S26HL512T RPC Reduced Pin Count Flash Memory Test-Cases List

Table of Contents

1 References....5
2 Introduction....5
3 Test Sequences....6
3.1 Power Up....6
3.1.1 Series 1. Negative Test....6
a. POWERUP_1_a....6
3.2 Program Volatile Configuration Register....6
3.2.1 Series 2. Positive Tests....6
a. READS_2_a....6
b. READS_2_b....7
c. READS_2_c....7
Method of verification:....7
Program VCR command is issued incomplete, without second unlock cycle. Programming of VCR is not done, read latency is unchanged. Read memory array to verify that latency is not changed....7
3.3 CRC ASO....7
3.3.1 Series 3. Positive and Negative Tests....8
a. CRCASO_3_a....8
b. CRCASO_3_b....8
3.4 Word Program and Word Program Suspend....8
3.4.1 Series 4. Positive Tests....8
a. WORDPG_4_a....8
b. WORDPG_4_b....9
c. WORDPG_4_c....9
d. WORDPG_4_d....10
e. WORDPG_4_e....10
3.5 Write Buffer Programing....11
3.5.1 Series 5. Positive Tests....11
a. BUFFPG_5_a....11
b. BUFFPG_5_b....11
3.6 Sector Erase....14
3.6.1 Series 6. Positive Tests....14
a. SECERS_6_a....14
b. SECERS_6_b....14
3.7 Erase Suspend....16
3.7.1 Series 7. Positive Tests....16
a. ERSSUSP_7_a....16
b. ERSSUSP_7_b....17
c. ERSSUSP_7_c....18
d. ERSSUSP_7_d....18
e. ERSSUSP_7_e....19
f. ERSSUSP_7_f....20
g. ERSSUSP_7_g....21
3.8 SSR ASO....23
3.8.1 Series 8. Positive Tests....23
a. SSRASO_8_a....23

b. SSRASO_8_b....23
c. SSRASO_8_c....24
d. SSRASO_8_d....24
e. SSRASO_8_e....25
f. SSRASO_8_f....25
g. SSRASO_8_g....25
h. SSRASO_8_h....26
i. SSRASO_8_i....27
3.9 ASP register....27
3.9.1 Series 9. Positive Test....27
a. ASPREG_9_a....27
b. ASPREG_9_b....28
3.10 RESET....28
3.10.1 Series 10. Negative Tests....28
a. RESET_10_a....28
b. RESET_10_b....29
3.11 DYB ASO....30
3.11.1 Series 11. Positive test....30
3.12 Password ASO....31
3.12.1 Series 12. Positive Test....31
3.13 PPB ASO....31
3.13.1 Series 13. Positive and negaitve tests....31
3.14. Protected sectors....33
3.14.1 Series 14. Negative tests....33
3.15 Chip Erase....34
3.15.1 Series 15. Positive Test....34
3.16 PPB Lock bit....34
3.16.1 Series 16. Positive and negative tests....34
3.17 Force DYB bits after HW Reset....36
3.17.1 Series 17. Positive test....36
3.18 xVCR Program negative, FREEZE set....36
3.18.1 Series 18. Negative tests....36
3.19 Measure Temperature....38
3.19.1 Series 19. Positive test....38
3.20 Deep Power Down....38
3.20.1 Series 20. Positive test....38
3.21 Sector erase....39
3.21.1 Series 21. Negative test....39
3.22 Persistent Protection Mode....39
3.22.1 Series 22. Positive test....39
3.23 Password Mode....40
3.23.1 Series 23. Positive and negative tests....40
3.24 Autoboot....43
3.24.1 Series 24. Positive tests....43
3.25 SFDP Test....44
3.25.1 Series 25. Positive test....44
3.26 Write Enable/Write Disable....44
3.26.1 Series 26. Positive Test....44
3.27 Initial configuration (TBPARM) 44
3.27.1 Series 27. Positive Test 44

3.28 READ Test....45
3.28.1 Series 28. Positive test....45
3.29 Read Registers....45
3.29.1 Series 29. 45
3.30 OTP READ....47
3.30.1 Series 30. Positive test....47
3.31SWL 47
3.31.1 Series 31. Positive test....47
3.32 OTP Program – Positive Tests....48
3.32.1 Series 32. Positive Tests....48
3.33 Page Programming (4 bytes address)....48
3.33.1 Series 33. Positive Tests....48
3.34 Program Suspend....49
3.34.1 Series 34. Positive Test 49
3.35 PPB Bit Programming....50
3.35.1 Series 35. Positive Test 50
3.36 PPB Erase....50
3.36.1 Series 36. PPB Bits Tests 50
3.37 DYB Bit Programming....50
3.37.1 Series 37. Positive test....50
3.38 Sector Erase....51
3.38.1 Series 38. Positive tests....51
3.39 Program During Erase Suspend....51
3.39.1 Series 39. 51
3.40 Bulk Erase....52
3.40.1 Series 40. Positive Tests 52
3.41 CRC ASO....52
3.41.1 Series 41. Positive Tests 52
3.42 SPI DPD....53
3.42.1 Series 42. Positive test....53
3.43 Sector Erase Count....53
3.43.1 Series 43. Positive test....53
3.44 JEDEC RESET....54
3.44.1 Series 44. Positive test....54
3.45 READ Jedec ID, SFDP, Unique ID, WRAR for STRV1,....54
3.45.1 Series 45. Positive test....54

1 References

- [1] 002_12337_0A_V_HB_512to4G_3V_18V.pdf, September 10, 2016
- [2] spansion_flash_testbench_struct.pdf

2 Introduction

This document defines the test cases which are together with testbench defined in [3] used to verify the functional behavior and timing of the S26HL512T Flash memory VITAL model. Test-Cases apply on flash memory models listed in the Table 1.

Table 1. Flash memory models

Model name	***Memory size***
S26HL512T	512Mb (64MB)

Functional behavior and timing is described in the appropriate data sheet [1].

Test cases are grouped in several sequences:

- Power up
- Various reads, memory array and different ASOs
- CRC ASO
- Word Programming and Suspend of Word Program
- Buffer Programming of 1 word, several words and full buffer
- Programming during Write Protect (WP# low)
- Sector Erase and Parameter Sector Erase
- Sector Erase Suspend, Programming in Erase Suspend
- SSR ASO
- ASP register
- Hard reset during program and erase operations
- DYB ASO
- Password ASO
- PPB ASO
- Protection of sectors
- Chip Erase
- PPB Lock bit
- Persistent Protection
- Password Protection

Each sequence consists of series of Positive and Negative tests

Each test case has the code name complying with following rule:

XXX_YYY_ZZZ where:

XXX ± Test Sequence name.

YYY ± Test series number

ZZZ ± Test case (a, b, c, ...)

3 Test Sequences

3.1 Power Up

3.1.1 Series 1. Negative Test

a. POWERUP_1_a

Testcase symbolic name*:* POWERUP_1_a

Description*:* During POR, the device can not be selected, will not accept commands, and does not drive outputs

What is being verified:

The device is selected during POR. Verify that command is ignored, output is not driven and warning is generated. After power-up the device resets the internal state machine to reading array data. All output signals should be Hi-Z and model should be ready to read preloaded memory data.

Method of verification:

During power up time initiate read operation to verify that outputs are Hi-Z.

3.2 Program Volatile Configuration Register

3.2.1 Series 2. Positive Tests

a. READS_2_a

Testcase symbolic name*:* READS_2_a

Description*:* Read from memory array and different ASOs

What is being verified:

Correct data from memory array and different ASOs

Method of verification:

Read commands are issued in both Continuous and Wrapped mode for memory array. Read commands are issued for Status register read, first cycle is read status register enter with combination of address and data 555h/70h, second cycle is when data is outputting and verified. Read of VCR and NVCR registers, with two unlock cycles (555h/AA and 2AAh/55h) followed by commands for VCR read (555h/C7h) and NVCR read (555h/C6h). Read of ID SFDP ASO is done. First ID SFDP enter command is set (555h/98h) and once the ID SFDP ASO is entered the read of ID SFDP data is verified. When read is done, ID SFDP is exited with ASO exit command (XXXh/F0h or Ffh).

b. READS_2_b

Testcase symbolic name*:* READS_2_b

Description*:* Read from memory array and different ASOs with different latencies

and burst length

What is being verified:

Correct data from memory array and different ASOs for different latency and burst length

Method of verification:

Program VCR to set read latency to 11 clock cycles. First set command for programming VCR, two unlock cycle followed by 555h/38h address/data command.

Read commands are issued in both Continuous and Wrapped mode for memory array. Program VCR again to set BurstLength to 64B.Read commands are issued in both Continuous and Wrapped mode for memory array. Program VCR for new latency and verify the data output for set latency.

Program VCR to set incorrect value for read latency and check if the default is automatically set to 16 clock cycles. Read data from memory array.

c. READS_2_c

Testcase symbolic name*:* READS_2_c

Description*:* Incomplete program VCR command to set read latency. Read latency unchanged

What is being verified:

Incomplete command of program VCR will not change read latency.

***Method of verification*:**
Program VCR command is issued incomplete, without second unlock cycle. Programming of VCR is not done, read latency is unchanged. Read memory array to verify that latency is not changed.

3.3 CRC ASO

3.3.1 Series 3. Positive and Negative Tests

a. CRCASO_3_a

Testcase symbolic name: CRCASO_3_a

Description: Test CRC ASO, CRC calculation and read of CRC registers

What is being verified:

Entering CRC ASO and starting CRC calculation over a part of memory array. Read of CRC registers to verify correct calculation.

Method of verification:

Set CRC ASO enter command cycles, two unlock cycles (555h/AAh, 2AAh/55h) followed by CRC ASO entry command (555h/78h). Set starting address for CRC calculation, address of the starting point and data part of command is C3h. Set ending address for CRC calculation, address of ending point and data part of command is 3Ch. This command will start the CRC calculation. Read status register to check if device is busy. Wait until calculation is done. Read status register to check if device is ready. Read CRC Low and High registers to match the calculated value. Set ASO reset command (XXXh/F0h) to exit CRC ASO.

b. CRCASO_3_b

Testcase symbolic name: CRCASO_3_b

Description: Negative Test for CRC ASO and CRC calculation

What is being verified:

CRC ASO is entered, command for CRC calculation is incorrect, ending address is not correct. CRC calculation is not done.

Method of verification:

Set CRC ASO enter command cycles, two unlock cycles (555h/AAh, 2AAh/55h) followed by CRC ASO entry command (555h/78h). Set starting address for CRC calculation, address of the starting point and data part of command is C3h. Set ending address which is not starting address +2 or higher. This will not cause CRC calculation. Read status register to check device ready/busy bit, device should be ready. Read status register to check abort status bit, after unsuccessful CRC calculation bit 6 of status register is set to 1. Exit CRC ASO by setting ASO exit command.

3.4 Word Program and Word Program Suspend

3.4.1 Series 4. Positive Tests

a. WORDPG_4_a

Testcase symbolic name: WORDPG_4_a

Description: Test Word Programming

What is being verified:

Correct execution of Word Program command

Method of verification:

Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by word program command (555h/A0h). After that is cycle with data that is programmed into the memory and address to which data is programmed. Read status register to check if device is busy (Status_reg[7] bit). Wait for tSWP which is time needed for single word program to execute. After tSWP, read status register to check if device is ready. Read memory array from the programmed location to check if data is programmed correctly.

b. WORDPG_4_b

Testcase symbolic name*:* WORDPG_4_b

Description*:* Test Word Programming Suspend

What is being verified:

Correct execution of Program Suspend command. Entering and reading ID SFDP ASO with two unlock cycles.

Method of verification:

Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by word program command (555h/A0h). After that is cycle with data that is programmed into the memory and address to which data is programmed. Read status register to check if device is busy (Status_reg[7] bit). Set Program Suspend command with XXXh/51h combination of address/data in the write cycle. Read status register to check if device is not yet in program suspend state. Wait for tPSL. After time tPSL, read status register to check if now device is in program suspend state. Set ID SFDP entry command cycles to enter ID SFDP ASO while in program suspend, first two unlock cycles followed by ID SFDP entry command 555h/90h. Read ID SFDP ASO from different locations, and after that exit ID SFDP ASO by setting ASO exit command XXXh/F0h. Resume programming by setting program resume command cycle XXXh/50h. Read status register to check if program is no longer in suspend state and that device is busy. Wait for tSWP, until word program is done. Read programmed location to check if the data is correctly programmed.

c. WORDPG_4_c

Testcase symbolic name*:* WORDPG_4_c

Description*:* Test Word Programming Suspend and ID SFDP ASO

What is being verified:

Correct execution of Program Suspend command. Entering and reading ID SFDP ASO without unlock cycles.

Method of verification:

Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by word program command (555h/A0h). After that is cycle with data that is programmed into the memory and address to which data is programmed. Read status register to check if device is busy (Status_reg[7]

bit). Set Program Suspend command with XXXh/51h combination of address/data in the write cycle. Read status register to check if device is not yet in program suspend state. Wait for tPSL. After time tPSL, read status register to check if now device is in program suspend state. Set ID SFDP entry command cycle to enter ID SFDP ASO while in program suspend, only one cycle with 55h/98h combination of address and data in write bus cycle. Read ID SFDP ASO from different locations, and after that exit ID SFDP ASO by setting ASO exit command XXXh/F0h. Resume programming by setting program resume command cycle XXXh/50h. Read status register to check if program is no longer in suspend state and that device is busy. Wait for tSWP, until word program is done. Read programmed location to check if the data is correctly programmed.

d. WORDPG_4_d

***Testcase symbolic name**:* WORDPG_4_d

***Description**:* Test Program oparation in Word Programing Suspend mode

What is being verified:

Program operation while device is in Program Suspend. Program should fail

***Method of verification*:**

Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by word program command (555h/A0h). After that is cycle with data that is programmed into the memory and address to which data is programmed. Read status register to check if device is busy (Status_reg[7] bit). Set Program Suspend command with XXXh/51h combination of address/data in the write cycle. Read status register to check if device is not yet in program suspend state. Wait for tPSL. After time tPSL, read status register to check if now device is in program suspend state. Read from a suspended line, device should return Xs.

Set Word Program command cycles, the device is in Program Suspend mode. Read Status register to check if device is not busy, to check if the last program operation has failed. Set Clear Status register command to clear the status bits set by the previous failed program attempt.

Resume programming by setting program resume command cycle XXXh/50h. Read status register to check if program is no longer in suspend state and that device is busy. Wait for tSWP, until word program is done. Read programmed location to check if the data is correctly programmed.

e. WORDPG_4_e

***Testcase symbolic name**:* WORDPG_4_e

***Description**:* Test Erase oparation in Word Programing Suspend mode

What is being verified:

Erase operation while device is in Program Suspend. Program should fail

***Method of verification*:**

Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by word program command (555h/A0h). After that is cycle with data that is programmed into the memory and address to which data is programmed. Read status register to check if device is busy (Status_reg[7] bit). Set Program Suspend command with XXXh/51h combination of address/data in the write cycle. Read status register to check if device is not yet in program suspend state. Wait for tPSL. After time tPSL, read status register to check if now device is in program suspend state. Read from a suspended line, device should return Xs.

Set Erase command cycles, the device is in Program Suspend mode. Read Status register to check if device is not busy, to check if the last erase operation has failed. Set Clear Status register command to clear the status bits set by the previous failed program attempt.

Resume programming by setting program resume command cycle XXXh/50h. Read status register to check if program is no longer in suspend state and that device is busy. Wait for tSWP, until word program is done. Read programmed location to check if the data is correctly programmed.

f. WORDPG_4_f

Testcase symbolic name. WORDPG_4_f

Description. Word programming using burst write capabilities

What is being verified:

Test model for Word programming using burst write capabilities

***Method of verification*:**

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and A0h data to enter word program state. Set write command cycle for burst write with first data and starting address. Read status register to check if device is busy (Status register bit 7). Wait for tSWP until programming is done. Read status register to check if device is ready and if programming was a success. Read memory array from programmed location to check if data is correctly programmed.

3.5 Write Buffer Programing

3.5.1 Series 5. Positive Tests

a. BUFFPG_5_a

Testcase symbolic name: BUFFPG_5_a

Description*:* Buffer Programming of one word

What is being verified: Test model for Buffer Programming of one word.

Method of verification*:*

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count equal to 0, to set only word to be programmed. After that, set write cycle with starting line address and data that is programmed into the memory array. Followed by data cycle is confirm write cycle to instruct the memory that buffer write cycle is done. Read status register to check if device is busy (Status register bit 7). Wait for tSWP until programming is done. Read status register to check if device is ready and if programming was a success. Read memory array from programmed location to check if data is correctly programmed.

b. BUFFPG_5_b

Testcase symbolic name: BUFFPG_5_b

Description*:* Buffer Programming of one page (16B)

What is being verified: Test model for Buffer Programming of one page.

Method of verification*:*

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count equal to 0, to set only word to be programmed. After that, set write cycles with starting line address and data that is programmed into the memory array. There are total of 8 write data cycles, one write cycle writes 2 bytes of data. Followed by data cycle is confirm write cycle to instruct the memory that buffer write cycle is done. Read status register to check if device is busy (Status register bit 7). Wait for tWBP until programming is done. Read status register to check if device is ready and if programming was a success. Read memory array from programmed location to check if data is correctly programmed.

c. BUFFPG_5_c

Testcase symbolic name: BUFFPG_5_c

Description*:* Buffer Programming of entire buffer

What is being verified: Test model for Buffer Programming of entire buffer (512B)

Method of verification*:*

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count equal to 0, to set only word to be programmed. After that, set write cycles with starting line address and data that is programmed into the memory array. There are total of 256 write data cycles, one write cycle writes 2 bytes of data. Followed by data cycle is confirm write cycle to instruct the memory that buffer write cycle is done. Read status register to check if device is busy (Status register bit 7). Wait for tWBP until programming is done. Read status register to check if device is ready and if programming was a success. Read memory array from programmed location to check if data is correctly programmed.

d. BUFFPG_5_d

Testcase symbolic name: BUFFPG_5_d

Description*:* Negative test of Buffer Programming, wrong sector address

What is being verified: Test model for incorect Buffer Programming command sequence

Method of verification*:*

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Following is incorrect write command cycle for Word Count with different sector address than the one used in previous write

cycle which, for entering Write Buffer state. Read status register to check if device is not busy and to check program fail and buffer abort status bits.

Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set clear status register command to clear the Status register bits set by incorrect Write Buffer command.

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set write command cycle for Word Count correctly, with proper SA. Set write command cycle for writing data into the buffer with incorrect sector address. Read status register to check if device is not busy and to check program fail and buffer abort status bits.

Set Write to Buffer Abort Reset command to exit Program Error State, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set clear status register command to clear the Status register bits set by incorrect Write Buffer command.

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set write command cycle for Word Count correctly, with proper SA and all following write command cycles for write of data into the buffer. Set incorrect write command cycle for Buffer Confirm command. Read status register to check if device is not busy and to check program fail and buffer abort status bits.

Set clear status register command to clear the Status register bits set by incorrect Write Buffer command.

Set clear status register command to clear the Status register bits set by incorrect Write Buffer command.

e. BUFFPG_5_e

Testcase symbolic name: BUFFPG_5_e

Description*:* Burst write

What is being verified: Test model for Burst Write Buffer Programming command sequence

Method of verification:

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count to be programmed. Set write command cycle for burst write with first data and starting address. In this command cycle CS# is kept low for entire period of data cycles. Number of data cycles is equal to Word Count +1. Read Status register to check if device is busy. Wait untili device complets Buffer Programing, tWBP time. Read Status register to check if device is not busy and if programming was a success.

Read memory array locations on which data was programmed.

3.6 Sector Erase

3.6.1 Series 6. Positive Tests

a. SECERS_6_a

***Testcase symbolic name**:* SECERS_6_a

***Description**:* Test Sector Erase operation.

What is being verified:

Sector Erase operation on non-empty sector.

Method of verification:

Read from several locations of the sector that is going to be erased. Verify that data is different than Ffh, which means that sector is not empty. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh) followed by write command cycle with SAh/30h address/data combination. This will put the device in Sector Erase state and the device will erase sector defined by SA in last command cycle. Read status register to check if device is busy. Wait for tSEO, time needed to complete erase operation. Read status register to check if erase operation is done and device is ready. Read status register to check if erase operation was successful, by checking Status register bit 5. Read from the same locations of memory array, like the ones used before erase operation, to check if data is FFh. Set Evaluate Erase Status command cycle with (SA)555h/D0h address/data combination to check if the previous erase operation was successful. Once the Evaluate Erase Status is set, device goes busy. Read status register to check if the device is busy. Wait for tEES, which the time needed to execute EES command. After that, read status register to check if the device is ready and that Sector Erase Status bit of Status register (bit 0) is set ot 1, indicating that last sector erase operation was successful.

b. SECERS_6_b

***Testcase symbolic name**:* SECERS_6_b

***Description**:* Test Sector Erase operation on parameter sector

What is being verified:

Sector Erase operation on Parameter Sector 4KB mapped on Low address sector

Method of verification:

Program VCR to set Parameter sector mapping to Low address, by writing value 2600 to bits [9:8] of VCR. Bits 9 and 8 of VCR register determine how sectors are mapped, in this case parameter sectors are mapped to Low address. This means that SA00 (256KB) is divided into 8 parameter sectors of 4KB and one of 224KB.

Read from the sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles

(555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination, where SA is 00. The rest of the address bits are set to 0 which means that the lowest parameter 4KB sector is going to be erased. Read status register to check if the device is busy. Wait for tPSEO, time needed for a parameter sector to be erased. After that, read status register to check if the device is ready. Read status register to check Erase Status bit, to check if the erase was successful. Read from the same locations as the ones on the start of the test, to check if the data is FFh which means that parameter sector is erased.

c. SECERS_6_c

***Testcase symbolic name**:* SECERS_6_c

***Description**:* Test Sector Erase operation on parameter sector

What is being verified:

Sector Erase operation on Parameter Sector 224KB mapped to Low address sector

Method of verification:

From previous test, VCR remained the same, so parameter sectors are mapped to Low address sector.

Read from the sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination, where SA is 00. The rest of the address bits are set to 4001h which means that the highest parameter 224KB sector is going to be erased. Read status register to check if the device is busy. Wait for tSEO, time needed for a parameter sector to be erased. After that, read status register to check if the device is ready. Read status register to check Erase Status bit, to check if the erase was successful. Read from the same locations as the ones on the start of the test, to check if the data is FFh which means that parameter sector is erased.

d. SECERS_6_d

***Testcase symbolic name**:* SECERS_6_d

***Description**:* Test Sector Erase operation on parameter sector

What is being verified:

Sector Erase operation on Parameter Sector 4KB mapped to High address sector

Method of verification:

Program VCR to set Parameter sector mapping to High address, by writing value 2'b01 to bits [9:8] of VCR. Bits 9 and 8 of VCR register determine how sectors are mapped, in this case parameter sectors are mapped to High address sector. This means that SA255 (256KB) is divided into 8 parameter sectors of 4KB and one of 224KB. Parameter sector 224KB is mapped to lowest address of SA255, and rest of the 4KB sectors are mapped to the highest address of SA255.

Read from the sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles

(555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination, where SA is 255. The rest of the address bits are set to 0 which means that the lowest parameter 224KB sector is going to be erased. Read status register to check if the device is busy. Wait for tSEO, time needed for a parameter sector to be erased. After that, read status register to check if the device is ready. Read status register to check Erase Status bit, to check if the erase was successful. Read from the same locations as the ones on the start of the test, to check if the data is FFh which means that parameter sector is erased.

e. SECERS_6_e

***Testcase symbolic name**:* SECERS_6_d

***Description**:* Test Sector Erase operation on parameter sector

What is being verified:

Sector Erase operation on Parameter Sector 4KB mapped to High address sector

Method of verification:

Read from the sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination, where SA is 255. The rest of the address bits are set to 1F800h which means that the highest parameter 4KB sector is going to be erased. Read status register to check if the device is busy. Wait for tSEO, time needed for a parameter sector to be erased. After that, read status register to check if the device is ready. Read status register to check Erase Status bit, to check if the erase was successful. Read from the same locations as the ones on the start of the test, to check if the data is FFh which means that parameter sector is erased.

3.7 Erase Suspend

3.7.1 Series 7. Positive Tests

a. ERSSUSP_7_a

***Testcase symbolic name**:* ERSSUSP_7_a

***Description**:* Test Erase Suspend operation

What is being verified:

Erase Suspend mode

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination

where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Read memory array from various locations, not sector that is set to be erased. Verify correct data on output. Read from suspended sector and verify that output is X. Enter ID SFDP ASO by setting write command cycle for ID SFDP, (SA)55h/98h address/data combination. Read ID SFDP ASO from various locations. If read command for ID SFDP is set with different SA than the one set for ID SFDP entry, then output is X. Read ID SFDP ASO with different SA and verify data on output to be undetermined (Xs). Exit ID SFDP ASO with ASO exit command write cycle. Read status register to check if device has returned to suspend state, check erase suspend status bit. Again, enter ID SFDP ASO this time with different command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with (SA)555h/90h address/data combination. Read ID SFDP ASO from various locations. Read ID SFDP ASO with different SA and verify data on output to be undetermined (Xs). Exit ID SFDP ASO with ASO exit command write cycle.

Set Erase Resume command write cycle with XXXh/30h address/data combination. Read status register to check if device is busy, considering that erase is resumed. Wait for tSEO, time needed for device to complete erase operation. Read status register to check if device is ready.

Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

b. ERSSUSP_7_b

***Testcase symbolic name**:* ERSSUSP_7_b

***Description**:* Test Program operation in Erase Suspend

What is being verified:

Erase Suspend mode, word program operation in Erase Suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Set word program command cycles, with address of non-suspended sector and data to be programmed into the memory array. Read status register to check if device is busy. Wait for tSWP, time needed for device to complete program operation. Read status register to check if device is ready. Read the programmed location and verify correct data programmed. Read status register to check if device is in erase suspend mode. Set erase resume command write cycle. Read status register to check if device is no longer in erase suspend mode. Read status register to check if device is busy. Wait for tSEO, time needed for erase operation. Read status register to check if device is ready.

Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Read memory array during Blank Check, device should return Xs. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

c. ERSSUSP_7_c

***Testcase symbolic name**:* ERSSUSP_7_b

***Description**:* Test Buffer Program operation in Erase Suspend

What is being verified:

Erase Suspend mode, buffer program operation in Erase Suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Set buffer program command cycles, with address of non-suspended sector and data to be programmed into the memory array. Read status register to check if device is busy. Wait for tWBP, time needed for device to complete program operation. Read status register to check if device is ready. Read the programmed location and verify correct data programmed. Read status register to check if device is in erase suspend mode. Set erase resume command write cycle. Read status register to check if device is no longer in erase suspend mode. Read status register to check if device is busy. Wait for tSEO, time needed for erase operation. Read status register to check if device is ready.
Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Read memory array during Blank Check, device should return Xs. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

d. ERSSUSP_7_d

***Testcase symbolic name**:* ERSSUSP_7_b

***Description**:* Test Buffer Program Suspend operation in Erase Suspend

What is being verified:

Erase Suspend mode, buffer program suspend operation in Erase Suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Set buffer program command cycles, with address of non-suspended sector and data to be programmed into the memory array. Read status register to check if device is busy. Set program suspend command write cycle, with XXXh/51h address/data combination. Read status register to check if device is still busy, it takes tPSL time for device to enter program suspend stata. After tPSL, read status register to check if device is in program suspend mode.
Enter ID SFDP ASO while in Erase Suspend/Program Suspend mode. Set ID SFDP entry command write cycles, two unlock cycles (555h/AAh, 2AAh/55h) followed by ID SFDP entry command with 555h/90h address/data combination. Read several ID SFDP locations and exit ID SFDP ASO by setting write command cycle for ASO exit. Clear status register by setting write command cycle with 555h/71h address/data combination.
Enter SSR ASO while in Erase Suspend/Program Suspend mode. Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by SSR ASO entry command with 55h/88h address/data combination. Read several SSR ASO locations and verify that correct data is read from SSR ASO. Exit SSR ASO by setting write command cycle for ASO exit.
Set program resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in program suspend state but still in erase suspend state. Read status register to check if the program operation was successful.
Set erase resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in erase suspend mode.
Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful. Read memory array from the programmed sector.

e. ERSSUSP_7_e

Testcase symbolic name: ERSSUSP_7_e

Description: Test Buffer Program Suspend operation in Erase Suspend, program resume command with 2 unlock cycles

What is being verified:

Program resume command with two unlock cycles while device is in erase suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is

performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Set buffer program command cycles, with address of non-suspended sector and data to be programmed into the memory array. Read status register to check if device is busy. Set program suspend command write cycle, with XXXh/51h address/data combination. Read status register to check if device is still busy, it takes tPSL time for device to enter program suspend stata. After tPSL, read status register to check if device is in program suspend mode.
Set read command for memory array of a suspended sector and verify that device returns undiefined data.
Set command cycles for Program Resume, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/50h address/data combination. Read Status register to check if device is no longer in program suspend mode but still in erase suspend mode.
Set erase resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in erase suspend mode.
Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful. Read memory array from the programmed sector.

f. ERSSUSP_7_f

***Testcase symbolic name**:* ERSSUSP_7_f

***Description**:* Negative test, Buffer program with wrong sector addres in Erase suspend mode

What is being verified:

Buffer program failure due to wrong sector address while in Erase Suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend. Set buffer program command cycles, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/25h. Set write command cycle with Word Count which has different sector address than the previous command. Read status register to check if device is not busy and if program status and write buffer abort status bits are set. Set Clear Status register command cycle to clear the status bits set by previous incorrect Buffer Programm commands.
Set buffer program command cycles, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/25h. Set write command cycle with Word Count with correct SA. Next write

command is data write into the buffer but the SA is incorrect. Read status register to check if device is not busy and if program status and write buffer abort status bits are set.
Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set Clear Status register command cycle to clear the status bits set by previous incorrect Buffer Programm commands.
Set buffer program command cycles, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/25h. Set write command cycle with Word Count with correct SA. All following write commands for data write into the buffer are with correct SA. Set Write Buffer Confirm command with wron SA. Read status register to check if device is not busy and if program status and write buffer abort status bits are set.
Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set Clear Status register command cycle to clear the status bits set by previous incorrect Buffer Programm commands.
Set erase resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in erase suspend mode.
Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

g. ERSSUSP_7_g

***Testcase symbolic name**:* ERSSUSP_7_g

***Description**:* Burst Write in Erase Suspend mode

What is being verified:

Burst Write program opartion while in Erase Suspend

Method of verification:

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend.
Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count to be programmed. Set write command cycle for burst write with first data and starting address. In this command cycle CS# is kept low for entire period of data cycles.

Number of data cycles is equal to Word Count +1. Read Status register to check if device is busy. Wait untili device complets Buffer Programing, tWBP time. Read Status register to check if device is not busy and if programming was a success. Read Status register to check if device is still in erase suspend mode. Read memory array locations on which data was programmed.

Set erase resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in erase suspend mode.

Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

h. ERSSUSP_7_h

***Testcase symbolic name**:* ERSSUSP_7_h

***Description**:* Program operation on Erase Suspended sector

What is being verified:

Program operation fails on Erase Suspended sector.

***Method of verification*:**

Read from sector that is going to be erased. Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy. Set Erase Suspend write command cycle with XXXh/B0h address/data combination. Read status register to check Erase Suspend status bit, should be 0 because takes time of tESL to enter Erase Suspend state. Wait for tESL, then read status register to check if Erase Suspend bit is set to 1, which indicates that device is in Erase Suspend.
Set Word Program command cycles, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle 555h/A0h. After this, set write command cycle for word program operation with PAh/PDh for address and data to be programmed. Program address is part of the sector that is in erase suspend, program should fail. Read status register to check if program status bit indicates failure. Set Clear Status register command to clear the status bit set by previous unsuccessful program operation.
Set Buffer Program command cycles, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle 555h/A0h. After this, set write command cycle for word count of buffer programming, SAh/WCh. Sector to be programmed at is the same as the erase suspended sector. Program operation should fail, read status register to check Program Status bit and verify that program operation has failed. Set Clear Status register command to clear the status bit set by previous unsuccessful program operation.
Set Erase command write cycles, while device is in Erase Suspend. First two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/80h. Erase attempt while device is in Erase Suspend mode should fail. Read status register to check Erase Status bit and verify that erase operation has failed. Set Clear Status register command to clear the status bit set by previous unsuccessful erase operation.
Set erase resume write command cycle with XXXh/50h address/data combination. Read status register to check if device is no longer in erase suspend mode.

Set Blank Check command write cycle with (SA)555h/33h address/data combination. Device enters Blank Check mode, to check if the sector SA is completely erased. Read status register to check if

device is busy. Wait for tBC, time needed for device to complete Blank Check. After that, read status register to check if device is ready and if erase was successful.

3.8 SSR ASO

3.8.1 Series 8. Positive Tests

a. SSRASO_8_a

***Testcase symbolic name**:* SSRASO_8_a

***Description**:* Test the SSR ASO

What is being verified:

Entry of SSR ASO and read of SSR ASO

Method of verification:

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode. Read several SSR ASO data and verify data correctness. Read command to the device with different SA than the one used to enter SSR ASO, returns memory array data. Read memory with different SA and verify it is correct memory array data. Set write command cycle for ASO exit to return the device to the default mode (read mode). Read memory with SA used to previously enter SSR ASO and verify it is memory array data.

b. SSRASO_8_b

***Testcase symbolic name**:* SSRASO_8_b

***Description**:* Test Word Program in SSR ASO

What is being verified:

Word Program of SSR ASO

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode. Read several SSR ASO data and verify data correctness.

Set write command cycles to enter word program mode in SSR ASO, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/A0h address/data combination. After this, set write command cycle with program address and data which is programmed on that address. Read status register to check if device is busy. Wait for tSWP, time needed for device to program one word in the SSR ASO. Read status register to check if device is ready. Read SSR ASO from the programmed location and verify that it is programmed data. Set write command cycle to exit SSR ASO.

c. SSRASO_8_c

***Testcase symbolic name**:* SSRASO_8_c

***Description**:* Test Buffer Program (512B) in SSR ASO

What is being verified:

Buffer Program (512B) of SSR ASO

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode.

Set write command cycles to enter buffer program mode in SSR ASO, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with SAh/25h address/data combination. After this, set write command cycle with .Sah/WCh where WC is word count, number of data - 1 programmed into the buffer. Following are write command cycles with addresses and data to be programmed at. After total number of 256 write cycles for data, set write command cycle for buffer confirm with SAh/29h address/data combination. Read status register to check if device is busy. Wait for tWBP, time needed for device to program entire buffer. Read status register to check if device is ready. Read SSR ASO from the programmed location and verify that it is programmed data. Set write command cycle to exit SSR ASO.

d. SSRASO_8_d

***Testcase symbolic name**:* SSRASO_8_d

***Description**:* Test Buffer Program (16B page) in SSR ASO

What is being verified:

Buffer Program (16B page) of SSR ASO

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode.

Set write command cycles to enter buffer program mode in SSR ASO, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with SAh/25h address/data combination. After this, set write command cycle with .Sah/WCh where WC is word count, number of data - 1 programmed into the buffer. Following are write command cycles with addresses and data to be programmed at. After total number of 8 write cycles for data, set write command cycle for buffer confirm with SAh/29h address/data combination. Read status register to check if device is busy. Wait for tWBP, time needed for device to program entire buffer. Read status register to check if device is ready. Read SSR ASO from the programmed location and verify that it is programmed data. Set write command cycle to exit SSR ASO.

e. SSRASO_8_e

***Testcase symbolic name**:* SSRASO_8_e

***Description**:* Test Burst Write Program in SSR ASO

What is being verified:

Burst Write Program operation of SSR ASO

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode.

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set command write cycle with SA and word count to be programmed. Set write command cycle for burst write with first data and starting address. In this command cycle CS# is kept low for entire period of data cycles. Number of data cycles is equal to Word Count +1. Read Status register to check if device is busy. Wait untili device complets Buffer Programing, tWBP time. Read Status register to check if device is not busy and if programming was a success.

Read SSR ASO from the programmed location and verify that it is programmed data. Set write command cycle to exit SSR ASO.

f. SSRASO_8_f

***Testcase symbolic name**:* SSRASO_8_f

***Description**:* Negative test, incorrect Word Program of SSR ASO

What is being verified:

Word Program operation of SSR ASO with wrong SSR address

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode. Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Following is write command cycle for data write with incorrect SA which is not equal to one used to enter SSR ASO. Read status register to check if device is ready and if program status bit indicates program failure. Set write command cycle to exit SSR ASO. Set Clear Status register command to clear status bits set by the incorrect program commands.

g. SSRASO_8_g

***Testcase symbolic name**:* SSRASO_8_g

***Description**:* Negative test, incorrect Word Program of SSR ASO

What is being verified:

Word Program operation of SSR ASO with wrong SA address

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode. Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set write command cycle with Word Count for Buffer Program but incorrect SA, diffrerent than the one used to enter SSR ASO. Read status register to check if device is ready and if program status bit indicates program failure.

Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set Clear status register command to clear all status bits set by previous incorrect Buffer Program commands.

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set write command cycle with Word Count for Buffer Program with correct SA the same as the one used for SSR ASO entry. Set number of write command cycle for write data into the buffer with incorrect SA. Read status register to check if device is ready and if program status bit indicates program failure.

Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set Clear status register command to clear all status bits set by previous incorrect Buffer Program commands.

Set command cycles for Buffer Program command, two unlock cycles (555h/AAh,2AA/55h) followed by a write cycle of SA sector address and 25h data to enter word program state. Set write command cycle with Word Count for Buffer Program with correct SA the same as the one used for SSR ASO entry. Set number of write command cycle for write data into the buffer with correct SA. Set write command cycle for Buffer Confirm command with incorrect SA. Read status register to check if device is ready and if program status bit indicates program failure.

Set Write to Buffer Abort Reset command to exit Program Error State, in which device enters after buffer abort error. Write to Buffer Abort Reset command consists of 3 cycles, first two cycles are unlock cycles (555h/AAh and 2AAh/55h) followed by third write cycle with 555h/F0h address/data combination.

Set Clear status register command to clear all status bits set by previous incorrect Buffer Program commands.

Set write command cycle to exit SSR ASO.

h. SSRASO_8_h

***Testcase symbolic name**:* SSRASO_8_h

***Description**:* Negative test, Programming SSR protected region

What is being verified:

Word Program operation of SSR ASO protected region

***Method of verification*:**

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode.

Set Word Program command cycle to program SSR region that is used for prtoection of other regions. The region that contains protection for the rest of SSR regions, is from address 10h until 2Fh. Programmed location is 13h (19). Read status register to check if device is busy. Wait until Word Programming is done, tSWP time. After that, read Status register to check if the device is ready. Read SSR programmed region and verify that correct data is programmed.

Set write command cycle to program protected SSR region. Read Status register to check if device is ready and if the program status bit indicates program failure.

Set write command cycle to exit SSR ASO.

i. SSRASO_8_i

***Testcase symbolic name**:* SSRASO_8_i

***Description**:* Negative test, Programming SSR region when SSRFREEZE bit is set

What is being verified:

Word Program operation of SSR ASO region when SSRFREEZE bit of xVCR register is set to ⓪©

***Method of verification*:**

Set write command cycles to programm VCR, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for VCR program (555h/38h). Following is a write command cycle with data value to be loaded into the VCR register. Bit [10] of VCR is set, thus SSR is frozen.

Set write command cycles for SSR ASO entry, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for SSR ASO entry with (SA)555h/88h address/data combination. Device is in SSR ASO mode.

Set Word Program command cycle to program SSR region. SSRFREEZE bit is set to ⓪© programming of SSR is not allowed. Read Status register to check if device is ready and if program status bit indicates program failure. Set Clear Status register command to clear the status bits set by the previous invalid program commands.

Set write command cycle to exit SSR ASO.

3.9 ASP register

3.9.1 Series 9. Positive Test

a. ASPREG_9_*a*

Testcase symbolic name: ASPREG_9_a

What is being verified:

Enter and read of ASP register

Method of verification:

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP enter with 555h/40h address/data combination. Read ASP register and verify the value of the register.

Set write command cycle to exit ASP, used ASPR ASO exit command with XXXh/90h address/data combination followed by write command cycle with XXXh/0h address/data combination.

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP enter with 555h/40h address/data combination. Read ASP register and verify the value of the register.

Exit ASP mode by setting reset command, write command cycle with XXXh/F0 address/data combination. Read ASP register and verify the value of the register.

b. ASPREG_9_b

Testcase symbolic name: ASPREG_9_b

What is being verified:

Enter and programming of ASP register

Method of verification:

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP enter with 555h/40h address/data combination. Read ASP register and verify the value of the register.

Read status register while in ASP ASO. Set clear status register write command cycle with 555h/71h address/data combination. Status register is cleared.

Program ASP register by setting write command cycle, first cycle with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be writen in the ASP register. Read status register to check if device is busy, while programming ASP register. Wait for tSWP, time needed for device to program data into the register. Read status register to check if device is ready. Read ASP register to check if the programmed data is written in the ASP register.

Set write command cycle to exit ASP, used ASPR ASO exit command with XXXh/90h address/data combination followed by write command cycle for ASO reset with XXXh/F0h address/data combination.

3.10 RESET

3.10.1 Series 10. Negative Tests

a. RESET_10_*a*

Testcase symbolic name: RESET_10_a

What is being verified:

Hard reset during program operation

Method of verification:

Set program PORTime register write command cycles, two unlock cycles (555h/AAh, 2AAh/55h) followed by PORTime program write cycle with 555h/34h address/data combination. Read status register to check if device is busy during programming of PORTime register. Wait tSPW, time needed for device to complete the programming of PORTime register. Read status register to check if device is ready. Read PORTime register to check if the programmed data is written in the register.

Program NVCR before reset to set new value of VCR after RESET is completed. Set NVCR program write command cycles, two unlock cycles (555h/AAh, 2AAh/55h) followed by NVCR program write command cycle with 555h/39h address/data combination. After that, set write command cycle with XXXh/PDh where PD is program data that is written in the NVCR register. Read status register to check if the device is busy while programming NVCR. Wait tSWP, time needed for the device to complete the programming of NVCR. After that, read status register to check if the device is ready. Read NVCR to check if the programmed data is correctly written in the NVCR register.

Start Word Program by setting the write command cycles, two unlock cycles followed by write cycle with 555h/A0h address/data combination. After that, set write cycle with program address and data to be programmed at that address. Read status register to check if the device is busy.

Set write cycle to reset the device. When RESET# is low all operations are terminated. Wait for 30us until device is reseted.

Read NVCR register to check if the previously programmed data is correct. Read VCR to check if the value is the same as in NVCR.

Repeat the interrupted word program operation. After any interrupted program or erase operation by reset, the same operation is needed to ensure integrity of the data. After programming is complete read the programmed location to check if data is written correctly.

b. RESET_10_b

Testcase symbolic name: RESET_10_b

What is being verified:

Hard reset during erase operations

Method of verification:

Set erase NVCR write command cycles to erase NVCR before reset. By erasing NVCR, the device is actually setting it to default value. First, two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for erase NVCR with 555h/C8h address/data combination. Read status register to check if the device is busy while erasing the NVCR. After tSEO, time needed for device to successfully erase NVCR, read status register to check if the device is ready.

Set command cycles for Sector Erase, first two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with 555h/80h address/data combination. The device is now in Erase state, but it is not yet determined which erase operation is performed, sector erase or chip erase. Set another write command cycles, again two unlock cycles (555h/AAh, 2AA/55h) followed by write command cycle with SAh/30h address/data combination where SA holds the value of sector to be erased. Read status register to check if device is busy.
Set write cycle to reset the device. When RESET# is low all operations are terminated. Wait for 30us until device is reseted.

Initiate Evaluate Erase Status command to check if the previous sector erase command was successful. Read status register to check if the device is busy while in EES state. Wait for tEES, time needed to complete the EES. Read status register to check if the device is ready and if the previous sector erase operation, interrupted with reset, was not successful.

Repeat the sector erase operation to ensure data integrity. After erasure is complete, initiate another Evaluate Erase Status to check if this sector erase opertaion was successful.

3.11 DYB ASO

3.11.1 Series 11. Positive test

a. DYBASO_11_a

Testcase symbolic name: DYBASO_11_a

Description*:* Enter and manage operations in DYB ASO

What is being verified: Entry of DYB ASO, read of DYB ASO

Method of verification:

Set write command cycles for DYB ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for DYB ASO entry. Set read status register command cycles to check status register. Set clear status register command cycle to clear the status register. Exit DYB ASO by setting Reset/ASO exit command cycle (XXXh/F0h).

Enter DYB ASO again with the same command cycles as described. Set read command cycle (SAh/XXh) to read DYB ASO status of the selected sector SA and verify the current status of DYB protection of the sector. Exit DYB ASO by setting Command Set Exit write cycles, first write cycle is XXXh/90h followed by write cycle with XXXh/00h.

b. DYBASO_11_b

Testcase symbolic name: DYBASO_11_b

Description*:* Enter and manage operations in DYB ASO

What is being verified: Entry of DYB ASO, set and clear DYB bits

Method of verification:

Set write command cycles for DYB ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for DYB ASO entry. Set write command cycles for programming DYB ASO, first write cycle with XXXh/A0h followed by write cycle with SAh/00h address/data combination. In this way, bit in DYB register that correspsonds to sector SA is set to 0, thus protecting the sector from program or erasure. Set read command cycle (SAh/XXh) to read DYB ASO status of the selected sector SA. Set write command cycles to clear the DYB bit of the selected sector, first write cycle with XXXh/A0h followed by write cycle with SAh/01h address/data combination. In this way, bit in DYB register that corresponds to sector SA is cleared to 1, thus the sector is unprotected. Set read command cycle (SAh/XXh) to read DYB ASO status of the selected sector SA. Exit DYB ASO with ASO Exit command.

3.12 Password ASO

3.12.1 Series 12. Positive Test

a. PASSWDASO_12_a

Testcase symbolic name: PASSWDASO_12_a

Description*:* Enter and manage operations in Password ASO

What is being verified: Entry of Password ASO, read password

Method of verification:

Set write command cycles for Password ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for Password ASO entry. Set four consecutive read commands, reads with address in 0,1,2,3 order, to read the password. Set ASO Exit command to exit Password ASO.

b. PASSWDASO_12_b

Testcase symbolic name: PASSWDASO_12_b

Description*:* Enter and manage operations in Password ASO

What is being verified: Entry of Password ASO, program password

Method of verification:

Set write command cycles for Password ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for Password ASO entry. Set write command cycle to program first word of the password, first write cycle is with XXXh/A0h address/data combination followed by write command cycle with 00h/PWD where PWD is password data word to be programmed. During password programming, device is busy, set read status regiset command to check ready/busy bit in status register.

Another 3 of these write cycles are needed to program entire 64bit password. After all program commands are executed, set clear status register command.

Set read commands to check the programmed password. There are four consecutive read commands with addresses 0,1,2 and 3 respectively. Exit Password ASO by setting write command cycles, first write cycle with XXXh/90h followed by write cycle with XXXh/00h address data combination.

3.13 PPB ASO

3.13.1 Series 13. Positive and negaitve tests

a. PPBASO_13_a

Testcase symbolic name: PPBASO_13_a

Description*:* Enter and manage operations in PPB ASO

What is being verified: Entry of PPB ASO, read PPBs

Method of verification:

Set write command cycles to program ASP register, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/40h address/data combinatio to enter ASP mode. In ASP mode, set read status register and clear status register commands to check and clear status register. Set write command cycles to program ASP register, first write cycle is with XXXh/A0h address/data combination followed by write command cycle with XXXh/PDh, where PD is program data to be written in the ASP register. While ASP is programmed, device goes busy, set read status register command to check if ready/busy bit is set. Wait until programming is done, then read ASP register to check if the programming of data was successful. Exit ASP mode by setting write command cycles, first write cycle is with XXXh/90h address/data combinatio followed by write cycle with XXXh/00h address/data combination.

Set write command cycles to enter PPB ASO, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C0h address/data combination. Set read command cycle with SA to read the PPB status of the selected sector SA. Exit PPB ASO by setting write command cycle for ASO exit (XXXh/F0h).

b. PPBASO_13_b

Testcase symbolic name: PPBASO_13_b

Description*:* Enter and manage operations in PPB ASO

What is being verified: Entry of PPB ASO, program PPBs

Method of verification:

Set write command cycles to enter PPB ASO, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C0h address/data combination. Set write command cycles to programm PPB bit, first write cycle is with XXXh/A0h address/data combination followed by write cycle with SAh/00h address/data combination, where SA is sector address for which PPB bit is set to 0 and thus protecting the sector from programming or erasure. While programming PPB bit, device goes busy. Set read status register command to check if the ready/busy bit is set. Wait until programming is done then read status register again to check if the device is ready. Set read command cycle (SAh) to read PPB status of the selected sector. Exit PPB ASO by setting write command cycles, first write cycle with XXXh/90h followed by write command cycle with XXXh/00h address/data combination.

c. PPBASO_13_c

Testcase symbolic name: PPBASO_13_c

Description*:* Enter and manage operations in PPB ASO

What is being verified: Entry of PPB ASO, erase PPBs

Method of verification:

Set write command cycles to enter PPB ASO, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C0h address/data combination. Set write command cycles to erase PPB bits, first write cycle is with XXXh/80h address/data combination followed by write cycle with 00h/30h address/data combination. When device enters PPB erase state it goes busy, set read status register command to check if the ready/busy bit is set. When PPB erase command is set, all PPBs are erased, set to 1, thus unprotecting all sectors. Wait until erasure is done, read status register to check if the device is ready. Set read command cycle to read PPB status for any sector in

the memory, they are all unprotected and PPBs should be 1. Exit PPB ASO by setting write command cycle for ASO Exit, write cycle with XXXh/F0h address/data combination.

3.14. Protected sectors

3.14.1 Series 14. Negative tests

a. PROTSEC_14_a

***Testcase symbolic name**:* PROTSEC_14_a

***Description**:* Operations on protected sectors

What is being verified: Impossible program operation on a protected sector

***Method of verification*:**

Set write command cycles to enter DYB ASO, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle 555h/E0h for DYB ASO entry. Set write command cycles for programming DYB ASO, first write cycle with XXXh/A0h followed by write cycle with SAh/00h address/data combination. In this way sector SA is protected by seting corresponding DYB bit. Set read command cycle to read the DYB status of the selected sector. Exit DYB ASO by setting write command cycles, first Command Set exit with XXXh/90h address/data combination followed by write cycle with XXXh/00h address/data combination.

Set word program write cycles to attempt progamming on a protected sector. First set two unlock write command cycle (555h/AAh, 2SSh/55h) followed by write command cycle with 555h/A0h address/data combination. Then set write command cycle with PAh/PD where PA is program addres and PD is program data. Program address is address within protected sector. Since the sector is protected, programming will be aborted. Device is in busy state for 20us to 100us, thus read status register command is set to check if the device is busy. Wait for 100us, then read status register to check if the device is now ready. When programming is attempted on a protected sector, status register bit 1 is set. Set read status register command cycles to check Sector Lock status bit. Also, status register bit 4 indicates if the programming has failed, set the read status register command cycles to verify the failure of previous programming operation. Set clear status register write command to clear the status bits set in this test.

b. PROTSEC_14_b

***Testcase symbolic name**:* PROTSEC_14_b

***Description**:* Operations on protected sectors

What is being verified: Impossible erase operation on a protected sector

***Method of verification*:**

Set buffer programming operation to program some data into the empty sector. First set two unlock write cycles (555h/AAh, 2AAh/55h) followed by write command cycle with SAh/25h address data combination. Then, set write command cycle with SAh/WCh where SA must be the same as the SA used in previous write command, and WC is word count -1 that is going to be programmed into the memory array. After this, set exact number of write command cycles, WC number cycles, to programm data into the memory. Read status register command cycles are set to check if device is busy. Wait until buffer programm is done, read status registe again to check if the device is ready and if the program operation was successful.

Set write command cycles for programming DYB ASO, first write cycle with XXXh/A0h followed by write cycle with SAh/00h address/data combination. In this way sector SA is protected by seting

corresponding DYB bit. Set read command cycle to read the DYB status of the selected sector. Exit DYB ASO by setting write command cycles, first Command Set exit with XXXh/90h address/data combination followed by write cycle with XXXh/00h address/data combination.

Set write command cycle to enter sector erase state, first 2 unlock write command cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/80h address/data combination. Device is in Erase mode, next is command for Sector Erase operation. Set write command cycles, first 2 unlock write command cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/30h address/data combination. Device is in busy state for 20us to 100us, thus read status register command is set to check if the device is busy. Wait for 100us, then read status register to check if the device is now ready. Because sector was protected, status register bit 1 is set, and since erase was not successful status register bit 5 is also set. Set read command cycles to check these bits, if they are set. Set clear status register write command cycle, to clear the bits which are set by the device.

Set write command cycles to initiate Blank Check of the sector for which erase was attempted. Blank check command will indicate that sector was not erased and status register bit 5 will be set. While device is in Blank Check mode, busy bit of status register is set. Read status register to check if the device is busy. Wait until Blank Check is done. Read status register to check if the device is ready. Read status register to check if the Erase status bit is set by Blank Check operation and verify that it is set. Clear status register command is set, to clear the status register bits.

3.15 Chip Erase

3.15.1 Series 15. Positive Test

a. CHIPERS_15_a

Testcase symbolic name: CHIPERS_15_a

Description: Test Chip Erase operation

What is being verified: Erasure of entire memory

Method of verification:

Set write command cycles for entering Chip Erase mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/80h address/data combination. Device is now in Erase Mode, but it is not yet determined if it is going to be sector or chip erase. By setting another two unlock cycles (555h/AAh, 2AAh/55h) and followed by write cycle of 555h/10h device will enter Chip Erase mode and start to erase entire memory.

Read status register to check if device is busy. Wait for tCE, time needed for device to erase entire array. Read status register to check if device is ready.

3.16 PPB Lock bit

3.16.1 Series 16. Positive and negative tests

a. PPBLB_16_a

Testcase symbolic name: PPBLB_16_a

Description: PPB Lock bit mode

What is being verified: PPB Lock bit entry mode, various exit commands

***Method of verification*:**

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status. Exit PPB Lock bit mode by setting Reset/ASO exit command cycle with XXXh/F0h address/data combination. Enter PPB Lock bit mode again. Read Lock bit status, then exit PPB Lock bit mode by setting Command Set Exit cycles, first write cycle with XXXh/90h followed by write cycle with XXXh/00h address/data combination.

b. PPBLB_16_b

***Testcase symbolic name**:* PPBLB_16_b

***Description**:* PPB Lock bit mode

What is being verified: PPB Lock bit entry mode, various invalid write commands

***Method of verification*:**

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status.

Set invalid write command cycle that will cause no transition from PPB Lock bit mode.

Set write command cycle to enter PPBLBEXT mode, write command cycle with XXXh/90h address/data combination. Device is in PPBLBEXT mode, next write command should be with XXXh/F0h address/data combination. Instead, write command cycle set is invalid and device returns to PPBLB state.

Set invalid PPB Lock bit clear write command cycles, first write command cycle is with XXXh/A0h followed by write command cycle which does not have 00h as data, thus the PPB Lock bit is not cleared. Set read command cycle to check the status of PPB Lock bit.

Exit PPB Lock bit mode by setting Command Set Exit write cycles.

c. PPBLB_16_c

***Testcase symbolic name**:* PPBLB_16_c

***Description**:* PPB Lock bit mode

What is being verified: PPB Lock bit entry mode, clear PPB Lock bit

***Method of verification*:**

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status.

Set write command cycles to clear PPB Lock bit, first write cycle is with XXXh/A0h address/data combination followed by write cycle with XXXh/00h. Set read command cycle to read PPB Lock bit status and verify if the PPB Lock bit is cleared.

Set read status register command while in PPB Lock bit mode. Clear status register command cycle to clear the status register while in PPB Lock bit mode.

Exit PPB Lock bit mode by setting Command Set Exit write cycles.

3.17 Force DYB bits after HW Reset

3.17.1 Series 17. Positive test

a. DYBRST_17_a

***Testcase symbolic name**:* DYBRST_17_a

***Description**:* DYB bits after Hard reset

What is being verified: Hard reset forces DYB bits to 0, unprotected state

Method of verification:

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP entry with 555h/40h address/data combination. Read ASP register and verify the value of the register. Program ASP register by setting write command cycle, first cycle with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be writen in the ASP register. Program data is set in a way that sets ASP register bit 4 to 0. ASP register bit 4 when set to 0 will cause all DYB bits to be set to 0 after hard reset. Read status register to check if device is busy during ASP programming. Wait until programming is done, read status register to check if device is ready. Read ASP register and verify the value of the register. Exit ASP register mode by setting Command Set Exit write cycles.

Set write command cycle to reset the device. Wait until device is reseted. Check if all DYB bits are set to 0 in DYB register.

3.18 xVCR Program negative, FREEZE set

3.18.1 Series 18. Negative tests

a. VCR_FREEZE_18_a

***Testcase symbolic name**:* VCRFREEZE_18_a

***Description**:* Load VCR fail, xVCR FREEZE bit set

What is being verified: Loading of VCR failes due to xVCR FREEZE bit set

Method of verification:

Set Load VCR command cycles to set xVCR FREEZE bit, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/38h. After this, set write command cycle with xxxh/PDh where PD is program data. Set command cycles to read VCR, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C7h. After this, set read command cycle to read and verify programmed VCR. Now, xVCR FREEZE bit is set.

Set Load VCR command cycles to attempt a load of VCR. Because xVCR FREEZE is set, this program command should fail. Set command cycles to read VCR and verify that program was not successful, old value of VCR should be read.

Set Hard Reset command, which forces NVCR to be loaded in the configuration register (VCR). After reset is done, xVCR FREEZE bit is not set any more. Set read command cycles to check the status of VCR register after reset has occurred and verify that the value is equal to NVCR.

b. VCR_FREEZE_18_b

***Testcase symbolic name**:* VCRFREEZE_18_b

***Description**:* Program and Erase of NVCR fail, xVCR FREEZE bit set

What is being verified: Program and Erase operations on VCR faile due to xVCR FREEZE bit set.

Method of verification:

Set Load VCR command cycles to set xVCR FREEZE bit, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/38h. After this, set write command cycle with xxxh/PDh where PD is program data. Set command cycles to read VCR, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C7h. After this, set read command cycle to read and verify programmed VCR. Now, xVCR FREEZE bit is set.

Set Program NVCR command cycles to attempt a program operation of NVCR. Because xVCR FREEZE is set, this program command should fail. Read status register to check if device is not busy. Set command cycles to read NVCR and verify that program was not successful, old value of NVCR should be read.

Set Erase NVCR command cycles to attempt an erase operation of NVCR. Because xVCR FREEZE is set, this erase command should fail. Read status register to check if device is not busy. Set command cycles to read NVCR and verify that erase was not successful, old value of NVCR should be read.

Set Hard Reset command, which forces NVCR to be loaded in the configuration register (VCR). After reset is done, xVCR FREEZE bit is not set any more. Set read command cycles to check the status of VCR register after reset has occurred and verify that the value is equal to NVCR.

c. VCR_FREEZE_18_c

***Testcase symbolic name**:* VCRFREEZE_18_c

***Description**:* Program of NVCR and VCR fail, xVCR FREEZE bit set

What is being verified: Program and Erase operations on NVCR and VCR faile due to xVCR FREEZE bit set.

Method of verification:

Set Program command cycles for NVCR to set xVCR FREEZE bit, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/39h. After this, set write command cycle with xxxh/PDh where PD is progam data for NVCR. Read Status register and verify that deivce is busy, program is in progress. Wait for tSWP, until programming is done. Read Status register to check if device is now ready, programming is completed. Now, xVCR FREEZE bit of NVCR is set.

Set command cycles to read NVCR, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/C6h. Aftert this, set read command cycle to read NVCR data and verify that the value is the same as programmed data. Bit xVCR FREEZE of NVCR will be effective only after hard reset, currently programming of VCR and NVCR is still allowed.

Set Load VCR command cycles to attempt a load of VCR. first two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle with 555h/38h. After this, set write command cycle with xxxh/PDh where PD is progam data for VCR. Set command cycles to read VCR and verify that data is programmed.

Set hard reset command, this forces NVCR to be loaded in VCR register. After reset is done, VCR will have the same value as NVCR. NVCR is programmed with xVCR FREEZE bit set to ©any program or erase operations on VCR or NVCR are not allowed.

Set command cycles to read VCR, verify that the value is the same as NVCR. Set command cycles to read NVCR, verify that the value is correct.

Set Erase NVCR command cycles to attempt an erase operation of NVCR. Because xVCR FREEZE is set, this erase command should fail. Read status register to check if device is not busy. Set command cycles to read NVCR and verify that erase was not successful, old value of NVCR should be read.

Set Program NVCR command cycles to attempt a program operation of NVCR. Because xVCR FREEZE is set, this program command should fail. Read status register to check if device is not busy. Set command cycles to read NVCR and verify that program was not successful, old value of NVCR should be read.

Set Load VCR command cycles to attempt a load of VCR. Because xVCR FREEZE is set, this program command should fail. Set command cycles to read VCR and verify that program was not successful, old value of VCR should be read.

3.19 Measure Temperature

3.19.1 Series 19. Positive test

a. MEASURETEMP_19_a

Testcase symbolic name: MEASURETEMP_19_a

Description: The temperature measurement is initiated by writing the three cycle Measure Temperature command sequence. The device then transitions to the BUSY state. The host can poll the status register until the device transitions back to the IDLE state.

To read the results of the measurement the host performs the four step Read Temperature Register command sequences.

What is being verified: Measure Temperature operation and Temperature register read

***Method of verification*:**

In the Idle state activate Measure Temperature operation by writing the three cycle command sequences. Verify the device busy state by polling the Status Register. After period of time specified for the operation verify the device ready state.

Initiate the four cycle Temperature Register read operation.

3.20 Deep Power Down

3.20.1 Series 20. Positive test

a. DPD_20_a

Testcase symbolic name: DPD_20_a

Description: Deep Power Down mode is entered using the DPD Entry command sequences. The device will take Tdpd time to return to the IDLE state. During the Tdpd period the device will ignore command sequences.

What is being verified: DPD Entry command

***Method of verification*:**

Write DPD Entry command sequences to enter DPD mode. Wait for Tdpd wake up time to return to IDLE state. To verify that the device is in ready state initiate the Status Register read operation.

3.21 Sector erase

3.21.1 Series 21. Negative test

a. SEC_ERASE_21_a

Testcase symbolic name: SEC_ERASE_21_a

Description: Erase Protected Sector

What is being verified: Verify that protected sector is not erased

***Method of verification*:**

Read from Sector that will be erased. DYB Set command cycles, first XXXh/A0h followed by SA/0h, device is in DYBEXT state, set any write command with SA=XXX and Addr=0h. Read status register. Sector erase command cycles. Verify that operation is not successful(Program Status Bit = 1).

b. SEC_ERASE_21_b

Testcase symbolic name: Positive, Sector Erase

Description: Erase Sector

What is being verified: Verify that not protected sector is not erased

***Method of verification*:**

Read from Sector that will be erased. DYB Set command cycles, first XXXh/A0h followed by SA/0h, device is in DYBEXT state, set any write command with SA=XXX and Addr=0h, protect sector 0. Read status register. Sector erase command cycles, erase sector 18. Verify that operation is not successful(Program Status Bit = 1).

3.22 Persistent Protection Mode

3.22.1 Series 22. Positive test

a. PPMODE_22_a

Testcase symbolic name: PPMODE_22_a

Description: Persistent Protection mode

What is being verified: Persistent Protection Mode, PPB Lock bit entry mode and clear PPB Lock bit, hard reset forces PPB Lock bit to 1

***Method of verification*:**

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP entry with 555h/40h address/data combination. Read ASP register and verify the value of the register. Program ASP register by setting write command cycle, first cycle with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be writen in the ASP register. Program data is set in a way that sets ASP register bits [2:1] to 2'b10, thus setting the Persistent Protection mode. Read status register to check if device is busy during ASP programming. Wait until programming is done, read status register to check if device is ready. Read ASP register and verify the value of the register. Exit ASP register mode by setting Command Set Exit write cycles.

Set write command cycle to reset the device. Wait until device is reseted. After hard reset, device forces PPB Lock bit to 1 when device is in Persistent Protection Mode.

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status. Exit PPB Lock bit mode by setting Command Set Exit write cycles.

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status. Set write command cycles to clear PPB Lock bit, first write cycle is with XXXh/A0h address/data combination followed by write cycle with XXXh/00h. Set read command cycle to read PPB Lock bit status and verify if the PPB Lock bit is cleared. Exit PPB Lock bit mode by setting Command Set Exit write cycles.

Set write command cycle to reset the device. Wait until device is reseted. After hard reset, device forces PPB Lock bit to 1 when device is in Persistent Protection Mode.

Set write command cycles to enter PPB Lock bit mode, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle with 555h/50h address/data combination to enter PPB Lock bit mode. Set read command cycle to read PPB Lock bit status. Exit PPB Lock bit mode by setting Command Set Exit write cycles.

ASP[11] is set to 0 to enable wrapped burst read in hybrid burst mode. Initiate read operations with different burst length to verify the hybrid burst read sequencing.

3.23 Password Mode

3.23.1 Series 23. Positive and negative tests

a. PWDMODE_23_a

***Testcase symbolic name**:* PWDMODE_23_a

***Description**:* Password Mode entry

What is being verified: Password mode entry, setting and clearing of PPB Lock bit

***Method of verification*:**

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP entry with 555h/40h address/data combination. Read ASP register and verify the value of the register. Program ASP register by setting write command cycle, first cycle

with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be written in the ASP register. Program data is set in a way that sets ASP register bits [2:1] to 2@01, thus setting the Password Protection mode. Read status register to check if device is busy during ASP programming. Wait until programming is done, read status register to check if device is ready. Read ASP register and verify the value of the register. Exit ASP register mode by setting Command Set Exit write cycles.

Set write command cycle to reset the device. Wait until device is reseted. After hard reset, device forces PPB Lock bit to 0 when device is in Password Protection Mode.

Set write command cycles for Password ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for Password ASO entry.

Set various invalid write commands to unlock the password. Valid combination for password unlock is: first write cycle ± 00h/25h, second write cycle ± 00h/03h, third write cycle 00h/PWD0 whre PWD0 is lowest 16 bits of password, fourth write cycle ± 01h/PWD1 where PWD1 is second 16bit part of the password, fifth write cycle ± 02h/PWD2 where PWD2 is third 16bit part of the 64bit password, sixth write cycle ± 03h/PWD3 where PWD3 is the last 16bit word of 64 bit password. If any of these write cycles are not correct, device returns to PP mode and the password remains unlocked.

Set correct write command cycles for password unlock, as described. Wait until password unlock time expires.

Set read command cycles for read password and verify that all 1s are on the output, because ASP register bit 2 is set to 0 and read of password returns all 1s.

Exit Password ASO by setting Command Set Exit write cycles.

b. PWDMODE_23_b

Testcase symbolic name*:* PWDMODE_23_b

Description*:* Wrong Password entered

What is being verified: Negative test, wrong unlock sequence for password unlock

***Method of verification*:**

Set write command cycles for Password ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for Password ASO entry. Set write command cycles for password unlock, with password words different than programmed. Read status register to check if device is busy, unlock sequence take around 100us. Wait for 100us then read status register to check if device is ready. Read status register to check program status bit, should be failed. Read status register to check sector lock status and verify that the value is @©Set clear status register to clear all status bits set by the previous command sequences.

Exit the password mode.

c. PWDMODE_23_c

Testcase symbolic name*:* PWDMODE_23_c

Description*:* Password Mode entry

What is being verified: Negative test, programming ASP register for Persistent Protection mode while device is in Password Protection mode

***Method of verification*:**

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP entry with 555h/40h address/data combination. Program ASP register by setting write command cycle, first cycle with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be writen in the ASP register. Program data is set in a way that bit 2 of ASP register is to be set to 0. When device is in Password Protection Mode, programming ASP register bit 2 to 0 is not allowed, program operation is aborted. Read status register to check if device is ready, considering that programming is aborted. Read status register to check program status bit, and verify that it indicates program failure. Exit ASP register mode by setting Command Set Exit write cycles.

d. PWDMODE_23_d

***Testcase symbolic name**:* PWDMODE_23_d

***Description**:* Negative, Program both Persistent and Password mode at the same time

What is being verified: Negative test, programming ASP register for Persistent Protection mode and for Password Protection mode will cause a failure

***Method of verification*:**

Set write command cycle for entering ASP mode, two unlock cycles (555h/AAh, 2AAh/55h) followed by write command cycle for ASP entry with 555h/40h address/data combination. Program ASP register by setting write command cycle, first cycle with XXXh/A0h address/data combination followed by write cycle with XXXh/PDh where PD is program data to be writen in the ASP register. Program data is set in a way that both ASP register bits for Password and Persistent Mode are set, ASPR[2:1] = 2b11. Read status register to check if device is ready, considering that programming is aborted. Read status register to check program status bit, and verify that it indicates program failure. Exit ASP register mode by setting Command Set Exit write cycles.

e. PWDMODE_23_e

Testcase symbolic name. PWDMODE_23_e

Description. Negative, Program ASPR register bits [2:1] with 2b11

What is being verified: Negative test, program ASPR register bits [2:1] when Password mode is selected, programming of the password is not allowed

***Method of verification*:** Enter Password ASO. For every Password word (2Bytes) one write command with Data=A0h is needed. It is required to program Password in sequential order PWD0, PWD1, PWD2, PWD3. If device is not busy, password program did not start. Exit Password ASO

f. PWDMODE_23_f

***Testcase symbolic name**:* PWDMODE_23_f

***Description**:* Negative, Password unlock when ASPR[2] is 0

What is being verified: Negative test, program password sequence when Password mode is selected, programming of the password is not allowed

***Method of verification*:**

Set write command cycles for Password ASO entry, first two unlock cycles (555h/AAh, 2AAh/55h) followed by write cycle for Password ASO entry. Set Password program command sequences, for each Password word (16 bit) initiate single program command. Verify for every password program command that device is not busy. ASPR bit [2] is set to ©, programming of the password is not allowed.

Read password to check if the programmed data is not written.

Exit the Password ASO mode.

3.24 Autoboot

3.24.1 Series 24. Positive tests

a. AUTOBOOT_24_a

***Testcase symbolic name**:* AUTOBOOT_24_a

***Description**:* Enter HyperFlash mode after reset

What is being verified: After reset enter HyperFlash mode

***Method of verification*:**

After reset AUTOBOOT activate bit in volatile register CR2 and then enter HyperFlash.

b. AUTOBOOT_24_b

***Testcase symbolic name**:* AUTOBOOT_24_b

***Description**:* Autoboot test program registers

What is being verified: AutoBoot program registers

***Method of verification*:**

Program AutoBoot register by setting write command cycle, first cycle with XXXh/A0h address/data combination . Wait for tSWP which is time needed for single word program to execute. After tSWP, read status register to check if device is ready. Then reset device.

c. AUTOBOOT_24_c

***Testcase symbolic name**:* AUTOBOOT_24_c

***Description**:* Load VCR1 to enter HyperFlash after reset

What is being verified: Load VCR1 to enter HyperFlash after reset

***Method of verification*:**

Load NVCR1 before reset. Verify that operation is in progress (Device Ready Bit = 0). Verify that operation is completed (Device Ready Bit = 1). Then read NVCR1.

d. AUTOBOOT_24_d

***Testcase symbolic name**:* AUTOBOOT_24_d

***Description**:* Enter Autoboot ASO after reset

What is being verified: Checking that device enter in ASO after reset

***Method of verification*:**

After reset command enter Autoboot ASO.

3.25 SFDP Test

3.25.1 Series 25. Positive test

a. SFDP_25_a

***Testcase symbolic name**:* SFDP_25_a

***Description**:* SFI test

What is being verified: Enter to SFDP

***Method of verification*:**

Enter ID SFDP ASO without unlock cycles then exit ID SFDP ASO with reset command.

3.26 Write Enable/Write Disable

3.26.1 Series 26. Positive Test

a. _WREN_26_a

***Testcase symbolic name**:* WREN_26_a

***Description**:* Test writes to Write Enable Latch (WEL) bit.

What is being verified: Write Enable instruction.

***Method of verification*:**Drive CS# signal low, initiate WREN instruction and then drive CS# signal high. Initiate read status register instruction to verify that WREN instruction sets the WEL bit.

b. WRDI_26_b

***Testcase symbolic name**:* WRDI_26_b

***Description**:* Test writes to Write Enable Latch (WEL) bit.

What is being verified: Write Disable instruction.

***Method of verification*:**Drive CS# signal low, initiate WRDI instruction and then drive CS# signal high. Initiate read status register instruction to verify that WRDI instruction resets the WEL bit.

3.27 Initial configuration (TBPARM)

3.27.1 Series 27. Positive Test

a. TBPARM_27_a

***Testcase symbolic name**:* TBPARM_27_a

***Description**:* TBPARM bit positive test

What is being verified: The desired state of TBPARM must be selected during the initial configuration of the device during system manufacture; before the first program or erase operation on

the main Flash array. TBPARM must not be programmed after programming or erasing is done in the main Flash array.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate WRR command with sixteen input cycles to change Configuration Register TBPARM bit. Initiate RDSR1 instruction to verify that write cycle is in progress (WIP=1). After t_w verify that write cycle is done by checking Status Register WIP bit(WIP=0).

After write cycle is over, initiate RDCR command to verify that TBPARM has correct value.

b. TBPARM_27_b

***Testcase symbolic name**:* TBPARM_27_b

***Description**:* TBPARM bit negative test

What is being verified: If TBPARM is already programmed to 1, an attempt to change it back to zero will fail and set the Program Error bit (P_ERR in SR1[6]).

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate WRR command with sixteen input cycles to change Configuration Register TBPARM bit. Initiate RDSR1 instruction to verify that operation is not accepted (WIP=0). Verify that P_ERR bit is set. Initiate read RDCR instruction to verify that TBPARM bit did not change its value (TBPARM=1).

3.28 READ Test

3.28.1 Series 28. Positive test

a. READ_SR1_28_a

***Testcase symbolic name**:* READ_SR1_28_a

***Description**:* Test read Status Register1.

What is being verified: The Read Status Register1 (RDSR1) instruction opcode allows the Status Register1 contents to be read out of the SO serial output pin. It is possible to read the Status Register1 continuously by providing multiples of eight clock cycles. CS# signal can be driven high after any bit of the data-out sequence is being shifted out to terminate the transaction.

***Method of verification*:** Initiate Read Status Register Read (RDSR1) instruction. Verify that correct data has been read. Initiate Read Status Register Read (RDSR1) with clock_num aux_parameter (bus_data_read cycle drives CS# low for number of clock cycles that isn't multiple of eight.). Verify that correct data has been read.

3.29 *Read Registers*

3.29.1 Series 29.

a. READ_SR1_29_a

***Testcase symbolic name**:* READ_SR1_29_a

***Description**:* Test read Status Register1.

What is being verified: The Read Status Register1 (RDSR1) instruction opcode allows the Status Register1 contents to be read out of the SO serial output pin. It is possible to read the Status Register1 continuously by providing multiples of eight clock cycles. CS# signal can be driven high after any bit of the data-out sequence is being shifted out to terminate the transaction.

***Method of verification*:** Initiate Read Status Register Read (RDSR1) instruction. Verify that correct data has been read. Initiate Read Status Register Read (RDSR1) with clock_num aux_parameter (bus_data_read cycle drives CS# low for number of clock cycles that isn©multiple of eight.). Verify that correct data has been read.

b. READ_SR2_29_b

***Testcase symbolic name**:* READ_SR2_29_b

***Description**:* Test read Status Register2.

What is being verified: The Read Status Register2 (RDSR2) instruction opcode allows the Status Register2 contents to be read out of the SO serial output pin. It is possible to read the Status Register2 continuously by providing multiples of eight clock cycles. CS# signal can be driven high after any bit of the data-out sequence is being shifted out to terminate the transaction.

***Method of verification*:** Initiate Read Status Register Read(RDSR2) instruction. Verify that correct data has been read. Initiate Read Status Register Read (RDSR2) with clock_num aux_parameter (bus_data_read cycle drives CS# low for number of clock cycles that isn©multiple of eight.). Verify that correct data has been read.

c. READ_PPBLOCK_29_c

***Testcase symbolic name**:* READ_PPBLOCK_29_c

***Description**:* Test read PPB Lock Register.

What is being verified: The PPB Lock Register (PLBRD) instruction opcode allows the PPB Lock Register contents to be read out of the SO serial output pin. It is possible to read the PPB Lock Register continuously by providing multiples of eight clock cycles. CS# signal can be driven high after any bit of the data-out sequence is being shifted out to terminate the transaction.

***Method of verification*:** Initiate PPB Lock Register Read (PLBRD) instruction. Verify that correct data has been read. Initiate PPB Lock Register Read (PLBRD) with clock_num aux_parameter (bus_data_read cycle drives CS# low for number of clock cycles that isn©multiple of eight.). Verify that correct data has been read.

d. READ_ACCREG_29_d

***Testcase symbolic name**:* READ_ACCREG_29_d

***Description**:* Test read PPB and DYB Access Registers.

What is being verified: The PPB Access Register (PPBRD) instruction opcode allows the PPB Access Register contents to be read out of the SO serial output pin. The device is first selected by driving the CS# chip select input pin to the logic low state. Then the instruction code is latched into the SI pin during the rising edge of the serial clock SCK signal. Then the 31-Bit address is shifted in. After address, the 8-bit PPB access register contents is shifted out on the serial output SO pin. Each bit is shifted out at the SCK frequency during the falling edge of the serial clock signal. It is possible to read the PPB Access Register continuously by providing multiples of eight clock cycles. CS# signal

can be driven high after any bit of the data-out sequence is being shifted out to terminate the transaction.

The DYB Access Register (DYBRD) instruction opcode allows the DYB Access Register contents to be read out of the SO serial output pin. Procedure for reading this register is the same as for PPB Access Register.

***Method of verification*:** Initiate PPB Access Register Read (PPBRD) instruction. Verify that correct data has been read. Initiate DYB Access Register Read (DYBRD) instruction. Verify that correct data has been read.

e. READ_ECCREG_29_e

***Testcase symbolic name**:* READ_ECCREG_29_e

***Description**:* Test read ECC Register.

What is being verified: To read the ECC Status Register, the command is followed by the ECC unit address, the four least significant bits (LSB) of address must be set to zero. This is followed by the number of dummy cycles selected by the read latency value in CR2V[3:0]. Then the 8-bit contents of the ECC Register, for the ECC unit selected, are shifted out on SO 16 times, once for each byte in the ECC Unit. If CS# remains low the next ECC unit status is sent thru SO 16 times, once for each byte in the ECC Unit. The maximum operating clock frequency for the ECC READ command is 133 MHz.

***Method of verification*:** Initiate ECC Register Read ECCRD instruction. Verify that correct data has been read. Initiate ECC Register Read ECCRD4 instruction. Verify that correct data has been read.

3.30 OTP READ

3.30.1 Series 30. Positive test

a. OTPREAD_30_a

***Testcase symbolic name**:* OTPREAD_30_a

***Description**:* OTP READ.

What is being verified: OTP READ positive test.

***Method of verification*:**

Reading OTP from addres 20h and 1FCh.

3.31SWL

3.31.1 Series 31. Positive test

a. SVL_31_a

***Testcase symbolic name**:* SVL_31_a

***Description**:* SVL command test

What is being verified: Verify that Wrap Length is set and WEL bit is not changed after setting Wrap Length.

***Method of verification*:**

Set Wrap Length to 16. Verify that Wrap Length is set to 16 (CR4V(1:0)). If the user set WEL bit before SWL command is issued, WEL bit will remainhigh after the command. Set Wrap Length to 32. Verify that Wrap Length is set to 32 (CR4V(1:0)). Verify that WEL bit is still "1".

3.32 *OTP Program ± Positive Tests*

3.32.1 Series 32. Positive Tests

a. OTPPROG_32_a

***Testcase symbolic name**:* OTPPROG_32_a

***Description**:* Write to OTP region - Single Byte Programming.

What is being verified: The OTP Program command programs data in the OTP region, which is in a different address space from the main array data. The protocol of the OTP Program command is the same as the Page Program command. Each region in the OTP memory space can be programmed one or more times, provided that the region is not locked. Subsequent OTP programming can be performed only on the unprogrammed bits.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate OTP_PP instruction to program a single byte to OTP region. Initiate read status register instruction to verify that OTP_PP cycle is in progress. After t_{PP} verify that programming is done by reading status register.

After programming is over verify that correct data has been programmed. Lock programmed OTP region.

b. OTPPROG_32_b

***Testcase symbolic name**:* OTPPROG_32_b

***Description**:* Write to OTP region - OTP Page Programming.

What is being verified: The OTP Program command programs data in the OTP region, which is in a different address space from the main array data. The protocol of the OTP Program command is the same as the Page Program command. Each region in the OTP space can be programmed one or more times, provided that the region is not locked. Subsequent OTP programming can be performed only on the unprogrammed bits.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate OTP_PP instruction to program a 200 bytes to OTP region. Initiate read status register instruction to verify that OTP_PP cycle is in progress. After t_{PP} verify that programming is done by reading status register.

After programming is over verify that correct data has been programmed.

3.33 *Page Programming (4 bytes address)*

3.33.1 Series 33. *Positive Tests*

a. PGPROG_33_a

***Testcase symbolic name**:* PGPROG_33_a

***Description**:* Write to memory array ± 4 bytes address, Page Size = 512

What is being verified: The Page Program instruction opcode allows bytes to be programmed in the memory. Before the Page Program instruction opcode can be accepted by the device internal state machine, a WREN instruction must be issued to set WEL bit. The Page Program instruction is entered by driving the CS# chip select input pin to the logic low state. The instruction opcode is followed by 4-byte address and at least one data byte on the SI serial input pin. Up to 512 data bytes can be placed into the SI serial input pin after the 4-bytes address. If the 8 least significant address bits are not all zero, all transmitted data goes beyond the end of the current page are programmed from the start address of the same page. The CS# chip select pin must be driven to the logic low state during the entire duration of the sequence. If more than 512 bytes are sent to the device, previously latched data are discarded and the last 512 data bytes are guaranteed to be programmed correctly within the same page.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate PP instruction to program single byte. Initiate read status register instruction to verify that PP cycle is in progress. After tPP verify that programming is done by reading status register.

After programming is over verify that correct data has been programmed.

Initiate WREN instruction to set WEL bit. Then, initiate PP instruction to program 3 bytes. Initiate read status register instruction to verify that PP cycle is in progress. After tPP verify that programming is done by reading status register.

After programming is over verify that correct data has been programmed.

Initiate WREN instruction to set WEL bit. Then, initiate PP instruction to program more then 512 bytes. Initiate read status register instruction to verify that PP cycle is in progress. After tPP verify that programming is done by reading status register.

After programming is over verify that correct data has been programmed.

3.34 *Program Suspend*

3.34.1 Series 34. Positive Test

a. PGSUSP_34_a

***Testcase symbolic name**:* PGSUSP_34_a

***Description**:* Test Program suspend operation

What is being verified: Program suspend instruction.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate PP4 instruction to program to flash memory. Initiate read status register instruction to verify that PP4 cycle is in progress. Initiate ESP instruction, and verify that after tEPS command has been accepted by reading status registers.

For all read commands initiate read from the top of memory array. Verify that when the highest address is reached, the address counter will wrap around and roll back to 000000h, allowing the read sequence to be continued indefinitely.

For all read commands initiate read from program suspended sector. Verify that output is undefined. Initiate command that is not allowed during Program Suspend. Verify that command is ignored. Initiate PGRS command and verify that programing again in-progress by reading status register.

After programming is over verify that correct data has been programmed.

3.35 *PPB Bit Programming*

3.35.1 Series 35. Positive Test

a. PGPPB_35_a

Testcase symbolic name: PGPPB_35_a

Description: Test PPB Bit program .

What is being verified: PPB bits can be changed if PPB LOCK = 0.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate PPB Bit program (PPBP) command. Verify that program operation failed by reading status register (WIP = 0, P_ERR=1).

3.36 *PPB Erase*

3.36.1 Series 36. PPB Bits Tests

a. PPBERS_36_a

Testcase symbolic name: PPBERS_36_a

Description: Test PPB Bit erase command.

What is being verified: PPB bits can be changed if PPB LOCK = 0.

***Method of verification*:** First, initiate WREN instruction to set WEL bit. Then, initiate PPB Lock register write (PLBWR) command to clear PPB LOCK. Initiate read status register instruction to verify that program cycle is in progress. After t_{PP} verify that programming is done by reading status register. Initiate read PPB Lock register instruction to verify that correct data has been programmed. Initiate WREN instruction to set WEL bit. Then, initiate PPB Erase (PPBE) command. Initiate read status register instruction to verify that erase cycle is in progress. After t_{PPBE} verify that erasure is done by reading status register. Verify that erase operation failed by reading status register (WIP = 0, E_ERR=1). Initiate Hardware Reset to set value of PPB LOCK bit to 1. After t_{RPH} initiate read PPB Lock register instruction to confirm that PPB Lock bit is set.

3.37 *DYB Bit Programming*

3.37.1 Series 37. Positive test

a. DYBPG_37_a

Testcase symbolic name: DYBPG_37_a

Description: Test DYB Bit program.

What is being verified: DYB Access register write (DYBWR) instruction.

***Method of verification*:** Initiate WREN instruction to set WEL bit. Then, initiate DYB Bit program (DYBWR) command. Initiate read status register instruction to verify that program cycle is in

progress. After t_{PP} verify that programming is done by reading status register. Initiate read DYB Bit instruction to verify that correct data has been programmed.

3.38 *Sector Erase*

3.38.1 Series 38. Positive tests

a. SECERS_38_a

***Testcase symbolic name**:* SECERS_38_a

***Description**:* Sector Erase, positive test.

What is being verified: Verificate Sector erase (256KB)

***Method of verification*:**

Verify that sector is erased. Read Status Register 1, then Read Status Register 2. Verify that Fast Read command is not allowed during Sector Erase. Verify that Parameter Sectors from selected 256KB region are not erased.

b. SECERS_38_b

***Testcase symbolic name**:* SECERS_38_b

***Description**:* Sector Erase, positive test.

What is being verified: Sector Erase applied to 256KB range that include 4KB sectors

***Method of verification*:**

Sector Erase command is applied to a 256KB range that include 4KB sectors. Verify that Parameter sectors from selected 256KB region are not erased, then verify that sector 255 is erased.

c. SECERS_38_c

***Testcase symbolic name**:* SECERS_38_c

***Description**:* Sector Erase, positive test.

What is being verified: Parameter 4 KB-sector Erase: positive test

***Method of verification*:**

Sector Erase command is applied to a 256KB range that include 4KB sectors. Read Status Register 1, then Read Status Register 2. Verify that Fast Read command is not allowed during Sector Erase. Verify that Parameter Sectors from selected 256KB region are not erased.

3.39 *Program During Erase Suspend*

3.39.1 Series 39.

a. ERSSUSPPG_39_a

***Testcase symbolic name**:* ERSSUSPPG_39_a

***Description**:* Test Program during Erase suspend operation .

What is being verified: Program instruction during erase suspend.

***Method of verification*:** Initiate WREN instruction to set WEL bit. Then, initiate sector erase instruction. Check WIP bit of status register to see if erase is in progress. Initiate erase suspend instruction and check if WIP bit is de-asserted. Initiate page program instruction on sector that is not suspended. Check WIP bit to see if programming has started. Initiate program suspend instruction. Initiate series of read instructions for memory array to check data. For read instructions that address the suspended sectors memory should output ©©©©. Initiate write bank register instruciton and check if write was successful by reading bank register. Initiate program resume instruction and check WIP bit to see if programing is in progress. Initiate read instruction to check programmed data.Initiate erase resume program and check WIP bit to see if erase is in progress. Initiate read array instructions to check if sector is erased (all ones)

3.40 *Bulk Erase*

3.40.1 Series 40. Positive Tests

a. BULKERS_40_a

***Testcase symbolic name*:** BULKERS_40_a

***Description*:** Test Bulk Erase operation

What is being verified: Bulk Erase Operation.

***Method of verification*:** Initiate WREN instruction to set WEL bit. Initiate WRR instruction to unprotect all sectors. After t_W verify that writing is done by reading status register. Initiate WREN instruction to set WEL bit. Initiate BE instruction and by reading status register verify that BULK_ERASE cycle is in progress. After t_{BE} verify that erasing is done by reading status register.

After erasing is over, verify that flash memory was erased by reading from them. Read from any location inside flash memory should return 0xFFFF.

3.41 *CRC ASO*

3.41.1 Series 41. Positive Tests

a. CRCASO_41_a

***Testcase symbolic name*:** CRCASO__41_a

***Description*:** Test CRC ASO, CRC calculation and read of CRC register

What is being verified: Entering CRC ASO and starting CRC calculation over a part of memory array. Read of CRC registers to verify correct calculation.

***Method of verification*:** Initiate Set CRC ASO enter instruction, set starting address for CRC calculation and set ending address for CRC calculation. Then, by reading status register verify that write cycle is in progress. After $t_{CRCSETUP}$ verify that calculation is done by reading status register. Initiate read RDAR instruction to verify that correct data has been read.

b.CRC_ASO _41_b

***Testcase symbolic name*:** CRC_ASO_41_b

***Description*:** CRC SUSPEND and read

What is being verified: CRC Suspend command during CRC operation

***Method of verification*:**

CRC ASO is entered, Load Starting CRC address. Read status register to check if device is busy.

When the CRC Suspend command is written during a CRC operation, the device requires a maximum of tCRCSL to suspend the program operation. Read status register to check if device is ready, enter CRC read command. Exit CRC, reset device.

3.42 SPI DPD

3.42.1 Series 42. Positive test

a. SPI_DPD_42_a

***Testcase symbolic name**:* SP_ DPD_42_a

***Description**:* Enter DPD from SPI mode

What is being verified: Enter to DPD mode from SPI mode command

***Method of verification*:**

Verificate does device enter to DPD from SPI mode.

b. SPI_DPD_42_b

***Testcase symbolic name**:* SPI_DPD_42_b

***Description**: DPD to Idle*

What is being verified: Testing if device is ready while going from DPD to Idle state during the tDPDOUT time and after tDPDOUT time.

***Method of verification*:**

After Idle state read status register to check if device is ready.The device should not be ready because a wakeup event did not occur. At the same time this access will be a wakeup event that will exit the device from DPD mode. During the tDPDOUT time the data will not be output After wakeup event wait for Deep Power Down to Idle wakeup time. Read status register to check if device is ready. Finish testing.

3.43 Sector Erase Count

3.43.1 Series 43. Positive test

a. HB_TO_SPI _43_a

***Testcase symbolic name**:* HB_TO_SPI_43_a

***Description**:* Enter to SPI mode from HyperBus.

What is being verified: Enter to SPI mode from HyperBus.

***Method of verification*:**

Verifiy Entering SPI mode from HyperBus mode.

b. SEC_ERASE_CNT_43_b

***Testcase symbolic name**:* SEC_ERASE_CNT_43_b

***Description**:* Read Sector Erase Count

What is being verified: Sector Erase Count command verified.

***Method of verification*:**

Verifiy Read Sector Erase Count command after multiple program-erase operations on same sector.

3.44 JEDEC RESET

3.44.1 Series 44. Positive test

a. JEDEC_44_a

***Testcase symbolic name**:* JEDEC_44_a

***Description**:* JEDEC Reset.

What is being verified: Test Software reset.

***Method of verification*:**

Verifiy JEDEC Reset.

3.45 READ Jedec ID, SFDP, Unique ID, WRAR for STRV1, Password program with wrar, 256B page program, and DYB bitts erasse

3.45.1 Series 45. Positive test

a. READ_JEDEC_ID_45_a

***Testcase symbolic name**:* READ_JEDEC_ID_45_a

***Description**:* READ Jedec ID

What is being verified: Reading Jedec ID

***Method of verification*:**

Verifiy READ Jedec ID.

b. READ_SFDP _45_b

***Testcase symbolic name**:* READ_SFDP_45_b

***Description**:* READ_SFDP.

What is being verified: Reading SFDP.

***Method of verification*:**

Verifiy reading SFDP.

c. READ_UID_45_c

***Testcase symbolic name**:* READ_UID_45_c

***Description**:* Reading UID.

What is being verified: Read UID.

***Method of verification*:**

Verifiy reading UID.

d. WRAR_STRV1_45_d

***Testcase symbolic name**:* WRAR_STRV1_45_d

***Description**:* Writing in

What is being verified: WRAR command for Status Registar 1, bitts 4,3,2

***Method of verification*:**

Verifiy that WRAR instruction can write STR1V[4:2]

e. WRAR_password_program_45_e

***Testcase symbolic name**:* WRAR,_password_program_45_e

***Description**:* Writting in and reading password

What is being verified: WRAR instruction for changing password

***Method of verification*:**

Verifiy WRAR instruction can write password

f. Page_Program_256B _45_f

***Testcase symbolic name**:* Page_Program256B_45_f

***Description**:* Programing 2 times with Page Program 256B instructio

What is being verified: pg_prog4 instruction for two consecutive instances

***Method of verification*:**

Verifiy that pg_prog4 instruction will program 2 pages consecutive

g. DYB_biits_erasse _45_g

***Testcase symbolic name**:* DYB_bitts_erasse_45_g

***Description**:* Errasing all DYB bitts with w_dyb4 instructio

What is being verified: That DYB bitts can be erassed and unlock those sectors

***Method of verification*:**

Verifiy that after DYB errasing, all sectors are unlocked