M58WR128ET
M58WR128EB

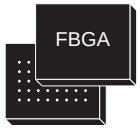
128 Mbit (8Mb x 16, Multiple Bank, Burst) 1.8V Supply Flash Memory

PRODUCT PREVIEW

FEATURES SUMMARY

- SUPPLY VOLTAGE
  - V_{DD} = 1.65V to 2.2V for Program, Erase and Read
  - V_{DDQ} = 1.65V to 3.3V for I/O Buffers
  - V_{PP} = 12V for fast Program (optional)
- SYNCHRONOUS / ASYNCHRONOUS READ
  - Synchronous Burst Read mode: 54MHz
  - Asynchronous/ Synchronous Page Read mode
  - Random Access: 70, 80, 100ns
- SYNCHRONOUS BURST READ SUSPEND
- PROGRAMMING TIME
  - 8µs 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
  - One parameter block permanently lockable
- COMMON FLASH INTERFACE (CFI)
- 100,000 PROGRAM/ERASE CYCLES per BLOCK

Figure 1. Package



VFBGA60 (ZB)
12.5 x 12mm

- ELECTRONIC SIGNATURE
  - Manufacturer Code: 20h
  - Top Device Code, M58WR128ET: 881Eh
  - Bottom Device Code, M58WR128EB: 881Fh

TABLE OF CONTENTS

SUMMARY DESCRIPTION . . . 7
Figure 2. Logic Diagram . . . 8
Table 1. Signal Names . . . 8
Table 2. Bank Architecture . . . 9
Figure 3. VFBGA Connections (Top view through package) . . . 9
Figure 4. Memory Map . . . 10

SIGNAL DESCRIPTIONS . . . 11
Address Inputs (A0-A22) . . . 11
Data Input/Output (DQ0-DQ15) . . . 11
Chip Enable (E) . . . 11
Output Enable (G) . . . 11
Write Enable (W) . . . 11
Write Protect (WP) . . . 11
Reset (RP) . . . 11
Latch Enable (L) . . . 11
Clock (K) . . . 11
Wait (WAIT) . . . 11
V_{DD} Supply Voltage . . . 11
V_{DDQ} Supply Voltage . . . 11
V_{PP} Program Supply Voltage . . . 11
V_{SS} Ground . . . 11
V_{SSQ} Ground . . . 11

BUS OPERATIONS . . . 12
Bus Read . . . 12
Bus Write . . . 12
Address Latch . . . 12
Output Disable . . . 12
Standby . . . 12
Reset . . . 12
Table 3. Bus Operations . . . 12

COMMAND INTERFACE . . . 13
Table 4. Command Codes . . . 13

COMMAND INTERFACE - STANDARD COMMANDS . . . 14
Read Array Command . . . 14
Read Status Register Command . . . 14
Read Electronic Signature Command . . . 14
Read CFI Query Command . . . 14
Clear Status Register Command . . . 14
Block Erase Command . . . 14
Bank Erase Command . . . 19
Program Command . . . 15

Program/Erase Suspend Command . . . 15
Program/Erase Resume Command . . . 15
Protection Register Program Command . . . 16
Set Configuration Register Command . . . 16
Block Lock Command . . . 16
Block Unlock Command . . . 16
Block Lock-Down Command . . . 16
Table 5. Standard Commands . . . 17
Table 6. Electronic Signature Codes . . . 18
Figure 5. Security Block and Protection Register Memory Map . . . 18

COMMAND INTERFACE - FACTORY PROGRAM COMMANDS . . . 19
Double Word Program Command . . . 19
Quadruple Word Program Command . . . 19
Enhanced Factory Program Command . . . 20
Setup Phase . . . 20
Program Phase . . . 20
Verify Phase . . . 20
Exit Phase . . . 21
Quadruple Enhanced Factory Program Command . . . 21
Setup Phase . . . 21
Load Phase . . . 21
Program and Verify Phase . . . 21
Exit Phase . . . 21
Table 7. Factory Program Commands . . . 22

STATUS REGISTER . . . 23
Program/Erase Controller Status Bit (SR7) . . . 23
Erase Suspend Status Bit (SR6) . . . 23
Erase Status Bit (SR5) . . . 23
Program Status Bit (SR4) . . . 23
V_{PP} Status Bit (SR3) . . . 23
Program Suspend Status Bit (SR2) . . . 24
Block Protection Status Bit (SR1) . . . 24
Bank Write/Multiple Word Program Status Bit (SR0) . . . 24
Table 8. Status Register Bits . . . 25

CONFIGURATION REGISTER . . . 26
Read Select Bit (CR15) . . . 26
X-Latency Bits (CR13-CR11) . . . 26
Wait Polarity Bit (CR10) . . . 26
Data Output Configuration Bit (CR9) . . . 26
Wait Configuration Bit (CR8) . . . 26
Burst Type Bit (CR7) . . . 26
Valid Clock Edge Bit (CR6) . . . 26
Wrap Burst Bit (CR3) . . . 26
Burst length Bits (CR2-CR0) . . . 27

Table 9. Configuration Register . . . 28
Table 10. Burst Type Definition . . . 29
Figure 6. X-Latency and Data Output Configuration Example . . . 31
Figure 7. Wait Configuration Example . . . 31

READ MODES . . . 32
Asynchronous Read Mode . . . 32
Synchronous Burst Read Mode . . . 32
Synchronous Burst Read Suspend . . . 33
Single Synchronous Read Mode . . . 33

DUAL OPERATIONS AND MULTIPLE BANK ARCHITECTURE . . . 34
Table 11. Dual Operations Allowed In Other Banks . . . 34
Table 12. Dual Operations Allowed In Same Bank . . . 34

BLOCK LOCKING . . . 35
Reading a Block's Lock Status . . . 35
Locked State . . . 35
Unlocked State . . . 35
Lock-Down State . . . 35
Locking Operations During Erase Suspend . . . 35
Table 13. Lock Status . . . 36

PROGRAM AND ERASE TIMES AND ENDURANCE CYCLES . . . 37
Table 14. Program, Erase Times and Program, Erase Endurance Cycles . . . 37

MAXIMUM RATING . . . 38
Table 15. Absolute Maximum Ratings . . . 38

DC and AC PARAMETERS . . . 39
Table 16. Operating and AC Measurement Conditions . . . 39
Figure 8. AC Measurement I/O Waveform . . . 39
Figure 9. AC Measurement Load Circuit . . . 39
Table 17. Capacitance . . . 39
Table 18. DC Characteristics - Currents . . . 40
Table 19. DC Characteristics - Voltages . . . 41
Figure 10. Asynchronous Random Access Read AC Waveforms . . . 42
Figure 11. Asynchronous Page Read AC Waveforms . . . 43
Table 20. Asynchronous Read AC Characteristics . . . 44
Figure 12. Synchronous Burst Read AC Waveforms . . . 45
Figure 13. Single Synchronous Read AC Waveforms . . . 46
Figure 14. Synchronous Burst Read Suspend AC Waveforms . . . 47
Figure 15. Clock input AC Waveform . . . 48
Table 21. Synchronous Read AC Characteristics . . . 48
Figure 16. Write AC Waveforms, Write Enable Controlled . . . 49
Table 22. Write AC Characteristics, Write Enable Controlled . . . 50

Figure 17. Write AC Waveforms, Chip Enable Controlled . . . 51
Table 23. Write AC Characteristics, Chip Enable Controlled . . . 52
Figure 18. Reset and Power-up AC Waveforms . . . 53
Table 24. Reset and Power-up AC Characteristics . . . 53

PACKAGE MECHANICAL . . . 54
Figure 19. VFBGA60 12.5x12mm - 8x7 ball array, 0.75mm pitch, Bottom View Package Outline . 54
Table 25. VFBGA60 12.5x12mm - 8x7 ball array, 0.75mm pitch, Package Mechanical Data. . . . 55
Figure 20. VFBGA60 Daisy Chain - Package Connections (Top view through package) . . . 56
Figure 21. VFBGA60 Daisy Chain - PCB Connection Proposal (Top view through package) . . . 57

PART NUMBERING . . . 58
Table 26. Ordering Information Scheme . . . 58
Table 27. Daisy Chain Ordering Scheme . . . 58

APPENDIX A. BLOCK ADDRESS TABLES . . . 59
Table 28. M58WR128ET - Parameter Bank Block Addresses . . . 60
Table 29. M58WR128ET -Main Bank Base Addresses. . . . 60
Table 30. M58WR128ET - Block Addresses in Main Banks . . . 60
Table 31. M58WR128EB - Parameter Bank Block Addresses . . . 61
Table 32. M58WR128EB - Main Bank Base Addresses . . . 61
Table 33. M58WR128EB - Block Addresses in Main Banks . . . 61

APPENDIX B. COMMON FLASH INTERFACE . . . 62
Table 34. Query Structure Overview . . . 62
Table 35. CFI Query Identification String . . . 62
Table 36. CFI Query System Interface Information . . . 63
Table 37. Device Geometry Definition . . . 64
Table 38. Primary Algorithm-Specific Extended Query Table . . . 65
Table 39. Protection Register Information . . . 66
Table 40. Burst Read Information . . . 66
Table 41. Bank and Erase Block Region Information . . . 66
Table 42. Bank and Erase Block Region 1 Information . . . 67
Table 43. Bank and Erase Block Region 2 Information . . . 68

APPENDIX C. FLOWCHARTS AND PSEUDO CODES. . . . 70
Figure 22. Program Flowchart and Pseudo Code . . . 70
Figure 23. Double Word Program Flowchart and Pseudo code . . . 71
Figure 24. Quadruple Word Program Flowchart and Pseudo Code . . . 72
Figure 25. Program Suspend & Resume Flowchart and Pseudo Code . . . 73
Figure 26. Block Erase Flowchart and Pseudo Code . . . 74
Figure 27. Erase Suspend & Resume Flowchart and Pseudo Code. . . . 75
Figure 28. Locking Operations Flowchart and Pseudo Code . . . 76
Figure 29. Protection Register Program Flowchart and Pseudo Code . . . 77
Figure 30. Enhanced Factory Program Flowchart . . . 78
Enhanced Factory Program Pseudo Code . . . 79

Figure 31. Quadruple Enhanced Factory Program Flowchart 80
Quadruple Enhanced Factory Program Pseudo Code 81

APPENDIX D. COMMAND INTERFACE STATE TABLES 82
Table 44. Command Interface States - Modify Table, Next State 82
Table 45. Command Interface States - Modify Table, Next Output 83
Table 46. Command Interface States - Lock Table, Next State 84
Table 47. Command Interface States - Lock Table, Next Output 85

REVISION HISTORY 86
Table 48. Document Revision History 86

SUMMARY DESCRIPTION

The M58WR128E is a 128 Mbit (8Mbit x16) non-volatile Flash memory that may be erased electrically at block level and programmed in-system on a Word-by-Word basis using a 1.65V to 2.2V V_{DD} supply for the circuitry and a 1.65V to 3.3V V_{DDQ} supply for the Input/Output pins. An optional 12V V_{PP} power supply is provided to speed up customer programming.

The device features an asymmetrical block architecture. M58WR128E has an array of 263 blocks, and is divided into 4 Mbit banks. There are 31 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 maps are shown in Figure 4. The Parameter Blocks are located at the top of the memory address space for the M58WR128ET, and at the bottom for the M58WR128EB.

Each block can be erased separately. Erase can be suspended, in order to perform 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 takes care of the timings necessary for program and erase operations. The end of a program or erase operation can be detected and any error conditions identified in the Status Register. The command set required to control the memory is consistent with JEDEC standards.

The device supports synchronous burst read and asynchronous read from all blocks of the memory array; at power-up the device is configured for asynchronous read. In synchronous burst mode, data is output on each clock cycle at frequencies of up to 54MHz. 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 M58WR128E features an instant, individual block locking scheme that allows any block to be locked or unlocked with no latency, enabling instant code and data protection. All blocks have three levels of protection. They can be locked and locked-down individually preventing any accidental programming or erasure. There is an additional hardware protection against program and erase. When $V_{PP} \leq V_{PPLK}$ all blocks are protected against program or erase. All blocks are locked at Power-Up.

The device includes a Protection Register and a Security Block 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. The Security Block, parameter block 0, can be permanently protected by the user. Figure 5, shows the Security Block and Protection Register Memory Map.

The memory is available in a VFBGA60 12.5x12mm package and is supplied with all the bits erased (set to '1').

Figure 2. Logic Diagram

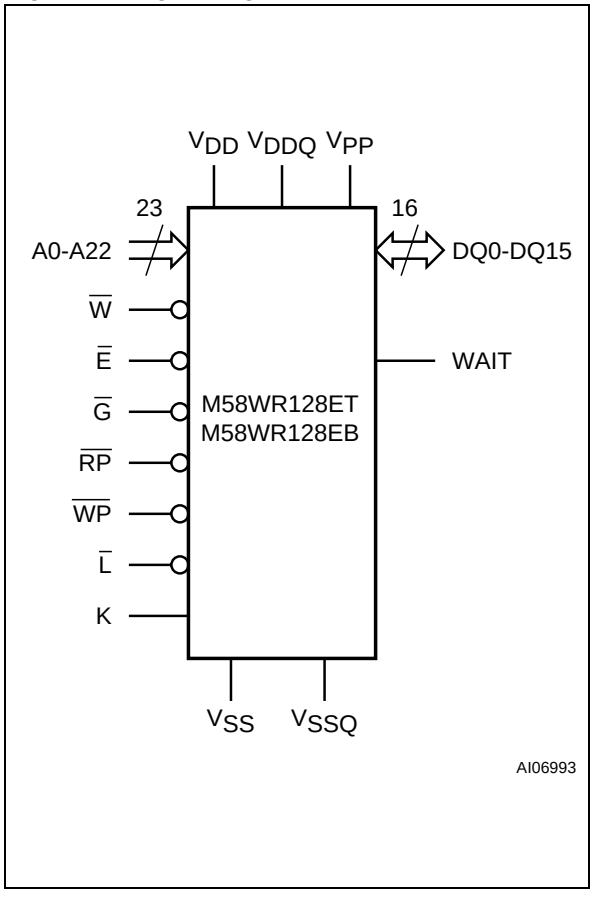


Table 1. Signal Names

A0-A22	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 & Erase
V_{SS}	Ground
V_{SSQ}	Ground Input/Output Supply
NC	Not Connected Internally
DU	Do Not Use

Figure 3. VFBGA Connections (Top view through package)

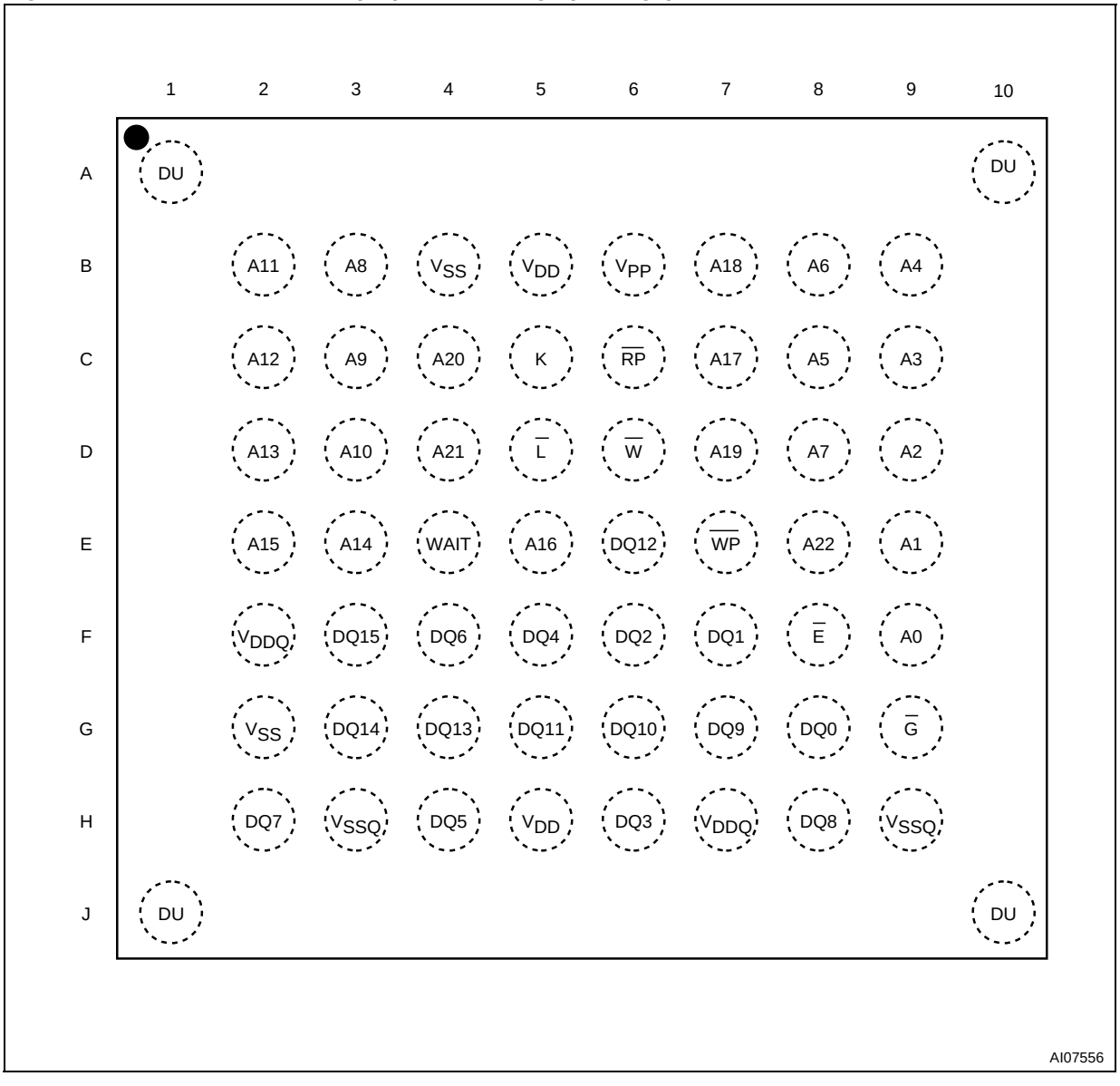
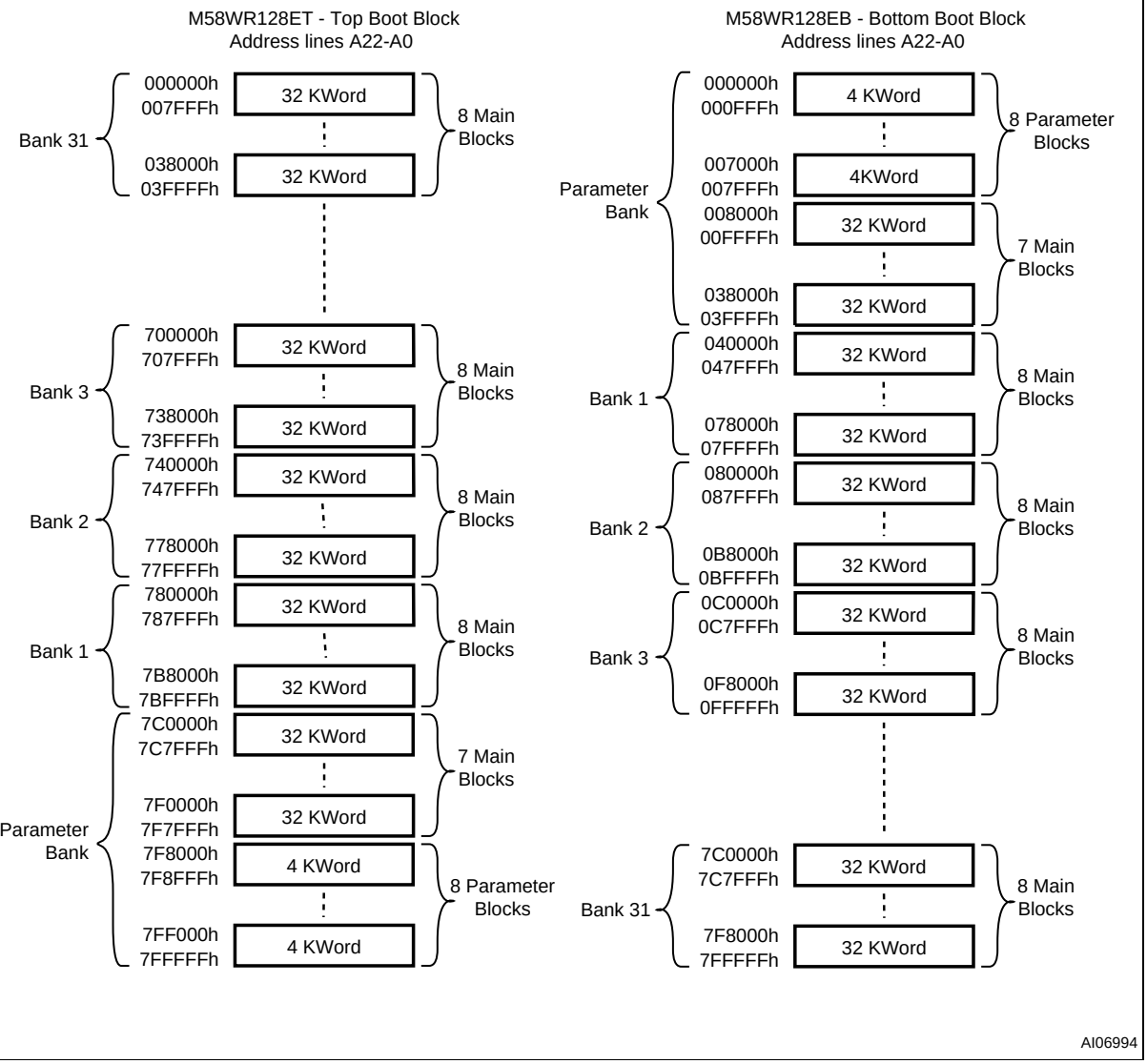


Table 2. Bank Architecture

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 30	4 Mbits	-	8 blocks of 32 KWords
Bank 31	4 Mbits	-	8 blocks of 32 KWords

Figure 4. Memory Map



SIGNAL DESCRIPTIONS

See Figure 2 Logic Diagram and Table 1,Signal Names, for a brief overview of the signals connected to this device.

Address Inputs (A0-A22). The Address Inputs select the cells in the memory array to access during Bus Read operations. During Bus Write operations they control the commands sent to the Command Interface of the internal state machine.

Data Input/Output (DQ0-DQ15). The Data I/O outputs the data stored at the selected address during a Bus Read operation or inputs a command or the data to be programmed during a Bus Write operation.

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 stand-by level.

Output Enable ($\overline{G}$). The Output Enable controls data outputs during the Bus Read operation of the memory.

Write Enable ($\overline{W}$). The Write Enable 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.

Write Protect ($\overline{WP}$). Write Protect is an input that gives an additional hardware protection for each block. When Write Protect is at V_{IL}, the Lock-Down is enabled and the protection status of the Locked-Down blocks cannot be changed. When Write Protect is at V_{IH}, the Lock-Down is disabled and the Locked-Down blocks can be locked or unlocked. (refer to Table 13, Lock Status).

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 18, 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. 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 3V logic without any additional circuitry. It can be tied to V_{RPH} (refer to Table 19, DC Characteristics).

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.

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.

Wait (WAIT). Wait is an output signal used during synchronous read to indicate whether the data on the output bus are valid. This output is high impedance when Chip Enable is at V_{IH} 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.

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).

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

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

If V_{PP} is kept in a low voltage range (0V to V_{DDQ}) V_{PP} is seen as a control input. In this case a voltage lower than V_{PPLK} gives an absolute protection against program or erase, while $V_{PP} > V_{PP1}$ enables these functions (see Tables 18 and 19, 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.

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

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

Note: Each device in a system should have V_{DD}, V_{DDQ} and V_{PP} decoupled with a 0.1µF ceramic capacitor close to the pin (high frequency, inherently low inductance capacitors should be as close as possible to the pack-

age). See Figure 9, AC Measurement Load Circuit. The PCB track widths should be sufficient to carry the required V_{PP} program and erase currents.

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 5ns on Chip Enable or Write Enable are ignored by the memory and do not affect Bus Write operations.

Bus Read. Bus Read operations are used to output the contents of the Memory Array, the Electronic Signature, the Status Register and the Common Flash Interface. Both Chip Enable and Output Enable must be at V_{IL} in order to perform a read operation. The Chip Enable input should be used to enable the device. Output Enable should be used to gate data onto the output. The data read depends on the previous command written to the memory (see Command Interface section). See Figures 10, 11, 12 and 13 Read AC Waveforms, and Tables 20 and 21 Read AC Characteristics, for details of when the output becomes valid.

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 16 and 17, Write AC Waveforms, and Tables 22 and 23, Write AC Characteristics, for details of the timing requirements.

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.

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

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.

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

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

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

COMMAND INTERFACE

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

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

Refer to Table 4, Command Codes and Appendix D, Tables 44, 45, 46 and 47, 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

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 following text descriptions.

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 will 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 will be executed but the output data is not guaranteed.

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.

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 will output 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).

If a Read Electronic Signature command is issued in a bank that is executing a Program or Erase operation the bank will go into Read Electronic Signature mode, subsequent Bus Read cycles will output the Electronic Signature data and the Program/Erase controller will continue 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.

Read CFI Query Command

The Read CFI Query command is used to read 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 will go into Read CFI Query mode, subsequent Bus Read cycles will output the CFI data and the Program/Erase controller will continue 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 11). 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.

See Appendix B, Common Flash Interface, Tables 34, 35, 36, 37, 38, 39, 40, 41, 42 and 43 for details on the information contained in the Common Flash Interface memory area.

Clear Status Register Command

The Clear Status Register command can be used to reset (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.

Block Erase Command

The Block Erase command can be used to erase a block. It sets all the bits within the selected block to '1'. All previous data in the block is lost. If the block is protected then the Erase operation will abort, the data in the block will not be changed and the Status Register will output 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 internal state machine and starts the Program/Erase Controller.

If the second bus cycle is not Write 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 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 will remain 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 will only accept the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and the Program/Erase Suspend command, all other commands will be ignored. Refer to Dual Operations section for detailed information about simultaneous operations allowed in banks not being erased. Typical Erase times are given in Table 14, Program, Erase Times and Program/Erase Endurance Cycles.

See Appendix C, Figure 26, Block Erase Flowchart and Pseudo Code, for a suggested flowchart for using the Block Erase command.

Program Command

The memory array can be programmed word-by-word. Only one Word in one bank can be programmed at any one time. 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 will only accept the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and the Program/Erase Suspend command. Refer to Dual Operations section for detailed information about simultaneous operations allowed in banks not being programmed. Typical Program times are given in Table 14, Program, Erase Times and Program/Erase 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 22, Program Flowchart and Pseudo Code, for the flowchart for using the Program command.

Program/Erase Suspend Command

The Program/Erase Suspend command is used to pause 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 will be set to '1'. The command can be addressed to any bank.

During Program/Erase Suspend the Command Interface will accept 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. Additionally, if the suspend operation was Erase then the Clear status Register, Program, Block Lock, Block Lock-Down or Block Unlock commands will also be 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 will complete. Refer to the Dual Operations section 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 25, Program Suspend & Resume Flowchart and Pseudo Code, and Figure 27, Erase Suspend & Resume Flowchart and Pseudo Code for flowcharts for using the Program/ Erase Suspend command.

Program/Erase Resume Command

The Program/Erase Resume command can be used to restart 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 will 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: suspend an erase operation, start a programming operation, suspend the

programming operation then read the array. See Appendix C, Figure 25, Program Suspend & Resume Flowchart and Pseudo Code, and Figure 27, Erase Suspend & Resume Flowchart and Pseudo Code for flowcharts for using the Program/Erase Resume command.

Protection Register Program Command

The Protection Register Program command is used to Program the 128 bit user One-Time-Programmable (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. Bit 1 of the Protection Lock Register also protects bit 2 of the Protection Lock Register. Programming bit 2 of the Protection Lock Register will result in a permanent protection of Parameter Block #0 (see Figure 5, Security Block and Protection Register Memory Map). Attempting to program a previously protected Protection Register will result in a Status Register error. The protection of the Protection Register and/or the Security Block is not reversible.

The Protection Register Program cannot be suspended. See Appendix C, Figure 29, Protection Register Program Flowchart and Pseudo Code, for a flowchart for using the Protection Register Program command.

Set Configuration Register Command

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

The Read mode of the banks is not modified when 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, etc.; the other address bits are ignored.

Block Lock Command

The Block Lock command is used to lock a block and prevent Program or Erase operations from changing the data in it. All blocks are locked 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. 13 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 the section, Block Locking, for a detailed explanation. See Appendix C, Figure 28, Locking Operations Flowchart and Pseudo Code, for a flowchart for using the Lock command.

Block Unlock Command

The Block Unlock command is used to unlock 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 13 shows the protection status after issuing a Block Unlock command. Refer to the section, Block Locking, for a detailed explanation and Appendix C, Figure 28, Locking Operations Flowchart and Pseudo Code, for a flowchart for using the Unlock command.

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. 13 shows the Lock Status after issuing a Block Lock-Down command. Refer to the section, Block Locking, for a detailed explanation and Appendix C, Figure 28, Locking Operations Flowchart and Pseudo Code, for a flowchart for using the Lock-Down command.

Table 5. Standard Commands

Commands	Cycles	Bus Operations					
		1st Cycle			2nd Cycle		
		Op.	Add	Data	Op.	Add	Data
Read Array	1+	Write	BKA	FFh	Read	WA	RD
Read Status Register	1+	Write	BKA	70h	Read	BKA(2)	SRD
Read Electronic Signature	1+	Write	BKA	90h	Read	BKA(2)	ESD
Read CFI Query	1+	Write	BKA	98h	Read	BKA(2)	QD
Clear Status Register	1	Write	BKA	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

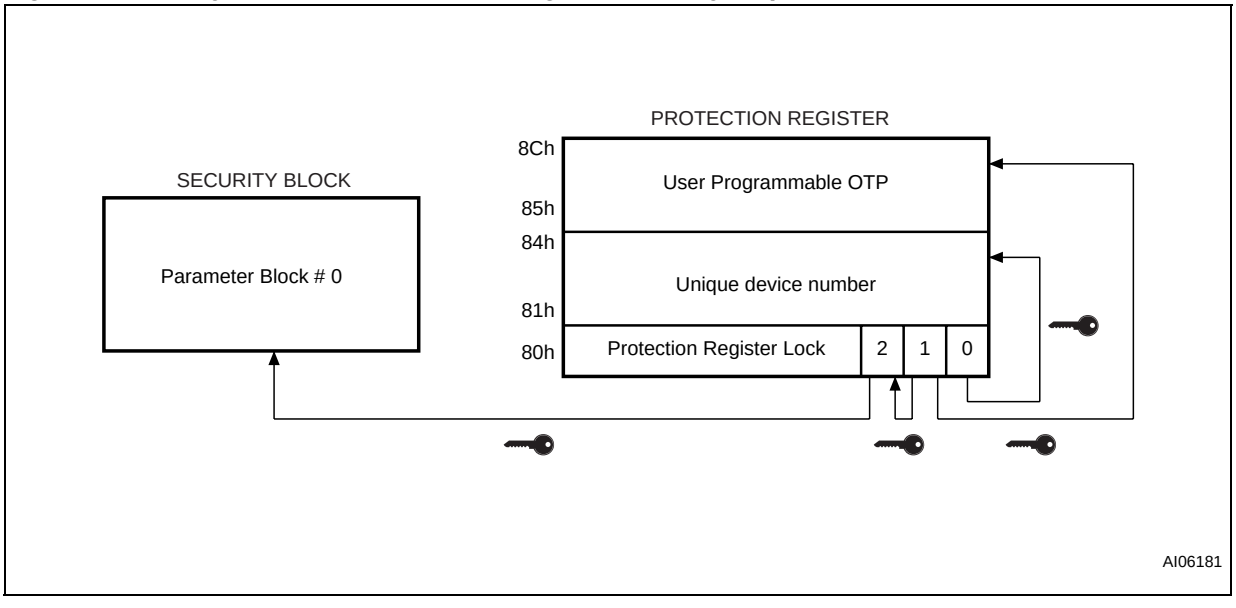
Note: 1. X = Don't Care, WA=Word Address in targeted bank, RD=Read Data, SRD=Status Register Data, ESD=Electronic Signature Data, QD=Query Data, BA=Block Address, BKA= Bank Address, PD=Program Data, PRA=Protection Register Address, PRD=Protection Register Data, CRD=Configuration Register Data.
2. Must 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	Bank Address + 01	881E
	Bottom	Bank Address + 01	881F
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
Protection Register Lock	ST Factory Default	Bank Address + 80	0006
	Security Block Permanently Locked		0002
	OTP Area Permanently Locked		0004
	Security Block and OTP Area Permanently Locked		0000
Protection Register		Bank Address + 81 Bank Address + 84	Unique Device Number
		Bank Address + 85 Bank Address + 8C	OTP Area

Note: CR=Configuration Register.

Figure 5. Security Block and Protection Register Memory Map



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 following text descriptions.

The use of the Factory Program commands require 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

Refer to Table 7, Factory Program Commands, in conjunction with the following text descriptions.

Bank Erase Command

The Bank Erase command can be used to erase 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 will ignore any protected blocks within the bank. If all blocks in the bank are protected then the Bank Erase operation will abort and the data in the bank will not be changed. The Status Register will 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 internal state machine and starts the Program/ Erase Controller.

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 will remain 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 will only accept the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query command, all other commands will be 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 will still operate 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 14, Program, Erase Times and Program/Erase Endurance Cycles.

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.

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 will only accept the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query command, all other commands will be ignored. Dual operations are not supported during Double Word Program operations and the command cannot be suspended. Typical Program times are given in Table 14, Program, Erase Times and Program/Erase 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 23, Double Word Program Flowchart and Pseudo Code, for the flowchart for using the Double Word Program command.

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 differ only for the addresses A0 and A1.

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.

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 will only accept the Read Array, Read Status Register, Read Electronic Signature and Read CFI Query command, all other commands will be ignored.

Dual operations are not supported during Quadruple Word Program operations and the command cannot be suspended. Typical Program times are given in Table 14, Program, Erase Times and Program/Erase Endurance Cycles.

See Appendix C, Figure 24, Quadruple Word Program Flowchart and Pseudo Code, for the flowchart for using the Quadruple Word Program command.

Enhanced Factory Program Command

The Enhanced Factory Program command can be used to program 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 will continue to work properly but some degradation in performance is possible. Typical Program times are given in Table 14.

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, Enhanced Factory Program Command and Figure 30, Enhanced Factory Program Flowchart.

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

Program Phase. The Program Phase requires n+1 cycles, where n is the number of Words (refer to Table 7, Enhanced Factory Program Command and Figure 30, 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. The Status Register Bank Write Status bit SR0 should be read to check that the P/E.C. 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/E.C. increments the address location or the address can be incremented in which case the P/E.C. jumps to the new address. If any address that is not in the same block as the Start Address is given with data FFFFh, 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/E.C. 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.

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 that is not in the same block as the Start Address is given with data FFFFh, the Verify Phase terminates. Status Register bit SR0 should be read to check that the P/E.C. 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 will be signaled in the Status Register.

Exit Phase. Status Register P/E.C. 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 the section on the Status Register for more details.

Quadruple Enhanced Factory Program Command

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

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

It 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.

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 as it will be interpreted as data to program.

Load Phase. The Load Phase requires 4 cycles to load the data (refer to Table 7, Factory Program Commands and Figure 31, 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. 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 7 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.

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 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.

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.

A full Status Register check should be done to ensure that the block has been sucessfully programmed. See the section on the Status Register for more details.

If the Program and Verify Phase has successfully completed the memory returns to Read mode. If the P/E.C. fails to program and reprogram a given

location, the error will be signaled in the Status Register.

Table 7. Factory Program Commands

Command	Phase	Cycles	Bus Write Operations									
			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(4)		3	BKA or WA1(8)	35h	WA1	PD1	WA2	PD2				
Quadruple Word Program(5)		5	BKA or WA1(8)	56h	WA1	PD1	WA2	PD2	WA3	PD3	WA4	PD4
Enhanced Factory Program (6)	Setup, Program	2+n +1	BKA or WA1(8)	30h	BA or WA1(9)	D0h	WA1(2)	PD1	WAn(3)	PAn	NOT WA1(2)	FFFFh
	Verify, Exit	n+1	WA1(2)	PD1	WA2(3)	PD2	WA3(3)	PD3	WAn(3)	PAn	NOT WA1(2)	FFFFh
Quadruple Enhanced Factory Program (5,6)	Setup, first Load	5	BKA or WA1(8)	75h	WA1(2)	PD1	WA2(7)	PD2	WA3(7)	PD3	WA4(7)	PD4
	First Program & Verify		Automatic									
	Subsequent Loads	4	WA1i (2)	PD1i	WA2i (7)	PD2i	WA3i (7)	PD3i			WA4i (7)	PD4i
	Subsequent Program & Verify		Automatic									
	Exit	1	NOT WA1 (2)	FFFFh								

Note: 1. WA=Word Address in targeted bank, BKA= Bank Address, PD=Program Data, BA=Block Address.
2. WA1 is the Start Address. NOT WA1 is any address that is not in the same block as WA1.
3. Address can remain Starting Address WA1 or be incremented.
4. Word Addresses 1 and 2 must be consecutive Addresses differing only for A0.
5. Word Addresses 1,2,3 and 4 must be consecutive Addresses differing only for A0 and A1.
6. 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.
7. 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.
8. Any address within the bank can be used.
9. Any address within the block can be used.

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 Read Status Register Command section 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.

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

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 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, 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.

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 rather than entering the Suspend mode.

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

Erase Status Bit (SR5). The Erase Status bit can be used to identify if the memory has failed to verify that the block or bank 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 or bank and still failed to verify that it has erased correctly. The Erase Status bit should be read once the Program/Erase Controller Status bit is 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 will appear to fail.

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 goes High (set to '1') only 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 will appear to fail.

V_{PP} Status Bit (SR3). The V_{PP} Status bit can be used to identify an invalid voltage on the V_{PP} pin during Program and Erase operations. The V_{PP} pin is only sampled at the beginning of a Program

or Erase operation. 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 will appear to fail.

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.

Block Protection Status Bit (SR1). The Block Protection Status bit can be used to identify if a Program or Block Erase operation has tried to modify the contents of a locked block.

When the Block Protection Status bit is High (set to '1'), a Program or Erase operation has been attempted on a locked 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 will appear to fail.

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 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.

Note: Refer to Appendix C, Flowcharts and Pseudo Codes, for using the Status Register.

Table 8. Status Register Bits

Bit	Name	Type	Logic Level	Definition	
SR7	P/E.C. 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	'0'	SR7 = '0'	Program or erase operation in addressed bank
				SR7 = '1'	No Program or erase operation in the device
			'1'	SR7 = '0'	Program or erase operation in a bank other than the addressed bank
				SR7 = '1'	Not Allowed
	Multiple Word Program Status (Enhanced Factory Program mode)	Status	'1'	SR7 = '0'	the device is NOT ready for the next word
				SR7 = '1'	Not Allowed
			'0'	SR7 = '0'	the device is ready for the next Word
				SR7 = '1'	the device is exiting from EFP

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

CONFIGURATION REGISTER

The Configuration Register is used to configure the type of bus access that the memory will perform. Refer to Read Modes section 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 9. They specify the selection of the burst length, burst type, burst X latency and the Read operation. Refer to Figures 6 and 7 for examples of synchronous burst configurations.

Read Select Bit (CR15)

The Read Select bit, CR15, is used to switch between asynchronous and synchronous Bus Read operations. When the Read Select bit is set to '1', read operations are asynchronous; when the Read Select bit is set to '0', read operations are synchronous. Synchronous Burst Read is supported in both parameter and main blocks and can be performed across banks.

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

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 9, Configuration Register.

The correspondence between X-Latency settings and the maximum sustainable frequency must be calculated taking into account some system parameters. Two conditions must be satisfied:

1. Depending on whether t_{AVK_CPU} or t_{DELAY} is supplied either one of the following two equations must be satisfied:

   $(n + 1)\ t_K \geq t_{ACC} - t_{AVK\_CPU} + t_{QVK\_CPU}$

   $(n + 2)\ t_K \geq t_{ACC} + t_{DELAY} + t_{QVK\_CPU}$

2. and also

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

where

n is the chosen X-Latency configuration code

t_K is the clock period

t_{AVK_CPU} is clock to address valid, $\overline{L}$ Low, or $\overline{E}$ Low, whichever occurs last

t_{DELAY} is address valid, $\overline{L}$ Low, or $\overline{E}$ Low to clock, whichever occurs last

t_{QVK_CPU} is the data setup time required by the system CPU,

t_{KQV} is the clock to data valid time

t_{ACC} is the random access time of the device.

Refer to Figure 6, X-Latency and Data Output Configuration Example.

Wait Polarity Bit (CR10)

In synchronous burst mode the Wait signal indicates whether the output data are 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 (default).

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, when the Data Output Configuration Bit 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 6, X-Latency and Data Output Configuration Example.

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 deasserted, 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' (default) the Wait output pin is asserted one clock cycle before the wait state.

Burst Type Bit (CR7)

The Burst Type bit is used to configure 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' (default) the memory outputs from sequential addresses. See Tables 10, Burst Type Definition, for the sequence of addresses output from a given starting address in each mode.

Valid Clock Edge Bit (CR6)

The Valid Clock Edge bit, CR6, is used to configure 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; when the Valid Clock Edge bit is '1' the rising edge of the Clock is active.

Wrap Burst Bit (CR3)

The burst reads can be confined inside the 4 or 8 Word boundary (wrap) or overcome the boundary

(no wrap). The Wrap Burst bit is used to select between wrap and no wrap. When the Wrap Burst bit is set to '0' the burst read wraps; when it is set to '1' the burst read does not wrap.

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, in 4, 8 words no-wrap, or in 16 words, 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 will be asserted for 1, 2 or 3 clock cycles when the burst sequence crosses the first 64 word boundary, or the 16 word boundary in the case of 16-word wrap burst, to indicate that the device needs an internal delay to read the successive words in the array. WAIT will be asserted only once during a continuous burst access. See also Table 10, Burst Type Definition.

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

Table 9. Configuration Register

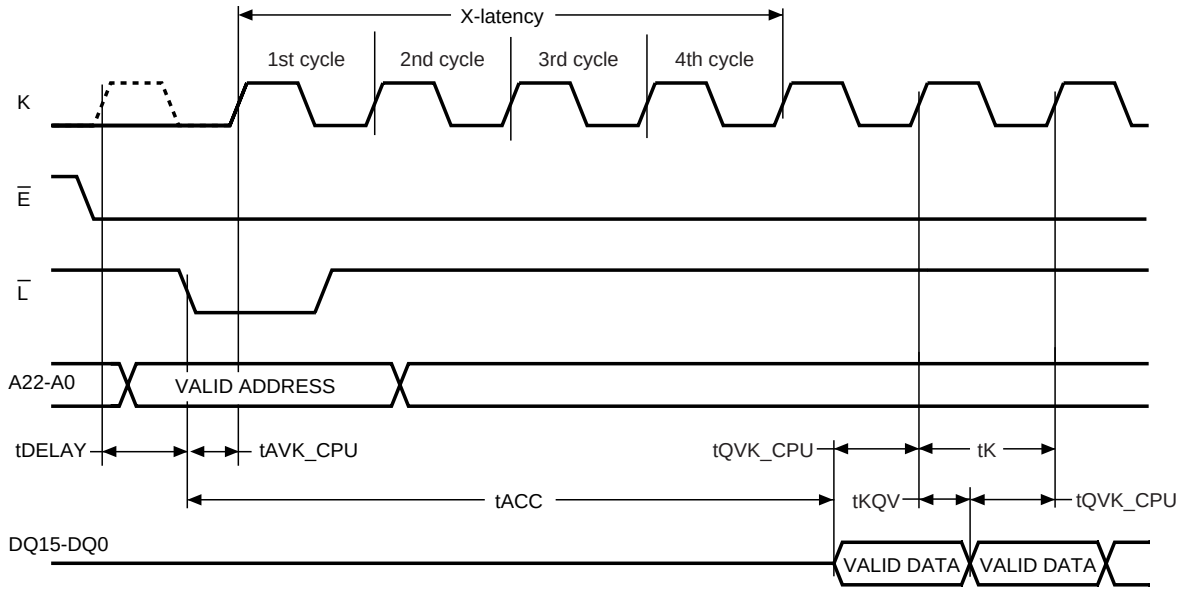
Bit	Description	Value	Description
CR15	Read Select	0	Synchronous Read
		1	Asynchronous Read (Default at power-on)
CR14			Reserved
CR13-CR11	X-Latency	010	2 clock latency
		011	3 clock latency
		100	4 clock latency
		101	5 clock latency
		111	Reserved (default)
		Other configurations reserved	
CR10	Wait 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 10. Burst Type Definition

Mode	Start Add.	4 Words		8 Words		16 Words		Continuous Burst
		Sequential	Inter-leaved	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-WAIT-0	1-0-3-2-5-4-7-6-9-8-11-10-13-12-15-14	1-2-3-4-5-6-7...
	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-WAIT-WAIT-0-1	2-3-0-1-6-7-4-5-10-11-8-9-14-15-12-13	2-3-4-5-6-7-8...
	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-WAIT-WAIT-WAIT-0-1-2	3-2-1-0-7-6-5-4-11-10-9-8-15-14-13-12	3-4-5-6-7-8-9...
	...							
	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-WAIT-WAIT-WAIT-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...
	...							
	60							60-61-62-63-64-65-66...
	61							61-62-63-WAIT-64-65-66...
	62							62-63-WAIT-WAIT-64-65-66...
	63							63-WAIT-WAIT-WAIT-64-65-66...

Mode	Start Add.	4 Words		8 Words		16 Words		Continuous Burst
		Sequential	Inter-leaved	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--9-10-11-12-13-14-15-16		
	2	2-3-4-5		2-3-4-5-6-7-8-9...		2-3-4-5--6-7-8-9-10-11-12-13-14-15-16-17		
	3	3-4-5-6		3-4-5-6-7-8-9-10		3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18		
	...							
	7	7-8-9-10		7-8-9-10-11-12-13-14		7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22		
	...							
	60	60-61-62-63		60-61-62-63-64-65-66-67		60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75		
	61	61-62-63-WAIT-64		61-62-63-WAIT-64-65-66-67-68		61-62-63-WAIT-64-65-66-67-68-69-70-71-72-73-74-75-76		
	62	62-63-WAIT-WAIT-64-65		62-63-WAIT-WAIT-64-65-66-67-68-69		62-63-WAIT-WAIT-64-65-66-67-68-69-70-71-72-73-74-75-76-77		
	63	63-WAIT-WAIT-WAIT-64-65-66		63-WAIT-WAIT-WAIT-64-65-66-67-68-69-70		63-WAIT-WAIT-WAIT-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78		

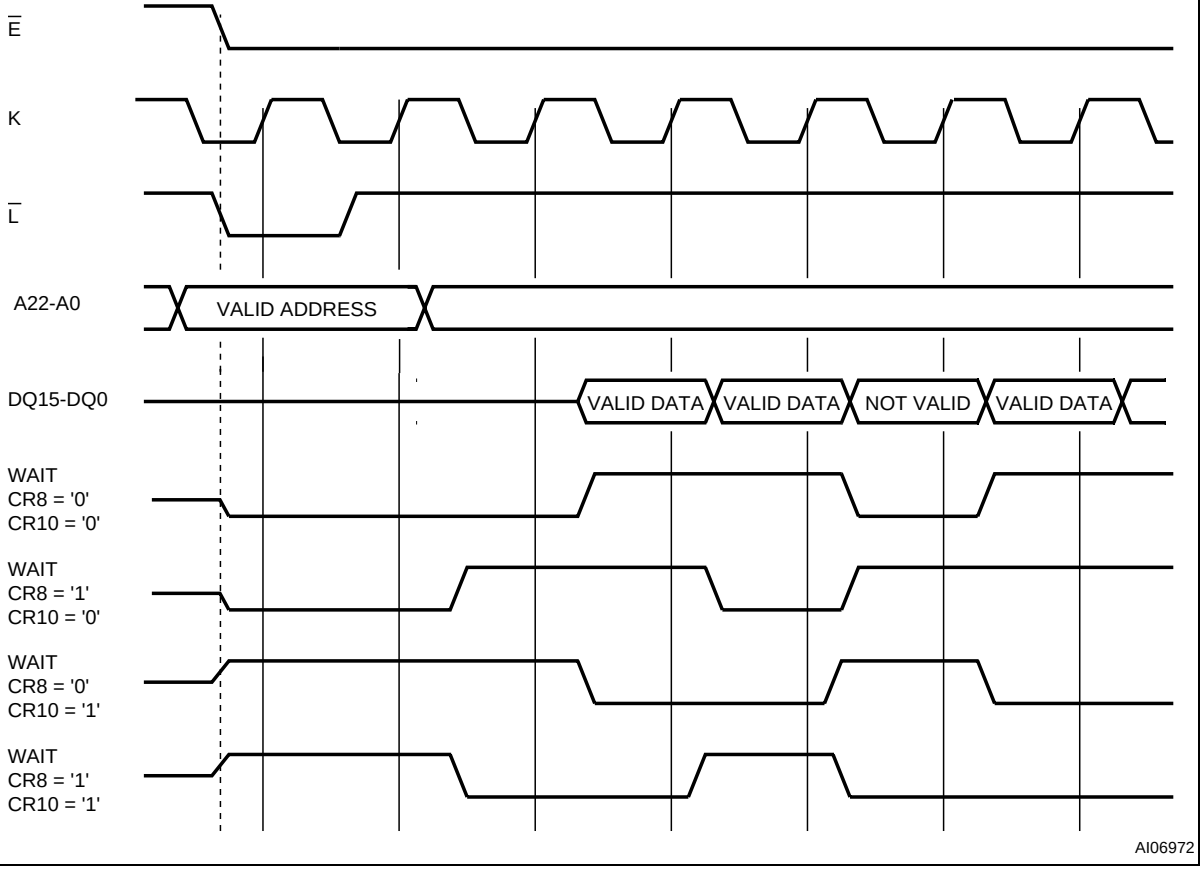
Figure 6. X-Latency and Data Output Configuration Example



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

AI06182

Figure 7. Wait Configuration Example



AI06972

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 Configuration Register section 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 11 and 12).

Asynchronous Read Mode

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

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), 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 150ns, the device automatically switches to the Automatic Standby mode. In this condition the power consumption is reduced to the standby value and the outputs are still driven.

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

See Table 20, Asynchronous Read AC Characteristics, Figure 10, Asynchronous Random Access Read AC Waveform and Figure 11, Asynchronous Page Read AC Waveform for details.

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 are output on each clock cycle.

The number of Words to be output during a Synchronous Burst Read operation can be configured as 4 or 8 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 or 8 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 will depend on the starting address of the burst sequence; the worst case delay will occur when the sequence is crossing a 64 word boundary and the starting address was at the end of a four word boundary.

WAIT is asserted during the X latency, the Wait state and at the end of 4- and 8-Word Burst. It is only deasserted when output data are valid. In Continuous Burst Read mode a Wait state will occur when crossing the first 64 Word boundary. If the burst starting address is aligned to a 4 Word Page, the Wait state will not occur.

The WAIT signal can be configured to be active Low or active High (default) 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 21, Synchronous Read AC Characteristics and Figure 12, Synchronous Burst Read AC Waveform for details.

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) in which case the initial latency time can be reduced to zero, 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 will not revert to high-impedance when $\overline{G}$ goes high. So 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 21, Synchronous Read AC Characteristics and Figure 14, Synchronous Burst Read Suspend AC Waveform for details.

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 21, Synchronous Read AC Characteristics and Figure 12, Single Synchronous Read AC Waveform for details.

DUAL OPERATIONS AND MULTIPLE BANK ARCHITECTURE

The Multiple Bank Architecture of the M58WR128E provides flexibility for software developers by allowing code and data to be split with 4Mbit 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, so 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.

Tables 11 and 12 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 11. 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 12. 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

Note: 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 not guaranteed until the Program or Erase has completed.

BLOCK LOCKING

The M58WR128E 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 13, defines all of the possible protection states ($\overline{WP}$, DQ1, DQ0), and Appendix C, Figure 28, shows a flowchart for the locking operations.

Reading a Block's Lock Status

The lock status of every block can be read in the Read Electronic Signature mode of the device. To enter this mode write 90h to the device. Subsequent reads at the address specified in Table 6, will 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.

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 will return 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.

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.

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.

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 will be changed. After completing any desired lock, read, or program operations, resume the erase operation with the Erase Resume command.

If a block is locked or locked-down during an erase suspend of the same block, the locking status bits will be changed immediately, but when the erase is resumed, the erase operation will complete. Locking operations cannot be performed during a program suspend. Refer to Appendix , Command Interface State Table, for detailed information on which commands are valid during erase suspend.

Table 13. 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)

Note: 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.

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 14. In the M58WR128E the maximum number of Program/ Erase cycles depends on the voltage supply used.

Table 14. Program, Erase Times and Program, Erase Endurance Cycles

	Parameter	Condition	Min	Typ	Typical after 100k W/E Cycles	Max	Unit
$V_{PP} = V_{DD}$	Parameter Block (4 KWord) Erase(2)			0.3	1	2.5	s
	Main Block (32 KWord) Erase	Preprogrammed		0.8	3	4	s
		Not Preprogrammed		1.1		4	s
	Bank (4Mbit) Erase	Preprogrammed		3			s
		Not Preprogrammed		4.5			s
	Parameter Block (4 KWord) Program(3)			40			ms
	Main Block (32 KWord) Program(3)			300			ms
	Word Program (3)			10	10	100	µs
	Program Suspend Latency			5		10	µs
	Erase Suspend Latency			5		20	µs
	Program/Erase Cycles (per Block)	Main Blocks	100,000				cycles
		Parameter Blocks	100,000				cycles
$V_{PP} = V_{PPH}$	Parameter Block (4 KWord) Erase			0.3		2.5	s
	Main Block (32 KWord) Erase			0.9		4	s
	Bank (4Mbit) Erase			3.5			s
	Bank (4Mbit) Program (Quad-Enhanced Factory Program)			t.b.a.(4)			s
	4Mbit Program	Quadruple Word		510			ms
	Word/ Double Word/ Quadruple Word Program(3)			8		100	µs
	Parameter Block (4 KWord) Program(3)	Quadruple Word		8			ms
		Word		32			ms
	Main Block (32 KWord) Program(3)	Quadruple Word		64			ms
		Word		256			ms
	Program/Erase Cycles (per Block)	Main Blocks				1000	cycles
		Parameter Blocks				2500	cycles

Note: 1. T_A = –40 to 85°C; V_{DD} = 1.65V to 2.2V; V_{DDQ} = 1.65V to 3.3V.
2. The difference between Preprogrammed and not preprogrammed is not significant (‹30ms).
3. Excludes the time needed to execute the command sequence.
4. t.b.a. = to be announced

MAXIMUM RATING

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

Table 15. Absolute Maximum Ratings

		Value		
Symbol	**Parameter**	**Min**	**Max**	**Unit**
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	3.6	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

DC AND AC PARAMETERS

This section summarizes the operating measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristics Tables that follow, are derived from tests performed under the Measurement Conditions summarized in Table 16, 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 16. Operating and AC Measurement Conditions

Parameter	M58WR128ET, M58WR128EB						Units
	70		80		100		
	Min	Max	Min	Max	Min	Max	
V_{DD} Supply Voltage	1.7	2.2	1.65	2.2	1.65	2.2	V
V_{DDQ} Supply Voltage	1.7	3.3	1.65	3.3	1.65	3.3	V
V_{PP} Supply Voltage (Factory environment)	11.4	12.6	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	-0.4	V_{DDQ} +0.4	V
Ambient Operating Temperature	– 40	85	– 40	85	– 40	85	°C
Load Capacitance (C_L)	30		30		30		pF
Input Rise and Fall Times		5		5		5	ns
Input Pulse Voltages	0 to V_{DDQ}		0 to V_{DDQ}		0 to V_{DDQ}		V
Input and Output Timing Ref. Voltages	$V_{DDQ}/2$		$V_{DDQ}/2$		$V_{DDQ}/2$		V

Figure 8. AC Measurement I/O Waveform

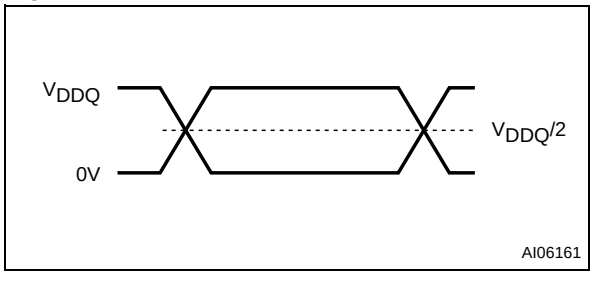


Figure 9. AC Measurement Load Circuit

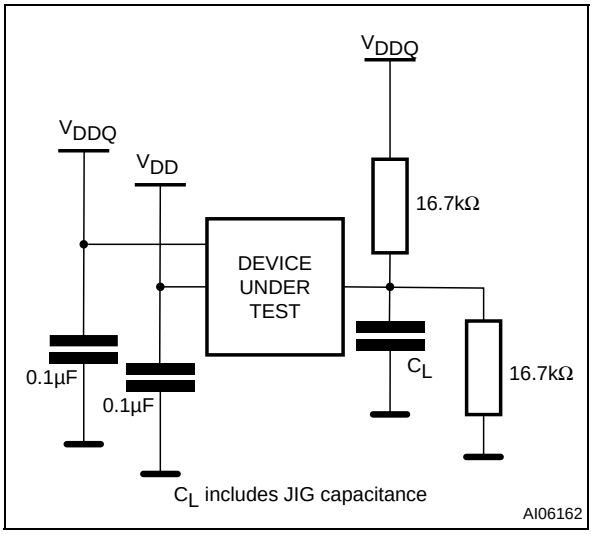


Table 17. Capacitance

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

Note: Sampled only, not 100% tested.

Table 18. DC Characteristics - Currents

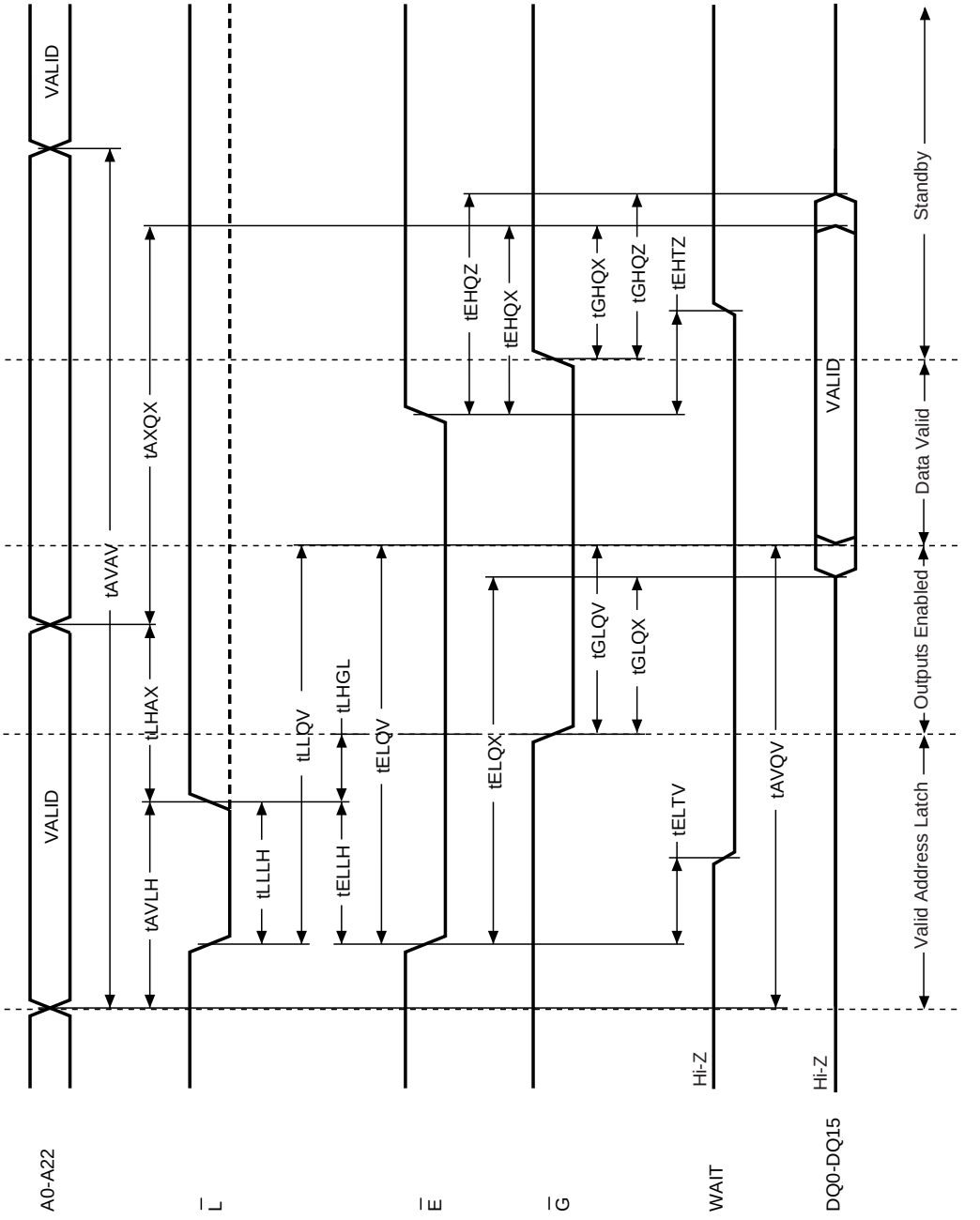
Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
I_{LI}	Input Leakage Current	$0V \le V_{IN} \le V_{DDQ}$			±1	µA
I_{LO}	Output Leakage Current	$0V \le V_{OUT} \le V_{DDQ}$			±1	µA
I_{DD1}	Supply Current Asynchronous Read (f=6MHz)	$\overline{E} = V_{IL}$, $\overline{G} = V_{IH}$		3	6	mA
	Supply Current Synchronous Read (f=40MHz)	4 Word		6	13	mA
		8 Word		8	14	mA
		Continuous		6	10	mA
	Supply Current Synchronous Read (f=54MHz)	4 Word		7	16	mA
		8 Word		10	18	mA
		Continuous		13	25	mA
I_{DD2}	Supply Current (Reset)	$\overline{RP} = V_{SS} \pm 0.2V$		10	50	µA
I_{DD3}	Supply Current (Standby)	$\overline{E} = V_{DD} \pm 0.2V$		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	15	mA
		$V_{PP} = V_{DD}$		10	20	mA
	Supply Current (Erase)	$V_{PP} = V_{PPH}$		8	15	mA
		$V_{PP} = V_{DD}$		10	20	mA
I_{DD6} (1,2)	Supply Current (Dual Operations)	Program/Erase in one Bank, Asynchronous Read in another Bank		13	26	mA
		Program/Erase in one Bank, Synchronous Read in another Bank		16	30	mA
I_{DD7}(1)	Supply Current Program/ Erase Suspended (Standby)	$\overline{E} = V_{DD} \pm 0.2V$		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

Note: 1. Sampled only, not 100% tested.
2. V_{DD} Dual Operation current is the sum of read and program or erase currents.

Table 19. 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} = 100µA			0.1	V
V_{OH}	Output High Voltage	I_{OH} = –100µA	V_{DDQ} –0.1			V
V_{PP1}	V_{PP} Program Voltage-Logic	Program, Erase	1	1.8	1.95	V
V_{PPH}	V_{PP} Program Voltage Factory	Program, Erase	11.4	12	12.6	V
V_{PPLK}	Program or Erase Lockout				0.9	V
V_{LKO}	V_{DD} Lock Voltage		1			V
V_{RPH}	$\overline{RP}$ pin Extended High Voltage				3.3	V

Figure 10. Asynchronous Random Access Read AC Waveforms



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

Figure 11. Asynchronous Page Read AC Waveforms

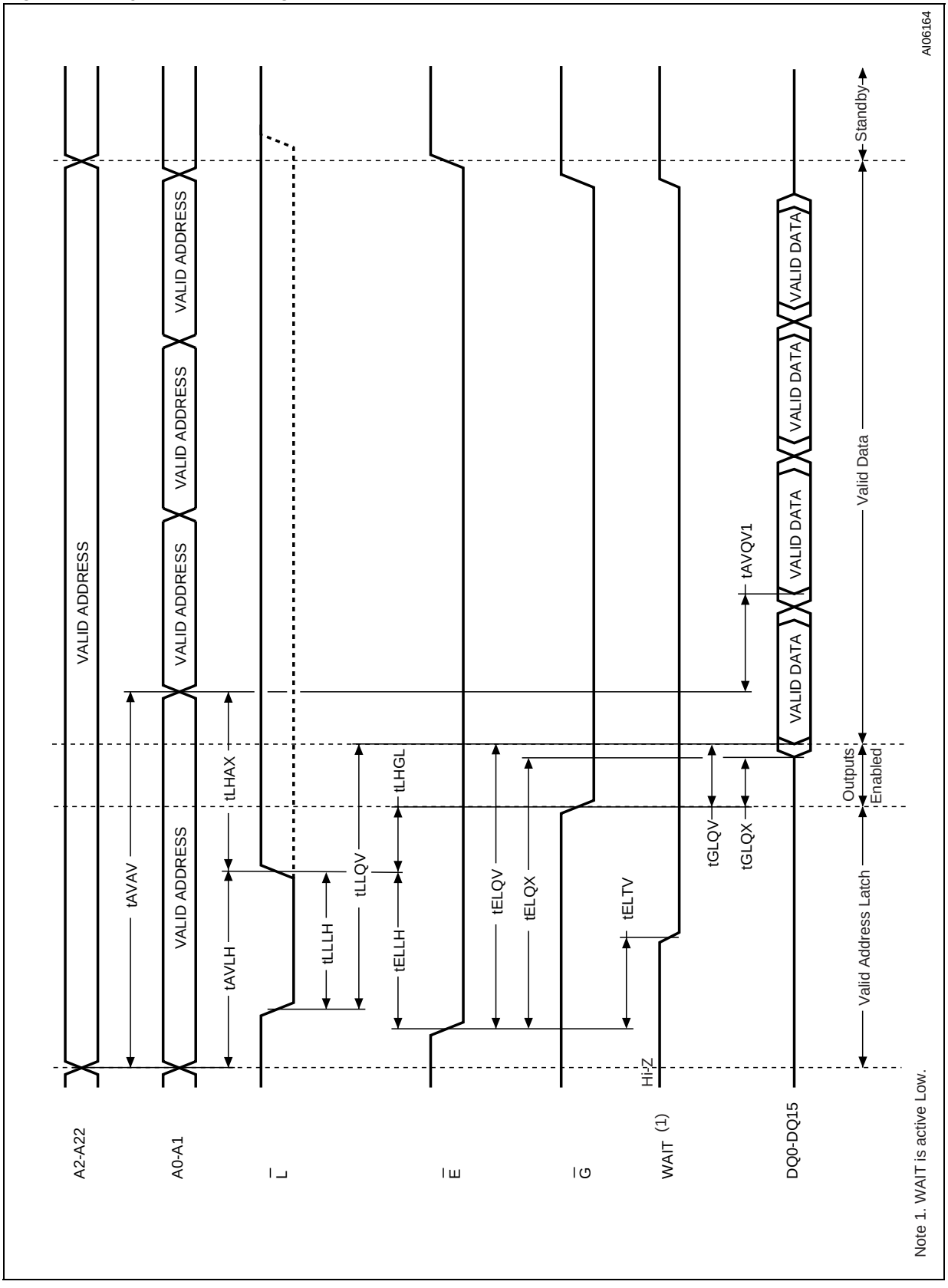
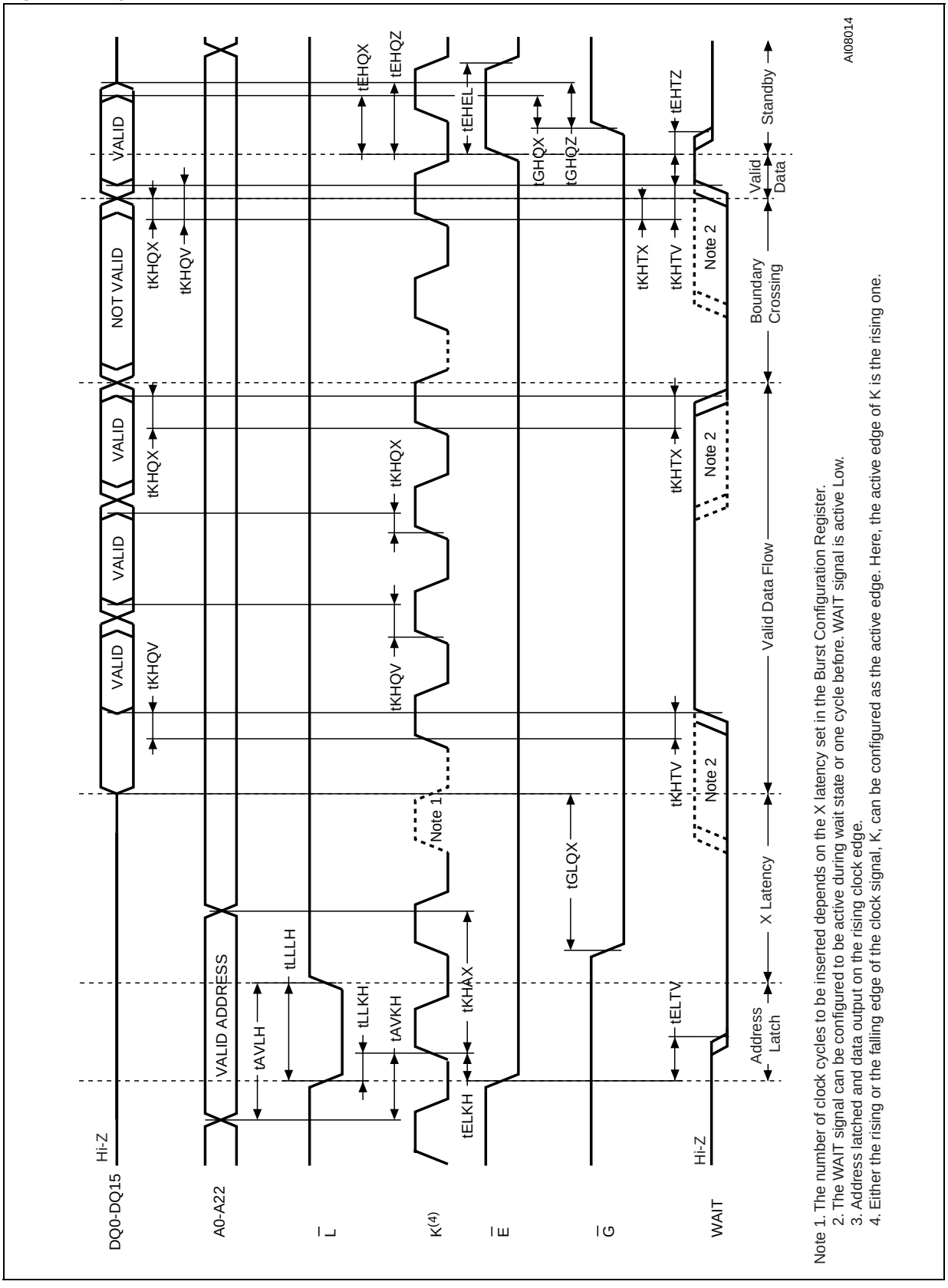


Table 20. Asynchronous Read AC Characteristics

Symbol		Alt	Parameter		V_{DDQ} = 1.65V-2.2V			V_{DDQ} = 2.2V-3.3V			Unit
					70	80	100	70	80	100	
Read Timings	t_{AVAV}	t_{RC}	Address Valid to Next Address Valid	Min	70	80	100	70	80	100	ns
	t_{AVQV}	t_{ACC}	Address Valid to Output Valid (Random)	Max	70	80	100	70	80	100	ns
	t_{AVQV1}	t_{PAGE}	Address Valid to Output Valid (Page)	Max	20	25	25	25	25	25	ns
	t_{AXQX} (1)	t_{OH}	Address Transition to Output Transition	Min	0	0	0	0	0	0	ns
	t_{ELTV}		Chip Enable Low to Wait Valid	Max	14	14	18	20	22	22	ns
	t_{ELQV} (2)	t_{CE}	Chip Enable Low to Output Valid	Max	70	80	100	70	80	100	ns
	t_{ELQX} (1)	t_{LZ}	Chip Enable Low to Output Transition	Min	0	0	0	0	0	0	ns
	t_{EHTZ}		Chip Enable High to Wait Hi-Z	Max	17	17	20	25	25	25	ns
	t_{EHQX} (1)	t_{OH}	Chip Enable High to Output Transition	Min	0	0	0	0	0	0	ns
	t_{EHQZ} (1)	t_{HZ}	Chip Enable High to Output Hi-Z	Max	17	17	20	20	20	20	ns
	t_{GLQV} (2)	t_{OE}	Output Enable Low to Output Valid	Max	20	25	25	30	30	30	ns
	t_{GLQX} (1)	t_{OLZ}	Output Enable Low to Output Transition	Min	0	0	0	0	0	0	ns
	t_{GHQX}(1)	t_{OH}	Output Enable High to Output Transition	Min	0	0	0	0	0	0	ns
	t_{GHQZ}(1)	t_{DF}	Output Enable High to Output Hi-Z	Max	17	17	20	17	17	20	ns
Latch Timings	t_{AVLH}	t_{AVADVH}	Address Valid to Latch Enable High	Min	9	9	10	10	10	12	ns
	t_{ELLH}	t_{ELADVH}	Chip Enable Low to Latch Enable High	Min	10	10	10	10	10	12	ns
	t_{LHAX}	t_{ADVHAX}	Latch Enable High to Address Transition	Min	9	9	10	9	9	10	ns
	t_{LLLH}	$t_{ADVLADVH}$	Latch Enable Pulse Width	Min	9	9	10	10	10	12	ns
	t_{LLQV}	t_{ADVLQV}	Latch Enable Low to Output Valid (Random)	Max	70	80	100	70	80	100	ns
	t_{LHGL}	t_{ADVHGL}	Latch Enable High to Output Enable Low	Min	0	0	0	0	0	0	ns

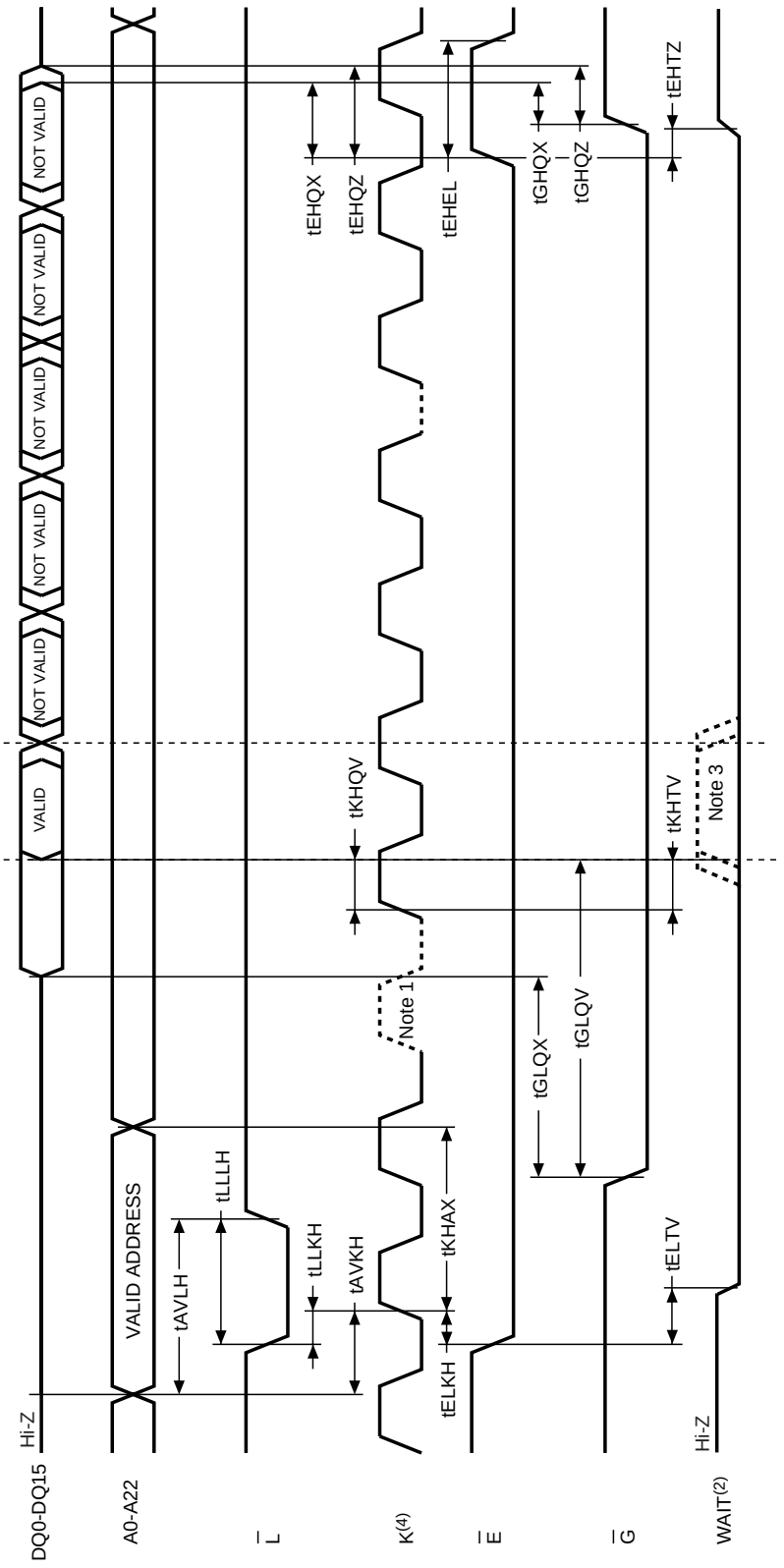
Note: 1. Sampled only, not 100% tested.
2. $\overline{G}$ may be delayed by up to t_{ELQV} - t_{GLQV} after the falling edge of $\overline{E}$ without increasing t_{ELQV}.

Figure 12. Synchronous Burst Read AC Waveforms



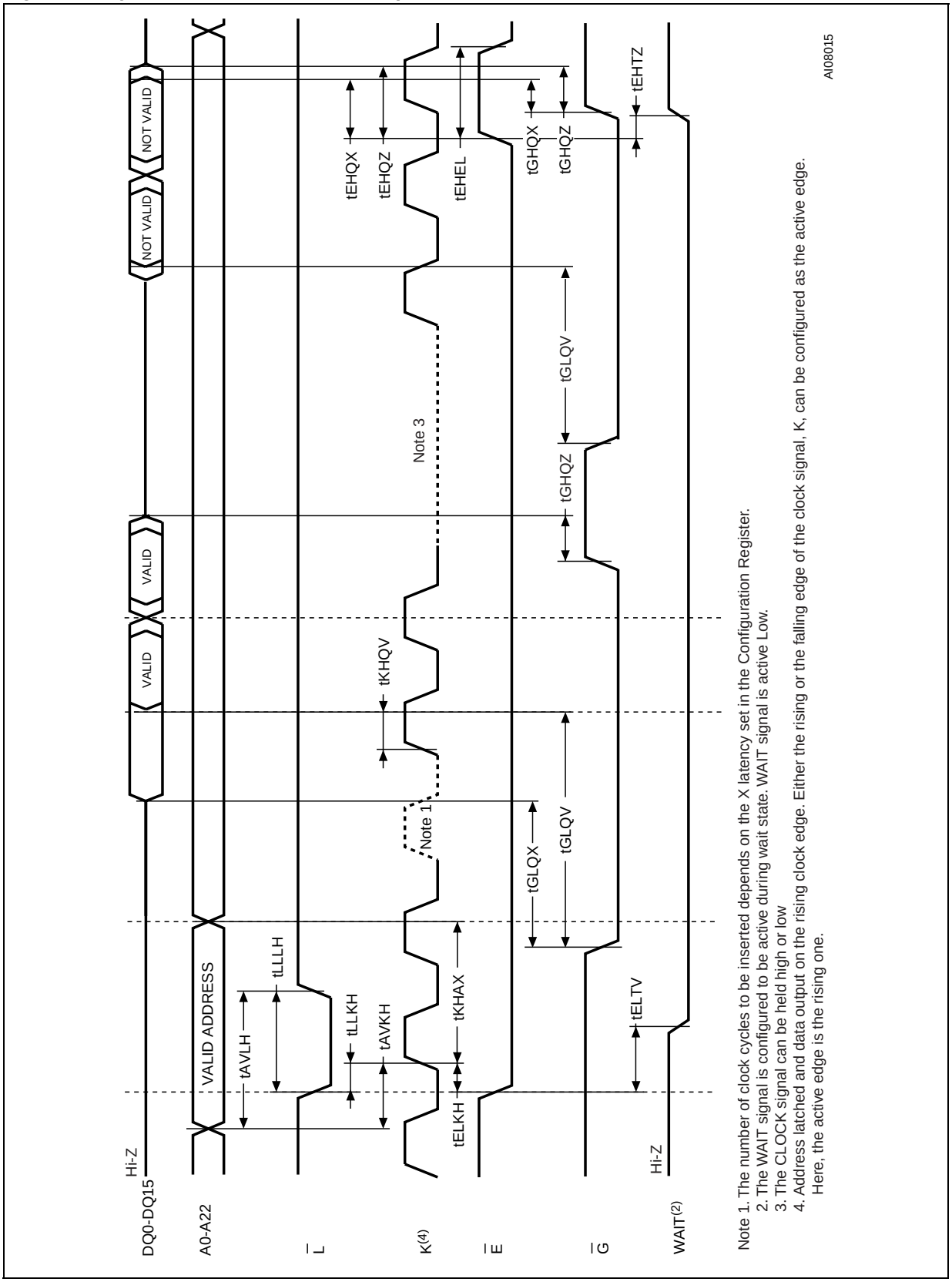
Note 1. The number of clock cycles to be inserted depends on the X latency set in the Burst Configuration Register.
2. The WAIT signal can be configured to be active during wait state or one cycle before. WAIT signal is active Low.
3. Address latched and data output on the rising clock edge.
4. Either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

Figure 13. 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 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. 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 15. Clock input AC Waveform

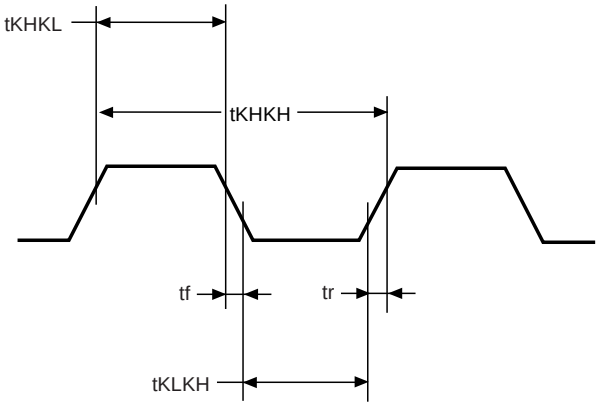


Table 21. Synchronous Read AC Characteristics

Symbol		Alt	Parameter		V_{DDQ} = 1.65V-2.2V			V_{DDQ} = 2.2V-3.3V			Unit
					70	80	100	70	80	100	
Synchronous Read Timings	t_{AVKH}	t_{AVCLKH}	Address Valid to Clock High	Min	9	9	9	9	9	10	ns
	t_{ELKH}	t_{ELCLKH}	Chip Enable Low to Clock High	Min	9	9	9	9	9	10	ns
	t_{ELTV}		Chip Enable Low to Wait Valid	Max	14	14	18	20	22	22	ns
	t_{EHEL}		Chip Enable Pulse Width (subsequent synchronous reads)	Min	14	14	14	20	20	20	ns
	t_{EHTZ}		Chip Enable High to Wait Hi-Z	Max	14	14	20	25	25	25	ns
	t_{KHAX}	t_{CLKHAX}	Clock High to Address Transition	Min	9	9	10	10	10	10	ns
	t_{KHQV} t_{KHTV}	t_{CLKHQV}	Clock High to Output Valid Clock High to WAIT Valid	Max	14	14	18	20	22	22	ns
	t_{KHQX} t_{KHTX}	t_{CLKHQX}	Clock High to Output Transition Clock High to WAIT Transition	Min	4	4	4	5	5	5	ns
	t_{LLKH}	$t_{ADVLCLKH}$	Latch Enable Low to Clock High	Min	9	9	9	10	10	10	ns
Clock Specifications	t_{KHKH}	t_{CLK}	Clock Period (f=33MHz)	Min	-	-	-	-	30	30	ns
			Clock Period (f=40MHz)		-	-	25	25	-	-	ns
			Clock Period (f=54MHz)		18.5	18.5	-	-	-	-	ns
	t_{KHKL} t_{KLKH}		Clock High to Clock Low Clock Low to Clock High	Min	4.5	4.5	5	9.5	9.5	9.5	ns
	t_f t_r		Clock Fall or Rise Time	Max	3	3	3	3	5	5	ns

Note: 1. Sampled only, not 100% tested.
2. For other timings please refer to Table 20, Asynchronous Read AC Characteristics.

Figure 16. Write AC Waveforms, Write Enable Controlled

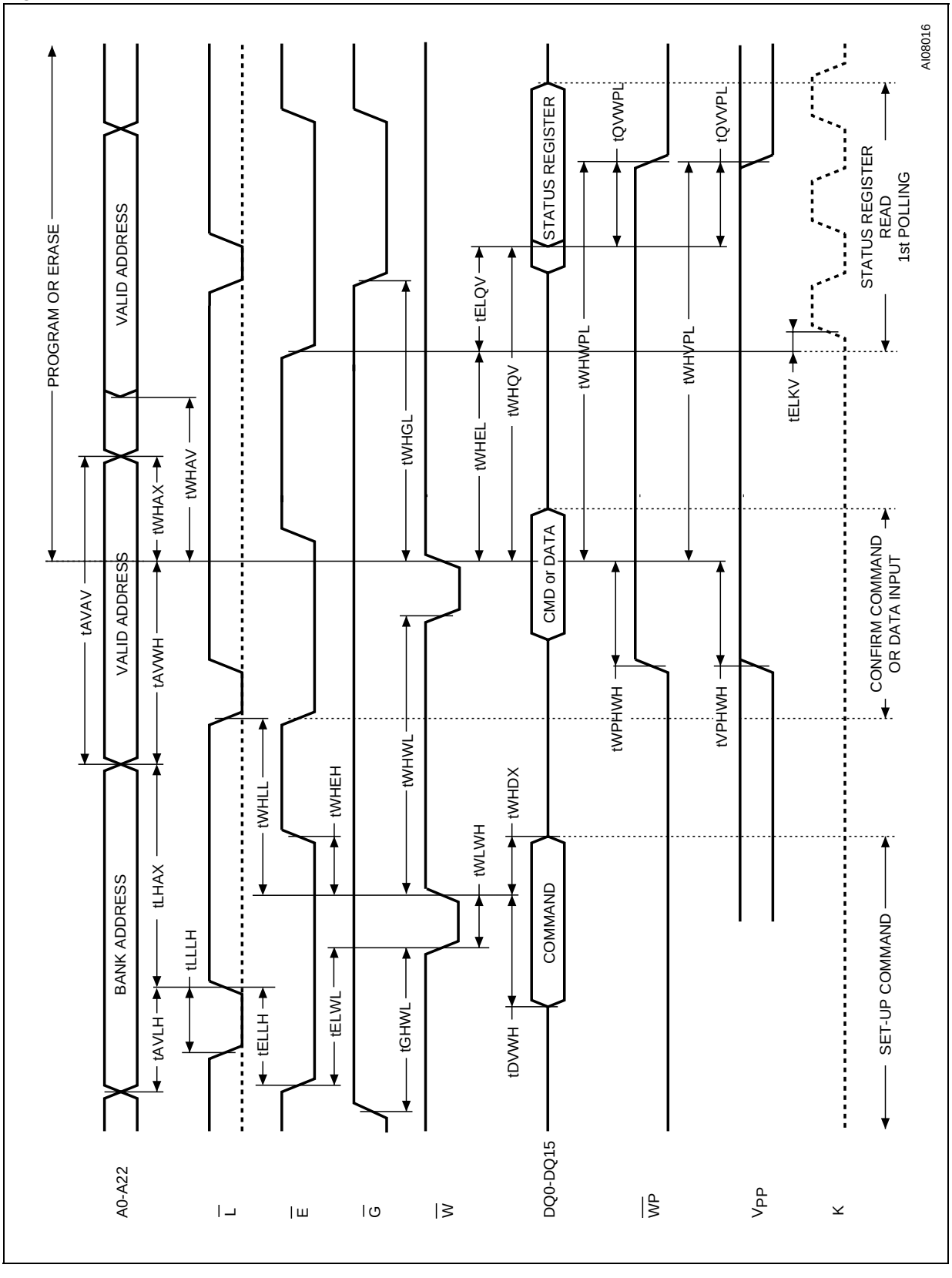


Table 22. Write AC Characteristics, Write Enable Controlled

Symbol		Alt	Parameter		M58WR128E			Unit
					70	80	100	
Write Enable Controlled Timings	t_{AVAV}	t_{WC}	Address Valid to Next Address Valid	Min	70	80	100	ns
	t_{AVLH}		Address Valid to Latch Enable High	Min	9	9	10	ns
	t_{AVWH}(3)	t_{WC}	Address Valid to Write Enable High	Min	45	50	50	ns
	t_{DVWH}	t_{DS}	Data Valid to Write Enable High	Min	45	50	50	ns
	t_{ELLH}		Chip Enable Low to Latch Enable High	Min	10	10	10	ns
	t_{ELWL}	t_{CS}	Chip Enable Low to Write Enable Low	Min	0	0	0	ns
	t_{ELQV}		Chip Enable Low to Output Valid	Min	70	80	100	ns
	t_{ELKV}		Chip Enable High to Clock Valid	Min	9	9	9	ns
	t_{GHWL}		Output Enable High to Write Enable Low	Min	17	17	20	ns
	t_{LHAX}		Latch Enable High to Address Transition	Min	9	9	10	ns
	t_{LLLH}		Latch Enable Pulse Width	Min	9	9	10	ns
	t_{WHAV}(3)		Write Enable High to Address Valid	Min	0	0	0	ns
	t_{WHAX}(3)	t_{AH}	Write Enable High to Address Transition	Min	0	0	0	ns
	t_{WHDX}	t_{DH}	Write Enable High to Input Transition	Min	0	0	0	ns
	t_{WHEH}	t_{CH}	Write Enable High to Chip Enable High	Min	0	0	0	ns
	t_{WHEL}(2)		Write Enable High to Chip Enable Low	Min	25	25	25	ns
	t_{WHGL}		Write Enable High to Output Enable Low	Min	0	0	0	ns
	t_{WHLL}		Write Enable High to Latch Enable Low	Min	0	0	0	ns
	t_{WHWL}	t_{WPH}	Write Enable High to Write Enable Low	Min	25	25	25	ns
	t_{WHQV}		Write Enable High to Output Valid	Min	95	105	125	ns
	t_{WLWH}	t_{WP}	Write Enable Low to Write Enable High	Min	45	50	50	ns
Protection Timings	t_{QVVPL}		Output (Status Register) Valid to V_{PP} Low	Min	0	0	0	ns
	t_{QVWPL}		Output (Status Register) Valid to Write Protect Low	Min	0	0	0	ns
	t_{VPHWH}	t_{VPS}	V_{PP} High to Write Enable High	Min	200	200	200	ns
	t_{WHVPL}		Write Enable High to V_{PP} Low	Min	200	200	200	ns
	t_{WHWPL}		Write Enable High to Write Protect Low	Min	200	200	200	ns
	t_{WPHWH}		Write Protect High to Write Enable High	Min	200	200	200	ns

Note: 1. Sampled only, not 100% tested.
2. t_{WHEL} has the values shown when reading in the targeted bank. 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 a command. If it is a Read Array operation in a different bank t_{WHEL} is 0ns.
3. Meaningful only if $\overline{L}$ is always kept low.

Figure 17. Write AC Waveforms, Chip Enable Controlled

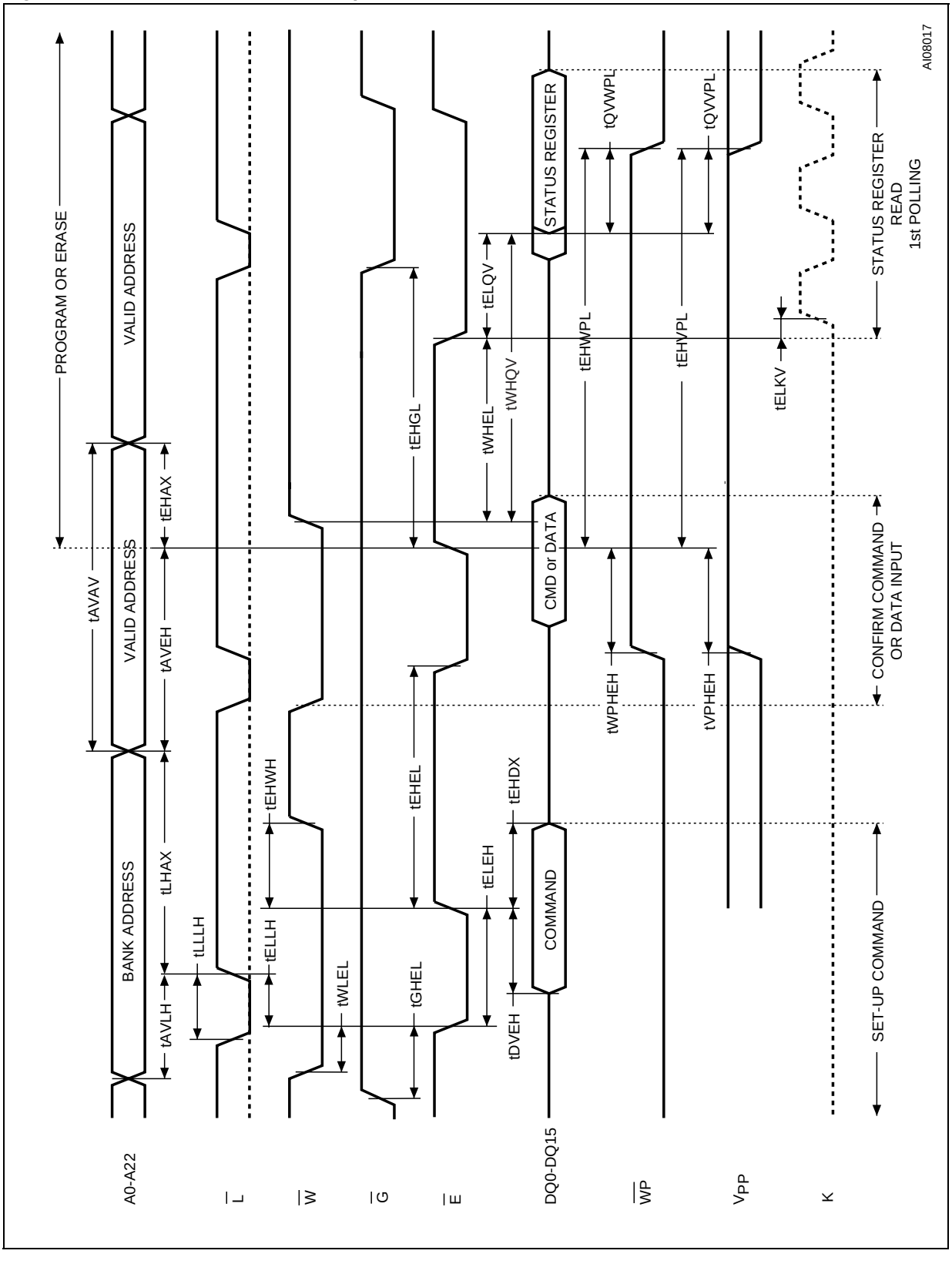


Table 23. Write AC Characteristics, Chip Enable Controlled

Symbol		Alt	Parameter		M58WR128E			Unit
					70	80	100	
Chip Enable Controlled Timings	t_{AVAV}	t_{WC}	Address Valid to Next Address Valid	Min	70	80	100	ns
	t_{AVEH}	t_{WC}	Address Valid to Chip Enable High	Min	45	50	50	ns
	t_{AVLH}		Address Valid to Latch Enable High	Min	9	9	10	ns
	t_{DVEH}	t_{DS}	Data Valid to Write Enable High	Min	45	50	50	ns
	t_{EHAX}	t_{AH}	Chip Enable High to Address Transition	Min	0	0	0	ns
	t_{EHDX}	t_{DH}	Chip Enable High to Input Transition	Min	0	0	0	ns
	t_{EHEL}	t_{WPH}	Chip Enable High to Chip Enable Low	Min	25	25	25	ns
	t_{EHGL}		Chip Enable High to Output Enable Low	Min	0	0	0	ns
	t_{EHWH}	t_{CH}	Chip Enable High to Write Enable High	Min	0	0	0	ns
	t_{ELKV}		Chip Enable Low to Clock Valid	Min	9	9	9	ns
	t_{ELEH}	t_{WP}	Chip Enable Low to Chip Enable High	Min	45	50	50	ns
	t_{ELLH}		Chip Enable Low to Latch Enable High	Min	10	10	10	ns
	t_{ELQV}		Chip Enable Low to Output Valid	Min	70	80	100	ns
	t_{GHEL}		Output Enable High to Chip Enable Low	Min	17	17	20	ns
	t_{LHAX}		Latch Enable High to Address Transition	Min	9	9	10	ns
	t_{LLLH}		Latch Enable Pulse Width	Min	9	9	10	ns
	t_{WHEL} (2)		Write Enable High to Chip Enable Low	Min	25	25	25	ns
	t_{WHQV}		Write Enable High to Output Valid	Min	95	105	125	ns
	t_{WLEL}	t_{CS}	Write Enable Low to Chip Enable Low	Min	0	0	0	ns
Protection Timings	t_{EHVPL}		Chip Enable High to V_{PP} Low	Min	200	200	200	ns
	t_{EHWPL}		Chip Enable High to Write Protect Low	Min	200	200	200	ns
	t_{QVVPL}		Output (Status Register) Valid to V_{PP} Low	Min	0	0	0	ns
	t_{QVWPL}		Output (Status Register) Valid to Write Protect Low	Min	0	0	0	ns
	t_{VPHEH}	t_{VPS}	V_{PP} High to Chip Enable High	Min	200	200	200	ns
	t_{WPHEH}		Write Protect High to Chip Enable High	Min	200	200	200	ns

Note: 1. Sampled only, not 100% tested.
2. t_{WHEL} has the values shown when reading in the targeted bank. 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 a command. If it is a Read Array operation in a different bank t_{WHEL} is 0ns.

Figure 18. Reset and Power-up AC Waveforms

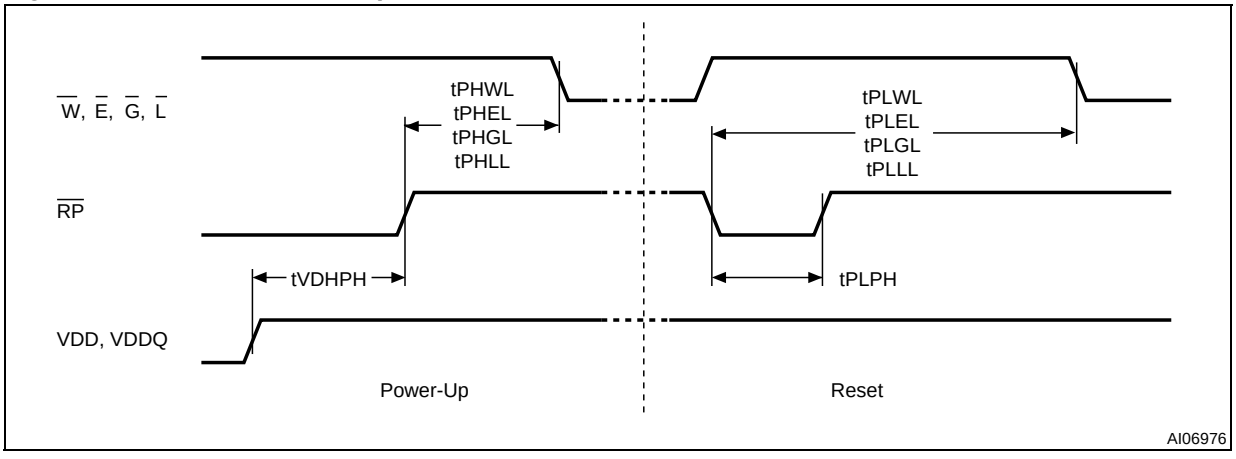


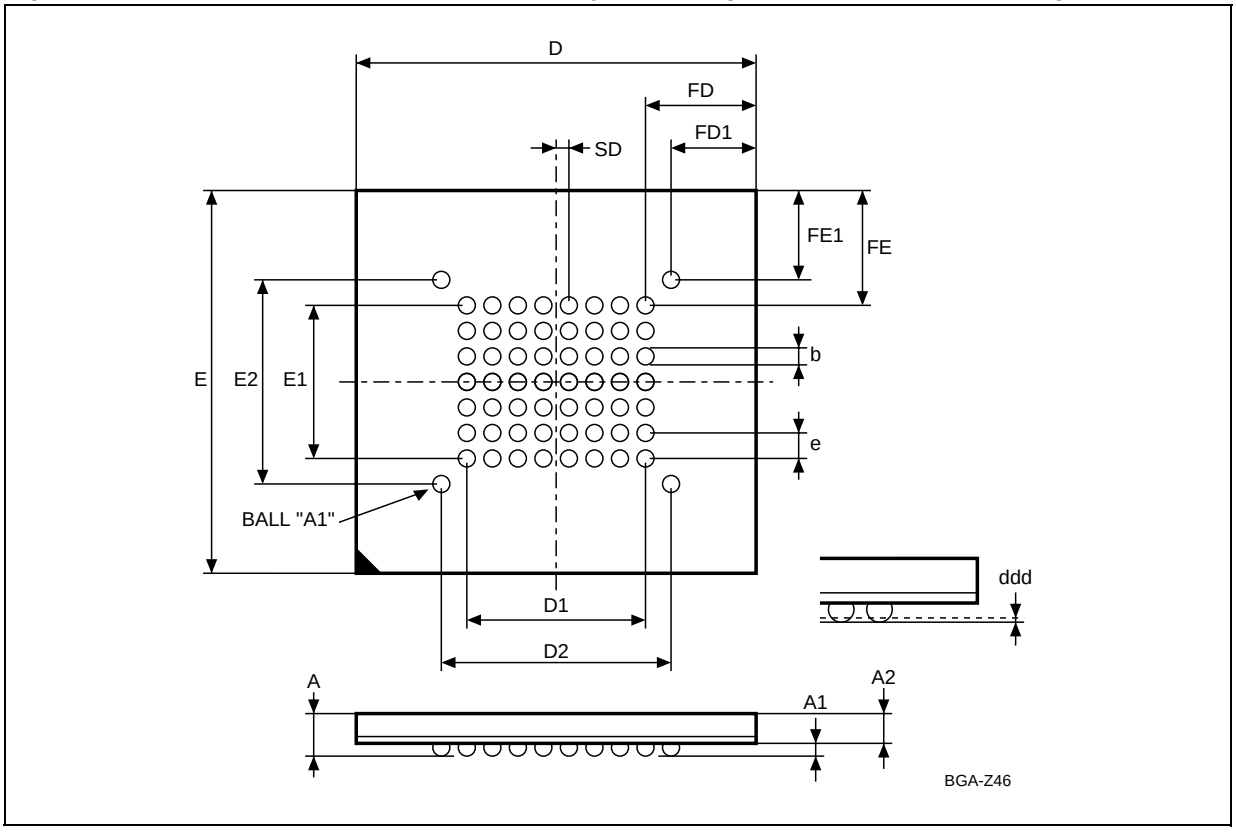
Table 24. Reset and Power-up AC Characteristics

Symbol	Parameter	Test Condition		70	80	100	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	10	µs
		During Erase	Min	20	20	20	µs
		Other Conditions	Min	80	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	30	ns
t_{PLPH} (1,2)	$\overline{RP}$ Pulse Width		Min	50	50	50	ns
t_{VDHPH} (3)	Supply Voltages High to Reset High		Min	50	50	50	µs

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

PACKAGE MECHANICAL

Figure 19. VFBGA60 12.5x12mm - 8x7 ball array, 0.75mm pitch, Bottom View Package Outline



Note: Drawing is not to scale.

Table 25. VFBGA60 12.5x12mm - 8x7 ball array, 0.75mm 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	0.370	0.320	0.420	0.0146	0.0126	0.0165
D	12.500	12.400	12.600	0.4921	0.4882	0.4961
D1	5.250			0.2067		
D2	6.750			0.2657		
ddd			0.100			0.0039
E	12.000	11.900	12.100	0.4724	0.4685	0.4764
E1	4.500			0.1772		
E2	6.000			0.2362		
e	0.750	–	–	0.0295	–	–
FD	3.625			0.1427		
FD1	2.875			0.1132		
FE	3.750			0.1476		
FE1	3.000			0.1181		
SD	0.375			0.0148		

Figure 20. VFBGA60 Daisy Chain - Package Connections (Top view through package)

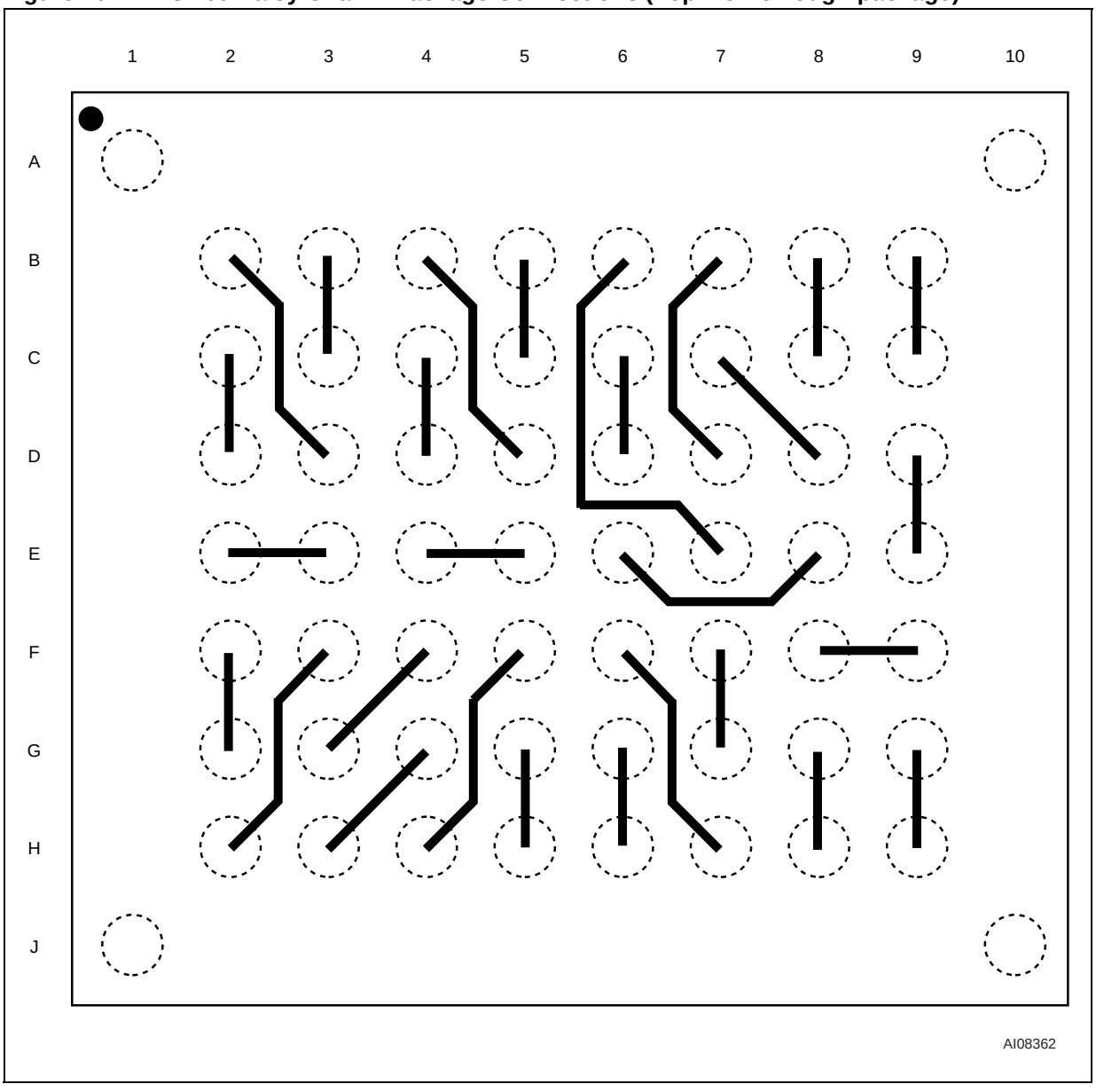
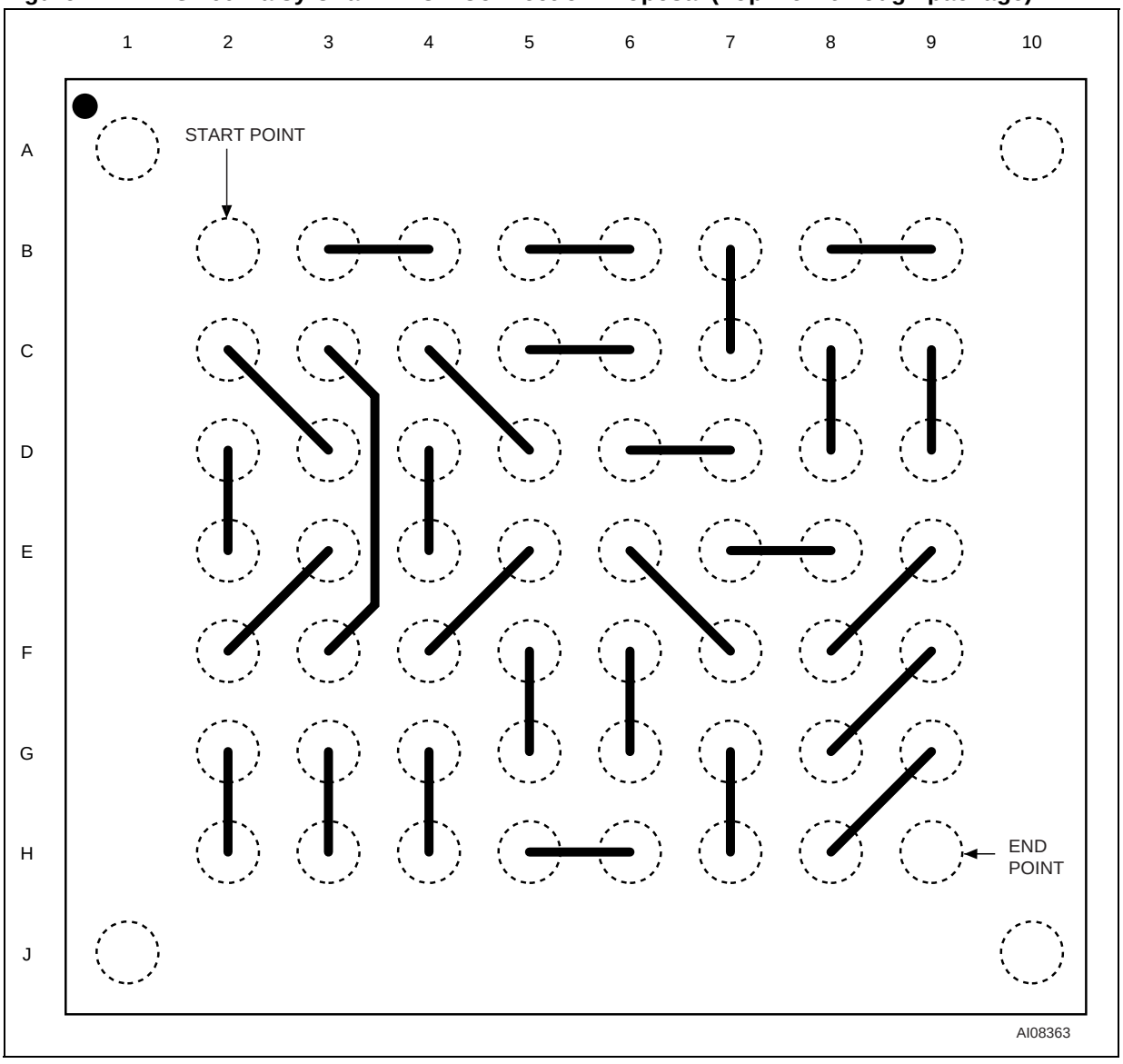


Figure 21. VFBGA60 Daisy Chain - PCB Connection Proposal (Top view through package)



PART NUMBERING

Table 26. Ordering Information Scheme

Example: M58WR128ET 80 ZB 6 T

Device Type
M58

Architecture
W = Multiple Bank, Burst Mode

Operating Voltage
R = V_{DD} = 1.65V to 2.2V, V_{DDQ} = 1.65V to 3.3V

Device Function
128ET = 128 Mbit (x16), Top Boot
128EB = 128 Mbit (x16), Bottom Boot

Speed
70 = 70 ns
80 = 80 ns
10 = 100 ns

Package
ZB = VFBGA60 12.5 x 12mm, 0.75mm pitch

Temperature Range
6 = –40 to 85°C

Option
T = Tape & Reel packing

Table 27. Daisy Chain Ordering Scheme

Example: M58WR128E -ZB T

Device Type
M58WR128E

Daisy Chain
ZB = VFBGA60 12.5 x 12mm, 0.75mm pitch

Option
T = Tape & Reel Packing

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

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

APPENDIX A. BLOCK ADDRESS TABLES

The following set of equations can be used to calculate a complete set of block addresses using the information contained in Tables 28, 29, 30, 31, 32 and 33.

To calculate the Block Base Address from the Block Number:

First it is necessary to calculate the Bank Number and the Block Number Offset. This can be achieved using the following formulas:

Bank_Number = (Block_Number – 7) / 8

Block_Number_Offset = Block_Number – 7 – (Bank_Number x 8)

If the Bank_Number = 0, the Block Base Address can be directly read from Table 28 or Table 31 (Parameter Bank Block Addresses) in the Block Number Offset row. Otherwise:

Block_Base_Address = Bank_Base_Address + Block_Address_Offset

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

If the address is in the range of the Parameter Bank, the Bank Number is 0 and the Block Number can be directly read from Table 28 or Table 31(Parameter Bank Block Addresses), in the row that corresponds to the address given. Otherwise, the Block Number can be calculated using the formulas below:

For the top configuration (M58WR128ET):

Block_Number = ((NOT address) / 2^{15}) + 7

For the bottom configuration (M58WR128EB):

Block_Number = (address / 2^{15}) + 7

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

Bank_Number = (Block_Number – 7) / 8

Block_Number_Offset = Block_Number – 7 – (Bank_Number x 8)

Table 28. M58WR128ET - Parameter Bank Block Addresses

Block Number	Size	Address Range
0	4	7FF000-7FFFFF
1	4	7FE000 - 7FEFFF
2	4	7FD000 - 7FDFFF
3	4	7FC000 - 7FCFFF
4	4	7FB000 - 7FBFFF
5	4	7FA000 - 7FAFFF
6	4	7F9000 - 7F9FFF
7	4	7F8000 - 7F8FFF
8	32	7F0000 - 7F7FFF
9	32	7E8000 - 7EFFFF
10	32	7E0000 - 7E7FFF
11	32	7D8000 - 7DFFFF
12	32	7D0000 - 7D7FFF
13	32	7C8000 - 7CFFFF
14	32	7C0000 - 7C7FFF

Table 29. M58WR128ET -Main Bank Base Addresses

Bank Number	First Block Number	Bank Base Address
1	22	780000
2	30	740000
3	38	700000
4	46	6C0000
5	54	680000
6	62	640000
7	70	600000
8	78	5C0000
9	86	580000
10	94	540000
11	104	500000
12	110	4C0000
13	118	480000
14	126	440000
15	134	400000
16	142	3C0000
17	150	380000
18	158	340000
19	166	300000
20	174	2C0000
21	182	280000
22	190	240000
23	198	200000
24	206	1C00000
25	214	180000
26	222	140000
27	230	100000
28	238	0C0000
29	246	080000
30	254	040000
31	262	000000

Table 30. M58WR128ET - Block Addresses in Main Banks

Block Number Offset	Block Base Address Offset
0	–038000
1	–030000
2	–028000
3	–020000
4	–018000
5	–010000
6	–008000
7	000000

Table 31. M58WR128EB - Parameter Bank Block Addresses

Block Number	Size	Address Range
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

Table 32. M58WR128EB - Main Bank Base Addresses

Bank Number	Last Block Number	Bank Base Address
31	255	7C0000
30	247	780000
29	239	740000
28	231	700000
27	223	6C0000
26	215	680000
25	207	640000
24	199	600000
23	191	5C0000
22	183	580000
21	175	540000
20	167	500000
19	159	4C0000
18	151	480000
17	143	440000
16	135	400000
15	127	3C0000
14	119	380000
13	111	340000
12	103	300000
11	95	2C0000
10	87	280000
9	79	240000
8	71	200000
7	63	1C0000
6	55	180000
5	47	140000
4	39	100000
3	31	0C0000
2	23	080000
1	15	040000

Table 33. M58WR128EB - Block Addresses in Main Banks

Block Number Offset	Block Base Address Offset
7	038000
6	030000
5	028000
4	020000
3	018000
2	010000
1	008000
0	000000

APPENDIX B. COMMON FLASH INTERFACE

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

When the Read CFI Query Command is issued the device enters CFI Query mode and the data structure is read from the memory. Tables 34, 35, 36, 37, 38, 39, 40, 41, 42 and 43 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 5, Security Block and 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 34. Query Structure Overview

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 & 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 Unique device Number and User Programmable OTP

Note: 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 35, 36, 37 and 38. Query data is always presented on the lowest order data outputs.

Table 35. CFI Query Identification String

Offset	Sub-section Name	Description	Value
00h	0020h	Manufacturer Code	ST
01h	881Eh 881Fh	Device Code	Top Bottom
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 37)	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 36. CFI Query System Interface Information

Offset	Data	Description	Value
1Bh	0017h	V_{DD} Logic Supply Minimum Program/Erase or Write voltage bit 7 to 4 BCD value in volts bit 3 to 0 BCD value in 100 millivolts	1.7V
1Ch	0022h	V_{DD} Logic Supply Maximum Program/Erase or Write voltage bit 7 to 4 BCD value in volts bit 3 to 0 BCD value in 100 millivolts	2.2V
1Dh	0017h	V_{PP} [Programming] Supply Minimum Program/Erase voltage bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 millivolts	1.7V
1Eh	00C0h	V_{PP} [Programming] Supply Maximum Program/Erase voltage bit 7 to 4 HEX value in volts bit 3 to 0 BCD value in 100 millivolts	12V
1Fh	0004h	Typical time-out per single byte/word program = 2^n µs	16µs
20h	0003h	Typical time-out for quadruple word program = 2^n µs	8µs
21h	000Ah	Typical time-out per individual block erase = 2^n ms	1s
22h	0000h	Typical time-out for full chip erase = 2^n ms	NA
23h	0003h	Maximum time-out for word program = 2^n times typical	128µs
24h	0004h	Maximum time-out for quadruple word = 2^n times typical	128µs
25h	0002h	Maximum time-out per individual block erase = 2^n times typical	4s
26h	0000h	Maximum time-out for chip erase = 2^n times typical	NA

Table 37. Device Geometry Definition

Offset Word Mode		Data	Description	Value
27h		0018h	Device Size = 2^n in number of Bytes	16 MBytes
28h 29h		0001h 0000h	Flash Device Interface Code description	x16 Async.
2Ah 2Bh		0003h 0000h	Maximum number of bytes in multi-byte program or page = 2^n	8 Byte
2Ch		0002h	Number of identical sized erase block regions within the device bit 7 to 0 = x = number of Erase Block Regions	2
M58WR128ET	2Dh 2Eh	00FEh 0000h	Region 1 Information Number of identical-size erase blocks = 00FEh+1	255
	2Fh 30h	0000h 0001h	Region 1 Information Block size in Region 1 = 0100h * 256 byte	64 KByte
	31h 32h	0007h 0000h	Region 2 Information Number of identical-size erase blocks = 0007h+1	8
	33h 34h	0020h 0000h	Region 2 Information Block size in Region 2 = 0020h * 256 byte	8 KByte
	35h 38h	Reserved	Reserved for future erase block region information	NA
M58WR128EB	2Dh 2Eh	0007h 0000h	Region 1 Information Number of identical-size erase block = 0007h+1	8
	2Fh 30h	0020h 0000h	Region 1 Information Block size in Region 1 = 0020h * 256 byte	8 KByte
	31h 32h	00FEh 0000h	Region 2 Information Number of identical-size erase block = 00FEh+1	255
	33h 34h	0000h 0001h	Region 2 Information Block size in Region 2 = 0100h * 256 byte	64 KByte
	35h 38h	Reserved	Reserved for future erase block region information	NA

Table 38. Primary Algorithm-Specific Extended Query Table

Offset	Data	Description	Value
(P)h = 39h	0050h		"P"
	0052h	Primary Algorithm extended Query table unique ASCII string “PRI”	"R"
	0049h		"I"
(P+3)h = 3Ch	0031h	Major version number, ASCII	"1"
(P+4)h = 3Dh	0030h	Minor version number, ASCII	"0"
(P+5)h = 3Eh	00E6h	Extended Query table contents for Primary Algorithm. Address (P+5)h contains less significant byte. bit 0 Chip Erase supported (1 = Yes, 0 = No) bit 1 Erase Suspend supported (1 = Yes, 0 = No) bit 2 Program Suspend supported (1 = Yes, 0 = No) bit 3 Legacy Lock/Unlock supported (1 = Yes, 0 = No) bit 4 Queued Erase supported (1 = Yes, 0 = No) bit 5 Instant individual block locking supported (1 = Yes, 0 = No) bit 6 Protection bits supported (1 = Yes, 0 = No) bit 7 Page mode read supported (1 = Yes, 0 = No) bit 8 Synchronous read supported (1 = Yes, 0 = No) bit 9 Simultaneous operation supported (1 = Yes, 0 = No) bit 10 to 31 Reserved; undefined bits are ‘0’. If bit 31 is ’1’ then another 31 bit field of optional features follows at the end of the bit-30 field.	
	0003h		
(P+7)h = 40h	0000h		No
(P+8)h = 41h	0000h		Yes Yes No No Yes Yes Yes Yes Yes
(P+9)h = 42h	0001h	Supported Functions after Suspend Read Array, Read Status Register and CFI Query bit 0 Program supported after Erase Suspend (1 = Yes, 0 = No) bit 7 to 1 Reserved; undefined bits are ‘0’	Yes
(P+A)h = 43h	0003h	Block Protect Status Defines which bits in the Block Status Register section of the Query are implemented. bit 0 Block protect Status Register Lock/Unlock bit active(1=Yes, 0 =No) bit 1 Block Lock Status Register Lock-Down bit active (1=Yes, 0 =No) bit 15 to 2 Reserved for future use; undefined bits are ‘0’	
(P+B)h = 44h	0000h		Yes Yes
(P+C)h = 45h	0018h	V_{DD} Logic Supply Optimum Program/Erase voltage (highest performance) bit 7 to 4HEX value in volts bit 3 to 0BCD value in 100 mV	1.8V
(P+D)h = 46h	00C0h	V_{PP} Supply Optimum Program/Erase voltage bit 7 to 4HEX value in volts bit 3 to 0BCD value in 100 mV	12V

Table 39. 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 Bits 0-7 Lower byte of protection register address Bits 8-15 Upper byte of protection register address	0080h
(P+10)h = 49h	0000h		
(P+11)h = 4Ah	0003h	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 40. Burst Read Information

Offset	Data	Description	Value
(P+13)h = 4Ch	0003h	Page-mode read capability 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 bit 3-7 Reserved bit 0-2 'n' such that 2^{n+1} HEX value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width. A value of 07h indicates that the device is capable of continuous linear bursts that 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 41. Bank and Erase Block Region Information

M58WR128ET (top)		M58WR128EB (bottom)		Description
Offset	Data	Offset	Data	
(P+19)h = 52h	02h	(P+19)h = 52h	02h	Number of Bank Regions within the device

Note: 1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, 1 contains all the banks that are made up of main blocks only, 2 contains the banks that are made up of the parameter and main blocks.

Table 42. Bank and Erase Block Region 1 Information

M58WR128ET (top)		M58WR128EB (bottom)		Description
Offset	Data	Offset	Data	
(P+1A)h =53h	1Fh	(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: Bits 0-3: Number of simultaneous program operations 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 Bits 0-3: Number of simultaneous program operations 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 Bits 0-3: Number of simultaneous program operations 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 n = number of erase block regions with contiguous same-size erase blocks. 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 Bits 0-15: n+1 = number of identical-sized erase blocks 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) 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 Bits 0-3: bits per cell in erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved
(P+27)h =60h	03h	(P+27)h =60h	03h	Bank Region 1 (Erase Block Type 1): Page mode and synchronous mode capabilities Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
		(P+28)h =61h	06h	Bank Region 1 Erase Block Type 2 Information Bits 0-15: n+1 = number of identical-sized erase blocks 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) 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 Bits 0-3: bits per cell in erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved

M58WR128ET (top)		M58WR128EB (bottom)		Description
Offset	Data	Offset	Data	
		(P+2F)h =68h	03h	Bank Region 1 (Erase Block Type 2): Page mode and synchronous mode capabilities Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved

Note: 1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, 1 contains all the banks that are made up of main blocks only, 2 contains the banks that are made up of the parameter and main blocks.

Table 43. Bank and Erase Block Region 2 Information

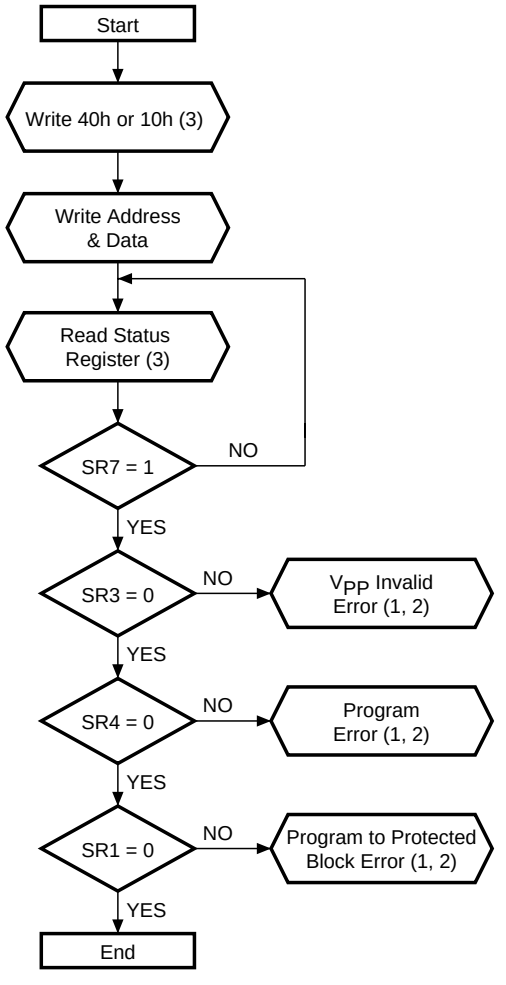
M58WR128ET (top)		M58WR128EB (bottom)		Description
Offset	Data	Offset	Data	
(P+28)h =61h	01h	(P+30)h =69h	1Fh	Number of identical banks within bank region 2
(P+29)h =62h	00h	(P+31)h =6Ah	00h	
(P+2A)h =63h	11h	(P+32)h =6Bh	11h	Number of program or erase operations allowed in bank region 2: Bits 0-3: Number of simultaneous program operations 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 Bits 0-3: Number of simultaneous program operations Bits 4-7: Number of simultaneous erase operations
(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 Bits 0-3: Number of simultaneous program operations 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 n = number of erase block regions with contiguous same-size erase blocks. 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 Bits 0-15: n+1 = number of identical-sized erase blocks 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) 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 Bits 0-3: bits per cell in erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved

M58WR128ET (top)		M58WR128EB (bottom)		Description
Offset	Data	Offset	Data	
(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 10) Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
(P+36)h =6Fh	07h			Bank Region 2 Erase Block Type 2 Information Bits 0-15: n+1 = number of identical-sized erase blocks 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) 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 Bits 0-3: bits per cell in erase region Bit 4: reserved for “internal ECC used” BIts 5-7: reserved
(P+3D)h =76h	03h			Bank Region 2 (Erase Block Type 2): Page mode and synchronous mode capabilities (defined in table 10) Bit 0: Page-mode reads permitted Bit 1: Synchronous reads permitted Bit 2: Synchronous writes permitted Bits 3-7: reserved
(P+3E)h =77h		(P+3E)h =77h		Feature Space definitions
(P+3F)h =78h		(P+3F)h =78h		Reserved

Note: 1. The variable P is a pointer which is defined at CFI offset 15h.
2. Bank Regions. There are two Bank Regions, Region 1 contains all the banks that are made up of main blocks only, Region 2 contains the banks that are made up of the parameter and main blocks.

APPENDIX C. FLOWCHARTS AND PSEUDO CODES

Figure 22. Program Flowchart and Pseudo Code

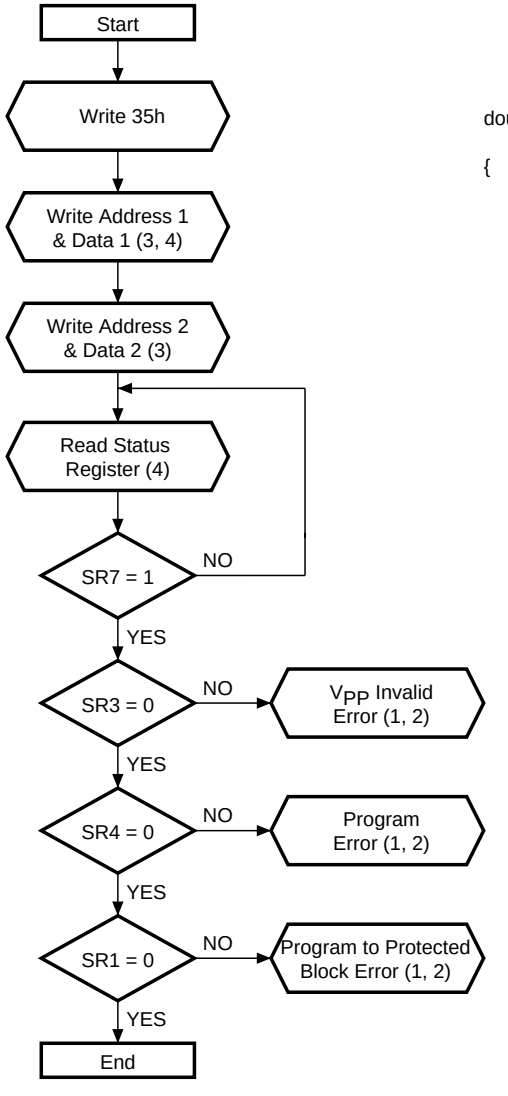



```
program_command (addressToProgram, dataToProgram) {:

        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

Note: 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 23. Double Word Program Flowchart and Pseudo code

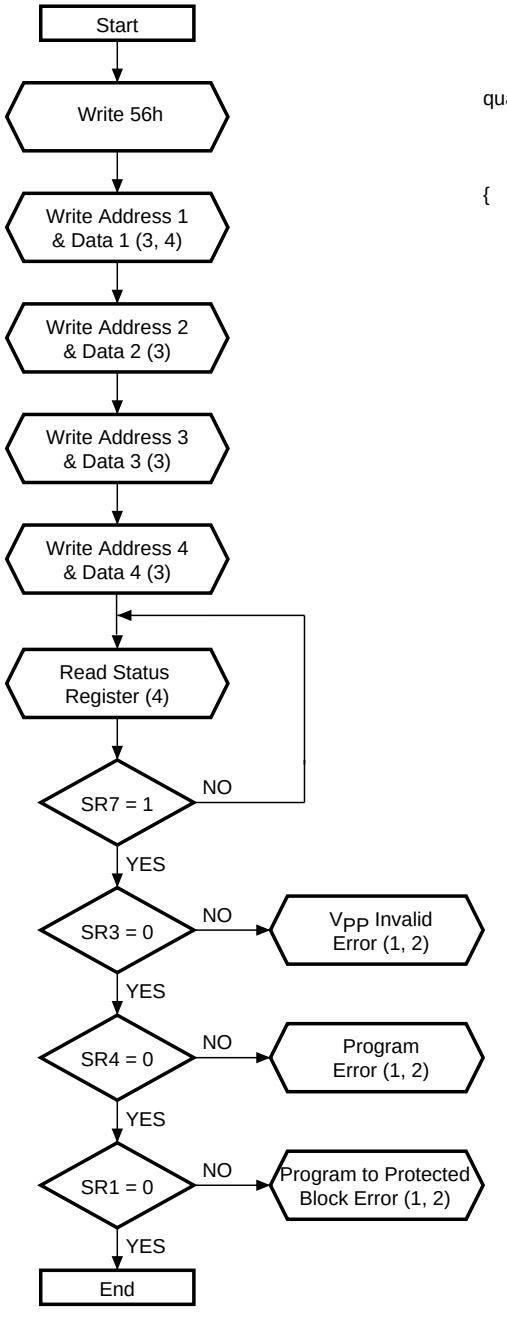



```
double_word_program_command (addressToProgram1, dataToProgram1,
                                        addressToProgram2, dataToProgram2)
{
        writeToFlash (addressToProgram1, 0x35);
                        /*see note (4)*/
        writeToFlash (addressToProgram1, dataToProgram1) ;
                                /*see note (3) */
        writeToFlash (addressToProgram2, dataToProgram2) ;
                                /*see note (3) */
         /*Memory enters read status state after
        the Program command*/

      do {
        status_register=readFlash (addressToProgram) ;
                                "see note (4)"
        /* E or G must be toggled*/

        } while (status_register.SR7== 0) ;

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

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

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

}
```

AI06171b

Note: 1. Status check of b1 (Protected Block), b3 (V_{PP} Invalid) and b4 (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 24. Quadruple Word Program Flowchart and Pseudo Code

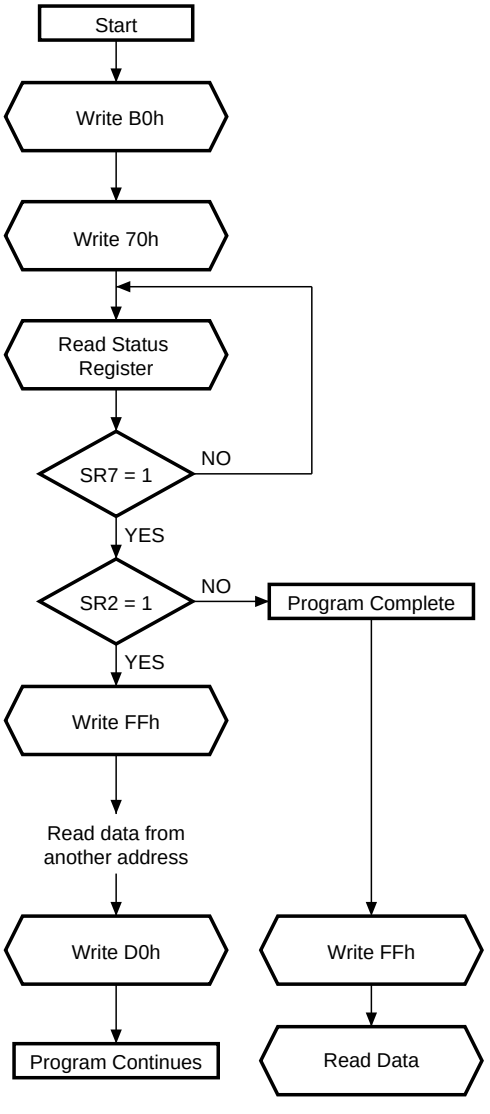



```
quadruple_word_program_command (addressToProgram1, dataToProgram1,
                                addressToProgram2, dataToProgram2,
                                addressToProgram3, dataToProgram3,
                                addressToProgram4, dataToProgram4)
{
        writeToFlash (addressToProgram1, 0x56);
                        /*see note (4) */

        writeToFlash (addressToProgram1, dataToProgram1) ;
                        /*see note (3) */

        writeToFlash (addressToProgram2, dataToProgram2) ;
                        /*see note (3) */

        writeToFlash (addressToProgram3, dataToProgram3) ;
                        /*see note (3) */

        writeToFlash (addressToProgram4, dataToProgram4) ;
                        /*see note (3) */

         /*Memory enters read status state after
        the Program command*/
      do {
        status_register=readFlash (addressToProgram) ;
                                /"see note (4) "/
        /* E or G must be toggled*/

        } while (status_register.SR7== 0) ;

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

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

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

}
```

AI06977b

Note: 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.

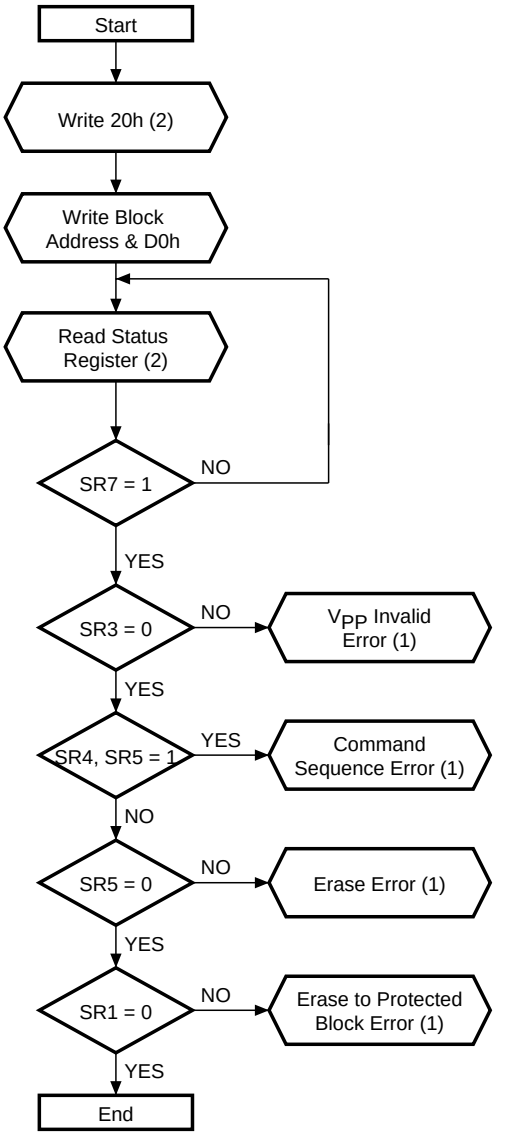
Figure 25. Program Suspend & Resume Flowchart and Pseudo Code




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

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


  do {
      status_register=readFlash (bank_address) ;
      /* E or G must be toggled*/




      } while (status_register.SR7== 0) ;



if  (status_register.SR2==0)  /*program completed */
      {  writeToFlash (bank_address, 0xFF)  ;
        read_data ( ) ; /*read data from another block*/
        /*The device returns to Read Array
        (as if program/erase suspend was not issued).*/



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

AI06173

Figure 26. Block Erase Flowchart and Pseudo Code

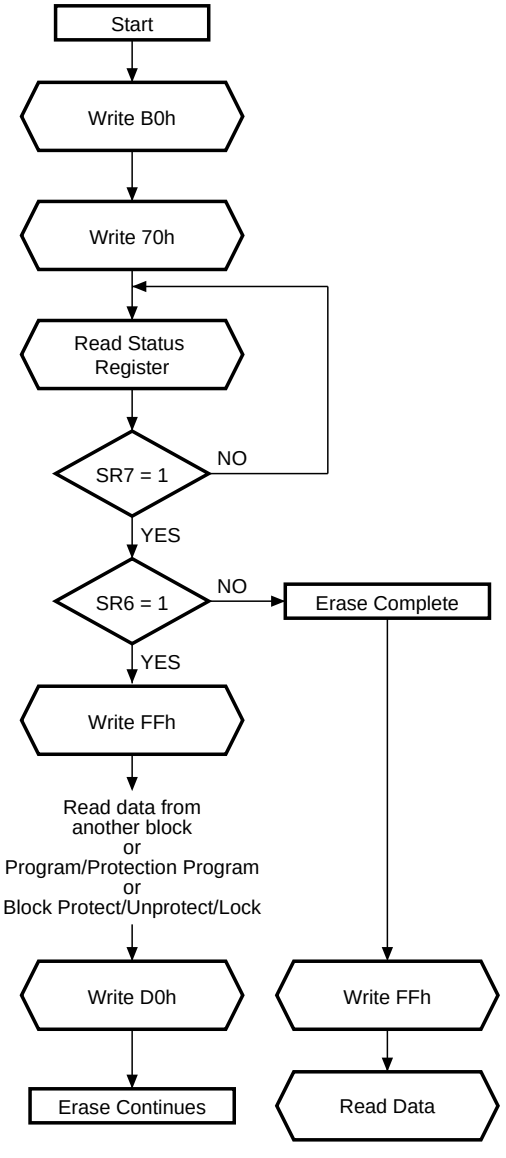



```
erase_command   ( blockToErase ) {
        writeToFlash (blockToErase, 0x20) ;
                                /*see note (2) */

        writeToFlash (blockToErase, 0xD0) ;
        /* only A12-A20 are significannt */
        /* Memory enters read status state after
        the Erase Command */


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




        } while (status_register.SR7== 0) ;


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


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


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


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


}
```

AI06174b

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

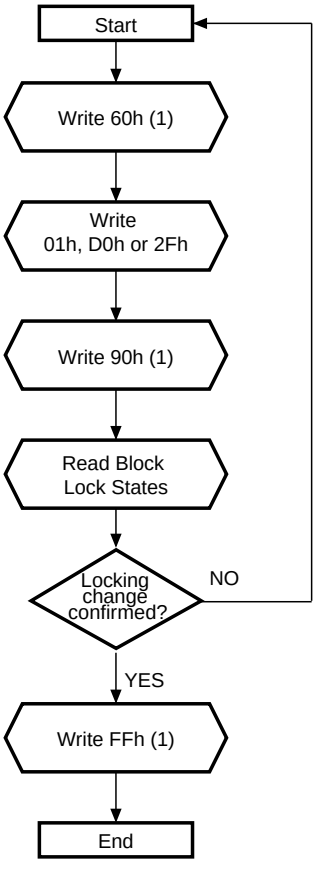
Figure 27. Erase Suspend & Resume Flowchart and Pseudo Code




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

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


     do {
         status_register=readFlash (bank_address) ;
         /* E or G must be toggled*/


        } while (status_register.SR7== 0) ;

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


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


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

AI06175

Figure 28. Locking Operations Flowchart and Pseudo Code

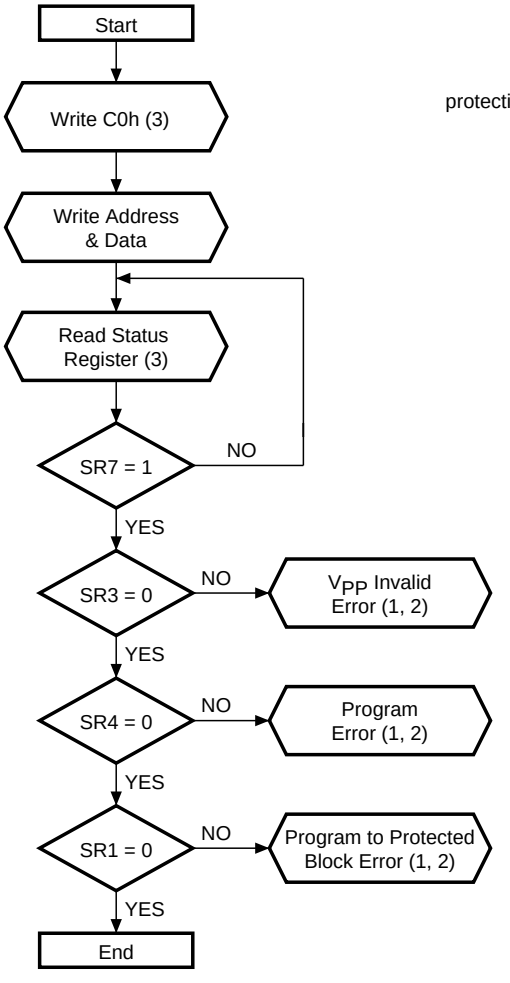



```
locking_operation_command   (address, lock_operation)  {
      writeToFlash (address, 0x60) ; /*configuration setup*/
                                    /* see note (1) */

      if  (lock_operation==LOCK)  /*to protect the block*/
            writeToFlash (address, 0x01)  ;
      else if  (lock_operation==UNLOCK)  /*to unprotect the block*/
            writeToFlash (address, 0xD0)  ;
      else if  (lock_operation==LOCK-DOWN)  /*to lock the block*/
            writeToFlash (address, 0x2F)  ;

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

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

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

}
```

AI06176b

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

Figure 29. Protection Register Program Flowchart and Pseudo Code

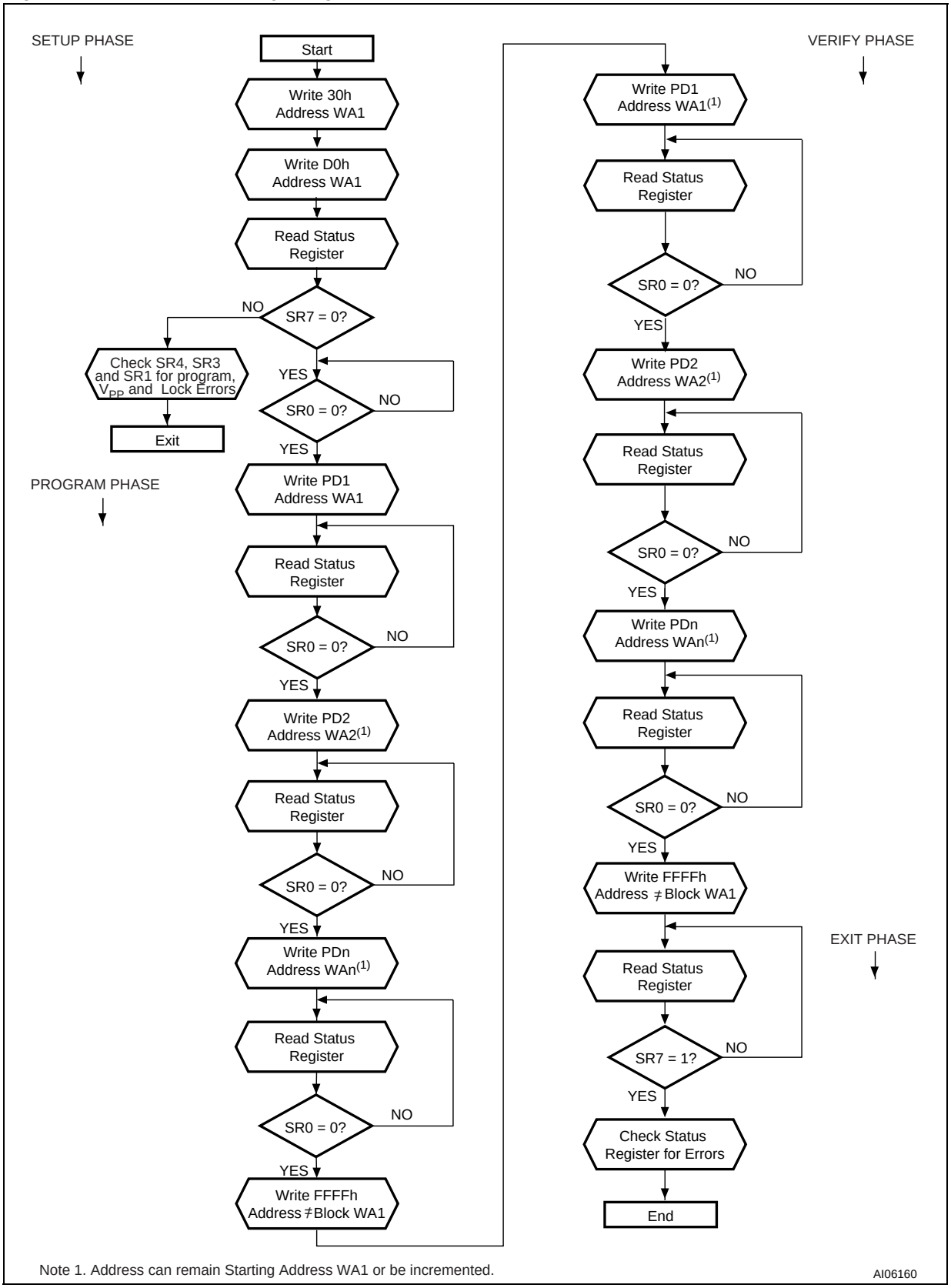



```
protection_register_program_command (addressToProgram, dataToProgram) {:
                        writeToFlash (addressToProgram, 0xC0) ;
                                        /*see note (3) */

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

                        do {
                          status_register=readFlash (addressToProgram) ;
                                                    /* see note (3) */
                          /* 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 ( ) ;

}
```

AI06177b

Note: 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 30. Enhanced Factory Program Flowchart



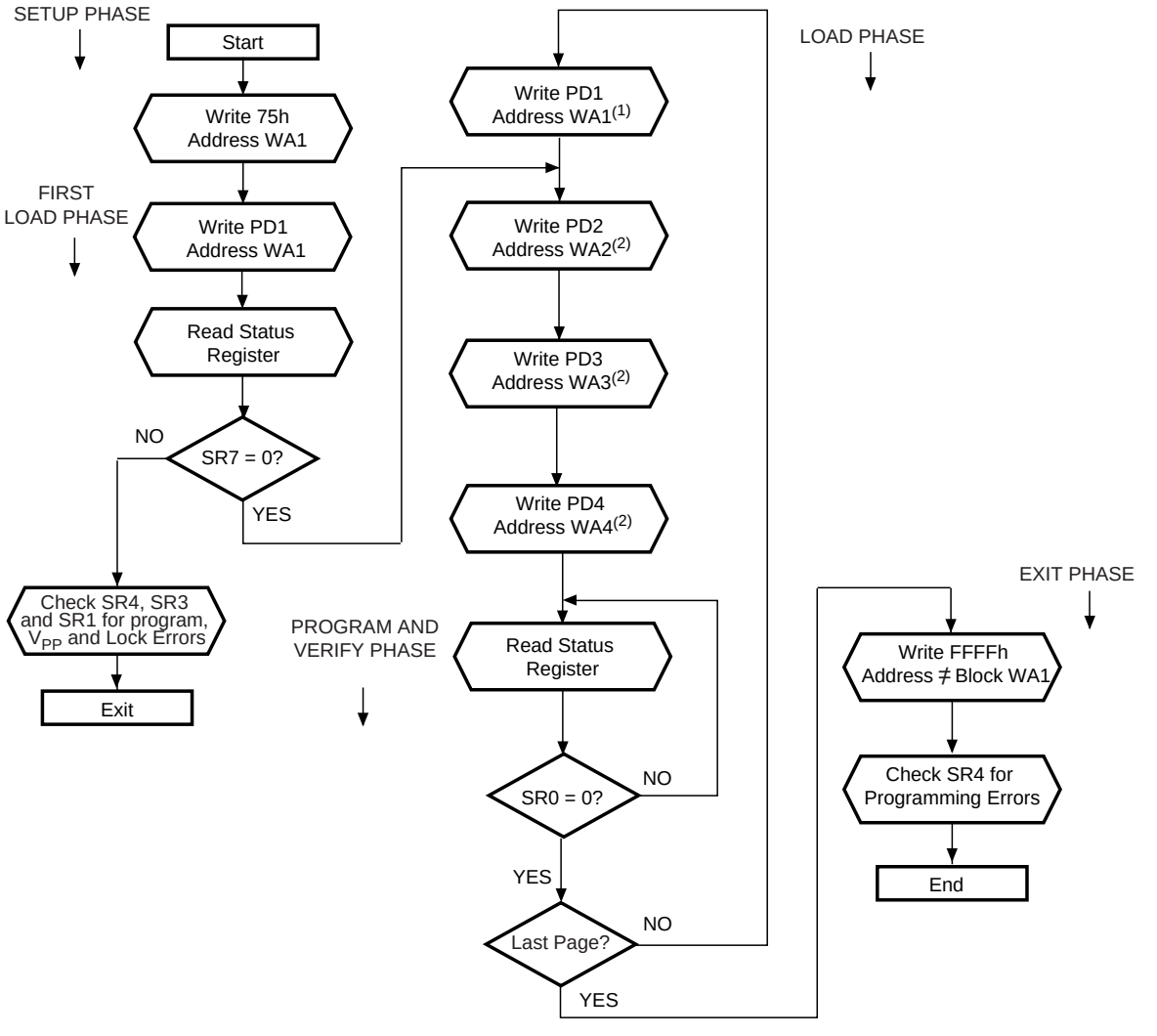
Note 1. Address can remain Starting Address WA1 or be incremented.

AI06160

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.b7==1){
                /*EFP aborted for an error*/
                if (status_register.b4==1) /*program error*/
                        error_handler();
                if (status_register.b3==1) /*VPP invalid error*/
                        error_handler();
                if (status_register.b1==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.b0==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.b0==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.b0==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.b7==0);
                if (status_register.b4==1) /*program failure error*/
                        error_handler();
                if (status_register.b3==1) /*VPP invalid error*/
                        error_handler();
                if (status_register.b1==1) /*program to protect block error*/
                        error_handler();
        }
}
```

Figure 31. Quadruple Enhanced Factory Program Flowchart



Note 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.

AI06178b

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 er-
ror*/
                                  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 44. Command Interface States - Modify Table, Next State

Current CI State		Next CI State After Command Input										
		Read Array(2)	Program WP setup (3,4)	Program DWP, QWP Setup (3,4)	Block Erase, Bank Erase Setup (3,4)	EFP Setup	Quad-EFP Setup	Erase Confirm P/E Resume, Block Unlock confirm, EFP Confirm	Program/ Erase Suspend	Read Status Register	Clear status Register (5)	Read Electronic signature, Read CFI Query
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 in Erase Suspend Busy										
	Busy	Program in Erase Suspend Busy							Program in Erase Suspend Suspended	Program in Erase Suspend Busy		
	Suspend	Program in Erase Suspend Suspended						Program in Erase Suspend Busy	Program in Erase Suspend Suspended			
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)										

Note: 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/E.C. is active, both cycles are ignored.
5. The Clear Status Register command clears the Status Register error bits except when the P/E.C. 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 45. Command Interface States - Modify Table, Next Output

Current CI State	Next Output State After Command Input (6)									
	Read Array(2)	Program DWP, QWP Setup (3,4)	Block Erase, Bank Erase Setup (3,4)	EFP Setup	Quad-EFP Setup	Erase Confirm P/E Resume, Block Unlock confirm, EFP Confirm	Program/ Erase Suspend	Read Status Register	Clear status Register (5)	Read Electronic signature, Read CFI Query
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	Status Register									
Lock/CR Setup in Erase Suspend										
OTP Busy	Array	Status Register				Output Unchanged		Status Register	Output Unchanged	Status Register
Ready	Array	Status Register				Output Unchanged		Status Register	Output Unchanged	Electronic Signature/ CFI
Program Busy										
Erase Busy										
Program/Erase										
Program in Erase Suspend Busy										
Program in Erase Suspend Suspended										

Note: 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/E.C. is active, both cycles are ignored.
5. The Clear Status Register command clears the Status Register error bits except when the P/E.C. is busy or suspended.
6. 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.

Table 46. Command Interface States - Lock Table, Next State

Current CI State		Next CI State After Command Input							
		Lock/CR Setup(4)	OTP Setup (4)	Block Lock Confirm	Block Lock-Down Confirm	Set CR Confirm	EFP Exit, Quad EFP Exit (3)	Illegal Command (5)	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 in Erase Suspend Busy							N/A
	Busy	Program in Erase Suspend Busy							Erase Suspended
	Suspend	Program in Erase Suspend Suspended							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 (2)					EFP Verify	EFP Busy(2)	N/A
	Verify	EFP Verify (2)					Ready	EFP Verify(2)	Ready
QuadEFP	Setup	Quad EFP Busy (2)							N/A
	Busy	Quad EFP Busy (2)					Ready	Quad EFP Busy(2)	Ready

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

Table 47. Command Interface States - Lock Table, Next Output

Current CI State	Next Output State After Command Input							
	Lock/CR Setup[3]	OTP Setup [3]	Block Lock Confirm	Block Lock-Down Confirm	Set CR Confirm	EFP Exit, Quad EFP Exit [2]	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		Output Unchanged
Lock/CR Setup in Erase Suspend								
OTP Busy	Status Register		Output Unchanged			Array	Output Unchanged	Output Unchanged
Ready	Status Register		Output Unchanged			Array	Output Unchanged	Output Unchanged
Program Busy								
EraseBusy								
Program/Erase								
Program in Erase Suspend Busy								
Program in Erase Suspend Suspended								

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

REVISION HISTORY

Table 48. Document Revision History

Date	Version	Revision Details
06-Sep-2002	1.0	First Issue
19-Dec-2002	1.1	Device Codes changed. VFBGA60 Package defined. 85ns Speed Class removed, 80ns Speed Class added. 70ns Speed Class characterized (certain timings modified). Command Interface description of invalid combinations clarified. Descriptions clarified: Clear Status Register Command, Program/Erase Suspend Command, Set Configuration Register Command, Factory Program Commands, WAIT signal's behavior. Tables 5, 7, 9 and 10 corrected. Notes to Figures 12, 13 and 14 modified. Flowcharts and Pseudo Code revised. CFI, Device Geometry Definition table address offsets 35h, 38h reserved. Revision History moved to end of document.
21-May-2003	1.2	Automatic Standby mode explained under Asynchronous Read Mode. Minor text changes in Clear Status Register Command, Quadruple Enhanced Factory Program Command and Synchronous Burst Read Mode. Bank Erase Command moved from the Standard to the Factory Program Commands. Number of Bank Erase cycles limited to 100. Erase replaced by Block Erase in Tables 11 and 12, Dual Operations Allowed in Other Banks and the Same Bank, respectively. I_{PP2} parameter for $V_{PP} = V_{PPH}$ removed from Table 18, DC Characteristics - Currents. Several cross-references corrected. V_{DDQ} range split into two in Tables 20 and 21, Asynchronous and Synchronous Read AC Characteristics: for V_{DDQ} = 2.2V to 3.3V, t_{AVQV1}, t_{ELTV}, t_{EHTZ}, t_{EHQZ}, t_{GLQV}, t_{AVLH}, t_{ELLH} and t_{LLLH} in Table 20 and all the timings in Table 21 were modified. Daisy chain information added (see Figures 20 and 21 and Table 27). Table 30, M58WR128ET - Block Addresses in Main Banks, corrected. CFI information corrected at offset (P+15)h = 4Eh in Table 40, and at offset (P+38)h = 71h in Table 43.