();
      if (status_register.SR3==1) /*VPP invalid error*/
         error_handler();
      if (status_register.SR1==1) /*program to protect block error*/
         error_handler();
   }
   else{
      /*Program Phase*/
      do{
         status_register=readFlash(any_address);
         /* E or G must be toggled*/
      } while (status_register.SR0==1)
      /*Ready for first data*/
      for (i=0; i++; i< n){
         writeToFlash(addressFlow[i],dataFlow[i]);
         /* status register polling*/
         do{
            status_register=readFlash(any_address);
            /* E or G must be toggled*/
         } while (status_register.SR0==1);
         /* Ready for a new data */
      }
      writeToFlash(another_block_address,FFFFh);

      /* Verify Phase */
      for (i=0; i++; i< n){
         writeToFlash(addressFlow[i],dataFlow[i]);
         /* status register polling*/
         do{
            status_register=readFlash(any_address);
            /* E or G must be toggled*/
         } while (status_register.SR0==1);
         /* Ready for a new data */
      }
      writeToFlash(another_block_address,FFFFh);
      /* exit program phase */

      /* Exit Phase */
      /* status register polling */
      do{
         status_register=readFlash(any_address);
         /* E or G must be toggled */
      } while (status_register.SR7==0);
      if (status_register.SR4==1) /*program failure error*/
         error_handler();
      if (status_register.SR3==1) /*VPP invalid error*/
         error_handler();
      if (status_register.SR1==1) /*program to protect block error*/
         error_handler();
   }
```

**Figure 28. Quadruple enhanced factory program flowchart**




1. Address can remain starting address WA1 (in which case the next page is programmed) or can be any address in the same block.
2. The address is only checked for the first word of each page as the order to program the words is fixed, so subsequent words in each page can be written to any address.

# Quadruple enhanced factory program pseudo code

```
quad_efp_command(addressFlow,dataFlow,n)
/* n is the number of pages to be programmed.*/
            {
      /* Setup phase */
   writeToFlash(addressFlow[0],0x75);
      for (i=0; i++; i< n){
                              /*Data Load Phase*/

      /*First Data*/
      writeToFlash(addressFlow[i],dataFlow[i,0]);
            /*at the first data of the first page, Quad-EFP may be aborted*/
      if (First_Page) {
                  status_register=readFlash(any_address);
                  if (status_register.SR7==1){
            /*EFP aborted for an error*/
                        if (status_register.SR4==1) /*program error*/
               error_handler();
                        if (status_register.SR3==1) /*VPP invalid error*/
               error_handler();
                        if (status_register.SR1==1) /*program to protect block
error*/
               error_handler();
         }
      }
      /*2nd data*/
            writeToFlash(addressFlow[i],dataFlow[i,1]);

      /*3rd data*/
                  writeToFlash(addressFlow[i],dataFlow[i,2]);

      /*4th data*/
            writeToFlash(addressFlow[i],dataFlow[i,3]);

      /* Program&Verify Phase */
      do{
         status_register=readFlash(any_address);
         /* E or G must be toggled*/
      }while (status_register.SR0==1)
   }
   /* Exit Phase */
   writeToFlash(another_block_address,FFFFh);
      /* status register polling */
   do{
      status_register=readFlash(any_address);
      /* E or G must be toggled */
   } while (status_register.SR7==0);
         if (status_register.SR1==1) /*program to protected block error*/
      error_handler();
      if (status_register.SR3==1) /*VPP invalid error*/
      error_handler();
      if (status_register.SR4==1) /*program failure error*/
      error_handler();
   }
}
```

# Appendix D Command interface state tables

**Table 41. Command interface states - modify table, next state(1)**

| Current CI State | | Command Input | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Read Array(2) (FFh) | WP setup(3)(4) (10/40h) | DWP, QWP Setup(3)(4) (35h, 56h) | Block Erase, Bank Erase Setup(3)(4) (20h, 80h) | EFP Setup (30h) | Quad-EFP Setup (75h) | Erase Confirm P/E Resume, Block Unlock confirm, EFP Confirm (D0h) | Program/ Erase Suspend (B0h) | Read Status Register (70h) | Clear status Register(5) (50h) | Read Electronic signature, Read CFI Query (90h, 98h) |
| **Ready** | | Ready | Program Setup | Program Setup | Erase Setup | EFP Setup | Quad-EFP Setup | Ready | | | | |
| **Lock/CR Setup** | | Ready (Lock Error) | | | | | | Ready | Ready (Lock Error) | | | |
| **OTP** | **Setup** | OTP Busy | | | | | | | | | | |
| | **Busy** | | | | | | | | | | | |
| **Program** | **Setup** | Program Busy | | | | | | | | | | |
| | **Busy** | Program Busy | | | | | | | Program Suspended | Program Busy | | |
| | **Suspend** | Program Suspended | | | | | | Program Busy | Program Suspended | | | |
| **Erase** | **Setup** | Ready (error) | | | | | | Erase Busy | Ready (error) | | | |
| | **Busy** | Erase Busy | | | | | | | Erase Suspended | Erase Busy | | |
| | **Suspend** | Erase Suspended | Program in Erase Suspend | Erase Suspended | | | | Erase Busy | Erase Suspended | | | |
| **Program in Erase Suspend** | **Setup** | Program Busy in Erase Suspend | | | | | | | | | | |
| | **Busy** | Program Busy in Erase Suspend | | | | | | | Program Suspend in Erase Suspend | Program Busy in Erase Suspend | | |
| | **Suspend** | Program Suspend in Erase Suspend | | | | | | Program Busy in Erase Suspend | Program Suspend in Erase Suspend | | | |
| **Lock/CR Setup in Erase Suspend** | | Erase Suspend (Lock Error) | | | | | | Erase Suspend | Erase Suspend (Lock Error) | | | |
| **EFP** | **Setup** | Ready (error) | | | | | | EFP Busy | Ready (error) | | | |
| | **Busy** | EFP Busy(6) | | | | | | | | | | |
| | **Verify** | EFP Verify(6) | | | | | | | | | | |
| **Quad EFP** | **Setup** | Quad EFP Busy(6) | | | | | | | | | | |
| | **Busy** | Quad EFP Busy(6) | | | | | | | | | | |

1. CI = Command Interface, CR = Configuration Register, EFP = Enhanced Factory Program, Quad EFP = Quadruple Enhanced Factory Program, DWP = Double Word Program, QWP = Quadruple Word Program, P/E. C. = Program/Erase Controller.
2. At Power-Up, all banks are in read array mode. A Read Array command issued to a busy bank, results in undetermined data output.
3. The two cycle command should be issued to the same bank address.
4. If the P/EC is active, both cycles are ignored.
5. The Clear Status Register command clears the Status Register error bits except when the P/EC is busy or suspended.
6. EFP and Quad EFP are allowed only when Status Register bit SR0 is set to ‘0’.EFP and Quad EFP are busy if Block Address is first EFP Address. Any other commands are treated as data.

**Table 42. Command interface states - modify table, next output(1)**

| Current CI State | Command Input(2) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Read Array(3) (FFh) | DWP, QWP Setup(4)(5) (35h, 56h) | Block Erase, Bank Erase Setup(4)(5) (20h, 80h) | EFP Setup (30h) | Quad-EFP Setup (75h) | Erase Confirm P/E Resume, Block Unlock confirm, EFP Confirm (D0h) | Program/ Erase Suspend (B0h) | Read Status Register (70h) | Clear status Register(6) (50h) | Read Electronic signature, Read CFI Query (90h, 98h) |
| Program Setup | Status Register | | | | | | | | | |
| Erase Setup | | | | | | | | | | |
| OTP Setup | | | | | | | | | | |
| Program in Erase Suspend | | | | | | | | | | |
| EFP Setup | | | | | | | | | | |
| EFP Busy | | | | | | | | | | |
| EFP Verify | | | | | | | | | | |
| Quad EFP Setup | | | | | | | | | | |
| Quad EFP Busy | | | | | | | | | | |
| Lock/CR Setup | | | | | | | | | | |
| Lock/CR Setup in Erase Suspend | | | | | | | | | | |
| OTP Busy | Array | Status Register | | | | Output Unchanged | | Status Register | Output Unchanged | Status Register |
| Ready | | | | | | | | | | Electronic Signature/CFI |
| Program Busy | | | | | | | | | | |
| Erase Busy | | | | | | | | | | |
| Program/Erase | | | | | | | | | | |
| Program Busy in Erase Suspend | | | | | | | | | | |
| Program Suspend in Erase Suspend | | | | | | | | | | |

1. CI = Command Interface, CR = Configuration Register, EFP = Enhanced Factory Program, Quad EFP = Quadruple Enhanced Factory Program, DWP = Double Word Program, QWP = Quadruple Word Program, P/E. C. = Program/Erase Controller.
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 Query mode, depending on the command issued. Each bank remains in its last output state until a new command is issued. The next state does not depend on the bank's output state.
3. At Power-Up, all banks are in read array mode. A Read Array command issued to a busy bank, results in undetermined data output.
4. The two cycle command should be issued to the same bank address.
5. If the P/EC is active, both cycles are ignored.
6. The Clear Status Register command clears the Status Register error bits except when the P/EC is busy or suspended.

**Table 43. Command interface states - lock table, next state(1)**

| Current CI State | | Command Input | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Lock/CR Setup(2) (60h) | OTP Setup(2) (C0h) | Block Lock Confirm (01h) | Block Lock-Down Confirm (2Fh) | Set CR Confirm (03h) | EFP Exit, Quad EFP Exit(3) | Illegal Command (4) | P/E. C. Operation Completed |
| Ready | | Lock/CR Setup | OTP Setup | Ready | | | | | N/A |
| Lock/CR Setup | | Ready (Lock error) | | Ready | | | Ready (Lock error) | | N/A |
| OTP | Setup | OTP Busy | | | | | | | N/A |
| | Busy | | | | | | | | Ready |
| Program | Setup | Program Busy | | | | | | | N/A |
| | Busy | Program Busy | | | | | | | Ready |
| | Suspend | Program Suspended | | | | | | | N/A |
| Erase | Setup | Ready (error) | | | | | | | N/A |
| | Busy | Erase Busy | | | | | | | Ready |
| | Suspend | Lock/CR Setup in Erase Suspend | Erase Suspended | | | | | | N/A |
| Program in Erase Suspend | Setup | Program Busy in Erase Suspend | | | | | | | N/A |
| | Busy | Program Busy in Erase Suspend | | | | | | | Erase Suspended |
| | Suspend | Program Suspend in Erase Suspend | | | | | | | N/A |
| Lock/CR Setup in Erase Suspend | | Erase Suspend (Lock error) | | Erase Suspend | | | Erase Suspend (Lock error) | | N/A |
| EFP | Setup | Ready (error) | | | | | | | N/A |
| | Busy | EFP Busy(5) | | | | | EFP Verify | EFP Busy(5) | N/A |
| | Verify | EFP Verify(5) | | | | | Ready | EFP Verify(5) | Ready |
| QuadEFP | Setup | Quad EFP Busy(5) | | | | | | | N/A |
| | Busy | Quad EFP Busy(5) | | | | | Ready | Quad EFP Busy(4) | Ready |

1. CI = Command Interface, CR = Configuration Register, EFP = Enhanced Factory Program, Quad EFP = Quadruple Enhanced Factory Program, P/E. C. = Program/Erase Controller.
2. If the P/EC is active, both cycles are ignored.
3. EFP and Quad EFP exit when Block Address is different from first Block Address and data is FFFFh.
4. Illegal commands are those not defined in the command set.
5. EFP and Quad EFP are allowed only when Status Register bit SR0 is set to '0'. EFP and Quad EFP are busy if Block Address is first EFP Address. Any other commands are treated as data.

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

| Current CI State | Command Input | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Lock/CR Setup(2) (60h) | OTP Setup(2) (C0h) | Block Lock Confirm (01h) | Block Lock-Down Confirm (2Fh) | Set CR Confirm (03h) | EFP Exit, Quad EFP Exit(3) | Illegal Command(4) | P/E. C. Operation Completed |
| Program Setup | Status Register | | | | | | | Output Unchanged |
| Erase Setup | | | | | | | | |
| OTP Setup | | | | | | | | |
| Program in Erase Suspend | | | | | | | | |
| EFP Setup | | | | | | | | |
| EFP Busy | | | | | | | | |
| EFP Verify | | | | | | | | |
| Quad EFP Setup | | | | | | | | |
| Quad EFP Busy | | | | | | | | |
| Lock/CR Setup | Status Register | | | | Array | Status Register | | |
| Lock/CR Setup in Erase Suspend | | | | | | | | |
| OTP Busy | Status Register | | Output Unchanged | | | Array | Output Unchanged | |
| Ready | | | | | | | | |
| Program Busy | | | | | | | | |
| EraseBusy | | | | | | | | |
| Program/Erase | | | | | | | | |
| Program Busy in Erase Suspend | | | | | | | | |
| Program Suspend in Erase Suspend | | | | | | | | |

1. CI = Command Interface, CR = Configuration Register, EFP = Enhanced Factory Program, Quad EFP = Quadruple Enhanced Factory Program, P/E. C. = Program/Erase Controller.
2. If the P/EC is active, both cycles are ignored.
3. EFP and Quad EFP exit when Block Address is different from first Block Address and data is FFFFh.
4. Illegal commands are those not defined in the command set.

# 17 Revision history

**Table 45. Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 12-Jan-2006 | 1 | Initial release. |
| 30-Mar-2006 | 2 | M58WR032HT and M58WR032HB part numbers removed.<br>First cycle address changed for Clear Status Register command in *Table 5: Standard commands*. Small text changes. |
| 04-Dec-2007 | 3 | Changed the $t_{GHWL}$ value from 14 and 17 to 5 ns for both speed classes in *Table 24: Write AC characteristics, Write Enable controlled*. |

**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.



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**