numonyx

N25Q256

256-Mbit 3 V, multiple I/O, 4-Kbyte subsector erase, XiP enabled, serial flash memory with 108 MHz SPI bus interface

Features

- SPI-compatible serial bus interface
- 108 MHz (maximum) clock frequency
- 2.7 V to 3.6 V single supply voltage
- Supports legacy SPI protocol and new Quad I/O or Dual I/O SPI protocol
- Quad/Dual I/O instructions resulting in an equivalent clock frequency up to 432 MHz:
- XIP mode for all three protocols
  - Configurable via volatile or non-volatile registers (enabling the memory to work in XiP mode directly after power on)
- Program/Erase suspend instructions
- Continuous read of entire memory via single instruction:
  - Fast Read
  - Quad or Dual Output Fast Read
  - Quad or Dual I/O Fast Read
- Flexible to fit application:
  - Configurable number of dummy cycles
  - Output buffer configurable
  - Software reset supported in all protocols
- 64-byte user-lockable, one-time programmable (OTP) area
- Erase capability
  - Subsector (4-Kbyte) granularity on the entire memory array.
  - Sector (64-Kbyte) granularity
- Write protections
  - Software write protection applicable to every 64-Kbyte sector (volatile lock bit)
  - Hardware write protection: protected area size defined by five non-volatile bits (BP0, BP1, BP2, BP3 and TB bit)
  - Additional smart protections available upon customer request
- Electronic signature
  - JEDEC standard two-byte signature (BA19h)
  - Additional 2 Extended Device ID (EDID) bytes to identify device factory options
  - Unique ID code (UID) with 14 bytes read-only
- More than 100,000 program/erase cycles per sector
- More than 20 years data retention
- Packages (All packages RoHS compliant):
  - F8 = VDFPN8 8 x 6 mm (MLP8) (Feasibility under evaluation)
  - SF = SO16W (SO16 Wide 300 mils body
  - width)
  - 12 = TBGA24 6 x 8 mm (Feasibility under evaluation)

Contents

1 Description 13

2 Signal descriptions 16

2.1 Serial data output (DQ1) 16

2.2 Serial data input (DQ0) 16

2.3 Serial Clock (C) 16

2.4 Chip Select (S) 16

2.5 Hold (HOLD) or Reset (Reset) 17

2.6 Write protect/enhanced program supply voltage ($\overline{W}$/VPP), DQ2 18

2.7 V_{CC} supply voltage 18

2.8 V_{SS} ground 18

3 SPI Modes 19

4 SPI Protocols 21

4.1 Selecting and Enabling a Protocol 21

4.2 Exiting DIO-SPI or QIO-SPI Protocols 21

4.3 Extended SPI protocol 21

4.4 Dual I/O SPI (DIO-SPI) protocol 21

4.5 Quad SPI (QIO-SPI) protocol 22

5 Operating features 24

5.1 Requirements for 256 Mb Address Space 24

5.1.1 3 Byte Address Mode (Segmentation) 24

5.1.2 4 Byte Address Mode 25

5.1.3 Instructions Not Affected by Addressing Mode 28

5.2 Extended SPI Protocol Operating features 28

5.2.1 Read Operations 28

5.2.2 Page programming 28

5.2.3 Dual Input Fast Program 28

5.2.4 Dual Input Extended Fast Program 28

5.2.5 Quad Input Fast Program 29

5.2.6 Quad Input Extended Fast Program 29

5.2.7 Subsector erase, sector erase and bulk erase 29
5.2.8 Polling during a write, program, or erase cycle 29
5.2.9 Active power and standby power modes 30
5.2.10 Hold (or Reset) condition 30
5.3 Dual SPI (DIO-SPI) Protocol 31
5.3.1 Multiple Read Identification 31
5.3.2 Dual Command Fast reading 31
5.3.3 Page programming 31
5.3.4 Subsector Erase, Sector Erase, and Bulk Erase 32
5.3.5 Polling during a Write, Program, or Erase cycle 32
5.3.6 Read and Modify registers 32
5.3.7 Active Power and Standby Power modes 32
5.3.8 HOLD (or Reset) condition 32
5.4 Quad SPI (QIO-SPI)Protocol 33
5.4.1 Multiple Read Identification 33
5.4.2 Quad Command Fast reading 33
5.4.3 QUAD Command Page programming 33
5.4.4 Subsector Erase, Sector Erase, and Bulk Erase 34
5.4.5 Polling during a Write, Program, or Erase cycle 34
5.4.6 Read and Modify registers 35
5.4.7 Active Power and Standby Power modes 35
5.4.8 HOLD (or Reset) condition 35
5.4.9 VPP pin Enhanced Supply Voltage feature 35

6 Volatile and Non Volatile Registers 36
6.1 Legacy SPI Status Register 38
6.1.1 WIP bit 38
6.1.2 WEL bit 38
6.1.3 BP3, BP2, BP1, BP0 bits 38
6.1.4 TB bit 38
6.1.5 SRWD bit 39
6.2 Non Volatile Configuration Register 39
6.2.1 Dummy clock cycle NV configuration bits (NVCR bits from 15 to 12) . . 41
6.2.2 XIP NV configuration bits (NVCR bits from 11 to 9) 42
6.2.3 Output Driver Strength NV configuration bits (NVCR bits from 8 to 6) . . 42
6.2.4 Hold (Reset) disable NV configuration bit (NVCR bit 4) 42
6.2.5 Quad Input NV configuration bit (NVCR bit 3) 42

6.2.6 Dual Input NV configuration bit (NVCR bit 2) 42
6.2.7 128Mb segment selection NV configuration bit (NVCR bit 1) 42
6.2.8 Address mode selection NV configuration bit (NVCR bit 0) 43
6.3 Volatile Configuration Register 43
6.3.1 Dummy clock cycle: VCR bits 7:4 44
6.3.2 XIP Volatile Configuration bits (VCR bit 3) 45
6.3.3 Wrap: VCR bits 1:0 45
6.4 Volatile Enhanced Configuration Register 46
6.4.1 Quad Input Command VECR<7> 46
6.4.2 Dual Input Command VECR<6> 47
6.4.3 Reset/Hold disable VECR<4> 47
6.4.4 Accelerator pin enable: QIO-SPI protocol / QIFP/QIEFP VECR<3> . . . 47
6.4.5 Output Driver Strength VECR<2:0> 48
6.5 Flag Status Register 48
6.5.1 P/E Controller Status bit 49
6.5.2 Erase Suspend Status bit 49
6.5.3 Erase Status bit 49
6.5.4 Program Status bit 49
6.5.5 VPP Status bit 50
6.5.6 Program Suspend Status bit 50
6.5.7 Protection Status bit 50
6.5.8 Protection Status bit (4-Byte) 51

7 Protection modes 52
7.1 SPI Protocol-related protections 52
7.2 Specific hardware and software protection 52

8 Memory organization 55

9 Instructions 60
9.1 Extended SPI Instructions 60
9.1.1 Read Identification (RDID) 64
9.1.2 Read Data Bytes (READ) 66
9.1.3 Read Data Bytes using 4 Byte Address (READ4BYTE) 67
9.1.4 Read Data Bytes at Higher Speed (FAST_READ) 68
9.1.5 Read Data Bytes at Higher Speed using 4 Byte Address (FAST_READ4BYTE) 69

9.1.6 Read Serial Flash Discovery Parameter 70
9.1.7 Dual Output Fast Read (DOFR) 71
9.1.8 Dual Output Fast Read using 4 Byte Address (DOFR4BYTE) 72
9.1.9 Dual I/O Fast Read 73
9.1.10 Dual Input/Output Fast Read using 4 Byte Address (DIOFR4BYTE) . . 74
9.1.11 Quad Output Fast Read 75
9.1.12 Quad Output Fast Read using 4 Byte Address (QOFR4BYTE) 76
9.1.13 Quad I/O Fast Read 78
9.1.14 Quad Input/Output Fast Read using 4 Byte Address (QIOFR4BYTE) . 79
9.1.15 Read OTP (ROTP) 80
9.1.16 Write Enable (WREN) 81
9.1.17 Write Disable (WRDI) 82
9.1.18 Page Program (PP) 83
9.1.19 Dual Input Fast Program (DIFP) 84
9.1.20 Dual Input Extended Fast Program 86
9.1.21 Quad Input Fast Program 86
9.1.22 Quad Input Extended Fast Program 88
9.1.23 Program OTP instruction (POTP) 88
9.1.24 Subsector Erase (SSE) 91
9.1.25 Sector Erase (SE) 92
9.1.26 Bulk Erase (BE) 93
9.1.27 Program/Erase Suspend 94
9.1.28 Program/Erase Resume 96
9.1.29 Read Status Register (RDSR) 96
9.1.30 Write status register (WRSR) 97
9.1.31 Read Lock Register (RDLR) 99
9.1.32 Write to Lock Register (WRLR) 100
9.1.33 Read Flag Status Register 101
9.1.34 Clear Flag Status Register 101
9.1.35 Read NV Configuration Register 102
9.1.36 Write NV Configuration Register 102
9.1.37 Read Volatile Configuration Register 103
9.1.38 Write Volatile Configuration Register 104
9.1.39 Read Volatile Enhanced Configuration Register 105
9.1.40 Write Volatile Enhanced Configuration Register 106
9.1.41 Read Extended Address Register 107
9.1.42 Write Extended Address Register 107

9.1.43 Enter 4-Byte Address Mode 108
9.1.44 Exit 4-Byte Address Mode 108
9.1.45 Reset Enable 108
9.2 DIO-SPI Instructions 110
9.2.1 Multiple I/O Read Identification protocol 113
9.2.2 Read Serial Flash Discovery Parameter 114
9.2.3 Dual Command Fast Read (DCFR) 115
9.2.4 Dual Command Fast Read using 4 Byte Address (DCFR4Byte) 116
9.2.5 Read OTP (ROTP) 116
9.2.6 Write Enable (WREN) 117
9.2.7 Write Disable (WRDI) 118
9.2.8 Dual Command Page Program (DCPP) 119
9.2.9 Program OTP instruction (POTP) 120
9.2.10 Subsector Erase (SSE) 121
9.2.11 Sector Erase (SE) 121
9.2.12 Bulk Erase (BE) 122
9.2.13 Program/Erase Suspend 122
9.2.14 Program/Erase Resume 123
9.2.15 Read Status Register (RDSR) 124
9.2.16 Write status register (WRSR) 124
9.2.17 Read Lock Register (RDLR) 125
9.2.18 Write to Lock Register (WRLR) 125
9.2.19 Read Flag Status Register 126
9.2.20 Clear Flag Status Register 126
9.2.21 Read NV Configuration Register, Dual I/O 127
9.2.22 Write NV Configuration Register, Dual I/O 127
9.2.23 Read Volatile Configuration Register, Dual I/O 128
9.2.24 Write Volatile Configuration Register, Dual I/O 128
9.2.25 Read Volatile Enhanced Configuration Register, Dual I/O 129
9.2.26 Write Volatile Enhanced Configuration Register, Dual I/O 129
9.2.27 Read Extended Address Register, Dual I/O 130
9.2.28 Write Extended Address Register, Dual I/O 130
9.2.29 Enter 4-Byte Address Mode, Dual I/O 131
9.2.30 Exit 4-Byte Address Mode, Dual I/O 131
9.2.31 Reset Enable and Reset Memory, Dual I/O 132
9.3 QIO-SPI Instructions 133
9.3.1 Multiple I/O Read Identification (MIORDID) 136

9.3.2 Read Serial Flash Discovery Parameter 137
9.3.3 Quad Command Fast Read (QCFR) 137
9.3.4 Quad Command Fast Read using 4 Byte Address (QCFR4Byte) 139
9.3.5 Read OTP (ROTP) 141
9.3.6 Write Enable (WREN) 142
9.3.7 Write Disable (WRDI) 142
9.3.8 Quad Command Page Program (QCPP) 143
9.3.9 Program OTP instruction (POTP) 146
9.3.10 Subsector Erase (SSE) 147
9.3.11 Sector Erase (SE) 147
9.3.12 Bulk Erase (BE) 148
9.3.13 Program/Erase Suspend 149
9.3.14 Program/Erase Resume 149
9.3.15 Read Status Register (RDSR) 150
9.3.16 Write status register (WRSR) 150
9.3.17 Read Lock Register (RDLR) 151
9.3.18 Write to Lock Register (WRLR) 151
9.3.19 Read Flag Status Register 152
9.3.20 Clear Flag Status Register 153
9.3.21 Read NV Configuration Register 153
9.3.22 Write NV Configuration Register 154
9.3.23 Read Volatile Configuration Register 155
9.3.24 Write Volatile Configuration Register 156
9.3.25 Read Volatile Enhanced Configuration Register 157
9.3.26 Write Volatile Enhanced Configuration Register 158
9.3.27 Read Extended Address Register, Quad I/O 159
9.3.28 Write Extended Address Register, Quad I/O 159
9.3.29 Enter 4-Byte Address Mode, Quad I/O 160
9.3.30 Exit 4-Byte Address Mode, Quad I/O 160
9.3.31 Reset Enable and Reset Memory, Quad I/O 161

10 XIP Operations 162

10.1 Enter XIP mode by setting the Non Volatile Configuration Register 163
10.2 Enter XIP mode by setting the Volatile Configuration Register 165
10.3 XIP mode hold and exit 166
10.4 XIP Memory reset after a controller reset 167

11 Power-up and power-down 168
11.1 Rescue sequence in case of power loss during WRNVCR 169

12 Initial delivery state 170

13 Maximum rating 170

14 DC and AC parameters 171

15 Package mechanical 179

16 Ordering information 183

17 Revision history 185

List of tables

Table 1. Signal names . . . 14
Table 2. Extended Address Register (EAR) . . . 24
Table 3. Status register format . . . 38
Table 4. Non-Volatile Configuration Register . . . 40
Table 5. Maximum operative frequency by dummy clock cycles . . . 41
Table 6. Volatile Configuration Register . . . 44
Table 7. Sequence of Bytes Read during Wrap Mode . . . 45
Table 8. Volatile Enhanced Configuration Register . . . 46
Table 9. Flag Status Register . . . 48
Table 10. Software protection truth table (Sectors 0 to 511, 64 Kbyte granularity) . . . 53
Table 11. Protected area sizes, Upper (TB bit = 0) . . . 54
Table 12. Memory Map Sectors 511:256 . . . 56
Table 13. Memory Map Sectors 255:128 . . . 57
Table 14. Memory Map Sectors 127:64 . . . 58
Table 15. Memory Map Sectors 63:0 . . . 59
Table 16. Instruction set: extended SPI protocol . . . 62
Table 17. Read Identification data-out sequence . . . 64
Table 18. Extended Device ID table (first byte) . . . 64
Table 19. Suspend Parameters . . . 94
Table 20. Operations Allowed / Disallowed During Device States . . . 95
Table 21. Protection modes . . . 98
Table 22. Lock Register out . . . 99
Table 23. Lock Register in . . . 100
Table 24. Instruction set: DIO-SPI protocol . . . 111
Table 25. Instruction set: QIO-SPI protocol . . . 134
Table 26. NVCR XIP bits setting example . . . 164
Table 27. VCR XIP bits setting example . . . 165
Table 28. Power-up timing and VWI threshold . . . 169
Table 29. Absolute maximum ratings . . . 170
Table 30. Operating conditions . . . 171
Table 31. AC measurement conditions . . . 171
Table 32. Capacitance . . . 171
Table 33. DC Characteristics . . . 172
Table 34. AC Characteristics . . . 173
Table 35. Reset Conditions . . . 175
Table 36. VDFPN8 (MLP8) Very Thin Dual Flat Package 8 leads 8×6x1 mm Dimensions . . . 179
Table 37. SO16 Wide – 16 Lead Plastic Small Outline, 300 mils Body Width, Dimensions . . . 180
Table 38. TBGA 6x8 mm 24-Ball, Dimensions, Symbols A to eE . . . 182
Table 39. TBGA 6x8 mm 24-Ball, Dimensions, Symbols FD to fff . . . 182
Table 40. Ordering information scheme . . . 183
Table 41. Valid Order Information Line Items . . . 184
Table 42. Document revision history . . . 185

List of figures

Figure 1. Logic diagram . . . 14
Figure 2. MLP8 Connections . . . 15
Figure 3. SO16 Connections . . . 15
Figure 4. BGA connections . . . 15
Figure 5. Bus master and memory devices on the SPI bus . . . 19
Figure 6. SPI Modes Supported . . . 20
Figure 7. Upper and Bottom 128 Bits . . . 24
Figure 8. 4-Byte Address on One Wire, Extended SPI . . . 25
Figure 9. 4-Byte Address on Two Wires, Extended SPI . . . 25
Figure 10. 4-Byte Address on Four Wires, Extended SPI . . . 26
Figure 11. 4-Byte Address on Two Wires, Dual SPI, Example . . . 26
Figure 12. 4-Byte Address on Four Wires, Quad SPI Example . . . 27
Figure 13. 4-Byte Address Mode . . . 27
Figure 14. Hold condition activation . . . 30
Figure 15. Non Volatile and Volatile configuration Register Scheme . . . 37
Figure 16. 4-Byte Address Mode Default by NVCR . . . 43
Figure 17. Block diagram . . . 55
Figure 18. Read identification instruction and data-out sequence . . . 65
Figure 19. Read Data Bytes instruction and data-out sequence . . . 66
Figure 20. Read Data Bytes using 4 Byte Address Instruction and Data-Out Sequence . . . 67
Figure 21. Read Data Bytes at Higher Speed instruction and data-out sequence . . . 68
Figure 22. Read Data Bytes at Higher Speed using 4 Byte Address Instruction . . . 69
Figure 23. Read Serial Flash Discovery Instruction and Data-out Sequence . . . 70
Figure 24. Dual Output Fast Read Instruction Sequence . . . 71
Figure 25. Dual Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence . . . 72
Figure 26. Dual I/O Fast Read Instruction Sequence . . . 73
Figure 27. Dual Input/Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence . 74
Figure 28. Quad Output Fast Read instruction sequence . . . 75
Figure 29. Quad Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence . . . 77
Figure 30. Quad Input/Output Fast Read instruction sequence . . . 78
Figure 31. Quad Input/Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence 79
Figure 32. Read OTP instruction and data-out sequence . . . 80
Figure 33. Write Enable instruction sequence . . . 81
Figure 34. Write Disable instruction sequence . . . 82
Figure 35. Page Program Instruction Sequence . . . 84
Figure 36. Dual Input Fast Program instruction sequence . . . 85
Figure 37. Dual Input Extended Fast Program instruction sequence . . . 86
Figure 38. Quad Input Fast Program Instruction Sequence . . . 87
Figure 39. Quad Input Extended Fast Program instruction sequence . . . 88
Figure 40. Program OTP instruction sequence . . . 89
Figure 41. How to permanently lock the OTP bytes . . . 90
Figure 42. Subsector Erase instruction sequence . . . 91
Figure 43. Sector Erase instruction sequence . . . 92
Figure 44. Bulk Erase instruction sequence . . . 93
Figure 45. Read Status Register instruction sequence . . . 96
Figure 46. Write Status Register instruction sequence . . . 97
Figure 47. Read Lock Register instruction and data-out sequence . . . 99
Figure 48. Write to Lock Register instruction sequence . . . 100

Figure 49. Read Flag Status Register instruction sequence . . . 101
Figure 50. Clear Flag Status Register instruction sequence . . . 101
Figure 51. Read NV Configuration Register instruction sequence . . . 102
Figure 52. Write NV Configuration Register instruction sequence . . . 103
Figure 53. Read Volatile Configuration Register instruction sequence . . . 103
Figure 54. Write Volatile Configuration Register instruction sequence . . . 104
Figure 55. Read Volatile Enhanced Configuration Register instruction sequence . . . 105
Figure 56. Write Volatile Enhanced Configuration Register instruction sequence . . . 106
Figure 57. Read Extended Address Register Instruction Sequence . . . 107
Figure 58. Write Extended Address Register Instruction Sequence . . . 107
Figure 59. Enter 4-Byte Address Mode Instruction Sequence . . . 108
Figure 60. Exit 4-Byte Address Mode Instruction Sequence . . . 108
Figure 61. Reset Enable and Reset Memory Instruction Sequence . . . 109
Figure 62. Multiple I/O Read Identification instruction and data-out sequence DIO-SPI . . . 113
Figure 63. Dual Read Serial Flash Discovery Parameter . . . 114
Figure 64. Dual Command Fast Read instruction and data-out sequence DIO-SPI . . . 115
Figure 65. Dual Command Fast Read using 4 Byte Address Instruction and Data-Out Sequence . . . 116
Figure 66. Read OTP instruction and data-out sequence DIO-SPI . . . 117
Figure 67. Write Enable instruction sequence DIO-SPI . . . 117
Figure 68. Write Disable instruction sequence DIO-SPI . . . 118
Figure 69. Dual Command Page Program instruction sequence DSP, 02h . . . 119
Figure 70. Dual Command Page Program instruction sequence DSP, A2h . . . 119
Figure 71. Dual Command Page Program instruction sequence DSP, D2h . . . 120
Figure 72. Program OTP instruction sequence DIO-SPI . . . 120
Figure 73. Subsector Erase instruction sequence DIO-SPI . . . 121
Figure 74. Sector Erase instruction sequence DIO-SPI . . . 122
Figure 75. Bulk Erase instruction sequence DIO-SPI . . . 122
Figure 76. Program/Erase Suspend instruction sequence DIO-SPI . . . 123
Figure 77. Program/Erase Resume instruction sequence DIO-SPI . . . 123
Figure 78. Read Status Register instruction sequence DIO-SPI . . . 124
Figure 79. Write Status Register instruction sequence DIO-SPI . . . 124
Figure 80. Read Lock Register instruction and data-out sequence DIO-SPI . . . 125
Figure 81. Write to Lock Register instruction sequence DIO-SPI . . . 125
Figure 82. Read Flag Status Register instruction sequence DIO-SPI . . . 126
Figure 83. Clear Flag Status Register instruction sequence DIO-SPI . . . 126
Figure 84. Read NV Configuration Register instruction sequence DIO-SPI . . . 127
Figure 85. Write NV Configuration Register instruction sequence DIO-SPI . . . 127
Figure 86. Read Volatile Configuration Register instruction sequence DIO-SPI . . . 128
Figure 87. Write Volatile Configuration Register instruction sequence DIO-SPI . . . 128
Figure 88. Read Volatile Enhanced Configuration Register instruction sequence DIO-SPI . . . 129
Figure 89. Write Volatile Enhanced Configuration Register Instruction Sequence, Dual I/O . . . 129
Figure 90. Read Extended Address Register Instruction Sequence, Dual I/O . . . 130
Figure 91. Write Extended Address Register Instruction Sequence, Dual I/O . . . 130
Figure 92. Enter 4-Byte Address Mode Instruction Sequence, Dual I/O . . . 131
Figure 93. Exit 4-Byte Address Mode Instruction Sequence, Dual I/O . . . 131
Figure 94. Reset Enable and Reset Memory Instruction Sequence, Dual I/O . . . 132
Figure 95. Multiple I/O Read Identification instruction and data-out sequence QIO-SPI . . . 136
Figure 96. Quad Read Serial Flash Discovery Parameter . . . 137
Figure 97. Quad Command Fast Read instruction and data-out sequence QSP, 0Bh . . . 138
Figure 98. Quad Command Fast Read instruction and data-out sequence QSP, 6Bh . . . 138
Figure 99. Quad Command Fast Read instruction and data-out sequence QSP, EBh . . . 139
Figure 100. Quad Command Fast Read using 4 Byte Address Instruction QSP, 0Ch . . . 140

Figure 101. Quad Command Fast Read using 4 Byte Address Instruction QSP, 6Ch 140
Figure 102. Quad Command Fast Read using 4 Byte Address Instruction QSP, ECh 141
Figure 103. Read OTP instruction and data-out sequence QIO-SPI 142
Figure 104. Write Enable instruction sequence QIO-SPI 142
Figure 105. Write Disable instruction sequence QIO-SPI 143
Figure 106. Quad Command Page Program instruction sequence QIO-SPI, 02h 144
Figure 107. Quad Command Page Program instruction sequence QIO-SPI, 12h 144
Figure 108. Quad Command Page Program instruction sequence QIO-SPI, 32h 145
Figure 109. Program OTP instruction sequence QIO-SPI 146
Figure 110. Subsector Erase instruction sequence QIO-SPI 147
Figure 111. Sector Erase instruction sequence QIO-SPI 147
Figure 112. Bulk Erase instruction sequence QIO-SPI 148
Figure 113. Program/Erase Suspend instruction sequence QIO-SPI 149
Figure 114. Program/Erase Resume instruction sequence QIO-SPI 149
Figure 115. Read Status Register instruction sequence QIO-SPI 150
Figure 116. Write Status Register instruction sequence QIO-SPI 150
Figure 117. Read Lock Register instruction and data-out sequence QIO-SPI 151
Figure 118. Write to Lock Register instruction sequence QIO-SPI 151
Figure 119. Read Flag Status Register instruction sequence QIO-SPI 152
Figure 120. Clear Flag Status Register instruction sequence QIO-SPI 153
Figure 121. Read NV Configuration Register instruction sequence QIO-SPI 153
Figure 122. Write NV Configuration Register instruction sequence QIO-SPI 154
Figure 123. Read Volatile Configuration Register instruction sequence QIO-SPI 155
Figure 124. Write Volatile Configuration Register instruction sequence QIO-SPI 156
Figure 125. Read Volatile Enhanced Configuration Register instruction sequence QIO-SPI 157
Figure 126. Write Volatile Enhanced Configuration Register instruction sequence QIO-SPI 158
Figure 127. Read Extended Address Register Instruction Sequence, Quad I/O 159
Figure 128. Write Extended Address Register Instruction Sequence, Quad I/O 159
Figure 129. Enter 4-Byte Address Mode Instruction Sequence, Quad I/O 160
Figure 130. Exit 4-Byte Address Mode Instruction Sequence, Quad I/O 160
Figure 131. Reset Enable and Reset Memory Instruction Sequence, Quad I/O 161
Figure 132. Read functionality Flow Chart 163
Figure 133. XIP mode directly after power on 164
Figure 134. XiP: enter by VCR 2/2 (QIOFR in normal SPI protocol example) 166
Figure 135. Power-up timing 168
Figure 136. AC measurement I/O waveform 171
Figure 137. Reset AC waveforms while a program or erase cycle is in progress 175
Figure 138. Reset Enable 175
Figure 139. Serial input timing 176
Figure 140. Write protect setup and hold timing during WRSR when SRWD=1 177
Figure 141. Hold timing 177
Figure 142. Output timing 178
Figure 143. VPP_H timing 178
Figure 144. VDFPN8 (MLP8) Very Thin Dual Flat Package 8 leads, 8×6x1 mm Drawing 179
Figure 145. SO16 wide - 16-lead plastic small outline, 300 mils body width, Drawing 180
Figure 146. TBGA - 6 x 8 mm, 24-ball, mechanical package outline (Feasability Evaluation) 181

1 Description

The N25Q256 is a 256 Mbit serial Flash memory, with advanced write protection mechanisms. It is accessed by a high speed SPI-compatible bus and features the possibility to work in XIP (“eXecution in Place”) mode.

The N25Q256 supports innovative, high-performance quad/dual I/O instructions, these new instructions allow to double or quadruple the transfer bandwidth for read and program operations.

Furthermore the memory can be operated with 3 different protocols:

- Standard SPI (Extended SPI protocol)
- Dual I/O SPI
- Quad I/O SPI

The Standard SPI protocol is enriched by the new quad and dual instructions (Extended SPI protocol). For Dual I/O SPI (DIO-SPI) all the instructions codes, the addresses and the data are always transmitted across two data lines. For Quad I/O SPI (QIO-SPI) the instructions codes, the addresses and the data are always transmitted across four data lines thus enabling a tremendous improvement in both random access time and data throughput.

The memory can work in “XIP mode”, that means the device only requires the addresses and not the instructions to output the data. This mode dramatically reduces random access time thus enabling many applications requiring fast code execution without shadowing the memory content on a RAM.

The XIP mode can be used with QIO-SPI, DIO-SPI, or Extended SPI protocol, and can be entered and exited using different dedicated instructions to allow maximum flexibility: for applications required to enter in XIP mode right after power up of the device, this can be set as default mode by using dedicated Non Volatile Register (NVR) bits.

Another feature is the ability to pause and resume program and erase cycles by using dedicated Program/Erase Suspend and Resume instructions.

The N25Q256 memory offers the following additional features to be configured by using the Non Volatile Configuration Register (NVCR) for default /Non-Volatile settings or by using the Volatile and Volatile Enhanced Configuration Registers for Volatile settings:

- the number of dummy cycles for fast read instructions (single, dual and, quad I/O) according to the operating frequency
- the output buffer impedance
- the type of SPI protocol (extended SPI, DIO-SPI or QIO-SPI)
- the required XIP mode
- the Hold (Reset) functionality enabling/disabling
- the Wrap mode enabling/disabling

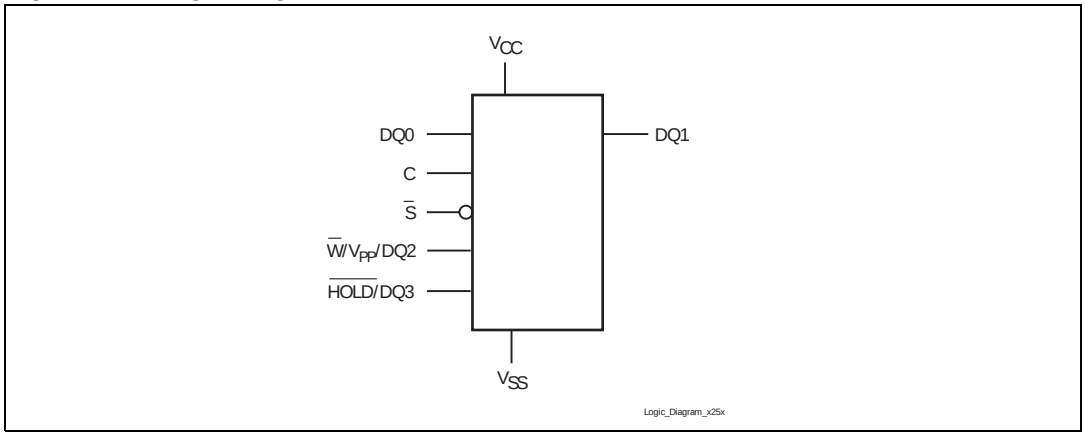
The memory is organized as 512 (64-Kbyte) main sectors that are further divided into 16 subsectors each (8192 subsectors in total). The memory can be erased a 4-KByte subsector at a time, a 64-KByte sector at a time, or as a whole.

The memory can be write protected by software using a mix of volatile and non-volatile protection features, depending on the application needs. The protection granularity is of 64-Kbyte (sector granularity) for volatile protections.

The N25Q256 has 64 one-time-programmable bytes (OTP bytes) that can be read and programmed using two dedicated instructions, Read OTP (ROTP) and Program OTP (POTP), respectively. These 64 bytes can be permanently locked by a particular Program OTP (POTP) sequence. Once they have been locked, they become read-only and this state cannot be reversed.

Many different N25Q256 configurations are available, please refer to the ordering scheme page for the possibilities. Additional features are available as security options (The Security features are described in a dedicated Application Note). Please contact your nearest Numonyx Sales office for more information.

Figure 1. Logic diagram



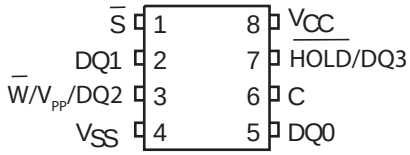
Note: Reset functionality is available in devices with a dedicated part number. See Section 16: Ordering information.

Table 1. Signal names

Signal	Description	I/O
C	Serial Clock	Input
DQ0	Serial Data input	I/O(1)
DQ1	Serial Data output	I/O(2)
$\overline{S}$	Chip Select	Input
$\overline{W}$/VPP/DQ2	Write Protect/Enhanced Program supply voltage/additional data I/O	I/O(3)
$\overline{HOLD}$/DQ3(4)	Hold (Reset function available upon customer request)/additional data I/O	I/O(3)
V_{CC}	Supply voltage	–
V_{SS}	Ground	–

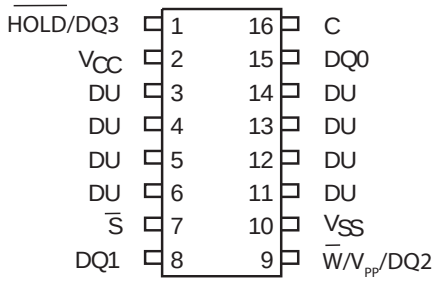
1. Provides dual and quad I/O for Extended SPI protocol instructions, dual I/O for Dual I/O SPI protocol instructions, and quad I/O for Quad I/O SPI protocol instructions.
2. Provides dual and quad instruction input for Extended SPI protocol, dual instruction input for Dual I/O SPI protocol, and quad instruction input for Quad I/O SPI protocol.
3. Provides quad I/O for Extended SPI protocol instructions, and quad I/O for Quad I/O SPI protocol instructions.
4. Reset functionality available with a dedicated part number. See *Section 16: Ordering information*.

Figure 2. MLP8 Connections



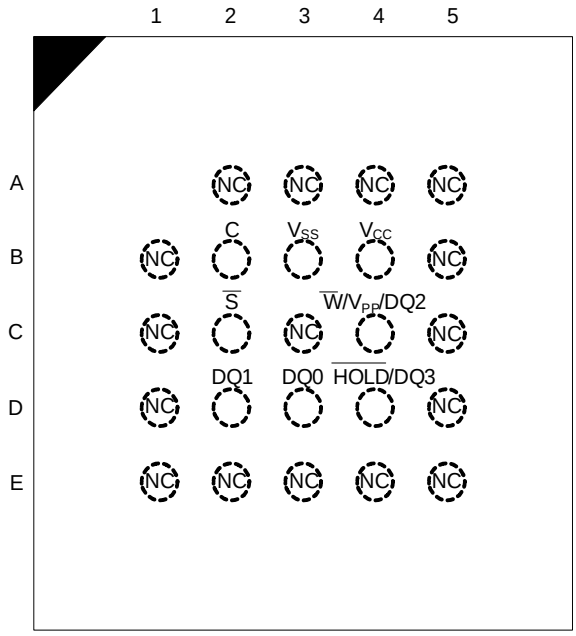
1. Reset functionality available in devices with a dedicated part number. See *Section 16: Ordering information*.

Figure 3. SO16 Connections



1. DU = don't use.
2. See *Package mechanical* section for package dimensions, and how to identify pin-1.
3. Reset functionality available in devices with a dedicated part number. See *Section 16: Ordering information*.

Figure 4. BGA connections



1. NC = No Connect.
2. See *Figure 146.: TBGA - 6 x 8 mm, 24-ball, mechanical package outline (Feasability Evaluation)*.

2 Signal descriptions

2.1 Serial data output (DQ1)

This output signal is used to transfer data serially out of the device. Data are shifted out on the falling edge of Serial Clock (C). When used as an Input, It is latched on the rising edge of the Serial Clock (C).

In the Extended SPI protocol, during the Quad and Dual Input Fast Program (QIFP, DIFP) instructions and during the Quad and Dual Input Extended Fast Program (QIEFP, DIEFP) instructions, pin DQ1 is used also as an input.

In the Dual I/O SPI protocol (DIO-SPI) the DQ1 pin always acts as an input/output.

In the Quad I/O SPI protocol (QIO-SPI) the DQ1 pin always acts as an input/output, with the exception of the Program or Erase cycle performed with the Enhanced Program Supply Voltage (VPP). In this case the device temporarily goes in Extended SPI protocol. The protocol then becomes QIO-SPI as soon as the VPP pin voltage goes low.

2.2 Serial data input (DQ0)

This input signal is used to transfer data serially into the device. It receives instructions, addresses, and the data to be programmed. Values are latched on the rising edge of Serial Clock (C). Data are shifted out on the falling edge of the Serial Clock (C).

In the Extended SPI protocol, during the Quad and Dual Output Fast Read (QOFR, DOFR) and the Quad and Dual Input/Output Fast Read (QIOFR, DIOFR) instructions, pin DQ0 is also used as an input/output.

In the DIO-SPI protocol the DQ0 pin always acts as an input/output.

In the QIO-SPI protocol, the DQ0 pin always acts as an input/output, with the exception of the Program or Erase cycle performed with the VPP. In this case the device temporarily goes in Extended SPI protocol. Then, the protocol returns to QIO-SPI as soon as the VPP pin voltage goes low.

2.3 Serial Clock (C)

This input signal provides the timing for the serial interface. Instructions, addresses, or data present at serial data input (DQ0) are latched on the rising edge of Serial Clock (C). Data are shifted out on the falling edge of the Serial Clock (C).

2.4 Chip Select ($\overline{S}$)

When this input signal is high, the device is deselected and serial data output (DQ1) is at high impedance. Unless an internal program, erase or write status register cycle is in progress, the device will be in the standby power mode. Driving Chip Select ($\overline{S}$) low enables the device, placing it in the active power mode. After power-up, a falling edge on Chip Select ($\overline{S}$) is required prior to the start of any instruction.

2.5 Hold ($\overline{\text{HOLD}}$) or Reset ($\overline{\text{Reset}}$)

The Hold ($\overline{\text{HOLD}}$) signal is used to pause any serial communications with the device without deselecting the device.

Reset functionality is present instead of Hold in devices with a dedicated part number. See *Section 16: Ordering information*.

During Hold condition, the Serial Data output (DQ1) is in high impedance, and Serial Data input (DQ0) and Serial Clock (C) are Don't Care.

To start the Hold condition, the device must be selected, with Chip Select ($\overline{\text{S}}$) driven Low.

For devices featuring Reset instead of Hold functionality, the Reset ($\overline{\text{Reset}}$) input provides a hardware reset for the memory.

When Reset ($\overline{\text{Reset}}$) is driven High, the memory is in the normal operating mode. When Reset ($\overline{\text{Reset}}$) is driven Low, the memory will enter the Reset mode. In this mode, the output is high impedance.

Driving Reset ($\overline{\text{Reset}}$) Low while an internal operation is in progress will affect this operation (write, program or erase cycle) and data may be lost.

In the Extended SPI protocol, during the QOFR, QIOFR, QIFP and the Quad Extended Fast Program (QIEFP) instructions, the Hold ($\overline{\text{Reset}}$) / DQ3 is used as an input/output (DQ3 functionality).

In QIO-SPI, the $\overline{\text{Hold}}$ ($\overline{\text{Reset}}$) / DQ3 pin acts as an I/O (DQ3 functionality), and the $\overline{\text{HOLD}}$ ($\overline{\text{Reset}}$) functionality disabled when the device is selected. When the device is deselected ($\overline{\text{S}}$ signal is high), in parts with Reset functionality, it is possible to reset the device unless this functionality is not disabled by mean of dedicated registers bits.

The HOLD (Reset) functionality can be disabled using bit 3 of the NVCR or bit 4 of the VECR.

2.6 Write protect/enhanced program supply voltage ($\overline{W}$/VPP), DQ2

$\overline{W}$/VPP/DQ2 can be used as:

- A protection control input.
- A power supply pin.
- I/O in Extended SPI protocol quad instructions and in QIO-SPI protocol instructions.

When the device is operated in Extended SPI protocol with single or dual instructions, the two functions $\overline{W}$ or VPP are selected by the voltage range applied to the pin. If the $\overline{W}$/VPP input is kept in a low voltage range (0 V to VCC) the pin is seen as a control input. This input signal is used to freeze the size of the area of memory that is protected against program or erase instructions (as specified by the values in the BP[0:3] bits of the Status Register. (See *Table 3.: Status register format*).

If VPP is in the range of VPPH, it acts as an additional power supply during the Program or Erase cycles (See *Table 30.: Operating conditions*). In this case VPP must be stable until the Program or Erase algorithm is completed.

During the Extended SPI protocol, the QOFR and QIOFR instructions, and the QIO-SPI protocol instructions, the pin $\overline{W}$/VPP/DQ2 is used as an input/output (DQ2 functionality).

Using the Extended SPI protocol the QIFP, QIEFP and the QIO-SPI Program/Erase instructions, it is still possible to use the VPP additional power supply to speed up internal operations. However, to enable this possibility it is necessary to set bit 3 of the Volatile Enhanced Configuration Register to 0.

In this case the $\overline{W}$/VPP/DQ2 pin is used as an I/O pin until the end of the instruction sequence. After the last input data is shifted in, the application should apply VPP voltage to $\overline{W}$/VPP/DQ2 within 200 ms to speed up the internal operations. If the VPP voltage is not applied within 200 ms the Program/Erase operations start with standard speed.

The default value of the VECR bit 3 is 1, and the VPP functionality for Quad I/O modify instruction is disabled.

2.7 V_{CC} supply voltage

V_{CC} is the supply voltage.

2.8 V_{SS} ground

V_{SS} is the reference for the V_{CC} supply voltage.

3 SPI Modes

These devices can be driven by a micro controller with its SPI peripheral running in either of the two following modes:

CPOL=0, CPHA=0

CPOL=1, CPHA=1

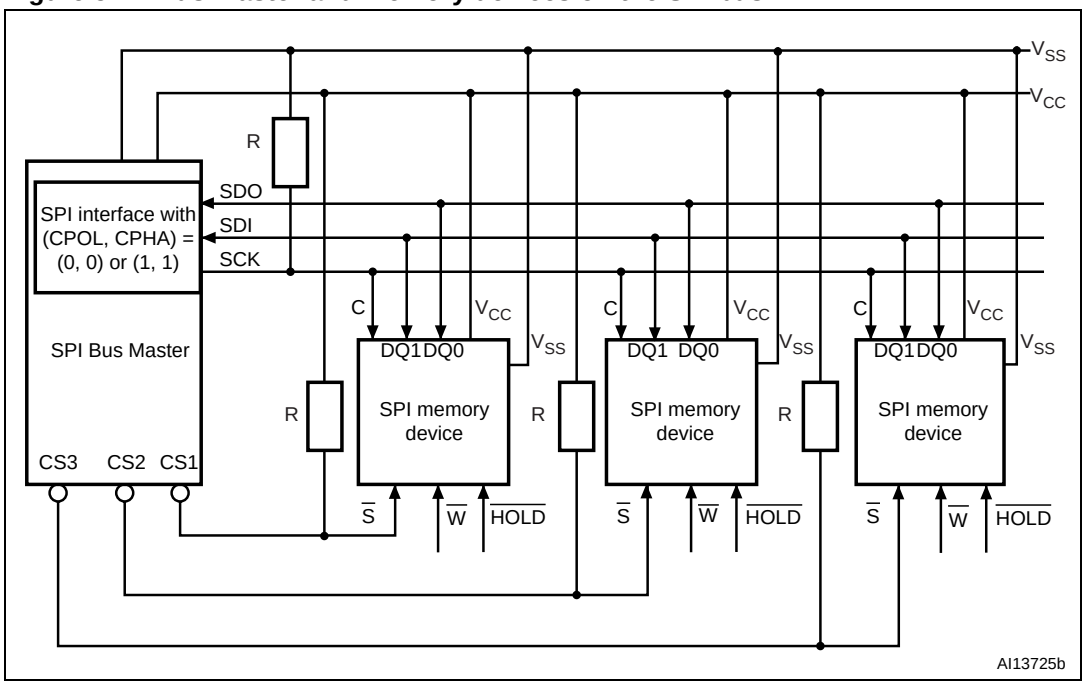
For these two modes, input data is latched in on the rising edge of Serial Clock (C), and output data is available from the falling edge of Serial Clock (C).

The difference between the two modes, as shown in *Figure 5*, is the clock polarity when the bus master is in standby mode and not transferring data:

C remains at 0 for (CPOL=0, CPHA=0)

C remains at 1 for (CPOL=1, CPHA=1)

Figure 5. Bus master and memory devices on the SPI bus

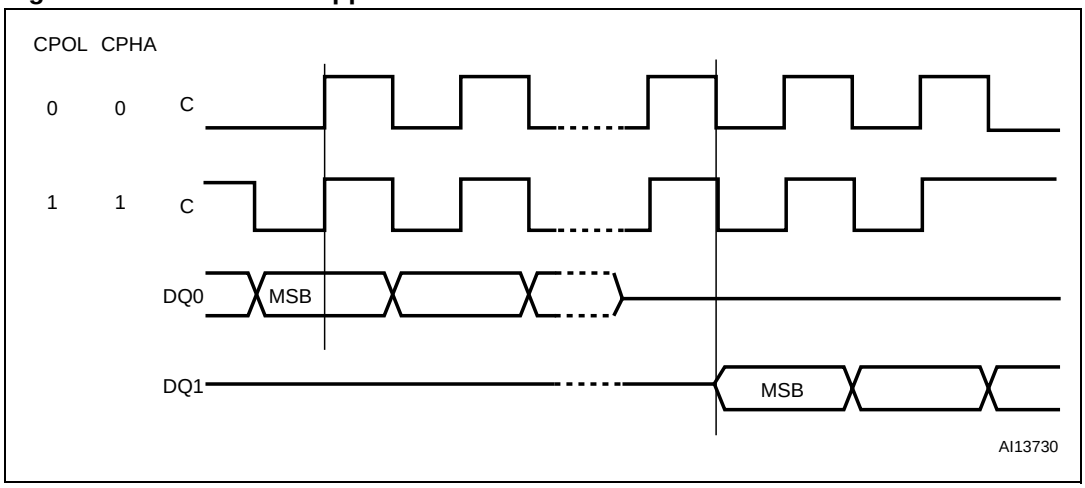


Shown here is an example of three devices working in Extended SPI protocol for simplicity connected to an MCU, on an SPI bus. Only one device is selected at a time, so only one device drives the serial data output (DQ1) line at a time; the other devices are high impedance. Resistors R ensures that the N25Q256 is not selected if the bus master leaves the $\overline{S}$ line in the high impedance state. As the bus master may enter a state where all inputs/outputs are in high impedance at the same time (for example, when the bus master is reset), the clock line (C) must be connected to an external pull-down resistor so that, when all inputs/outputs become high impedance, the $\overline{S}$ line is pulled High while the C line is pulled Low. This ensures that $\overline{S}$ and C do not become High at the same time, and so that the t_{SHCH} requirement is met. The typical value of R is 100 kΩ, assuming that the time constant $R*C_p$

(C_p = parasitic capacitance of the bus line) is shorter than the time during which the bus master leaves the SPI bus in high impedance.

Example: C_p = 50 pF, that is R*C_p = 5 µs <=> the application must ensure that the bus master never leaves the SPI bus in the high impedance state for a time period shorter than 5 µs. The Write Protect (W) and Hold (HOLD) signals should be driven, High or Low as appropriate.

Figure 6. SPI Modes Supported



4 SPI Protocols

The N25Q256 memory can work with 3 different Serial protocols:

- Extended SPI protocol.
- Dual I/O SPI (DIO-SPI) protocol.
- Quad I/O SPI (QIO-SPI) protocol.

4.1 Selecting and Enabling a Protocol

It is possible to choose among and enable or disable any of the three protocols by the user volatile or non-volatile configuration bit settings (VECR or NVCR bits). It's not possible to mix Extended SPI, DIO-SPI, and QIO-SPI protocols. However, the device can operate in XIP mode in all three protocols.

4.2 Exiting DIO-SPI or QIO-SPI Protocols

In addition to exiting the DIO-SPI or QIO-SPI protocols by the volatile or nonvolatile configuration bit settings (VECR or NVCR bits), it is also possible to exit from either of these protocols by using the following FFh sequence:

- DQ0 (PAD DATA) and DQ3 (PAD HOLD) = 1 for 8 clock cycles within S low
- S becomes high before the 9th clock cycle
- After this sequence, the Extended SPI protocol is active.

Note: This sequence does not work when the device is functioning in DIO-SPI or QIO-SPI in XiP mode.

4.3 Extended SPI protocol

This is an extension of the standard (legacy) SPI protocol. Instructions are transmitted on a single data line (DQ0), while addresses and data are transmitted by one, two or four data lines (DQ0, DQ1, W/VPP(DQ2) and HOLD / (DQ3) according to the instruction.

When used in the Extended SPI protocol, these devices can be driven by a micro controller in either of the two following modes:

- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

Please refer to the SPI modes for a detailed description of these two modes

4.4 Dual I/O SPI (DIO-SPI) protocol

Dual I/O SPI (DIO-SPI) protocol: instructions, addresses and I/O data are always transmitted on two data lines (DQ0 and DQ1).

Also when in DIO-SPI mode, the device can be driven by a micro controller in either of the two following modes:

- CPOL= 0, CPHA= 0
- CPOL= 1, CPHA= 1

Please refer to the SPI modes for a detailed description of these two modes.

Note: Extended SPI protocol Dual I/O instructions allow only address and data to be transmitted over two data lines. However, DIO-SPI *allows instructions, addresses, and data to be transmitted on two data lines.*

This mode can be set using two ways

- **Volatile:** by setting bit 6 of the VECR to 0. The device enters DIO-SPI protocol immediately after the Write Enhanced Volatile Configuration Register sequence completes. The device returns to the default working mode (defined by NVCR) on power on.
- **Default/ Non-Volatile:** This is default mode on power-up. By setting bit 2 of the NVCR to 0. The device enters DIO-SPI protocol on the subsequent power-on. After all subsequent power-on sequences, the device still starts in DIO-SPI protocol unless bit 2 of NVCR is set to 1 (default value, corresponding to Extended SPI protocol) or bit 3 of NVCR is set to 0 (corresponding to QIO-SPI protocol).

4.5 Quad SPI (QIO-SPI) protocol

Quad SPI (QIO-SPI) protocol: instructions, addresses, and I/O data are always transmitted on four data lines DQ0, DQ1, $\overline{W}$/VPP(DQ2), and $\overline{HOLD}$ / (DQ3).

The exception is the Program/Erase cycle performed with the VPP, in which case the device temporarily goes to Extended SPI protocol. Going temporarily into Extended SPI protocol allows the application either to:

- check the polling bits: WIP bit in the Status Register or Program/Erase Controller bit in the Flag Status Register
- perform Program/Erase suspend functions.

Note: As soon as the VPP pin voltage goes low, the protocol returns to the QIO-SPI protocol.

In QIO-SPI protocol the $\overline{W}$ and HOLD/ (RESET) functionality is disabled when the device is selected ($\overline{S}$ signal low).

When used in the QIO-SPI mode, these devices can be driven by a micro controller in either of the two following modes:

- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

Please refer to the SPI modes for a detailed description of the 2 modes.

Note: In the Extended SPI protocol only Address and data are allowed to be transmitted on 4 data lines, However in QIO-SPI protocol, the address, data and instructions are transmitted across 4 data lines.

This working mode is set in either bit 7 of the Volatile Enhanced Configuration Register (VECR) or in bit 3 of the Non Volatile Configuration Register (NVCR).

This mode can be set using two ways

- **Volatile:** by setting bit 7 of the VECR to 0, the device enters QIO-SPI protocol immediately after the Write Enhanced Volatile Configuration Register sequence completes. The device returns to the default working protocol (defined by the NVCR) on the next power on.
- **Default/ Non- Volatile:** This is default protocol on power up. By setting bit 3 of the NVCR to 0, the device enters QIO-SPI protocol on the subsequent power-on. After all subsequent power-on sequences, the device still starts in QIO-SPI protocol unless bit 3 of the NVCR is set to 1 (default value, corresponding to Extended SPI mode).

5 Operating features

5.1 Requirements for 256 Mb Address Space

When the memory size is 256Mb, memory needs to be addressed using a 32 bit address. Therefore, one additional bit is necessary to addressability in N25Q256 respect 24 bit address used to address memory size lower than 256Mb and bits A31'A25 are don't care.

The management of the added address bit will be implemented in two different ways:

- by means a segmentation of the array in two segments of 128Mbit each one, so that user can choose the segment in which run a modify operation or start a reading operation;
- by means an additional address byte so that the total number of address bytes to be specified directly by user is four.

5.1.1 3 Byte Address Mode (Segmentation)

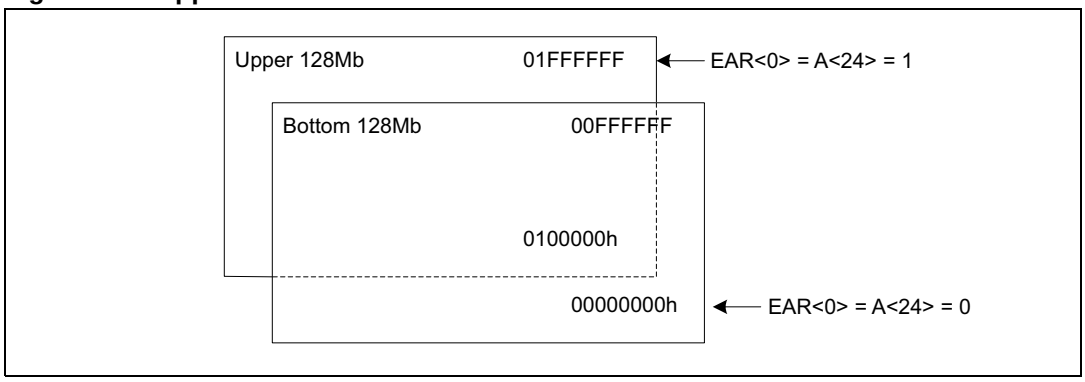
The 256 Mb device includes an 8-bit volatile register to manage segmentations: it contains the 4th byte address (A31 ' A24) to allow addressability beyond 128Mb when 3-byte address mode is enabled.

Table 2. Extended Address Register (EAR)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A31	A30	A29	A28	A27	A26	A25	A24

A32 to A25 are Don’t Care. During EAR, reading these bits will read as 0. The default value of bit <0> in EAR is 0; this value can be changed by NVCR bit.

Figure 7. Upper and Bottom 128 Bits



Erase/Program operations act in the selected segment, Bulk Erase instruction erases the entire device, Read operations start in the selected 128Mbit, but they are not bounded in it: when the last byte of the segment is reached, the next out byte (in a continuous reading) is the first of the other segment so that download of the whole array is possible with one instruction. The value in the EAR does not change when a read operation crosses the selected 128Mbit boundary.

In 3-address byte modality, XiP is possible only in the selected segment.

5.1.2 4 Byte Address Mode

In 4-byte address mode all instructions requiring an address must be issued with 4 address byte. Two dedicated instructions are available to enter/exit this modality:

- Enter 4-byte address mode (EN4BYTEADDR)
- Exit 4-byte address mode (EX4BYTEADDR)

In this modality EAR<0> becomes 'don't care'

When 4-byte address mode is enabled, all instructions requiring address must be issued with 4 address byte. The figures below show a typical read instruction using the Extended SPI protocol and a 4-byte address.

Figure 8. 4-Byte Address on One Wire, Extended SPI

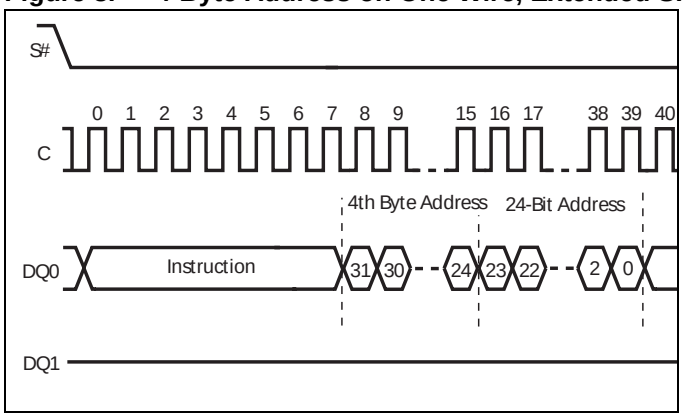


Figure 9. 4-Byte Address on Two Wires, Extended SPI

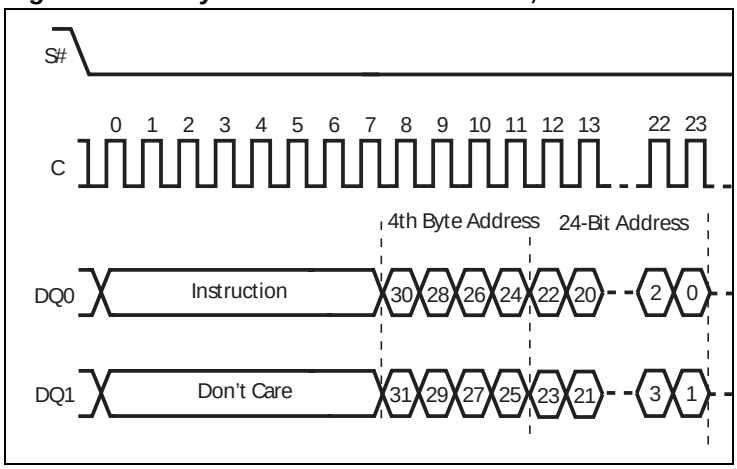
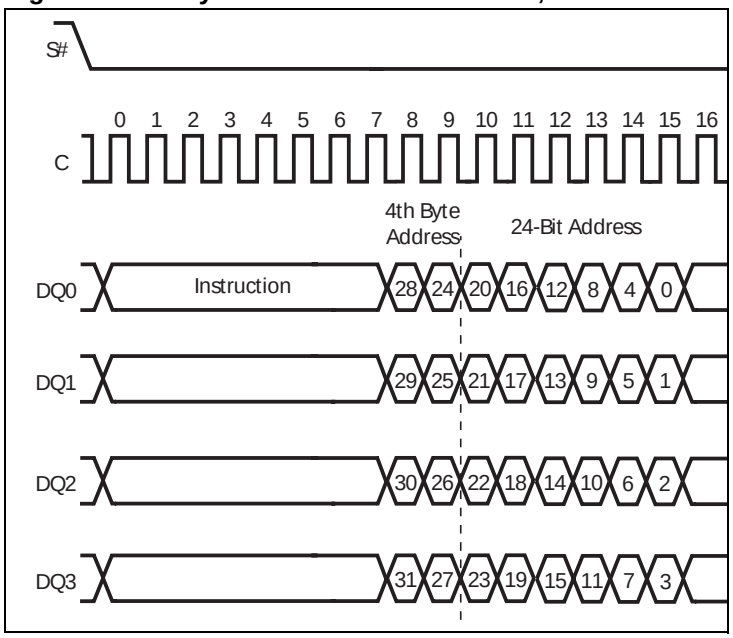
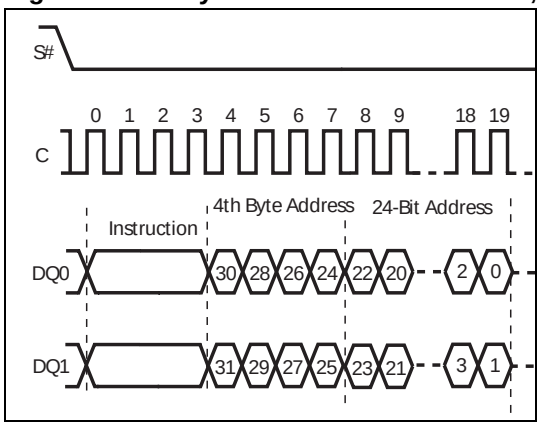


Figure 10. 4-Byte Address on Four Wires, Extended SPI



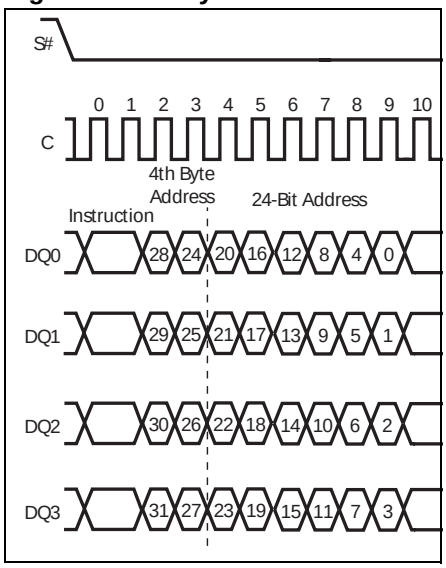
The following figure shows a Dual SPI Read instruction.

Figure 11. 4-Byte Address on Two Wires, Dual SPI, Example



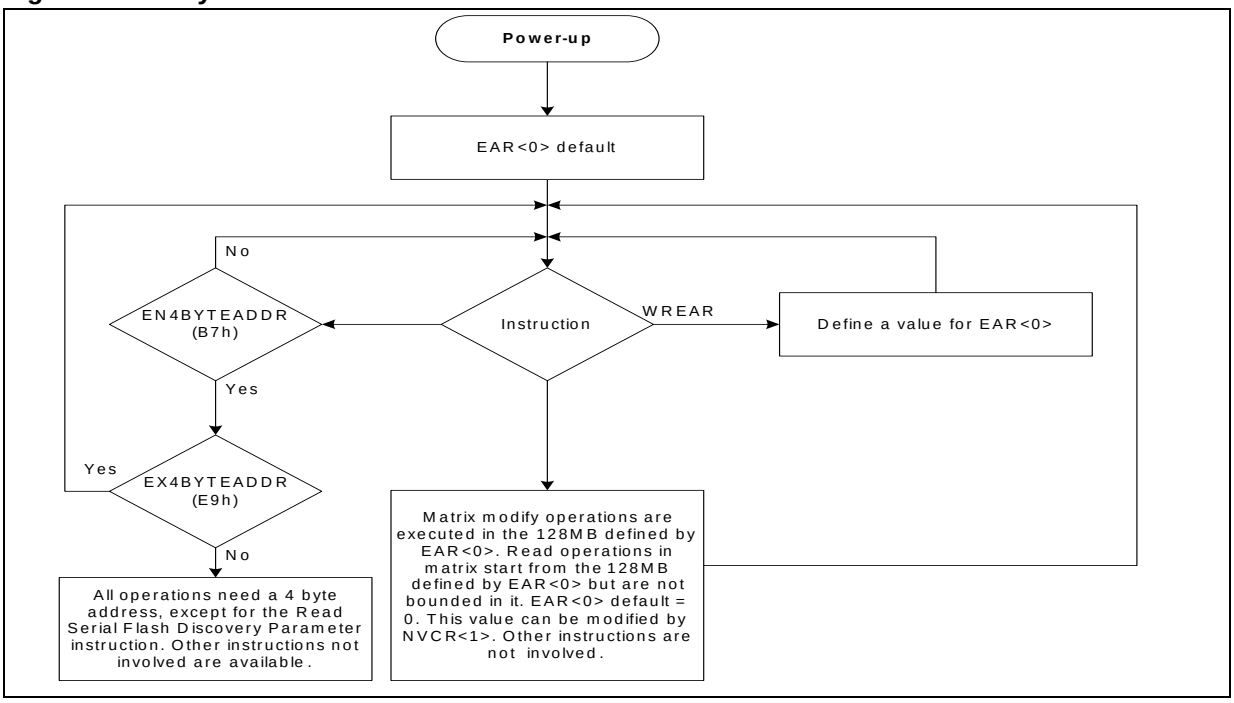
The following figure shows a Quad SPI Read instruction.

Figure 12. 4-Byte Address on Four Wires, Quad SPI Example



The default address mode is segmentation (3-Byte address mode). The mode can be changed by setting the NVCR bit.

Figure 13. 4-Byte Address Mode



Note: WREN operation is implied in EN4BYTEADDR, EX4BYTEADDR, and WREAR; to check the address mode enabled, a bit in FSR is enabled.

5.1.3 Instructions Not Affected by Addressing Mode

Also available are a set of dedicated array reading instructions that always require 4 byte address regardless active addressing mode. See *Table 16.: Instruction set: extended SPI protocol*.

5.2 Extended SPI Protocol Operating features

5.2.1 Read Operations

To read the memory content in Extended SPI protocol different instructions are available: READ, Fast Read, Dual Output Fast Read, Dual Input Output Fast Read, Quad Output Fast Read and Quad Input Output Fast read, allowing the application to choose an instruction to send addresses and receive data by one, two or four data lines.

Note: In the Extended SPI protocol the instruction code is always sent on one data line (DQ0): to use two or four data lines the user must use either the DIO-SPI or the QIO-SPI protocol respectively.

For fast read instructions the number of dummy clock cycles is configurable by using VCR bits [7:4] or NVCR bits [15:12].

After a successful reading instruction a reduced tSHSL equal to 20 ns is allowed to further improve random access time (in all the other cases tSHSL should be at least 50 ns). See *Table 34.: AC Characteristics*.

5.2.2 Page programming

To program one data byte, two instructions are required: write enable (WREN), which is one byte, and a page program (PP) sequence, which consists of four bytes plus data. This is followed by the internal program cycle (of duration t_{PP}).

To spread this overhead, the page program (PP) instruction allows up to 256 bytes to be programmed at a time (changing bits from ‘1’ to ‘0’), provided that they lie in consecutive addresses on the same page of memory.

For optimized timings, it is recommended to use the page program (PP) instruction to program all consecutive targeted bytes in a single sequence versus using several page program (PP) sequences with each containing only a few bytes (see *Section 5.3.3: Page programming* and *Table 34: AC Characteristics*).

5.2.3 Dual Input Fast Program

The dual input fast program (DIFP) instruction makes it possible to program up to 256 bytes using two input pins at the same time (by changing bits from ‘1’ to ‘0’).

For optimized timings, it is recommended to use the DIFP instruction to program all consecutive targeted bytes in a single sequence rather using several DIFP sequences each containing only a few bytes (see *Section 9.1.19: Dual Input Fast Program (DIFP)*).

5.2.4 Dual Input Extended Fast Program

The Dual Input Extended Fast Program (DIEFP) instruction is an enhanced version of the Dual Input Fast Program instruction, allowing to transmit address across two data lines.

For optimized timings, it is recommended to use the DIEFP instruction to program all consecutive targeted bytes in a single sequence rather than using several DIEFP sequences, each containing only a few bytes.

5.2.5 Quad Input Fast Program

The Quad Input Fast Program (QIFP) instruction makes it possible to program up to 256 bytes using 4 input pins at the same time (by changing bits from 1 to 0).

For optimized timings, it is recommended to use the QIFP instruction to program all consecutive targeted bytes in a single sequence rather than using several QIFP sequences each containing only a few bytes.

5.2.6 Quad Input Extended Fast Program

The Quad Input Extended Fast Program (QIEFP) instruction is an enhanced version of the Quad Input Fast Program instruction, allowing parallel input on the 4 input pins, including the address being sent to the device.

For optimized timings, it is recommended to use the QIEFP instruction to program all consecutive targeted bytes in a single sequence rather than using several QIEFP sequences each containing only a few bytes.

5.2.7 Subsector erase, sector erase and bulk erase

The page program (PP) instruction allows bits to be reset from ‘1’ to’0’. In order to do this the bytes of memory need to be erased to all 1s (FFh).

This can be achieved as follows:

- a subsector at a time, using the subsector erase (SSE) instruction. See *Section 16: Ordering information*;
- a sector at a time, using the sector erase (SE) instruction;
- throughout the entire memory, using the bulk erase (BE) instruction.

This starts an internal erase cycle (of duration t_{SSE}, t_{SE} or t_{BE}). The erase instruction must be preceded by a write enable (WREN) instruction.

5.2.8 Polling during a write, program, or erase cycle

A further improvement in the time to Write Status Register (WRSR), POTP, PP, DIFP,DIEFP,QIFP, QIEFP or Erase (SSE, SE or BE) can be achieved by not waiting for the worst case delay (tW, tPP, tSSE, tSE, or tBE). The application program can monitor if the required internal operation is completed, by polling the dedicated register bits to establish when the previous Write, Program or Erase cycle is complete.

The information on the memory being in progress for a Program, Erase, or Write instruction can be checked either on the Write In Progress (WIP) bit of the Status Register or in the Program/Erase Controller bit of the Flag Status Register.

Note: The Program/Erase Controller bit is the opposite state of the WIP bit in the Status Register.

In the Flag Status Register additional information can be checked, as eventual Program/Erase failures by mean of the Program or erase Error bits.

5.2.9 Active power and standby power modes

When Chip Select ($\overline{S}$) is Low, the device is selected, and in the active power mode.

When Chip Select ($\overline{S}$) is High, the device is deselected, but could remain in the active power mode until all internal cycles have completed (program, erase, write status register). The device then goes in to the standby power mode. The device consumption drops to I_{CC1}.

5.2.10 Hold (or Reset) condition

The Hold ($\overline{HOLD}$) signal is used to pause serial communications with the device without resetting the clocking sequence. However, taking this signal Low does not terminate any write status register, program or erase cycle that is currently in progress.

To enter the hold condition, the device must be selected, with Chip Select ($\overline{S}$) Low.

The hold condition starts on the falling edge of the Hold ($\overline{HOLD}$) signal, provided that the Serial Clock (C) is Low (as shown in *Figure 14*).

The hold condition ends on the rising edge of the Hold ($\overline{HOLD}$) signal, provided that the Serial Clock (C) is Low.

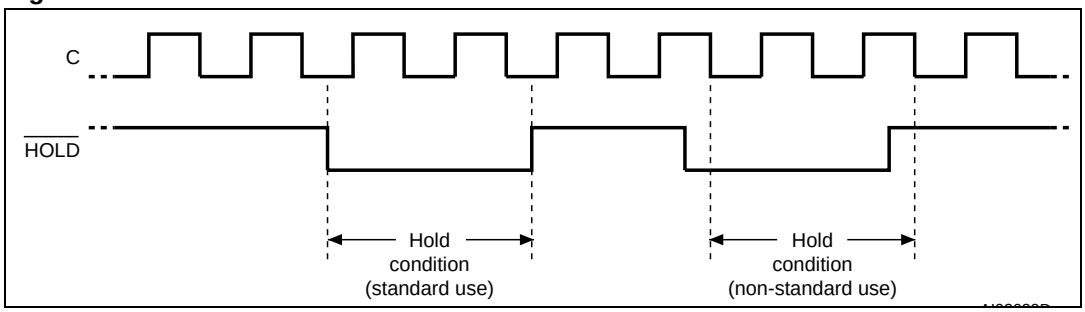
If the falling edge does not coincide with Serial Clock (C) being Low, the hold condition starts after Serial Clock (C) next goes Low. Similarly, if the rising edge does not coincide with Serial Clock (C) being Low, the hold condition ends after Serial Clock (C) next goes Low (this is shown in *Figure 14*).

During the hold condition, the serial data output (DQ1) is high impedance, and serial data input (DQ0) and Serial Clock (C) are don't care.

Normally, the device is kept selected, with Chip Select ($\overline{S}$) driven Low for the whole duration of the hold condition. This is to ensure that the state of the internal logic remains unchanged from the moment of entering the hold condition.

If Chip Select ($\overline{S}$) goes High while the device is in the Hold condition, this has the effect of resetting the internal logic of the device. To restart communication with the device, it is necessary to drive Hold ($\overline{HOLD}$) High, and then to drive Chip Select ($\overline{S}$) Low. This prevents the device from going back to the hold condition.

Figure 14. Hold condition activation



Reset functionality is available instead of Hold in parts with a dedicated part number. See *Section 16: Ordering information*.

Driving Reset ($\overline{Reset}$) Low while an internal operation is in progress (or suspended) will affect this operation (write, program or erase cycle) and data may be lost. On $\overline{Reset}$ going Low, the device enters the reset mode and a time of tRHSL is then required before the device can be reselected by driving Chip Select ($\overline{S}$) Low. For the value of tRHSL, see *Table*

34.: AC Characteristics. All the lock bits and volatile configuration registers are reset Power On Reset default condition after a Reset Low pulse. Power On Reset condition will depend on Non Volatile configuration register content.

The $\overline{\text{Hold}}$/$\overline{\text{Reset}}$ feature is not available when the $\overline{\text{Hold}}$ ($\overline{\text{Reset}}$) / DQ3 pin is used as I/O (DQ3 functionality) during Quad Instructions: QOFR, QIOFR,QIFP and QIEFP.

The $\overline{\text{Hold}}$/$\overline{\text{Reset}}$ feature can be disabled by using of the bit 4 of the VECR.

5.3 Dual SPI (DIO-SPI) Protocol

In the Dual SPI (DIO-SPI) protocol all the instructions, addresses and I/O data are transmitted on two data lines. All the functionality available in the Extended SPI protocol is also available in the DIO-SPI protocol. The DIO-SPI instructions are comparable with the Extended SPI instructions; however, in DIO-SPI, the instructions are multiplexed on the two data lines, DQ0 and DQ1.

The only exceptions are the READ, Quad Read, and Program instructions, which are not available in DIO-SPI protocol, and the RDID instruction, which is replaced in the DIO-SPI protocol by the Multiple I/O Read Identification (MIORDID) instruction.

The Multiple I/O Read Identification Instruction reads just the standard SPI electronic ID (3 bytes), while the Extended SPI protocol RDID instruction allows access to the UID bytes.

To help the application code port from Extended SPI to DIO-SPI protocol, the instructions available in the DIO-SPI protocol have the same operation code as the Extended SPI protocol, the only exception being the MIORDID instruction.

5.3.1 Multiple Read Identification

The Multiple I/O Read Identification (MIORDID) instruction is available to read the device electronic ID.With respect to the RDID instruction of the Extended SPI protocol, the output data, shifted out on the 2 data lines DQ0 and DQ1.

Since the read ID instruction in the DIO-SPI protocol is limited to 3 bytes of the standard electronic ID, the UID bytes are not read with the MIORDID instruction

5.3.2 Dual Command Fast reading

Reading the memory data multiplexing the instruction, the addresses and the output data on 2 data lines can be achieved in DIO-SPI protocol by mean of the Dual Command Fast Read instruction, that has 3 instruction codes (BBh, 3Bh and 0Bh) to help the application code porting from Extended SPI protocol to DIO-SPI protocol. Of course quad and single I/O Read instructions are not available in DIO-SPI mode.

For Dual Command fast read instructions the number of dummy clock cycles is configurable by using VCR bits [7:4] or NVCR bits [15:12].

After a successful reading instruction, a reduced tSHSL equal to 20ns is allowed to further improve random access time (in all the other cases tSHSL should be at least 50 ns). See *Table 34.: AC Characteristics*.

5.3.3 Page programming

Programming the memory by transmitting the instruction, addresses and the output data on 2 data lines can be achieved in DIO-SPI protocol by using the Dual Command Page

Program instruction, that has 3 instruction codes (D2h, A2h and 02h) to help port from Extended SPI protocol to DIO-SPI protocol

Quad and single input Program instructions are not available in DIO-SPI mode.

The DIO-SPI protocol is similar to the Extended SPI protocol i.e., to program one data byte two instructions are required:

- Write Enable (WREN), which is one byte, and a
- Dual Command Page Program (DCPP) sequence, which consists of four bytes plus data.

This is followed by the internal Program cycle (of duration tPP).

To spread this overhead, the Dual Command Page Program (DCPP) instruction allows up to 256 bytes to be programmed at a time (changing bits from 1 to 0), provided that they are consecutive addresses on the same page of memory.

For optimized timings, it is recommended to use the DCPP instruction to program all consecutive targeted bytes in a single sequence versus using several DCPP sequences with each containing only a few bytes. See *Table 34.: AC Characteristics*.

5.3.4 Subsector Erase, Sector Erase, and Bulk Erase

Similar to the Extended SPI protocol, in the DIO-SPI protocol to erase the memory bytes to all 1s (FFh) the Subsector Erase (SSE), the Sector Erase (SE) and the Bulk Erase (BE) instructions are available. These instructions start an internal Erase cycle (of duration tSSE, tSE or tBE).

The Erase instruction must be preceded by a Write Enable (WREN) instruction.

5.3.5 Polling during a Write, Program, or Erase cycle

Similar to the Extended SPI protocol, in the DIO-SPI protocol it is possible to monitor if the internal write, program or erase operation is completed, by polling the dedicated register bits by using the Read Status Register (RDSR) or Read Flag Status Register (RFSR) instructions, the only obvious difference is that instruction codes, addresses and output data are transmitted across two data lines.

5.3.6 Read and Modify registers

Similar to the Extended SPI protocol, the only obvious difference is that instruction codes, addresses and output data are transmitted across two data lines

5.3.7 Active Power and Standby Power modes

Similar to the Extended SPI protocol, when Chip Select ($\overline{S}$) is Low, the device is selected, and in the Active Power mode. When Chip Select ($\overline{S}$) is High, the device is deselected, but could remain in the Active Power mode until all internal cycles have completed (Program, Erase, Write Cycles). The device then goes in to the Standby Power mode. The device consumption drops to ICC1.

5.3.8 HOLD (or Reset) condition

The $\overline{HOLD}$ (or $\overline{Reset}$ i.e. for parts having the reset functionality instead of $\overline{hold}$ pin) signal has exactly the same behavior in DIO-SPI protocol as do in Extended SPI protocol, so

please refer to section 5.1.10, $\overline{\text{Hold}}$ (or $\overline{\text{Reset}}$) condition" in the Extend SPI protocol section for further details.

5.4 Quad SPI (QIO-SPI)Protocol

In the Quad SPI (QIO-SPI) protocol all the Instructions, addresses and I/O data are transmitted on four data lines, with the exception of the polling instructions performed during a Program or Erase cycle performed with VPP, in this case the device temporarily goes in Extended SPI protocol. The protocol again becomes QIO-SPI as soon as the VPP voltage goes low.

All the functionality available in the Extended SPI protocol are also available in the QIO-SPI mode, with equivalent instruction transmitted on the 4 data lines DQ0, DQ1, DQ2 and DQ3. The exceptions are the READ, Dual Read and Dual Program instructions, that are not available in QIO-SPI protocol, and the RDID instruction, that is replaced in the QIO-SPI protocol by the Multiple I/O Read Identification (MIORDID) instruction. The Multiple I/O Read Instruction reads just the standard SPI electronic ID (3 bytes), while with the Extended SPI protocol RDID instruction is possible to access also the UID bytes.

To help the application code port from Extended SPI to QIO-SPI protocol, the instructions available in the QIO-SPI protocol have the same operation code as in the Extended SPI protocol, the only exception is the MIORDID instruction.

5.4.1 Multiple Read Identification

The Multiple I/O Read Identification (MIORDID) instruction is available to read the device electronic ID. With respect to the RDID instruction of the Extended SPI protocol, the output data, shifted out on the 4 data lines DQ0, DQ1, DQ2 and DQ3.

Since in the QIO-SPI protocol the Read ID instruction is limited to 3 bytes of the standard electronic ID, the UID bytes are not read with the MIORDID instruction.

5.4.2 Quad Command Fast reading

The Array Data can be read by the Quad Command Fast Read instruction using 3 instructions (EBh, 6Bh and 0Bh) to help the application code port from Extended SPI protocol to DIO-SPI protocol. The instruction, address and output data are transmitted across 4 data lines.

The Dual and Single I/O Read instructions are not available in QIO-SPI protocol.

5.4.3 QUAD Command Page programming

The memory can be programmed in QIO-SPI protocol by the Quad Command Page Program instruction using (02h, 12h and 32h). The instruction, address and input data are transmitted across 4 data lines.

The Dual and Single I/O Program instructions are not available in QIO-SPI protocol.

Programming the memory by multiplexing the instruction, the addresses and the output data on 4 wires can be achieved in QIO-SPI protocol by mean of the Quad Command Page Program instruction, that has 3 instruction codes (02h, 12h and 32h) to help the application code porting from Extended SPI protocol to QIO-SPI protocol.

Similar to the Extended SPI protocol in the QIO-SPI protocol, to program one data byte two instructions are required:

- Write Enable (WREN), which is one byte, and
- Quad Command Page Program (QCPP) sequence, which consists of instruction (one byte), address (3 bytes) and input data.

This is followed by the internal Program cycle (of duration tPP).

To spread this overhead, the Quad Command Page Program (QCPP) instruction allows up to 256 bytes to be programmed at a time (changing bits from 1 to 0), provided that they are in consecutive addresses on the same page of memory.

For optimized timings, it is recommended to use the QCPP instruction to program all consecutive targeted bytes in a single sequence versus using several QCPP sequences with each containing only a few bytes. See *Table 34.: AC Characteristics*.

The QCPP instruction is transmitted across 4 data lines except when VPP is raised to VPPH.

The VPP can be raised to VPPH to decrease programming time (provided that the bit 3 of the VECR has been set to 0 in advance). When bit 3 of VECR is set to 0 after the Quad Command Page Program instruction sequence has been received, the memory temporarily goes in Extended SPI protocol, and is possible to perform polling instructions (checking the WIP bit of the Status Register or the Program/Erase Controller bit of the Flag Status Register) or Program/Erase Suspend instruction even if DQ2 is temporarily used in this VPP functionality. The memory automatically comes back in QIO-SPI protocol as soon as the VPP pin goes Low.

5.4.4 Subsector Erase, Sector Erase, and Bulk Erase

Similar to the Extended SPI protocol, Subsector Erase (SSE), the Sector Erase (SE) and the Bulk Erase (BE) instructions are used to erase the memory in the QIO-SPI protocol. These instructions start an internal Erase cycle (of duration tSSE, tSE or tBE).

The Erase instruction must be preceded by a Write Enable (WREN) instruction.

The erase instructions are transmitted across 4 data lines unless the VPP is raised to VPPH.

The VPP can be raised to VPPH to decrease erasing time, provided that the bit 3 of the VECR has been set to 0 in advance. In this case, after the erase instruction sequence has been received, the memory temporarily goes in extended SPI protocol, and it is possible to perform polling instructions (checking the WIP bit of the Status Register or the Program/Erase Controller bit of the Flag Status Register) or Program/Erase Suspend instruction even if DQ2 is temporarily used in this VPP functionality. The memory automatically comes back in QIO-SPI protocol as soon as the VPP pin goes Low.

5.4.5 Polling during a Write, Program, or Erase cycle

It is possible to check if the internal write, program or erase operation is completed, by polling the dedicated register bits of the Read Status Register (RDSR) or Read Flag Status Register (FSR).

When the Program or Erase cycle is performed with the VPP, the device temporarily goes in single I/O SPI mode. The protocol became again QIO-SPI as soon as the VPP pin voltage goes low.

5.4.6 Read and Modify registers

The read and modify register instructions are available and behave in QIO-SPI protocol exactly as they do in Extended SPI protocol, the only difference is that instruction codes, addresses and output data are transmitted across 4 data lines.

5.4.7 Active Power and Standby Power modes

Exactly as in Extended SPI protocol, when Chip Select ($\overline{S}$) is Low, the device is selected, and in the Active Power mode. When Chip Select ($\overline{S}$) is High, the device is deselected, but could remain in the Active Power mode until all internal (Program, Erase, Write) Cycles have completed. The device then goes in to the Standby Power mode. The device consumption drops to ICC1.

5.4.8 HOLD (or Reset) condition

The HOLD ($\overline{Hold}$) feature (or Reset feature, for parts having the reset functionality instead of hold) is disabled in QIO-SPI protocol when the device is selected: the $\overline{Hold}$ (or $\overline{Reset}$)/ DQ3 pin always behaves as an I/O pin (DQ3 function) when the device is deselected. For parts with reset functionality, it is still possible to reset the memory when it is deselected (C signal high).

5.4.9 VPP pin Enhanced Supply Voltage feature

It is possible in the QIO-SPI protocol to use the VPP pin as an enhanced supply voltage, but the intention to use VPP as accelerated supply voltage must be declared by setting bit 3 of the VECR to 0.

In this case, to accelerate the Program cycle the VPP pin must be raised to VPPH after the device has received the last data to be programmed within 200ms. If the VPP is not raised within 200ms, the program operation starts with the standard internal cycle speed as if the Vpp high voltage were not used, and a flag error appears on Flag Status Register bit 3".

6 Volatile and Non Volatile Registers

The device features many different registers to store, in volatile or non volatile mode, many parameters and operating configurations:

- Legacy SPI Status Register
- 3 configuration registers:
  - Non Volatile Configuration Register (NVCR), 16 bits
  - Volatile Configuration Register (VCR), 8 bits
  - Volatile Enhanced Configuration Register (VECR), 8 bits

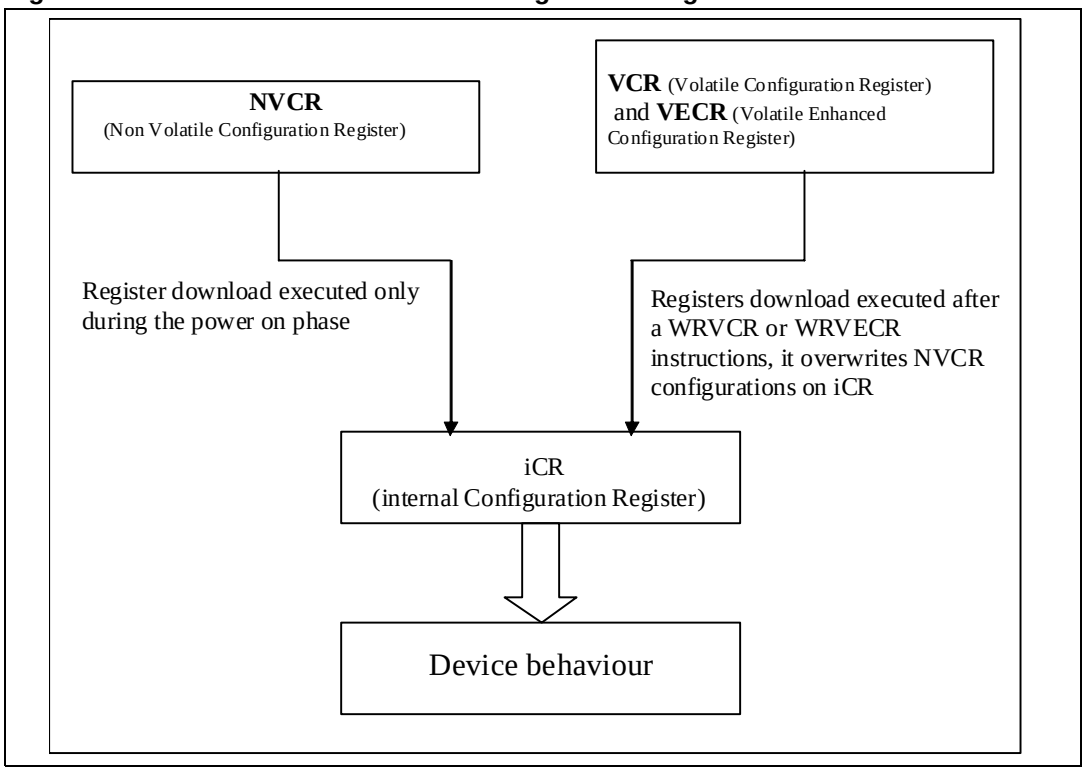
The Non Volatile Configuration Register (NVCR) affects the memory configuration starting from the successive power-on. It can be used to make the memory start in a determined condition.

The VCR and VECR affect the memory configuration after every execution of the related Write Volatile configuration Register (WRVCR) and Write Enhanced Volatile Configuration register (WRVECR) instructions. These instructions overwrite the memory configuration set at POR by NVCR.

As described in *Figure 15.: Non Volatile and Volatile configuration Register Scheme*, the working condition of the memory is set by an internal configuration register, which is not accessible by the user. The working parameters of the internal configuration register are loaded from the NVCR during the boot phase of the device. In this sense the NVCR can be seen as having the default settings of the memory.

During the normal life of the application, every time a write volatile or enhanced volatile configuration register instruction is performed, the new configuration parameters set in the volatile registers are also copied in the internal configuration register, thus instantly affecting the memory behavior. Please note that on the next power on the memory will start again in the working protocol set by the Non Volatile Register parameters.

Figure 15. Non Volatile and Volatile configuration Register Scheme



A Flag Status Register (FSR), 8 bits, is also available to check the status of the device, detecting possible errors or a Program/Erase internal cycle in progress.

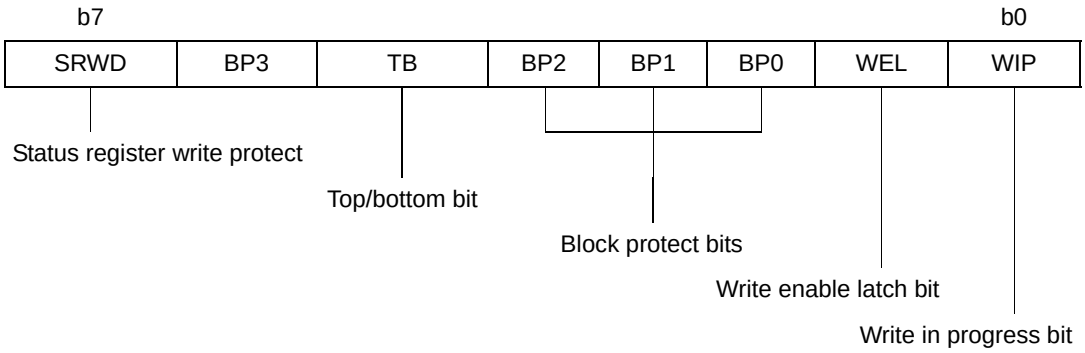
Each register can be read and modified by means of dedicated instructions in all the 3 protocols (Extended SPI, DIO-SPI, and QIO-SPI).

Reading time for all registers is comparable; writing time instead is very different: NVCR bits are set as Flash Cell memory content requiring a longer time to perform internal writing cycles. See *Table 34.: AC Characteristics*.

6.1 Legacy SPI Status Register

The Status Register contains a number of status and control bits that can be read or set by specific instructions: Read Status Register (RDSR) and Write Status Register (WRSR). This is available in all the 3 protocols (Extended SPI, DIO-SPI, and QIO-SPI).

Table 3. Status register format



6.1.1 WIP bit

The Write In Progress (WIP) bit set to 1 indicates that the memory is busy with a Write Status Register, Program or Erase cycle. 0 indicates no cycle is in progress.

6.1.2 WEL bit

The Write Enable Latch (WEL) bit set to 1 indicates that the internal Write Enable Latch is set. When set to 0 the internal Write Enable Latch is reset and no Write Status Register, Program or Erase instruction is accepted.

6.1.3 BP3, BP2, BP1, BP0 bits

The Block Protect (BP3, BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against Program and Erase instructions. These bits are written with the Write Status Register (WRSR) instruction. When one or more of the Block Protect (BP3, BP2, BP1, BP0) bits is set to 1, the relevant memory area, as defined in *Table 11.: Protected area sizes, Upper (TB bit = 0)* becomes protected against all program and erase instructions. The Block Protect (BP3, BP2, BP1, BP0) bits can be written provided that the Hardware Protected mode has not been set. The Bulk Erase (BE) instruction is executed if, and only if, all Block Protect (BP3, BP2, BP1, BP0) bits are 0.

6.1.4 TB bit

The Top/Bottom (TB) bit is non-volatile. It can be set and reset with the Write Status Register (WRSR) instruction provided that the Write Enable (WREN) instruction has been issued.

The Top/Bottom (TB) bit is used in conjunction with the Block Protect (BP3, BP2, BP1, BP0) bits to determine if the protected area defined by the Block Protect bits starts from the top or the bottom of the memory array:

- When TB is reset to '0' (default value), the area protected by the Block Protect bits starts from the top of the memory array.
- When TB is set to '1', the area protected by the Block Protect bits starts from the bottom of the memory array.

The TB bit cannot be written when the SRWD bit is set to '1' and the $\overline{W}$ pin is driven Low.

6.1.5 SRWD bit

The Status Register Write Disable (SRWD) bit is operated in conjunction with the Write Protect ($\overline{W}$/VPP) signal. The Status Register Write Disable (SRWD) bit and the Write Protect ($\overline{W}$/VPP) signal allow the device to be put in the hardware protected mode (when the Status Register Write Disable (SRWD) bit is set to '1', and Write Protect (($\overline{W}$/VPP) is driven Low). In this mode, the non-volatile bits of the Status Register (TB, BP3, BP2, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

6.2 Non Volatile Configuration Register

The Non Volatile Configuration Register (NVCR) bits affects the default memory configuration after power-on. It can be used to make the memory start in the configuration to fit the application requirements.

The device is delivered with Non Volatile Configuration Register (NVCR) bits all erased to 1 (FFFFh).

The purpose of the NVCR is to define the default memory settings after the power-on sequence related to many features:

- The number of dummy clock cycle for fast read instructions,
- XIP mode configurations,
- output driver strengths,
- Reset (or Hold) disabling
- Multiple I/O protocol enabling.

The NVCR can be read by the Read Non Volatile Configuration Register (RDNVCR) instruction and written by the Write Non Volatile Configuration Register (WRNVCR) in all the 3 available SPI protocols. See the sections that follow as well as *Table 4.: Non-Volatile Configuration Register*.

Table 4. Non-Volatile Configuration Register

Bit	Parameter	Value	Description	Note
NVCR<15:12>	Dummy clock cycle	0000	As '1111'	To optimize instruction execution (FASTREAD, DOFR,DIOFR,QOFR, QIOFR, ROTP) according to the frequency
		0001	1	
		0010	2	
		0011	3	
		0100	4	
		0101	5	
		0110	6	
		0111	7	
		1000	8	
		1001	9	
		1010	10	
		1011	11	
		1100	12	
		1101	13	
		1110	14	
		1111	Target on maximum allowed frequency fc (108MHz) and to guarantee backward compatibility (default)	
NVCR<11:9>	XIP enabling at POR	000	XIP for SIO Read	
		001	XIP for DOFR	
		010	XIP for DIOFR	
		011	XIP for QOFR	
		100	XIP for QIOFR	
		101	reserved	
		110	reserved	
		111	XIP disabled (default)	
NVCR<8:6>	Output Driver Strength	000	reserved	Impedance at Vcc/2
		001	90	
		010	60	
		011	45	
		100	reserved	
		101	20	
		110	15	
		111	30 (default)	
NVCR<5>	Reserved	X	Don't Care	Default value = 1

Table 4. Non-Volatile Configuration Register

Bit	Parameter	Value	Description	Note
NVCR<4>	Reset/Hold disable	0	disabled	Disable Pad Hold/Reset functionality
		1	enabled (default)	
NVCR<3>	Quad Input Command	0	enabled	Enable command on four input line
		1	disabled (default)	
NVCR<2>	Dual Input Command	0	enabled	Enable command on two input line
		1	disabled (default)	
NVCR<1>	128Mb selection	0	Upper 128Mb segment	Define the 128Mb segment enabled for 3-Byte operation
		1	Bottom 128Mb segment	
NVCR<0>	Address mode selection	0	4-Byte address mode	Define the number of address bytes for memory instructions
		1	3-Byte address mode (default)	

6.2.1 Dummy clock cycle NV configuration bits (NVCR bits from 15 to 12)

The bits from 15 to 12 of the Non Volatile Configuration register store the default settings for the dummy clock cycles number after the fast read instructions (in all the 3 available protocols). The dummy clock cycles number can be set from 1 up to 15 as described here, according to operating frequency (the higher is the operating frequency, the bigger must be the dummy clock cycle number) to optimize the fast read instructions performance.

The default values of these bits allow the memory to be safely used with fast read instructions at the maximum frequency (108 MHz). Please note that if the dummy clock number is not sufficient for the operating frequency, the memory reads wrong data.

Table 5. Maximum operative frequency by dummy clock cycles

Maximum allowed frequency (MHz) (1)					
Dummy Clock	**FASTREAD**	**DOFR**	**DIOFR**	**QOFR**	**QIOFR**
1	54	50	39	43	20
2	95	85	59	56	39
3	105	95	75	70	49
4	108	105	88	83	59
5	108	108	94	94	69
6	108	108	105	105	78
7	108	108	108	108	86
8	108	108	108	108	95
9	108	108	108	108	105
10	108	108	108	108	108

1. All the values are guaranteed by characterization and not 100% tested in production

6.2.2 XIP NV configuration bits (NVCR bits from 11 to 9)

The bits from 11 to 9 of the Non Volatile Configuration register store the default settings for the XIP operation, allowing the memory to start working directly on the required XIP mode after successive POR sequence: the device then accepts only address on one, two, or four wires (skipping the instruction) depending on the NVCR XIP bits settings.

The default settings for the XIP bits of the NVCR enable the memory to start working in Extended SPI mode after the POR sequence (XIP directly after POR is disabled).

6.2.3 Output Driver Strength NV configuration bits (NVCR bits from 8 to 6)

The bits from 8 to 6 of the Non Volatile Configuration register store the default settings for the output driver strength, enabling to optimize the impedance at Vcc/2 output voltage for the specific application.

The default values of Output Driver Strength bits of the NVCR set the output impedance at Vcc/2 equal to 30 Ohms.

6.2.4 Hold (Reset) disable NV configuration bit (NVCR bit 4)

The Hold (RESET) disable bit can be used to disable the Hold (Reset) functionality of the Hold (Reset) / DQ3 pin as described in *Table 4.: Non-Volatile Configuration Register*. This feature can be useful to avoid accidental Hold or Reset condition entries in applications that never require the Hold (Reset) functionality.

The default values of Hold (Reset) bit of the NVCR is set to enable the Hold (Reset) functionality.

Note: *Reset functionality is available instead of Hold* in devices with a dedicated part number. *See Section 16: Ordering information.*

6.2.5 Quad Input NV configuration bit (NVCR bit 3)

The Quad Input NV configuration bit can be used to make the memory start working in QIO-SPI protocol directly after the power on sequence. The products are delivered with this set to 1, making the memory default in Extended SPI protocol, if the application sets this bit to 0 the device will enter in QIO-SPI protocol right after the next power on.

Please note that in case both QIO-SPI and DIO-SPI are enabled (both bit 3 and bit 2 of the Non Volatile Configuration Register set to 0), the memory will work in QIO-SPI.

6.2.6 Dual Input NV configuration bit (NVCR bit 2)

The Dual Input NV configuration bit can be used to make the memory start working in DIO-SPI protocol directly after the power on sequence. The products are delivered with this set to 1, making the memory default in Extended SPI protocol, if the application sets this bit to 0 the device will enter in QIO-SPI protocol right after the next power on.

Please note that in case both QIO-SPI and DIO-SPI are enabled (both bit 3 and bit 2 of the Non Volatile Configuration Register set to 0), the memory will work in QIO-SPI.

6.2.7 128Mb segment selection NV configuration bit (NVCR bit 1)

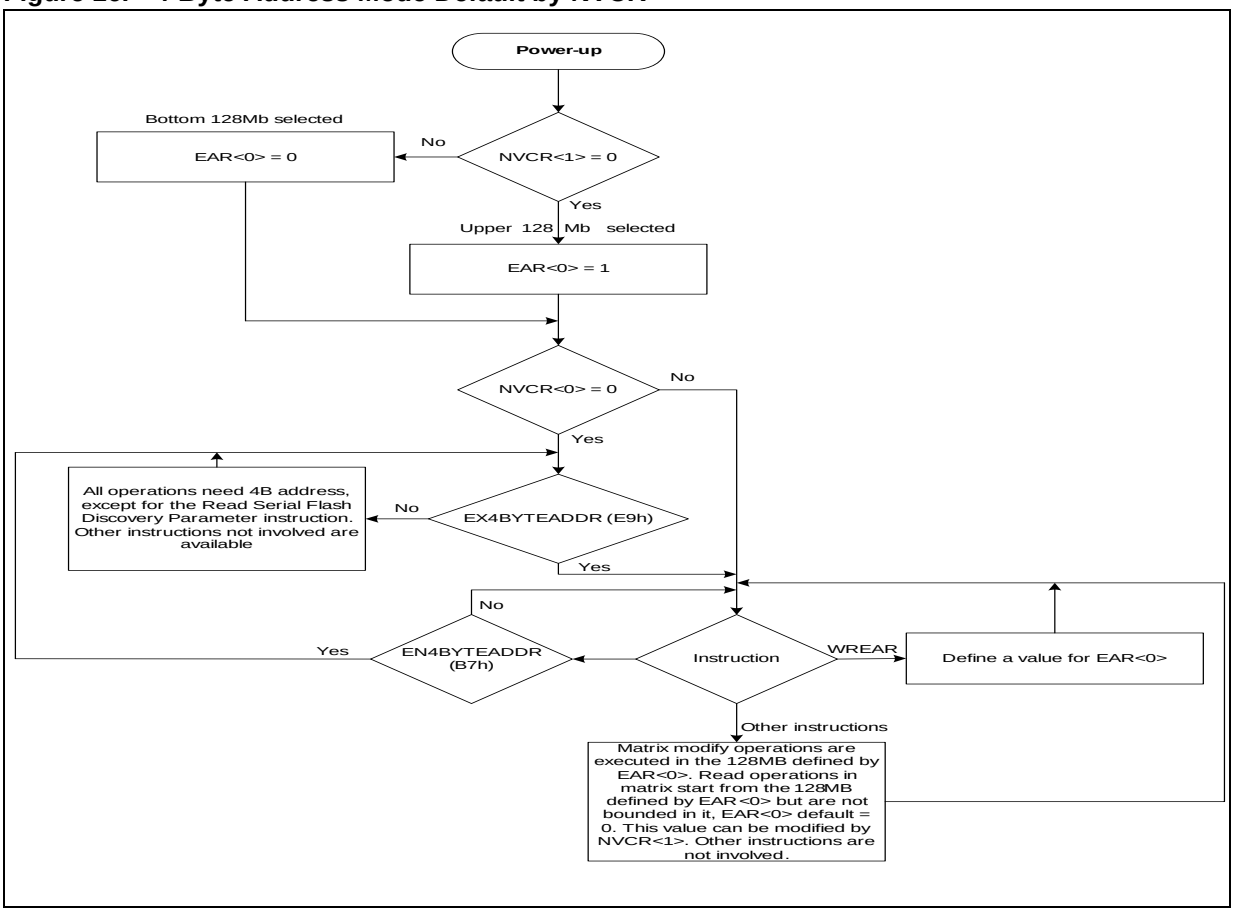
The 128Mb segment selection NV configuration bit can be used to set the segment to address directly after the power on sequence. The products are delivered with this set to 1,

making segment default in bottom 128Mb segment, if the application sets this bit to 0 the selected segment will be the upper 128Mb right after the next power on.

6.2.8 Address mode selection NV configuration bit (NVCR bit 0)

The address mode selection NV configuration bit can be used to set the addressing mode directly after the power on sequence. The products are delivered with this set to 1, making default address mode in 3-byte address mode, if the application sets this bit to 0 the address mode will be the 4-byte address mode right after the next power on. The figure here shows how NVCR<1:0> bits affect working diagram.

Figure 16. 4-Byte Address Mode Default by NVCR



6.3 Volatile Configuration Register

The Volatile Configuration Register (VCR) affects the memory configuration after every execution of Write Volatile Configuration Register (WRVCR) instruction: this instruction overwrite the memory configuration set at POR by the Non Volatile Configuration Register (NVCR). Its purpose is to define the dummy clock cycles number and to make the device ready to enter in the required XIP mode.

Table 6. Volatile Configuration Register

Bit	Parameter	Binary Value	Decimal Value	Description
VCR<7:4>	Dummy clock cycle (1)	0000	As '1111'	
		0001	1	
		0010	2	
		0011	3	
		0100	4	
		0101	5	
		0110	6	
		0111	7	
		1000	8	
		1001	9	
		1010	10	
		1011	11	
		1100	12	
		1101	13	
		1110	14	
		1111	15	Targets maximum allowed frequency fc (108MHz) and guarantees backward compatibility (default)
VCR<3>	XIP (2)	0	0	Ready to enter XIP mode
		1	1	XIP disabled (default)
VCR<2>	Reserved	x	reserved	Fixed value = 0b
VCR<1:0>	Wrap	00	0	16-byte wrap: Output data wraps within an aligned 16-byte boundary starting from the 3-byte address issued after the command code.
		01	1	32-byte wrap: Output data wraps within an aligned 32-byte boundary starting from the 3-byte address issued after the command code.
		10	2	64-byte wrap: Output data wraps within an aligned 64-byte boundary starting from the 3-byte address issued after the command code.
		11	3	Continous reading (Default): All bytes are read sequentially

1. To optimize instruction execution (FASTREAD, DOFR,DIOFR,QOFR, QIOFR, ROTP) according to the frequency.
2. To make the data on DQ0 during the first dummy clock NOT “Don’t Care.” For devices with feature set digit equal to 2 or 4 in the part number (Basic XiP), this bit is always Don't Care."

6.3.1 Dummy clock cycle: VCR bits 7:4

The bits from 7 to 4 of the Volatile Configuration Register, as the bits from 15 to 12 of the Volatile Configuration register, set the dummy clock cycles number after the fast read instructions (in all the 3 available protocols). The dummy clock cycles number can be set from 1 up to 15 as described in *Table 6.: Volatile Configuration Register*, according to operating frequency (the higher is the operating frequency, the bigger must be the dummy

clock cycle number, according to *Table 5.: Maximum operative frequency by dummy clock cycles*) to optimize the fast read instructions performance.

Note: If the dummy clock number is not sufficient for the operating frequency, the memory reads wrong data.

6.3.2 XIP Volatile Configuration bits (VCR bit 3)

The bit 3 of the Volatile Configuration Register is the XIP enabling bit, this bit must be set to 0 to enable the memory working on XIP mode. For devices with a feature set digit equal to 2 or 4 in the part number (Basic XiP), this bit is always Don't Care, and it is possible to operate the memory in XIP mode without setting it to 0. See *Section 16: Ordering information*.

6.3.3 Wrap: VCR bits 1:0

The Wrap mode provides the ability to read from memory sequentially ((standard mode); or to read from memory by confined 16-byte, 32-byte, or 64-byte boundaries. For Wrap setting options, see *Table 6.: Volatile Configuration Register*. The following table shows an example of the sequence of bytes in the 16-byte, 32-byte, and 64-byte options, according to the starting address.

Table 7. Sequence of Bytes Read during Wrap Mode

Starting Address	16-Byte Wrap	32-Byte Wrap	64-Byte Wrap
0	0-1-2- . . . -15-0-1- . .	0-1-2- . . . -31-0-1- . .	0-1-2- . . . -63-0-1- . .
1	1-2- . . . -15-0-1-2- . .	1-2- . . . -31-0-1-2- . .	1-2- . . . -63-0-1-2- . .
16	15-0-1-2-3- . . . -15-0-1- . .	15-16-17- . . . -31-0-1- . .	15-16-17- . . . -63-0-1- . .
31	31-16-17- . . . -31-16-17- . .	31-0-1-2-3- . . . -31-0-1- . .	31-32-33- . . . -63-0-1- . .
63	63-48-49- . . . -63-48-49- . .	63-32-33- . . . -63-32-33- . .	63-0-1- . . . -63-0-1- . .

6.4 Volatile Enhanced Configuration Register

The Volatile Enhanced Configuration Register (VECR) affects the memory configuration after every execution of Write Volatile Enhanced Configuration Register (WRVECR) instruction. This instruction overwrites the memory configuration set during the POR sequence by the Non Volatile Configuration Register (NVCR). Its purpose is enabling of QIO-SPI protocol and DIO-SPI protocol.

Warning: WARNING: in case of both QIO-SPI and DIO-SPI enabled, the memory works in QIO-SPI

- HOLD (Reset) functionality disabling
- To enable the VPP functionality in Quad I/O modify operations
- To define output driver strength (3 bit)

Table 8. Volatile Enhanced Configuration Register

Bit	Parameter	Value	Description	Note
VECR<7>	Quad Input Command	0	Enabled	Enable command on four input lines
		1	Disabled (default)	
VECR<6>	Dual Input Command	0	Enabled	Enable command on two input lines
		1	Disabled (default)	
VECR<5>	Reserved	x	Reserved	Fixed value = 0b
VECR<4>	Reset/Hold disable	0	Disabled	Disable Pad Hold/Reset functionality
		1	Enabled (default)	
VECR<3>	Accelerator pin enable in QIO-SPI protocol or in QIFP/QIEFP	0	Enabled	The bit must be considered in case of QIFP, QIEFP, or QIO-SPI protocol. It is “Don’t Care” otherwise.
		1	Disabled (default)	
VECR<2:0>	Output Driver Strength	000	reserved	Impedance at $V_{CC}/2$
		001	90	
		010	60	
		011	45	
		100	reserved	
		101	20	
		110	15	
		111	30 (default)	

6.4.1 Quad Input Command VECR<7>

The Quad Input Command configuration bit can be used to make the memory start working in QIO-SPI protocol directly after the Write Volatile Enhanced Configuration Register

(WRVECR) instruction. The default value of this bit is 1, corresponding to Extended SPI protocol, If this bit is set to 0 the memory works in QIO-SPI protocol. If VECR bit 7 is set back to 1 the memory start working again in Extended SPI protocol, unless the bit 6 is set to 0 (in this case the memory start working in DIO-SPI mode).

Please note that in case both QIO-SPI and DIO-SPI are enabled (both bit 7and bit 6 of the VECR set to 0), the memory will work in QIO-SPI.

6.4.2 Dual Input Command VECR<6>

The Dual Input Command configuration bit can be used to make the memory start working in DIO-SPI protocol directly after the Write Volatile Enhanced Configuration Register (WVECR) instruction. The default value of this bit is 1, corresponding to Extended SPI protocol, if this bit is set to 0 the memory works in DIO-SPI protocol (unless the Volatile Enhanced Configuration Register bit 7 is also set to 0). If the Volatile Enhanced Configuration Register bit 6 is set back to 1 the memory start working again in Extended SPI protocol.

Please note that in case both QIO-SPI and DIO-SPI are enabled (both bit 7 and bit 6 of the VECR are set to 0), the memory will work in QIO-SPI.

6.4.3 Reset/Hold disable VECR<4>

The Hold (RESET) disable bit can be used to disable the Hold (Reset) functionality of the Hold (Reset) / DQ3 pin right after the Write Volatile Enhanced Configuration Register (WVECR) instruction. This feature can be useful to avoid accidental Hold or Reset condition entries in applications that never require the Hold (Reset) functionality. If this bit is set to 0 the Hold (Reset) functionality is disabled, it is possible to enable it back by setting this bit to 1.

Please note that after the next power on the Hold (Reset) functionality will be enabled again unless the bit 4 of the Non Volatile Configuration Register is set to 0.

Note: Reset functionality is available instead of Hold in devices with a dedicated part number. See Section 16: Ordering information.

6.4.4 Accelerator pin enable: QIO-SPI protocol / QIFP/QIEFP VECR<3>

The bit 3 of the Volatile Enhanced Configuration Register determine whether is possible or not to use the Vpp accelerating voltage to speed up internal modify operation with Quad program and erase instructions (both in Extended or QIO-SPI protocols).

If we want to use the Vpp voltage with Quad I/O modify instructions, we must set previously this bit to 0 (his default value is 1, in this case the Vpp pin functionality is disabled in all Quad I/O operations: both in Extended SPI and QIO-SPI protocols).

If the Volatile Enhanced Configuration Register bit 3 is set to 0, using the QIO-SPI protocol, after a Quad Command Page Program instruction or an Erase instruction is received (with all input data in the Program case) and the memory is de-selected, the protocol temporarily switches to Extended SPI protocol until Vpp passes from Vpph to normal I/O value (this transition is mandatory to come back to QIO-SPI protocol), to enable the possibility to perform polling instructions (to check if the internal modify cycle is finished by means of the WIP bit of the Status Register or of the Program/Erase controller bit of the Flag Status register) or Program/Erase Suspend instruction even if the DQ2 pin is temporarily used in his Vpp functionality.

If the Volatile Enhanced Configuration Register bit 3 is set to 0, after any quad modify instruction (both in Extended SPI protocol and QIO-SPI protocol) there is a maximum allowed time-out of 200ms after the last instruction input is received and the memory is de-selected to raise the Vpp signal to Vpph, otherwise the modify instruction start at normal speed, without the Vpph enhancement, and a flag error appears on Flag Status Register bit 3.

6.4.5 Output Driver Strength VECR<2:0>

The bits from 2 to 0 of the VECR set the value of the output driver strength, enabling to optimize the impedance at Vcc/2 output voltage for the specific application as described in *Table 8.: Volatile Enhanced Configuration Register*.

The default values of Output Driver Strength is set by the dedicated bits of the Non Volatile Configuration Register (NVCR), the parts are delivered with the output impedance at Vcc/2 equal to 30 Ohms.

6.5 Flag Status Register

The Flag Status Register is a powerful tool to investigate the status of the device, checking information regarding what the memory is actually doing and detecting possible error conditions. The Flag Status Register is composed of 8 bits, which are read by the Read Status Register (RFSR) instruction.

Three bits provide a Status Indicator, and are set/reset automatically by the memory:

- Program/Erase Controller bit
- Erase Suspend bit
- Program Suspend bit)

Four bits are error Indicators bits, and are set by the memory when some program or erase operation fails or the user tries to perform a forbidden operation. The user can clear the error indicators bits by mean of the Clear Flag Status Register (CLFSR) instruction:

- Erase error bit
- Program error bit
- VPP 1 to 0 error bit
- Protection error bit

Table 9. Flag Status Register

BIT	Description	Note
7	P/E Controller (not WIP)	Status
6	Erase Suspend	Status
5	Erase	Error
4	Program	Error
3	VPP	Error
2	Program Suspend	Status
1	Protection	Error
0	4-byte Address Enabling	Status

6.5.1 P/E Controller Status bit

The bit 7 of the Flag Status register represents the Program/Erase Controller Status bit, It indicates whether there is a Program/Erase internal cycle active. When P/E Controller Status bit is Low (FSR<7>=0) the device is busy; when the bit is High (FSR<7>=1) the device is ready to process a new command.

This bit has the same meaning of Write In Progress (WIP) bit of the standard SPI Status Register, but with opposite logic: FSR<7> = not WIP

It's possible to make the polling instructions, to check if the internal modify operations are finished, both on the Flag Status register bit 7 or on WIP bit of the Status Register.

6.5.2 Erase Suspend Status bit

The bit 6 of the Flag Status register represents the Erase Suspend Status bit, It indicates that an Erase operation has been suspended or is going to be suspended.

The bit is set (FSR<6>=1) within the Erase Suspend Latency time, that is as soon as the Program/Erase Suspend command (PES) has been issued, therefore the device may still complete the operation before entering the Suspend Mode.

The Erase Suspend Status should be considered valid when the P/E Controller bit is high (FSR<7>=1).

When a Program/Erase Resume command (PER) is issued the Erase Suspend Status bit returns Low (FSR<6>=0)

6.5.3 Erase Status bit

The bit 5 of the Flag Status Register represents the Erase Status bit. It indicates an erase failure or a protection error when an erase operation is issued.

When the Erase Status bit is High (FSR<5>=1) after an Erase failure that means that the P/E Controller has applied the maximum pulses number to the portion to be erased and still failed to verify that it has correctly erased.

The Erase Status bit should be read once the P/E Controller Status bit is High.

The Erase Status bit is related to all possible erase operations: Sector Erase, Sub Sector Erase, and Bulk Erase in all the three available protocols (SPI, DIO-SPI and QIO-SPI).

Once the bit 5 is set High, it can only be reset Low (FSR<5>=0) by a Clear Flag Status Register command (CLFSR).

If set High it should be reset before a new Erase command is issued; otherwise the new command will appear to fail.

6.5.4 Program Status bit

The bit 4 of the Flag Status Register represents the Program Status bit. It indicates:

- a Program failure
- an attempt to program a '1' on '0' when VPP=VPPH (only when the pattern is a multiple of 64 bits, otherwise this bit is "Don't care").
- a protection error when a program is issued

When the Program Status bit is High (FSR<4>=1) after a Program failure that means that the P/E Controller has applied the maximum pulses number to the bytes and it still failed to verify that the required data have been correctly programmed.

After an attempt to program '1' on '0', the FSR<4> only goes High (FSR<4>=1) if VPP=VPPH and the data pattern is a multiple of 64 bits: if VPP is not VPPH, FSR<4> remains Low and the attempt is not shown while if VPP is equal to VPPh but the pattern is not a 64 bits multiple the bit 4 is Don't Care. The Program Status bit should be read once the P/E Controller Status bit is High.

The Program Status bit is related to all possible program operations in the Extended SPI protocol: Page Program, Dual and Quad Input Fast Program, Dual and Quad Input Extended Fast Program, and OTP Program.

The Program Status bit is related to the following program operations in the DIO-SPI and QIO-SPI protocols: Dual and Quad Command Page program and OTP program.

Once the bit is set High, it can only be reset Low (FSR<4>=0) by a Clear Flag Status Register command (CLFSR). If set High it should be reset before a new Program command is issued, otherwise the new command will appear to fail.

6.5.5 VPP Status bit

The bit 3 of the Flag Status Register represents the VPP Status bit. It indicates an invalid voltage on the VPP pin during Program and Erase operations. The VPP pin is sampled at the beginning of a Program or Erase operation.

If VPP becomes invalid during an operation, that is the voltage on VPP pin is below the VPPH Voltage (9V), the VPP Status bit goes High (FSR<3>=1) and indeterminate results can occur.

Once set High, the VPP Status bit can only be reset Low (FSR<3>=0) by a Clear Flag Status Register command (CLFSR). If set High it should be reset before a new Program or Erase command is issued, otherwise the new command will appear to fail.

6.5.6 Program Suspend Status bit

The bit 2 of the Flag Status register represents the Program Suspend Status bit, It indicates that an Program operation has been suspended or is going to be suspended.

The bit is set (FSR<2>=1) within the Erase Suspend Latency time, that is as soon as the Program/Erase Suspend command (PES) has been issued, therefore the device may still complete the operation before entering the Suspend Mode.

The Program Suspend Status should be considered valid when the P/E Controller bit is high (FSR<7>=1).

When a Program/Erase Resume command (PER) is issued the Program Suspend Status bit returns Low (FSR<2>=0)

6.5.7 Protection Status bit

The bit 1 of the Flag Status Register represents the Protection Status bit. It indicates that an Erase or Program operation has tried to modify the contents of a protected array sector, or

that a modify operation has tried to access to a locked OTP space. The Protection Status bit is related to all possible protection violations as follows:

- The sector is protected by Software Protection Mode 1 (SPM1) Lock registers,
- The sector is protected by Software Protection Mode 2 (SPM2) Block Protect Bits (standard SPI Status Register),
- An attempt to program OTP when locked,
- A Write Status Register command (WRSR) on STD SPI Status Register when locked by the SRWD bit in conjunction with the Write Protect ($\overline{W}$/VPP) signal (Hardware Protection Mode).

Once set High, the Protection Status bit can only be reset Low (FSR<1>=0) by a Clear Flag Status Register command (CLFSR). If set High it should be reset before a new command is issued, otherwise the new command will appear to fail.

6.5.8 Protection Status bit (4-Byte)

Bit 0 of the Flag Status register represents the 4-byte address mode Status bit. It indicates that 4-byte address mode was enabled by the Enter 4-byte Address Mode (EN4BYTEADDR) instruction.

The bit is set (FSR<0>=1) when EN4BYTEADDR has been correctly decoded by memory.

When a Exit 4-byte address mode (EX4BYTEADDR) is issued the 4-byte address mode Status bit returns Low (FSR<0>=0)

Clear Flag Status Register command (CLFSR) has no effect on this bit

7 Protection modes

There are protocol-related and specific hardware and software protection modes. They are described below.

7.1 SPI Protocol-related protections

This applies to all three protocols. The environments where non-volatile memory devices are used can be very noisy. No SPI device can operate correctly in the presence of excessive noise. To help combat this, the N25Q256 features the following data protection mechanisms:

- Power On Reset and an internal timer (tPUW) can provide protection against inadvertent changes while the power supply is outside the operating specification.
- Program, Erase, and Write Status Register instructions are checked to ensure the instruction includes a number of clock pulses that is a multiple of a byte before they are accepted for execution.
- All instructions that modify data must be preceded by a Write Enable (WREN) instruction to set the Write Enable Latch (WEL) bit. This bit is returned to its reset state by the following events (in Extended SPI protocol mode):
  - Power-up
  - Write Disable (WRDI) instruction completion
  - Write Status Register (WRSR) instruction completion
  - Write to Lock Register (WRLR) instruction completion
  - Program OTP (POTP) instruction completion
  - Page Program (PP) instruction completion
  - Dual Input Fast Program (DIFP) instruction completion
  - Dual Input Extended Fast Program (DIEFP) instruction completion
  - Quad Input Fast Program (QIFP) instruction completion
  - Quad Input Extended Fast Program (QIEFP) instruction completion
  - Subsector Erase (SSE) instruction completion
  - Sector Erase (SE) instruction completion
  - Bulk Erase (BE) instruction completion

This bit is also returned to its reset state after all the analogous events in DIO-SPI and QIO-SPI protocol modes.

7.2 Specific hardware and software protection

There are two software protected modes, SPM1 and SPM2, that can be combined to protect the memory array as required. The SPM2 can be locked by hardware with the help of the $\overline{W}$ input pin.

SPM1

The first software protected mode (SPM1) is managed by specific Lock Registers assigned to each 64 Kbyte sector.

The Lock Registers can be read and written using the Read Lock Register (RDLR) and Write to Lock Register (WRLR) instructions.

In each Lock Register two bits control the protection of each sector: the Write Lock bit and the Lock Down bit.

- Write Lock bit: The Write Lock bit determines whether the contents of the sector can be modified (using the Write, Program, or Erase instructions). When the Write Lock bit is set to '1', the sector is write protected - any operations that attempt to change the data in the sector will fail. When the Write Lock bit is reset to '0', the sector is not write protected by the Lock Register, and may be modified.
- Lock Down bit: The Lock Down bit provides a mechanism for protecting software data from simple hacking and malicious attack. When the Lock Down bit is set to '1', further modification to the Write Lock and Lock Down bits cannot be performed. A powerup is required before changes to these bits can be made. When the Lock Down bit is reset to '0', the Write Lock and Lock Down bits can be changed.

The definition of the Lock Register bits is given in Table 9: Lock Register out.

SPM2

The second software protected mode (SPM2) uses the Block Protect bits (BP3, BP2, BP1, BP0) and the Top/Bottom bit (TB bit) to allow part of the memory to be configured as read-only. See *Section 16: Ordering information*.

Table 10. Software protection truth table (Sectors 0 to 511, 64 Kbyte granularity)

Sector Lock Register		Protection Status
Lock Down bit	Write Lock bit	
0	0	Sector unprotected from Program/Erase/Write operations, protection status reversible.
0	1	Sector protected from Program/Erase/Write operations, protection status reversible.
1	0	Sector unprotected from Program/Erase/Write operations. Sector protection status cannot be changed except by a power-up.
1	1	Sector protected from Program/Erase/Write operations. Sector protection status cannot be changed except by a power-up.

As a second level of protection, the Write Protect signal (applied on the $\overline{W}$/VPP pin) can freeze the Status Register in a read-only mode. In this mode, the Block Protect bits (BP3, BP2, BP1, BP0) and the Status Register Write Disable bit (SRWD) are protected.

If a sector is write locked and/or locked-down during an erase or subsector erase suspend on that sector, the locking status bits are changed immediately. However, when that sector or subsector erase resumes and then completes, the sector lock register content affects the sector protection status. Locking operations cannot be performed during a program suspend. or during subsector erase suspend.

Table 11. Protected area sizes, Upper (TB bit = 0)

Status Register Content					Memory Content	
TB bit (Upper = 0; Lower = 1)	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	0	0	None	All sectors
0	0	0	0	1	Upper 512th	Sectors 0 to 511
0	0	0	1	0	Upper 256th	Sectors 0 to 510
0	0	0	1	1	Upper 128th	Sectors 0 to 508
0	0	1	0	0	Upper 64th	Sectors 0 to 504
0	0	1	0	1	Upper 32nd	Sectors 0 to 496
0	0	1	1	0	Upper 16th	Sectors 0 to 480
0	0	1	1	1	Upper eighth	Sectors 0 to 448
0	1	0	0	0	Upper quarter	Sectors 0 to 384
0	1	0	0	1	Upper half	Sectors 0 to 256
0	1	0	1	0	All sectors	None
0	1	0	1	1	All sectors	None
0	1	1	0	0	All sectors	None
0	1	1	0	1	All sectors	None
0	1	1	1	0	All sectors	None
0	1	1	1	1	All sectors	None
1	0	0	0	0	None	All sectors
1	0	0	0	1	Lower 512th	Sectors 1 to 511
1	0	0	1	0	Lower 256th	Sectors 2 to 511
1	0	0	1	1	Lower 128th	Sectors 4 to 511
1	0	1	0	0	Lower 64th	Sectors 8 to 511
1	0	1	0	1	Lower 32nd	Sectors 16 to 511
1	0	1	1	0	Lower 16th	Sectors 32 to 511
1	0	1	1	1	Lower eighth	Sectors 64 to 511
1	1	0	0	0	Lower quarter	Sectors 128 to 511
1	1	0	0	1	Lower half	Sectors 256 to 511
1	1	0	1	0	All sectors	None
1	1	0	1	1	All sectors	None
1	1	1	0	0	All sectors	None
1	1	1	0	1	All sectors	None
1	1	1	1	0	All sectors	None
1	1	1	1	1	All sectors	None

8 Memory organization

The memory is organized as:

- 33,554,432 bytes (8 bits each)
- 512 sectors (64 Kbytes each)
- 8,192 subsectors (4 Kbytes each)
- 131,1072 pages (256 bytes each)
- 64 OTP bytes located outside the main memory array

Each page can be individually programmed: bits are programmed from 1 to 0. The device is Subsector eraseble, Sector eraseble or Bulk Erasable but not Page Erasable: bits are erased from 0 to 1.

Figure 17. Block diagram

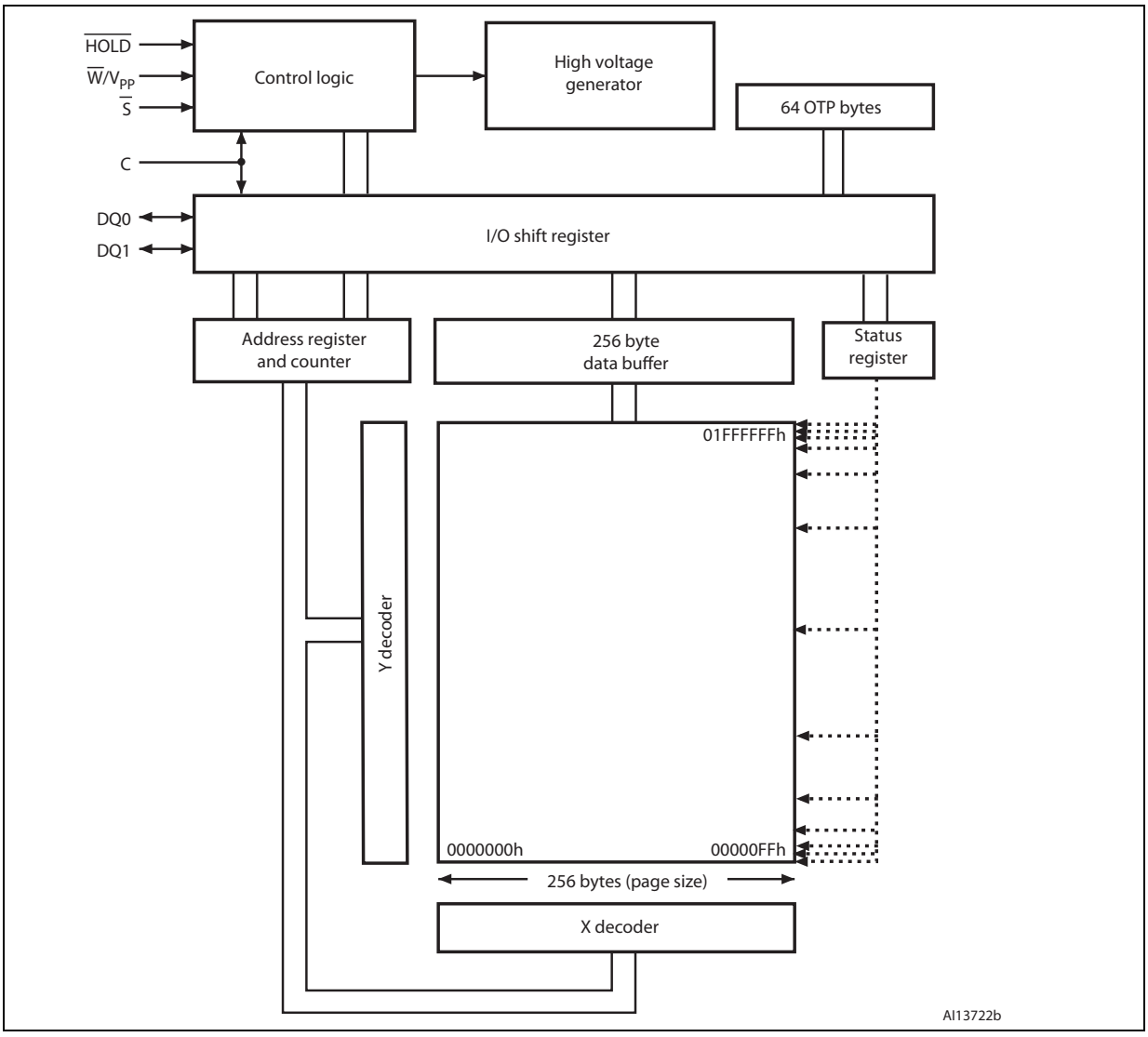


Table 12. Memory Map Sectors 511:256

Sector	Subsector	Address Range	
511	4095	00FF FFFFh	00FF F000h
	4094	00FF EFFFh	00FF E000h
	4093	00FF DFFFh	00FF D000h
	2092:2083	00FF _FFFh	00FF _000h
	4082	00FF 2FFFh	00FF 2000h
	4081	00FF 1FFFh	00FF 1000h
	4080	00FF 0FFFh	00FF 0000h
510	4079	00FE FFFFh	00FE F000h
	4078	00FE EFFFh	00FE E000h
	4077	00FE DFFFh	00FE D000h
	2076:4067	00FE _FFFh	00FE _000h
	4066	00FE 2FFFh	00FE 2000h
	4065	00FE 1FFFh	00FE 1000h
	4064	00FE 0FFFh	00FE 0000h
.			
.			
257	4127	0101 FFFFh	0101 F000h
	4126	0101 EFFFh	0101 E000h
	4125	0101 DFFFh	0101 D000h
	4124:2083	0101 _FFFh	0101 _000h
	4082	0101 2FFFh	0101 2000h
	4113	0101 1FFFh	0101 1000h
	4112	0101 0FFFh	0101 0000h
256	4111	0100 FFFFh	0100 F000h
	4110	0100 EFFFh	0100 E000h
	4109	0100 DFFFh	0100 D000h
	4108:4099	0100 _FFFh	0100 _000h
	4098	0100 2FFFh	0100 2000h
	4097	0100 1FFFh	0100 1000h
	4096	0100 0FFFh	0100 0000h

Table 13. Memory Map Sectors 255:128

Sector	Subsector	Address Range	
255	4095	00FF FFFFh	00FF F000h
	4094	00FF EFFFh	00FF E000h
	4093	00FF DFFFh	00FF D000h
	2092:2083	00FF _FFFh	00FF _000h
	4082	00FF 2FFFh	00FF 2000h
	4081	00FF 1FFFh	00FF 1000h
	4080	00FF 0FFFh	00FF 0000h
254	4079	00FE FFFFh	00FE F000h
	4078	00FE EFFFh	00FE E000h
	4077	00FE DFFFh	00FE D000h
	2076:4067	00FE _FFFh	00FE _000h
	4066	00FE 2FFFh	00FE 2000h
	4065	00FE 1FFFh	00FE 1000h
	4064	00FE 0FFFh	00FE 0000h
.			
.			
129	1055	0081 FFFFh	0081 F000h
	1054	0081 EFFFh	0081 E000h
	1053	0081 DFFFh	0081 D000h
	1052:1043	0081 _FFFh	0081 _000h
	1042	0081 2FFFh	0081 2000h
	1041	0081 1FFFh	0081 1000h
	1040	0081 0FFFh	0081 0000h
128	2063	0080 FFFFh	0080 F000h
	2062	0080 EFFFh	0080 E000h
	2061	0080 DFFFh	0080 D000h
	2060:2051	0080 _FFFh	0080 _000h
	2050	0080 2FFFh	0080 2000h
	2049	0080 1FFFh	0080 1000h
	2048	0080 0FFFh	0080 0000h

Table 14. Memory Map Sectors 127:64

Sector	Subsector	Address Range	
127	2047	007F FFFFh	007F F000h
	2046	007F EFFFh	007F E000h
	2045	007F DFFFh	007F D000h
	2044:2035	007F _FFFh	007F _000h
	2034	007F 2FFFh	007F 2000h
	2033	007F 1FFFh	007F 1000h
	2032	007F 0FFFh	007F 0000h
126	2031	007E FFFFh	007E F000h
	2030	007E EFFFh	007E E000h
	2029	007E DFFFh	007E D000h
	2028:2019	007E _FFFh	007E _000h
	2018	007E 2FFFh	007E 2000h
	2017	007E 1FFFh	007E 1000h
	2016	007E 0FFFh	007E 0000h
.			
.			
65	1055	0041 FFFFh	0041 F000h
	1054	0041 EFFFh	0041 E000h
	1053	0041 DFFFh	0041 D000h
	1052:1043	0041 _FFFh	0041 _000h
	1042	0041 2FFFh	0041 2000h
	1041	0041 1FFFh	0041 1000h
	1040	0041 0FFFh	0041 0000h
64	1039	0040 FFFFh	0040 F000h
	1038	0040 EFFFh	0040 E000h
	1037	0040 DFFFh	0040 D000h
	1036:1027	0040 _FFFh	0040 _000h
	1026	0040 2FFFh	0040 2000h
	1025	0040 1FFFh	0040 1000h
	1024	0040 0FFFh	0040 0000h

Table 15. Memory Map Sectors 63:0

Sector	Subsector	Address Range	
63	1023	003F FFFFh	003F F000h
	1022	003F EFFFh	003F E000h
	1021	003F DFFFh	003F D000h
	1020:1011	003F _FFFh	003F _000h
	1010	003F 2FFFh	003F 2000h
	1009	003F 1FFFh	003F 1000h
	1008	003F 0FFFh	003F 0000h
62	1007	003E FFFFh	003E F000h
	1006	003E EFFFh	003E E000h
	1005	003E DFFFh	003E D000h
	1004:995	003E _FFFh	003E _000h
	994	003E 2FFFh	003E 2000h
	993	003E 1FFFh	003E 1000h
	992	003E 0FFFh	003E 0000h
.			
.			
1	31	0001 FFFFh	0001 F000h
	30	0001 EFFFh	0001 E000h
	29	0001 DFFFh	0001 D000h
	28:19	0001 _FFFh	0001 _000h
	18	0001 2FFFh	0001 2000h
	17	0001 1FFFh	0001 1000h
	16	0001 0FFFh	0001 0000h
0	15	0000 FFFFh	0000 F000h
	14	0000 EFFFh	0000 E000h
	13	0000 DFFFh	0000 D000h
	12:3	0000 _FFFh	0000 _000h
	2	0000 2FFFh	0000 2000h
	1	0000 1FFFh	0000 1000h
	0	0000 0FFFh	0000 0000h

9 Instructions

The device can work in three different protocols: Extended SPI, DIO-SPI and QIO-SPI. Each protocol has a dedicated instruction set, and each instruction set features the same functionality:

- Read, program and erase the memory and the 64 byte OTP area,
- Suspend and resume the program or erase operations,
- Read and modify all the registers and to read the device ID: please note that in this case there is a small functionality difference among the single and the multiple I/O read ID instructions. See *Section 9.2.1: Multiple I/O Read Identification protocol* and *Section 9.3.1: Multiple I/O Read Identification (MIORDID)*.

The application can choose in every time of the device life which protocol to use by setting the dedicated bits either in the Non Volatile Configuration Register or the Volatile Enhanced Configuration Register.

Note: In multiple SPI protocols, all instructions, addresses, and data are parallel on two lines (DIO-SPI protocol) or four lines (QIO-SPI protocol).

All instructions, addresses and data are shifted in and out of the device, most significant bit first.

Serial Data input(s) is (are) sampled on the first rising edge of Serial Clock (C) after Chip Select ($\overline{S}$) is driven Low. Then, the one-byte instruction code must be shifted in to the device, most significant bit first, on Serial Data input(s), each bit being latched on the rising edges of Serial Clock (C). Instruction code is shifted into the device just on DQ0 in Extended SPI protocol, on DQ0 and DQ1 in DIO-SPI protocol and on DQ0, DQ1, DQ2, and DQ3 in QIO-SPI protocol.

In standard mode every instruction sequence starts with a one-byte instruction code. Depending on the instruction, this might be followed by address bytes, or by data bytes, or by both or none.

In XIP modes only read operation and exit XIP mode can be performed, and to read the memory content no instructions code are needed: the device directly receives addresses and after a configurable number of dummy clock cycle it outputs the required data.

9.1 Extended SPI Instructions

In Extended SPI protocol instruction set the instruction code is always shifted into the device just on DQ0 pin, while depending on the instruction addresses and input/output data can run on single, two or four wires.

In the case of a Read Instructions Data Bytes (READ), Read Data Bytes using 4 bytes address (READ4BYTE), Read Data Bytes at Higher Speed (FAST_READ), Read Data Bytes using 4 bytes at Higher Speed s(FAST_READ4BYTE), Dual Output Fast Read (DOFR), Dual Output Fast Read using 4 bytes address (DOFR4BYTE), Dual Input/Output Fast Read (DIOFR), Dual Input/Output Fast Read using 4 bytes address (DIOFR4BYTE), Quad Output Fast Read (QOFR), Quad Output Fast Read using 4 bytes address (QOFR4BYTE), Quad Input/Output Fast Read (QIOFR), Quad Input/Output Fast Read using 4 bytes address (QIOFR4BYTE), Read OTP (ROTP), Read Lock Registers (RDLR), Read Status Register (RDSR), Read Flag Status Register (RFSR), Read NV Configuration Register (RDNVCR), Read Volatile Configuration Register (RDVCR), Read Volatile

Enhanced Configuration Register (RDVECR), Read Serial Flash Discovery Parameter (RDSFDP), Read Extended Address Register (RDEAR) and Read Identification (RDID) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (S) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a Page Program (PP), Program OTP (POTP), Dual Input Fast Program (DIFP), Dual Input Extended Fast Program (DIEFP), Quad Input Fast Program (QIFP), Quad Input Extended Fast Program (QIEFP), Subsector Erase (SSE), Sector Erase (SE), Bulk Erase (BE), Write Status Register (WRSR), Clear Flag Status Register (CLFSR), Write to Lock Register (WRLR), Write Configuration Register (WRVCR), Write Enhanced Configuration Register (WRVECR), Write NV Configuration Register (WRNVCR), Write Enable (WREN) or Write Extended Address Register (WREAR), enter 4-byte address mode (EN4BYTEADDR), Exit 4-byte address mode (EX4BYTEADDR) or Write Disable (WRDI) instruction, Chip Select (S) must be driven High exactly at a byte boundary. Otherwise the instruction is rejected, and is not executed. That is, Chip Select (S) must driven High when the number of clock pulses after Chip Select (S) being driven Low is an exact multiple of eight.

All attempts to access the memory array are ignored during:

- Write Status Register cycle
- Write Non Volatile Configuration Register
- Program cycle
- Erase cycle

The following continue unaffected, with one exception:

- Internal Write Status Register cycle,
- Write Non Volatile Configuration Register,
- Program cycle,
- Erase cycle

The only exception is the Program/Erase Suspend instruction (PES), that can be used to pause all the program and the erase cycles except for:

- Program OTP (POTP),
- Bulk Erase,
- Write Non Volatile Configuration Register.

The suspended program or erase cycle can be resumed by the Program/Erase Resume instruction (PER). During the program/erase cycles, the polling instructions (both on the Status register and on the Flag Status register) are also accepted to allow the application to check the end of the internal modify cycles.

Note: These polling instructions don't affect the internal cycles performing.

Table 16. Instruction set: extended SPI protocol (page 1 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
RDID	Read Identification	1001 111x	9Eh / 9Fh	0	0	1 to 20
READ	Read Data Bytes	0000 0011	03h	3/4(1)	0	1 to ∞
READ4BYTE	Read Data Bytes using 4 Bytes Address	0001 0011	13h	4	0	1 to ∞
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0Bh	3/4(1)	8 (2)	1 to ∞
FAST_READ4BYTE	Read Data Bytes at Higher Speed using 4 Bytes Address	0000 1100	0Ch	4	8 (2)	1 to ∞
RDSFDP	Read Serial Flash Discovery Parameter	01011010	5Ah	3	8	1 to ∞
DOFR	Dual Output Fast Read	0011 1011	3Bh	3/4(1)	8 **(2)**	1 to ∞
DOFR4BYTE	Dual Output Fast Read using 4 Byte Address	0011 1100	3Ch	4	8 (2)	1 to ∞
DIOFR	Dual Input/Output Fast Read	1011 1011	BB	3/4(1)	8 **(2)**	1 to ∞
DIOFR4BYTE	Dual Input/Output Fast Read using 4 Byte Address	1011 1100	BCh	4	8 (2)	1 to ∞
QOFR	Quad Output Fast Read	0110 1011	6Bh	3/4(1)	8 **(2)**	1 to ∞
QOFR4BYTE	Quad Output Fast Read using 4 Byte Address	0110 1100	6Ch	4	8 (2)	1 to ∞
QIOFR	Quad Input/Output Fast Read	1110 1011	EBh	3/4(1)	10 **(2**	1 to ∞
QIOFR4BYTE	Quad Input/Output Fast Read using 4 Byte Address	1110 1100	ECh	4	10 (2)	1 to ∞
ROTP	Read OTP (Read of OTP area)	0100 1011	4Bh	3/4(1)	8 **(2)**	1 to 65
WREN	Write Enable	0000 0110	06h	0	0	0
WRDI	Write Disable	0000 0100	04h	0	0	0
PP	Page Program	0000 0010	02h	3/4(1)	0	1 to 256
DIFP	Dual Input Fast Program	1010 0010	A2h	3/4(1)	0	1 to 256
DIEFP	Dual Input Extended Fast Program	1101 0010	D2h	3/4(1)	0	1 to 256
QIFP	Quad Input Fast Program	0011 0010	32h	3/4(1)	0	1 to 256
QIEFP	Quad Input Extended Fast Program	0001 0010	12h	3/4(1)	0	1 to 256

Table 16. Instruction set: extended SPI protocol (page 2 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
POTP	Program OTP (Program of OTP area)	0100 0010	42h	3/4(1)	0	1 to 65
SSE	SubSector Erase	0010 0000	20h	3/4(1)	0	0
SE	Sector Erase	1101 1000	D8h	3/4(1)	0	0
BE	Bulk Erase	1100 0111	C7h	0	0	0
PER	Program/Erase Resume	0111 1010	7Ah	0	0	0
PES	Program/Erase Suspend	0111 0101	75h	0	0	0
RDSR	Read Status Register	0000 0101	05h	0	0	1 to ∞
WRSR	Write Status Register	0000 0001	01h	0	0	1
RDLR	Read Lock Register	1110 1000	E8h	3/4(1)	0	1 to ∞
WRLR	Write to Lock Register	1110 0101	E5h	3/4(1)	0	1
RFSR	Read Flag Status Register	0111 0000	70h	0	0	1 to ∞
CLFSR	Clear Flag Status Register	0101 0000	50h	0	0	0
RDNVCR	Read NV Configuration Register	1011 0101	B5h	0	0	2
WRNVCR	Write NV Configuration Register	1011 0001	B1h	0	0	2
RDVCR	Read Volatile Configuration Register	1000 0101	85h	0	0	1 to ∞
WRVCR	Write Volatile Configuration Register	1000 0001	81h	0	0	1
RDVECR	Read Volatile Enhanced Configuration Register	0110 010	65h	0	0	1 to ∞
WRVECR	Write Volatile Enhanced Configuration Register	0110 0001	61h	0	0	1
EN4BYTEADDR	Enter 4-byte address mode	1011 0111	B7h	0	0	0
EX4BYTEADDR	Exit 4-byte address mode	1110 1001	E9h	0	0	0
WREAR	Write Extended Address Register	1100 0101	C5h	0	0	0
RDEAR	Read Extended Address Register	1100 1000	C8h	0	0	0
RSTEN	Reset Enable	0110 0110	66h	0	0	0
RST	Reset Memory	1001 1001	99h	0	0	0

1. Accordingly to address mode (default is 3).
2. The Number of dummy clock cycles is configurable by user.

9.1.1 Read Identification (RDID)

The Read Identification (RDID) instruction allows to read the device identification data:

- Manufacturer identification (1 byte)
- Device identification (2 bytes)
- A Unique ID code (UID) (17 bytes, of which 14 factory programmed

The manufacturer identification is assigned by JEDEC, and has the value 20h. The device identification is assigned by the device manufacturer, and indicates the memory type in the first byte (BAh), and the memory capacity of the device in the second byte (19h). The UID is composed by 17 read only bytes, containing the length of the following data in the first byte (set to 10h), 2 bytes of Extended Device ID (EDID) to identify the specific device configuration (Top, Bottom or uniform architecture, Hold or Reset functionality), and 14 bytes of the Customized Factory Data (CFD) content. The CFD bytes are factory programmed.

The CFD bytes are factory programmed with a unique identification number for each single device. This area can be accessed in read mode only. It is impossible to change the security number after it has been written by manufacturing.

Any Read Identification (RDID) instruction while an Erase or Program cycle is in progress, is not decoded, and has no effect on the cycle that is in progress.

The device is first selected by driving Chip Select ($\overline{S}$) Low. Then, the 8-bit instruction code for the instruction is shifted in. After this, the 24-bit device identification, stored in the memory, the 17 bytes of UID content will be shifted out on Serial Data output (DQ1). Each bit is shifted out during the falling edge of Serial Clock (C).

The Read Identification (RDID) instruction is terminated by driving Chip Select ($\overline{S}$) High at any time during data output. When Chip Select ($\overline{S}$) is driven High, the device is put in the Standby Power mode. Once in the Standby Power mode, the device waits to be selected, so that it can receive, decode and execute instructions.

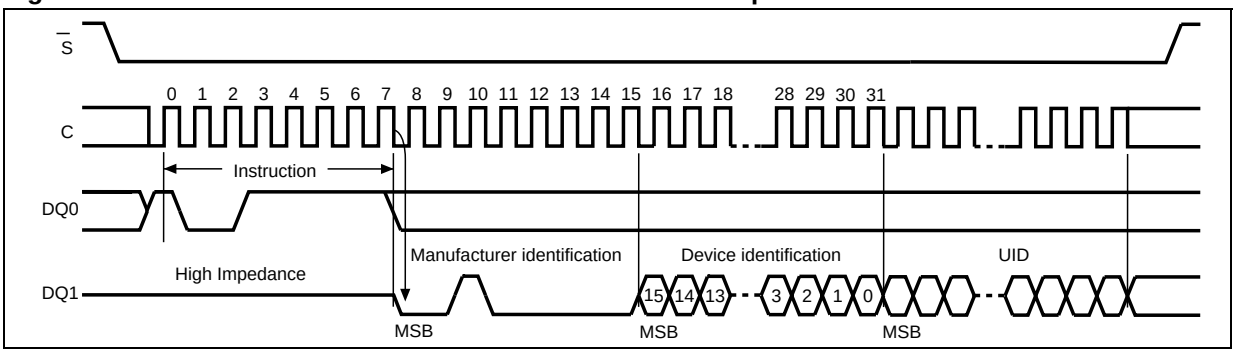
Table 17. Read Identification data-out sequence

Manufacturer Identification	Device identification		UID		
	Memory type	Memory capacity	EDID+CFD length	EDID	CFD
20h	BAh	19h	10h	2 bytes	14 bytes

Table 18. Extended Device ID table (first byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	VCR XIP bit setting: 0 = required, 1 = not required	Hold/Reset function: 0 = HOLD, 1 = Reset	Addressing: 0 = by Byte,	Architecture: 00 = Uniform,	

Figure 18. Read identification instruction and data-out sequence



9.1.2 Read Data Bytes (READ)

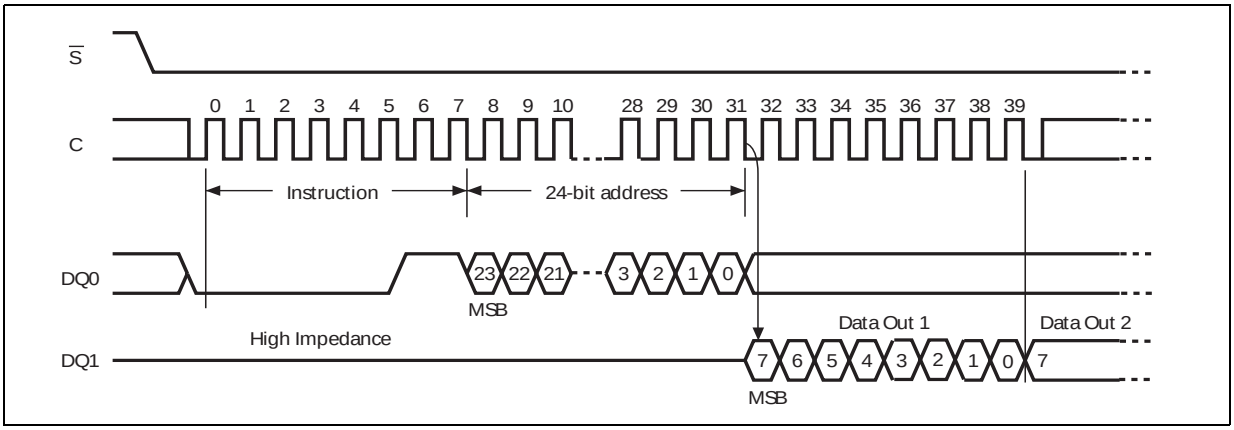
The device is first selected by driving Chip Select ($\overline{S}$) Low. The instruction code for the Read Data Bytes (READ) instruction is followed by a 3-byte address (A23-A0), each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, is shifted out on Serial Data output (DQ1), each bit being shifted out, at a maximum frequency fR, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes (READ) instruction. When the highest address is reached, the address counter rolls over to 00000000h, allowing the read sequence to be continued indefinitely.

The Read Data Bytes (READ) instruction is terminated by driving Chip Select ($\overline{S}$) High. Chip Select ($\overline{S}$) can be driven High at any time during data output. Any Read Data Bytes (READ) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 19. Read Data Bytes instruction and data-out sequence



9.1.3 Read Data Bytes using 4 Byte Address (READ4BYTE)

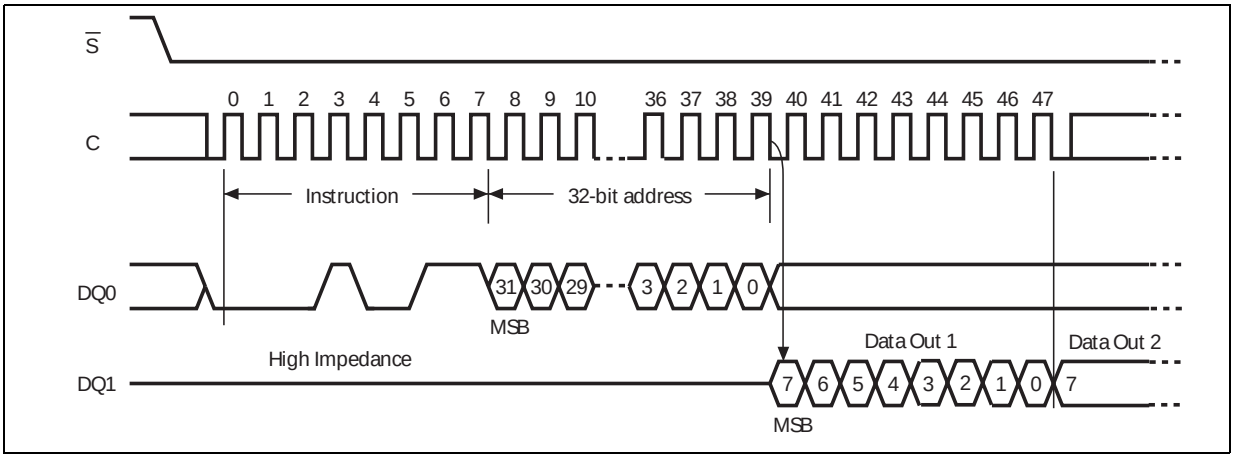
The device is first selected by driving Chip Select (S) Low. The instruction code for the Read Data Bytes using 4 Bytes Address (READ4BYTE) instruction is followed by a 4-byte address (A31-A0), each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, is shifted out on Serial Data output (DQ1), each bit being shifted out, at a maximum frequency fR, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single instruction. When the highest address is reached, the address counter rolls over to 00000000h, allowing the read sequence to be continued indefinitely.

This instruction is terminated by driving Chip Select (S) High. Chip Select (S) can be driven High at any time during data output. Any instruction, while an Erase, Program, or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

EAR becomes 'don't care' while the instruction code is executed, and neither FSR<0> not NVCR<0> are modified by this command, so the addressing mode returns in the formerly selected mode after the operation is completed.

Figure 20. Read Data Bytes using 4 Byte Address Instruction and Data-Out Sequence



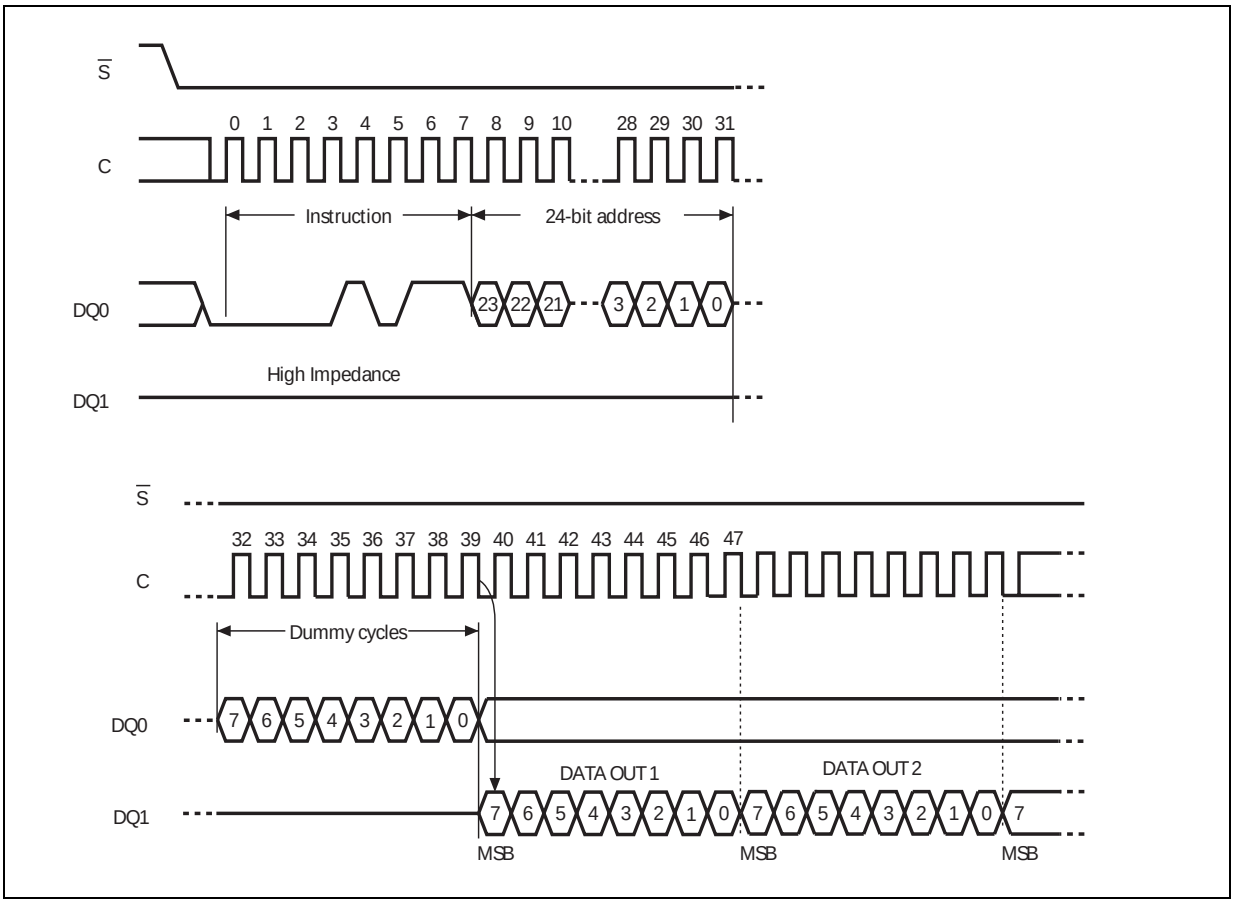
9.1.4 Read Data Bytes at Higher Speed (FAST_READ)

The device is first selected by driving Chip Select (S) Low. The instruction code for the Read Data Bytes at Higher Speed (FAST_READ) instruction is followed by a 3-byte address (A23-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on Serial Data output (DQ1) at a maximum frequency fC, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes at Higher Speed (FAST_READ) instruction. When the highest address is reached, the address counter rolls over to 00000000h, allowing the read sequence to be continued indefinitely.

The Read Data Bytes at Higher Speed (FAST_READ) instruction is terminated by driving Chip Select (S) High. Chip Select (S) can be driven High at any time during data output. Any Read Data Bytes at Higher Speed (FAST_READ) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress. If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 21. Read Data Bytes at Higher Speed instruction and data-out sequence



9.1.5 Read Data Bytes at Higher Speed using 4 Byte Address (FAST_READ4BYTE)

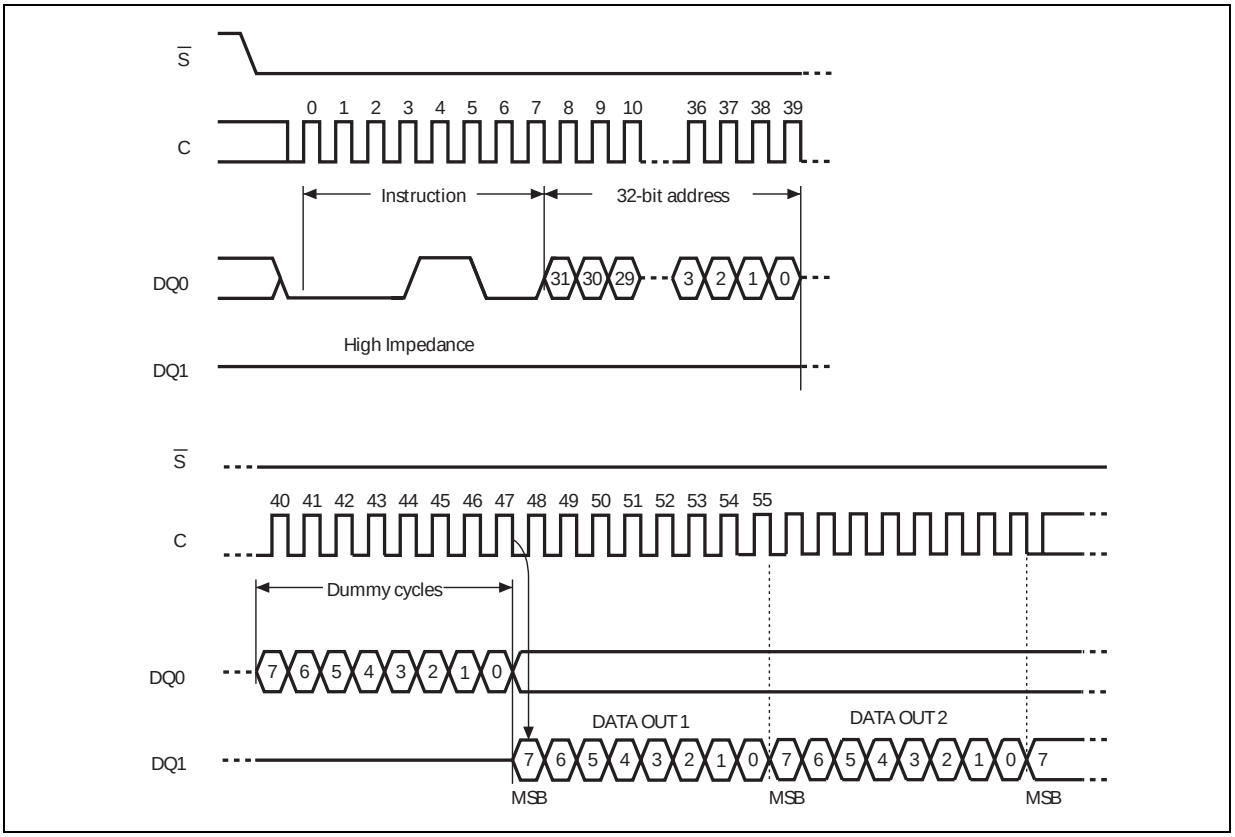
The device is first selected by driving Chip Select (S) Low. The instruction code for the Read Data Bytes at Higher Speed using 4 bytes address (FAST_READ4BYTE) instruction is followed by a 4-byte address (A31-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on Serial Data output (DQ1) at a maximum frequency fC, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single instruction. When the highest address is reached, the address counter rolls over to 00000000h, allowing the read sequence to be continued indefinitely.

This instruction is terminated by driving Chip Select (S) High. Chip Select (S) can be driven High at any time during data output. Any instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

EAR becomes 'don't care' while the instruction code is executed, and neither FSR<0> not NVCR<0> are modified by this command, so the addressing mode returns in the formerly selected mode after the operation is completed.

Figure 22. Read Data Bytes at Higher Speed using 4 Byte Address Instruction



9.1.6 Read Serial Flash Discovery Parameter

The Read Serial Flash Discovery Parameter (RDSFDP) instruction allows reading the Serial Flash Discovery Parameter area (SFDP).

This SFDP area is composed of 2048 read-only bytes containing operating characteristics and vendor specific information. The SFDP area is factory programmed.

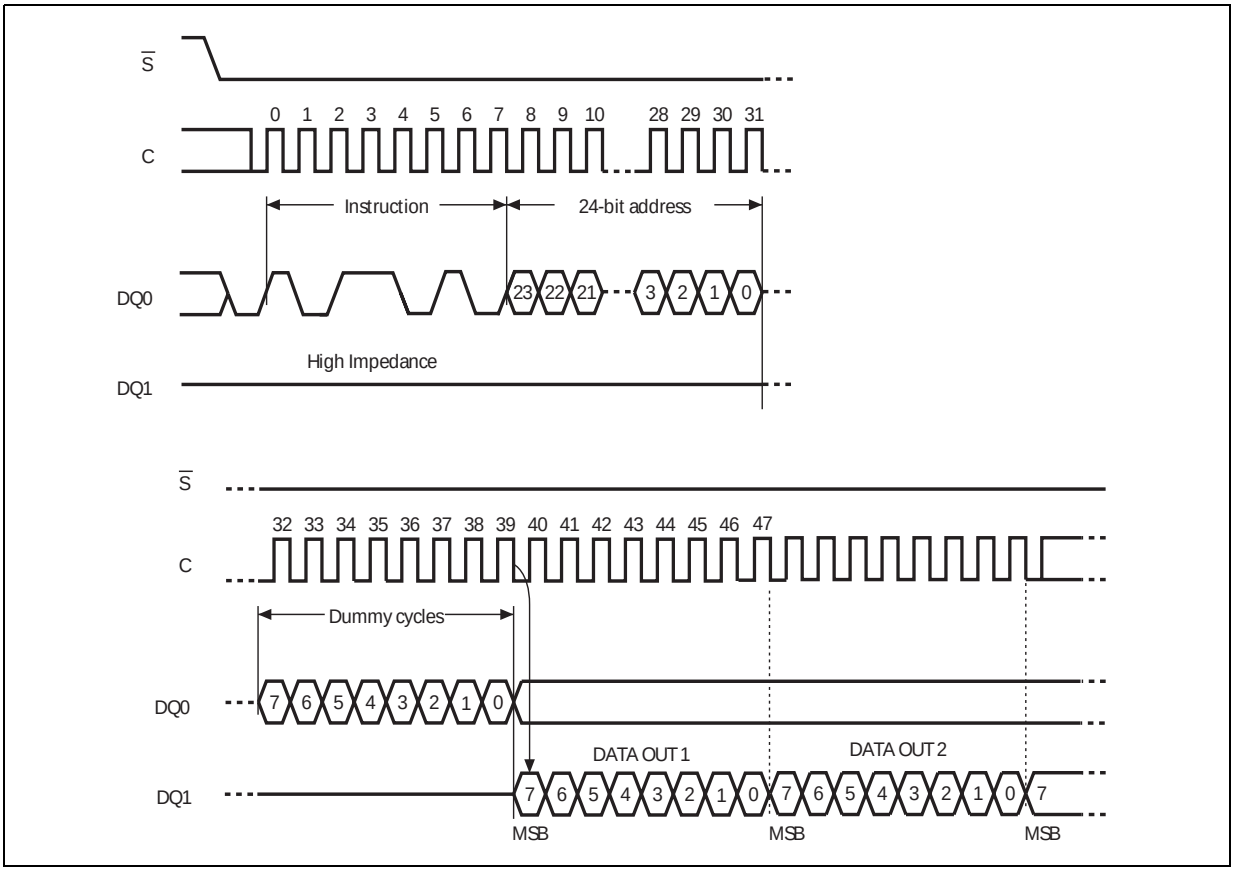
Note: Data to be written to the SFDP area is in definition phase.

If the SFDP area is blank, the device is shipped with all the SFDP bytes at FFh. If only a portion of the SFDP area is written to, the portion not used is shipped with bytes in erased state (FFh).

The instruction sequence for RDSFDP has the same structure as that of a Fast Read instruction. First, the device is selected by driving Chip Select (S) Low. Next, the 8-bit instruction code (5Ah) and the 24-bit address are shifted in, followed by 8 dummy clock cycles.

The bytes of SFDP content are shifted out on the Serial Data Output (DQ1) starting from the specified address. Each bit is shifted out during the falling edge of Serial Clock (C). The Read SFDP instruction is terminated by driving Chip Select (S) High at any time during data output. SFDP area is always addressable by means 3 byte regardless active address mode

Figure 23. Read Serial Flash Discovery Instruction and Data-out Sequence



9.1.7 Dual Output Fast Read (DOFR)

The Dual Output Fast Read (DOFR) instruction is very similar to the Read Data Bytes at Higher Speed (FAST_READ) instruction, except that the data are shifted out on two pins (pin DQ0 and pin DQ1) instead of only one. Outputting the data on two pins instead of one doubles the data transfer bandwidth compared to the Read Data Bytes at Higher Speed (FAST_READ) instruction.

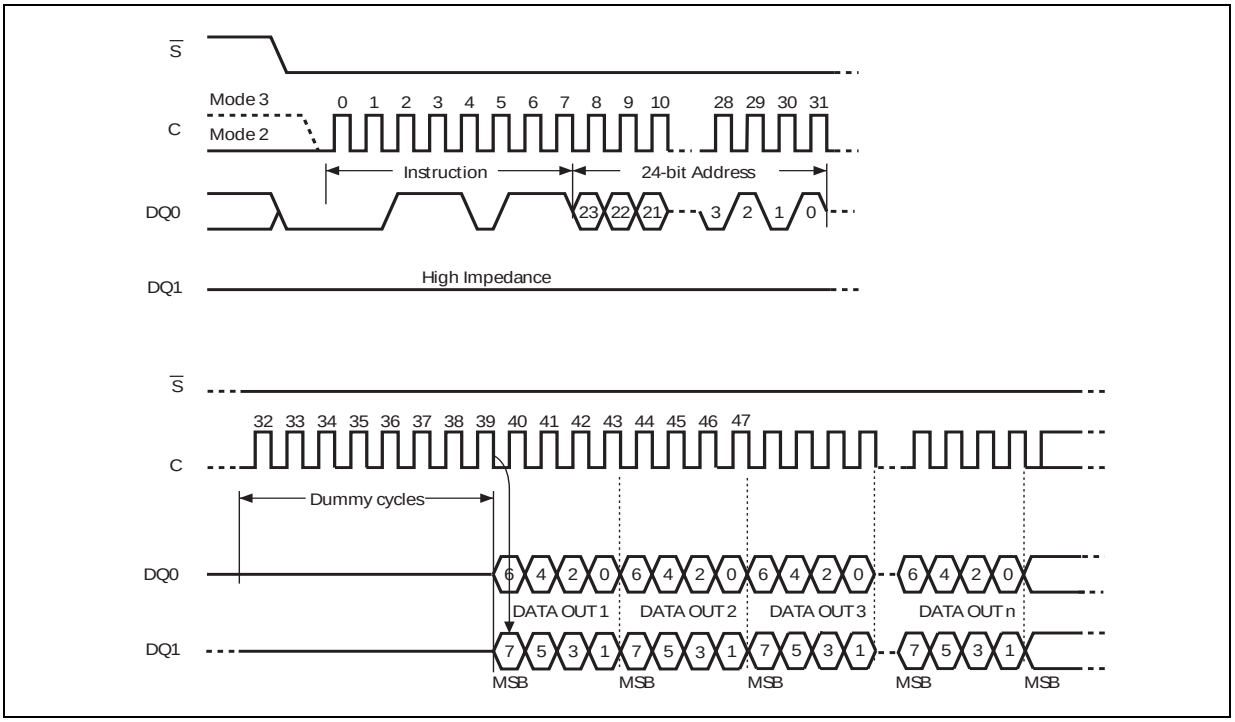
The device is first selected by driving Chip Select ($\overline{S}$) Low. The instruction code for the Dual Output Fast Read instruction is followed by a 3-byte address (A23-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on DQ0 and DQ1 at a maximum frequency Fc, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out on DQ0 and DQ1. The whole memory can, therefore, be read with a single Dual Output Fast Read (DOFR) instruction.

When the highest address is reached, the address counter rolls over to 00000000h, so that the read sequence can be continued indefinitely.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 24. Dual Output Fast Read Instruction Sequence



9.1.8 Dual Output Fast Read using 4 Byte Address (DOFR4BYTE)

The Dual Output Fast Read using 4 bytes address (DOFR4BYTE) instruction is very similar to the Read Data Bytes at Higher Speed using 4 bytes address (FAST_READ4BYTE) instruction, except that the data are shifted out on two pins (pin DQ0 and pin DQ1) instead of only one. Outputting the data on two pins instead of one doubles the data transfer bandwidth compared to the Read Data Bytes at Higher Speed using 4 bytes address (FAST_READ4BYTE) instruction.

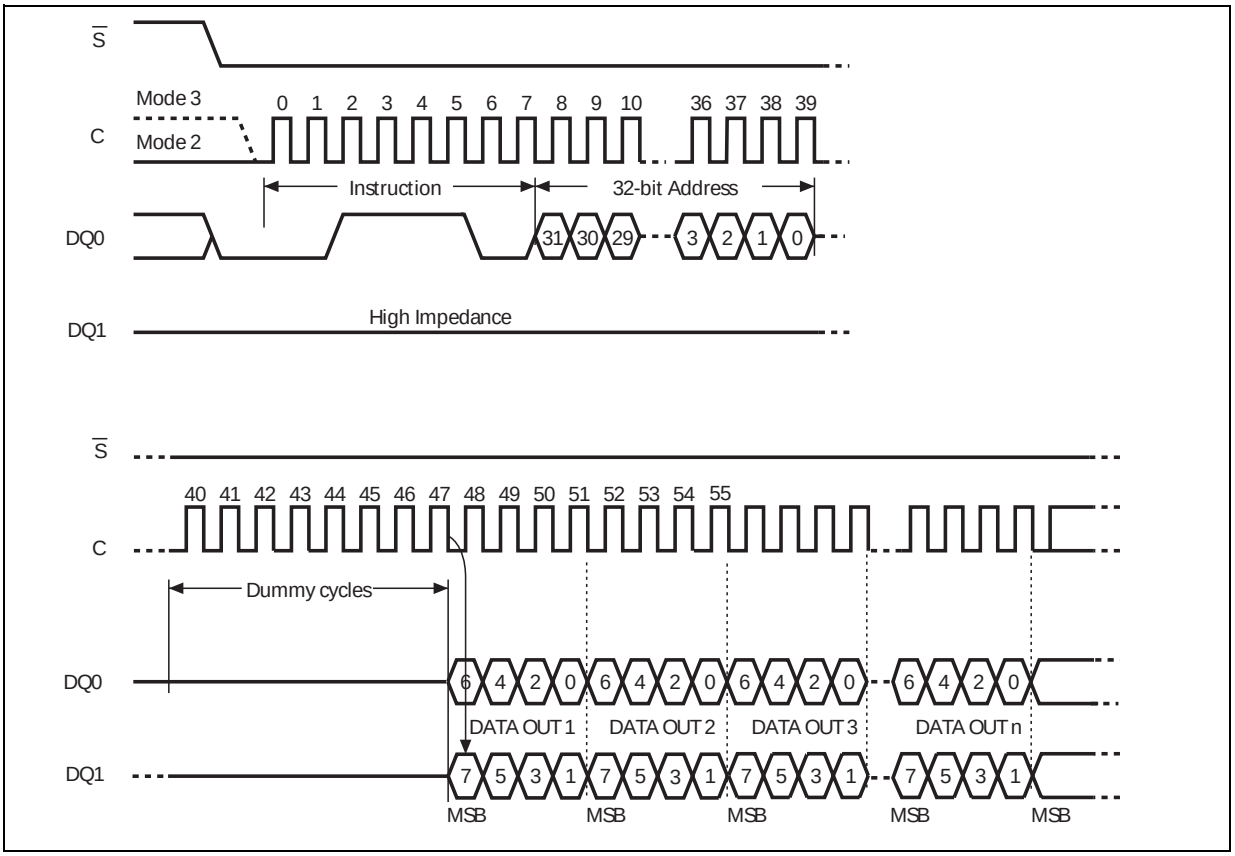
The device is first selected by driving Chip Select (S) Low. The instruction code is followed by a 4-byte address (A31-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on DQ0 and DQ1 at a maximum frequency Fc, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out on DQ0 and DQ1. The whole memory can, therefore, be read with a instruction.

When the highest address is reached, the address counter rolls over to 00000000h, so that the read sequence can be continued indefinitely.

EAR becomes 'don't care' while the instruction code is executed, and neither FSR<0> not NVCR<0> are modified by this command, so the addressing mode returns in the formerly selected mode after the operation is completed.

Figure 25. Dual Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence

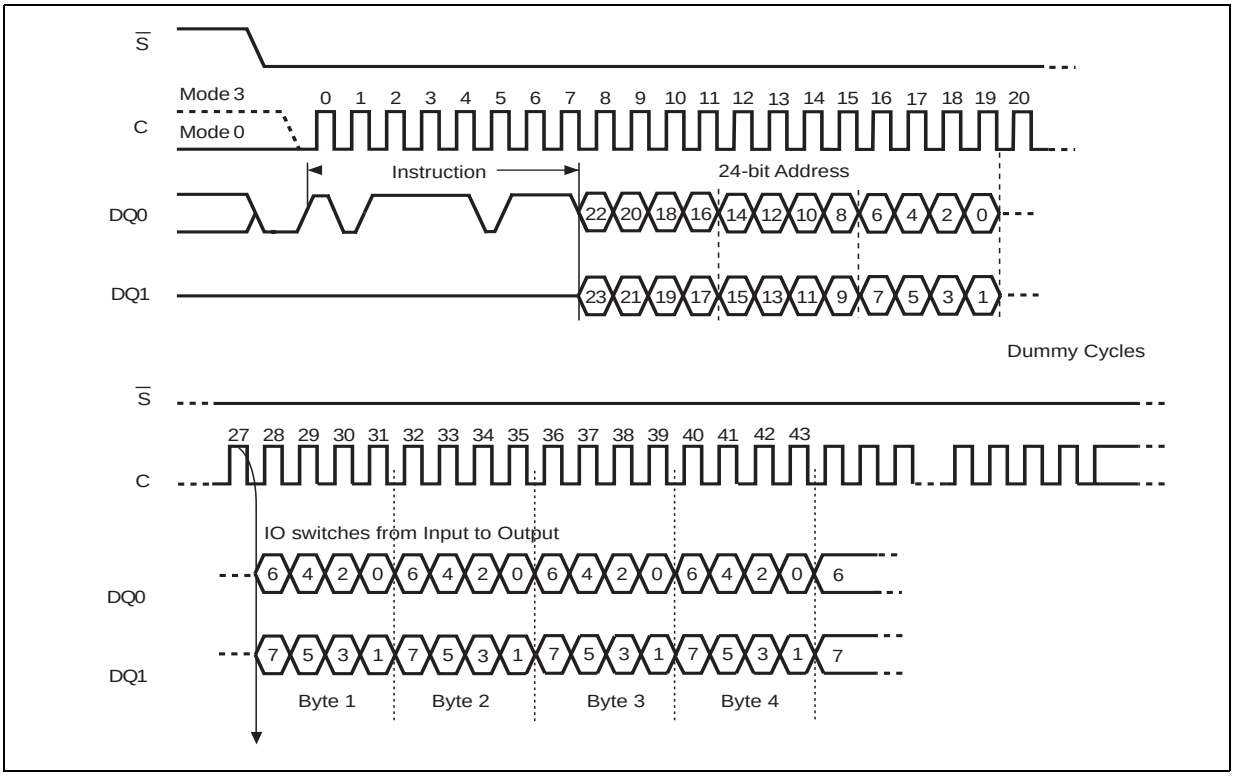


9.1.9 Dual I/O Fast Read

The Dual I/O Fast Read (DIOFR) instruction is very similar to the Dual Output Fast Read (DOFR), except that the address bits are shifted in on two pins (pin DQ0 and pin DQ1) instead of only one.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

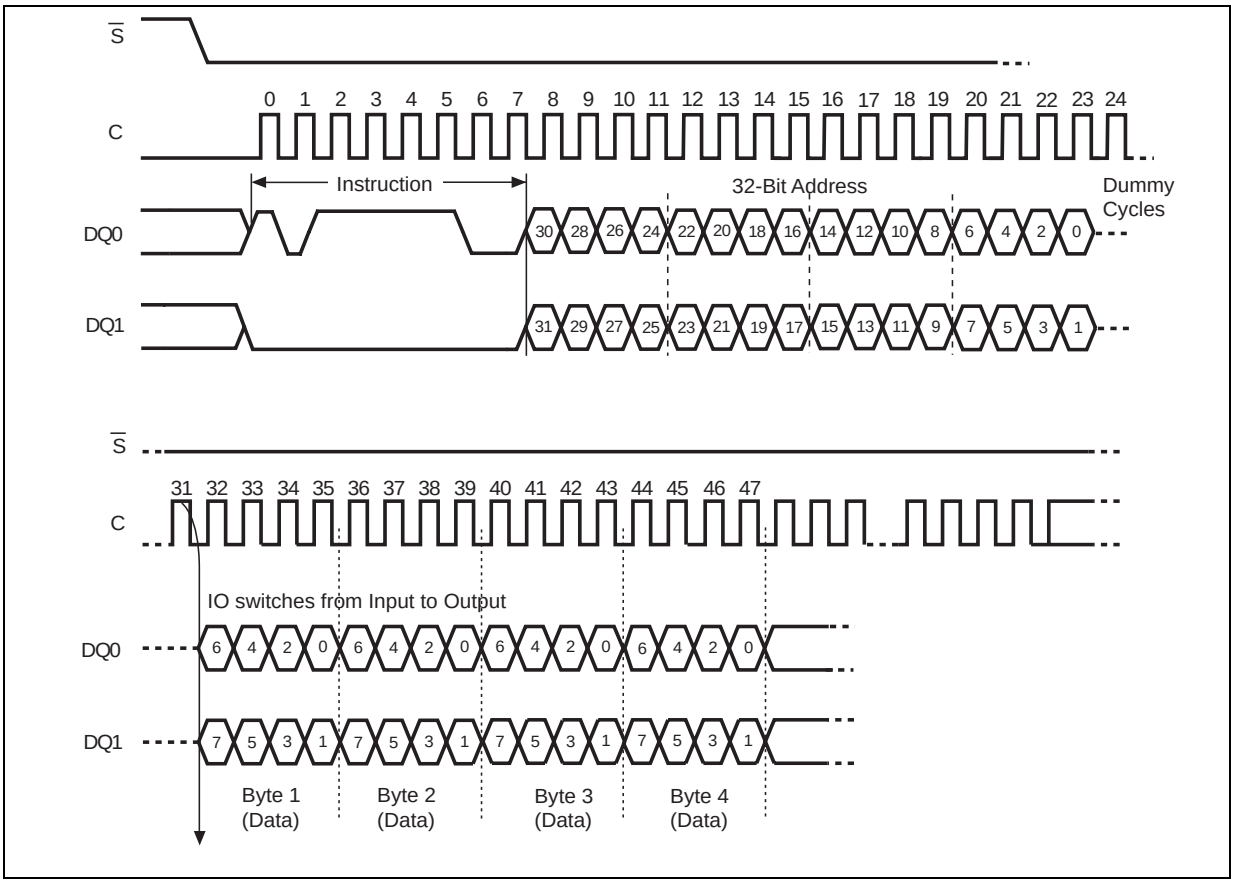
Figure 26. Dual I/O Fast Read Instruction Sequence



9.1.10 Dual Input/Output Fast Read using 4 Byte Address (DIOFR4BYTE)

The Dual Input/Output Fast Read using 4 bytes address (DIOFR4BYTE) instruction is very similar to the Dual Output Fast Read using 4 bytes address (DOFR4BYTE), except that the address bits are shifted in on two pins (pin DQ0 and pin DQ1) instead of only one.

Figure 27. Dual Input/Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence



9.1.11 Quad Output Fast Read

The Quad Output Fast Read (QOFR) instruction is very similar to the Dual Output Fast Read (DOFR) instruction, except that the data are shifted out on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1) instead of only two. Outputting the data on four pins instead of one doubles the data transfer bandwidth compared to the Dual Output Fast Read (DOFR) instruction.

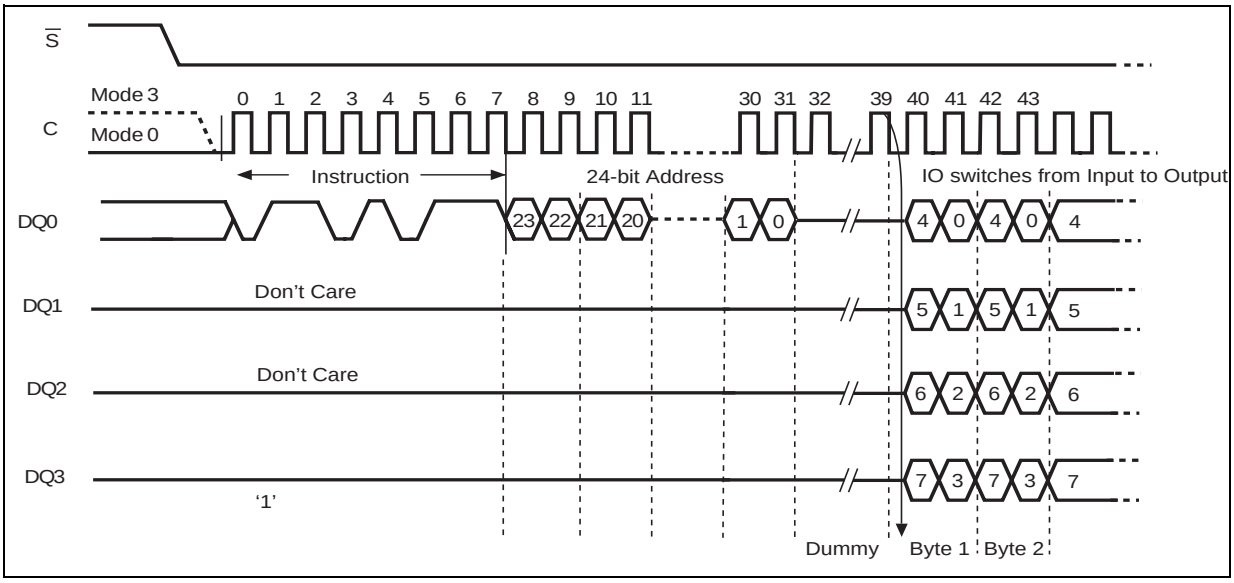
The device is first selected by driving Chip Select (S) Low. The instruction code for the Quad Output Fast Read instruction is followed by a 3-byte address (A23-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1) at a maximum frequency fC, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out on pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1). The whole memory can, therefore, be read with a single Quad Output Fast Read (QOFR) instruction. When the highest address is reached, the address counter rolls over to 00000000h, so that the read sequence can be continued indefinitely.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Note: Reset functionality is available instead of Hold in devices with a dedicated part number. See Section 16: Ordering information.

Figure 28. Quad Output Fast Read instruction sequence



9.1.12 Quad Output Fast Read using 4 Byte Address (QOFR4BYTE)

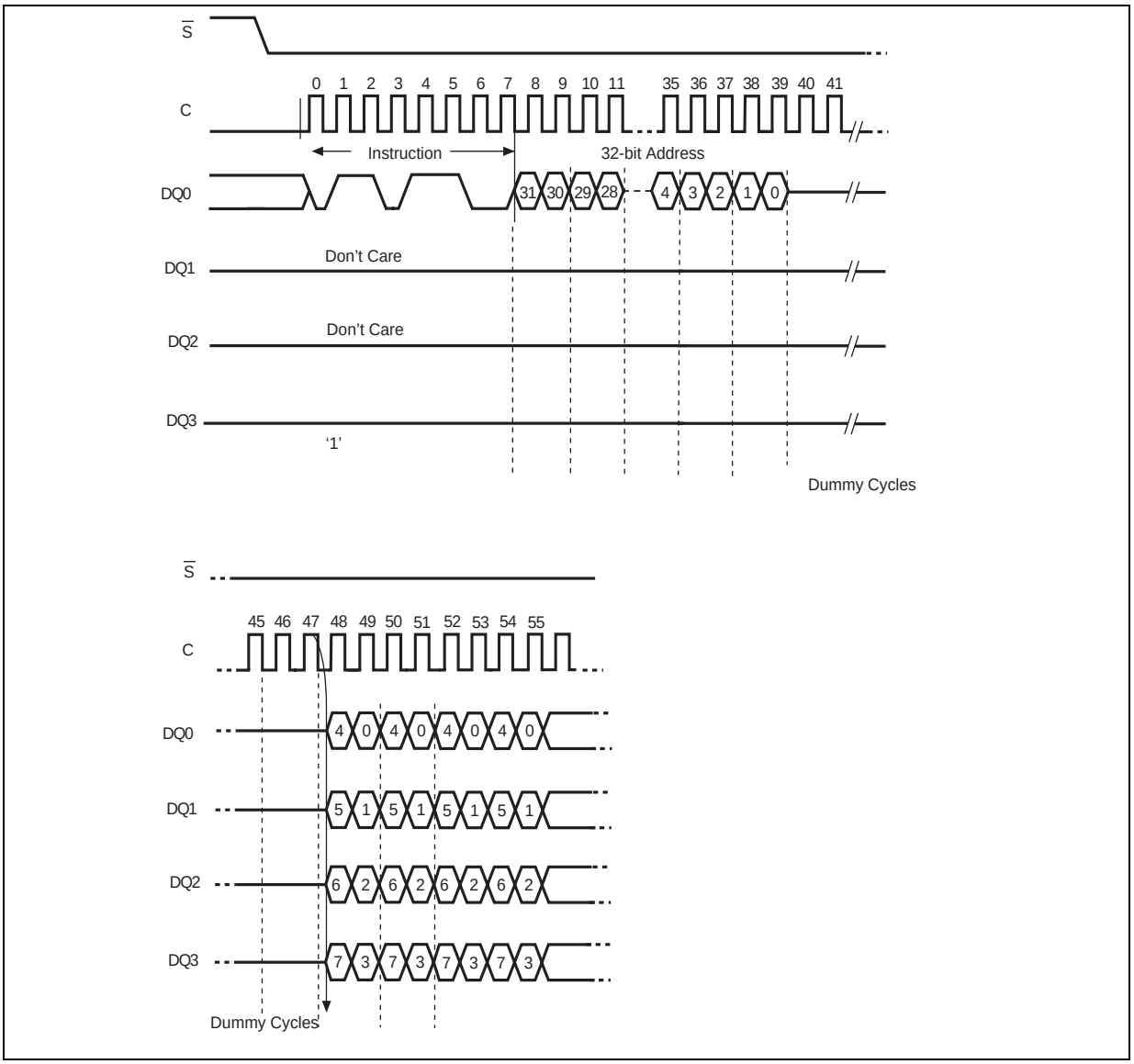
The Quad Output Fast Read using 4 bytes address (QOFR4BYTE) instruction is very similar to the Dual Output Fast Read using 4 bytes address (DOFR4BYTE) instruction, except that the data are shifted out on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1) instead of only two. Outputting the data on four pins instead of one doubles the data transfer bandwidth compared to the Dual Output Fast Read using 4 bytes address (DOFR4BYTE) instruction.

The device is first selected by driving Chip Select (S) Low. The instruction code for is followed by a 4-byte address (A31-A0) and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1) at a maximum frequency fC, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out on pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1). The whole memory can, therefore, be read with a single instruction. When the highest address is reached, the address counter rolls over to 00000000h, so that the read sequence can be continued indefinitely.

EAR becomes 'don't care' while the instruction code is executed, and neither FSR<0> not NVCR<0> are modified by this command, so the addressing mode returns in the formerly selected mode after the operation is completed. Reset functionality is available instead of Hold in devices with a dedicated part number as explained under Ordering Information.

Figure 29. Quad Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence



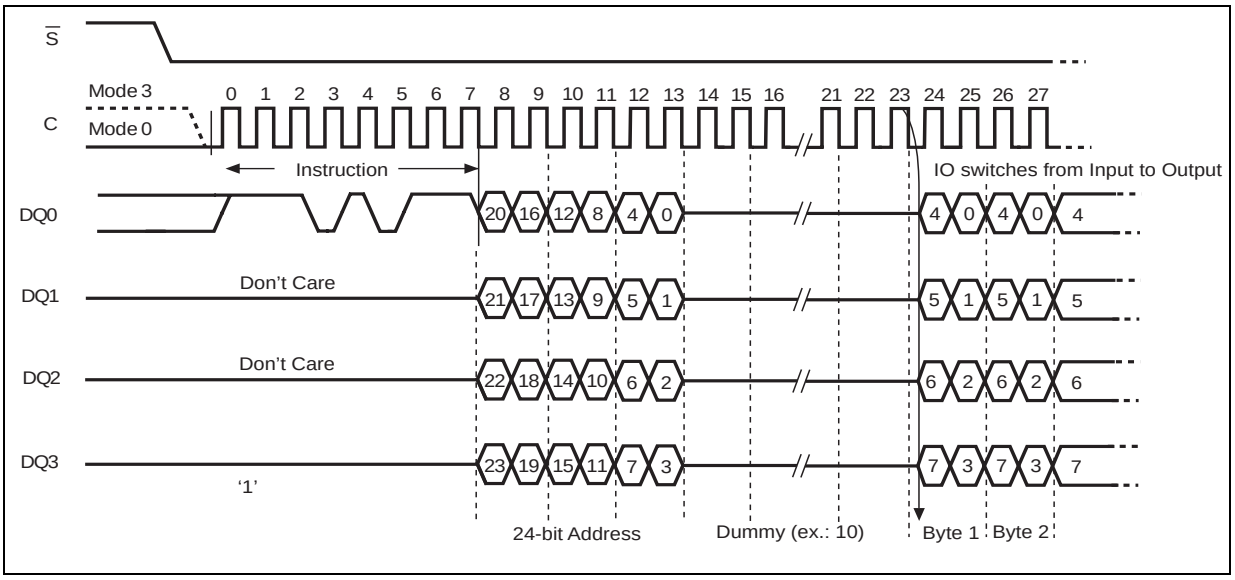
9.1.13 Quad I/O Fast Read

The Quad I/O Fast Read (QIOFR) instruction is very similar to the Quad Output Fast Read (QOFR), except that the address bits are shifted in on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1)) instead of only one.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Note: Reset functionality is available instead of Hold in devices with a dedicated part number. See Section 16: Ordering information.

Figure 30. Quad Input/Output Fast Read instruction sequence

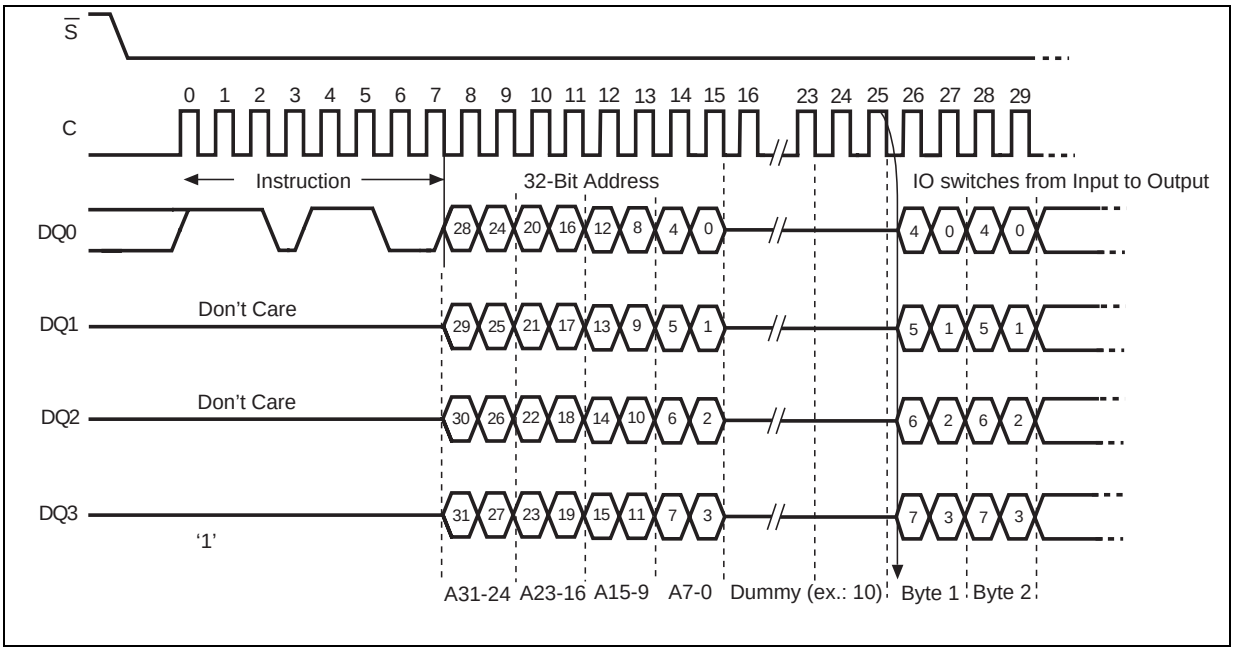


9.1.14 Quad Input/Output Fast Read using 4 Byte Address (QIOFR4BYTE)

The Quad Input/Output Fast Read using 4 bytes address (QIOFR4BYTE) instruction is very similar to the Dual Input/Output Fast Read using 4 bytes address (DIOFR4BYTE), except that the address bits are shifted in on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3 (1)) instead of only one.

Reset functionality is available instead of Hold in devices with a dedicated part number as explained under Ordering Information.

Figure 31. Quad Input/Output Fast Read using 4 Byte Address Instruction and Data-Out Sequence



9.1.15 Read OTP (ROTP)

The device is first selected by driving Chip Select ($\overline{S}$) Low. The instruction code for the Read OTP (ROTP) instruction is followed by a 3-byte address (A23- A0) and a dummy byte. Each bit is latched in on the rising edge of Serial Clock (C).

Then the memory contents at that address are shifted out on Serial Data output (DQ1).

Each bit is shifted out at the maximum frequency, fCmax, on the falling edge of Serial Clock (C). The instruction sequence is shown in Figure 17.

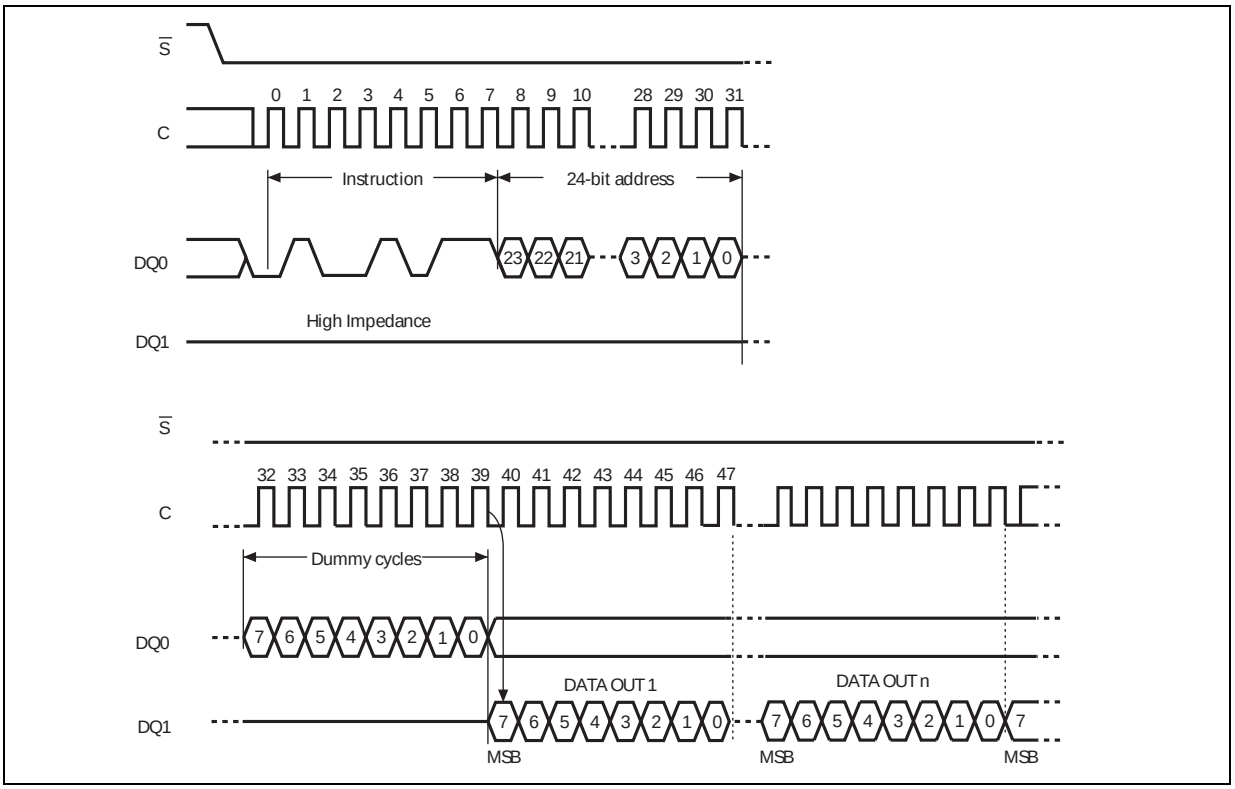
The address is automatically incremented to the next higher address after each byte of data is shifted out.

There is no rollover mechanism with the Read OTP (ROTP) instruction. This means that the Read OTP (ROTP) instruction must be sent with a maximum of 65 bytes to read. All other bytes outside the OTP area are “Don’t Care.”

The Read OTP (ROTP) instruction is terminated by driving Chip Select ($\overline{S}$) High. Chip Select ($\overline{S}$) can be driven High at any time during data output. Any Read OTP (ROTP) instruction issued while an Erase, Program or Write cycle is in progress, is rejected without having any effect on the cycle that is in progress.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 32. Read OTP instruction and data-out sequence



9.1.16 Write Enable (WREN)

The Write Enable (WREN) instruction sets the Write Enable Latch (WEL) bit.

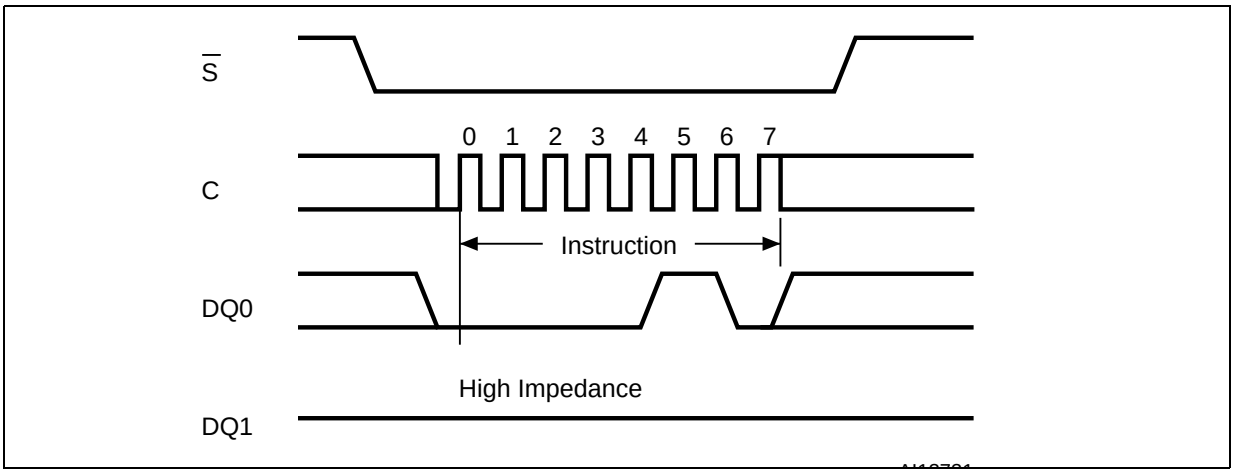
The Write Enable Latch (WEL) bit must be set prior to every Program, Erase, or Write instruction:

Page Program (PP), Dual Input Fast Program (DIFP), Dual Input Extended Fast Program (DIEFP), Quad Input Fast Program (QIFP), Quad Input Extended Fast Program (QIEFP), Program OTP (POTP), Write to Lock Register (WRLR), Subsector Erase (SSE), Sector Erase (SE), Bulk Erase (BE), Write Status Register (WRSR), Write Configuration Register (WRCR), Write Enhanced Configuration Register (WRECR), and Write Extended Address Register (WREAR), Enter 4-byte address mode (EN4BYTEADDR), Exit 4-byte address mode (EX4BYTEADDR), and Write NV Configuration Register (WRNVCR) instruction.

The Write Enable (WREN) instruction is entered by driving Chip Select (S) Low, sending the instruction code, and then driving Chip Select (S) High.

At the end of the POR sequence the WEL bit is low, so the next modify instruction can be accepted.

Figure 33. Write Enable instruction sequence

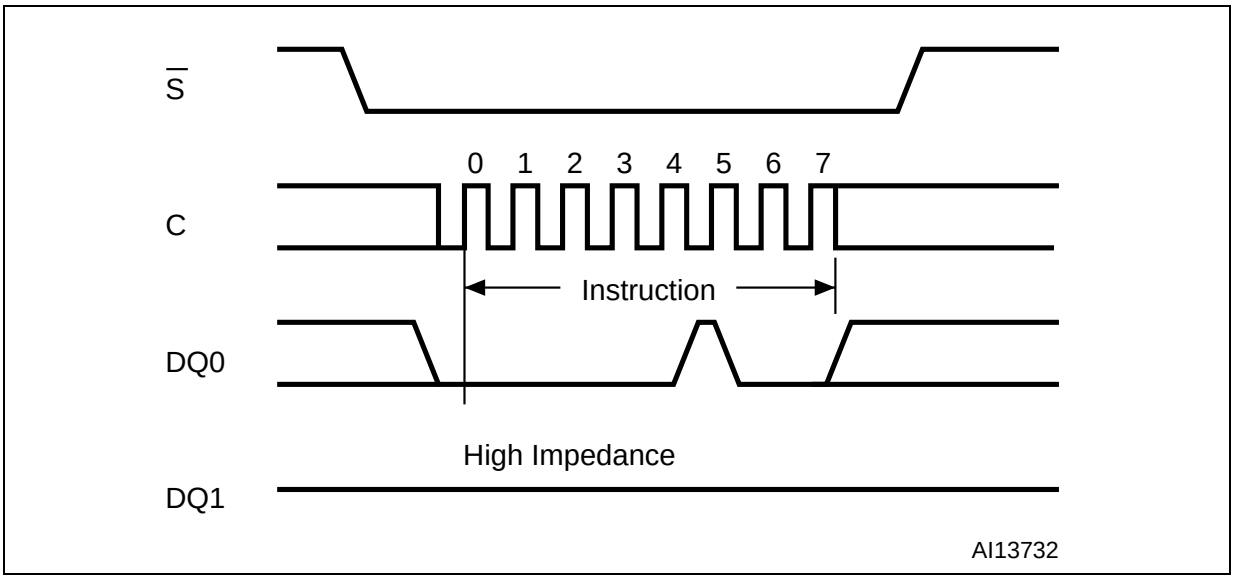


9.1.17 Write Disable (WRDI)

The Write Disable (WRDI) instruction (Figure 9) resets the Write Enable Latch (WEL) bit. The Write Disable (WRDI) instruction is entered by driving Chip Select ($\overline{S}$) Low, sending the instruction code, and then driving Chip Select ($\overline{S}$) High. The Write Enable Latch (WEL) bit is reset under the following conditions:

- Power-up
- Write Disable (WRDI) instruction completion
- Write Status Register (WRSR) instruction completion
- Write lo Lock Register (WRLR) instruction completion
- Write Non Volatile Configuration Register (WRNVCR) instruction completion
- Write Volatile Configuration Register (WRVCR) instruction completion
- Write Volatile Enhanced Configuration Register (WRVECR) instruction completion
- Page Program (PP) instruction completion
- Dual Input Fast Program (DIFP) instruction completion
- Dual Input Extended Fast Program (DIEFP) instruction completion
- Quad Input Fast Program (QIFP) instruction completion
- Quad Input Extended Fast Program (QIEFP) instruction completion
- Program OTP (POTP) instruction completion
- Subsector Erase (SSE) instruction completion
- Sector Erase (SE) instruction completion
- Bulk Erase (BE) instruction completion
- Write Extended Address Register (WREAR) instruction completion
- Enter 4-Byte Address Mode (EN4BYTEADDR) instruction completion
- Exit 4-Byte Address Mode (EX4BYTEADDR) instruction completion

Figure 34. Write Disable instruction sequence



9.1.18 Page Program (PP)

The Page Program (PP) instruction allows bytes to be programmed in the memory (changing bits from 1 to 0). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Page Program (PP) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data input (DQ0). If the 8 least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without having any effects on the other bytes of the same page.

For optimized timings, it is recommended to use the Page Program (PP) instruction to program all consecutive targeted bytes in a single sequence versus using several Page Program (PP) sequences with each containing only a few bytes. See *Table 34.: AC Characteristics*.

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the Page Program (PP) instruction is not executed.

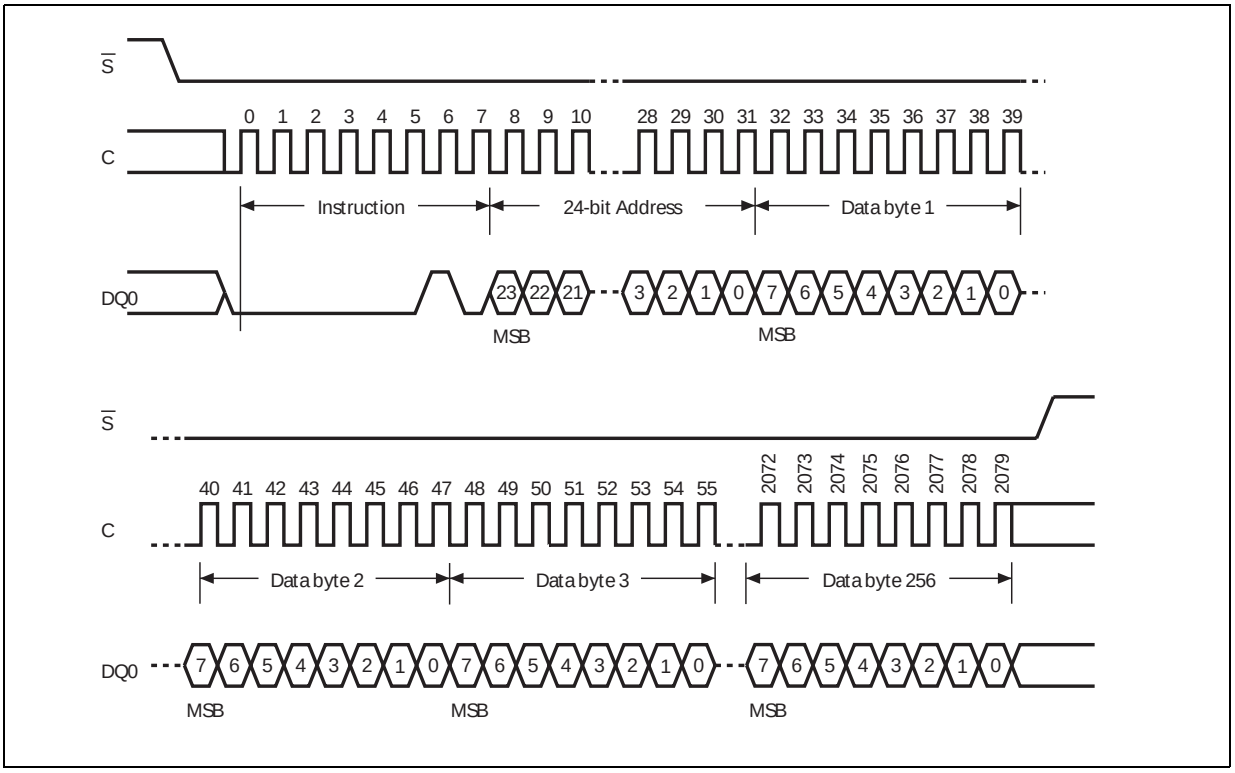
As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Page Program cycle (whose duration is top) is initiated. While the Page Program cycle is in progress, the Status Register and the Flag Status Register may be read to check if the internal modify cycle is finished. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Page Program (PP) instruction applied to a page which is protected by the Block Protect (BP3,BP2, BP1, BP0 and TB) bits is not executed.

Page Program cycle can be paused by mean of Program/Erase Suspend (PES) instruction and resumed by mean of Program/Erase Resume (PER) instruction.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 35. Page Program Instruction Sequence



9.1.19 Dual Input Fast Program (DIFP)

The Dual Input Fast Program (DIFP) instruction is very similar to the Page Program (PP) instruction, except that the data are entered on two pins (pin DQ0 and pin DQ1) instead of only one. Inputting the data on two pins instead of one doubles the data transfer bandwidth compared to the Page Program (PP) instruction.

The Dual Input Fast Program (DIFP) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data input (DQ0).

If the 8 least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without having any effects on the other bytes in the same page.

For optimized timings, it is recommended to use the Dual Input Fast Program (DIFP) instruction to program all consecutive targeted bytes in a single sequence rather to using several Dual Input Fast Program (DIFP) sequences each containing only a few bytes. See *Table 34.: AC Characteristics*.

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the Dual Input Fast Program (DIFP) instruction is not executed.

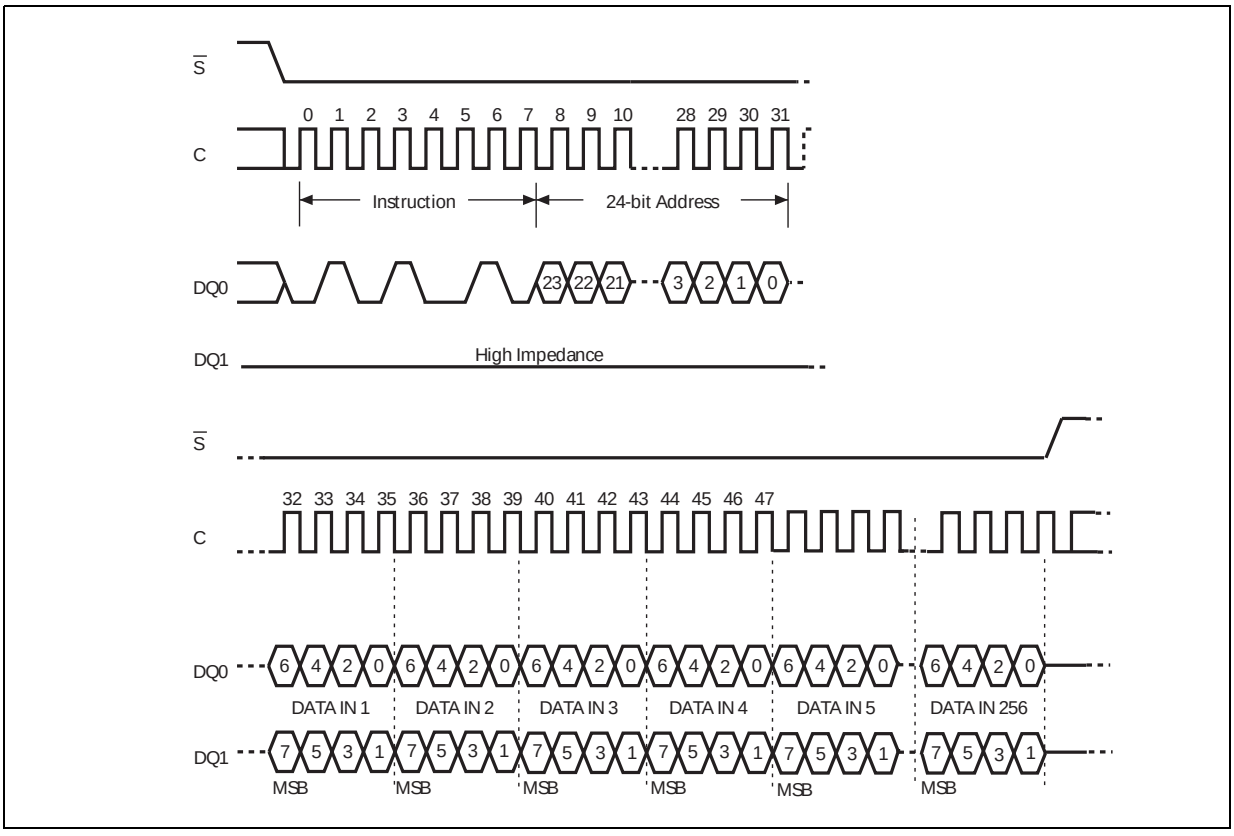
As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Page Program cycle (whose duration is top) is initiated. While the Dual Input Fast Program (DIFP) cycle is in progress, the Status Register and the Flag Status Register may be read to check if the internal modify cycle is finished. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Dual Input Fast Program (DIFP) instruction applied to a page that is protected by the Block Protect (BP3, BP2, BP1, BP0 and TB) bits is not executed.

Dual Input Fast Program cycle can be paused by mean of Program/Erase Suspend (PES) instruction and resumed by mean of Program/Erase Resume (PER) instruction.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 36. Dual Input Fast Program instruction sequence

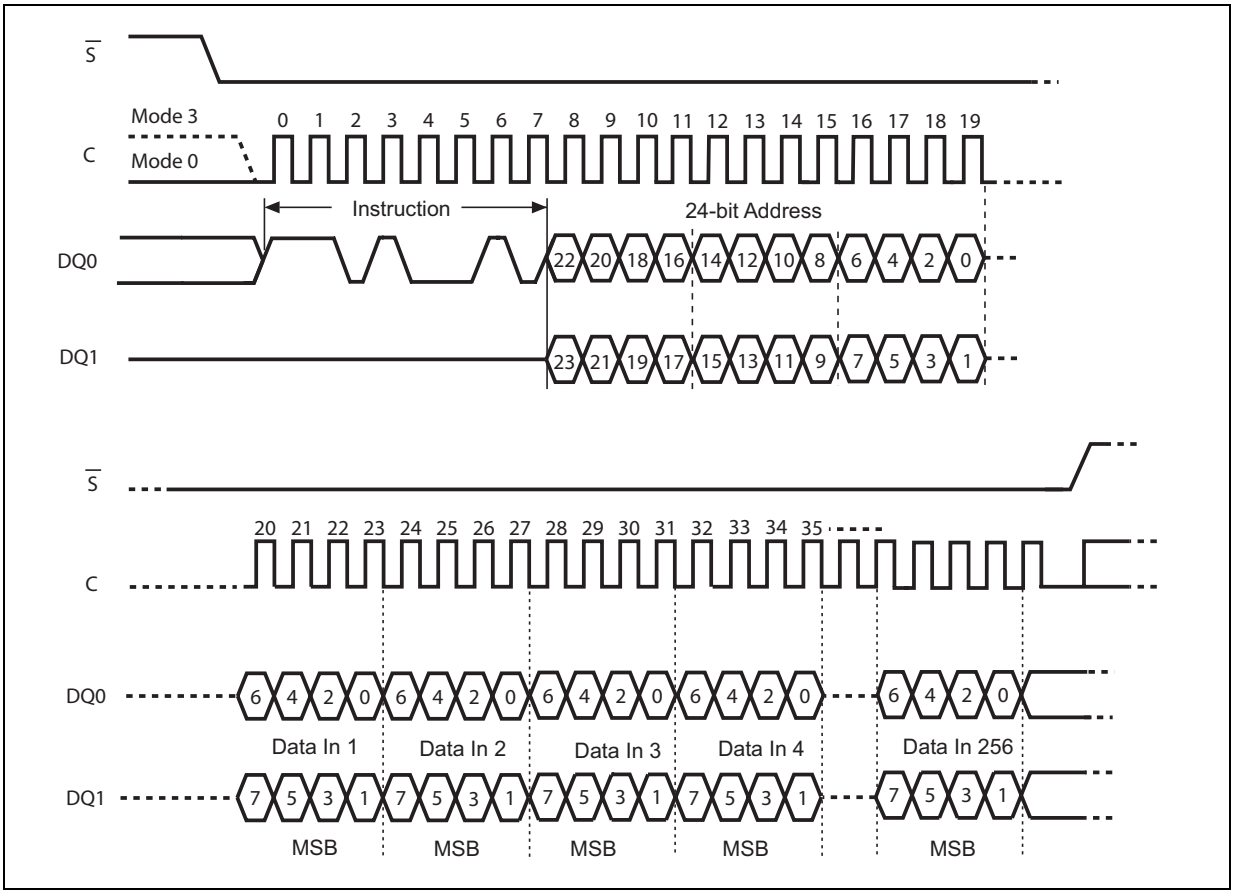


9.1.20 Dual Input Extended Fast Program

The Dual Input Extended Fast Program (DIEFP) instruction is very similar to the Dual Input Fast Program (DIFP), except that the address bits are shifted in on two pins (pin DQ0 and pin DQ1) instead of only one.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 37. Dual Input Extended Fast Program instruction sequence



9.1.21 Quad Input Fast Program

The Quad Input Fast Program (QIFP) instruction is very similar to the Dual Input Fast Program (DIFP) instruction, except that the data are entered on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/ (DQ3) instead of only two. Inputting the data on four pins instead of two doubles the data transfer bandwidth compared to the Dual Input Fast Program (DIFP) instruction.

The Quad Input Fast Program (QIFP) instruction is entered by driving Chip Select (S) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data input (DQ0).

If the 8 least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same

page (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without having any effects on the other bytes in the same page.

For optimized timings, it is recommended to use the Quad Input Fast Program (QIFP) instruction to program all consecutive targeted bytes in a single sequence rather to using several Quad Input Fast Program (QIFP) sequences each containing only a few bytes See *Table 34.: AC Characteristics*.

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the Quad Input Fast Program (QIFP) instruction is not executed.

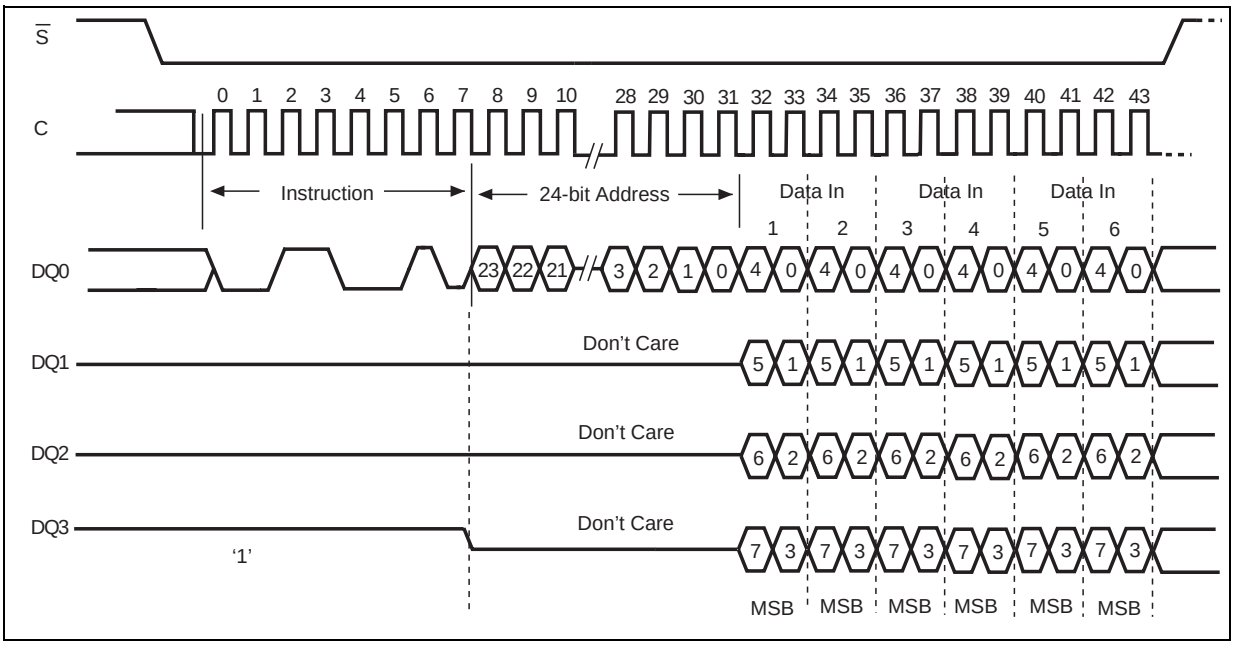
As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Page Program cycle (whose duration is tPP) is initiated. While the Quad Input Fast Program (QIFP) cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Page Program cycle, and 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Quad Input Fast Program (QIFP) instruction applied to a page that is protected by the Block Protect (BP3, BP2, BP1, BP0 and TB) bits is not executed.

A Quad Input Fast Program cycle can be paused by mean of Program/Erase Suspend (PES) instruction and resumed by mean of Program/Erase Resume (PER) instruction.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 38. Quad Input Fast Program Instruction Sequence

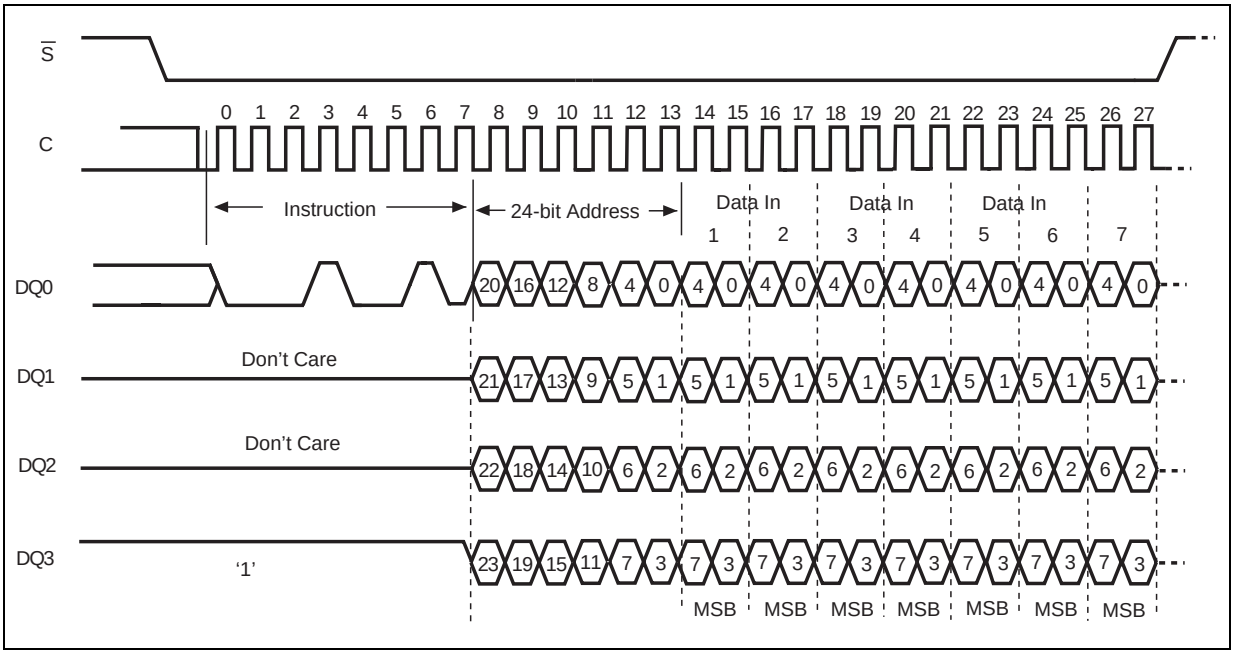


9.1.22 Quad Input Extended Fast Program

The Quad Input Extended Fast Program (QIEFP) instruction is very similar to the Quad Input Extended Fast Program (QIFP), except that the address bits are shifted in on four pins (pin DQ0, pin DQ1, pin W/VPP/DQ2 and pin HOLD/DQ3) instead of only one.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 39. Quad Input Extended Fast Program instruction sequence



9.1.23 Program OTP instruction (POTP)

The Program OTP instruction (POTP) is used to program at most 64 bytes to the OTP memory area (by changing bits from 1 to 0, only). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL) bit.

The Program OTP instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction opcode, three address bytes and at least one data byte on Serial Data input (DQ0). Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the Program OTP instruction is not executed.

There is no rollover mechanism with the Program OTP (POTP) instruction. This means that the Program OTP (POTP) instruction must be sent with a maximum of 65 bytes to program, once all 65 bytes have been latched in, any following byte will be discarded.

As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Page Program cycle (whose duration is tPP) is initiated. While the Program OTP cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Program OTP cycle, and it is 0 when it is completed. At some unspecified time before the cycle is complete, the Write Enable Latch (WEL) bit is reset. To lock the OTP memory:

Bit 0 of the OTP control byte, that is byte 64, is used to permanently lock the OTP memory array.

- When bit 0 of byte 64 = '1', the 64 bytes of the OTP memory array can be programmed.
- When bit 0 of byte 64 = '0', the 64 bytes of the OTP memory array are read-only and cannot be programmed anymore.

Once a bit of the OTP memory has been programmed to '0', it can no longer be set to '1'.

Therefore, as soon as bit 0 of byte 64 (control byte) is set to '0', the 64 bytes of the OTP memory array become read-only in a permanent way.

Any Program OTP (POTP) instruction issued while an Erase, Program or Write cycle is in progress is rejected without having any effect on the cycle that is in progress.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 40. Program OTP instruction sequence

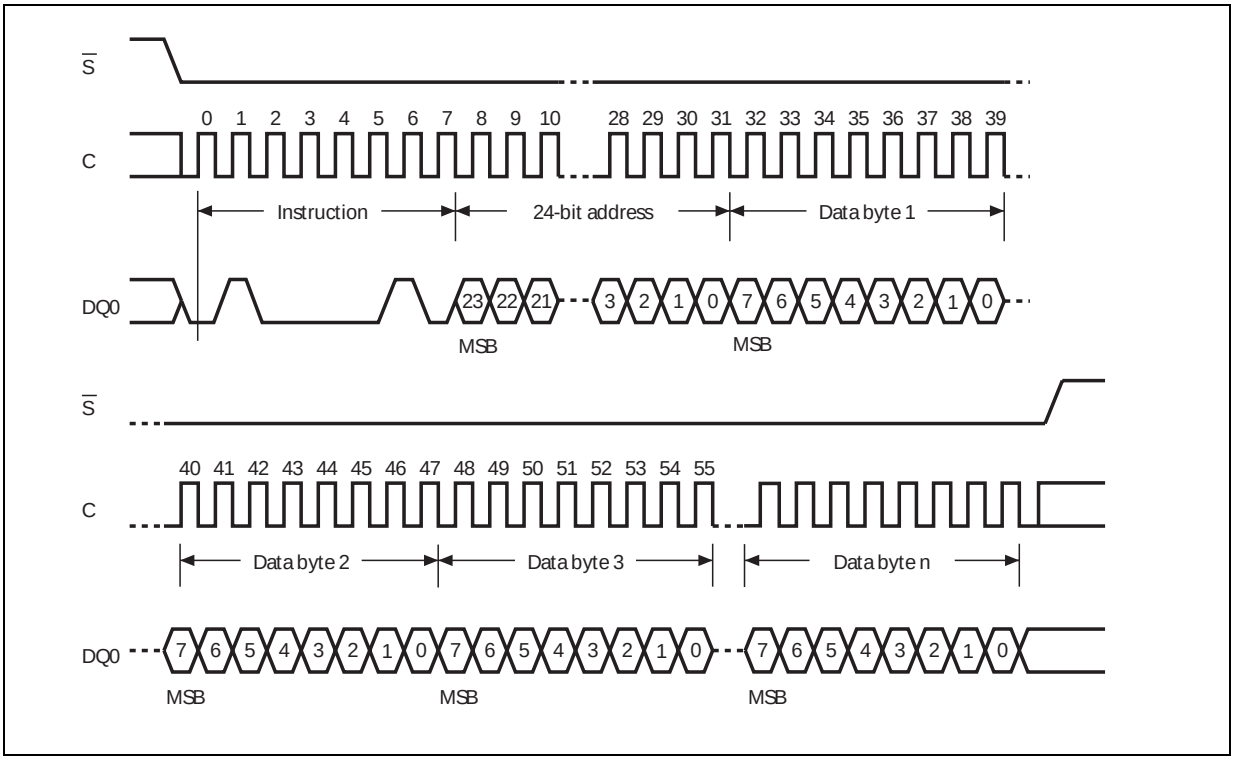
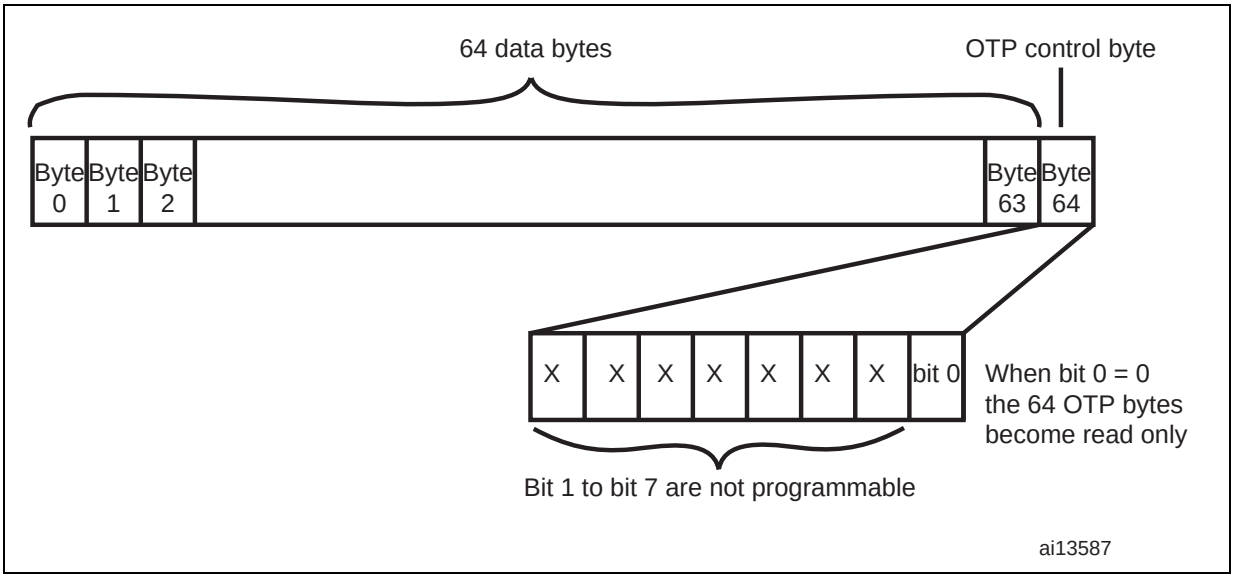


Figure 41. How to permanently lock the OTP bytes



9.1.24 Subsector Erase (SSE)

The Subsector Erase (SSE) instruction sets to '1' (FFh) all bits inside the chosen subsector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Subsector Erase (SSE) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code, and three address bytes on Serial Data input (DQ0). Any address inside the subsector is a valid address for the Subsector Erase (SSE) instruction. Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

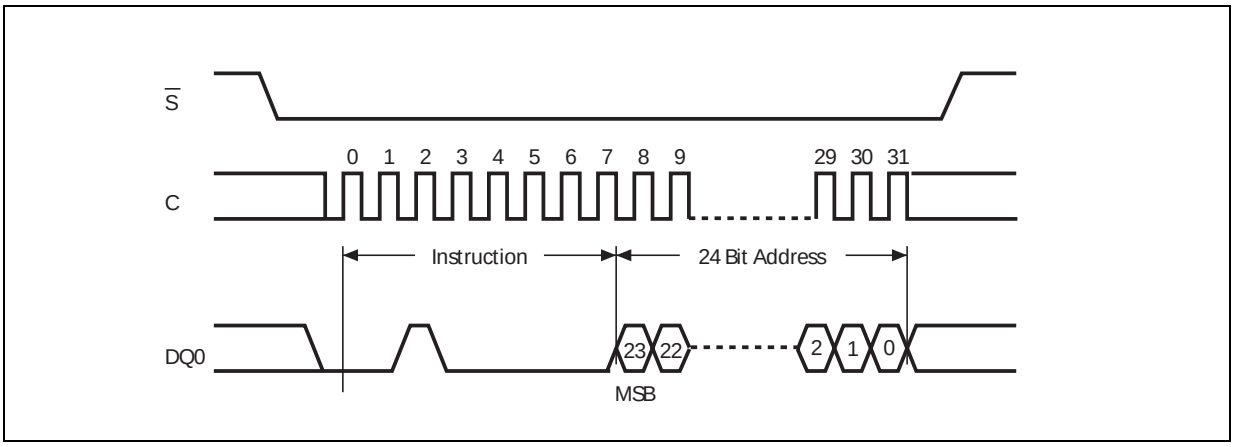
Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Subsector Erase (SSE) instruction is not executed. As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Subsector Erase cycle (whose duration is tSSE) is initiated. While the Subsector Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Subsector Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is complete, the Write Enable Latch (WEL) bit is reset.

A Subsector Erase (SSE) instruction issued to a sector that is hardware or software protected, is not executed.

Any Subsector Erase (SSE) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 42. Subsector Erase instruction sequence



9.1.25 Sector Erase (SE)

The Sector Erase (SE) instruction sets to '1' (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Sector Erase (SE) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code, and three address bytes on Serial Data input (DQ0). Any address inside the sector is a valid address for the Sector Erase (SE) instruction. Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

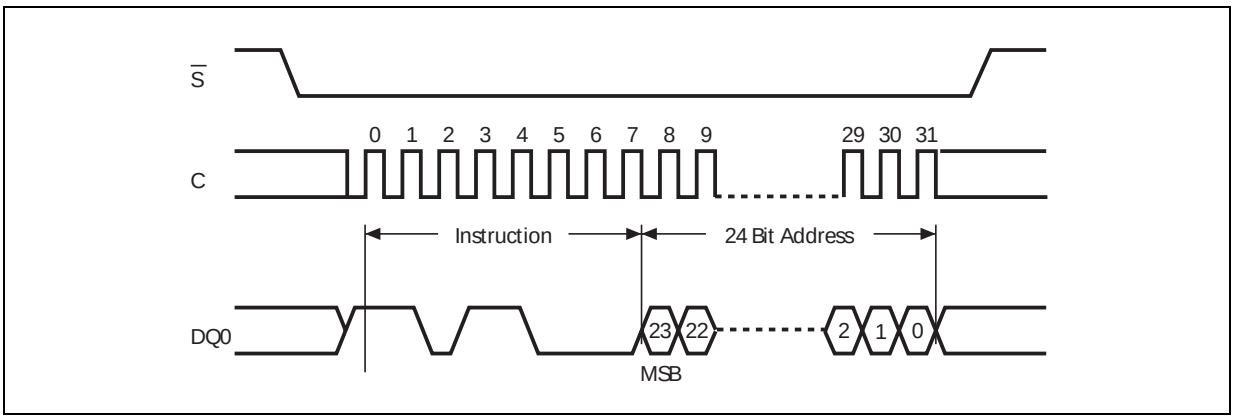
Chip Select ($\overline{S}$) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Sector Erase (SE) instruction is not executed. As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Sector Erase cycle (whose duration is tSE) is initiated. While the Sector Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Sector Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Sector Erase (SE) instruction applied to a page which is protected by the Block Protect (BP3, BP2, BP1, BP0 and TB) bits is not executed.

A Sector Erase cycle can be paused by mean of Program/Erase Suspend (PES) instruction and resumed by mean of Program/Erase Resume (PER) instruction.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 43. Sector Erase instruction sequence



9.1.26 Bulk Erase (BE)

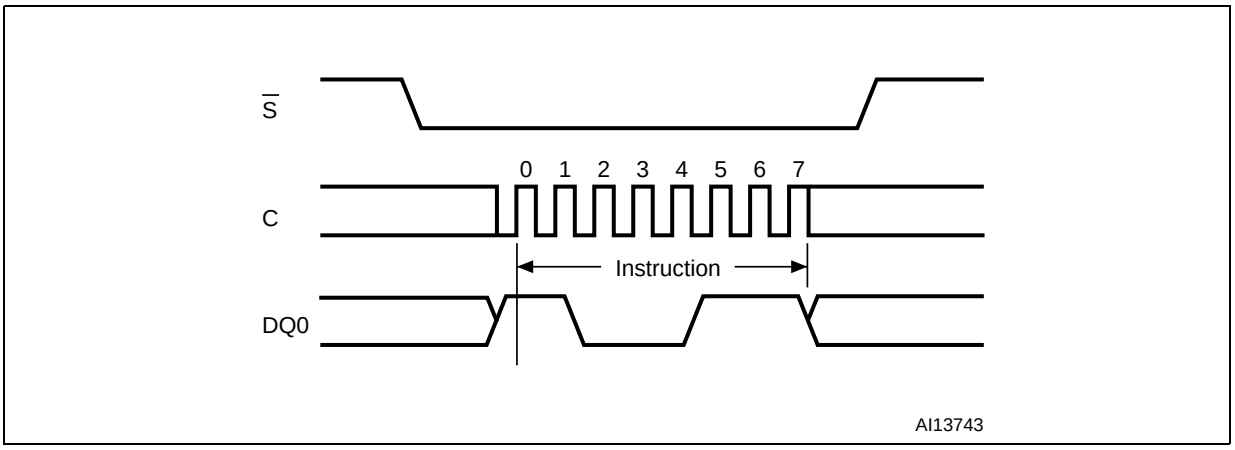
The Bulk Erase (BE) instruction sets all bits to '1' (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Bulk Erase (BE) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code on Serial Data input (DQ0). Chip Select ($\overline{S}$) must be driven Low for the entire duration of the sequence.

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Bulk Erase instruction is not executed. As soon as Chip Select ($\overline{S}$) is driven High, the self-timed Bulk Erase cycle (whose duration is tBE) is initiated. While the Bulk Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Bulk Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

The Bulk Erase (BE) instruction is executed only if all Block Protect (BP3, BP2, BP1, BP0) bits are 0. The Bulk Erase (BE) instruction is ignored if one, or more, sectors are protected.

Figure 44. Bulk Erase instruction sequence



9.1.27 Program/Erase Suspend

The Program/Erase Suspend instruction allows the controller to interrupt a Program or an Erase instruction, in particular: Sector Erase, Subsector Erase, Page Program, Dual Input Page Program, Dual Input Extended Page program, Quad Input Page Program and Quad Input Extended Page program can be suspended and erased.

Note: Bulk Erase, Write Non Volatile Configuration register and Program OTP cannot be suspended.

After a Program/Erase Suspend instruction the bit 2 of the Flag Status register is immediately set to 1 and, after a latency time, both the WIP bit of the Status Register and the Program/Erase controller bit (Not WIP) of the Flag Status Register are cleared (to 0 and to 1 respectively).

The Suspended state is reset if a power-off is performed or after resume. After a sector erase instruction has been suspended, another erase instruction is not allowed; however, it is possible to perform program and reading instructions on all the sectors except the one whose erase cycle is suspended. Any read instruction issued on this sector outputs Don't Care data.

After a subsector erase instruction has been suspended, neither an erase instruction or a program instruction is allowed; only a read instruction is allowed on all sectors except the one containing the subsector whose erase cycle is suspended. Any read instruction issued on this sector outputs Don't Care data.

After a program instruction has been suspended, neither a program instruction or an erase instructions is allowed; however, it is possible to perform a read instruction on all pages except the one whose program cycle is suspended. Any read instruction issued on this page outputs Don't Care data.

It's possible to nest a suspend instruction inside another suspended one just once, meaning that it's possible for example to send to the device an erase instruction, then suspend it, then send a program instruction and in the end suspend it as well. In this case the next Program/Erase Resume Instruction resumes the more recent suspended modify cycles, another Program/Erase Resume Instruction is need to resume also the former one.

Table 19. Suspend Parameters

Parameter	Condition	Typ	Max	Unit	Note
Erase to Suspend	Sector Erase or Erase Resume to Erase Suspend	700		µs	Timing not internally controlled
Program to Suspend	Program Resume to Program Suspend	5		µs	Timing not internally controlled
SSErase to Suspend	Sub Sector Erase or Sub Sector Erase Resume to Erase Suspend	50		µs	Timing not internally controlled
Suspend Latency	Program	7		µs	Any Read instruction accepted
Erase	Sub Sector Erase	15		µs	Any Read instruction accepted
Erase	Erase	15		µs	Any instruction accepted but DP, SE, SSE, BE, WRSR, WRNVCR, POTP

Note: Device states are shown in Table 20.: Operations Allowed / Disallowed During Device States. The device can be in only one state at a time, such as Standby, Program, Erase, and so on.

Table 20. Operations Allowed / Disallowed During Device States

Operation	Device States and Sector (Same/Other) in Which Operation is Allowed/Disallowed (Yes/No)											
	Standby State		Program State		Erase State (SE/SSE)		Subsector Erase Suspended State		Program Suspended State		Erase Suspended State	
	Sector		Sector		Sector		Sector		Sector		Sector	
	Same	Other	Same	Other	Same	Other	Same	Other	Same	Other	Same	Other
All Reads except RDSR / RDFSR	Yes	Yes	No	No	No	No	Yes(1)	Yes	Yes	Yes	Yes(1)	Yes
Array Program: PP / DIFP / QIFP / DIEFP / QIEFP	Yes	Yes	No	No	No	No	No	No	No	No	No	Yes
Sector Erase	Yes	Yes	No	No	No	No	No	No	No	No	No	No
Sub-Sector Erase	Yes	Yes	No	No	No	No	No	No	No	No	No	No
WRLR / POTP / BE / WRSR / WRNVCR	Yes		No		No		No		No		No	
WVCR / WVECR	Yes		No		No		Yes		Yes		Yes	
RDSR / RDFSR	Yes		Yes		Yes		Yes		Yes		Yes	
Program / Erase Suspend	No		Yes		Yes		No		No		No	

1. The Read operation is accepted but the data output is not guaranetted until the program or erase has completed.

9.1.28 Program/Erase Resume

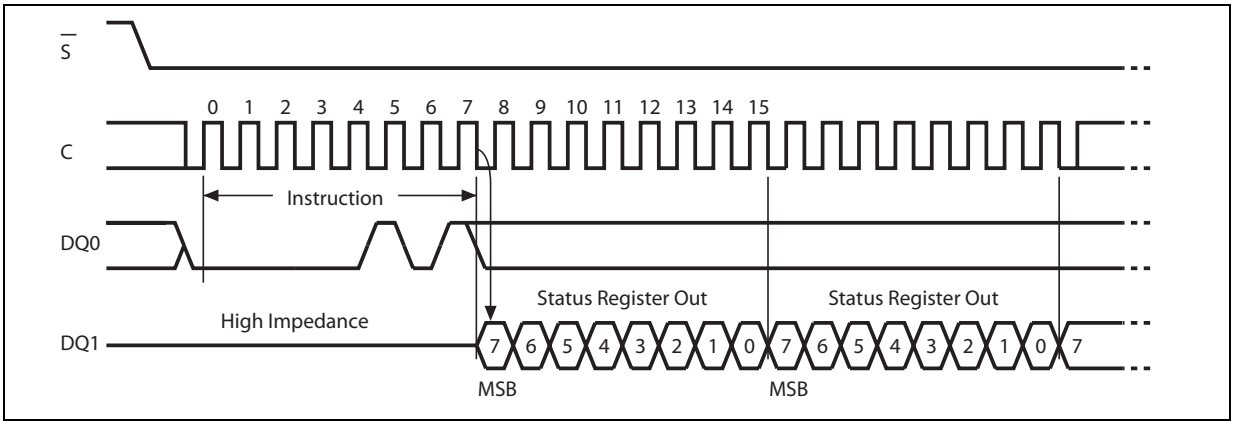
After a Program/Erase suspend instruction, a Program/Erase Resume instruction is required to continue performing the suspended Program or Erase sequence. Program/Erase Resume instruction is ignored if the device is not in a Program/Erase Suspended status. The WIP bit of the Status Register and Program/Erase controller bit (Not WIP) of the Flag Status Register both switch to the busy state (1 and 0 respectively) after Program/Erase Resume instruction until the Program or Erase sequence is completed.

In this case the next Program/Erase Resume Instruction resumes the more recent suspended modify cycles, another Program/Erase Resume Instruction is need to resume also the former one.

9.1.29 Read Status Register (RDSR)

The Read Status Register (RDSR) instruction allows the Status Register to be read. The Status Register may be read at any time, even while a Program, Erase or Write Status Register cycle is in progress. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit (or the Program/Erase controller bit of the Flag Status Register) before sending a new instruction to the device. It is also possible to read the Status Register continuously, as shown here.

Figure 45. Read Status Register instruction sequence



9.1.30 Write status register (WRSR)

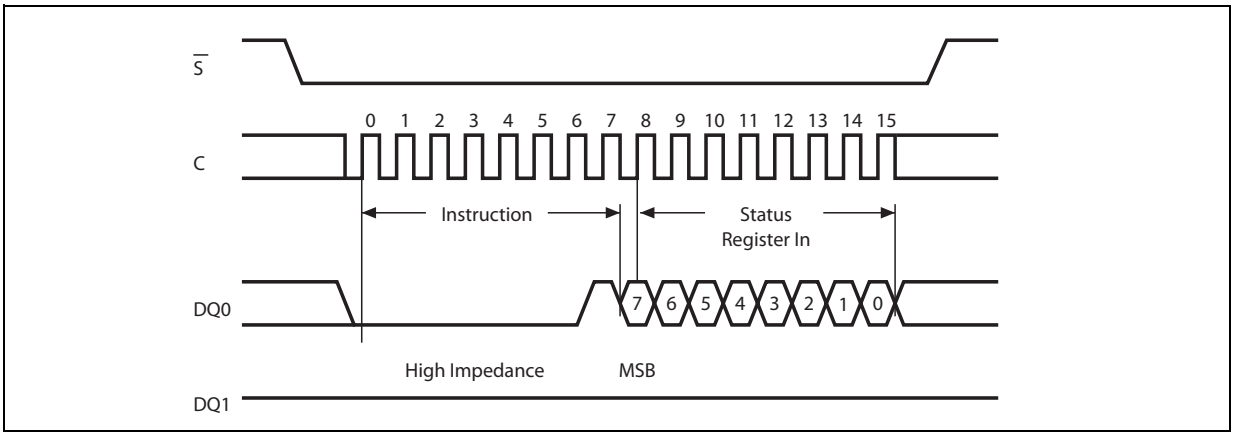
The write status register (WRSR) instruction allows new values to be written to the status register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

The write status register (WRSR) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code and the data byte on serial data input (DQ0). The write status register (WRSR) instruction has no effect on b1 and b0 of the status register.

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the data byte has been latched in. If not, the write status register (WRSR) instruction is not executed. As soon as Chip Select ($\overline{S}$) is driven High, the self-timed write status register cycle (whose duration is tow) is initiated. While the write status register cycle is in progress, the status register may still be read to check the value of the write in progress (WIP) bit. The write in progress (WIP) bit is 1 during the self-timed write status register cycle, and is 0 when it is completed. When the cycle is completed, the write enable latch (WEL) is reset.

The write status register (WRSR) instruction allows the user to change the values of the block protect (BP3, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Table 3. The write status register (WRSR) instruction also allows the user to set and reset the status register write disable (SRWD) bit in accordance with the Write Protect ($\overline{W}$/VPP) signal. The status register write disable (SRWD) bit and Write Protect ($\overline{W}$/VPP) signal allow the device to be put in the hardware protected mode (HPM). The write status register (WRSR) instruction is not executed once the hardware protected mode (HPM) is entered.

Figure 46. Write Status Register instruction sequence



The protection modes of the device are summarized below.

When the Status Register Write Disable (SRWD) bit of the Status Register is 0 (its initial delivery state), it is possible to write to the Status Register provided that the Write Enable Latch (WEL) bit has previously been set by a Write Enable (WREN) instruction, regardless of the whether Write Protect ($\overline{W}$/VPP) is driven High or Low.

When the Status Register Write Disable (SRWD) bit of the Status Register is set to '1', two cases need to be considered, depending on the state of Write Protect ($\overline{W}$/VPP):

- If Write Protect ($\overline{W}$/VPP) is driven High, it is possible to write to the Status Register provided that the Write Enable Latch (WEL) bit has previously been set by a Write Enable (WREN) instruction.
- If Write Protect ($\overline{W}$/VPP) is driven Low, it is not possible to write to the Status Register even if the Write Enable Latch (WEL) bit has previously been set by a Write Enable (WREN) instruction (attempts to write to the Status Register are rejected, and are not accepted for execution). As a consequence, all the data bytes in the memory area that are software protected (SPM) by the Block Protect (BP3, BP2, BP1, BP0) bits of the Status Register, are also hardware protected against data modification.

Regardless of the order of the two events, the Hardware Protected mode (HPM) can be entered in either of the following ways:

- setting the Status Register Write Disable (SRWD) bit after driving Write Protect ($\overline{W}$/VPP) Low
- driving Write Protect ($\overline{W}$/VPP) Low after setting the Status Register Write Disable (SRWD) bit.

The only way to exit the Hardware Protected mode (HPM) once entered is to pull Write Protect ($\overline{W}$/VPP) High.

If Write Protect ($\overline{W}$/VPP) is permanently tied High, the Hardware Protected mode (HPM) can never be activated, and only the Software Protected mode (SPM), using the Block Protect (BP3, BP2, BP1, BP0) bits of the Status Register, can be used.

Table 21. Protection modes

$\overline{W}$ / VPP Signal	SRWD bit	Mode	Write protection of the status register	Memory content	
				Protected area (1)	Unprotected area (1)
1	0	Software protected (SPM)	Status register is writeable, if the WREN instruction has set the WEL bit. The values in the SRWD, TB, BP3, BP2, BP1, and BP0 bits can be changed.	Protected against PP, DIFP, DIEFP, QIFP, QIEFP, SSE, SE and BE instructions.	Ready to accept PP, DIFP, DIEFP, QIFP, QIEFP, SSE, and SE instructions.
0	0				
1	1				
0	1	Hardware protected (HPM)	Status Register is hardware write protected. The values in the SRWD, TB, BP3, BP2, BP1 and BP0 bits cannot be changed	PP, DIFP, DIEFP, QIFP, QIEFP, SSE, SE and BE instructions.	PP, DIFP, DIEFP, QIFP, QIEFP, SSE, and SE instructions.

1. As defined by the values in the Block Protect (TB, BP3, BP2, BP1, BP0) bits of the Status Register, as shown in *Table 3: Status register format*.

9.1.31 Read Lock Register (RDLR)

The device is first selected by driving Chip Select ($\overline{S}$) Low. The instruction code for the Read Lock Register (RDLR) instruction is followed by a 3-byte address (A23-A0) pointing to any location inside the concerned sector. Each address bit is latched-in during the rising edge of Serial Clock (C). Then the value of the Lock Register is shifted out on Serial Data output (DQ1), each bit being shifted out, at a maximum frequency fC, during the falling edge of Serial Clock (C).

The Read Lock Register (RDLR) instruction is terminated by driving Chip Select ($\overline{S}$) High at any time during data output.

Any Read Lock Register (RDLR) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 47. Read Lock Register instruction and data-out sequence

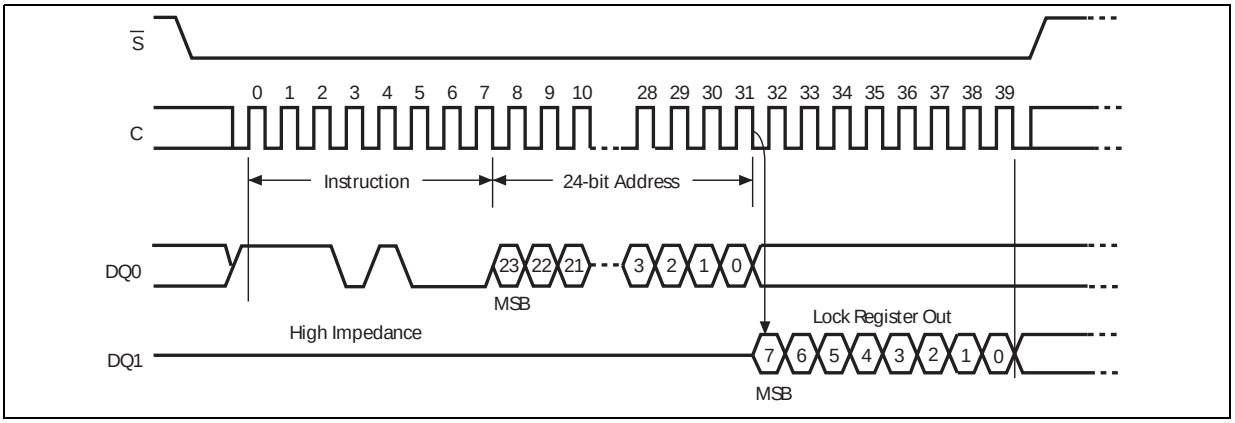


Table 22. Lock Register out(1)

Bit	Bit name	Value	Function
b7-b2	Reserved		
b1	Sector Lock Down	‘1’	The Write Lock and Lock Down bits cannot be changed. Once a ‘1’ is written to the Lock Down bit it cannot be cleared to ‘0’, except by a power-up.
		‘0’	The Write Lock and Lock Down bits can be changed by writing new values to them.
b0	Sector Write Lock	‘1’	Write, Program and Erase operations in this sector will not be executed. The memory contents will not be changed.
		‘0’	Write, Program and Erase operations in this sector are executed and will modify the sector contents.

1. Values of (b1, b0) after power-up are defined in *Section 7: Protection modes*.

9.1.32 Write to Lock Register (WRLR)

The Write to Lock Register (WRLR) instruction allows bits to be changed in the Lock Registers. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Write to Lock Register (WRLR) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code, three address bytes (pointing to any address in the targeted sector and one data byte on Serial Data input (DQ0). Chip Select ($\overline{S}$) must be driven High after the eighth bit of the data byte has been latched in, otherwise the Write to Lock Register (WRLR) instruction is not executed.

Lock Register bits are volatile, and therefore do not require time to be written. When the Write to Lock Register (WRLR) instruction has been successfully executed, the Write Enable Latch (WEL) bit is reset after a delay time less than tSHSL minimum value.

Any Write to Lock Register (WRLR) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress. If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 48. Write to Lock Register instruction sequence

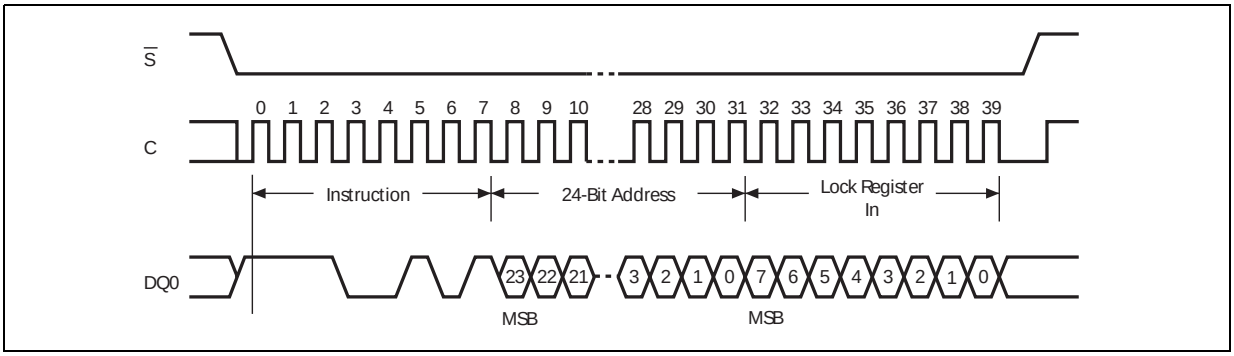


Table 23. Lock Register in(1)

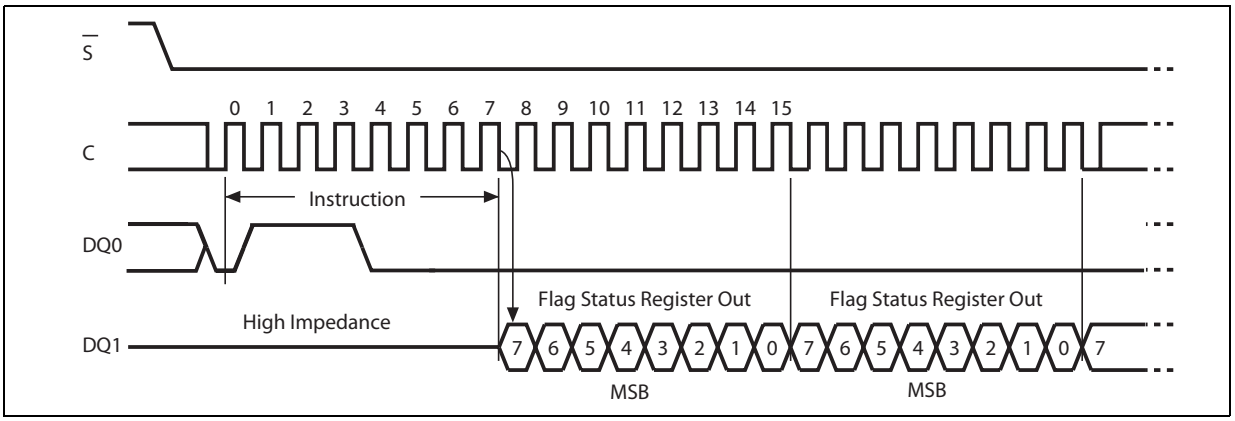
Sector	Bit	Value
All sectors	b7-b2	'0'
	b1	Sector Lock Down bit value (refer to *Table 22*)
	b0	Sector Write Lock bit value (refer to *Table 22*)

1. Values of (b1, b0) after power-up are defined in *Section 7: Protection modes*.

9.1.33 Read Flag Status Register

The Read Flag Status Register (RFSR) instruction allows the Flag Status Register to be read. The Status Register may be read at any time, even while a Program, Erase. When one of these cycles is in progress, it is recommended to check the P/E Controller bit (Not WIP) bit before sending a new instruction to the device. It is also possible to read the Flag Register continuously, as shown here.

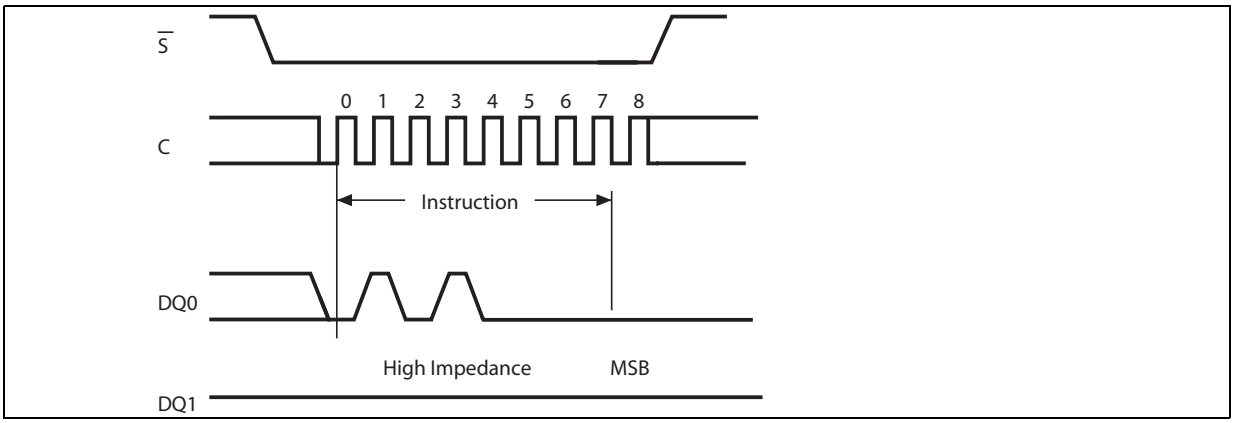
Figure 49. Read Flag Status Register instruction sequence



9.1.34 Clear Flag Status Register

The Clear Flag Status Register (CLFSR) instruction reset the error Flag Status Register bits (Erase Error bit, Program Error bit, VPP Error bit, Protection Error bit). It is not necessary to set the WEL bit before the Clear Flag Status Register instruction is executed. The WEL bit will be unchanged after this command is executed.

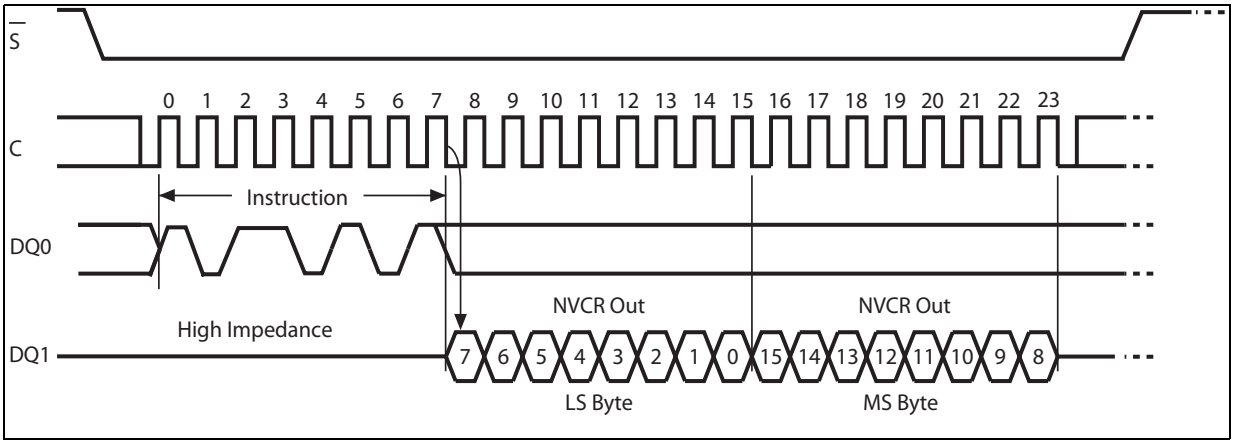
Figure 50. Clear Flag Status Register instruction sequence



9.1.35 Read NV Configuration Register

The Read Non Volatile Configuration Register (RDNVCR) instruction allows the Non Volatile Configuration Register to be read.

Figure 51. Read NV Configuration Register instruction sequence



9.1.36 Write NV Configuration Register

The Write Non Volatile Configuration register (WRNVCR) instruction allows new values to be written to the Non Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

The Write Non Volatile Configuration register (WRNVCR) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code and the data bytes on serial data input (DQ0).

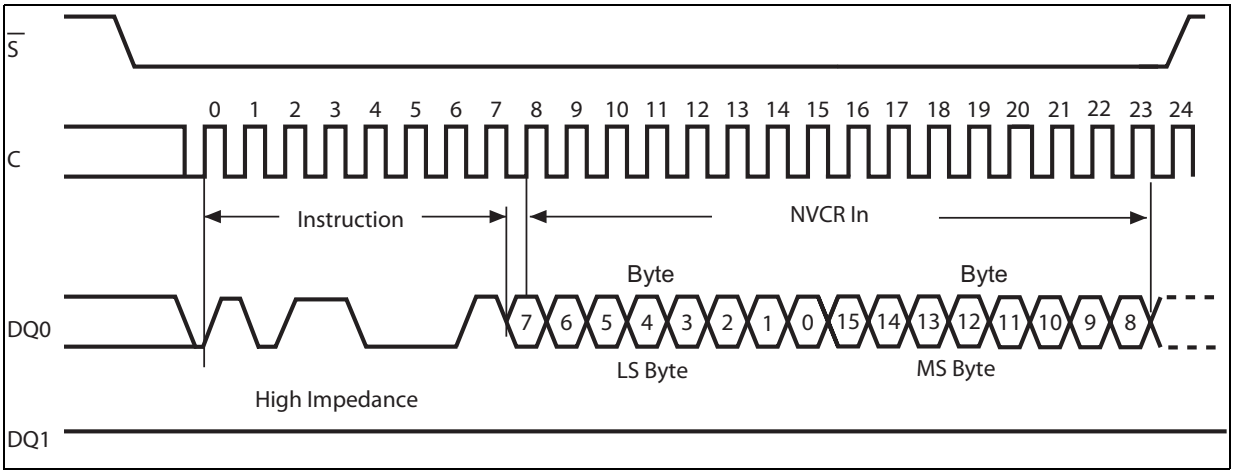
Chip Select ($\overline{S}$) must be driven High after the 16th bit of the data bytes has been latched in. If not, the Write Non Volatile Configuration register (WRNVCR) instruction is not executed. As soon as Chip Select ($\overline{S}$) is driven High, the self-timed write NV configuration register cycle (whose duration is tnvcr) is initiated.

While the Write Non Volatile Configuration register cycle is in progress, it is possible to monitor the end of the process by polling status Register write in progress (WIP) bit or the Flag Status Register Program/Erase Controller bit. The write in progress (WIP) bit is 1 during the self-timed Write Non Volatile Configuration register cycle, and is 0 when it is completed. When the cycle is completed, the write enable latch (WEL) is reset.

The Write Non Volatile Configuration register (WRNVCR) instruction allows the user to change the values of all the Non Volatile Configuration Register bits, described in *Table 4.: Non-Volatile Configuration Register*.

The Write Non Volatile Configuration Register impacts the memory behavior only after the next power on sequence.

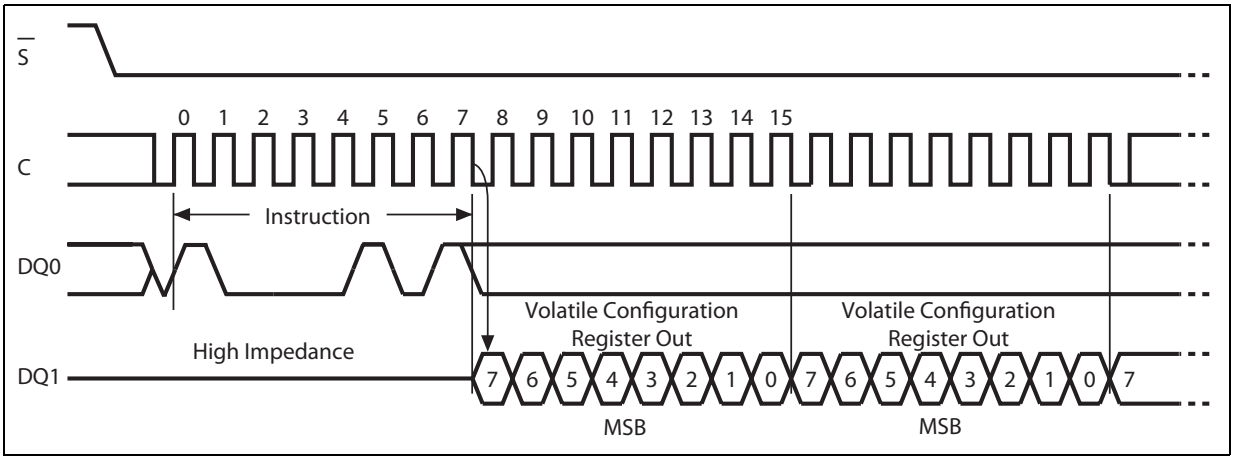
Figure 52. Write NV Configuration Register instruction sequence



9.1.37 Read Volatile Configuration Register

The Read Volatile Configuration Register (RDVCR) instruction allows the Volatile Configuration Register to be read. See *Table 6.: Volatile Configuration Register*.

Figure 53. Read Volatile Configuration Register instruction sequence



9.1.38 Write Volatile Configuration Register

The Write Volatile Configuration register (WRVCR) instruction allows new values to be written to the Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

The Write Volatile Configuration register (WRVCR) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code and the data byte on serial data input (DQ0).

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the data byte has been latched in.

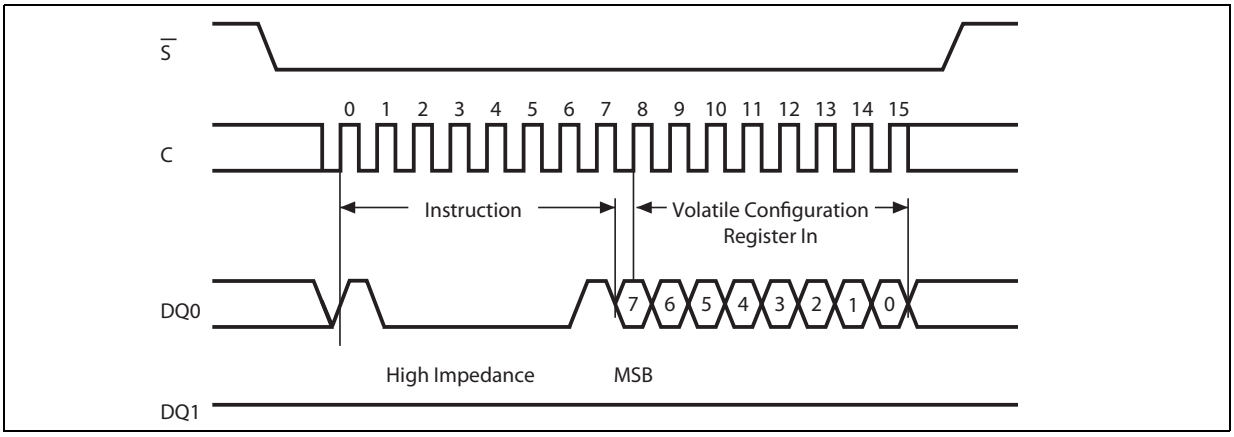
If not, the Write Volatile Configuration register (WRVCR) instruction is not executed.

When the new data are latched, the write enable latch (WEL) is reset.

The Write Volatile Configuration register (WRVCR) instruction allows the user to change the values of all the Volatile Configuration Register bits, described in *Table 6.: Volatile Configuration Register*.

The Write Volatile Configuration Register impacts the memory behavior right after the instruction is received by the device.

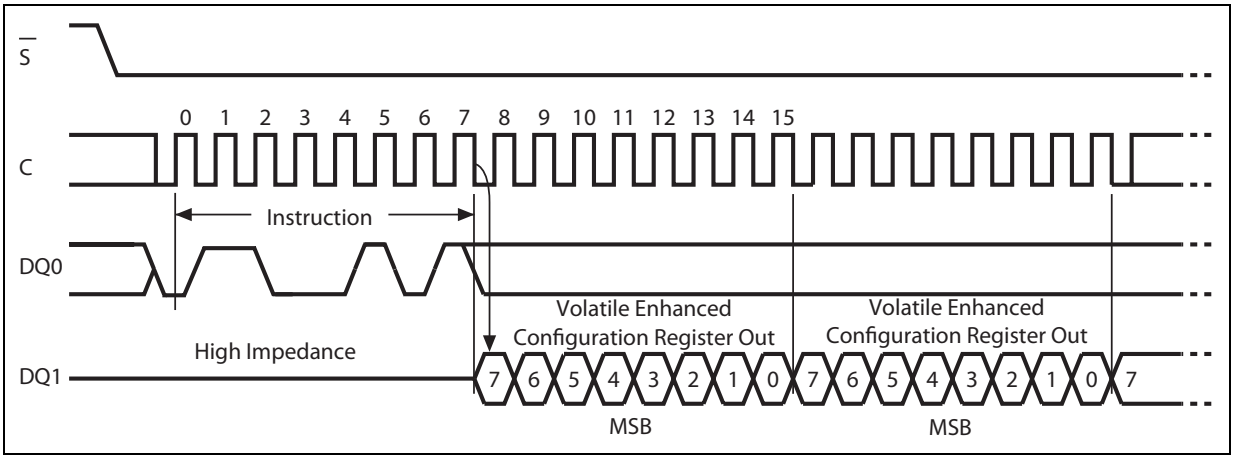
Figure 54. Write Volatile Configuration Register instruction sequence



9.1.39 Read Volatile Enhanced Configuration Register

The Read Volatile Enhanced Configuration Register (RDVECR) instruction allows the Volatile Configuration Register to be read.

Figure 55. Read Volatile Enhanced Configuration Register instruction sequence



9.1.40 Write Volatile Enhanced Configuration Register

The Write Volatile Enhanced Configuration register (WRVECR) instruction allows new values to be written to the Volatile Enhanced Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

The Write Volatile Enhanced Configuration register (WRVECR) instruction is entered by driving Chip Select ($\overline{S}$) Low, followed by the instruction code and the data byte on serial data input (DQ0).

Chip Select ($\overline{S}$) must be driven High after the eighth bit of the data byte has been latched in.

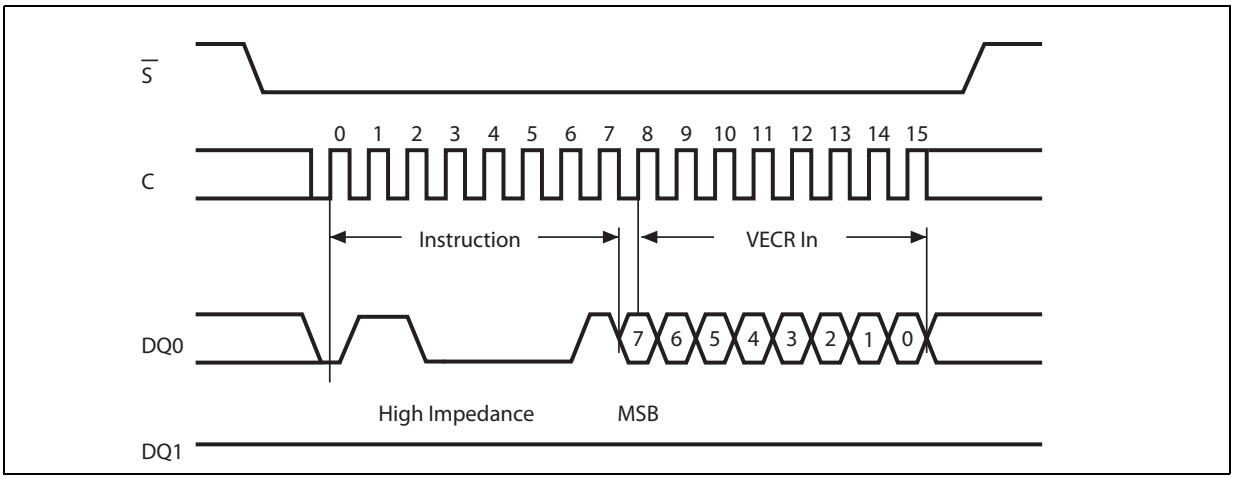
If not, the Write Volatile Enhanced Configuration register (WRVECR) instruction is not executed.

When the new data are latched, the write enable latch (WEL) is reset.

The Write Volatile Enhanced Configuration register (WRVECR) instruction allows the user to change the values of all the Volatile Enhanced Configuration Register bits, described in *Table 8.: Volatile Enhanced Configuration Register*.

The Write Volatile Enhanced Configuration Register impacts the memory behavior right after the instruction is received by the device.

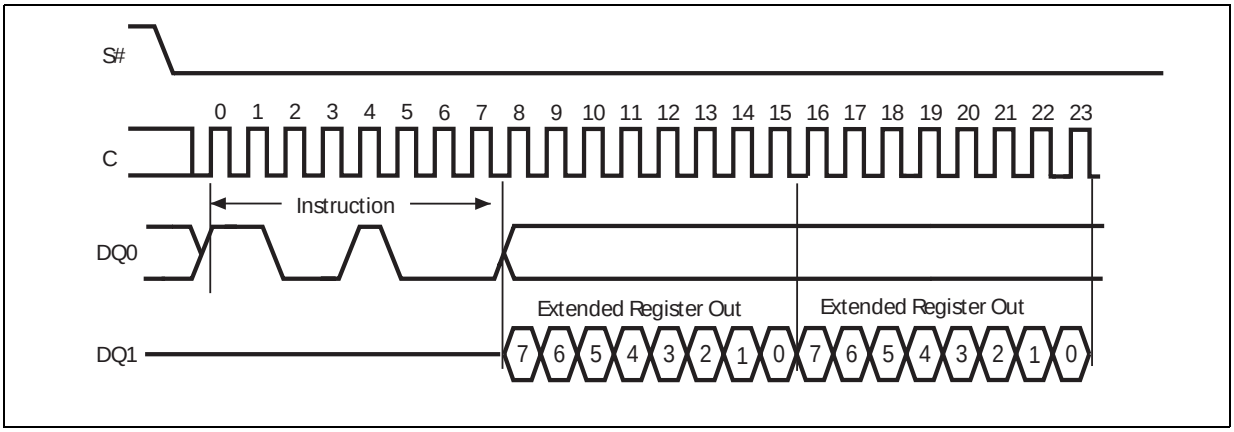
Figure 56. Write Volatile Enhanced Configuration Register instruction sequence



9.1.41 Read Extended Address Register

The Read Extended Address Register (RDEAR) instruction allows the Extended Address Register to be read.

Figure 57. Read Extended Address Register Instruction Sequence



9.1.42 Write Extended Address Register

The Write Extended Address register (WREAR) instruction allows new values to be written to the Extended Address register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

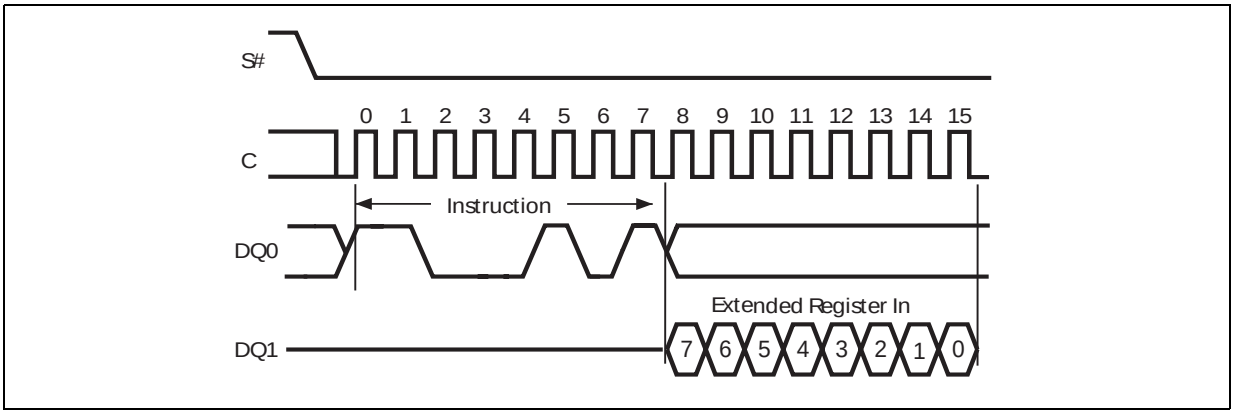
The Write Extended Address register (WREAR) instruction is entered by driving Chip Select (S) Low, followed by the instruction code and the data byte on serial data input (DQ0).

Chip Select (S) must be driven High after the eighth bit of the data byte has been latched in. If not, the The Write Extended Address register (WREAR) instruction is not executed. When the new data are latched, the write enable latch (WEL) is reset.

The Write Extended Address register (WREAR) instruction allows the user to change the values of all the Extended Address register bits, described in Table to be defined

The Write Extended Address register impacts the memory behavior right after the instruction is received by the device.

Figure 58. Write Extended Address Register Instruction Sequence

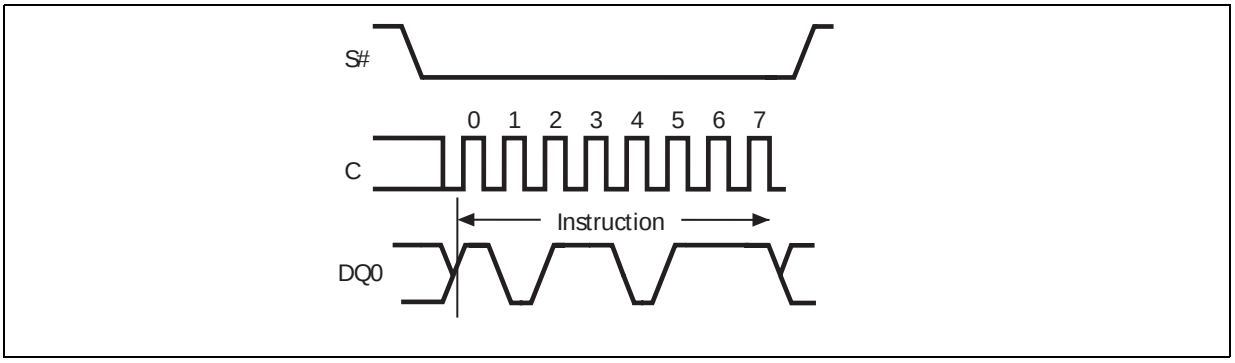


9.1.43 Enter 4-Byte Address Mode

The Enter 4-byte address mode (EN4BYTEADDR) instruction enables 4-byte address mode.

Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

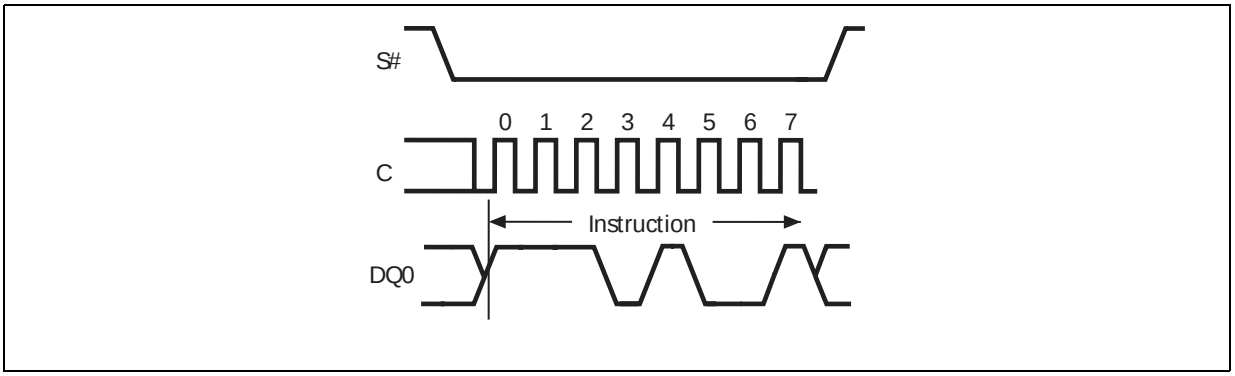
Figure 59. Enter 4-Byte Address Mode Instruction Sequence



9.1.44 Exit 4-Byte Address Mode

The Exit 4-byte address mode (EX4BYTEADDR) instruction disables 4-byte address mode. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

Figure 60. Exit 4-Byte Address Mode Instruction Sequence



9.1.45 Reset Enable

The Reset operation is used as a system software reset that puts the device in the power-on reset condition. All the lock bits, volatile configuration registers, and the extended address register are reset to the power-on reset default condition after the reset software sequence has been accepted. The power-on reset condition depends on non volatile configuration register content. This Reset operation consists of two instructions: Reset Enable and Reset Memory.

The Reset operation requires the Reset Enable instruction followed by the Reset Memory instruction. If the Reset Enable instruction is followed by any instruction other than Reset

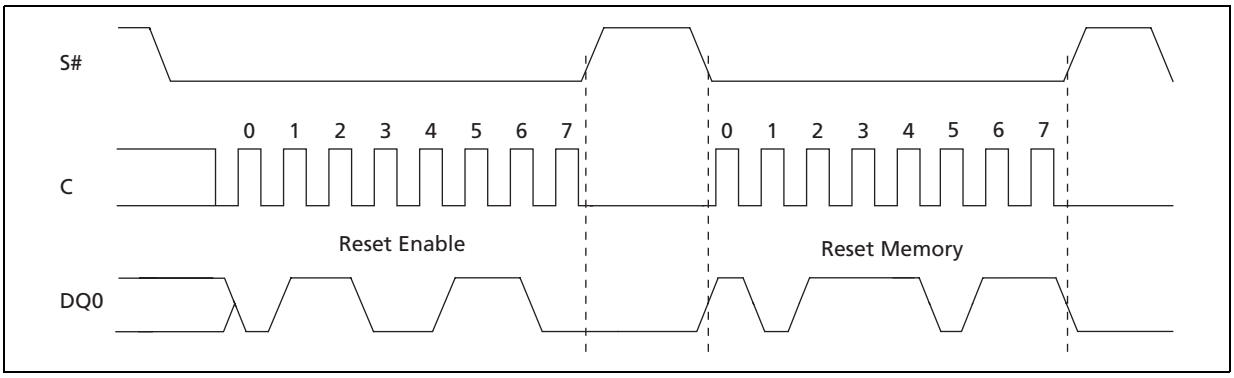
Memory, it is disabled. Reset Memory is also disabled if the device is selected by driving chip select (S) and Clock (C) low.

The Reset Enable instruction is entered by driving S low, followed by the instruction code on serial data input (DQ0). The Reset Memory instruction is entered by driving S low, followed by the instruction code on DQ0.

Minimum deselection time between the Reset Enable instruction and the Reset Memory instruction must be set according to the tSHSL2 specification; otherwise, reset software is not guaranteed.

If a Reset operation is begun while an internal Write, Program, or Erase operation is in progress or suspended, the internal operation is affected and data might be lost. As soon as S is driven high after the Reset Memory instruction is issued, the device enters the Reset mode and a time of tSHSL3 is then required before the device can be reselected by driving S low. You should exit XiP mode before entering software reset as described in *10.3: XIP mode hold and exit*.

Figure 61. Reset Enable and Reset Memory Instruction Sequence



9.2 DIO-SPI Instructions

In DIO-SPI protocol, instructions, addresses and input/Output data always run in parallel on two wires: DQ0 and DQ1.

In the case of a Dual Command Fast Read (DCFR), Dual Command Fast Read using 4-byte Address (DCFR4BYTE), Read OTP (ROTP), Read Lock Registers (RDLR), Read Status Register (RDSR), Read Flag Status Register (RFSR), Read NV Configuration Register (RDNVCR), Read Volatile Configuration Register (RDVCR), Read Volatile Enhanced Configuration Register (RDVECR), Read Serial Flash Discovery Parameter (RDSFDP), Read Extended Address Register (RDEAR) and Multiple I/O Read Identification (MIORDID) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (S) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a Dual Command Page Program (DCPP), Program OTP (POTP), Subsector Erase (SSE), Sector Erase (SE), Bulk Erase (BE), Program/Erase Suspend (PES), Program/Erase Resume (PER), Write Status Register (WRSR), Clear Flag Status Register (CLFSR), Write to Lock Register (WRLR), Write Configuration Register (WRVCR), Write Enhanced Configuration Register (WRVECR), Write NV Configuration Register (WRNVCR), Write Enable (WREN), Write Extended Address Register (WREAR), Enter 4-byte address mode (EN4BYTEADDR), Exit 4-byte address mode (EX4BYTEADDR) or Write Disable (WRDI) instruction, Chip Select (S) must be driven High exactly at a byte boundary, otherwise the instruction is rejected, and is not executed.

All attempts to access the memory array during a Write Status Register cycle, a Write Non Volatile Configuration Register, a Program cycle or an Erase cycle are ignored, and the internal Write Status Register cycle, Write Non Volatile Configuration Register, Program cycle or Erase cycle continues unaffected, the only exception is the Program/Erase Suspend instruction (PES), that can be used to pause all the program and the erase cycles but the Program OTP (POTP), Bulk Erase (BE) and Write Non Volatile Configuration Register. The suspended program or erase cycle can be resumed by mean of the Program/Erase Resume instruction (PER). During the program/erase cycles also the polling instructions (to check if the internal modify cycle is finished by mean of the WIP bit of the Status Register or of the Program/Erase controller bit of the Flag Status register) are also accepted to allow the application checking the end of the internal modify cycles, of course these polling instructions don't affect the internal cycles performing.

Table 24. Instruction set: DIO-SPI protocol (page 1 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
MIORDID	Multiple I/O read identification	1010 1111	AFh	0	0	1 to 3
RDSFDP	Read Serial Flash Discovery Parameter	01011010	5Ah	3	8	1 to ∞
DCFR	Dual Command Fast Read	0000 1011	0Bh	3/4(1)	8 (2)	1 to ∞
		0011 1011	3Bh	3/4(1)	8(2)	1 to ∞
		1011 1011	BBh	3/4(1)	8 (2)	1 to ∞
DCFR4BYTE	Dual Command Fast Read using 4 Byte Address	0000 1100	0Ch	4	8 (2)	1 to ∞
		0011 1100	3Ch	4	8 (2)	1 to ∞
		1011 1100	BCh	4	8 (2)	1 to ∞
ROTP	Read OTP	0100 1011	4Bh	3/4(1)	8 (2)	1 to 65
WREN	Write Enable	0000 0110	06h	0	0	0
WRDI	Write Disable	0000 0100	04h	0	0	0
DCPP	Dual Command Page Program	0000 0010	02h	3/4(1)	0	1 to 256
		1010 0010	A2h	3/4(1)	0	1 to 256
		1101 0010	D2h	3/4(1)	0	1 to 256
POTP	Program OTP	0100 0010	42h	3/4(1)	0	1 to 65
SSE	SubSector Erase	0010 0000	20h	3/4(1)	0	0
SE	Sector Erase	1101 1000	D8h	3/4(1)	0	0
BE	Bulk Erase	1100 0111	C7h	0	0	0
PER	Program/Erase Resume	0111 1010	7Ah	0	0	0
PES	Program/Erase Suspend	0111 0101	75h	0	0	0
RDSR	Read Status Register	0000 0101	05h	0	0	1 to ∞
WRSR	Write Status Register	0000 0001	01h	0	0	1
RDLR	Read Lock Register	1110 1000	E8h	3/4(1)	0	1 to ∞
WRLR	Write to Lock Register	1110 0101	E5h	3/4(1)	0	1
RFSR	Read Flag Status Register	0111 0000	70h	0	0	1 to ∞
CLFSR	Clear Flag Status Register	0101 0000	50h	0	0	0
RDNVCR	Read NV Configuration Register	1011 0101	B5h	0	0	2
WRNVCR	Write NV Configuration Register	1011 0001	B1h	0	0	2
RDVCR	Read Volatile Configuration Register	1000 0101	85h	0	0	1 to ∞
WRVCR	Write Volatile Configuration Register	1000 0001	81h	0	0	1

Table 24. Instruction set: DIO-SPI protocol (page 2 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
RDVECR	Read Volatile Enhanced Configuration Register	0110 0101	65h	0	0	1 to ∞
WRVECR	Write Volatile Enhanced Configuration Register	0110 0001	61h	0	0	1
EN4BYTEADDR	Enter 4-byte address mode	1011 0111	B7h	0	0	0
EX4BYTEADDR	Exit 4-byte address mode	1110 1001	E9h	0	0	0
WREAR	Write Extended Address Register	1100 0101	C5h	0	0	0
RDEAR	Read Extended Address Register	1100 1000	C8h	0	0	0
RSTEN	Reset Enable	0110 0110	66h	0	0	0
RST	Reset Memory	1001 1001	99h	0	0	0

1. Accordingly to address mode (default is 3).
2. The number of Dummy Clock cycles is configurable by the user.

9.2.1 Multiple I/O Read Identification protocol

The Multiple Input/Output Read Identification (MIORDID) instruction allows to read the device identification data in the DIO-SPI protocol:

- Manufacturer identification (1 byte)
- Device identification (2 bytes)

Unlike the RDID instruction of the Extended SPI protocol, the Multiple Input/Output instruction can not read the Unique ID code (UID) (17 bytes).

For further details on the manufacturer and device identification codes please refer to *Section 9.1.1: Read Identification (RDID)*.

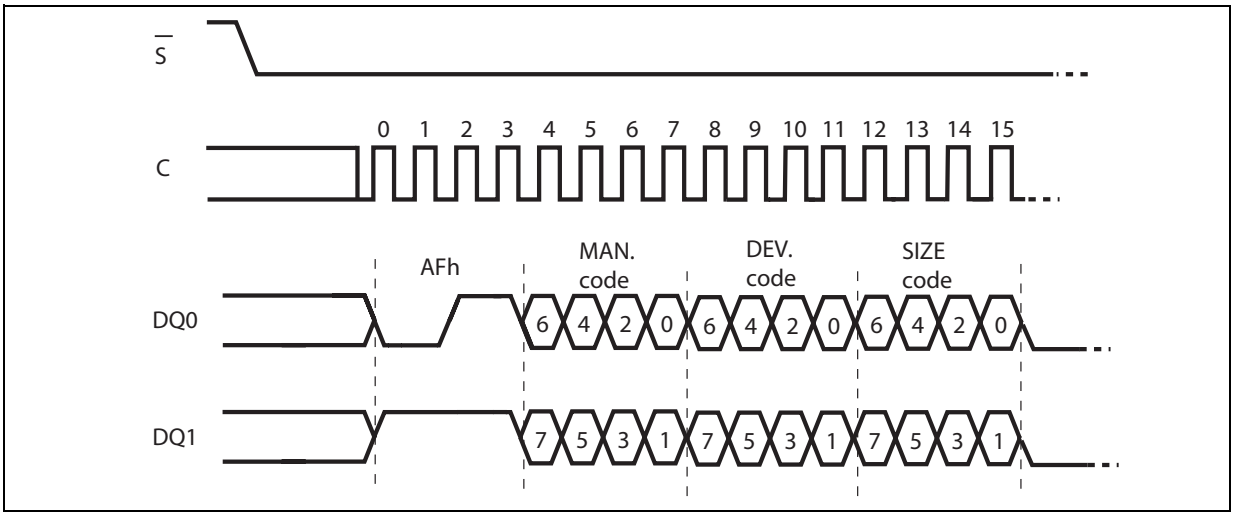
Any Multiple Input/Output Read Identification (MIORDID) instruction while an Erase or Program cycle is in progress, is not decoded, and has no effect on the cycle that is in progress.

The device is first selected by driving Chip Select ($\overline{S}$) Low. Then, the 8-bit instruction code for the instruction is shifted in parallel on the 2 pins DQ0 and DQ1. After this, the 24-bit device identification, stored in the memory, will be shifted out on again in parallel on DQ1 and DQ0. Each two bits are shifted out during the falling edge of Serial Clock (C).

The Read Identification (RDID) instruction is terminated by driving Chip Select ($\overline{S}$) High at any time during data output.

When Chip Select ($\overline{S}$) is driven High, the device is put in the Standby Power mode. Once in the Standby Power mode, the device waits to be selected, so that it can receive, decode and execute instructions.

Figure 62. Multiple I/O Read Identification instruction and data-out sequence DIO-SPI



9.2.2 Read Serial Flash Discovery Parameter

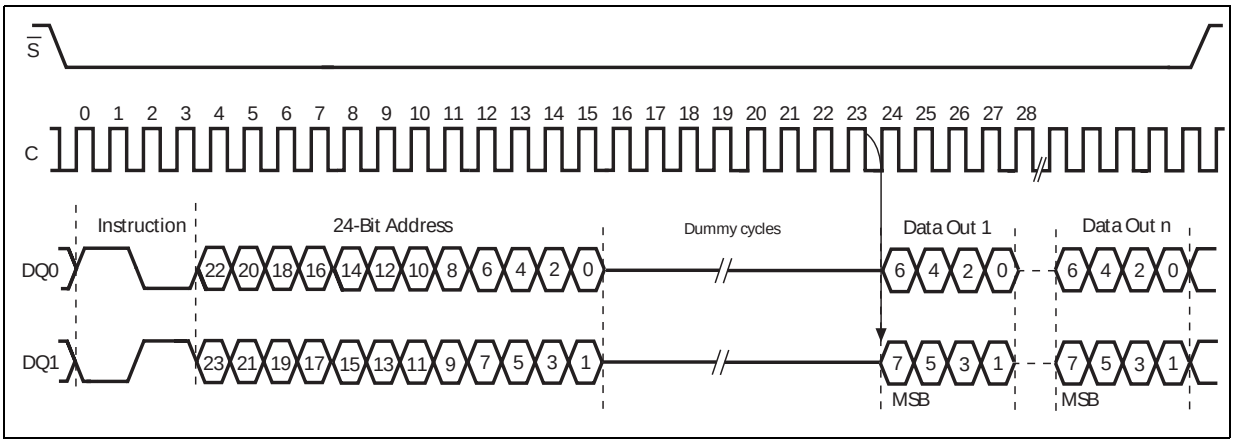
The Read Serial Flash Discovery Parameter (RDSFDP) instruction allows to read the Serial Flash Discovery Parameter area (SFDP) in the DIO-SPI protocol.

The instruction functionality is exactly the same as the Read Serial Flash Discovery Parameter instruction of the Extended SPI protocol; the only difference is that in the DIO-SPI protocol instruction code, address and output data are all parallelized on the two pins DQ0 and DQ1.

The serial flash discovery parameter area is always addressable by means 3 byte regardless active address mode.

Note: The dummy bits can not be parallelized since these clock cycles are requested to perform the internal reading operation.

Figure 63. Dual Read Serial Flash Discovery Parameter



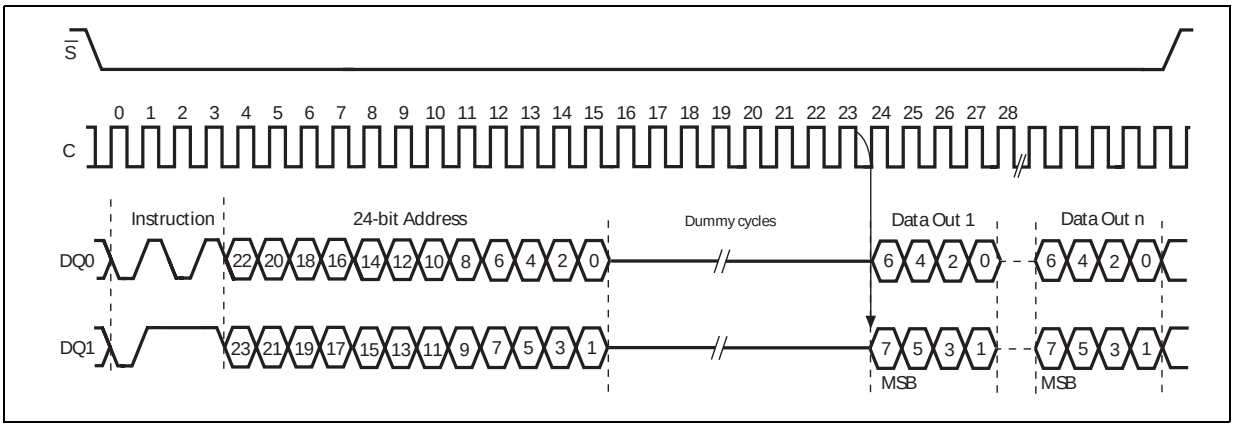
9.2.3 Dual Command Fast Read (DCFR)

The Dual Command Fast Read (DCFR) instruction allows to read the memory in DIO-SPI protocol, parallelizing the instruction code, the address and the output data on two pins (DQ0 and DQ1). The Dual Command Fast Read (DCFR) instruction can be issued, when the device is set in DIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 0Bh, 3Bh or BBh, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to DIO-SPI protocol.

Apart for the parallelizing on two pins of the instruction code, the Dual Command Fast Read instruction functionality is exactly the same as the Dual I/O Fast Read of the Extended SPI protocol, please refer to *Section 9.1.9: Dual I/O Fast Read* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 64. Dual Command Fast Read instruction and data-out sequence DIO-SPI

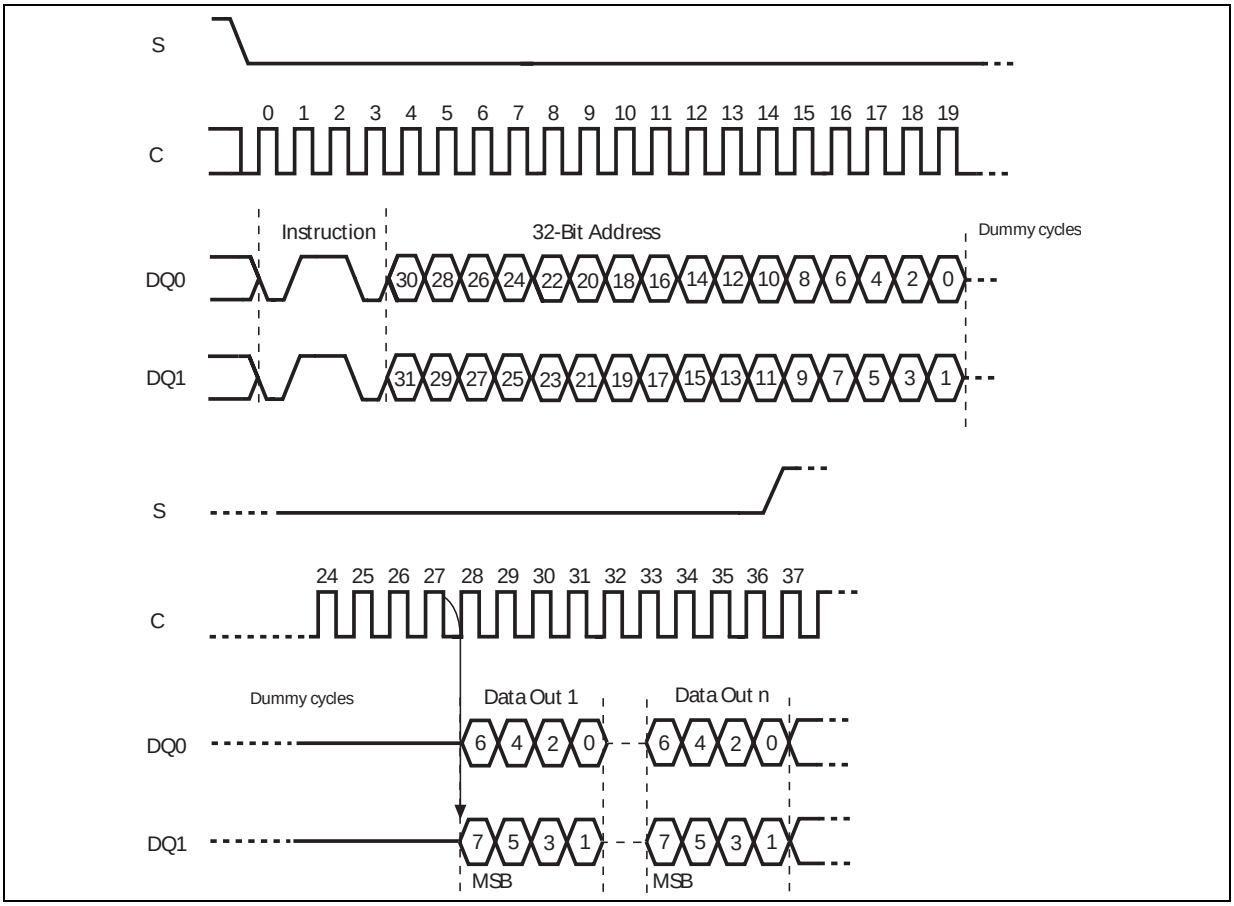


9.2.4 Dual Command Fast Read using 4 Byte Address (DCFR4Byte)

The Dual Command Fast Read using 4 bytes address (DCFR4BYTE) instruction allows to read the memory in DIO-SPI protocol, parallelizing the instruction code, the address and the output data on two pins (DQ0 and DQ1). The instruction can be issued, when the device is set in DIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 0Ch, 3Ch or BCh, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to DIO-SPI protocol.

Apart for the parallelizing on two pins of the instruction code, the Dual Command Fast Read using 4 bytes address instruction functionality is exactly the same as the Dual I/O Fast Read using 4 byte address of the Extended SPI protocol.

Figure 65. Dual Command Fast Read using 4 Byte Address Instruction and Data-Out Sequence



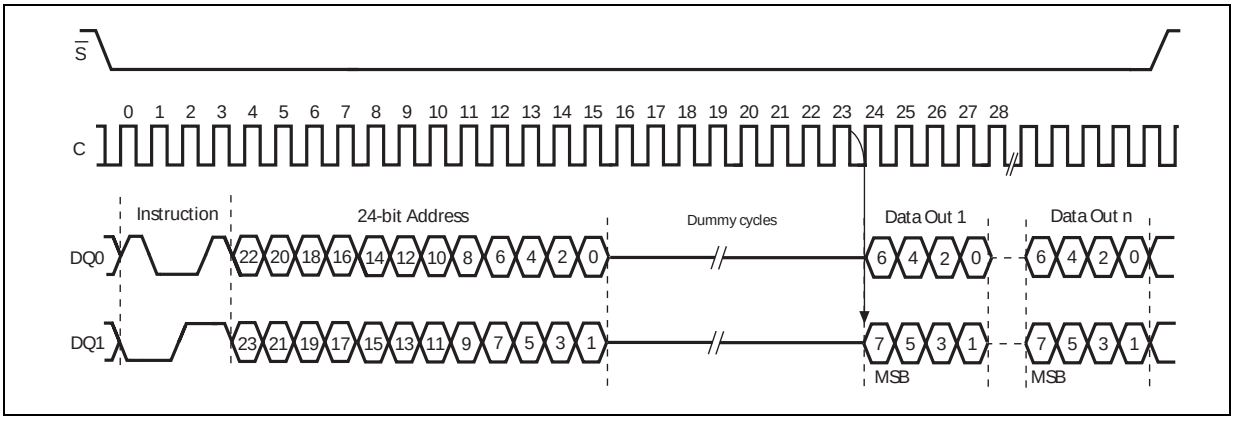
9.2.5 Read OTP (ROTP)

The Read OTP (ROTP) instruction is used to read the 64 bytes OTP area in the DIO-SPI protocol. The instruction functionality is exactly the same as the Read OTP instruction of the Extended SPI protocol; the only difference is that in the DIO-SPI protocol instruction code, address and output data are all parallelized on the two pins DQ0 and DQ1.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Note: The dummy bits can not be parallelized since these clock cycles are requested to perform the internal reading operation.

Figure 66. Read OTP instruction and data-out sequence DIO-SPI

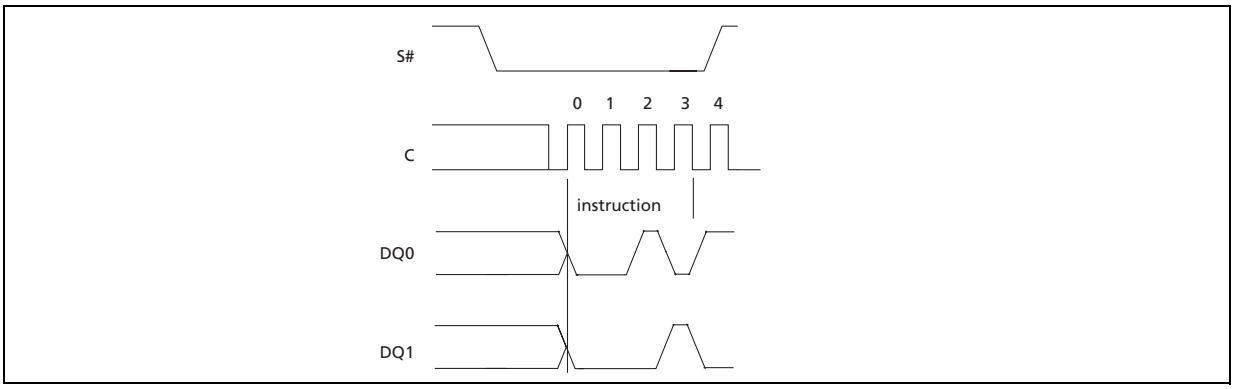


9.2.6 Write Enable (WREN)

The Write Enable (WREN) instruction sets the Write Enable Latch (WEL) bit.

Apart form the parallelizing of the instruction code on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write Enable (WREN) instruction of the Extended SPI protocol.

Figure 67. Write Enable instruction sequence DIO-SPI

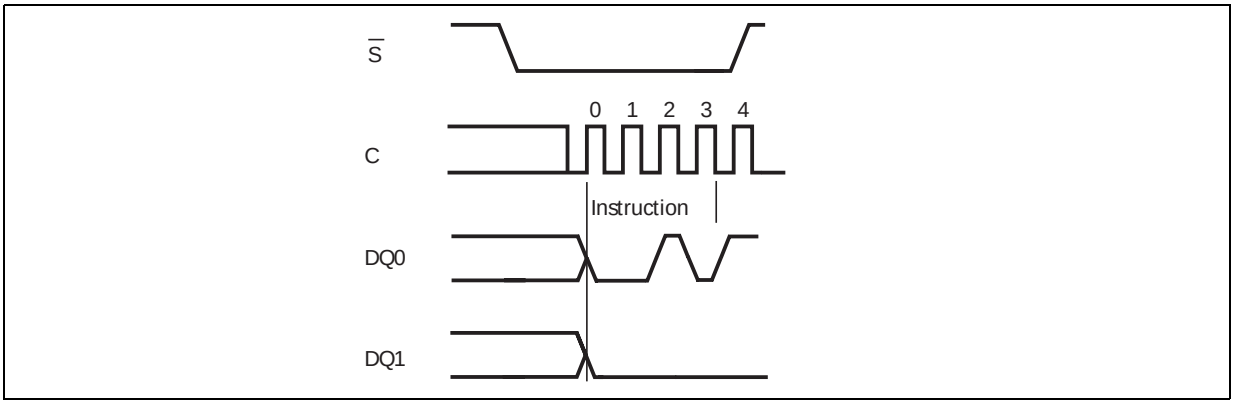


9.2.7 Write Disable (WRDI)

The Write Disable (WRDI) instruction resets the Write Enable Latch (WEL) bit.

Apart form the parallelizing of the instruction code on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write Disable (WRDI) instruction of the Extended SPI protocol, please refer to *Section 9.1.17: Write Disable (WRDI)* for further details.

Figure 68. Write Disable instruction sequence DIO-SPI



9.2.8 Dual Command Page Program (DCPP)

The Dual Command Page Program (DCPP) instruction allows to program the memory content in DIO-SPI protocol, parallelizing the instruction code, the address and the input data on two pins (DQ0 and DQ1). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. The Dual Command Page Program (DCPP) instruction can be issued, when the device is set in DIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 02h, A2h or D2h, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to DIO-SPI protocol.

Apart for the parallelizing on two pins of the instruction code, the Dual Command Page Program instruction functionality is exactly the same as the Dual Input Extended Fast Program of the Extended SPI protocol, please refer to *Section 9.1.20: Dual Input Extended Fast Program* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 69. Dual Command Page Program instruction sequence DSP, 02h

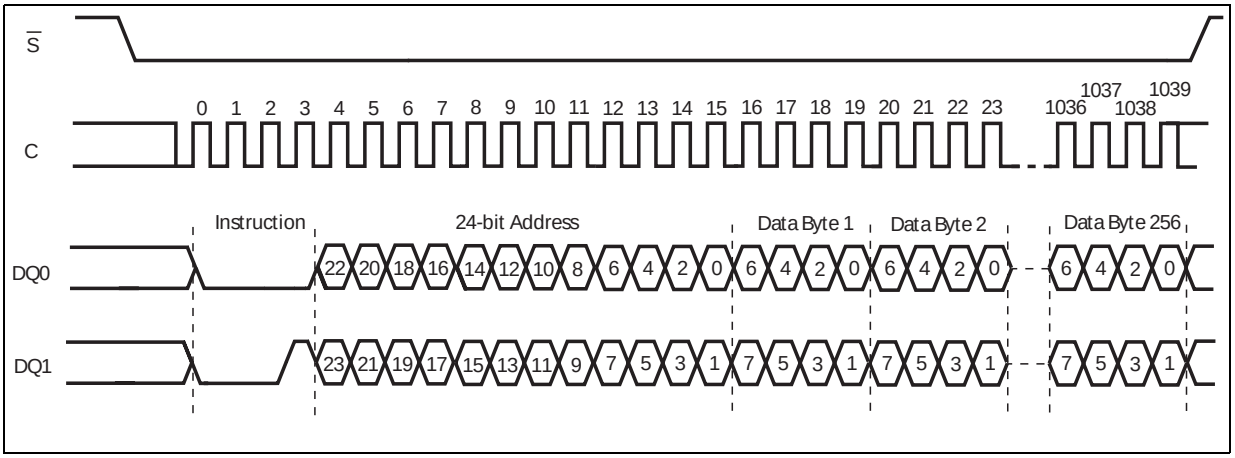


Figure 70. Dual Command Page Program instruction sequence DSP, A2h

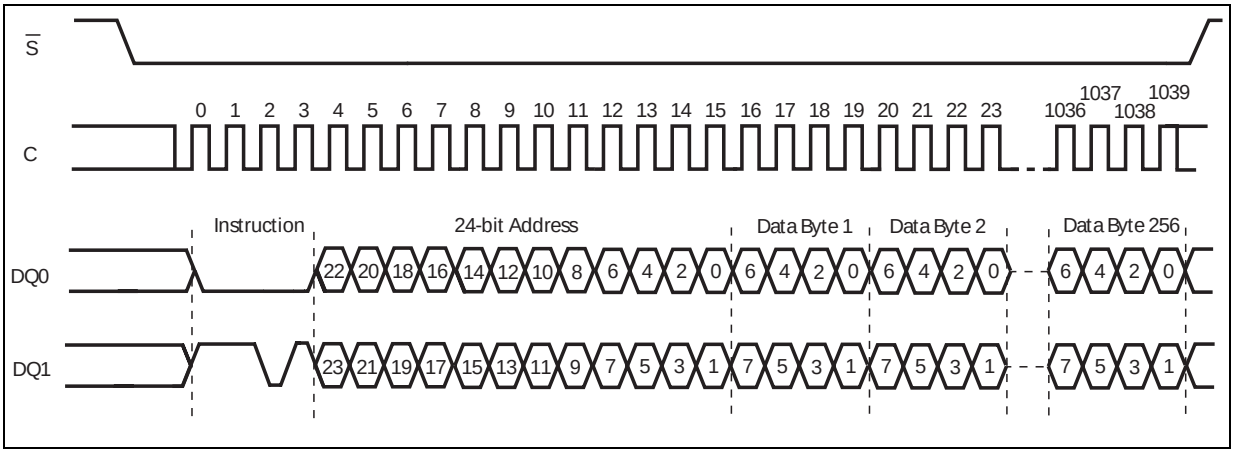
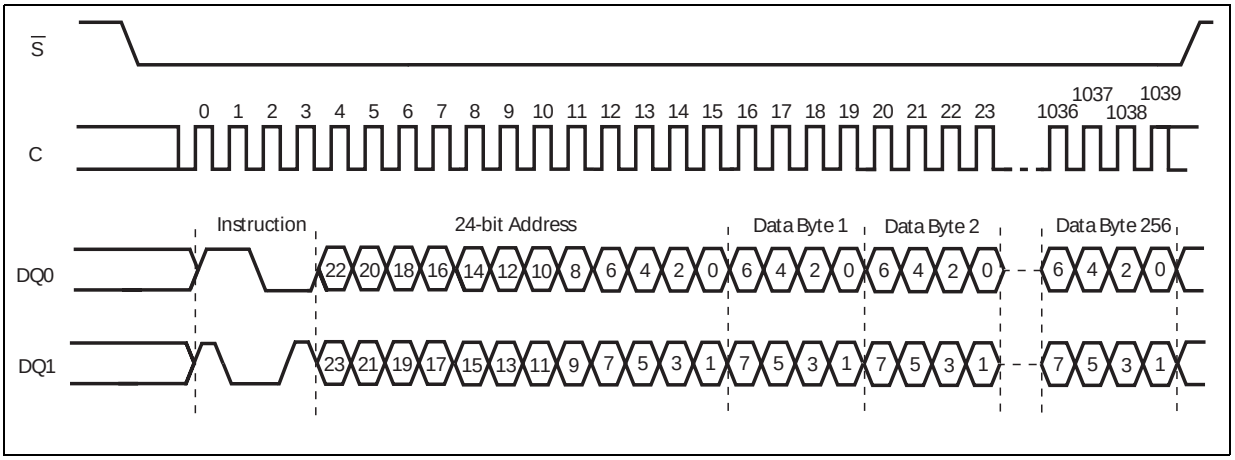


Figure 71. Dual Command Page Program instruction sequence DSP, D2h



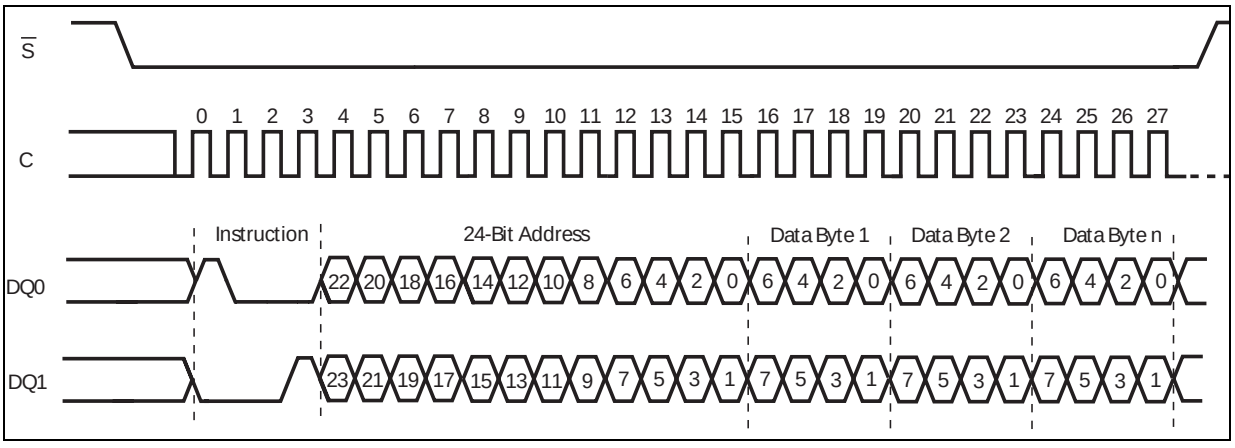
9.2.9 Program OTP instruction (POTP)

The Program OTP instruction (POTP) is used to program at most 64 bytes to the OTP memory area (by changing bits from 1 to 0, only). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

Apart form parallelizing the instruction code, address, and input data on pins DQ0 and DQ1, the instruction functionality and the locking OTP method are the same as the Program OTP (POTP) instruction of the Extended SPI protocol; refer to *Section 9.1.23: Program OTP instruction (POTP)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 72. Program OTP instruction sequence DIO-SPI



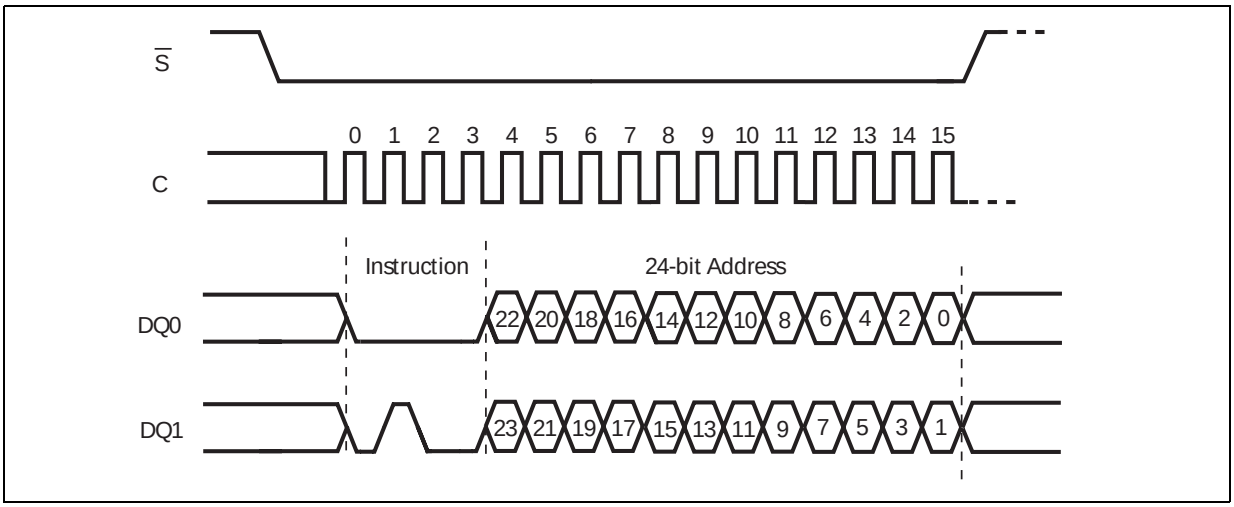
9.2.10 Subsector Erase (SSE)

The Subsector Erase (SSE) instruction sets to '1' (FFh) all bits inside the chosen subsector. Before it can be accepted, a Write Enable (WREN) instruction must have been executed previously.

Apart form parallelizing the instruction code and the address on pins DQ0 and DQ1, the instruction functionality is the same as the Subsector Erase (SSE) instruction of the Extended SPI protocol; Refer to *Section 9.1.24: Subsector Erase (SSE)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 73. Subsector Erase instruction sequence DIO-SPI



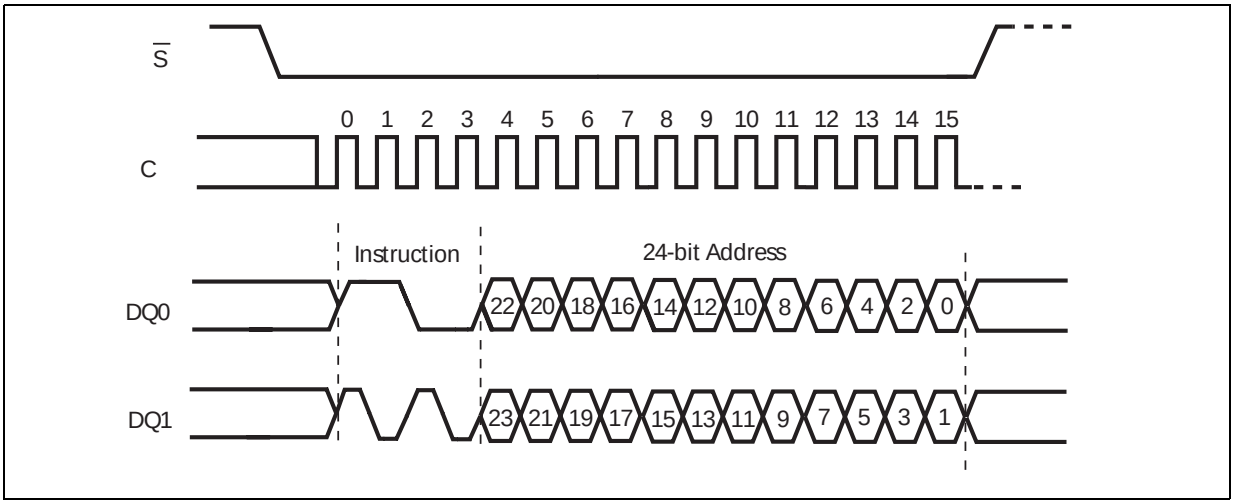
9.2.11 Sector Erase (SE)

The Sector Erase (SE) instruction sets to '1' (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

Apart form parallelizing the instruction code and the address on the two pins DQ0 and DQ1, the instruction functionality is the same as the Sector Erase (SE) instruction of the Extended SPI protocol, please refer to *Section 9.1.25: Sector Erase (SE)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 74. Sector Erase instruction sequence DIO-SPI

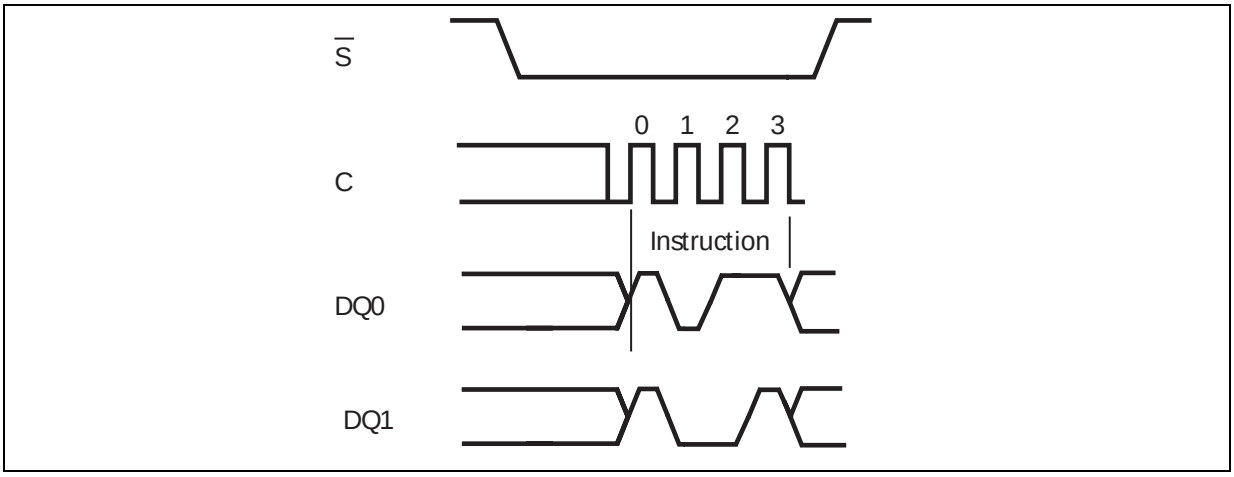


9.2.12 Bulk Erase (BE)

The Bulk Erase (BE) instruction sets all bits to '1' (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Bulk Erase (BE) instruction of the Extended SPI protocol, please refer to *Section 9.1.26: Bulk Erase (BE)* for further details.

Figure 75. Bulk Erase instruction sequence DIO-SPI

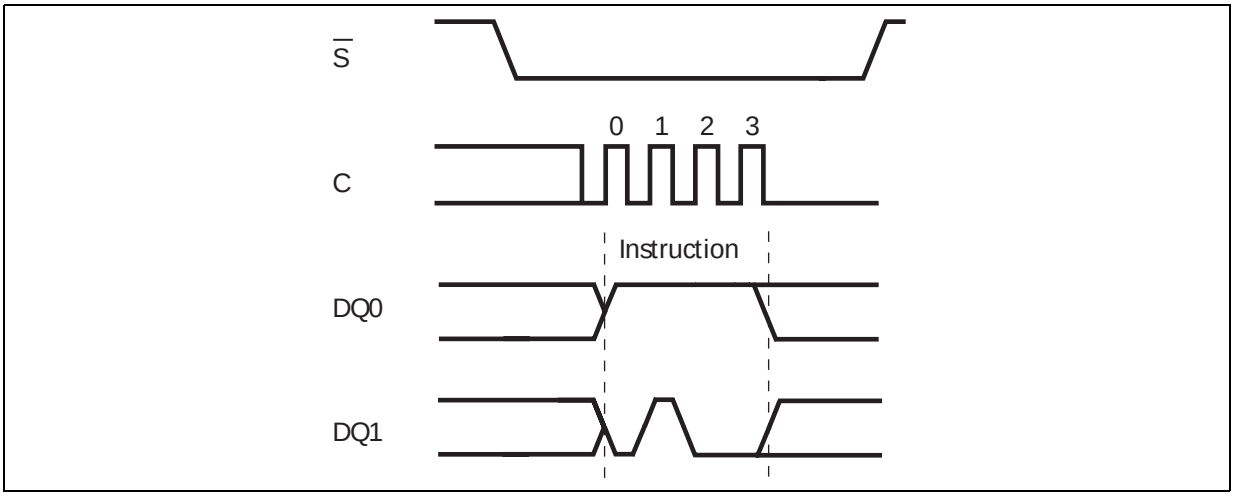


9.2.13 Program/Erase Suspend

The Program/Erase Suspend instruction allows the controller to interrupt a Program or an Erase instruction, in particular: Sector Erase and Dual Command Page Program can be suspended and erased while Subsector Erase, Bulk Erase, Write Non Volatile Configuration register, and Program OTP cannot be suspended.

Apart form the parallelizing of the instruction code on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Program/Erase Suspend (PES) instruction of the Extended SPI protocol.

Figure 76. Program/Erase Suspend instruction sequence DIO-SPI

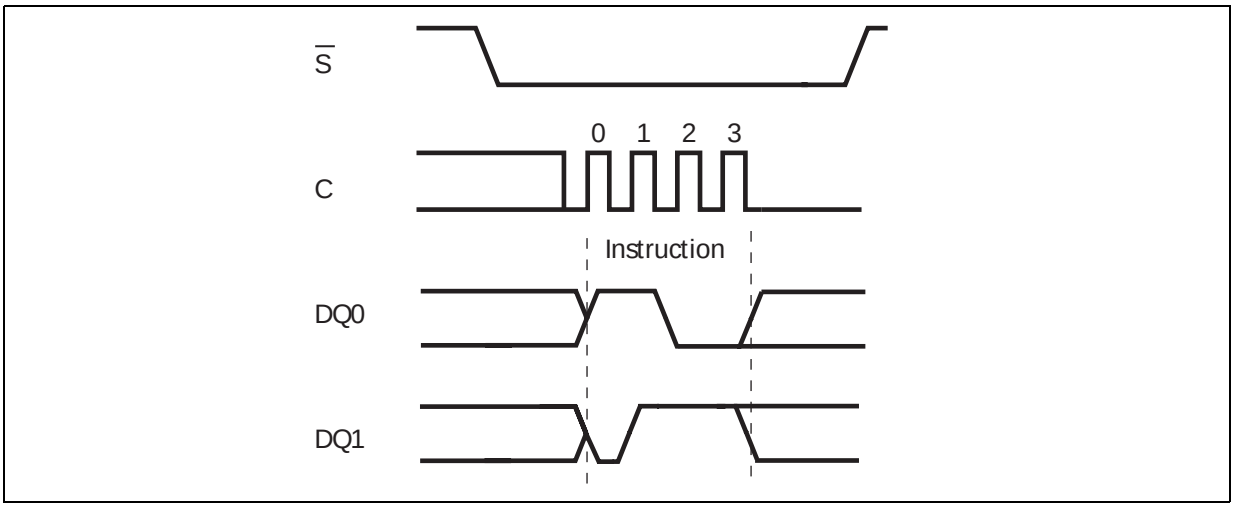


9.2.14 Program/Erase Resume

After a Program/Erase suspend instruction, a Program/Erase Resume instruction is required to continue performing the suspended Program or Erase sequence.

Apart form the parallelizing of the instruction code on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Program/Erase Resume (PER) instruction of the Extended SPI protocol, please refer to *Section 9.1.28: Program/Erase Resume* for further details.

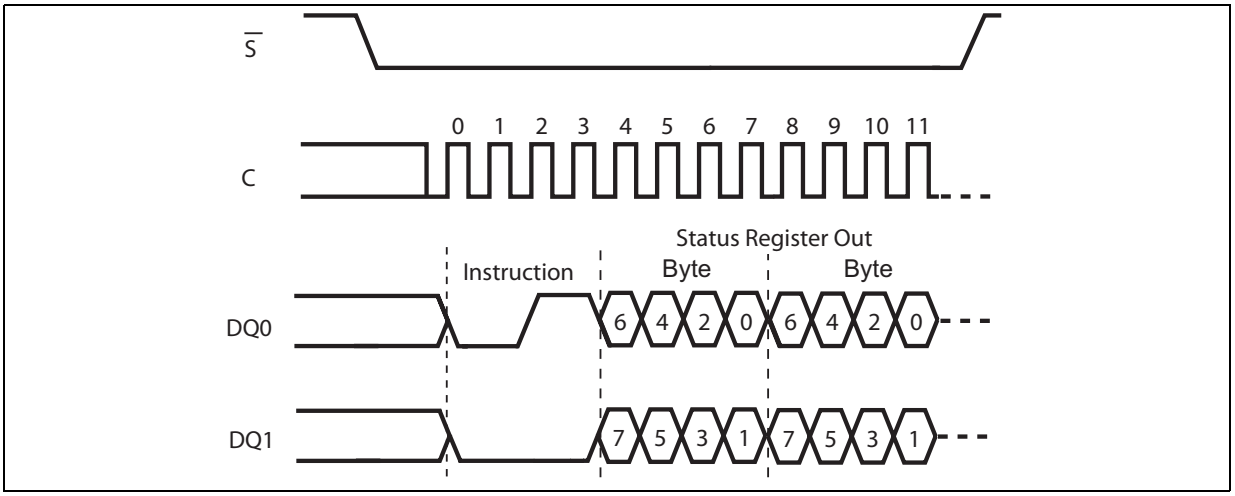
Figure 77. Program/Erase Resume instruction sequence DIO-SPI



9.2.15 Read Status Register (RDSR)

The Read Status Register (RDSR) instruction allows the Status Register to be read. Apart form the parallelizing of the instruction code and the output data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Read Status Register (RDSR) instruction of the Extended SPI protocol, please refer to *Section 9.1.29: Read Status Register (RDSR)* for further details.

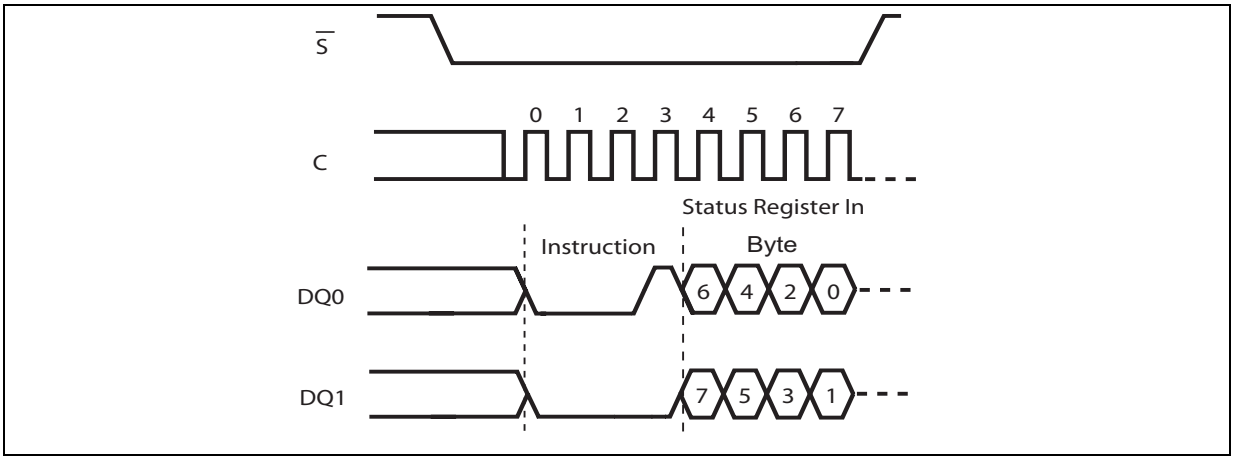
Figure 78. Read Status Register instruction sequence DIO-SPI



9.2.16 Write status register (WRSR)

The write status register (WRSR) instruction allows new values to be written to the status register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. Apart form the parallelizing of the instruction code and the input data on the two pins DQ0 and DQ1, the instruction functionality and the protection feature management is exactly the same as the Write Status Register (WRSR) instruction of the Extended SPI protocol, please refer to *Section 9.1.30: Write status register (WRSR)* for further details.

Figure 79. Write Status Register instruction sequence DIO-SPI



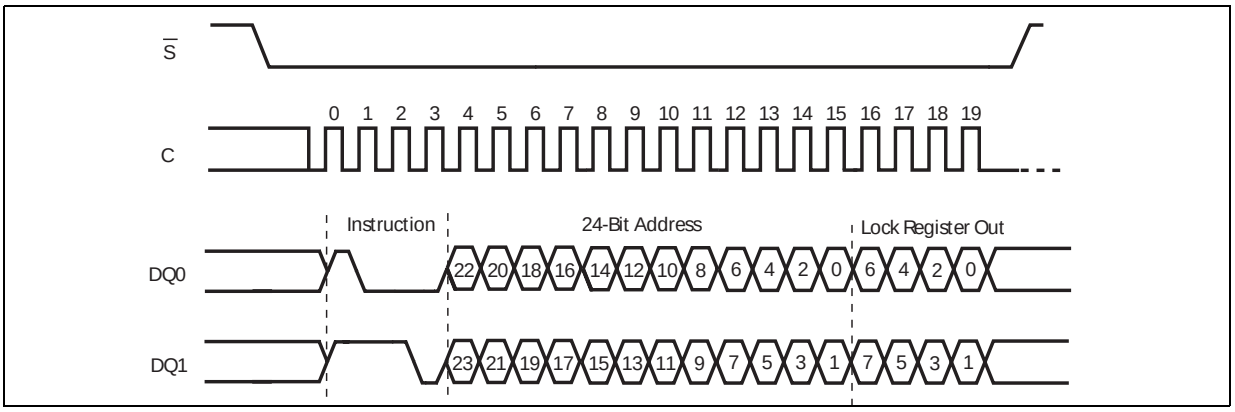
9.2.17 Read Lock Register (RDLR)

The Read Lock Register instructions is used to read the lock register content.

Apart form the parallelizing of the instruction code, the address and the output data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Read Lock Register (RDLR) instruction of the Extended SPI protocol, please refer to *Section 9.1.31: Read Lock Register (RDLR)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 80. Read Lock Register instruction and data-out sequence DIO-SPI



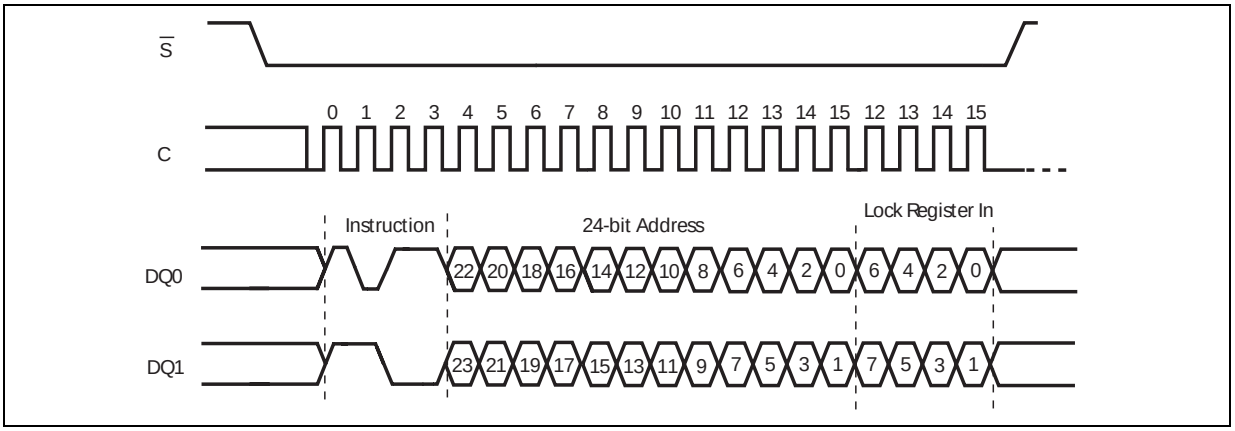
9.2.18 Write to Lock Register (WRLR)

The Write to Lock Register (WRLR) instruction allows bits to be changed in the Lock Registers. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code, the address and the input data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write to Lock Register (WRLR) instruction of the Extended SPI protocol, please refer to *Section 9.1.32: Write to Lock Register (WRLR)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 81. Write to Lock Register instruction sequence DIO-SPI

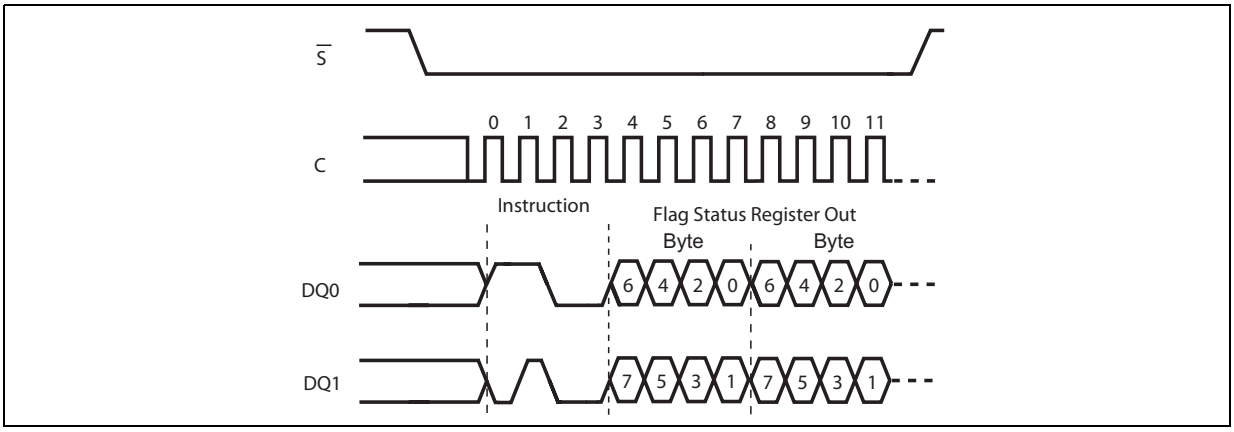


9.2.19 Read Flag Status Register

The Read Flag Status Register (RFSR) instruction allows the Flag Status Register to be read.

Apart form the parallelizing of the instruction code and the output data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Read Flag Status Register (RFSR) instruction of the Extended SPI protocol, please refer to *Section 9.1.33: Read Flag Status Register* for further details.

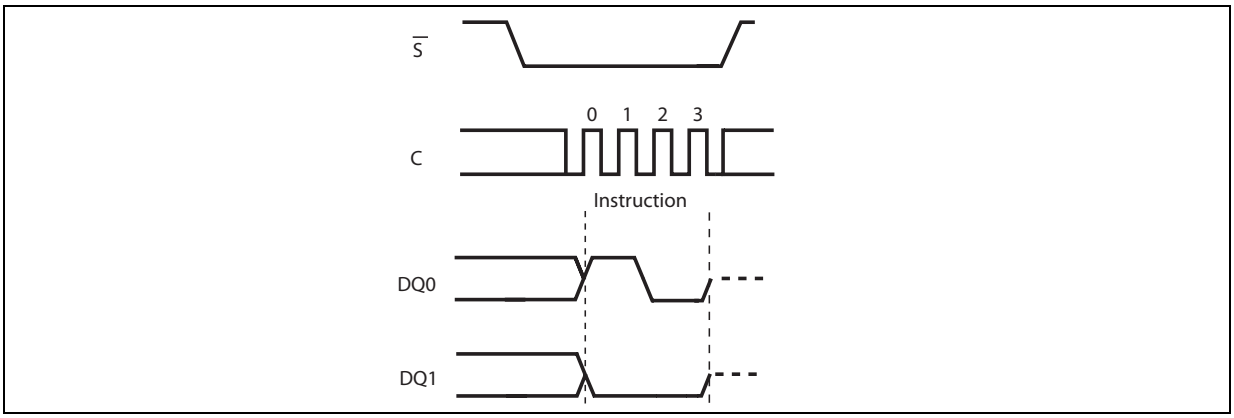
Figure 82. Read Flag Status Register instruction sequence DIO-SPI



9.2.20 Clear Flag Status Register

The Clear Flag Status Register (CLFSR) instruction reset the error Flag Status Register bits (Erase Error bit, Program Error bit, VPP Error bit, Protection Error bit). It is not necessary to set the WEL bit before the Clear Flag Status Register instruction is executed. The WEL bit will be unchanged after this command is executed.

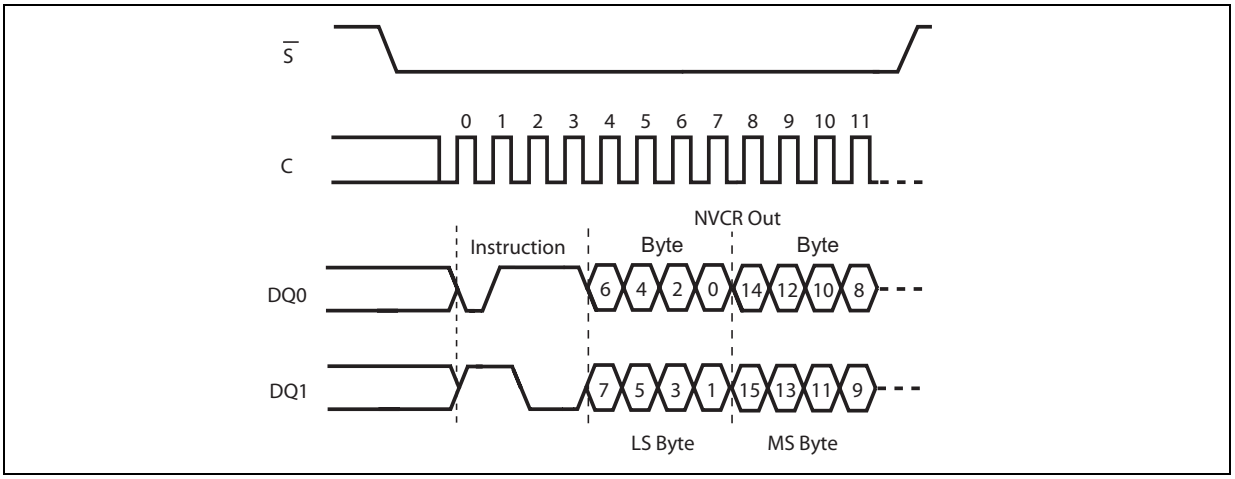
Figure 83. Clear Flag Status Register instruction sequence DIO-SPI



9.2.21 Read NV Configuration Register, Dual I/O

The Read Non Volatile Configuration Register (RDNVCR) instruction allows the Non Volatile Configuration Register to be read.

Figure 84. Read NV Configuration Register instruction sequence DIO-SPI

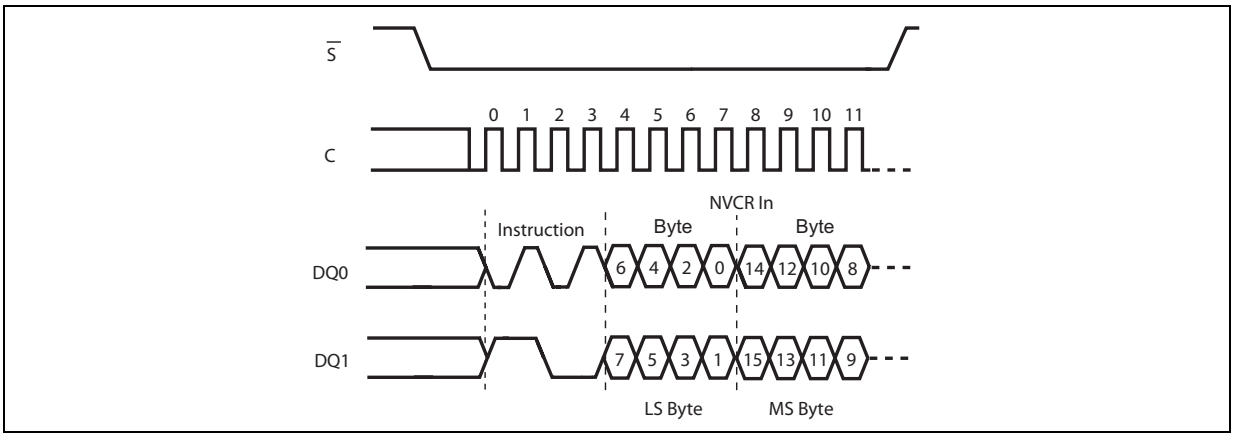


9.2.22 Write NV Configuration Register, Dual I/O

The Write Non Volatile Configuration register (WRNVCR) instruction allows new values to be written to the Non Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code and the input data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write Non Volatile Configuration Register (WNVCR) instruction of the Extended SPI protocol, please refer to *Section 9.1.36: Write NV Configuration Register* for further details.

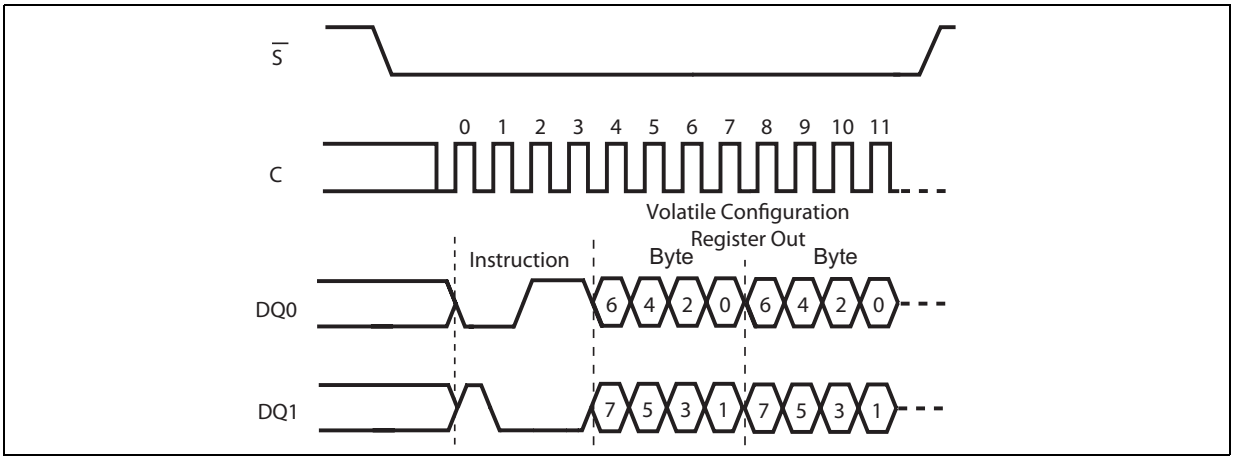
Figure 85. Write NV Configuration Register instruction sequence DIO-SPI



9.2.23 Read Volatile Configuration Register, Dual I/O

The Read Volatile Configuration Register (RDVCR) instruction allows the Volatile Configuration Register to be read. See *Table 6.: Volatile Configuration Register*.

Figure 86. Read Volatile Configuration Register instruction sequence DIO-SPI

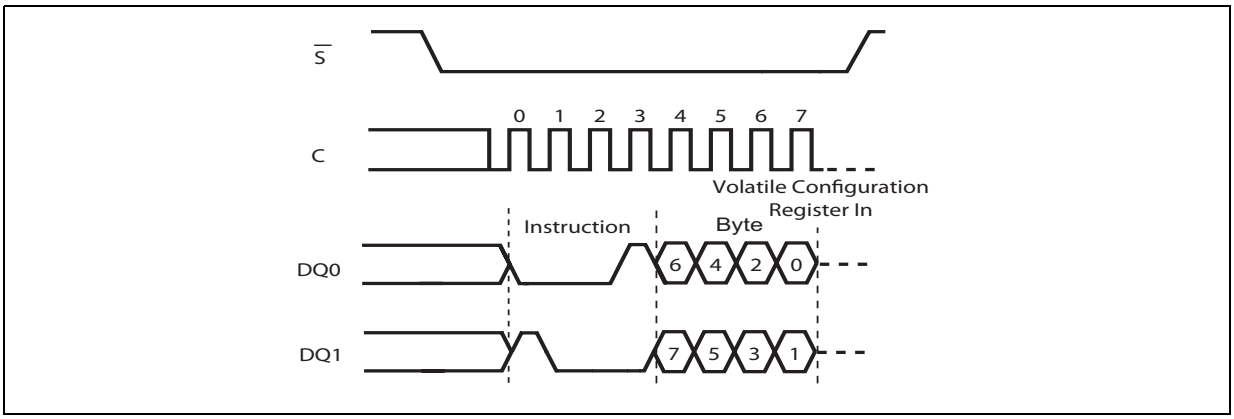


9.2.24 Write Volatile Configuration Register, Dual I/O

The Write Volatile Configuration register (WRVCR) instruction allows new values to be written to the Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must have been executed previously.

Apart form the parallelizing of the instruction code and the input data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write Volatile Configuration Register (WVCR) instruction of the Extended SPI protocol, please refer to *Section 9.1.38: Write Volatile Configuration Register* for further details.

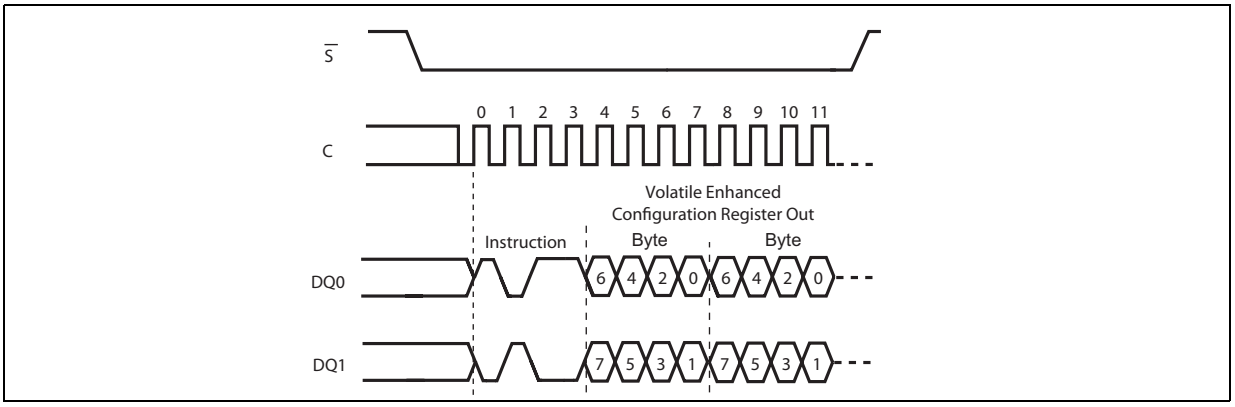
Figure 87. Write Volatile Configuration Register instruction sequence DIO-SPI



9.2.25 Read Volatile Enhanced Configuration Register, Dual I/O

The Read Volatile Enhanced Configuration Register (RDVECR) instruction allows the Volatile Configuration Register to be read.

Figure 88. Read Volatile Enhanced Configuration Register instruction sequence DIO-SPI

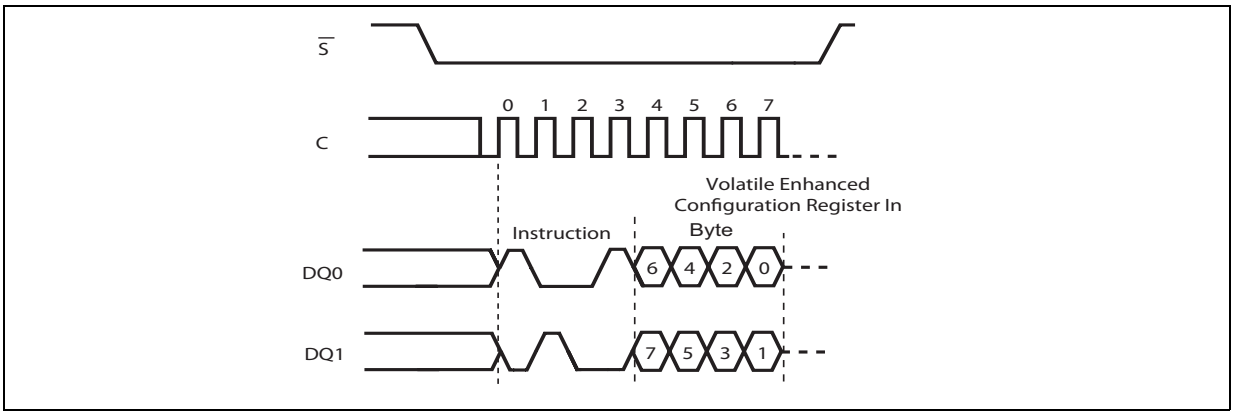


9.2.26 Write Volatile Enhanced Configuration Register, Dual I/O

The Write Volatile Enhanced Configuration Register instruction allows new values to be written to the Volatile Enhanced Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code and the input data on the two pins DQ0 and DQ1, the instruction functionality is exactly the same as the Write Volatile Enhanced Configuration Register (WRVECR) instruction of the Extended SPI protocol, please refer to *Section 9.1.40: Write Volatile Enhanced Configuration Register* for further details.

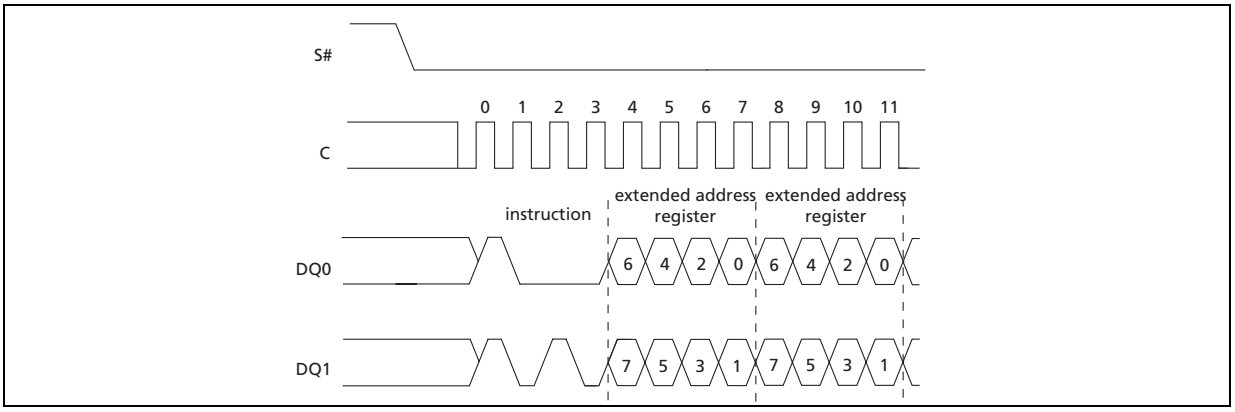
Figure 89. Write Volatile Enhanced Configuration Register Instruction Sequence, Dual I/O



9.2.27 Read Extended Address Register, Dual I/O

The Read Extended Address Register instruction allows the volatile configuration register to be read.

Figure 90. Read Extended Address Register Instruction Sequence, Dual I/O



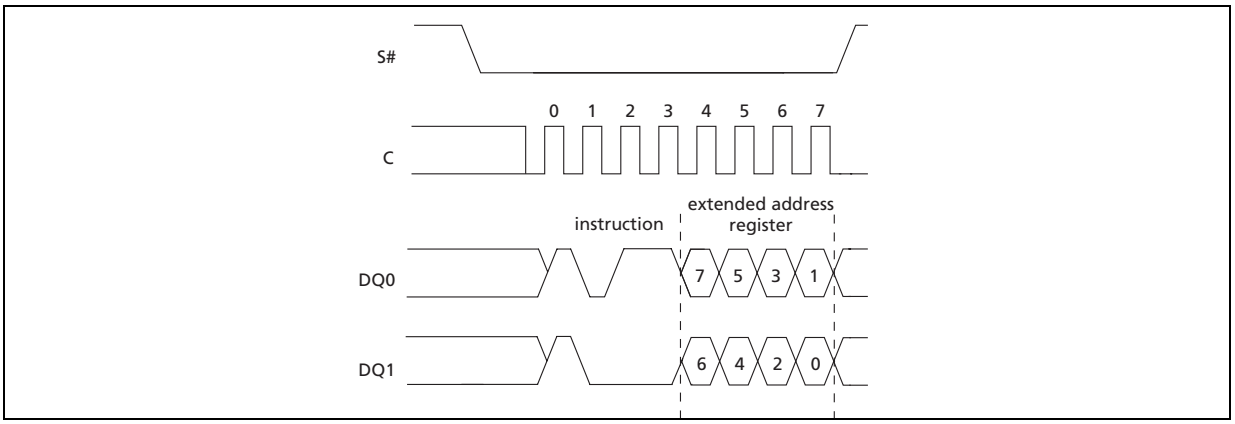
9.2.28 Write Extended Address Register, Dual I/O

The Write Extended Address Register, dual I/O instruction allows new values to be written to the extended address register. Before this instruction can be accepted, a Write Enable instruction must have been executed previously.

Apart form the parallelizing of the instruction code and the input data on the two pins DQ0 and DQ1,

This instruction functions exactly as the Write Extended Address Register instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on two pins, DQ0 and DQ1.

Figure 91. Write Extended Address Register Instruction Sequence, Dual I/O

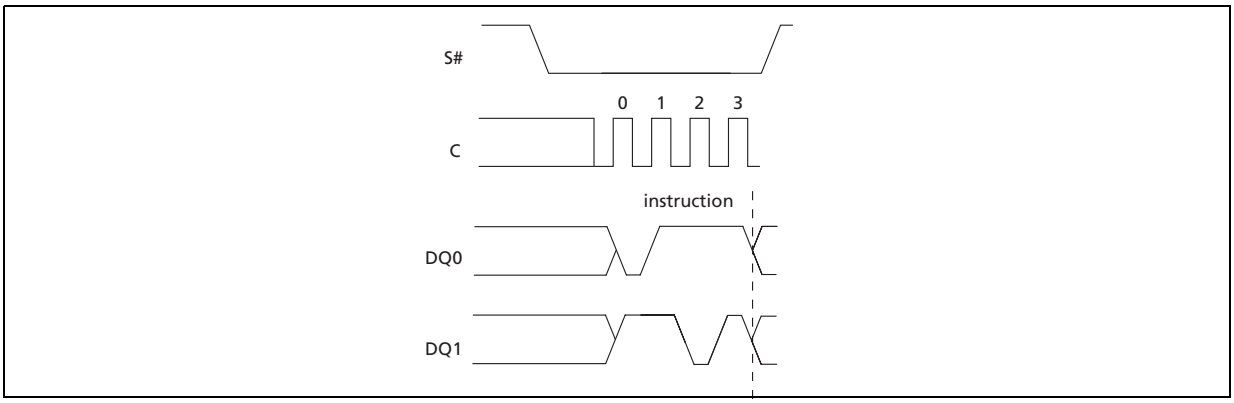


9.2.29 Enter 4-Byte Address Mode, Dual I/O

The Enter 4-Byte Address Mode instruction enables 4-byte address mode. Before this command can be accepted, a Write Enable instruction must have been executed previously. After the Write Enable instruction has been decoded and executed, the device sets the write enable latch (WEL) bit.

This instruction functions exactly as the Enter 4-Byte Address Mode instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on two pins, DQ0 and DQ1.

Figure 92. Enter 4-Byte Address Mode Instruction Sequence, Dual I/O

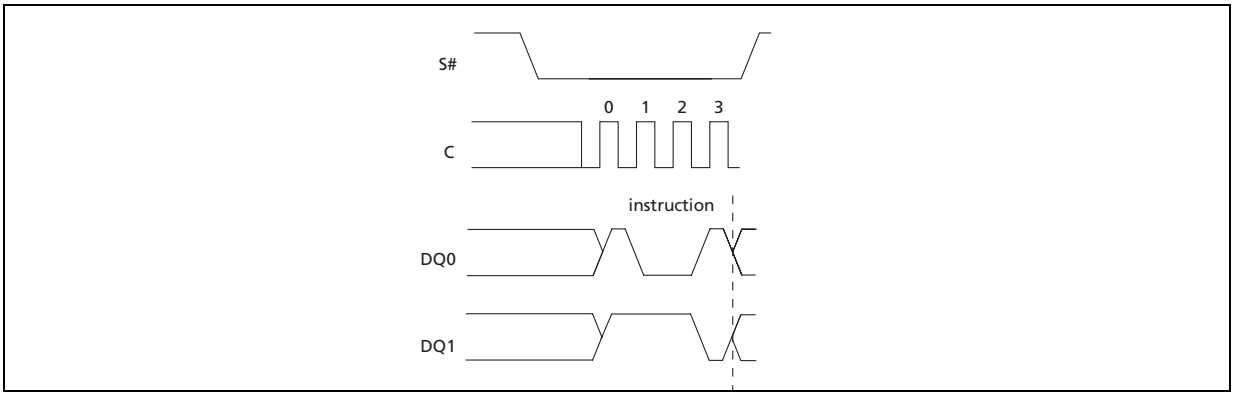


9.2.30 Exit 4-Byte Address Mode, Dual I/O

The Exit 4-Byte Address Mode instruction disables 4-byte address mode. Before this instruction can be accepted, a Write Enable instruction must have been executed previously. After the Write Enable instruction has been decoded and executed, the device sets the write enable latch (WEL) bit.

This instruction functions exactly as the Exit 4-Byte Address Mode instruction of the extended SPI protocol, except that for this instruction the instruction code and input data are on two pins, DQ0 and DQ1.

Figure 93. Exit 4-Byte Address Mode Instruction Sequence, Dual I/O



9.2.31 Reset Enable and Reset Memory, Dual I/O

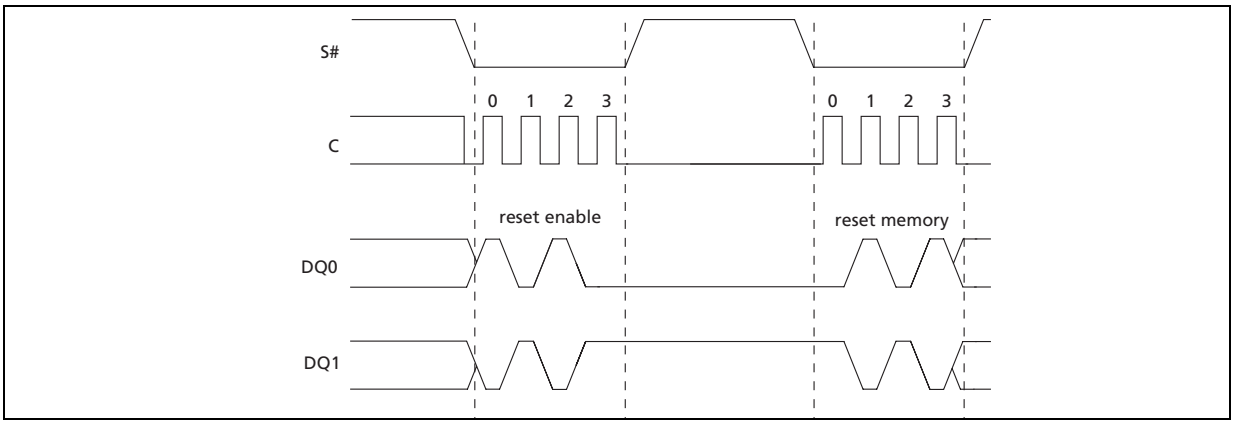
The Reset Enable and Reset Memory operation is used as a system software reset that puts the device in the power-on reset condition.

This operation consists of two instructions: Reset Enable and Reset Memory.

The Reset operation requires the Reset Enable instruction followed by the Reset Memory instruction. If the Reset Enable instruction is followed by any instruction other than Reset Memory, it is disabled. Reset Memory is also disabled if the device is selected by driving chip select (S) and Clock (C) low.

This instruction functions exactly as the Reset Enable instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on two pins, DQ0 and DQ1.

Figure 94. Reset Enable and Reset Memory Instruction Sequence, Dual I/O



9.3 QIO-SPI Instructions

In QIO-SPI protocol, instructions, addresses and Input/Output data always run in parallel on four wires: DQ0, DQ1, DQ2 and DQ3 with the already mentioned exception of the modify instruction (erase and program) performed with the VPP=VPPh.

In the case of a Quad Command Fast Read (QCFR), Quad Command Fast Read using 4-byte Address (QCFR4BYTE), Read OTP (ROTP), Read Lock Registers (RDLR), Read Status Register (RDSR), Read Flag Status Register (RFSR), Read NV Configuration Register (RDNVCR), Read Volatile Configuration Register (RDVCR), Read Volatile Enhanced Configuration Register (RDVECR) , Read Serial Flash Discovery Parameter (RDSFDP), Read Extended Address Register (RDEAR) and Multiple I/O Read Identification (MIORDID) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (S) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a Quad Command Page Program (QCPP), Program OTP (POTP), Subsector Erase (SSE), Sector Erase (SE), Bulk Erase (BE), Program/Erase Suspend (PES), Program/Erase Resume (PER), Write Status Register (WRSR), Clear Flag Status Register (CLFSR), Write to Lock Register (WRLR), Write Configuration Register (WRVCR), Write Enhanced Configuration Register (WRVECR), Write NV Configuration Register (WRNVCR), Write Enable (WREN) , Write Extended Address Register (WREAR), Enter 4-byte address mode (EN4BYTEADDR), Exit 4-byte address mode (EX4BYTEADDR) or Write Disable (WRDI) instruction, Chip Select (S) must be driven High exactly at a byte boundary, otherwise the instruction is rejected, and is not executed.

All attempts to access the memory array during a Write Status Register cycle, a Write Non Volatile Configuration Register, a Program cycle or an Erase cycle are ignored, and the internal Write Status Register cycle, Write Non Volatile Configuration Register, Program cycle or Erase cycle continues unaffected, the only exception is the Program/Erase Suspend instruction (PES), that can be used to pause all the program and the erase cycles but the Program OTP (POTP), Bulk Erase (BE) and Write Non Volatile Configuration Register. The suspended program or erase cycle can be resumed by mean of the Program/Erase Resume instruction (PER). During the program/erase cycles also the polling instructions (to check if the internal modify cycle is finished by mean of the WIP bit of the Status Register or of the Program/Erase controller bit of the Flag Status register) are also accepted to allow the application checking the end of the internal modify cycles, of course these polling instructions don't affect the internal cycles performing.

Table 25. Instruction set: QIO-SPI protocol (page 1 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
MIORDID	Multiple I/O read identification	1010 1111	AFh	0	0	1 to 3
RDSFDP	Read Serial Flash Discovery Parameter	01011010	5Ah	3	8	1 to ∞
QCFR	Quad Command Fast Read	0000 1011	0Bh	3/4(1)	10 (2)	1 to ∞
		0110 1011	6Bh	3/4(1)	10 **(2)**	1 to ∞
		1110 1011	EBh	3/4(1)	10 **(2)**	1 to ∞
QCFR4BYTE	Quad Command Fast Read using 4 Byte Address	0000 1100	0Ch	4	10 **(2)**	1 to ∞
		0110 1100	6Ch	4	10 **(2)**	1 to ∞
		1110 1100	ECh	4	10 **(2)**	1 to ∞
ROTP	Read OTP (Read of OTP area)	0100 1011	4Bh	3/4(1)	10 **(2)**	1 to 65
WREN	Write Enable	0000 0110	06h	0	0	0
WRDI	Write Disable	0000 0100	04h	0	0	0
QCPP	Quad Command Page Program	0000 0010	02h	3/4(1)	0	1 to 256
		0011 0010	32h	3/4(1)	0	1 to 256
		0001 0010	12h	3/4(1)	0	1 to 256
POTP	Program OTP (Program of OTP area)	0100 0010	42h	3/4(1)	0	1 to 65
SSE	SubSector Erase	0010 0000	20h	3/4(1)	0	0
SE	Sector Erase	1101 1000	D8h	3/4(1)	0	0
BE	Bulk Erase	1100 0111	C7h	0	0	0
PER	Program/Erase Resume	0111 1010	7Ah	0	0	0
PES	Program/Erase Suspend	0111 0101	75h	0	0	0
RDSR	Read Status Register	0000 0101	05h	0	0	1 to ∞
WRSR	Write Status Register	0000 0001	01h	0	0	1
RDLR	Read Lock Register	1110 1000	E8h	3/4(1)	0	1 to ∞
WRLR	Write to Lock Register	1110 0101	E5h	3/4(1)	0	1
RFSR	Read Flag Status Register	0111 0000	70h	0	0	1 to ∞
CLFSR	Clear Flag Status Register	0101 0000	50h	0	0	0
RDNVCR	Read NV Configuration Register	1011 0101	B5h	0	0	2
WRNVCR	Write NV Configuration Register	1011 0001	B1h	0	0	2
RDVCR	Read Volatile Configuration Register	1000 0101	85h	0	0	1 to ∞

Table 25. Instruction set: QIO-SPI protocol (page 2 of 2)

Instruction	Description	One-byte Instruction Code (BIN)	One-byte Instruction Code (HEX)	Address bytes	Dummy clock cycle	Data bytes
WRVCR	Write Volatile Configuration Register	1000 0001	81h	0	0	1
RDVECR	Read Volatile Enhanced Configuration Register	0110 0101	65h	0	0	1 to ∞
WRVECR	Write Volatile Enhanced Configuration Register	0110 0001	61h	0	0	1
EN4BYTEADDR	Enter 4-byte address mode	1011 0111	B7h	0	0	0
EX4BYTEADDR	Exit 4-byte address mode	1110 1001	E9h	0	0	0
WREAR	Write Extended Address Register	1100 0101	C5h	0	0	0
RDEAR	Read Extended Address Register	1100 1000	C8h	0	0	0
RSTEN	Reset Enable	0110 0110	66h	0	0	0
RST	Reset Memory	1001 1001	99h	0	0	0

1. Accordingly to address mode (default is 3).
2. The number of Dummy Clock cycles is configurable by the user.

9.3.1 Multiple I/O Read Identification (MIORDID)

The Multiple Input/Output Read Identification (MIORDID) instruction allows to read the device identification data in the QIO-SPI protocol:

- Manufacturer identification (1 byte)
- Device identification (2 bytes)

Unlike the RDID instruction of the Extended SPI protocol, the Multiple Input/Output instruction can not read the Unique ID code (UID) (17 bytes).

For further details on the manufacturer and device identification codes, see *9.1.1: Read Identification (RDID)*.

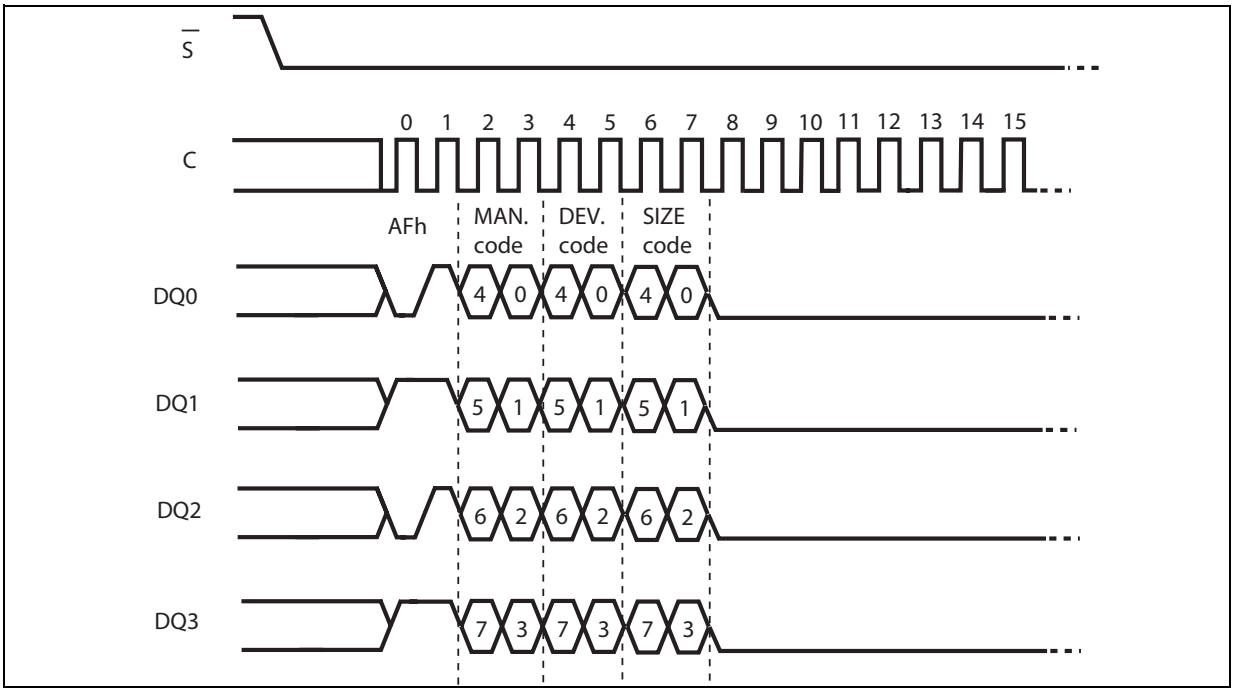
Any Multiple Input/Output Read Identification (MIORDID) instruction while an Erase or Program cycle is in progress, is not decoded, and has no effect on the cycle that is in progress.

The device is first selected by driving Chip Select ($\overline{S}$) Low. Then, the 8-bit instruction code for the instruction is shifted in parallel on the 4 pins DQ0, DQ1, DQ2 and DQ3. After this, the 24-bit device identification, stored in the memory, will be shifted out on again in parallel on DQ0, DQ1, DQ2 and DQ3. The identification bits are shifted out 4 at a time during the falling edge of Serial Clock (C).

The Read Identification (RDID) instruction is terminated by driving Chip Select ($\overline{S}$) High at any time during data output.

When Chip Select ($\overline{S}$) is driven High, the device is put in the Standby Power mode. Once in the Standby Power mode, the device waits to be selected, so that it can receive, decode and execute instructions.

Figure 95. Multiple I/O Read Identification instruction and data-out sequence QIO-SPI



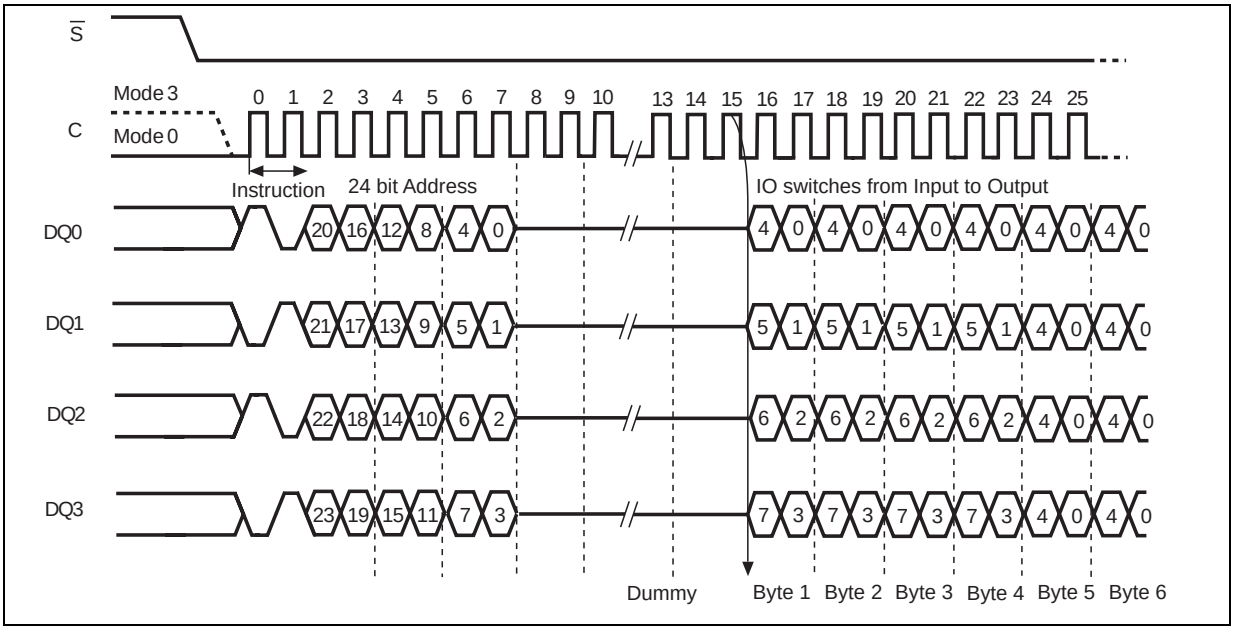
9.3.2 Read Serial Flash Discovery Parameter

The Read Serial Flash Discovery Parameter (RDSFDP) instruction allows reading the Serial Flash Discovery Parameter area (SFDP) in the QIO-SPI protocol. The instruction functionality is exactly the same as the Read Serial Flash Discovery Parameter instruction of the Extended SPI protocol. The only difference is that in the QIO-SPI protocol instruction code, address and output data are all parallelized on the four pins DQ0, DQ1, DQ2 and DQ3.

The serial flash discovery parameter area is always addressable by means 3 byte regardless active address mode.

Note: The dummy byte bits can not be parallelized: 8 clock cycles are requested to perform the internal reading operation.

Figure 96. Quad Read Serial Flash Discovery Parameter



9.3.3 Quad Command Fast Read (QCFR)

The Quad Command Fast Read (QCFR) instruction allows to read the memory in QIO-SPI protocol, parallelizing the instruction code, the address and the output data on four pins (DQ0, DQ1, DQ2 and DQ3). The Quad Command Fast Read (QCFR) instruction can be issued, after the device is set in QIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 0Bh, 6Bh or EBh, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to QIO-SPI protocol.

Apart for the parallelizing on four pins of the instruction code, the Quad Command Fast Read instruction functionality is exactly the same as the Quad I/O Fast Read of the Extended SPI protocol, please refer to *Section 9.1.13: Quad I/O Fast Read* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 97. Quad Command Fast Read instruction and data-out sequence QSP, 0Bh

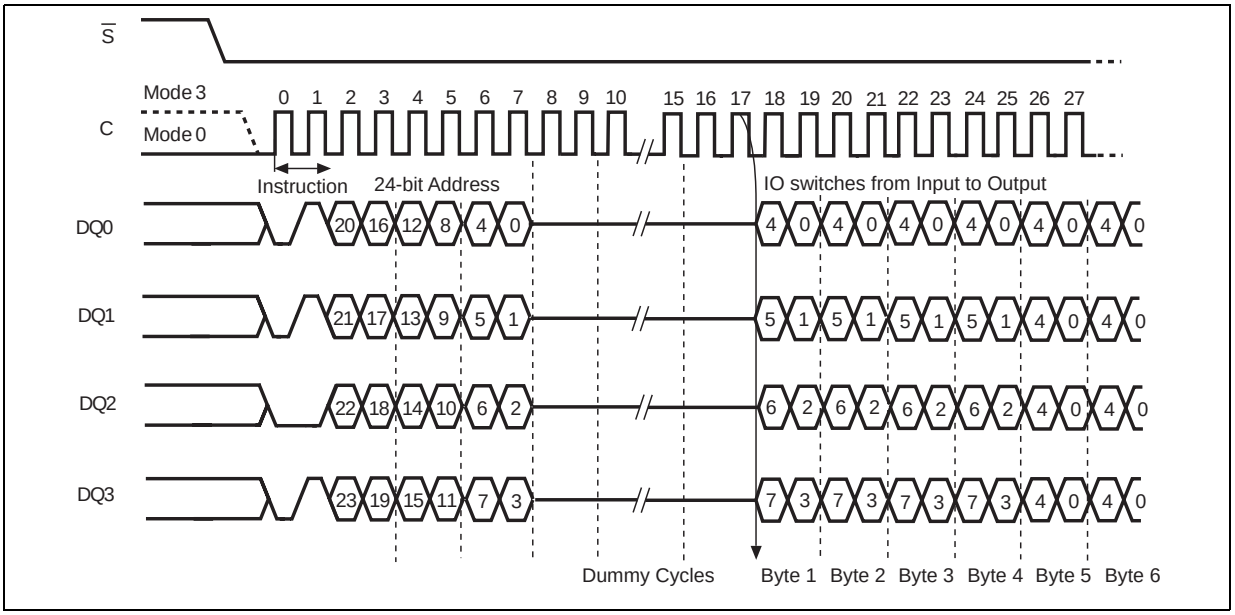


Figure 98. Quad Command Fast Read instruction and data-out sequence QSP, 6Bh

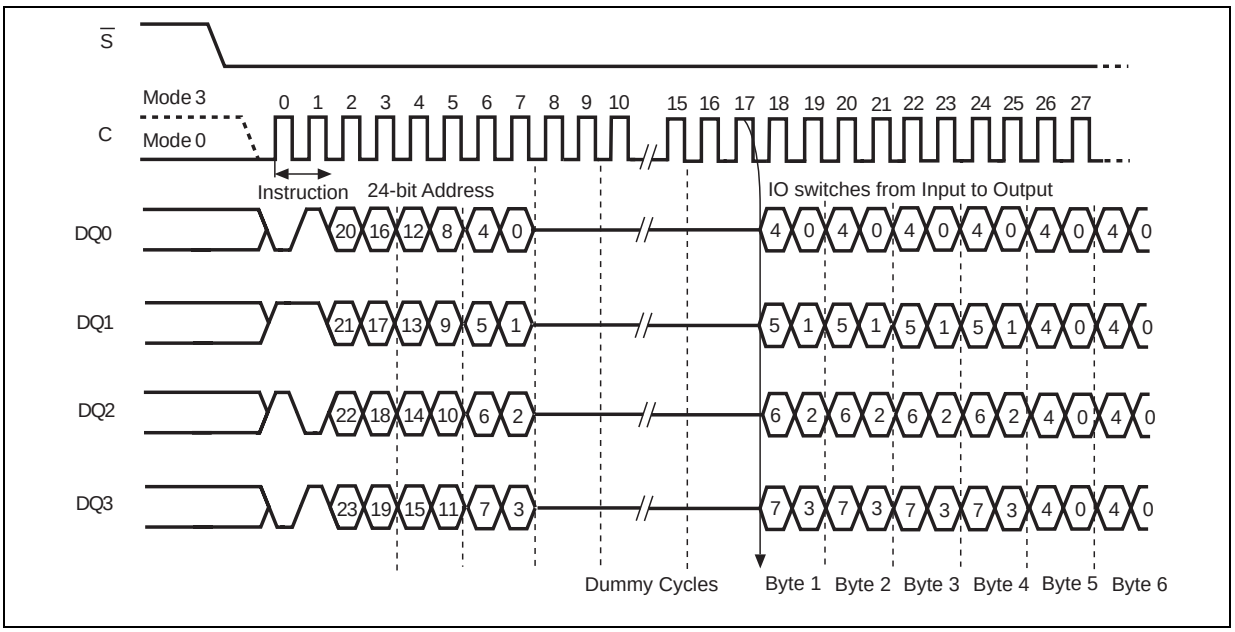
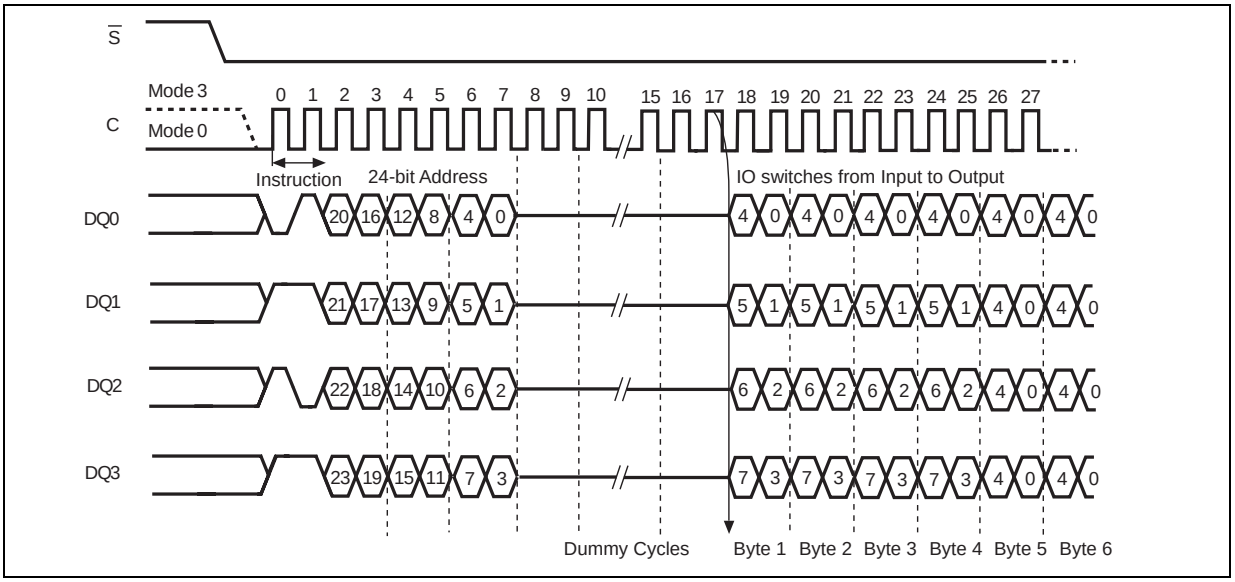


Figure 99. Quad Command Fast Read instruction and data-out sequence QSP, EBh



9.3.4 Quad Command Fast Read using 4 Byte Address (QCFR4Byte)

The Quad Command Fast Read using 4 byte address (QCFR4BYTE) instruction allows to read the memory in QIO-SPI protocol, parallelizing the instruction code, the address and the output data on four pins (DQ0, DQ1, DQ2 and DQ3). The instruction can be issued, after the device is set in QIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 0Ch, 6Ch or ECh, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to QIO-SPI protocol.

Apart for the parallelizing on four pins of the instruction code, the Quad Command Fast Read using 4 byte address instruction functionality is exactly the same as the Quad I/O Fast Read of the Extended SPI protocol.

Figure 100. Quad Command Fast Read using 4 Byte Address Instruction QSP, 0Ch

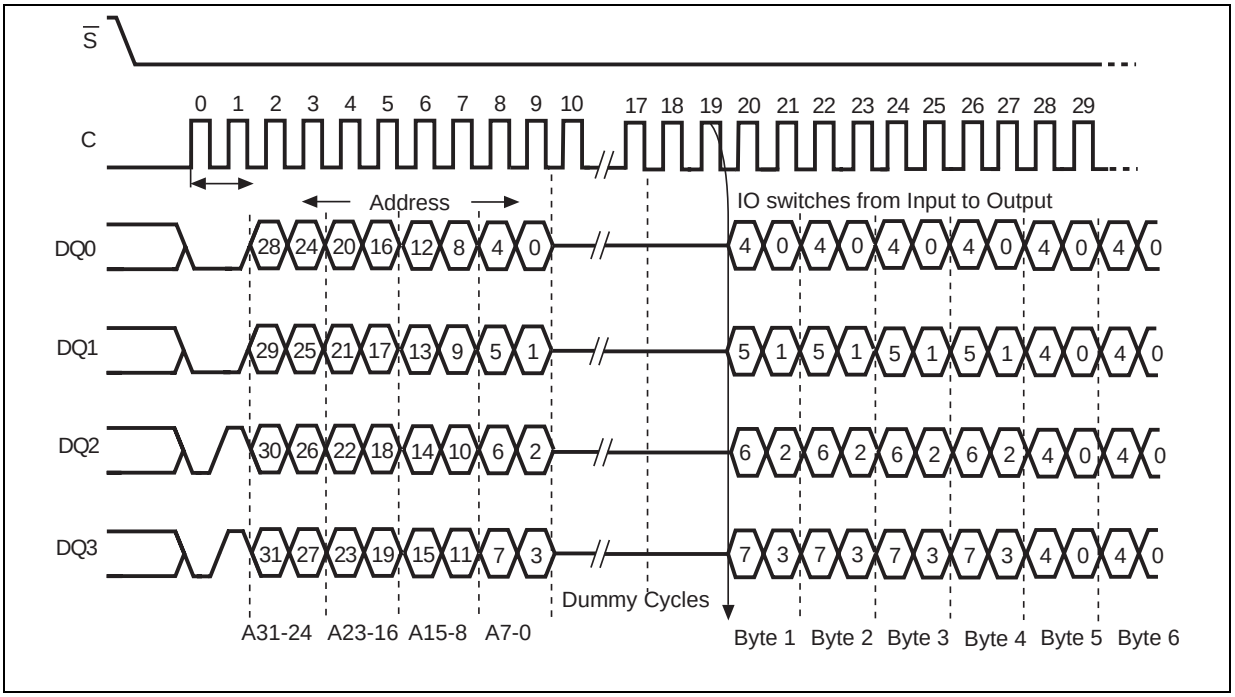


Figure 101. Quad Command Fast Read using 4 Byte Address Instruction QSP, 6Ch

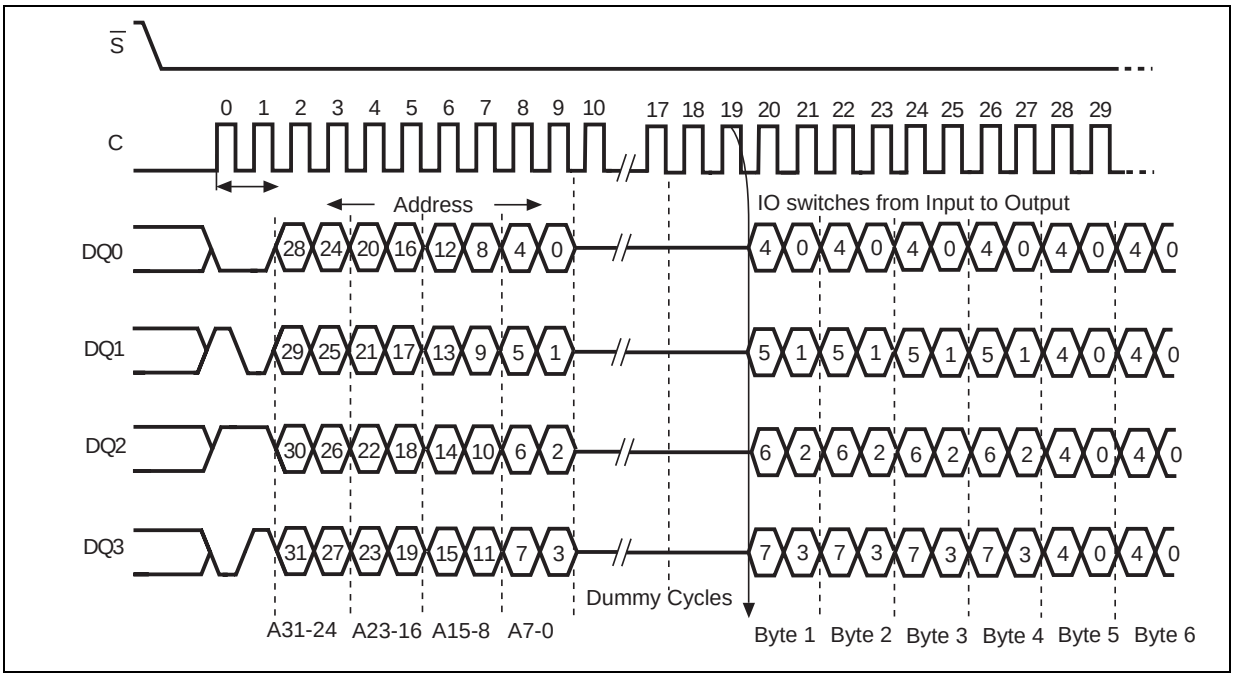
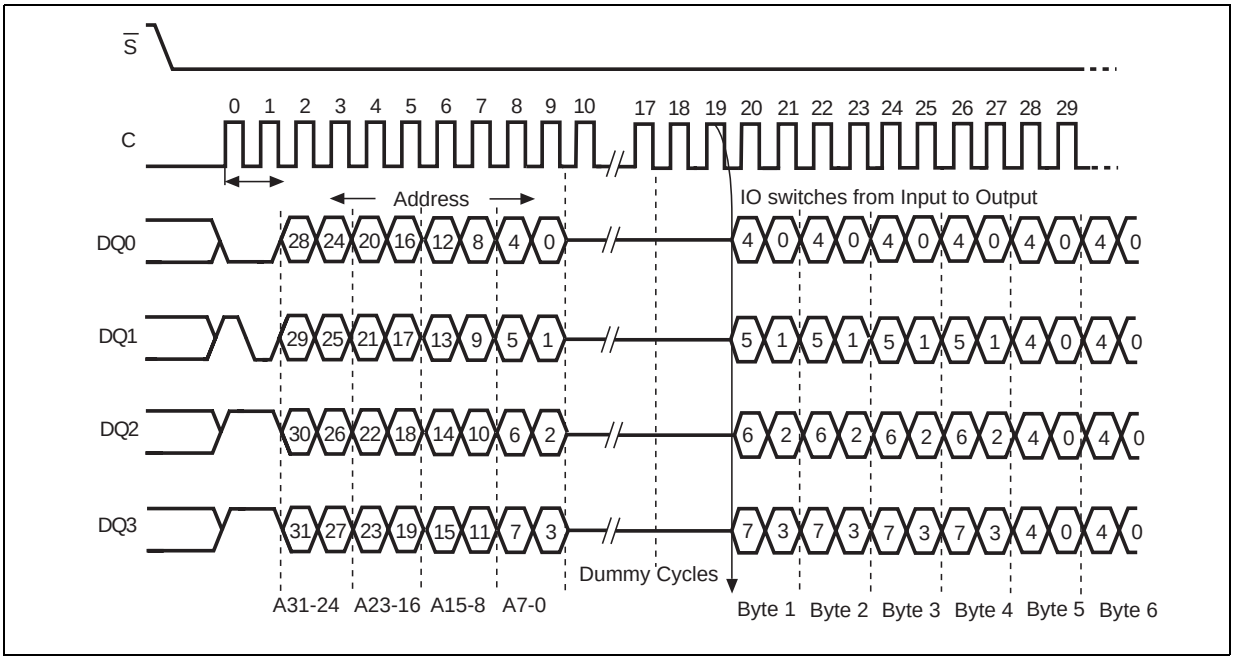


Figure 102. Quad Command Fast Read using 4 Byte Address Instruction QSP, ECh



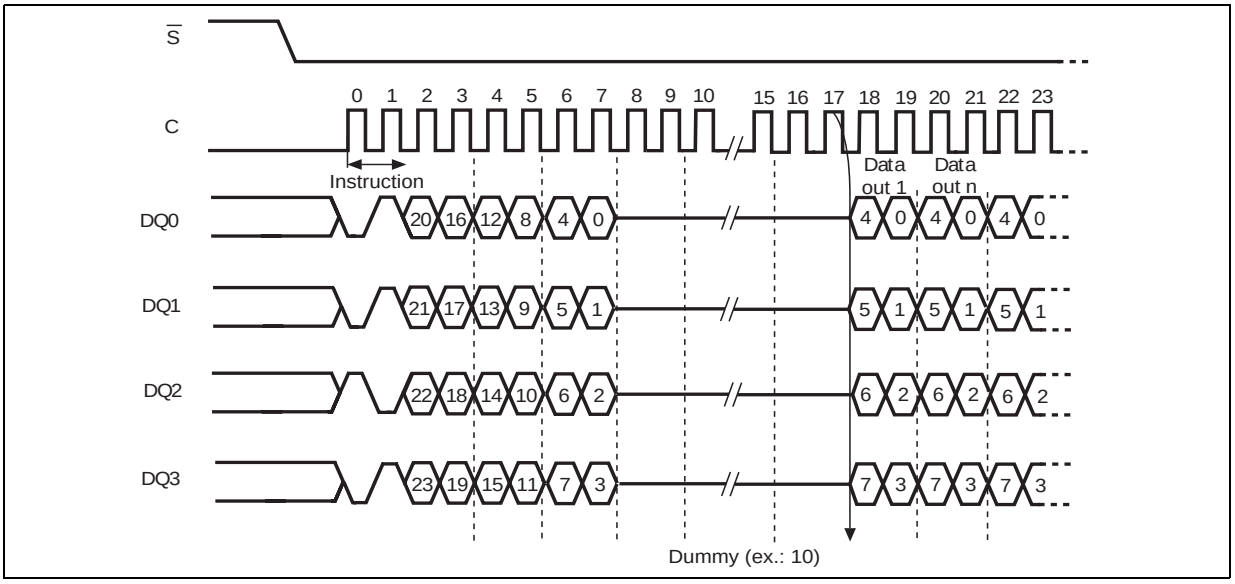
9.3.5 Read OTP (ROTP)

The Read OTP (ROTP) instruction is used to read the 64 bytes OTP area in the QIO-SPI protocol. The instruction functionality is exactly the same as the Read OTP instruction of the Extended SPI protocol. The only difference is that in the QIO-SPI protocol instruction code, address and output data are all parallelized on the four pins DQ0, DQ1, DQ2 and DQ3.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Note: The dummy byte bits can not be parallelized: 8 clock cycles are requested to perform the internal reading operation.

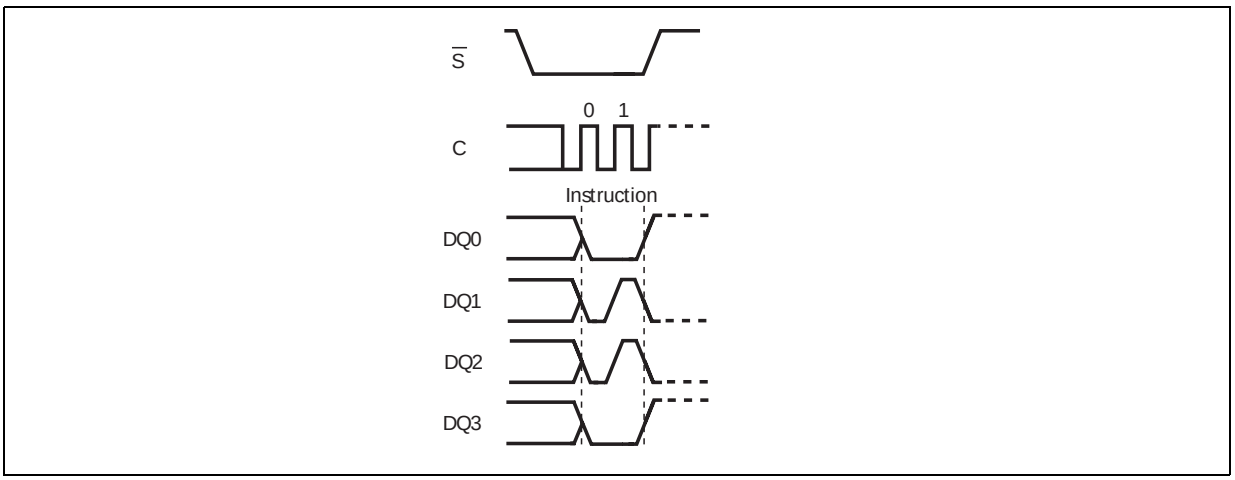
Figure 103. Read OTP instruction and data-out sequence QIO-SPI



9.3.6 Write Enable (WREN)

The Write Enable (WREN) instruction sets the Write Enable Latch (WEL) bit. Apart form the parallelizing of the instruction code on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Write Enable instruction of the Extended SPI protocol, please refer to *Section 9.1.16: Write Enable (WREN)* for further details.

Figure 104. Write Enable instruction sequence QIO-SPI

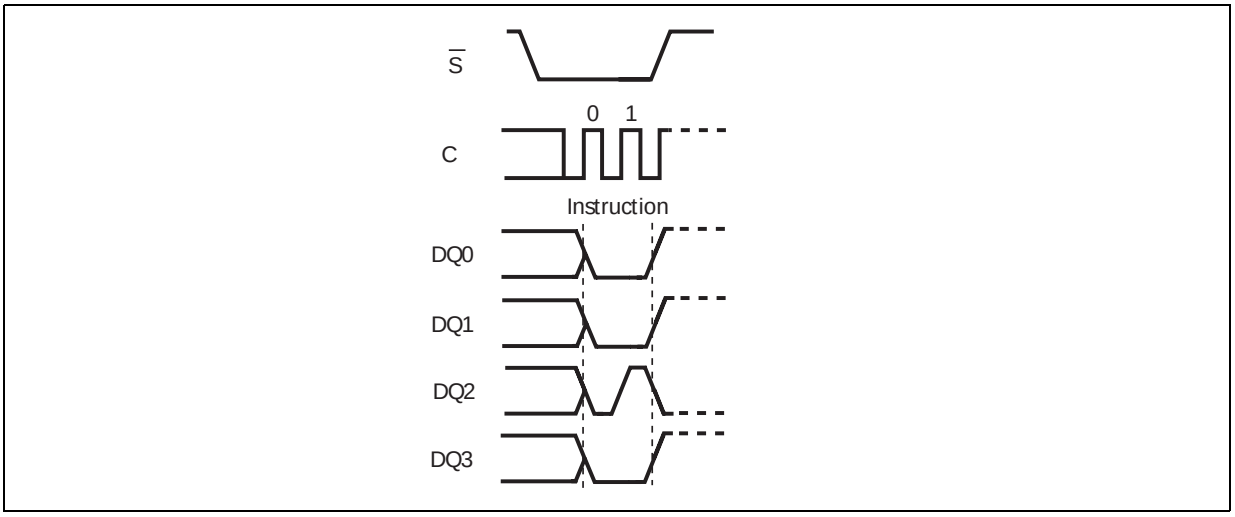


9.3.7 Write Disable (WRDI)

The Write Disable (WRDI) instruction resets the Write Enable Latch (WEL) bit.

Apart form the parallelizing of the instruction code on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Write Disable (WRDI) instruction of the Extended SPI protocol, please refer to *Section 9.1.17: Write Disable (WRDI)* for further details.

Figure 105. Write Disable instruction sequence QIO-SPI



9.3.8 Quad Command Page Program (QCPP)

The Quad Command Page Program (QCPP) instruction allows to program the memory content in DIO-SPI protocol, parallelizing the instruction code, the address and the input data on four pins (DQ0, DQ1, DQ2 and DQ3). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. The Quad Command Page Program (QCPP) instruction can be issued, when the device is set in QIO-SPI mode, by sending to the memory indifferently one of the 3 instructions codes: 02h, 12h or 32h, the effect is exactly the same. The 3 instruction codes are all accepted to help the application code porting from Extended SPI protocol to QIO-SPI protocol.

Apart for the parallelizing on four pins of the instruction code, the Quad Command Page Program instruction functionality is exactly the same as the Quad Input Extended Fast Program of the Extended SPI protocol, please refer to *Section 9.1.22: Quad Input Extended Fast Program* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 106. Quad Command Page Program instruction sequence QIO-SPI, 02h

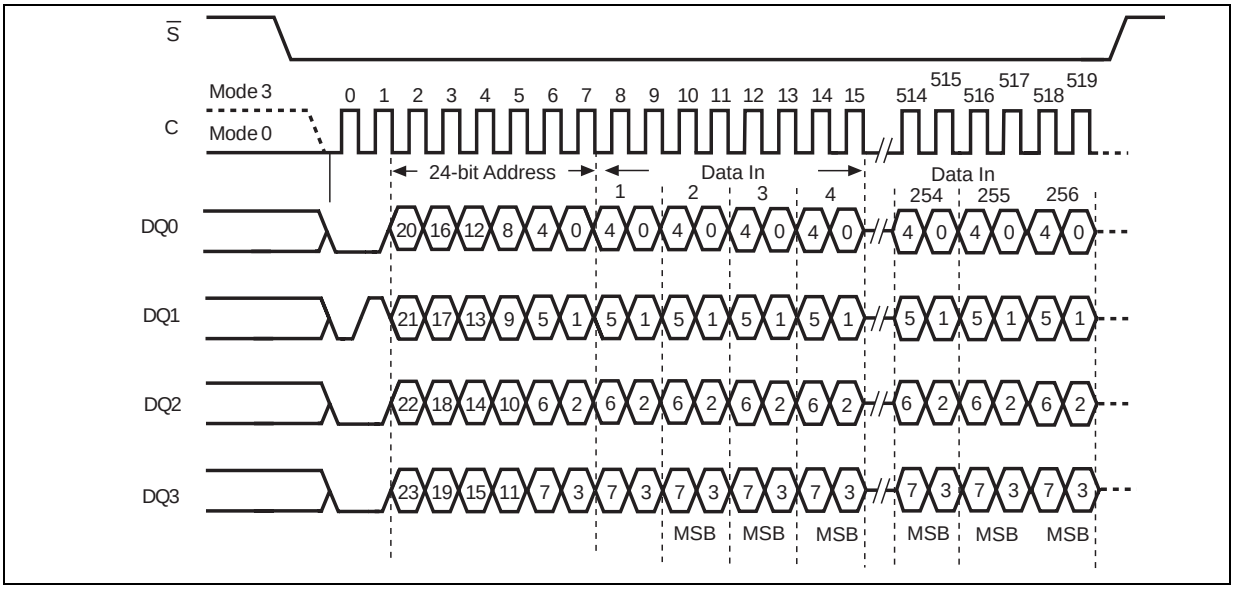


Figure 107. Quad Command Page Program instruction sequence QIO-SPI, 12h

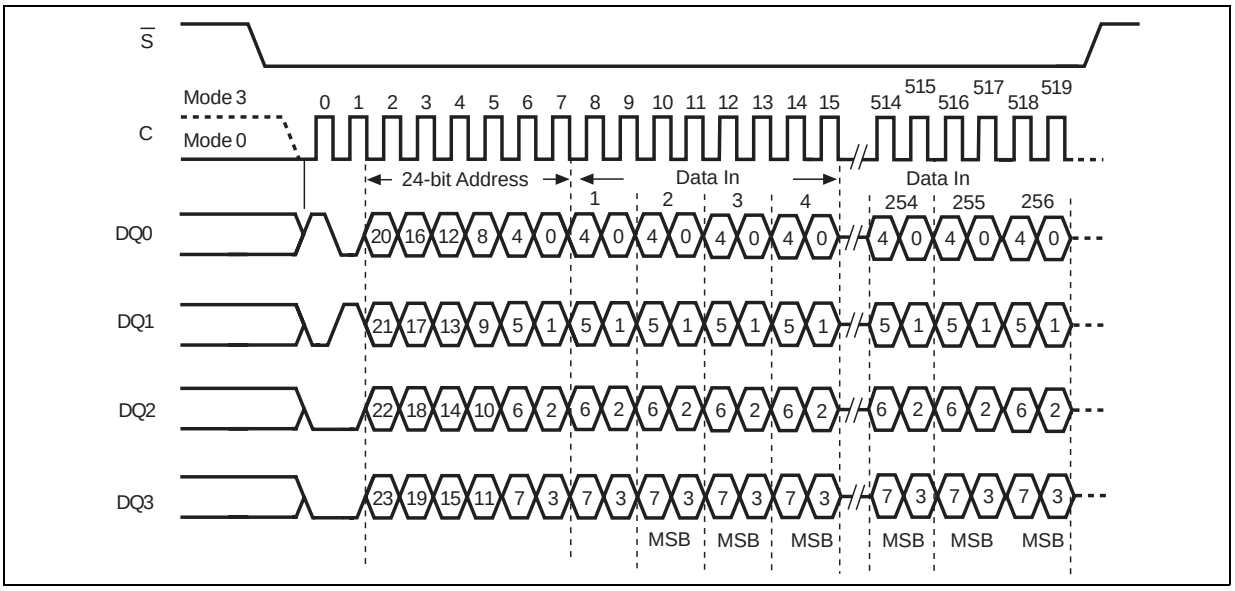
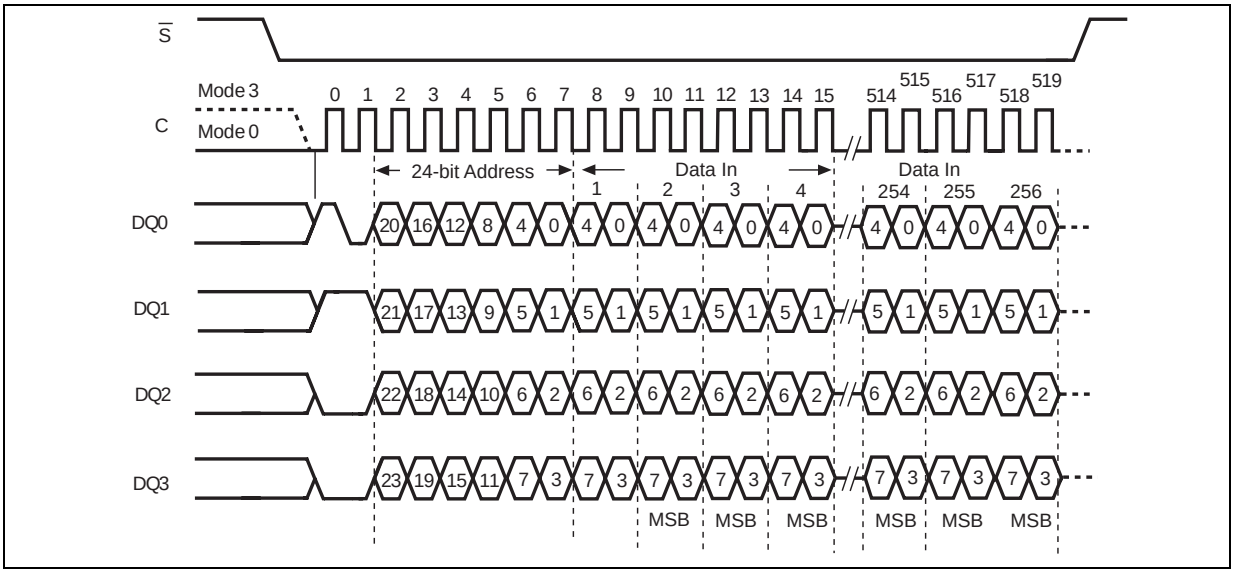


Figure 108. Quad Command Page Program instruction sequence QIO-SPI, 32h



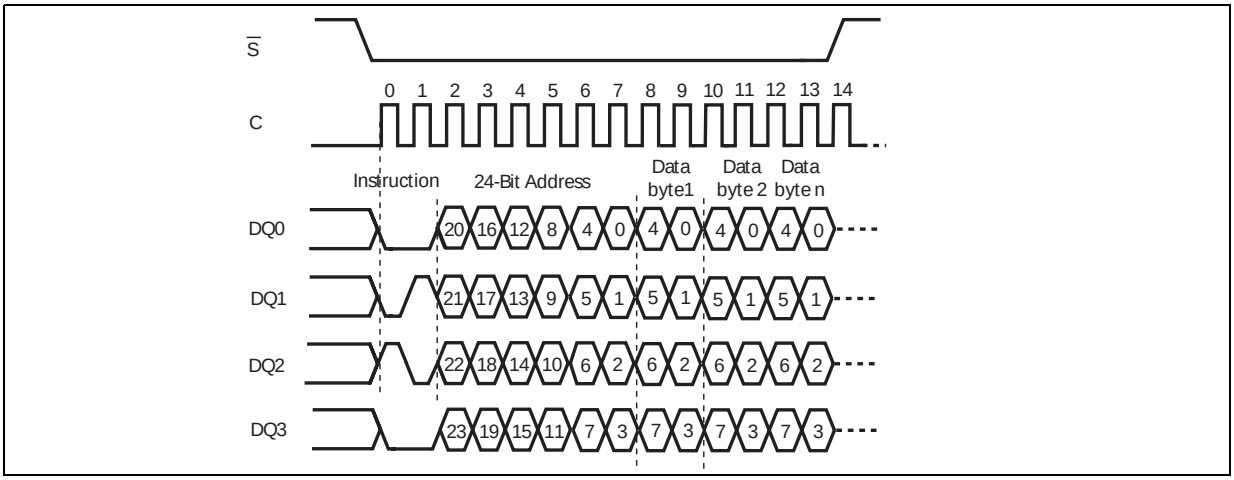
9.3.9 Program OTP instruction (POTP)

The Program OTP instruction (POTP) is used to program at most 64 bytes to the OTP memory area (by changing bits from 1 to 0, only). Before it can be accepted, a Write Enable (WREN) instruction must have been executed.

Apart form the parallelizing of the instruction code, address and input data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality (as well as the locking OTP method) is exactly the same as the Program OTP (POTP) instruction of the Extended SPI protocol, please refer to *Section 9.1.23: Program OTP instruction (POTP)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 109. Program OTP instruction sequence QIO-SPI



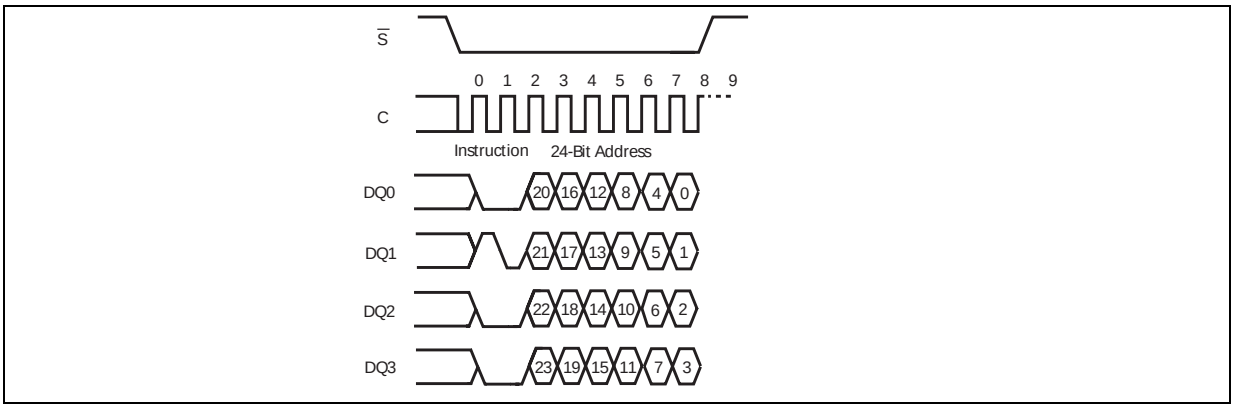
9.3.10 Subsector Erase (SSE)

The Subsector Erase (SSE) instruction sets to '1' (FFh) all bits inside the chosen subsector. Before it can be accepted, a Write Enable (WREN) instruction must have been executed.

Apart form the parallelizing of the instruction code and the address on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Subsector Erase (SSE) instruction of the Extended SPI protocol, please refer to *Section 9.1.24: Subsector Erase (SSE)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 110. Subsector Erase instruction sequence QIO-SPI



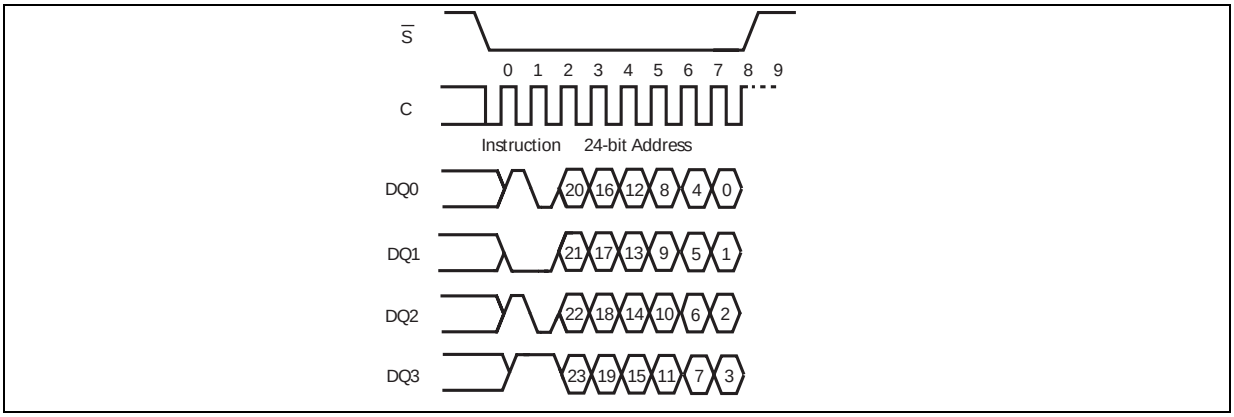
9.3.11 Sector Erase (SE)

The Sector Erase (SE) instruction sets to '1' (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must have been executed.

Apart form the parallelizing of the instruction code and the address on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Sector Erase (SE) instruction of the Extended SPI protocol, please refer to *Section 9.1.25: Sector Erase (SE)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 111. Sector Erase instruction sequence QIO-SPI

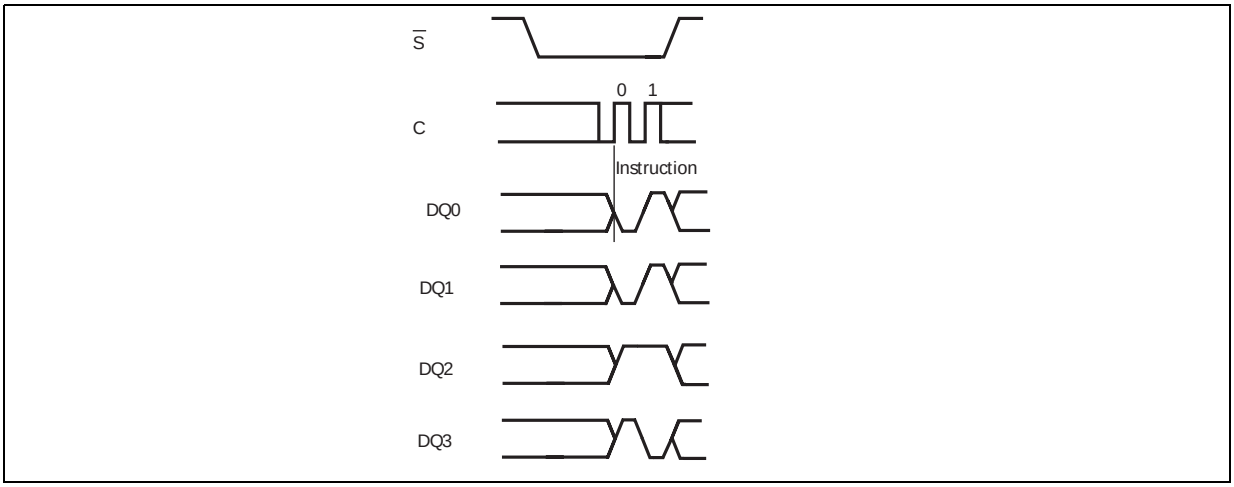


9.3.12 Bulk Erase (BE)

The Bulk Erase (BE) instruction sets all bits to '1' (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

Apart form parallelizing the instruction code on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Bulk Erase (BE) instruction of the Extended SPI protocol, please refer to *Section 9.1.26: Bulk Erase (BE)* for further details.

Figure 112. Bulk Erase instruction sequence QIO-SPI

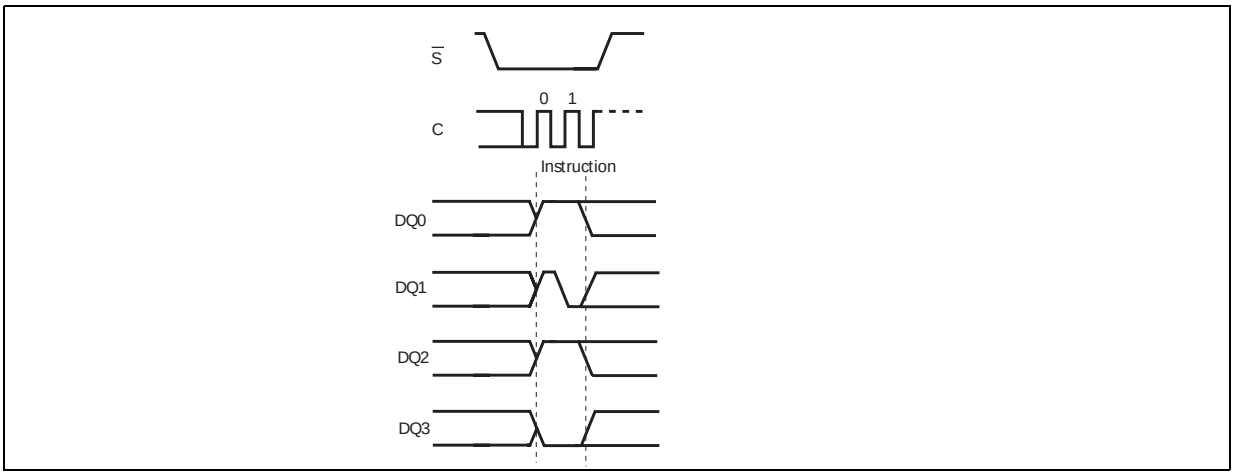


9.3.13 Program/Erase Suspend

The Program/Erase Suspend instruction allows the controller to interrupt a Program or an Erase instruction, in particular: Sector Erase and Quad Command Page Program can be suspended and erased while that Subsector Erase, Bulk Erase, Write Non Volatile Configuration register and Program OTP can not be suspended.

Apart form parallelizing the instruction code on four pins (DQ0, DQ1, DQ2, DQ3) the instruction functionality is the same as the Program/Erase Suspend (PES) instruction of the Extended SPI protocol. Refer to *Section 9.1.27: Program/Erase Suspend* for further details.

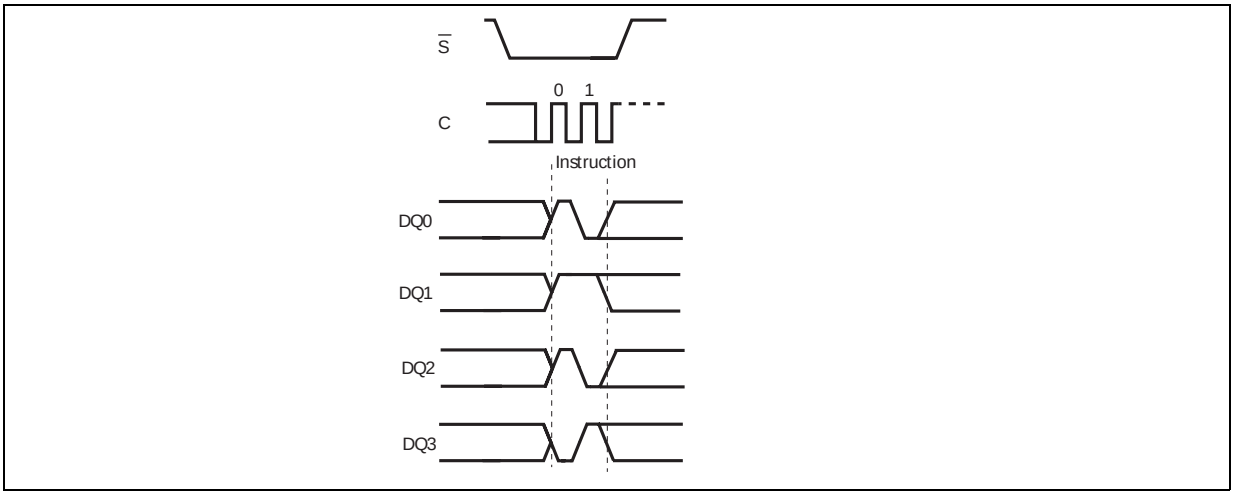
Figure 113. Program/Erase Suspend instruction sequence QIO-SPI



9.3.14 Program/Erase Resume

After a Program/Erase suspend instruction, a Program/Erase Resume instruction is required to continue performing the suspended Program or Erase sequence. Apart form parallelizing the instruction code on four pins (DQ0, DQ1, DQ2, DQ3) the instruction functionality is the same as the Program/Erase Resume (PER) instruction of the Extended SPI protocol. Refer to *Section 9.1.28: Program/Erase Resume* for further details.

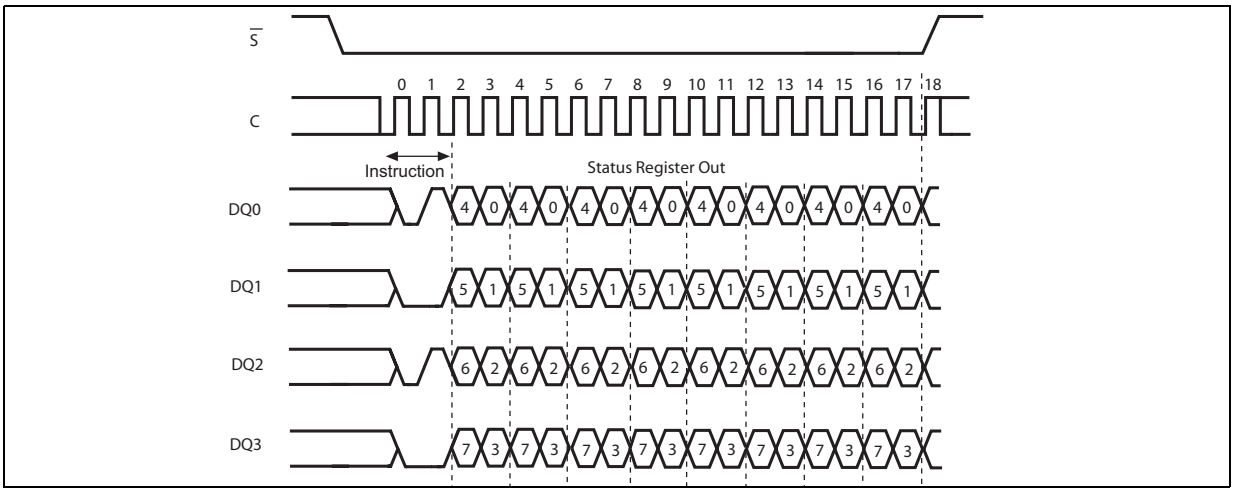
Figure 114. Program/Erase Resume instruction sequence QIO-SPI



9.3.15 Read Status Register (RDSR)

The Read Status Register (RDSR) instruction allows the Status Register to be read. Apart form the parallelizing of the instruction code and the output data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Read Status Register (RDSR) instruction of the Extended SPI protocol, please refer to *Section 9.1.29: Read Status Register (RDSR)* for further details.

Figure 115. Read Status Register instruction sequence QIO-SPI

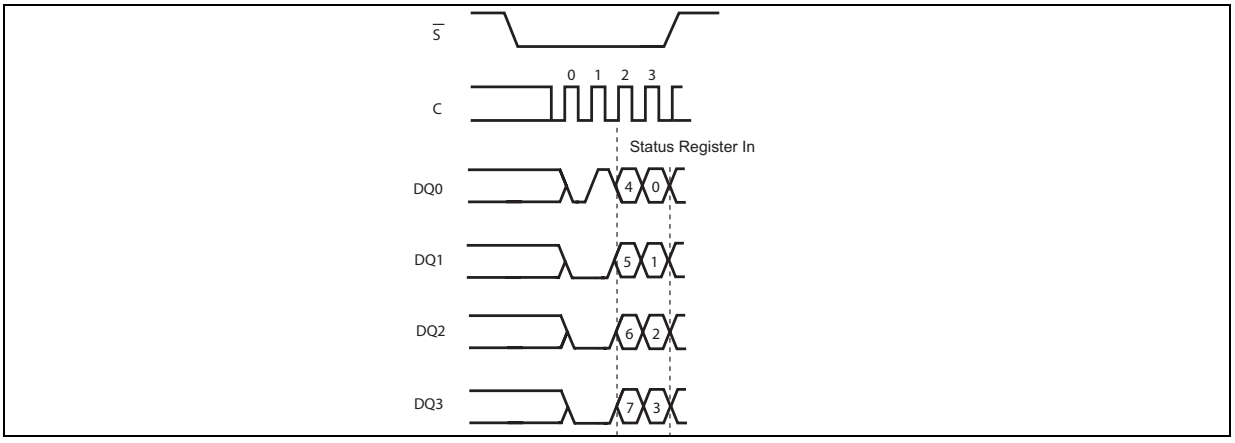


9.3.16 Write status register (WRSR)

The write status register (WRSR) instruction allows new values to be written to the status register. Before it can be accepted, a write enable (WREN) instruction must have been executed. The instruction code and input data are sent on four pins DQ0, DQ1, DQ2 DQ3. The instruction functionality is exactly the same as the Write Status Register (WRSR) instruction of the Extended SPI protocol (See *Section 9.1.30: Write status register (WRSR)*).

However, the protection feature management is different. In particular, once SRWD bit is set to '1' the device enters in the hardware protected mode (HPM) independently from Write Protect (W/VPP) signal value. To exit the HPM mode is needed to switch temporarily to the Extended SPI protocol.

Figure 116. Write Status Register instruction sequence QIO-SPI

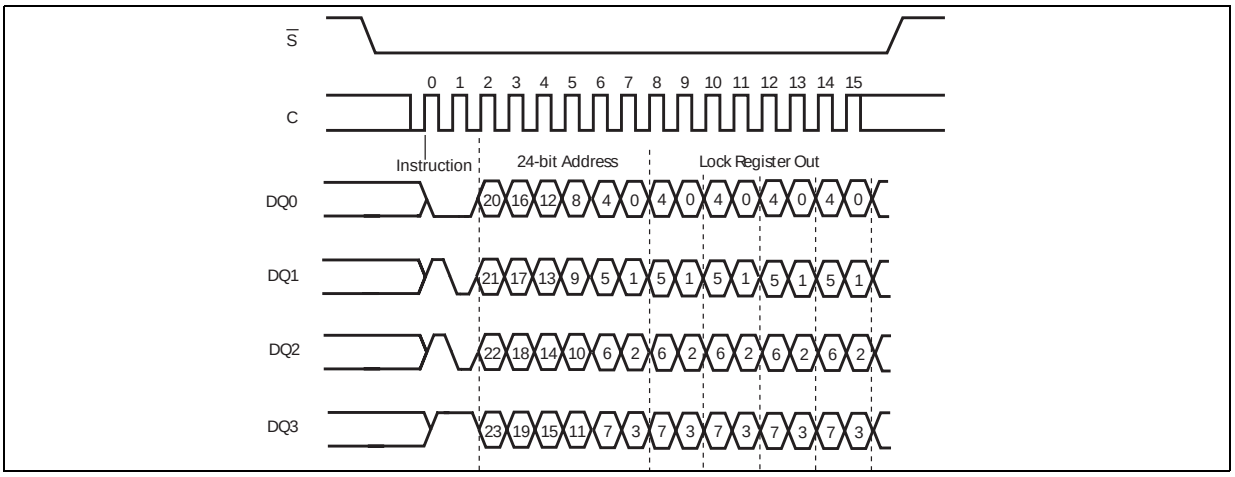


9.3.17 Read Lock Register (RDLR)

The Read Lock Register instructions is used to read the lock register content. Apart from parallelizing the instruction code, the address, and the output data on four pins (DQ0, DQ1, DQ2, DQ3) the instruction functionality is the same as the Read Lock Register (RDLR) instruction of the Extended SPI protocol. Refer to *Section 9.1.31: Read Lock Register (RDLR)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 117. Read Lock Register instruction and data-out sequence QIO-SPI

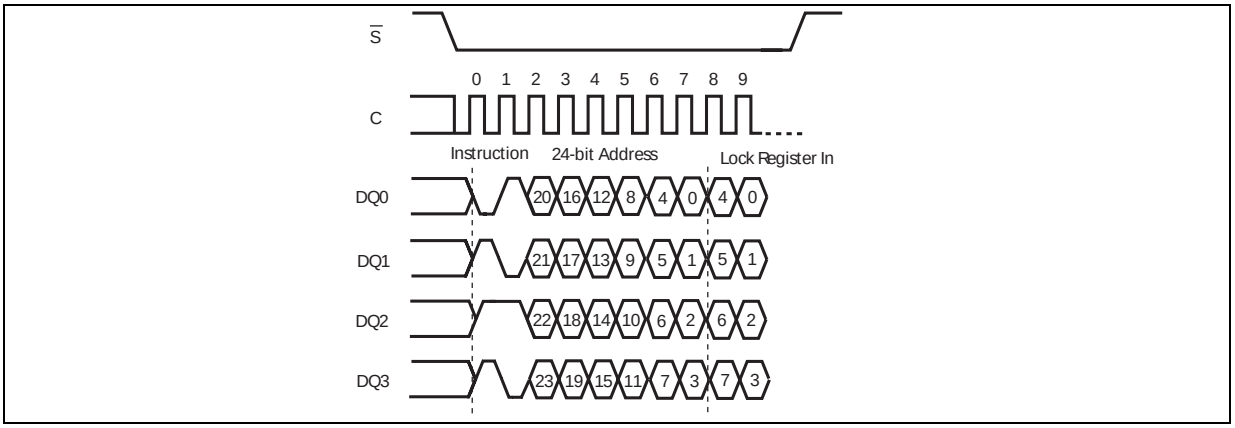


9.3.18 Write to Lock Register (WRLR)

The Write to Lock Register (WRLR) instruction allows bits to be changed in the Lock Registers. Before it can be accepted, a Write Enable (WREN) instruction must have been executed. Apart form the parallelizing of the instruction code, the address, and the input data on the four pins DQ0, DQ1, DQ2 DQ3, the instruction functionality is exactly the same as the Write to Lock Register (WRLR) instruction of the Extended SPI protocol. Please refer to *Section 9.1.32: Write to Lock Register (WRLR)* for further details.

If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Figure 118. Write to Lock Register instruction sequence QIO-SPI

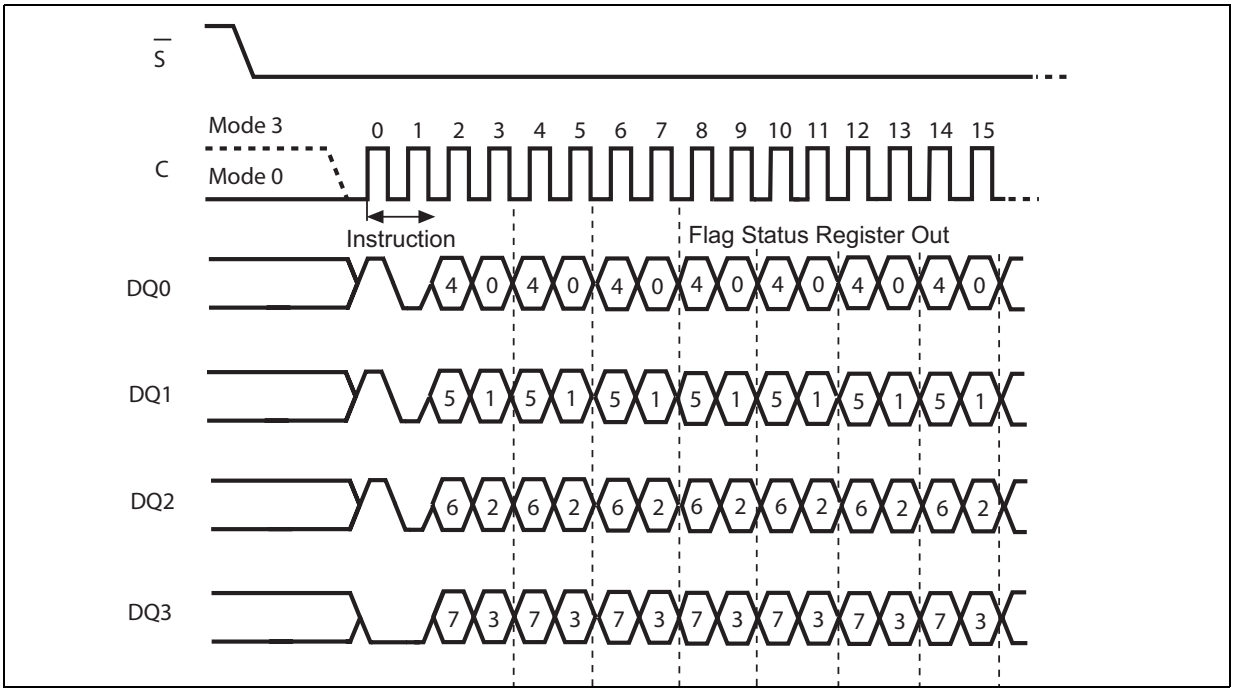


9.3.19 Read Flag Status Register

The Read Flag Status Register (RFSR) instruction allows the Flag Status Register to be read.

Apart form the parallelizing of the instruction code and the output data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Read Flag Status Register (RFSR) instruction of the Extended SPI protocol, please refer to *Section 9.1.33: Read Flag Status Register* for further details.

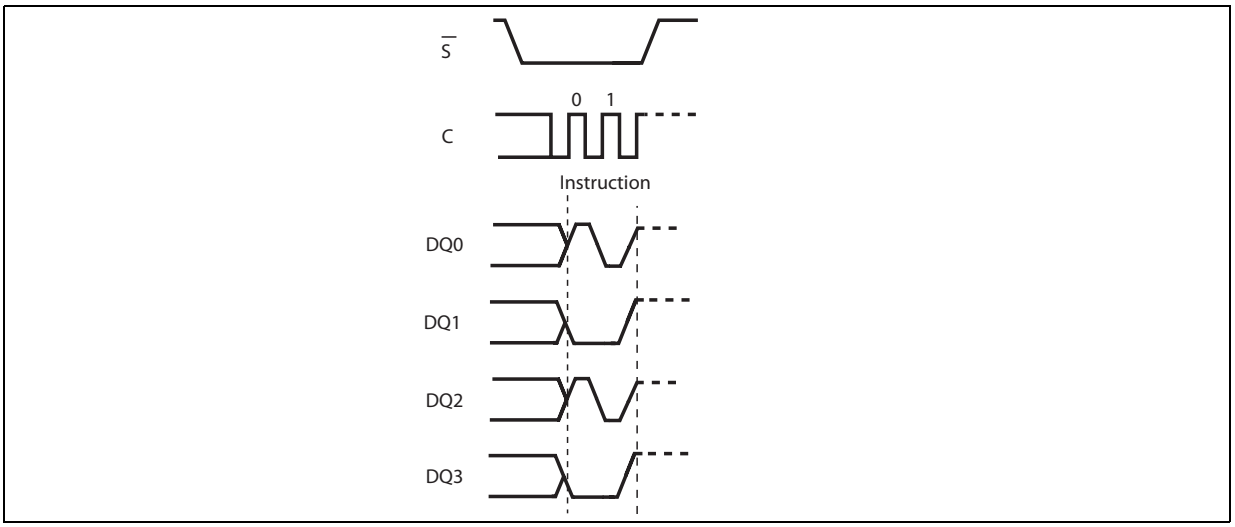
Figure 119. Read Flag Status Register instruction sequence QIO-SPI



9.3.20 Clear Flag Status Register

The Clear Flag Status Register (CLFSR) instruction reset the error Flag Status Register bits (Erase Error bit, Program Error bit, VPP Error bit, Protection Error bit). It is not necessary to set the WEL bit before the Clear Flag Status Register instruction is executed. The WEL bit will be unchanged after this command is executed.

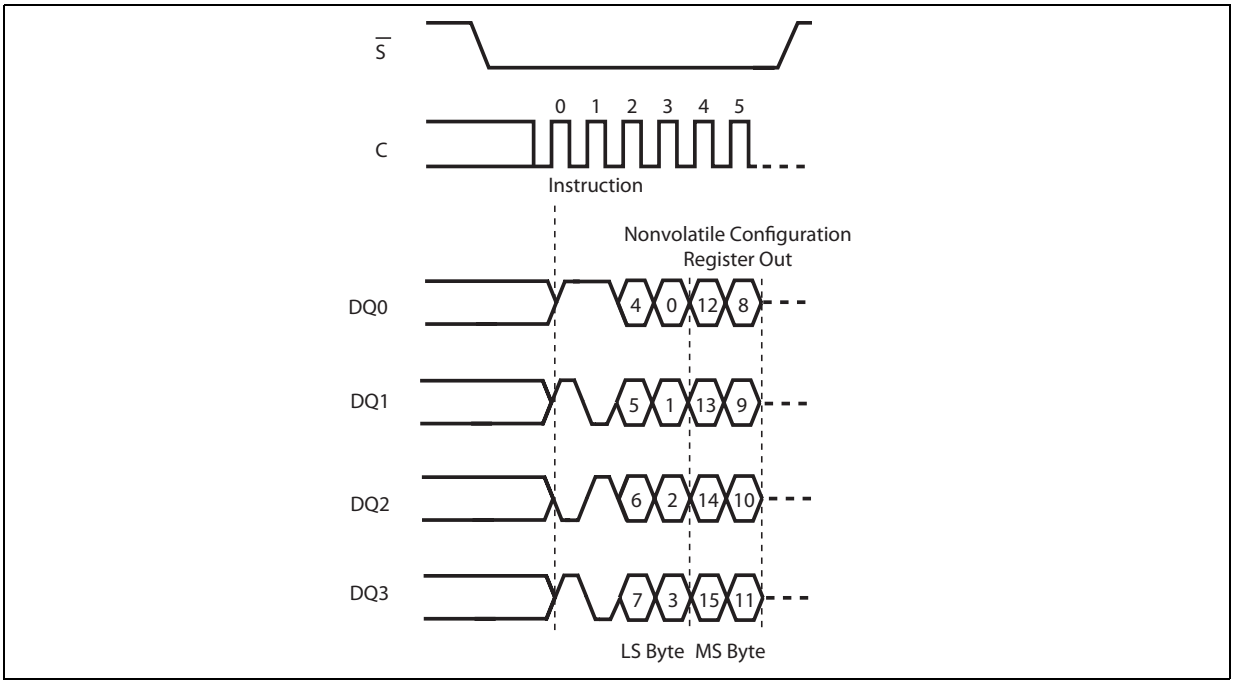
Figure 120. Clear Flag Status Register instruction sequence QIO-SPI



9.3.21 Read NV Configuration Register

The Read Non Volatile Configuration Register (RDNVCR) instruction allows the Non Volatile Configuration Register to be read.

Figure 121. Read NV Configuration Register instruction sequence QIO-SPI

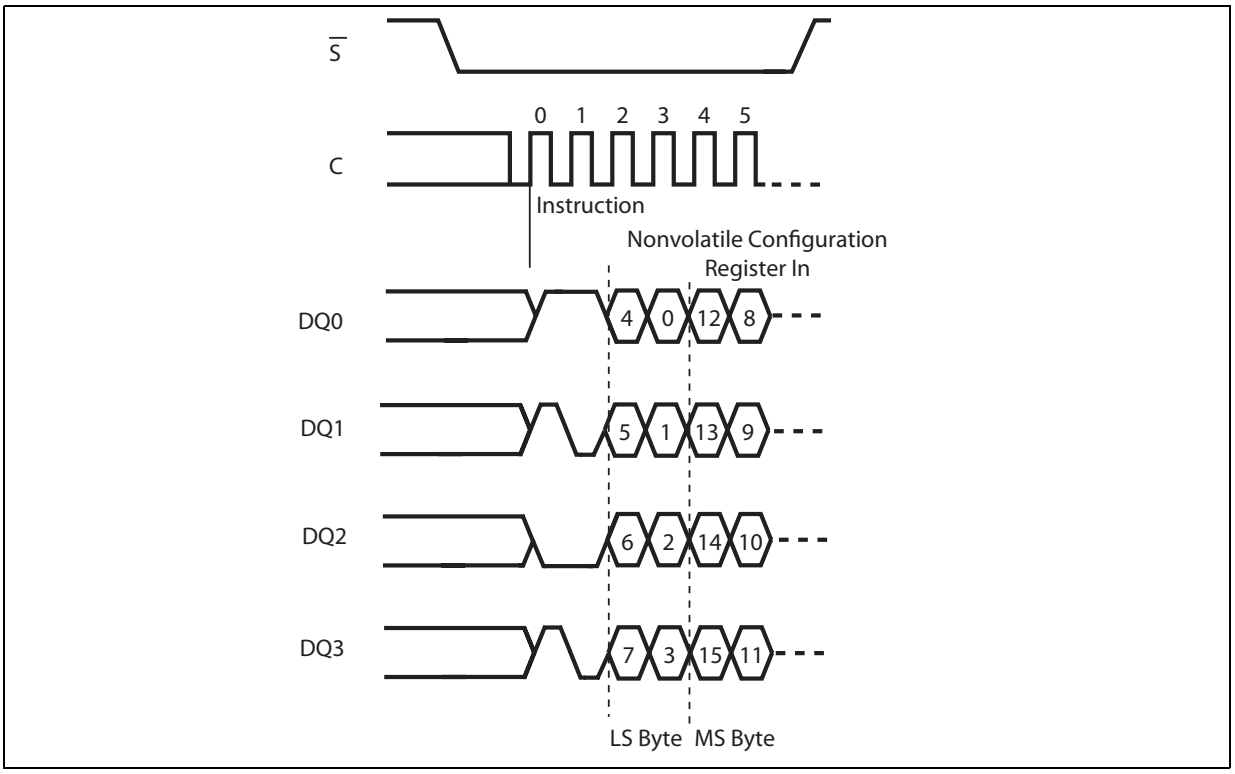


9.3.22 Write NV Configuration Register

The Write Non Volatile Configuration register (WRNVCR) instruction allows new values to be written to the Non Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code and the input data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Write Non Volatile Configuration Register (WRNVCR) instruction of the Extended SPI protocol, please refer to *Section 9.1.36: Write NV Configuration Register* for further details.

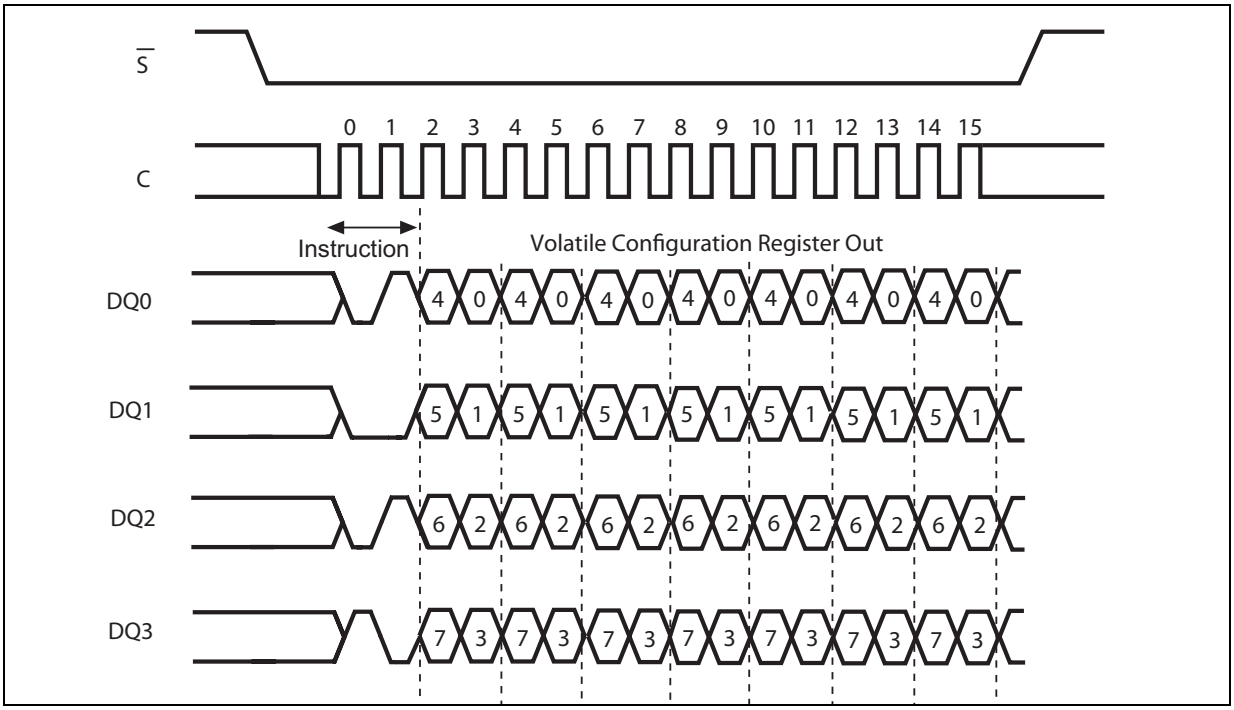
Figure 122. Write NV Configuration Register instruction sequence QIO-SPI



9.3.23 Read Volatile Configuration Register

The Read Volatile Configuration Register (RDVCR) instruction allows the Volatile Configuration Register to be read.

Figure 123. Read Volatile Configuration Register instruction sequence QIO-SPI

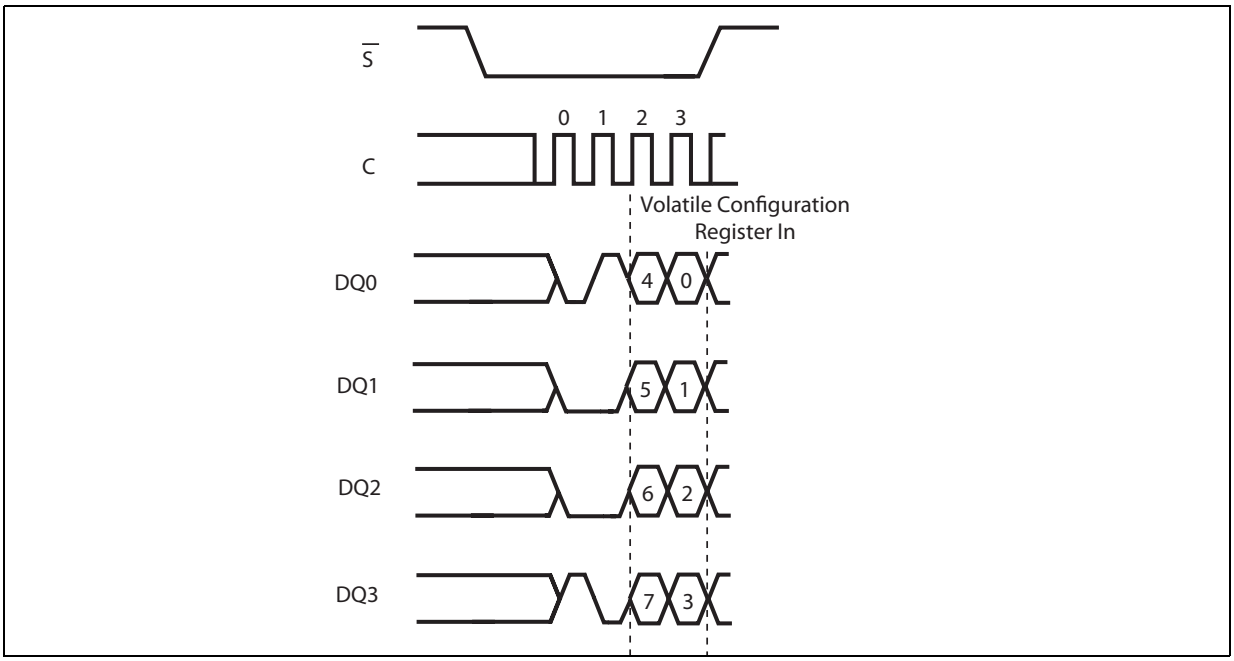


9.3.24 Write Volatile Configuration Register

The Write Volatile Configuration register (WRVCR) instruction allows new values to be written to the Volatile Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code and the input data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Write Volatile Configuration Register (WRVCR) instruction of the Extended SPI protocol, please refer to *Section 9.1.38: Write Volatile Configuration Register* for further details.

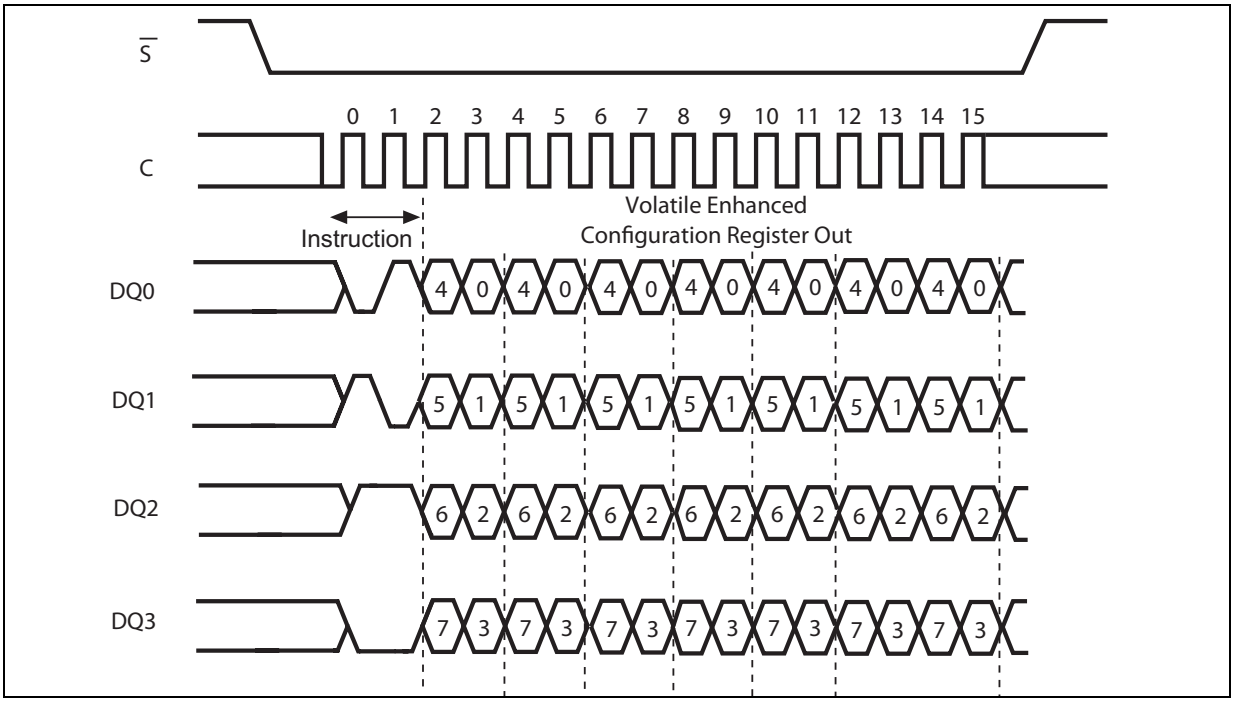
Figure 124. Write Volatile Configuration Register instruction sequence QIO-SPI



9.3.25 Read Volatile Enhanced Configuration Register

The Read Volatile Enhanced Configuration Register (RDVECR) instruction allows the Volatile Configuration Register to be read.

Figure 125. Read Volatile Enhanced Configuration Register instruction sequence QIO-SPI

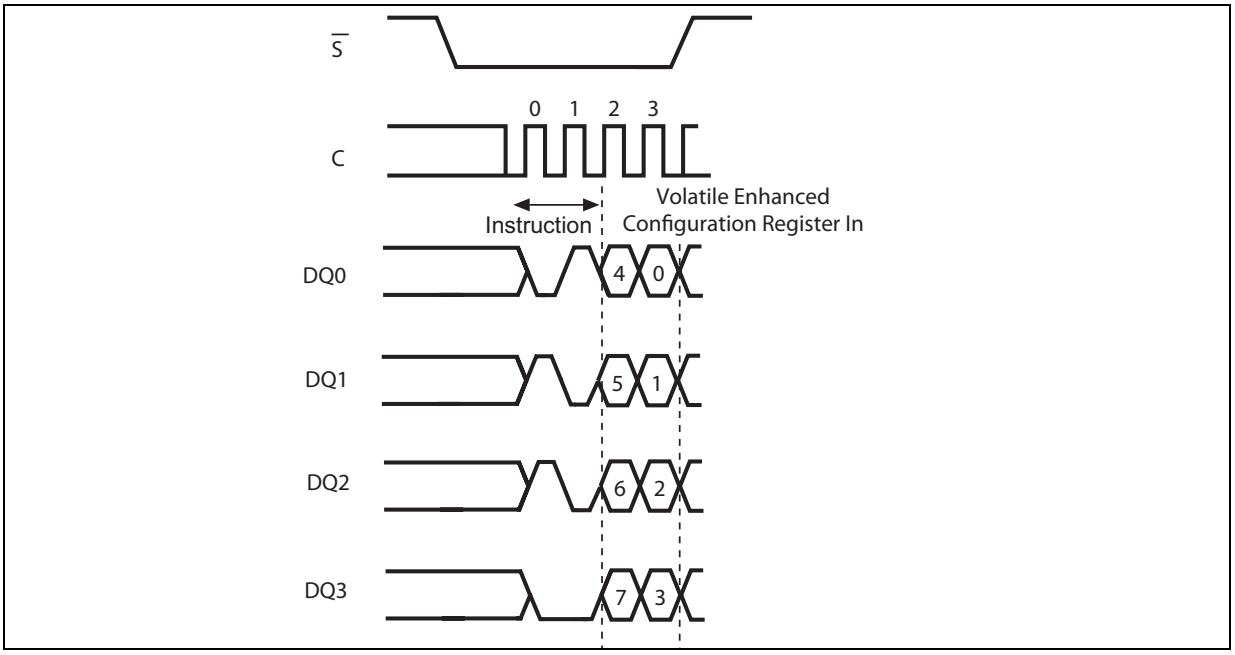


9.3.26 Write Volatile Enhanced Configuration Register

The Write Volatile Enhanced Configuration register (WRVECR) instruction allows new values to be written to the Volatile Enhanced Configuration register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed.

Apart form the parallelizing of the instruction code and the input data on the four pins DQ0, DQ1, DQ2 and DQ3, the instruction functionality is exactly the same as the Write Volatile Enhanced Configuration Register (WRVECR) instruction of the Extended SPI protocol, please refer to *Section 9.1.40: Write Volatile Enhanced Configuration Register* for further details.

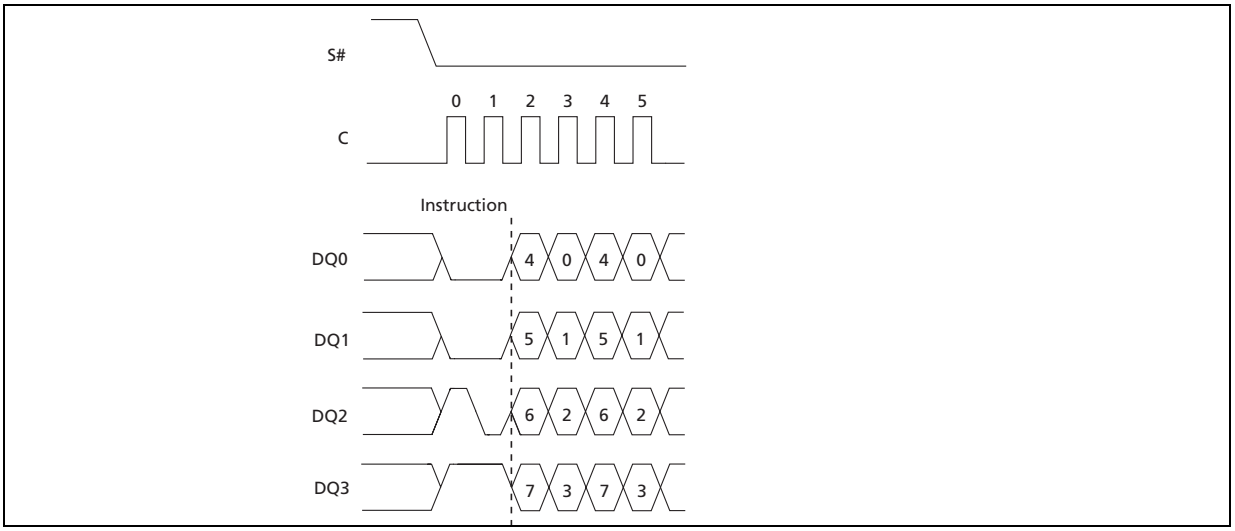
Figure 126. Write Volatile Enhanced Configuration Register instruction sequence QIO-SPI



9.3.27 Read Extended Address Register, Quad I/O

The Read Extended Address Register instruction allows the volatile configuration register to be read.

Figure 127. Read Extended Address Register Instruction Sequence, Quad I/O

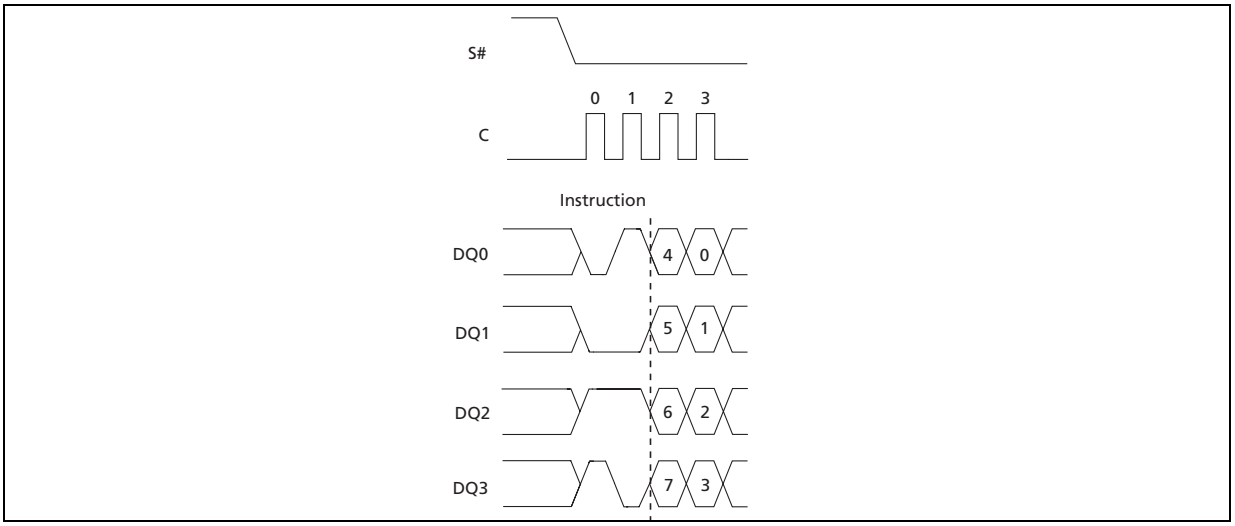


9.3.28 Write Extended Address Register, Quad I/O

The Write Extended Address Register, dual I/O instruction allows new values to be written to the extended address register. Before this instruction can be accepted, a Write Enable instruction must have been executed previously.

This instruction functions exactly as the Write Extended Address Register instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on four pins, DQ0, DQ1, DQ2 and DQ3.

Figure 128. Write Extended Address Register Instruction Sequence, Quad I/O

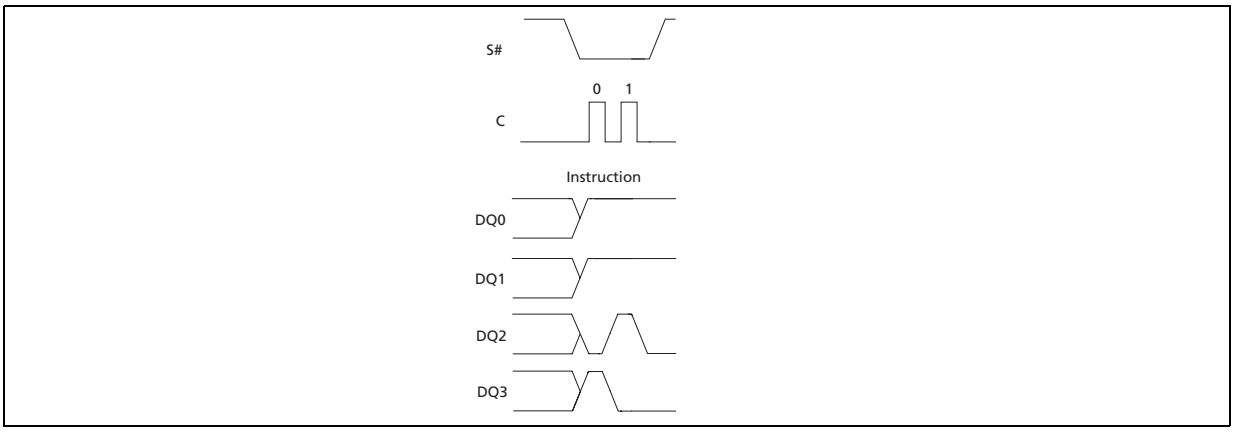


9.3.29 Enter 4-Byte Address Mode, Quad I/O

The Enter 4-Byte Address Mode instruction enables 4-byte address mode. Before this command can be accepted, a Write Enable instruction must have been executed previously. After the Write Enable instruction has been decoded and executed, the device sets the write enable latch (WEL) bit.

This instruction functions exactly as the Enter 4-Byte Address Mode instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on four pins, DQ0, DQ1, DQ2 and DQ3.

Figure 129. Enter 4-Byte Address Mode Instruction Sequence, Quad I/O

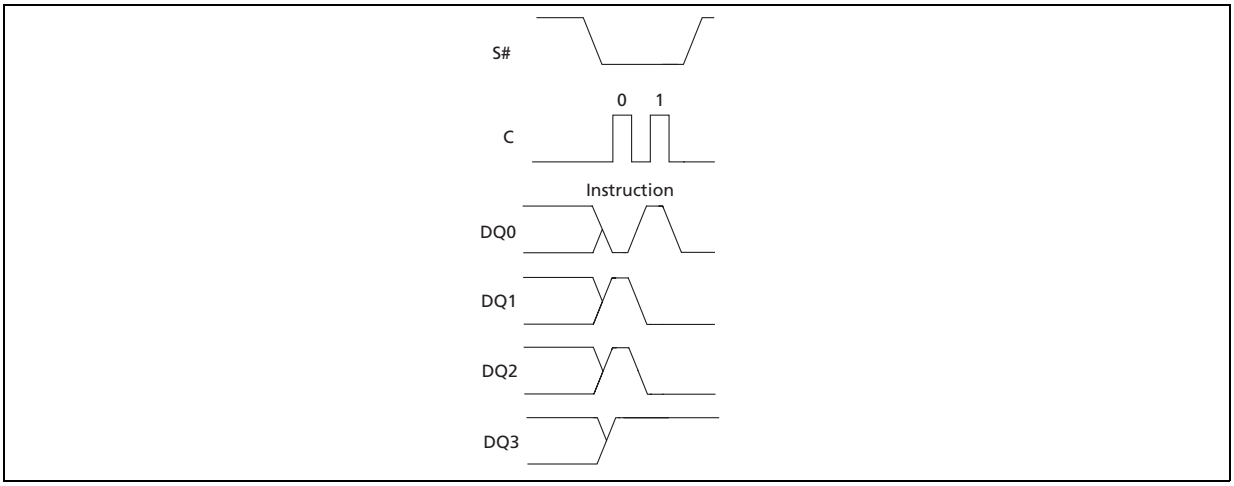


9.3.30 Exit 4-Byte Address Mode, Quad I/O

The Exit 4-Byte Address Mode instruction disables 4-byte address mode. Before this instruction can be accepted, a Write Enable instruction must have been executed previously. After the Write Enable instruction has been decoded and executed, the device sets the write enable latch (WEL) bit.

This instruction functions exactly as the Exit 4-Byte Address Mode instruction of the extended SPI protocol, except that for this instruction the instruction code and input data are on four pins, DQ0, DQ1, DQ2 and DQ3.

Figure 130. Exit 4-Byte Address Mode Instruction Sequence, Quad I/O



9.3.31 Reset Enable and Reset Memory, Quad I/O

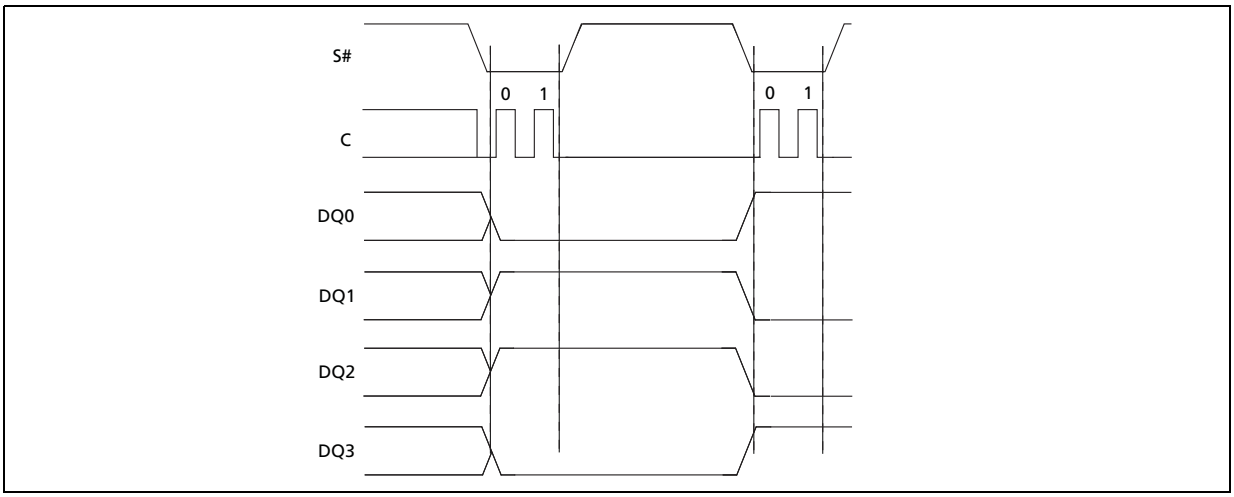
The Reset Enable and Reset Memory operation is used as a system software reset that puts the device in the power-on reset condition.

This operation consists of two instructions: Reset Enable and Reset Memory.

The Reset operation requires the Reset Enable instruction followed by the Reset Memory instruction. If the Reset Enable instruction is followed by any instruction other than Reset Memory, it is disabled. Reset Memory is also disabled if the device is selected by driving chip select (S) and Clock (C) low.

This instruction functions exactly as the Reset Enable instruction of the Extended SPI protocol, except that for this instruction the instruction code and input data are on four pins, DQ0, DQ1, DQ2 and DQ3.

Figure 131. Reset Enable and Reset Memory Instruction Sequence, Quad I/O



10 XIP Operations

XIP (eXecution in Place) mode is available in each protocol: Extended SPI, DIO-SPI, and QIO-SPI. XIP mode allows the memory to be read simply by sending an address to the device and then receiving the data on one, two, or four pins in parallel, depending on the customer requirements. It offers maximum flexibility to the application, saves instruction overhead, and allows a dramatic reduction to the Random Access time.

You can enable XIP mode in two ways:

- Using the Volatile Configuration Register: this is dedicated to applications that boot in SPI mode (Extended SPI, DIO-SPI or QIO-SPI) and then during the application life need to switch to XIP mode to directly execute some code in the flash.
- Using the Non Volatile Configuration Register: this is dedicated to applications that need to boot directly in XIP mode.

Setting to 0 the bit 3 of the Volatile Configuration Register the device is ready to enter in XIP mode right after the next fast read instruction (by 1, 2 or 4 pin).

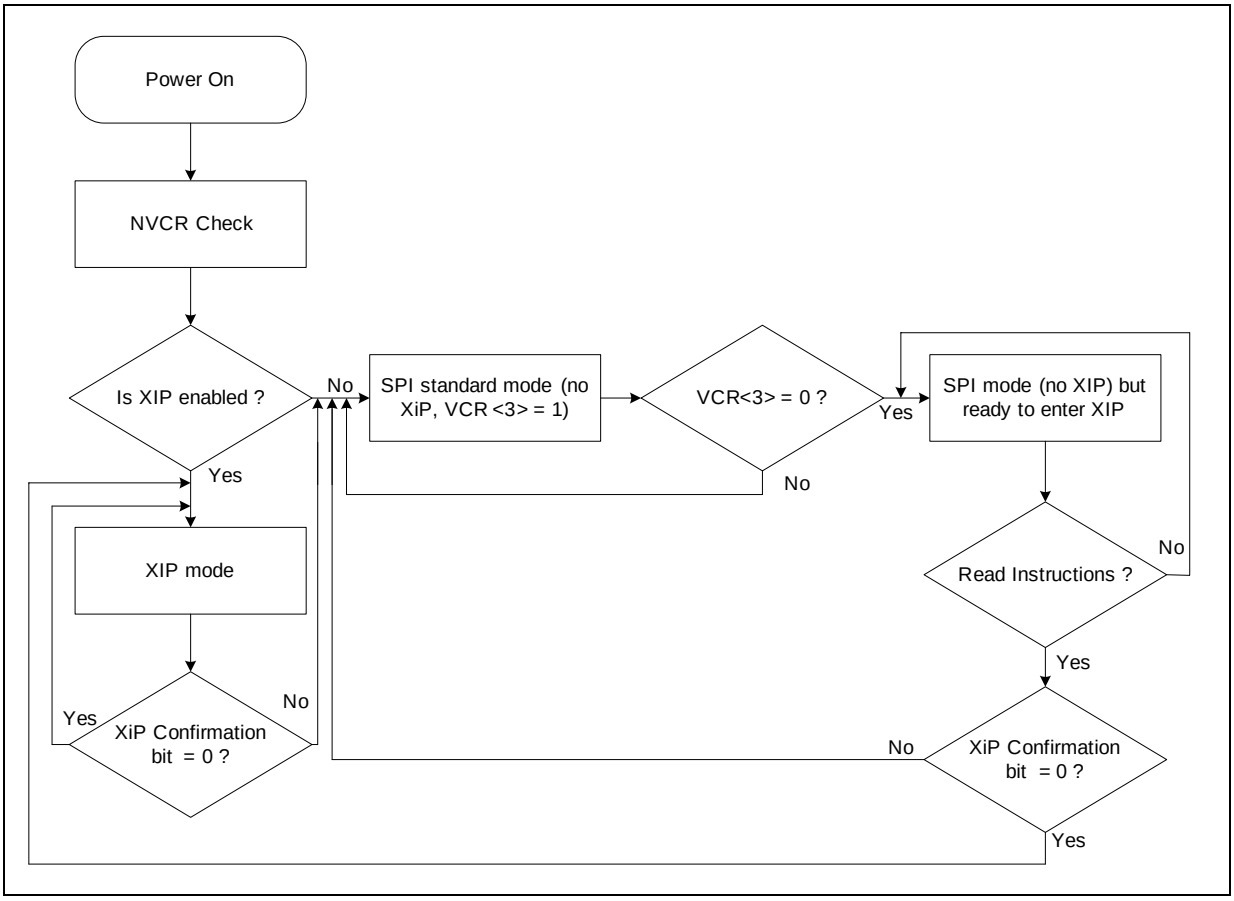
While acting on the Non Volatile Configuration Register (bit 11 to bit 9, depending on which XIP type is required, single, dual or quad I/O) the memory enters in the selected XIP mode only after the next power-on sequence. The Non Volatile Configuration Register XIP configuration bits allows the memory to start directly in the required XIP mode (Single, Dual or Quad) after the power on.

The XIP mode status must be confirmed forcing the XIP confirmation bit to "0", the XIP confirmation bit is the value on the DQ0 pin during the first dummy clock cycle after the address in XIP reading instruction. Forcing the bit "1" on DQ0 during the first dummy clock cycle after the address (XIP Confirmation bit) the memory returns in the previous standard read mode, that means it will codify as an instruction code the next byte received on the input pin(s) after the next chip select. Instead, if the XIP mode is confirmed (by forcing the XIP confirmation bit to 0), after the device next de-selection and selection cycle, the memory codify the first 3 bytes received on the inputs pin(s) as a new address.

Besides not confirming the XIP mode during the first dummy clock cycle, it is possible to exit the XIP mode by mean of a dedicated rescue sequence.

Note: For devices with a feature set digit equal to 2 or 4 in the part number (Basic XiP), it is not necessary to set the Volatile Configuration Register bit 3 to enter XIP mode: it is possible to enter XIP mode directly by setting XIP Confirmation bit to 0 during the first dummy clock cycle after a fast read instruction.See Section 16: Ordering information.

Figure 132. Read functionality Flow Chart



10.1 Enter XIP mode by setting the Non Volatile Configuration Register

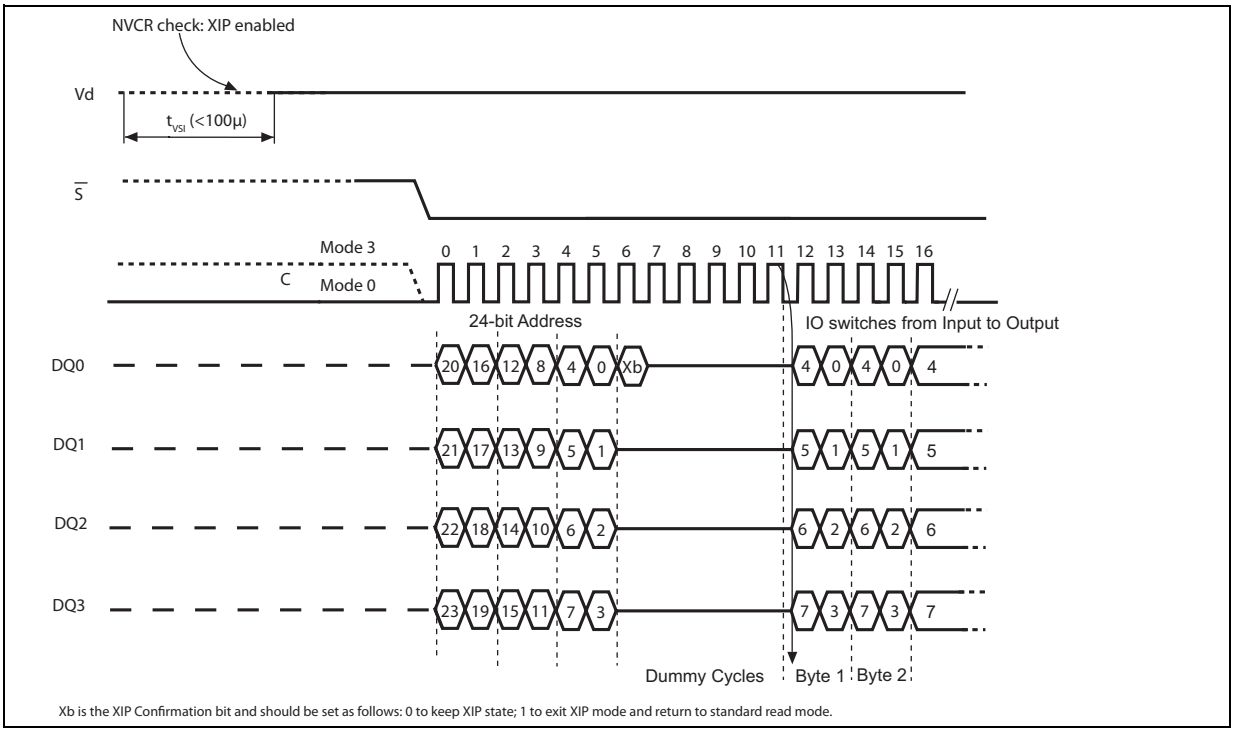
To use the Non Volatile Configuration Register method to enter in XIP mode it is necessary to set the Non Volatile Configuration Register bits from 11 to 9 with the pattern corresponding to the required XIP mode by mean of the Write Non Volatile Configuration Register (WRNVCR) instruction. (See *Table 26.: NVCR XIP bits setting example*.)

This instruction doesn't affect the XIP state until the next Power on sequence. In this case, after the next power on sequence, the memory directly accept addresses and then, after the dummy clock cycles (configurable), outputs the data as described in *Table 26.: NVCR XIP bits setting example*. For example to enable XIP on QIOFR in normal SPI protocol with six dummy clock cycles the following pattern must be issued:

Table 26. NVCR XIP bits setting example

B1h (WRNVCR opcode)	+ 0110	100	111	x	1	11	11
	6 dummy cycles for fast read instructions	XIP set as default; Quad I/O mode	Output Buffer driver strength default	Don't Care	Hold/Reset not disabled	Extended SPI protocol	Activate 4-byte Address (Default)

Figure 133. XIP mode directly after power on



Xb is the XIP Confirmation bit and should be set as follows: 0 to keep XIP state; 1 to exit XIP mode and return to standard read mode.

10.2 Enter XIP mode by setting the Volatile Configuration Register

To use the Volatile Configuration Register method to enter XIP mode, it is necessary to write a 0 to bit 3 of the Volatile Configuration Register to make the device ready to enter XIP mode (2). This instruction doesn't permit to enter XIP state directly: a Fast Read instruction (either Single, Dual or Quad) is needed once to start the XIP Reading.

After the Fast Read instruction (Single, Dual or Quad) the XIP confirmation bit must be set to 0. (first bit on DQ0 during the first dummy cycle after the address has been received),

Then after the next de-select and select cycle ($\overline{S}$ pin set to 1 and then to 0) the memory codify the first 3 bytes received on the input pin(s) directly as an address, without any instruction code, and after the dummy clock cycles (configurable) directly outputs the data.

For example to enable the XIP (without enter) with six dummy clock cycles, the pattern in *Table 27.: VCR XIP bits setting example* must be issued, and after that it is possible to enter, for example, in XIP mode from extended SPI read mode by mean of Quad Input Output Fast Read instruction, as described in *Table 27.: VCR XIP bits setting example*.

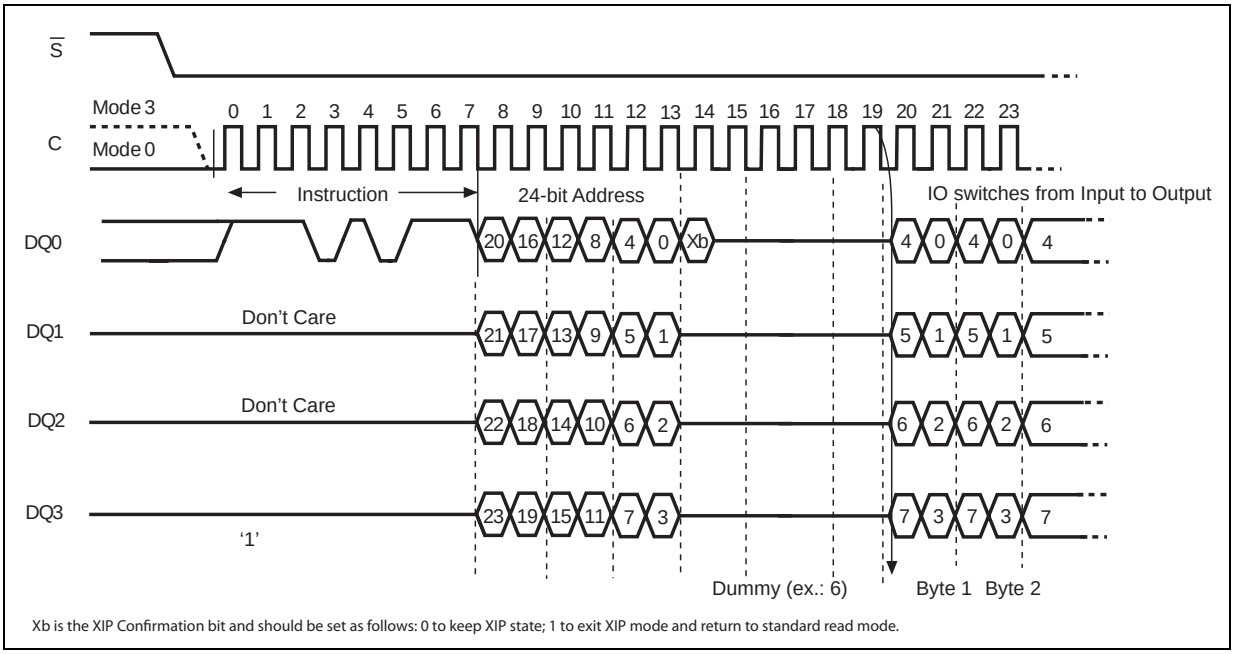
If 4-byte address mode is enabled, the device uses a 32-bit address as explained and shown in *Section 5.1.2: 4 Byte Address Mode on page 25*

Note: For devices with a feature set digit equal to 2 or 4 in the part number (Basic XiP), it is not necessary to set the Volatile Configuration Register bit 3 to enter in XIP mode: it is possible to enter directly in XIP mode by setting XIP Confirmation bit to 0 during the first dummy clock cycle after a fast read instruction. See Section 16: Ordering information.

Table 27. VCR XIP bits setting example

81h (WRVCR opcode)	+ 0110	0	0	11
	6 dummy cycles	Ready for XIP	Reserved	Wrap Mode Configuration

Figure 134. XiP: enter by VCR 2/2 (QIOFR in normal SPI protocol example)



10.3 XIP mode hold and exit

The XIP mode does require at least one additional clock cycle to allow the XIP Confirmation bit to be sent to the memory on DQ0 during the first dummy clock cycle.

The device decodes the XIP Confirmation bit with the scheme:

- XIP Confirmation bit=0 means to hold XIP Mode
- XIP Confirmation bit=1 means to exit XIP Mode and comes back to read mode, that means codifying the first byte after the next chip select as an instruction code.

In Dual I/O XIP mode, the values of DQ1 during the first dummy clock cycle after the addresses is always Don't Care.

In Quad I/O XIP mode, the values of DQ3, DQ2 and DQ1 during the first dummy clock cycle after the addresses are always Don't Care.

In Dual and Single I/O XIP mode, in presence of the $\overline{\text{RESET}}$ pin enabled (in devices with a dedicated part number), a low pulse on that pin resets the XIP protocol as defined by the Volatile Configuration Register, reporting the memory at the state of last power up, as defined by the Non Volatile Configuration Register. In Quad I/O XiP modes, it is possible to reset the memory (for devices with a dedicated part number) only when the device is deselected. See *Section 16: Ordering information*.

10.4 XIP Memory reset after a controller reset

If during the application life the system controller is reset during operation, and the device features the RESET functionality (in devices with a dedicated part number), and the feature has not been disabled, after the controller resets, the memory returns to POR state and there is no issue. See *Section 16: Ordering information*.

In all the other cases, it is possible to exit the memory from the XIP mode by sending the following rescue sequence at the first chip selection after a system reset:

DQ0 (PAD DATA) equal to '1' for:DQ0= '1' for:

7 clock cycles within S low (S becomes high before 8th clock cycle)

+ 9 clock cycles within S low (S becomes high before 10th clock cycle)

+ 13 clock cycles within S low (S becomes high before 14th clock cycle)

+ 17 clock cycles within S low (S becomes high before 18th clock cycle)

+ 25 clock cycles within S low (S becomes high before 26th clock cycle)

+ 33 clock cycles within S low (S becomes high before 34th clock cycle)

The global effect is only to exit from XIP without any other reset.

Power-on Reset state means that all the lock bits, volatile configuration registers, and the extended address register have been reset in the power-on reset default condition. The power-on reset condition depends on non volatile configuration register content.

11 Power-up and power-down

At power-up and power-down, the device must not be selected (that is Chip Select ($\overline{S}$) must follow the voltage applied on VCC) until VCC reaches the correct value:

- VCC(min) at power-up
- VSS at power-down

A safe configuration is provided in *Section 3: SPI Modes*.

To avoid data corruption and inadvertent write operations during power-up, a Power On Reset (POR) circuit is included. The logic inside the device is held reset while VCC is less than the Power On Reset (POR) threshold voltage, VWI - all operations are disabled, and the device does not respond to any instruction.

During a standard power-up phase the device ignores all the instructions but RDSR and RFSR (they can be used to check the memory internal state according to *Figure 135.: Power-up timing*.

After power-up, the device is in the following state:

- The device is in the Standby Power mode
- The Write Enable Latch (WEL) bit is reset
- The Write In Progress (WIP) bit is reset
- The Lock Registers are configured as: (Write Lock bit, Lock Down bit) = (0,0).

Normal precautions must be taken for supply line decoupling, to stabilize the VCC supply.

Each device in a system should have the VCC line decoupled by a suitable capacitor close to the package pins (generally, this capacitor is of the order of 100 nF).

At power-down, when VCC drops from the operating voltage, to below the Power On Reset (POR) threshold voltage, VWI, all operations are disabled and the device does not respond to any instruction (the designer needs to be aware that if power-down occurs while a Write, Program or Erase cycle is in progress, some data corruption may result).

VPPH must be applied only when VCC is stable and in the VCC(min) to VCC(max) voltage range.

Figure 135. Power-up timing

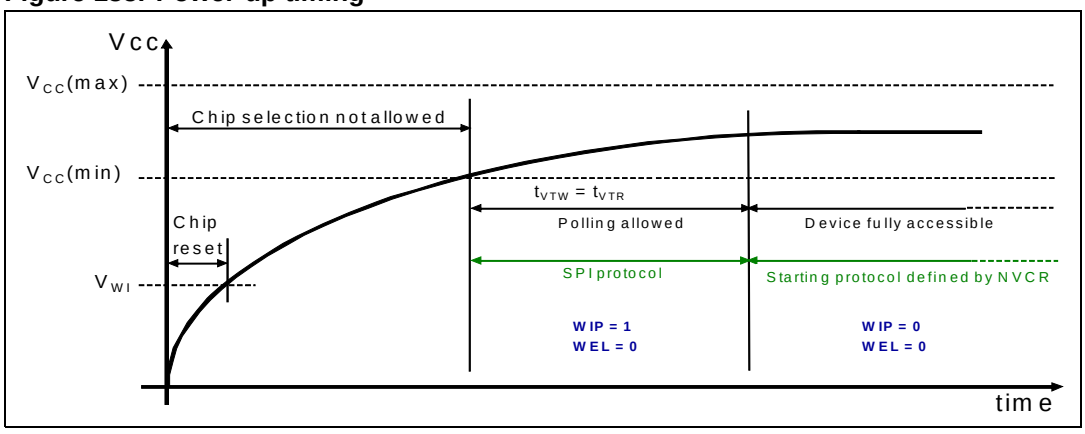


Table 28. Power-up timing and V_{WI} threshold

Symbol	Parameter	Min	Max	Unit
t_{VTR}(1)	V_{CC}(min) to Read		150	µs
t_{VTW}(1)	V_{CC}(min) to device fully accessible		150	µs
V_{WI}(1)	Write inhibit voltage	1.5	2.5	V

1. These parameters are characterized only.

11.1 Rescue sequence in case of power loss during WRNVCR

If a power loss occurs during a Write Non Volatile Configuration Register instruction, after the next power on the device could eventually wake up in a not determined state, for example a not required protocol or XIP mode. In that case a particular rescue sequence must be used to recover the device at a fixed state (Extended SPI protocol without XIP) until the next power up. Then to fix the problem definitively is recommended to run the Write Non Volatile configuration Register again.

The rescue sequence is composed of two parts that have to be run in the correct order. During all the sequence the TSHSL2 must be 50ns at least. The first part of the sequence is:

DQ0 (PAD DATA) and DQ3 (PAD HOLD) equal to '1' for:

7 clock cycles within S low (S becomes high before 8th clock cycle)

+ 9 clock cycles within S low (S becomes high before 10th clock cycle)

+ 13 clock cycles within S low (S becomes high before 14th clock cycle)

+ 17 clock cycles within S low (S becomes high before 18th clock cycle)

+ 25 clock cycles within S low (S becomes high before 26th clock cycle)

+ 33 clock cycles within S low (S becomes high before 34th clock cycle)

The second part of the sequence is exiting from DIO-SPI or QIO-SPI by using the following FFh sequence:

- DQ0 (PAD DATA) and DQ3 (PAD HOLD) equal to '1' for 8 clock cycles within S low; S becomes high before 9th clock cycle. After this sequence the Extended SPI protocol is active.

To exit XIP, see *10.4: XIP Memory reset after a controller reset on page 167*

12 Initial delivery state

The device is delivered with the memory array erased: all bits are set to 1 (each byte contains FFh). The Status Register contains 00h (all Status Register bits are 0).

13 Maximum rating

Stressing the device outside the ratings listed here may cause permanent damage to the device. These are stress ratings only, and operation of the device at these, or any other conditions outside those indicated in the operating sections of this specification, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 29. Absolute maximum ratings

Symbol	Parameter	Min	Max	Unit
T_{STG}	Storage temperature	–65	150	°C
T_{LEAD}	Lead temperature during soldering		see(1)	°C
V_{IO}	Input and output voltage (with respect to ground)	–0.6	V_{CC} + 0.6	V
V_{CC}	Supply voltage	–0.6	4.0	V
VPP	Fast program/erase voltage	–0.2	10.0	V
V_{ESD}	Electrostatic discharge voltage (human body model)(2)	–2000	2000	V

1. Compliant with JEDEC Std. J-STD-020C (for small body, Sn-Pb or Pb assembly), the Numonyx ECOPACK® 7191395 specification, and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU.
2. JEDEC Std JESD22-A114A (C1 = 100 pF, R1 = 1500 Ω, R2 = 500 Ω).

14 DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristics tables that follow are derived from tests performed under the measurement conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 30. Operating conditions

Symbol	Parameter	Min	Typ	Max	Unit
V_{CC}	Supply voltage	2.7		3.6	V
VPP_H	Supply voltage on VPP	8.5		9.5	V
T_A	Ambient operating temperature	–40		85	°C

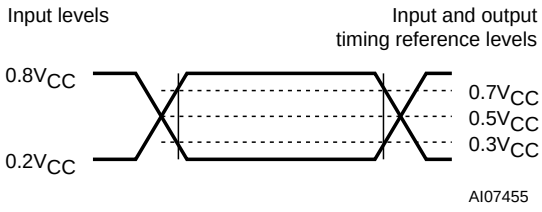
Table 31. AC measurement conditions

Symbol	Parameter	Min	Max	Unit
C_L	Load capacitance	30(1)		pF
	Input rise and fall times		5	ns
	Input pulse voltages	$0.2V_{CC}$ to $0.8V_{CC}$(2)		V
	Input timing reference voltages	$0.3V_{CC}$ to $0.7V_{CC}$		V
	Output timing reference voltages	V_{CC} / 2		V

1) Output Buffers are configurable by user.

2) For QUAD/DUAL operations: 0 to Vcc.

Figure 136. AC measurement I/O waveform



Note: For 0.8VCC: for QUAD/DUAL operations, this is V_{CC}; for 0.2VCC: for QUAD/DUAL operations, this is 0V

Table 32. Capacitance(1)

Symbol	Parameter	Test condition	Min	Max	Unit
$C_{IN/OUT}$	Input/output capacitance (DQ0/DQ1/DQ2/DQ3)	V_{OUT} = 0 V		8	pF
C_{IN}	Input capacitance (other pins)	V_{IN} = 0 V		6	pF

1. Sampled only, not 100% tested, at T_A=25 °C and a frequency of 54 MHz.

Table 33. DC Characteristics

Symbol	Parameter	Test condition (in addiction to those in *Table 30.: Operating conditions*)	Min	Max	Unit
ILI	Input leakage current			± 2	µA
ILO	Output leakage current			± 2	µA
ICC1	Standby current	S = VCC, VIN = VSS or VCC		100	µA
ICC3	Operating current (Fast Read Single I/O)	C = 0.1VCC / 0.9VCC at 108 MHz, DQ1 = open		15	mA
		C = 0.1VCC / 0.9VCC at 54 MHz, DQ1 = open		6	mA
	Operating current (Fast Read Dual I/O)	C = 0.1VCC / 0.9VCC at 108 MHz		18	mA
	Operating current (Fast Read Quad I/O)	C = 0.1VCC / 0.9VCC at 108 MHz		20	mA
ICC4	Operating current (Page Program Single, Dual and Quad I/O)	$\overline{S}$ = VCC		20	mA
ICC5	Operating current (WRSR)	$\overline{S}$= VCC		20	mA
ICC6	Operating current (SE)	$\overline{S}$ = VCC		20	mA
VIL	Input low voltage		– 0.5	0.3VCC	V
VIH	Input high voltage		0.7VCC	VCC+0.4	V
VOL	Output low voltage	IOL = 1.6 mA		0.4	V
VOH	Output high voltage	IOH = –100 uA	VCC–0.2		V

Note: The AC Characteristics data is preliminary.

Table 34. AC Characteristics (page 1 of 2)

Symbol	Alt.	Parameter	Min	Typ(2)	Max	Unit
fC	fC	Clock frequency for the all the instructions (Extended SPI, DIO-SPI and QIO-SPI protocol) but the READ instruction	D.C.		108	MHz
fR		Clock frequency for read instructions	D.C.		54	MHz
tCH(1)	tCLH	Clock High time	4			ns
tCL(2)	tCLL	Clock Low time	4			ns
tCLCH(3)		Clock rise time(4) (peak to peak)	0.1			V/ns
tCHCL(3)		Clock fall time(4) (peak to peak)	0.1			V/ns
tSLCH	tCSS	$\overline{S}$ active setup time (relative to C)	4			ns
tCHSL			4			ns
tDVCH	tDSU	Data in setup time	2			ns
tCHDX	tDH	Data in hold time	3			ns
tCHSH		$\overline{S}$ active hold time (relative to C)	4			ns
tSHCH		$\overline{S}$ not active setup time (relative to C)	4			ns
tSHSL1	tCSH	$\overline{S}$ deselect time after a correct read instruction	20			ns
tSHSL2	tCSH	$\overline{S}$ deselect time after a not correct read or after any different instruction	50			ns
tSHQZ(3)	tDIS	Output disable time			8	ns
tCLQV	tV	Clock Low to Output valid under 30 pF			7	ns
		Clock Low to Output valid under 10 pF			5	ns
tCLQX	tHO	Output hold time	1			ns
tHLCH		HOLD setup time (relative to C)	4			ns
tCHHH		HOLD hold time (relative to C)	4			ns
tHHCH		HOLD setup time (relative to C)	4			ns
tCHHL		HOLD hold time (relative to C)	4			ns
tHHQX(3)	tLZ	HOLD to Output Low-Z			8	ns
tHLQZ(3)	tHZ	HOLD to Output High-Z			8	ns
tWHSL(5)		Write protect setup time	20			ns
tSHWL(5)		Write protect hold time	100			ns

Table 34. AC Characteristics (page 2 of 2)

Symbol	Alt.	Parameter	Min	Typ(2)	Max	Unit
tVPPHSL(6)		Enhanced program supply voltage High (VPPH) to Chip Select Low for Single and Dual I/O Page Program	200			ns
tW		Write status register cycle time		1.3	8	ms
tCFSR		Clear flag status register cycle time		40		ns
tWNVCR		Write non volatile configuration register cycle time		0.2	3	s
tWVCR		Write volatile configuration register cycle time		40		ns
tWRVECR		Write volatile enhanced configurationregister cycle time		40		ns
tWREAR		Write extended address register cycle time		40		ns
tPP(7)		Page Program Cycle Time (256 Bytes)		0.5	5	ms
		Page Program Cycle Time (n Bytes)		int(n/8) × 0.015(8)	5	ms
		Page Program Cycle Time Vpp=VPPH (256 Bytes)		0.4	5	ms
		Program OTP cycle time (64 bytes)		0.2		ms
tSSE		Subsector erase cycle time		0.3	3	s
tSE		Sector erase cycle time		0.7	3	s
		Sector erase cycle time VPP = VPPH		0.6	3	s
tBE		Bulk erase cycle time		240	480	s
		Bulk erase cycle time (with VPP=VPPH)		200	480	s

1. tCH + tCL must be greater than or equal to 1/ fC.
2. Typical values given for TA = 25 °C
3. Value guaranteed by characterization, not 100% tested in production.
4. Expressed as a slew-rate.
5. Only applicable as a constraint for a WRSR instruction when SRWD is set to '1'.
6. VPPH should be kept at a valid level until the program or erase operation has completed and its result (success or failure) is known.
7. When using the page program (PP) instruction to program consecutive bytes, optimized timings are obtained with one sequence including all the bytes versus several sequences of only a few bytes ($1 \leq n \leq 256$).
8. int(A) corresponds to the upper integer part of A. For example int(12/8) = 2, int(32/8) = 4 int(15.3) =16.

Figure 137. Reset AC waveforms while a program or erase cycle is in progress

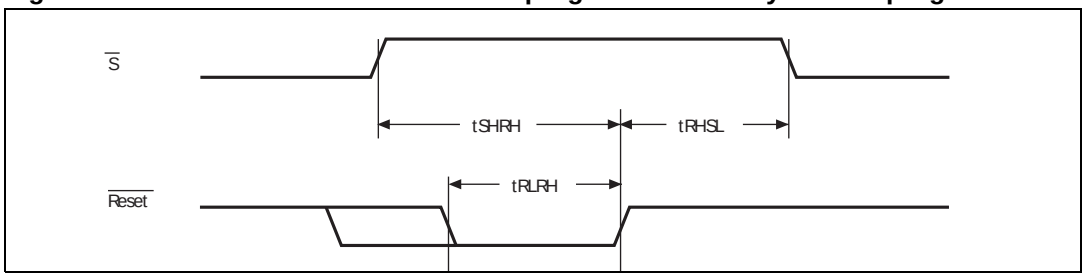


Figure 138. Reset Enable

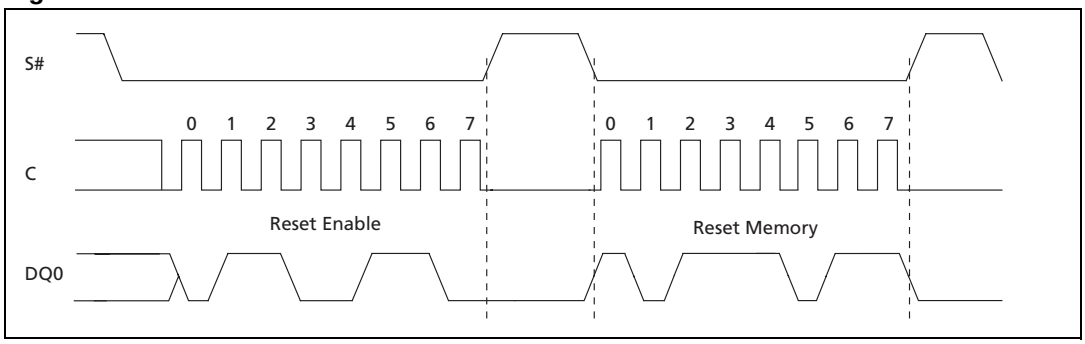


Table 35. Reset Conditions (page 1 of 2)

Symbol	Alt.	Parameter	Conditions	Min	Typ	Max	Unit
tRLRH(1)(2)	tRST	Reset pulse width				50	ns
tRHSL(1)	tREC	Reset Recovery Time	Device selected ($\overline{S}$ low), while decoding any modify instruction, during all read operations, CLFSR, WRDI, WREN, WRLR, WRVCR, WRVECR.			40	ns
			Under completion (or suspension) of an internal erase or program cycle related to POTP, PP, DIEFP, DIFP, QIEFP, QIFP, SE, BE, PER, PES.			30	µs
			Under completion or suspension of an SSE operation		t_{SSE}		ms
			Under completion of an WRSR operation.		t_W		ms
			Under completion of an WRNVCR operation.		t_{WNVCR}		ms
			Device deselected ($\overline{S}$ high) and in XiP mode.			40	ns
			Device deselected ($\overline{S}$ high) and in Standby mode.			40	ns

Table 35. Reset Conditions (page 2 of 2)

Symbol	Alt.	Parameter	Conditions	Min	Typ	Max	Unit
tSHSL3	tREC1	Software Reset Recovery Time	Under completion of an internal erase or program cycle related to POTP, PP, DIEFP, DIFP, QIEFP, QIFP, SE, BE, PER, PES.			30	µs
			Under completion or suspension of an SSE operation.		t_{SSE}		
			Under completion of an WRSR operation.		t_W		
			Under completion of an WRNVCR operation.		t_{WNVCR}		
			Device deselected (S high) and in Standby mode.			90	ns
tSHRV(1)		S# deselect to R valid	Deselect to R valid in Quad Output or in QIO-SPI.			2	ns

1. All values are guaranteed by characterization and not 100% tested in production.
2. The device reset is possible but not guaranteed if tRLRH < 50 ns.

Figure 139. Serial input timing

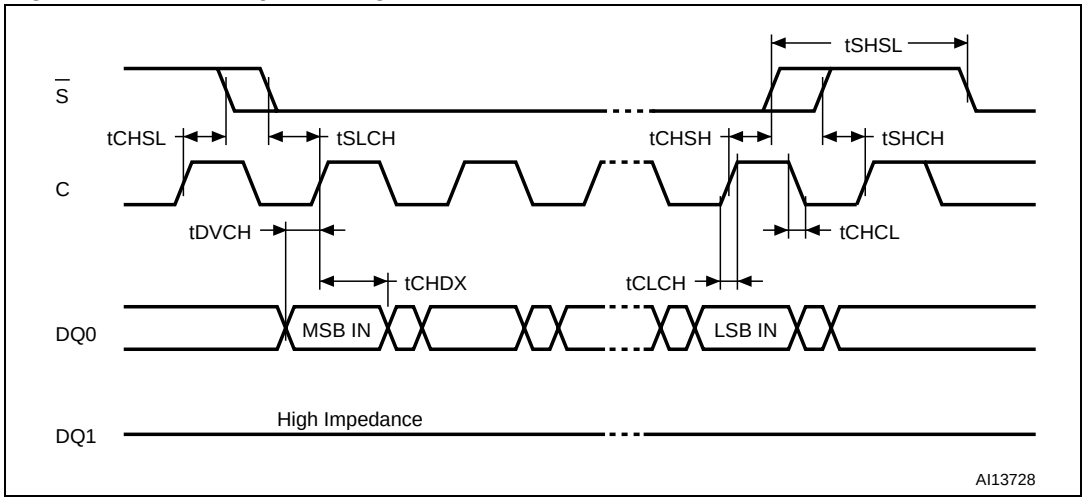


Figure 140. Write protect setup and hold timing during WRSR when SRWD=1

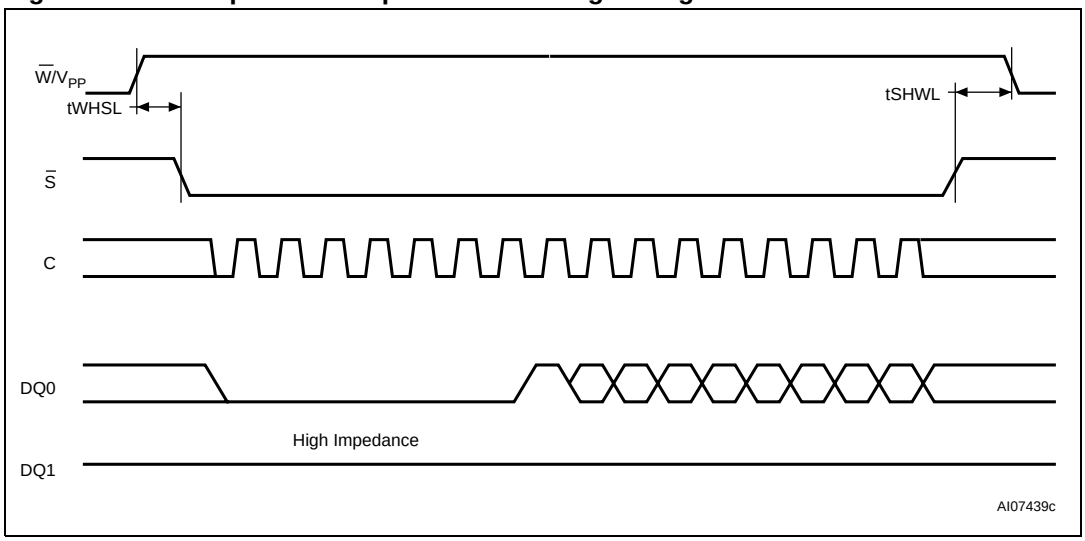


Figure 141. Hold timing

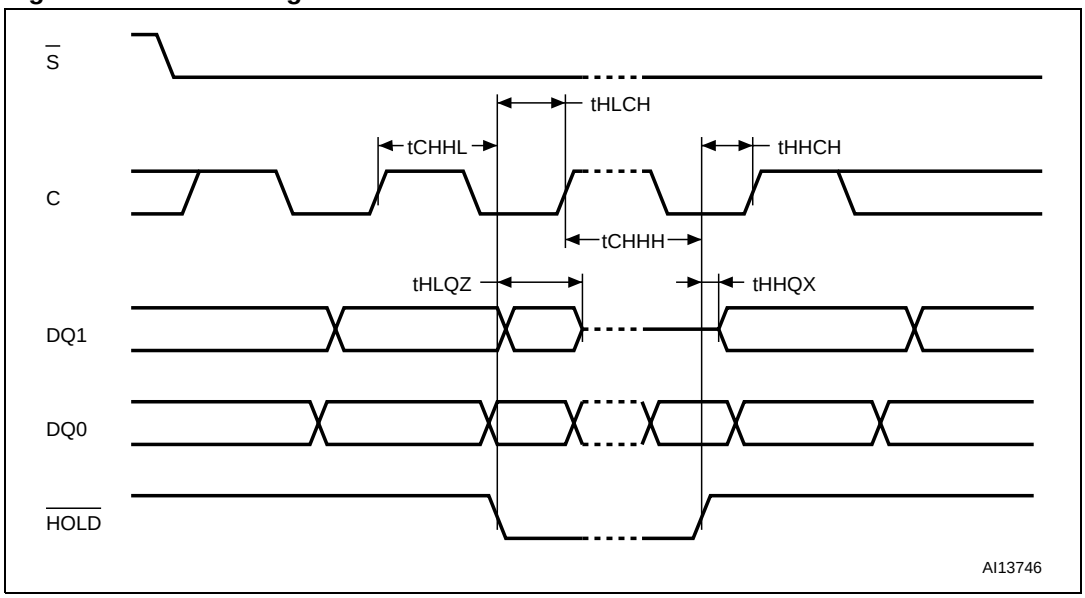


Figure 142. Output timing

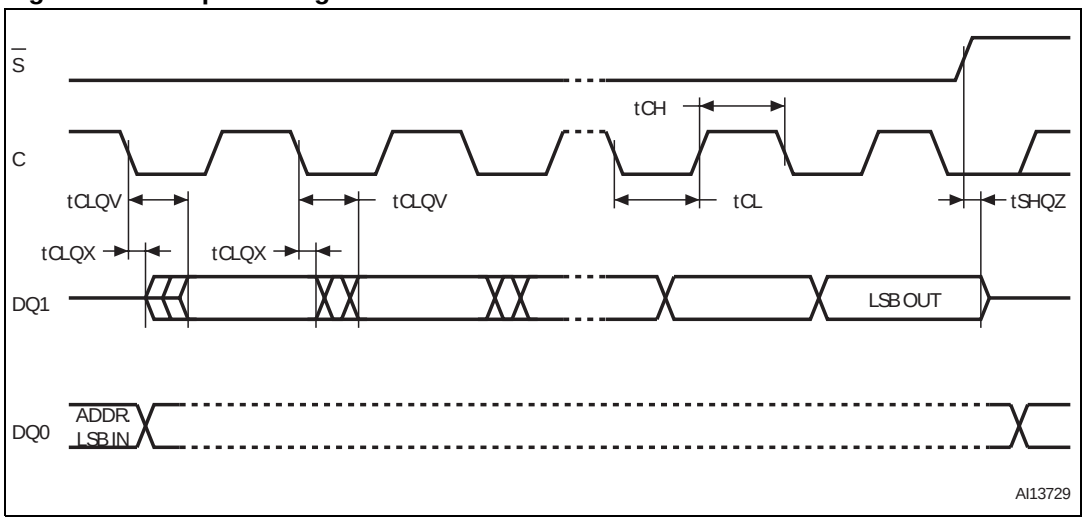
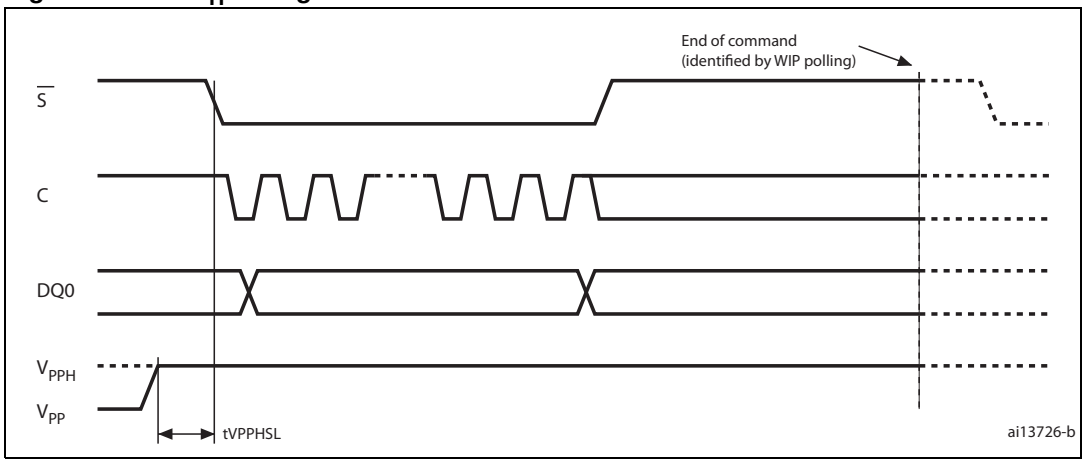


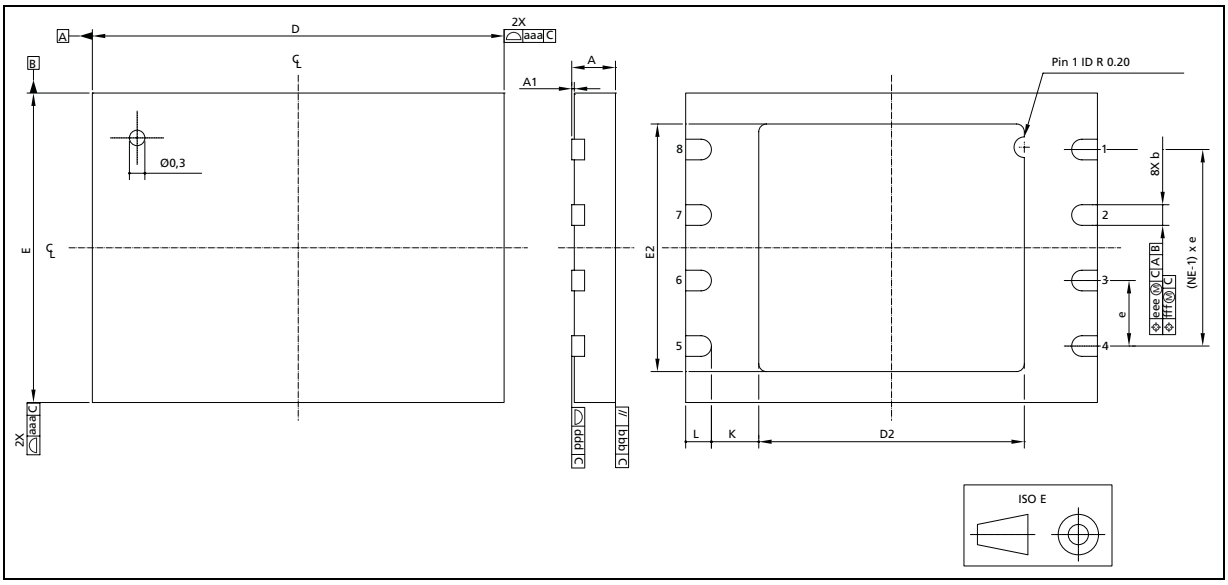
Figure 143. VPP_H timing



15 Package mechanical

In order to meet environmental requirements, Numonyx offers these devices in RoHS compliant packages. These packages have a lead-free second level interconnect. The category of second level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

Figure 144. VDFPN8 (MLP8) Very Thin Dual Flat Package 8 leads, 8×6x1 mm Drawing

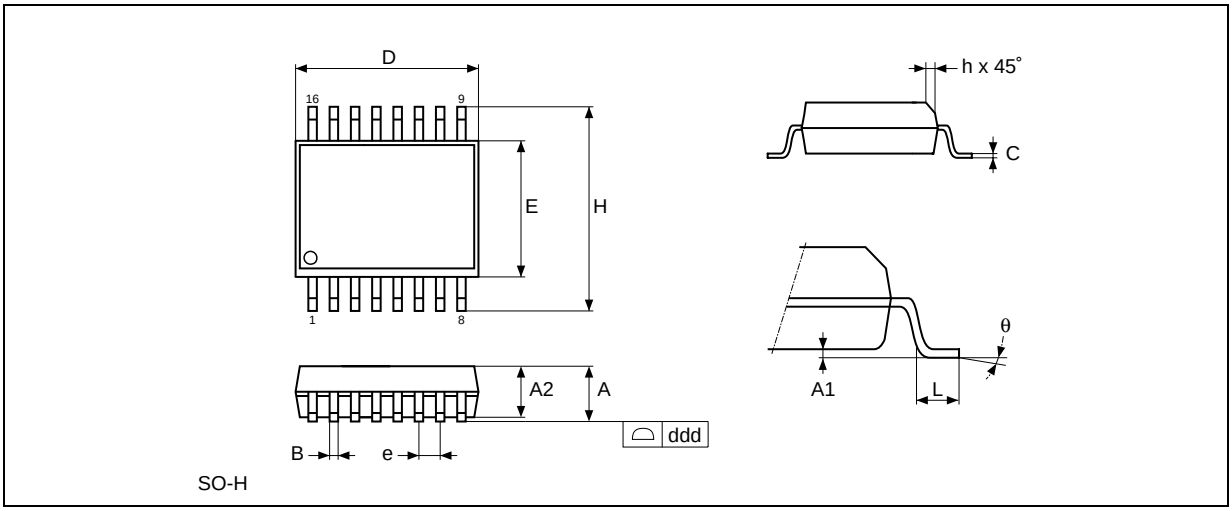


1. Drawing is not to scale.
2. The circle in the top view of the package indicates the position of pin 1.

Table 36. VDFPN8 (MLP8) Very Thin Dual Flat Package 8 leads 8×6x1 mm Dimensions

mm	Symbol									
	A	A1	b	D	D2	E	E2	e	K	L
Typ	0.85	—	0.4	8	5.16	6	4.8	1.27 (basic)	—	0.40
Min	—	0	0.35	—	—	—	—	1.27 (basic)	0.2	0.35
Max	1	0.05	0.48	—	—	—	—	1.27 (basic)	—	0.45

Figure 145. SO16 wide - 16-lead plastic small outline, 300 mils body width, Drawing

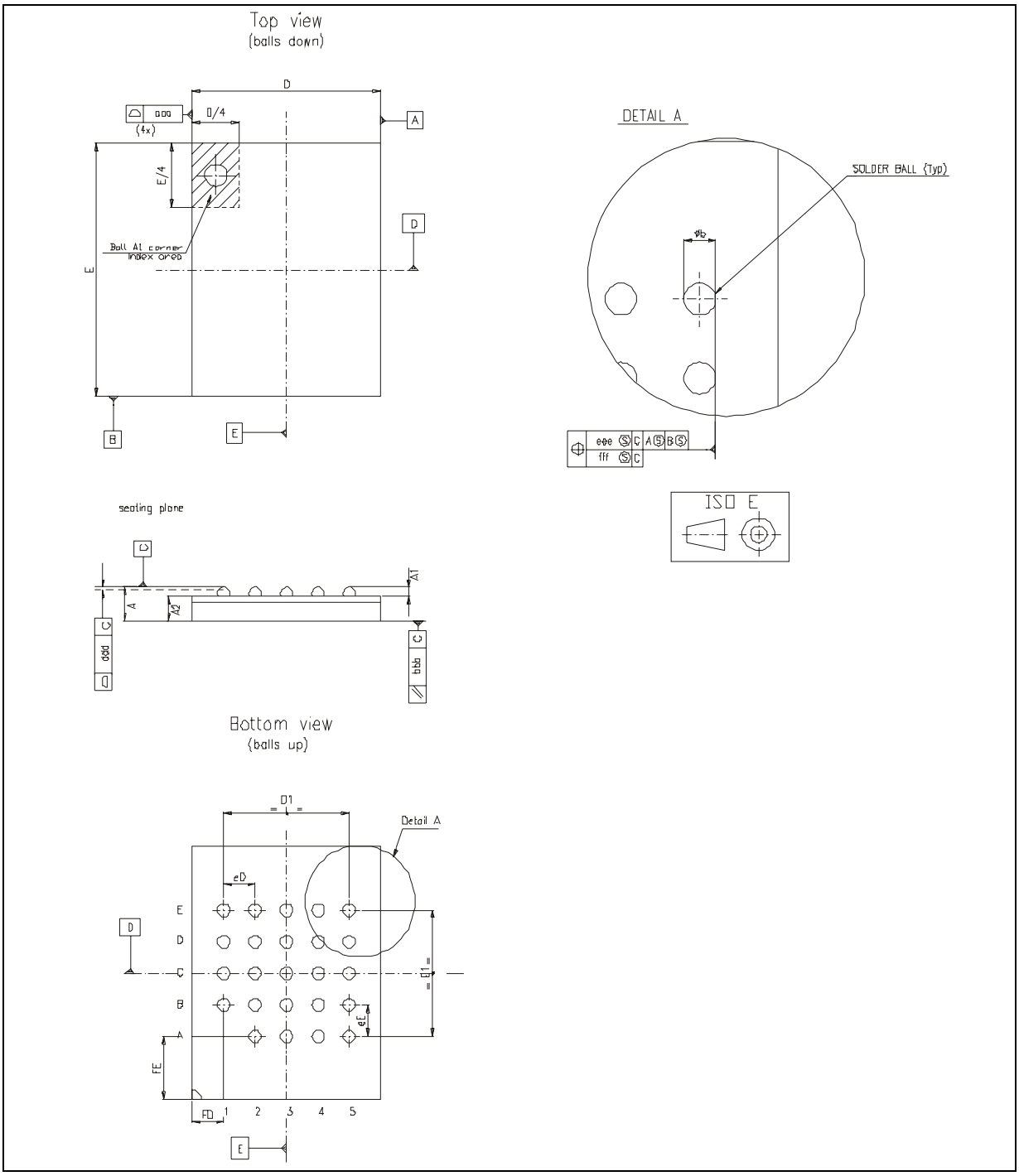


1. Drawing is not to scale.

Table 37. SO16 Wide – 16 Lead Plastic Small Outline, 300 mils Body Width, Dimensions

mm	Symbol											
	A	A1	B	C	D	E	e	H	h	L	θ	ddd
Typ	—	—	—	—	—	—	1.27	—	—	—	—	—
Min	2.35	0.10	0.33	0.23	10.10	7.40	—	10	0.25	0.40	0°	—
Max	2.65	0.30	0.51	0.32	10.50	7.60	—	10.65	0.75	1.27	8°	0.10

Figure 146. TBGA - 6 x 8 mm, 24-ball, mechanical package outline (Feasability Evaluation)



1. Drawing is not to scale.

Table 38. TBGA 6x8 mm 24-Ball, Dimensions, Symbols A to eE

mm	Symbol									
	A	A1	A2	θb	D	D1	E	E1	eD	eE
Typ	—	0.20	—	0.35	5.90	—	7.90	—	—	—
Min	—	—	0.79	0.40	6.00	4	8	4	1	1
Max	1.20	—	—	0.45	6.10	—	8.10	—	—	—

Table 39. TBGA 6x8 mm 24-Ball, Dimensions, Symbols FD to fff

mm	Symbol									
	FD	FE	MD	ME	n	aaa	bbb	ddd	eee	fff
Typ	—	—	5	5	24 balls	—	—	—	—	—
Min	1	2	5	5	24 balls	—	—	—	—	—
Max	—	—	5	5	24 balls	0.15	0.10	0.10	0.15	0.08

16 Ordering information

Note: For further information on line items not listed here or on any aspect of this device, please contact your nearest Numonyx Sales Office.

Table 40. Ordering information scheme

Example: N25Q256 A 1 3 E F8 4 0 E

Device type

N25Q = serial Flash memory, Quad I/O, XiP

Device density

256 = 256 Mbit

Technology

A = 65 nm

Feature set

1 = Byte addressability, Hold pin, Numonyx XiP (1)
2 = Byte addressability, Hold pin, Basic XiP (1)
3 = Byte addressability, Reset pin, Numonyx XiP (1)
4 = Byte addressability, Reset pin, Basic XiP (1)
7 = Byte addressability, Hold pin, Numonyx XiP (2)

Operating voltage

3 = VCC = 2.7 V to 3.6 V

Block Structure

E = Uniform (no boot sectors)

Package (all packages are RoHS compliant)

F8 = VDFPN8 8 x 6 mm (MLP8)
SF = SO16W (SO16 Wide 300 mils body width)
12 = TBGA24 6 x 8 mm

Temperature and test flow

4 = Industrial temperature range, –40 to 85 °C: Device tested with standard test flow
A = Automotive temperature range, –40 to 125 °C; Device tested with high reliability certified test flow
H = Industrial temperature range, –40 to 85 °C; Device tested with high reliability certified test flow

Security features (3)

0 = Default

Packing options

E = Tray packing
F = Tape and reel packing
G = Tube packing

1. Enter 4-Byte Addrress Mode and Exit 4-Byte Address Mode supported
2. Only Enter 4-Byte Addrress Mode supported
3. Additional secure options are available upon customer request.

Note: For further information on line items not listed here or on any aspect of this device, please contact your nearest Numonyx Sales Office.

Table 41. Valid Order Information Line Items

Part Number	Features	Block Structure	Package	Temperature and Test Flow	Security	Notes
N25Q256A13E1240E N25Q256A13E1240F	Byte addressability, Hold pin, Numonyx XiP	Uniform	TBGA24 6x8 mm	Industrial temp; Standard test flow	Default	
N25Q256A13EF840E N25Q256A13EF840F	Byte addressability, Hold pin, Numonyx XiP	Uniform	VDFPN8 8 x 6 mm	Industrial temp; Standard test flow	Default	
N25Q256A13ESF40F N25Q256A13ESF40G	Byte addressability, Hold pin, Numonyx XiP	Uniform	SO16	Industrial temp; Standard test flow	Default	

Note: 1 Applies to the entire table: Packing information details: E= tray, F= tape-n-reel, G= tube (16th digit of part number).

17 Revision history

Table 42. Document revision history

Date	Revision	Changes
29-June-2010	1	Initial release.
6-August-2010	2	Added information to clarify 4-Byte Address Mode; added reset information including *Figure 138.: Reset Enable on page 175* and new rows to *Table 35.: Reset Conditions on page 175*.
24-November-2010	3	Added the following sections: – *9.1.45: Reset Enable* – *9.2.27: Read Extended Address Register, Dual I/O* through – *9.2.31: Reset Enable and Reset Memory, Dual I/O* – *9.3.27: Read Extended Address Register, Quad I/O* through – *9.3.31: Reset Enable and Reset Memory, Quad I/O*
30-November-2010	4	Added *Figure 2.: MLP8 Connections*.
22-Feb-2011	5	Added a power-on reset clarification to cover page and to the section entitled, “XIP Memory reset after a controller reset.”

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
www.micron.com/productsupport Customer Comment Line: 800-932-4992

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein.
Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.