# 2 Mb EXCELON™ Ultra Ferroelectric RAM (F-RAM)

## Serial (quad SPI), 256K × 8, 108 MHz, industrial

## Features

- 2-Mbit ferroelectric random access memory (F-RAM) logically organized as 256K × 8
  - Virtually unlimited endurance of 100 trillion ($10^{14}$) read/write cycles
  - 151-year data retention (See **“Data retention and endurance”** on page 96)
  - Infineon instant non-volatile write technology
  - Advanced high-reliability ferroelectric process
- Single and multi I/O SPI
  - Serial bus interface SPI protocols
  - Supports SPI mode 0 (0, 0) and mode 3 (1, 1) for all SDR mode transfers
  - Supports SPI mode 0 (0, 0) for all DDR mode transfers
  - Extended I/O SPI protocols
  - Dual SPI (DPI) protocols
  - Quad SPI (QPI) protocols
- SPI clock frequency
  - Up to 108-MHz frequency SPI SDR
  - Up to 54-MHz frequency SPI DDR
- Execute-in-place (XIP) for memory read/write
- Write protection, data security, and data integrity
- Hardware protection using the write protect ($\overline{WP}$) pin
- Software block protection
- Embedded ECC and CRC for enhanced data integrity
  - ECC detects and corrects 1-bit error. If a 2-bit error occurs, it does not correct but reports through the ECC Status register
  - CRC detects any accidental change to raw data
- Extended electronic signatures
  - Device ID includes manufacturer ID and product ID
  - Unique ID
  - User programmable serial number
- Dedicated 256-byte special sector F-RAM
  - Dedicated special sector write and read
  - Content can survive up to three standard reflow cycles
- Low-power consumption at high speed
  - 10 mA (typ) active current for 108 MHz SPI SDR
  - 16 mA (typ) active current for 108 MHz QSPI SDR and 54-MHz QSPI DDR
  - 110 µA (typ) standby current
  - 0.80 µA (typ) deep power down mode current
  - 0.1 µA (typ) hibernate mode current
- Low-voltage operation:
  - CY15V102QSN: $V_{DD}$ = 1.71 V to 1.89 V
  - CY15B102QSN: $V_{DD}$ = 1.8 V to 3.6 V

- Operating temperature: –40°C to +85°C
- 8-pin small outline integrated circuit (SOIC) package
- Restriction of hazardous substances (RoHS) compliant

## Functional description

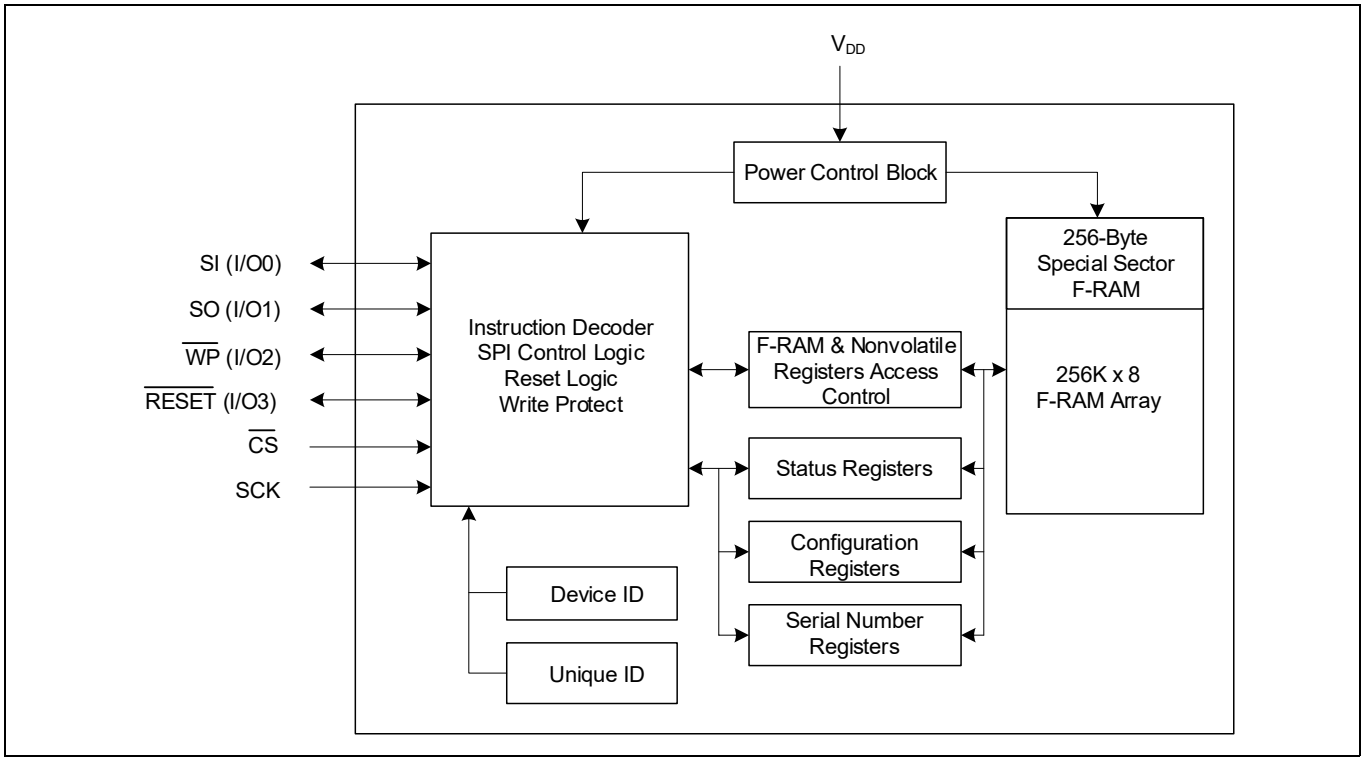
The EXCELON™ Ultra CY15X102QSN is a high-performance, 2-Mbit nonvolatile memory employing an advanced ferroelectric process. A ferroelectric random access memory or F-RAM is nonvolatile and performs reads and writes similar to a RAM. It provides reliable data retention for 151 years while eliminating the complexities, overhead, and system-level reliability problems caused by serial flash and other nonvolatile memories.

Unlike serial flash, the CY15X102QSN performs write operations at bus speed. No write delays are incurred. Data is written to the memory array immediately after each byte is successfully transferred to the device. The next bus cycle can commence without the need for data polling. In addition, the product offers substantial write endurance compared to other nonvolatile memories. The CY15X102QSN is capable of supporting $10^{14}$ read/write cycles, or 100 million times more write cycles than EEPROM. These capabilities make the CY15X102QSN ideal for nonvolatile memory applications, requiring frequent or rapid writes. Examples range from data collection, where the number of write cycles may be critical, to demanding industrial controls where the long write time of serial flash can cause data loss.

The CY15X102QSN combines a 2-Mbit F-RAM with the high-speed quad SPI (QPI) SDR and DDR interfaces which enhances the nonvolatile write capability of F-RAM technology. The device incorporates a read-only device ID and unique ID features which allow the SPI bus master to determine the manufacturer, product density, product revision and unique ID for each part. The device is also offered with a unique serial number that is read-only and can be used to identify a board or a system.

The device supports on-die ECC logic which can detect and correct 1-bit error in every 8-byte unit data. The device also extends capability to report 2-bit error in 8-byte unit data. The CY15X102QSN also supports the cyclic redundancy check (CRC) feature which can be used to check the data integrity of the stored data in the memory array.

## Logic block diagram



# Table of contents

**Features** ..... **1**
**Functional description** ..... **2**
**Logic block diagram** ..... **2**
**Table of contents** ..... **3**
**1 Pinouts** ..... **5**
**2 Pin definitions** ..... **6**
**3 Functional overview** ..... **7**
3.1 Memory architecture ..... 7
3.2 SPI bus ..... 7
3.2.1 Single channel SPI ..... 8
3.2.2 Extended SPI ..... 8
3.2.3 Dual SPI (DPI) ..... 8
3.2.4 Quad SPI (QPI) ..... 9
3.3 Terms used in SPI protocol ..... 9
3.3.1 SPI master ..... 9
3.3.2 SPI slave ..... 9
3.3.3 Chip select ($\overline{CS}$) ..... 9
3.3.4 Serial clock (SCK) ..... 9
3.3.5 Data transmission (SI/SO) ..... 10
3.3.6 Most significant bit (MSb) ..... 10
3.3.7 Serial opcode ..... 10
3.3.8 Invalid opcode ..... 11
3.3.9 Instruction ..... 11
3.3.10 Mode byte ..... 11
3.3.11 Wait states or dummy cycles ..... 11
3.4 SPI modes ..... 12
3.4.1 SDR ..... 12
3.4.2 DDR ..... 12
3.5 Power-up to first access ..... 13
**4 CY15x102QSN registers** ..... **15**
4.1 Status registers ..... 15
4.1.1 Status register 1 (SR1) ..... 15
4.1.2 Status register 2 (SR2) ..... 18
4.2 Configuration registers ..... 19
4.2.1 Configuration register 1 (CR1) ..... 19
4.2.2 Configuration register 2 (CR2) ..... 22
4.2.3 Configuration register 4 (CR4) ..... 25
4.2.4 Configuration register 5 (CR5) ..... 27
**5 Functional description** ..... **29**
5.1 Command structure ..... 29
5.1.1 Write enable control commands ..... 31
5.1.2 Register access commands ..... 34
5.1.3 Memory operation ..... 45
5.1.4 Memory write operation commands ..... 45
5.1.5 Memory read operation commands ..... 56
5.1.6 Special sector memory access commands ..... 67
5.1.7 Error correction code (ECC) and cyclic redundancy check commands ..... 70
5.1.8 Identification and serial number commands ..... 79
5.1.9 Low power modes and resets ..... 84
**6 Electrical characteristics** ..... **91**
6.1 Maximum ratings ..... 91

Table of contents

**7 Operating range ..... 92**
**8 DC electrical characteristics ..... 93**
**9 Data retention and endurance ..... 96**
**10 Capacitance ..... 97**
**11 Thermal resistance ..... 98**
**12 AC test conditions ..... 99**
**13 SDR AC switching characteristics ..... 100**
**14 DDR AC switching characteristics ..... 103**
**15 Write protect (WP) timing parameters ..... 105**
**16 Reset (RESET) timing parameters ..... 106**
**17 Power cycle timing ..... 107**
**18 Ordering information ..... 109**
18.1 Ordering code definitions ..... 109
**19 Package diagram ..... 110**
**20 Acronyms ..... 111**
**21 Document conventions ..... 112**
21.1 Units of measure ..... 112
**Revision history ..... 113**

# 1 Pinouts

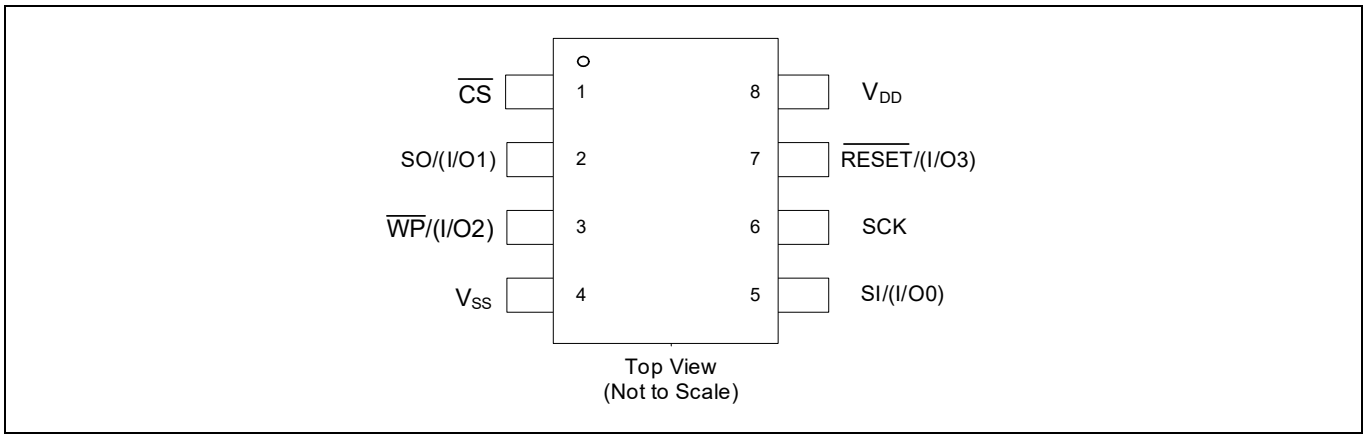



**Figure 1 8-pin SOIC pinout**

# 2 Pin definitions

**Table 1 Pin definitions**

| Pin name | I/O type | Description |
|---|---|---|
| $\overline{CS}$ | Input | **Chip select.** This active LOW input activates the device. When HIGH, the device enters low-power standby mode, ignores other inputs, and the output is tristated. When LOW, the device internally activates the SCK signal. A falling edge on $\overline{CS}$ must occur before a new opcode is issued. |
| SCK | Input | **Serial clock.** All I/O activity is synchronized to the serial clock. Inputs are latched on the rising edge and outputs occur on the falling edge. Because the device is synchronous, the clock frequency can vary between 0 and 108 MHz and may be interrupted at any time. |
| SI / (I/O0) | Input | **Serial input.** All data is input to the device on this pin. The pin is sampled on the rising edge of SCK and is ignored at other times. |
| | Input/ output | **I/O0:** When the part is either in dual mode or quad mode, the SI pin becomes input/output (I/O0) pin and acts as input during command and address cycles and output during the data output cycle. |
| SO / (I/O1) | Output | **Serial output.** This is the data output pin. It is driven during a read and remains tristated at all other times including when $\overline{RESET}$ is LOW. Data transitions are driven on the falling edge of the serial clock. |
| | Input/ output | **I/O1:** When the part is either in dual mode or quad mode, the SO pin becomes input/output (I/O1) pin and acts as input during command and address cycles and output during the data output cycle. |
| $\overline{WP}$ / (I/O2) | Input | **Write protect.** This active LOW pin prevents write operation to the status and configuration registers when SRWD bit (SR1[7]) is set to '1'. A complete explanation of write protection is provided in **"Status register 1 (SR1)"** on page 15. This pin must be tied to $V_{DD}$ if not used. |
| | Input/ output | **I/O2:** When the part is in quad mode, the $\overline{WP}$ pin becomes input/output (I/O2) pin and acts as input during command and address cycles and output during the data output cycle. |
| $\overline{RESET}$ / (I/O3) | Input | **Hardware reset pin.** This active LOW pin resets the device. When $\overline{RESET}$ is LOW, the device will self-initialize and will return to either standby state or active state depending on $\overline{CS}$ HIGH or LOW status after the $\overline{RESET}$ input is released to HIGH. This pin must be tied to $V_{DD}$ if not used. $\overline{RESET}$ / (I/O3) behavior is described in **Table 21**. |
| | Input/ output | **I/O3:** When the part is in quad mode, the $\overline{RESET}$ pin becomes input/output (I/O3) pin and acts as input during command and address cycles and output during the data output cycle. |
| $V_{SS}$ | Power supply | **Ground for the device.** Must be connected to the system ground. |
| VDD | Power supply | **Power supply input to the device.** |

# 3 Functional overview

The CY15X102QSN is a serial F-RAM memory. The memory array is logically organized as 262,144 × 8 bits and is accessed using an industry-standard serial peripheral interface (SPI) bus. The functional operation of the F-RAM is similar to single SPI EEPROM or single/dual/quad SPI flash. The key differences between the CY15X102QSN and a serial flash, with the same pinout, is the F-RAM's superior write performance, high endurance, and lower power consumption.

## 3.1 Memory architecture

When accessing the CY15X102QSN, the user addresses 256K locations of eight data bits each. These eight data bits are shifted in or out serially either on single, dual, or quad I/Os. The addresses are accessed using the SPI protocol, which includes a chip select (to permit multiple devices on the bus), an opcode, and a three-byte (24-bit) address. However, since CY15X102QSN requires only 18 bits to address its entire 256K byte locations, the upper six bits of the most significant address byte are 'don't care' values. The 18-bit address uniquely identifies each data byte location in the 256K memory array.

The access time for the memory operation is essentially zero, beyond the time needed for the serial protocol. That is, the memory is read or written at the speed of the SPI bus. Unlike a serial flash or EEPROM, it is not necessary to poll the device for a ready condition before initiating a new command. This is explained in more detail in **"Functional description"** on page 29.

## 3.2 SPI bus

The SPI is a synchronous serial interface, which uses clock and data pins for memory access and supports multiple devices on the data bus. A device on the SPI bus is activated using the $\overline{CS}$ pin. The relationship between chip select, clock, and data is dictated by the SPI mode. This device supports SPI modes 0 and 3. In both of these modes, data is clocked into the F-RAM on the rising edge of SCK starting from the first rising edge after $\overline{CS}$ goes active. The SPI protocol is controlled by opcodes. The $\overline{CS}$ must go inactive after an operation is complete and before a new opcode can be issued.

The CY15X102QSN is an SPI slave device and operates at speeds up to 108 MHz in single data rate (SDR) mode and at speeds up to 54 MHz in DDR mode. This high-speed serial bus provides high-performance serial communication to an SPI master. The CY15X102QSN supports four different SPI interface/protocol options: single channel SPI, extended SPI, dual SPI, quad SPI.

**Table 2** provides the I/O signaling details during opcode, address, and data phase in various SPI modes discussed above.

**Table 2 SPI modes and signal details**

| Interface | Single channel SPI | Extended SPI[1] | | | | Multi-channel SPI | |
|---|---|---|---|---|---|---|---|
| | | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI |
| Signals | $\overline{CS}$, SCK, SI, SO | $\overline{CS}$, SCK, I/O0, I/O1 | $\overline{CS}$, SCK, I/O0, I/O1, I/O2, I/O3 | $\overline{CS}$, SCK, I/O0, I/O1 | $\overline{CS}$, SCK, I/O0, I/O1, I/O2, I/O3 | $\overline{CS}$, SCK, I/O0, I/O1 | $\overline{CS}$, SCK, I/O0, I/O1, I/O2, I/O3 |
| Opcode | SI | I/O0 | I/O0 | I/O0 | I/O0 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |
| Address | SI | I/O0 | I/O0 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |
| Data | SI/SO | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |

**Note**
1. There is no user setting for the extended SPI modes. Device always starts with SPI mode and then changes to the respective extended SPI mode based on the opcode received.

### 3.2.1 Single channel SPI

The single channel SPI is a four-pin interface with chip select ($\overline{CS}$), serial input (SI), serial output (SO), and serial clock (SCK) pins. After $\overline{CS}$ is activated, the first byte transferred from the bus master is the opcode. Following the opcode, any addresses and data are then transferred. The $\overline{CS}$ must go HIGH (inactive) after an operation is complete and before a new opcode can be issued. This mode uses SI and SO pins for input and output respectively. Opcode and address is transferred by the master on the SI line, while data is read by the master on SO.

### 3.2.2 Extended SPI

The CY15X102QSN has the capability to reconfigure the standard SPI pins to work in dual or quad I/O modes called extended SPI modes. The extended SPI mode provides: dual data, dual input/output (I/O), quad data, and quad input/output (I/O) modes. The $\overline{CS}$ going HIGH after extended SPI command or device reset (either POR or hardware/software reset) brings the device back to the single channel SPI mode. Extended SPI mode has the following I/O configurations:

- When the part is in dual output or dual I/O mode, the SI pin and SO pin become I/O0 pin and I/O1 pin respectively.
- When the part is in quad output or quad I/O mode, the SI pin, SO pin, $\overline{WP}$ pin, and $\overline{RESET}$ pin become I/O0 pin, I/O1 pin, I/O2 pin, I/O3 pin respectively.
- Dual or quad data commands and addresses are sent to the memory only on the SI signal. Data will be returned to the host as a sequence of bit pairs on I/O0 and I/O1 or four bit (nibble) groups on I/O0, I/O1, I/O2, and I/O3.
- Dual or quad input/output (I/O) commands are sent to the memory only on SI signal while an address is sent from the host as bit pairs on I/O0 and I/O1 or, four bit (nibble) groups on I/O0, I/O1, I/O2, and I/O3 respectively. Data is returned to the host similarly as bit pairs on I/O0 and I/O1 or, four bit (nibble) groups on I/O0, I/O1, I/O2, and I/O3.

### 3.2.3 Dual SPI (DPI)

The CY15X102QSN DPI mode is enabled by writing ‘1’ at bit 4 of configuration register 2 (CR2), CR2[4] = ‘1’. Since configuration register 2 (CR2) has both volatile and nonvolatile space, user setting in the nonvolatile register will survive power and hardware reset cycles. Therefore, once the dual SPI (DPI) mode is set in the nonvolatile CR2, it always returns to the DPI mode until the host clears the DPI bit by writing ‘0’ in the nonvolatile CR2[4]. The host can change the device interface to DPI mode by writing ‘1’ to the volatile register CR2[4]; but this volatile setting will not survive the power and hardware reset cycles, and the volatile CR2[4] setting will be overwritten with default settings stored at associated nonvolatile location at power up or after the hardware reset cycle.

When the part is in dual SPI mode, the SI pin and SO pin become I/O0 pin and I/O1 pin respectively. Command, address, and data bits are sent to the memory from the host as bit pairs on I/O0 and I/O1. Data bits are returned to the host similarly as bit pairs on I/O0 and I/O1.

### 3.2.4 Quad SPI (QPI)

The CY15X102QSN multichannel QPI mode is enabled by writing '1' at bit 6 of configuration register 2 (CR2), CR2[6] = '1'. Since configuration register 2 (CR2) has both volatile and nonvolatile space, user setting in the nonvolatile register will survive power and hardware reset cycles. Therefore, once the quad SPI (QPI) mode is set in the nonvolatile CR2, it always returns to the QPI mode until the host clears the QPI bit by writing '0' in the nonvolatile CR2[6]. The host can change the device interface to QPI mode by writing '1' to the volatile register CR2[6]; but this volatile setting will not survive the power and hardware reset cycles, and the volatile CR2[6] setting will be overwritten with default settings stored at associated nonvolatile location at power up or after the hardware reset cycle.

When the part is in quad SPI mode, the SI pin, SO pin, $\overline{WP}$ pin, and $\overline{RESET}$ pins become I/O0 pin, I/O1 pin, I/O2 pin, I/O3 pin respectively. Command, address, and data bits are sent to the memory from the host as four bit (nibble) groups on I/O0, I/O1, I/O2, and I/O3. Data bits are returned to the host similarly as four bit (nibble) groups on I/O0, I/O1, I/O2, and I/O3.

The QPI mode also supports DDR through special opcodes where byte transfer occurs on both edges of the clock for address, mode, and data bytes. There is no DDR mode during the opcode phase; that is, opcodes are always transmitted in SDR mode. The device enters DDR mode after a specific command is transmitted in SDR mode, which then determines the address, mode, and data cycles in DDR. There is no setting for enabling the DDR mode. The quad SPI DDR mode is only supported for memory write and read operations with special opcodes.

## 3.3 Terms used in SPI protocol

The commonly used terms in the SPI protocol are as follows:

### 3.3.1 SPI master

The SPI master device controls the operations on the SPI bus. An SPI bus may have only one master with one or more slave devices. All the slaves share the same SPI bus lines and the master may select any of the slave devices using the $\overline{CS}$ pin. All of the operations must be initiated by the master activating a slave device by pulling the $\overline{CS}$ pin of the slave LOW. The master also generates the SCK and all the data transmission on SI and SO lines are synchronized with this clock.

### 3.3.2 SPI slave

The SPI slave device is activated by the master through the chip select line. A slave device gets the SCK as an input from the SPI master and all the communication is synchronized with this clock. An SPI slave never initiates a communication on the SPI bus and acts only on the instruction from the master.

The CY15X102QSN operates as an SPI slave and may share the SPI bus with other SPI slave devices.

### 3.3.3 Chip select ($\overline{CS}$)

To select any slave device, the master needs to pull down the corresponding $\overline{CS}$ pin. Any instruction can be issued to a slave device only while the $\overline{CS}$ pin is LOW. When the device is not selected, data through the SI pin is ignored and the serial output pin (SO) remains in a high-impedance state.

**Note:** A new instruction must begin with the falling edge of $\overline{CS}$. Therefore, only one opcode can be issued for each active $\overline{CS}$ HIGH to LOW transition.

### 3.3.4 Serial clock (SCK)

The serial clock is generated by the SPI master and the communication is synchronized with this clock after $\overline{CS}$ goes LOW.

The CY15X102QSN enables SPI modes 0 and 3 for data communication. In both of these modes, the inputs are latched by the slave device on the rising edge of SCK and outputs are issued on the falling edge. Therefore, the first rising edge of SCK signifies the arrival of the first Most Significant Bit (MSb) of an SPI instruction on the SI pin. Further, all data inputs and outputs are synchronized with SCK.

### 3.3.5 Data transmission (SI/SO)

The SPI data bus consists of two lines, SI and SO, for serial data communication. SI is also referred to as master-out-slave-in (MOSI) and SO is referred to as master-in-slave-out (MISO). The master issues instructions to the slave through the SI pin, while the slave responds through the SO pin. Multiple slave devices may share the SI and SO lines as described earlier.

The CY15X102QSN has two separate pins for SI and SO, which can be connected with the master as shown in **Figure 2**. When in dual or quad I/O modes, these pins are configured as I/O pins. **Figure 3** shows such a system interface with a QSPI port.

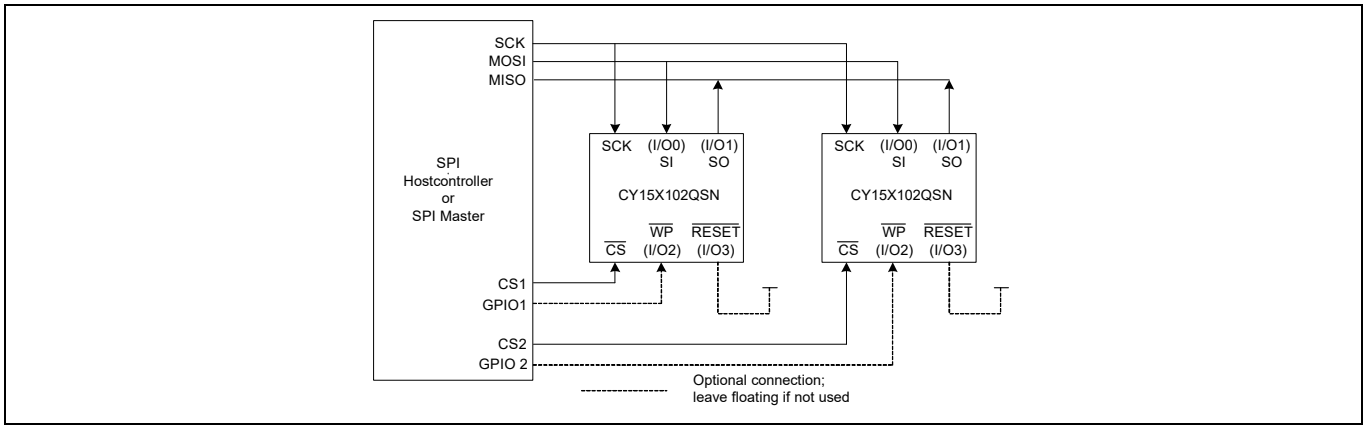



**Figure 2 System configuration with SPI port**

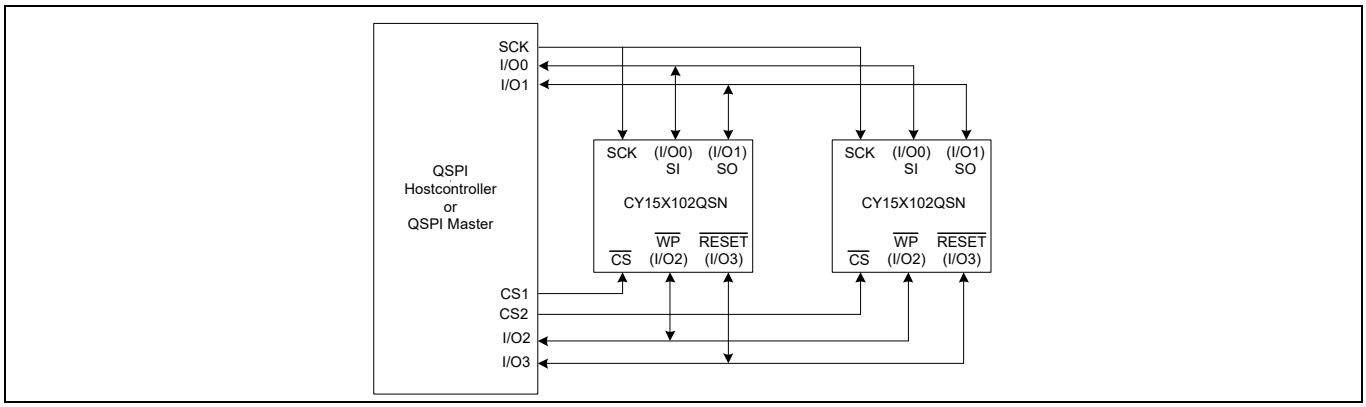



**Figure 3 System configuration with QSPI port**

### 3.3.6 Most significant bit (MSb)

The SPI protocol requires that the first bit to be transmitted is the most significant bit (MSb). This is valid for both address and data transmission.

The 2-Mbit serial F-RAM requires a 3-byte address for any read or write operation. Because the address is only 18 bits, the six bits, which are fed in are ignored by the device. Although these five bits are ‘don’t care’, Infineon recommends that these bits be set to 0s to enable seamless transition to higher memory densities.

### 3.3.7 Serial opcode

After the slave device is selected with $\overline{CS}$ going LOW, the first byte received is treated as the opcode for the intended operation. CY15X102QSN uses the standard opcodes (refer to **Table 32**) for memory accesses.

### 3.3.8 Invalid opcode

If a reserved opcode is received, the opcode may internally trigger unintended operation and start driving the I/O pin(s) with a non-deterministic data output. Hence, all opcodes under the reserved category should be avoided to transmit over SI pin when CY15X102QSN chip select $\overline{CS}$ is LOW.

### 3.3.9 Instruction

Instruction is the combination of the opcode, address, mode and/or dummy bytes/cycles used to access the memory and registers.

### 3.3.10 Mode byte

The mode byte is applicable for all write and read commands that support execute-in-place (XIP). The XIP is a method of executing the program (code) directly from an external memory rather than copying or shadowing the code into RAM. When the XIP is set for a write or read command, the device stays in XIP mode after the command cycle is terminated ($\overline{CS}$ toggles HIGH) so that the subsequent command cycle with $\overline{CS}$ LOW directly starts with the address phase (opcode phase is skipped). When in XIP, the device executes the same operation as in previous cycle. To initiate a new operation while in XIP, for example to switch from memory write to memory read or vice versa, the device should first exit the XIP for the current command cycle and initiate the next command cycle with opcode phase. Opcodes with the mode phase only support the XIP. See **Table 32** for the list of opcodes that require mode phase.

Following the opcode and 3-byte address cycles, the mode byte 0xAX (X don't care bits) or 0xA5 (depending on the opcode) transmitted during the mode phase keeps the device in XIP for the next command cycle. The XIP must be set during every command cycle to remain in XIP for the next command cycle. Any other value than 0xAX or 0xA5 (!0xAX or !0xA5) transmitted during the mode phase will exit the XIP for the current operation. In this case, the next command cycle must always start with the opcode phase to start the same operation or a new operation. Depending upon the SPI mode and the interface type, the number of clocks to transmit the mode byte will vary from one clock (quad, DDR) to eight clocks (SPI, SDR).

### 3.3.11 Wait states or dummy cycles

The wait states, also called dummy cycles, are appended after the address bits and mode bits (if applicable). The number of wait state cycles are programmable through configuration register 1 (CR1) and configuration register 2 (CR2) for both memory and registers reads respectively. A valid data is driven on the output bus only after specific number of dummy cycles are elapsed following memory and register read commands that support wait state. A dummy cycle is a full clock cycle irrespective of the SPI modes and data rates (SDR or DDR). The status of I/Os are don't care during dummy cycle.

## 3.4 SPI modes

CY15X102QSN may be driven by a microcontroller with its SPI peripheral running in either of the following two modes:

- SPI mode 0 (CPOL = 0, CPHA = 0)
- SPI mode 3 (CPOL = 1, CPHA = 1)

The device detects the SPI mode from the status of the SCK pin when the device is selected by bringing the $\overline{CS}$ pin LOW. If the SCK pin is LOW when the device is selected, SPI mode 0 is assumed and if the SCK pin is HIGH, it works in SPI mode 3. The two SPI modes are shown in **Figure 4** and **Figure 5**. The status of the clock SCK when the bus master is not transferring data is:

- SCK remains at 0 for mode 0
- SCK remains at 1 for mode 3

SPI mode 0 and SPI mode 3 are supported for all SDR mode commands. While, all DDR mode commands support only SPI mode 0.

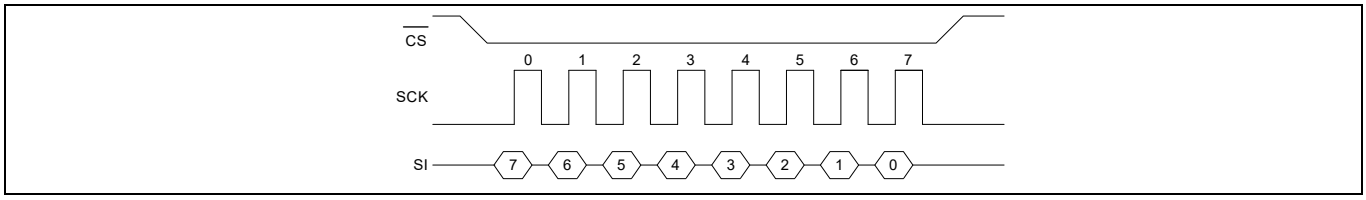



**Figure 4 SPI mode 0**

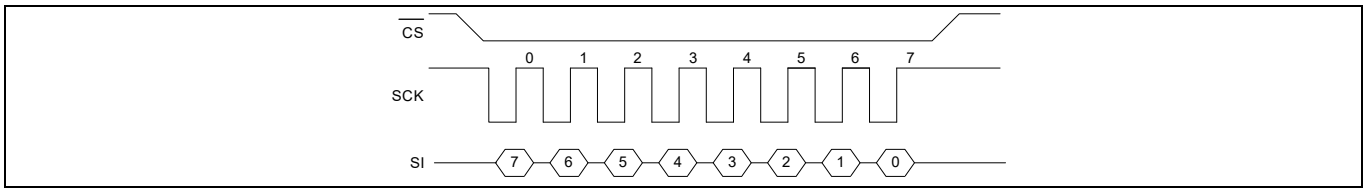



**Figure 5 SPI mode 3**

### 3.4.1 SDR

The input data bits (includes instruction, address, and data) are always latched in on the rising edge of SCK starting from the first rising edge after $\overline{CS}$ goes active. If the clock starts from a HIGH state (in mode 3), the first rising edge after the clock toggles is considered. The output data is available on the falling edge of SCK.

### 3.4.2 DDR

The instruction bits are always latched on the rising edge of SCK starting from the first rising edge after CS goes active. If the clock starts from a HIGH state (in mode 3), the first rising edge after the clock toggles is considered. However, the address and input data that follow the instruction are latched on both the rising and falling edges of SCK. The first address bit is latched on the first rising edge of SCK following the falling edge at the end of the last instruction bit. The first bit of output data is driven on the falling edge of SCK at the end of the last access latency (dummy) cycle.

## 3.5 Power-up to first access

When the CY15X102QSN power supply ($V_{DD}$) falls below $V_{DD}$(low), the power-up cycle starts. CY15X102QSN waits for the $V_{DD}$ power supply to rise above the minimum $V_{DD}$(min), after which the device stars its internal boot-up sequence. The boot-up sequence for CY15X102QSN includes internal power-on-reset (POR) followed by loading the internal device configuration and trim registers as well as setting the user accessible registers. All user accessible registers (status and configuration, mode, ID, ECC, and CRC) are set to their default values after a successful boot-up cycle. **Table 3** shows the status of each register of CY15X102QSN after a successful power-up (or POR) sequence.

CY15X102QSN ignores all instructions until a time delay of $t_{PU}$ has elapsed after the moment $V_{DD}$ rises above $V_{DD}$(min). No instruction should be sent to the device until the end of $t_{PU}$. After the $t_{PU}$, if $\overline{CS}$ is HIGH, the device enters standby mode and draws standby current ($I_{SB}$). The device enters deep power-down mode after $t_{PU}$ if the deep power-down mode upon POR (DPDPOR) in configuration register 4 (CR4) is set to '1' (CR4 [2] = '1').

The WIP bit of status register 1 (SR1[0]) cannot be used to poll the device readiness after the POR event because device is still not accessible for executing any command including RDSR1 until the $t_{PU}$ time is over. However, if the WIP status remains HIGH even after $t_{PU}$ time or device remains inaccessible, indicates device didn't boot up correctly (boot error). Once the boot error occurs, the device enters the following default states:

- The interface mode is set to single SPI (SDR)
- IO3R bit of CR2 (CR2[5]) is internally set '1' to enable the hardware reset ($\overline{RESET}$) on IO3
- Register latency is set to three-clock cycle (max value)
- Output impedance is set to 45 ohm
- Only RDSR1 and RDAR commands are allowed (in SPI SDR mode only) to read the SR1. All other commands will remain disabled and will return undefined data if executed.
- Reading the SR1 returns 0x61 as boot error signature

CY15X102QSN will require power cycle or hardware reset to restart the boot-up again. The above default settings will be replaced with actual user configurations after a successful boot-up.

**Table 3 CY15X102QSN registers status after POR**

| Function | Register type | CY15X102QSN registers status after POR |
|---|---|---|
| Device status | Status register 1 (SR1) | Default to corresponding nonvolatile bits |
| | Status register 2 (SR2) | 0x00 |
| Device configuration[2] | Configuration register 1 (CR1) | Default to corresponding nonvolatile bits |
| | Configuration register 2 (CR2) | Default to corresponding nonvolatile bits |
| | Configuration register 4 (CR4) | Default to corresponding nonvolatile bits |
| | Configuration register 5 (CR5) | Default to corresponding nonvolatile bits |
| Identification | Identification register | Default to corresponding nonvolatile bits (factory set) |
| | Unique identification register | Default to corresponding nonvolatile bits (factory set) |
| | Serial number register | Default to corresponding nonvolatile bits (factory set to 0x0000000000000000) |
| Error correction | ECC status register | 0x00 |
| | ECC count register | 0x0000 |
| | ECC address trap register | 0x00000000 |
| Cyclic redundancy check | CRC register | 0x00000000 |

**Note**
2. Configuration register 3 (CR3) is reserved for future use.

# 4 CY15x102QSN registers

CY15X102QSN supports various status and configuration registers for device status update and configuration settings. CY15X102QSN registers and their access details are discussed in follow on sections.

## 4.1 Status registers

The CY15X102QSN supports two status registers - status register 1 (SR1) and status register 2 (SR2) to provide the write protect settings as well ready/CRC status of the device. The SR1 register has a volatile and an associated nonvolatile register space in the F-RAM. The nonvolatile register retains the device configuration during power down which is then copied to the respective volatile register during power up or after the hardware reset (JEDEC reset or RESET pin). The CY15X102QSN state machine uses only the volatile register settings to change the device configuration during normal access. Since the CY15X102QSN provides independent space for both volatile and nonvolatile configuration registers, the host can program the volatile register only to make the configuration effective for the current power cycle. The nonvolatile write changes the content of both volatile and nonvolatile registers. Therefore, the new configurations become effective immediately for the current power cycle as well as subsequent power cycles or hardware reset cycles. The SR2 is a read only register.

Read from status registers either uses dedicated status register read opcodes (RDSR1, RDSR2) or RDAR followed status register address. The status register read always returns the volatile register content. Individual status register details are provided in follow on sections.

### 4.1.1 Status register 1 (SR1)

The status register 1 (SR1), as shown in **Table 4**, contains both status and write protect control bits. The SR1 is accessible by WRSR and WRAR command for write and the RDSR1 or the RDAR command for read operations. The SR1 access details are provided in **"Register access commands"** on page 34.

WRAR nonvolatile write address - 0x000000

WRAR volatile write address - 0x070000

RDAR read address - 0x000000 or 0x070000

The default state shown after each bit in **Table 4** is the factory programmed value.

**Table 4 Status register 1 (SR1)**

| SR1[7] | SR1[6] | SR1[5] | SR1[4] | SR1[3] | SR1[2] | SR1[1] | SR1[0] |
|---|---|---|---|---|---|---|---|
| SRWD (0) | RFU (0) | TBPROT (0) | BP2 (0) | BP1 (0) | BP0 (0) | WEL (0) | WIP (0) |

**Table 5 Status register 1 (SR1) - nonvolatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| SR1[7] | SRWD | Status register write disable | NV | R/W | 1 = Locks state of status & configuration registers when $\overline{WP}$ is LOW<br>0 = No register protection irrespective of the status of $\overline{WP}$ pin |
| SR1[6] | RFU | Reserved (0) | | | Reserved for future use |
| SR1[5] | TBPROT | Top/bottom relative protection | NV | R/W | 1 = Protection starts at memory array bottom<br>0 = Protection starts at memory array top |
| SR1[4] | BP2 | Block protect bit | NV | R/W | Protects the selected address range of memory array |
| SR1[3] | BP1 | | NV | | |
| SR1[2] | BP0 | | NV | | |
| SR1[1] | WEL | Write enable latch | V | R | WEL indicates if the device is write enabled. This bit defaults to '0' (disabled) on power-up.<br>WEL = '1' --> Write enabled<br>WEL = '0' --> Write disabled |
| SR1[0] | WIP | Work in progress | V | R | 1 = Device busy<br>0 = Device ready |

NV - Nonvolatile; V - Volatile

**Table 6 Status register 1 (SR1) - volatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| SR1[7] | SRWD | Status register write disable | V | R/W | 1 = Locks state of status & configuration registers when $\overline{WP}$ is LOW<br>0 = No register protection irrespective of the status of $\overline{WP}$ pin |
| SR1[6] | RFU | Reserved (0) | | | Reserved for future use |
| SR1[5] | TBPROT | Top/bottom relative protection | V | R/W | 1 = Protection starts at memory array bottom<br>0 = Protection starts at memory array top |
| SR1[4] | BP2 | Block protect bit | V | R/W | Protects the selected address range of memory array |
| SR1[3] | BP1 | | V | | |
| SR1[2] | BP0 | | V | | |
| SR1[1] | WEL | Write enable latch | V | R | WEL indicates if the device is write enabled. This bit defaults to '0' (disabled) on power-up.<br>WEL = '1' --> Write enabled<br>WEL = '0' --> Write disabled |
| SR1[0] | WIP | Work in progress | V | R | 1 = Device busy<br>0 = Device ready |

V - Volatile

### 4.1.1.1 Status register protect (SRWD) SR1 [7]

This bit enables write protect for the status and configuration registers when set to '1' and the write protect ($\overline{WP}$) pin is driven LOW. In this mode, any instruction that changes the status registers or configuration registers content is ignored, effectively locking the state of the device. If the SRWD is set to '0', irrespective of the $\overline{WP}$ status (LOW or HIGH), status and configuration registers write protection remains disabled. Refer to **Table 9** for the memory and status register protection options.

### 4.1.1.2 Top or bottom protection (TBPROT) SR1 [5]

This bit defines the operation of the block protection bits BP2, BP1, and BP0. This bit controls the starting point of the memory array (from top or bottom) memory that gets protected by the block protection bits.

**Table 7 Start of protection from top (TBPROT = '0')**

| Status register content | | | Protected fraction of memory array | Protected address range |
|---|---|---|---|---|
| BP2 | BP1 | BP0 | | |
| 0 | 0 | 0 | None | None |
| 0 | 0 | 1 | Upper 1/64$^{th}$ of memory array | 0x03F000–0X03FFFF |
| 0 | 1 | 0 | Upper 1/32$^{nd}$ of memory array | 0x03E000–0X03FFFF |
| 0 | 1 | 1 | Upper 1/16$^{th}$ of memory array | 0x03C000–0X03FFFF |
| 1 | 0 | 0 | Upper 1/8$^{th}$ of memory array | 0x038000–0X03FFFF |
| 1 | 0 | 1 | Upper 1/4$^{th}$ of memory array | 0x030000–0x03FFFF |
| 1 | 1 | 0 | Upper half of memory array | 0x020000–0x03FFFF |
| 1 | 1 | 1 | Full memory | 0x000000–0x03FFFF |

**Table 8 Start of protection from bottom (TBPROT = '1')**

| Status register content | | | Protected fraction of memory array | Protected address range |
|---|---|---|---|---|
| BP2 | BP1 | BP0 | | |
| 0 | 0 | 0 | None | None |
| 0 | 0 | 1 | Lower 1/64$^{th}$ of memory array | 0x000000–0x000FFF |
| 0 | 1 | 0 | Lower 1/32$^{nd}$ of memory array | 0x000000–0x001FFF |
| 0 | 1 | 1 | Lower 1/16$^{th}$ of memory array | 0x000000–0x003FFF |
| 1 | 0 | 0 | Lower 1/8$^{th}$ of memory array | 0x000000–0x007FFF |
| 1 | 0 | 1 | Lower 1/4$^{th}$ of memory array | 0x000000–0x00FFFF |
| 1 | 1 | 0 | Lower half of memory array | 0x000000–0x01FFFF |
| 1 | 1 | 1 | Full memory | 0x000000–0x03FFFF |

### 4.1.1.3 Block protection (BP2, BP1, and BP0) SR1 [4:2]

These bits define the memory array to be write-protected against memory write commands. When one or more of the BP bits is set to '1', the respective memory address is protected from write. The block protect bits (BP2, BP1, and BP0) in combination with the TBPROT bit can be used to protect an address range of the memory array. The size of the range is determined by the value of the BP bits and the upper or lower starting point of the range which is selected by the TBPROT. **Table 7** and **Table 8** show CY15X102QSN protected address range for BP[2:0] bits setting.

### 4.1.1.4 Write enable latch (WEL) SR1 [1]

The WEL bit must be set to 1 to enable write operations to memory array or registers, as shown in **Table 9**. This bit is set to '1' only by executing the write enable (WREN) command. The WEL bit (SR1[1]) automatically clears to '0' on the rising edge of CS following opcodes including: WRDI (04h), WRSR (01h), SSWR (42h), WRAR (71h), and WRSN (C2h). The WEL bit (SR1[1]) doesn't clear to '0' on the rising edge of CS following memory write opcodes. The WEL bit is volatile and returns to its default '0' state after POR and all reset events.

**Table 9 Write protection**

| SRWD | $\overline{WP}$ | WEL | Protected blocks | Unprotected blocks | Status and configuration registers[3] |
|---|---|---|---|---|---|
| X | X | 0 | Protected | Protected | Protected |
| 0 | X | 1 | Protected | Writable | Writable |
| 1 | 0 | 1 | Protected | Writable | Protected |
| 1 | 1 | 1 | Protected | Writable | Writable |

**Note**
3. All bits except read only and reserved bits.

#### 4.1.1.5 Work-in-progress (WIP) SR1 [0]

This is a read-only bit and indicates device ready or busy status during normal operation. The CY15X102QSN sets this bit to '1' while executing the CRC calculation. No other command(s) and event(s) set the WIP to '1' in CY15X102QSN. When WIP is '1', the CY15X102QSN can execute only read status registers using RDSR1/RDSR2 or read any register (RDAR followed by status register address), CRC suspend (EPCS), and software reset (RSTEN followed by RST) commands. Other commands will be ignored while WIP is '1'. The WIP bit can't be used to poll the device ready status during power up or reset cycles. This bit is volatile and returns to its default state after POR and all reset events.

### 4.1.2 Status register 2 (SR2)

The status register 2 (SR2), as shown in **Table 10**, provides the device status on CRC operations. The SR2 is a read-only volatile register and is accessible by RDSR2 or the RDAR command for read operations. The SR1 access details are provided in **"Register access commands"** on page 34.

RDAR read address - 0x000001 or 0x070001

The default state shown after each bit in **Table 11** is the factory programmed value.

**Table 10 Status register 2 (SR2)**

| SR2[7] | SR2[6] | SR2[5] | SR2[4] | SR2[3] | SR2[2] | SR2[1] | SR2[0] |
|---|---|---|---|---|---|---|---|
| RFU (0) | RFU (0) | RFU (0) | CRCS (0) | CRCA (0) | RFU (0) | RFU (0) | RFU (0) |

**Table 11 Status register 2 (SR2) - volatile only**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| SR2[7] | RFU | Reserved (0) | | | Reserved for future use |
| SR2[6] | RFU | Reserved (0) | | | Reserved for future use |
| SR2[5] | RFU | Reserved (0) | | | Reserved for future use |
| SR2[4] | CRCS | CRC suspend | V | R | 1 = in CRC suspend mode<br>0 = not in CRC suspend mode |
| SR2[3] | CRCA | CRC abort | V | R | 1 = CRC command aborted<br>0 = CRC command not aborted |
| SR2[2] | RFU | Reserved (0) | | | Reserved for future use |
| SR2[1] | RFU | Reserved (0) | | | Reserved for future use |
| SR2[0] | RFU | Reserved (0) | | | Reserved for future use |

V - Volatile

#### 4.1.2.1 CRC suspend (CRCS) SR2 [4]

The CRC suspend (CRCS) bit is used to determine whether the device is in CRC suspend mode. When the device CRC calculation is in progress, executing the CRC suspend command (EPCS) will set this bit to '1' to indicate the CRC suspend status. The CRC resume (EPCR) command clears the CRCS bit to '0', indicates device exited the CRC suspend mode. This is a read only bit. This bit also gets cleared after resets (POR, hardware, and software).

#### 4.1.2.2 CRC abort (CRCA) SR2 [3]

This bit indicates whether the CRC calculation (CRCC) operation is aborted. The CRC calculation is aborted when end address and start address criteria (EA < SA + 3), which is ending address should be at least 32-bit aligned word higher than the starting address, doesn't meet. This bits gets clears when subsequent CRC calculation starts successfully. This bit also gets cleared after reset (POR, hardware, and software).

## 4.2 Configuration registers

The CY15X102QSN supports four user configuration registers - CR1, CR2, CR4, and CR5 to program various controls in the device. Each configuration register has a volatile and an associated nonvolatile register space in the F-RAM. The nonvolatile registers retain the device configuration during power down which are then copied to their respective volatile registers during power up or after the hardware reset (JEDEC reset or RESET pin). The CY15X102QSN state machine uses only the volatile register settings to change the device configuration during normal access. Since the CY15X102QSN provides independent space for both volatile and nonvolatile configuration registers, the host can program volatile register only to make the configuration effective for the current power cycle. The nonvolatile write changes the content of both volatile and nonvolatile registers. Therefore, new configurations become effective immediately for the current power cycle as well as subsequent power cycles or hardware reset cycles.

Read from configuration registers either using dedicated configuration register read opcodes (RDCR1, RDCR2, RDCR3, RDCR4) or RDAR always returns the volatile register content. Individual configuration register details are provided in follow on sections.

### 4.2.1 Configuration register 1 (CR1)

The configuration register 1 (CR1), as shown in **Table 12**, configures the latency (dummy) cycles for memory and special sector reads and enables the quad I/O during extended SPI access. The CR1 is accessible by the WRAR command for write and the RDCR1 or the RDAR command for read operations. The CR1 access details are provided in **"Register access commands"** on page 34.

WRAR nonvolatile write address - 0x000002

WRAR volatile write address - 0x070002

RDAR read address - 0x000002 or 0x070002

The default state shown after each bit in **Table 12** is the factory programmed value.

**Table 12 Configuration register 1 (CR1)**

| CR1[7] | CR1[6] | CR1[5] | CR1[4] | CR1[3] | CR1[2] | CR1[1] | CR1[0] |
|---|---|---|---|---|---|---|---|
| MLC3 (0) | MLC2 (0) | MLC1 (0) | MLC0 (0) | RFU (0) | RFU (0) | QUAD (0) | RFU (0) |

**Table 13 Configuration register 1 (CR1) - nonvolatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR1[7] | MLC3 | Memory latency code | NV | R/W | Configures number of latency (dummy) cycles for the memory as well as special sector read opcodes.<br>Example-<br>0000 - 0 cycle<br>0110 - 6 cycles<br>1111 - 15 cycles |
| CR1[6] | MLC2 | | NV | | |
| CR1[5] | MLC1 | | NV | | |
| CR1[4] | MLC0 | | NV | | |
| CR1[3] | RFU | Reserved (0) | | | Reserved for future use |

**Table 13 Configuration register 1 (CR1) - nonvolatile** (continued)

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR1[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR1[1] | QUAD | Quad | NV | R/W | 1 = Quad<br>0 = Dual or serial |
| CR1[0] | RFU | Reserved (0) | | | Reserved for future use |

NV - Nonvolatile

**Table 14 Configuration register 1 (CR1) - volatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR1[7] | MLC3 | Memory latency code | V | R/W | Configures number of latency (dummy) cycles for the memory as well as special sector read opcodes.<br>Example-<br>0000 - 0 cycle<br>0110 - 6 cycles<br>1111 - 15 cycles |
| CR1[6] | MLC2 | | V | | |
| CR1[5] | MLC1 | | V | | |
| CR1[4] | MLC0 | | V | | |
| CR1[3] | RFU | Reserved (0) | | | Reserved for future use |
| CR1[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR1[1] | QUAD | Quad | V | R/W | 1 = Quad<br>0 = Dual or serial |
| CR1[0] | RFU | Reserved (0) | | | Reserved for future use |

V - Volatile

### 4.2.1.1 Memory latency code (MLC) CR1 [7:4]

These four bits configures the latency (dummy) cycles for all variable latency memory read instructions. It enables the user to adjust the memory read latency during normal operation to optimize the latency for different instructions at different operating frequencies. Dummy cycles are full clock cycles on SCK irrespective of the SPI modes and data rates (SDR and DDR).

Some read opcodes support dummy cycles following address cycles. These dummy cycles provide additional latency that is needed to complete the initial read access of the memory array before data can be returned to the host system. As the SPI clock (SCK) frequency increase, number of dummy cycles need to increase to meet the latency.

**Table 15** to **Table 17** show the max SPI clock frequency versus clock latency for each opcodes that support dummy cycles. The host controller can determine to optimize the timing by setting individual latency cycle for each opcode or can set the worst case latency which meets the latency requirement of all opcodes for a desired operating frequency. The memory read latency set for a higher frequency also applies for all lower frequencies. Hence, when the host lowers the SPI clock (SCK) from higher frequency to a lower frequency, adjusting the clock latency becomes optional.

The format (CMD, ADD, DATA) in **Table 15** header represents the transmission of these bytes over number of I/Os in different SPI modes. For example: (2, 2, 2) represents all command (CMD), address (ADDR), and data (DATA) bytes are transmitted over two I/Os (I/O0 and I/O1) in DPI mode. Similarly, (1, 2, 2) represents CMD byte is transmitted over a single I/O (I/O0), while ADDR and DATA bytes are transmitted over two I/Os (I/O0, I/O1) in dual I/O mode. (1, 1, 4) represents CMD and ADDR bytes are transmitted over a single I/O (I/O0), while DATA bytes are transmitted over four I/Os (I/O0, I/O1, I/O2, I/O3) in Quad Data mode.

Mode represents number of clock cycles required in various SPI interface modes to transmit the mode byte after address bits. Since mode bits are transmitted after the address cycles, clock cycles required to transmit mode bits are internally added to the latency calculation.

**Table 15 Latency (dummy) cycles for memory read commands - with XIP mode (SDR)**

| Latency (dummy) cycles - decimal | SPI (SDR) | DPI (SDR) | QPI (SDR) | Dual data (SDR) | Dual I/O (SDR) | Quad data (SDR) | Quad I/O (SDR) |
|---|---|---|---|---|---|---|---|
| | FAST_READ | FAST_READ | FAST_READ, QIOR | DOR | DIOR | QOR | QIOR |
| | (1, 1, 1) | (2, 2, 2) | (4, 4, 4) | (1, 1, 2) | (1, 2, 2) | (1, 1, 4) | (1, 4, 4) |
| | Mode = 8 | Mode = 4 | Mode = 2 | Mode = 8 | Mode = 4 | Mode = 8 | Mode = 2 |
| **0** | 108 MHz | 55 MHz[4] | 10 MHz[4] | 108 MHz | 55 MHz[4] | 108 MHz | 10 MHz[4] |
| **1** | 108 MHz | 70 MHz[4] | 25 MHz[4] | 108 MHz | 70 MHz[4] | 108 MHz | 25 MHz[4] |
| **2** | 108 MHz | 80 MHz[4] | 40 MHz[4] | 108 MHz | 80 MHz[4] | 108 MHz | 40 MHz[4] |
| **3** | 108 MHz | 95 MHz[4] | 55 MHz[4] | 108 MHz | 95 MHz[4] | 108 MHz | 55 MHz[4] |
| **4** | 108 MHz | 108 MHz | 70 MHz[4] | 108 MHz | 108 MHz | 108 MHz | 70 MHz[4] |
| **5** | 108 MHz | 108 MHz | 80 MHz[4] | 108 MHz | 108 MHz | 108 MHz | 80 MHz[4] |
| **6** | 108 MHz | 108 MHz | 95 MHz[4] | 108 MHz | 108 MHz | 108 MHz | 95 MHz[4] |
| **7–15** | 108 MHz | 108 MHz | 108 MHz | 108 MHz | 108 MHz | 108 MHz | 108 MHz |

**Note**
4. This parameter is guaranteed by characterization; not tested in production.

**Table 16 Latency (dummy) cycles for memory read commands - with XIP mode (DDR)**

| Latency (dummy) cycles - decimal | QPI (DDR) | Quad I/O (DDR) |
|---|---|---|
| | DDRFR, DDRQIOR | DDRQIOR |
| | (4, 4, 4) | (1, 4, 4) |
| | Mode = 1 | Mode = 1 |
| **0** | NA | NA |
| **1** | NA | NA |
| **2** | 10 MHz[5] | 10 MHz[5] |
| **3** | 25 MHz[5] | 25 MHz[5] |
| **4** | 33 MHz[5] | 33 MHz[5] |
| **5** | 40 MHz[5] | 40 MHz[5] |
| **6** | 50 MHz[5] | 50 MHz[5] |
| **7–15** | 54 MHz | 54 MHz |

**Note**
5. This parameter is guaranteed by characterization; not tested in production.

**Table 17 Latency (dummy) cycles for memory read commands - without XIP mode**

| Latency (dummy) cycles - decimal | SPI (SDR) | DPI (SDR) | QPI (SDR) |
|---|---|---|---|
| | READ, ECCRD, SSRD | | |
| | (1, 1, 1) | (2, 2, 2) | (4, 4, 4) |
| | Mode = NA | Mode = NA | Mode = NA |
| **0** | 40 MHz[6] | NA | NA |
| **1** | 55 MHz[6] | NA | NA |
| **2** | 70 MHz[6] | 25 MHz[6] | 10 MHz[6] |
| **3** | 80 MHz[6] | 40 MHz[6] | 25 MHz[6] |
| **4** | 95 MHz[6] | 55 MHz[6] | 40 MHz[6] |
| **5** | 108 MHz | 70 MHz[6] | 55 MHz[6] |
| **6** | 108 MHz | 80 MHz[6] | 70 MHz[6] |
| **7** | 108 MHz | 95 MHz[6] | 80 MHz[6] |
| **8** | 108 MHz | 108 MHz | 95 MHz[6] |
| **9–15** | 108 MHz | 108 MHz | 108 MHz |

**Note**
6. This parameter is guaranteed by characterization; not tested in production.

### 4.2.1.2 Quad data width (QUAD) CR1 [1]

When set to '1', this bit switches the data width of the device to 4 I/Os – quad mode, that is $\overline{WP}$ becomes I/O2 and $\overline{RESET}$ / (I/O3) becomes I/O3. If the alternate function is enabled on I/O3 by setting IO3R bit in configuration register 2 (CR2[5]), $\overline{RESET}$ / (I/O3) works as I/O3 when $\overline{CS}$ is low and $\overline{RESET}$ input when $\overline{CS}$ is HIGH. The $\overline{WP}$ input is disabled and is internally set to '1'. The QUAD bit must be set to '1' when executing the extended SPI read commands: quad output read, and quad I/O Read, and DDR quad I/O Read. The impact of "QUAD" bit setting on various SPI interfaces are shown in **Table 21**.

## 4.2.2 Configuration register 2 (CR2)

The configuration register 2 (CR2), as shown in **Table 18**, controls the serial interface settings. The CR2 is accessible by the WRAR command for write and the RDCR2 or the RDAR command for read operations. The CR2 access details are provided in **"Register access commands"** on page 34.

WRAR nonvolatile write address - 0x000003

WRAR volatile write address - 0x070003

RDAR read address - 0x000003 or 0x070003

The default state shown after each bit in **Table 18** is the factory programmed value.

**Table 18 Configuration register 2 (CR2)**

| CR2[7] | CR2[6] | CR2[5] | CR2[4] | CR2[3] | CR2[2] | CR2[1] | CR2[0] |
|---|---|---|---|---|---|---|---|
| RFU (0) | QPI (0) | IO3R (0) | DPI (0) | RFU (0) | RFU (0) | RFU (0) | RFU (0) |

**Table 19 Configuration register 2 (CR2) - nonvolatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR2[7] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[6] | QPI | Quad SPI enable | NV | R/W | 1 = Enable QPI protocol<br>0 = Enable SPI protocol, if DPI bit is set to '0' |
| CR2[5] | IO3R | IO3 reset | NV | R/W | 1 = I/O3 is used as $\overline{\text{RESET}}$ input when $\overline{\text{CS}}$ is HIGH<br>0 = I/O3 has no alternate function |
| CR2[4] | DPI | Dual SPI enable | NV | R/W | 1 = Enable DPI protocol<br>0 = Enable SPI protocol, if QPI bit is set to '0' |
| CR2[3] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[0] | RFU | Reserved (0) | | | Reserved for future use |

NV - Nonvolatile

**Table 20 Configuration register 2 (CR2) - volatile**

| Bit | Bit name | Bit function | Type | Read/Write | Description |
|---|---|---|---|---|---|
| CR2[7] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[6] | QPI | Quad SPI enable | V | R/W | 1 = Enable QPI protocol<br>0 = Enable SPI protocol, if DPI bit is set to '0' |
| CR2[5] | IO3R | IO3 reset | V | R/W | 1 = I/O3 is used as $\overline{\text{RESET}}$ input when $\overline{\text{CS}}$ is HIGH<br>0 = I/O3 has no alternate function |
| CR2[4] | DPI | Dual SPI enable | V | R/W | 1 = Enable DPI protocol<br>0 = Enable SPI protocol, if QPI bit is set to '0' |
| CR2[3] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR2[0] | RFU | Reserved (0) | | | Reserved for future use |

NV - Nonvolatile

### 4.2.2.1 Quad SPI (QPI) CR2 [6]

This bit controls the instruction and data widths in Quad SPI mode. In this mode, all transfers between the host system and memory are 4 bits wide on I/O0 to I/O3, including all instructions. The QUAD bit set '1' in CR1 [1] is not necessary, hence ignored for the QPI mode. Refer to **Table 22** for details.

### 4.2.2.2 IO3 Reset (IO3R) CR2 [5]

This bit controls the $\overline{\text{RESET}}$ / (I/O3) pin behavior. When this bit is set '1', enables the $\overline{\text{RESET}}$ input during normal operation. **Table 21** shows the $\overline{\text{RESET}}$ / (I/O3) functionality based on the interface mode.

### 4.2.2.3 Dual (DPI) CR2 [4]

This bit controls the instruction and data widths in Dual SPI mode. In this mode, all transfers between the host system and memory are 2 bits wide on I/O0 to I/O1, including all instructions. Refer to **Table 22** for details.

**Table 21 $\overline{\text{RESET}}$ / (I/O3) pin function**

| Interface mode | Quad bit (CR1[7]) | RESET / (I/O3) pin function | | | |
|---|---|---|---|---|---|
| | | IO3R (CR2[5]) = 0 (IO3 reset disable) | | IO3R (CR2[5]) = 1 (IO3 reset enable) | |
| | | $\overline{\text{CS}}$ = 0 | $\overline{\text{CS}}$ = 1 | $\overline{\text{CS}}$ = 0 | $\overline{\text{CS}}$ = 1 |
| SPI | QUAD = '0' | No function | No function | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ |
| SPI | QUAD = '1' | I/O3[8] | No function | I/O3[8] | $\overline{\text{RESET}}$ |
| DPI | QUAD = '0' | No function | No function | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ |
| DPI | QUAD = '1' | No function | No function | No function | $\overline{\text{RESET}}$ |
| QPI | QUAD = x (Don't care) | I/O3 | No function | I/O3 | $\overline{\text{RESET}}$ |

**Notes**
7. All Extended SPIs start in the SPI mode.
8. No function in SPI and DPI modes. I/O3 in Quad data or Quad I/O mode.

**Table 22 SPI operation modes setting**

| QUAD[10] CR1[9] | DPI CR2[9] | QPI CR2[9] | Operational mode |
|---|---|---|---|
| 0 | 0 | 0 | SPI, extended SPI (dual) |
| 1 | 0 | 0 | SPI, extended SPI (dual/quad) |
| X | 1 | 0 | DPI |
| X | 0 | 1 | QPI |
| 0 | 1 | 1 | SPI[11], extended SPI (dual) – not a recommended configuration |
| 1 | 1 | 1 | SPI[11], extended SPI (dual/quad) - not a recommended configuration |

**Notes**
9. All Extended SPIs start in the SPI mode.
10. QUAD = '1' reconfigures I/O to QUAD mode and affects $\overline{\text{WP}}$ and $\overline{\text{RESET}}$ operations (see **Table 21**).
11. Register reads will always return what is written to them, even though not a recommended configuration.

### 4.2.3 Configuration register 4 (CR4)

The configuration register 4 (CR4), as shown in **Table 23**, controls the output drive impedance and the deep-power-down (DPD) mode settings. The CR4 is accessible by the WRAR command for write and the RDCR4 or the RDAR command for read operations. The CR4 access details are provided in **"Register access commands"** on page 34.

WRAR nonvolatile write address - 0x000005

WRAR volatile write address - 0x070005

RDAR read address - 0x000005 or 0x070005

The default state shown after each bit in **Table 23** is the factory programmed value.

**Table 23 Configuration register 4 (CR4)**

| CR4[7] | CR4[6] | CR4[5] | CR4[4] | CR4[3] | CR4[2] | CR4[1] | CR4[0] |
|---|---|---|---|---|---|---|---|
| OI (0) | OI (0) | OI (0) | RFU (0) | RFU (1) | DPDPOR (0) | RFU (0) | RFU (0) |

**Table 24 Configuration register 4 (CR4) - nonvolatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR4[7] | OI | Output impedance | NV | R/W | Output impedance selection |
| CR4[6] | | | NV | R/W | |
| CR4[5] | | | NV | R/W | |
| CR4[4] | RFU | Reserved (0) | | | Reserved for future use |
| CR4[3] | RFU | Reserved (1) | | | Reserved for future use[12] |
| CR4[2] | DPDPOR | Deep power-down mode on POR | NV | R/W | 1 = Deep power-down is entered upon completion of POR or hardware reset (including JEDEC reset) when $\overline{CS}$ is HIGH. 0 = Standby mode is entered upon completion of power-up or POR or hardware reset (including JEDEC reset) when $\overline{CS}$ is HIGH. |
| CR4[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR4[0] | RFU | Reserved (0) | | | Reserved for future use |

NV - Nonvolatile

**Note**

12. The SPI bus master must make sure bit CR4 [3] remains '1' when writing to this configuration register. Writing a '0' to this bit may impact device functionality.

**Table 25 Configuration register 4 (CR4) - volatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR4[7] | OI | Output impedance | V | R/W | Output impedance selection |
| CR4[6] | | | V | R/W | |
| CR4[5] | | | V | R/W | |
| CR4[4] | RFU | Reserved (0) | | | Reserved for future use |
| CR4[3] | RFU | Reserved (1) | | | Reserved for future use[13] |
| CR4[2] | DPDPOR | Deep power-down mode on POR | V | R/W | 1 = deep power-down is entered upon completion of POR or hardware reset (including JEDEC reset) when $\overline{CS}$ is HIGH. 0 = standby mode is entered upon completion of power-up or POR or hardware reset (including JEDEC reset) when $\overline{CS}$ is HIGH. |
| CR4[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR4[0] | RFU | Reserved (0) | | | Reserved for future use |

**Note**
13. The SPI bus master must make sure bit CR4 [3] remains '1' when writing to this configuration register. Writing a '0' to this bit may impact device functionality.

### 4.2.3.1 Output impedance (OI) CR4 [7:5]

These three bits control the output impedance (drive strength) of the I/O pins. The output impedance configuration bits enable the user to adjust the drive strength for a better signal integrity on the printed circuit board.

**Table 26 Impedance selection**

| Impedance selection | Typical impedance (Ω)[14] | Comments |
|---|---|---|
| 000 | 45 | 45 Ω is the factory default configuration. Other drive strength can be programmed by writing into impedance selection bits in CR4[7:5]. |
| 001 | 120 | |
| 010 | 90 | |
| 011 | 60 | |
| 100 | 45 | |
| 101 | 30 | |
| 110 | 20 | |
| 111 | | |

**Note**
14. Typical impedance measured at $V_{DD}/2$.

### 4.2.3.2 Deep-power-down mode on POR (DPDPOR) CR4 [2]

This bit controls whether the device enters the deep-power-down (DPD) or the standby mode after the completion of power-on-reset (POR), hardware reset ($\overline{RESET}$ pin or JEDEC reset), or exit the hibernate mode. The DPDPOR configuration bit enables the device to start in DPD mode, instead of standby mode when $\overline{CS}$ is HIGH. A $\overline{CS}$ pulse-width of $t_{CSDPD}$, or hardware reset will exit the DPD mode after $t_{EXTDPD}$ time. The $\overline{CS}$ pulse-width can be generated by toggling $\overline{CS}$ alone while SCK and I/Os are don't care. The DPDPOR bit status is ignored during the software reset and the device always enters standby after the software reset.

### 4.2.4 Configuration register 5 (CR5)

The configuration register 5 (CR5), as shown in **Table 27**, configures the read latency (dummy) cycles for register read. The CR5 is accessible by the WRAR command for write and the RDCR5 or the RDAR command for read operations. The CR5 access details are provided in **"Register access commands"** on page 34.

WRAR nonvolatile write address - 0x000006

WRAR volatile write address - 0x070006

RDAR read address - 0x000006 or 0x070006

The default state shown after each bit in **Table 27** is the factory programmed value.

**Table 27 Configuration register 5 (CR5)**

| CR5[7] | CR5[6] | CR5[5] | CR5[4] | CR5[3] | CR5[2] | CR5[1] | CR5[0] |
|---|---|---|---|---|---|---|---|
| RLC1 (0) | RLC0 (0) | RFU (0) | RFU (0) | RFU (0) | RFU (0) | RFU (0) | RFU (0) |

**Table 28 Configuration register 5 (CR5) - nonvolatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR5[7] | RLC1 | Register latency code | NV | R/W | Selects number of register read latency cycles between 0 to 3 clock cycles for register accesses |
| CR5[6] | RLC0 | | | R/W | |
| CR5[5] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[4] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[3] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[0] | RFU | Reserved (0) | | | Reserved for future use |

NV - Nonvolatile

**Table 29 Configuration register 5 (CR5) - volatile**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| CR5[7] | RLC1 | Register latency code | V | R/W | Selects number of register read latency cycles between 0 to 3 clock cycles for register accesses |
| CR5[6] | RLC0 | | | R/W | |
| CR5[5] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[4] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[3] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[2] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[1] | RFU | Reserved (0) | | | Reserved for future use |
| CR5[0] | RFU | Reserved (0) | | | Reserved for future use |

V - Volatile

### 4.2.4.1 Register latency code (RLC [1:0]) CR5 [7:6]

These two bits control the read latency (dummy cycle) delay in all variable latency register read instructions. It enables users to adjust the read latency during normal operation to optimize the latency for different register read instructions at different operating frequencies. **Table 30** shows latency cycles for register read command.

**Table 30 Dummy cycles for register read commands**

| Latency (dummy cycles) | SPI (SDR) | DPI (SDR) | QPI (SDR) |
|---|---|---|---|
| | RDSR1, RDSR2, RDCR1, RDCR2, RDCR4, RDCR5, RDAR, RUID, RDID2, RDSN | | |
| 0 | 50 MHz[15] | 50 MHz[15] | 50 MHz[15] |
| 1–3 | 108 MHz | 108 MHz | 108 MHz |

**Note**
15.This parameter is guaranteed by characterization; not tested in production.

# 5 Functional description

The CY15X102QSN has an 8-bit instruction register. All instructions and their opcodes are listed in the following. All instructions, addresses, and data are transferred with a HIGH to LOW $\overline{CS}$ transition. Furthermore, the $\overline{WP}$ and $\overline{RESET}$ pins provide additional hardware controlled functions.

## 5.1 Command structure

The CY15X102QSN command cycle consists of up to five different command phases - opcode, address, mode, dummy (latency), and data. The number of command phases per command cycle varies from one to five depending on the opcode sent in the opcode phase. The opcode, address, mode, and data phases are configurable in terms of number of lines 1, 2, or 4 needed to transmit them in SPI, DPI, or QPI interface, respectively. **Table 31** shows the command phases for each command cycle in different SPI interfaces.

**Table 31 Command transmission over I/Os in different SPI modes**

| Command phases | Command transmission on I/Os | | | | | | |
|---|---|---|---|---|---|---|---|
| | Single channel SPI | Extended SPI | | | | Multi-channel SPI | |
| | | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI |
| **Opcode** | SI | I/O0 | I/O0 | I/O0 | I/O0 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |
| **Address** | SI | I/O0 | I/O0 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |
| **Mode** | SI | I/O0 | I/O0 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |
| **Dummy (latency)** | Fixed number of dummy SPI clocks, independent of SPI interface.<br>0 to 15 clocks for memory access (configurable via CR1[7:4])<br>0 to 3 clocks for register access (configurable via CR5[7:6]) | | | | | | |
| **Data** | SI/SO | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 | I/O0, I/O1 | I/O0, I/O1, I/O2, I/O3 |

There are 44 commands, called opcodes that can be issued by the bus master to the CY15X102QSN as shown in **Table 32**. These opcodes control the functions performed by the memory.

**Table 32 Opcode Commands**

| Command | | SPI bus interface | | | | | | | Data transfer | | Latency | | XIP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Command** | **Opcode (HEX)** | **SPI** | **Dual data** | **Quad data** | **Dual I/O** | **Quad I/O** | **DPI** | **QPI** | **SDR** | **DDR** | **Register latency** | **Memory latency** | **Execute -in-place** |
| **Write enable control** | | | | | | | | | | | | | |
| WREN | 06 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| WRDI | 04 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| **Register access** | | | | | | | | | | | | | |
| WRSR | 01 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| RDSR1 | 05 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDSR2 | 07 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDCR1 | 35 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDCR2 | 3F | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDCR4 | 45 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDCR5 | 5E | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| WRAR | 71 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |

**Table 32 Opcode Commands** (continued)

| Command | | SPI bus interface | | | | | | | Data transfer | | Latency | | XIP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Command** | **Opcode (HEX)** | **SPI** | **Dual data** | **Quad data** | **Dual I/O** | **Quad I/O** | **DPI** | **QPI** | **SDR** | **DDR** | **Register latency** | **Memory latency** | **Execute -in-place** |
| RDAR | 65 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| **Memory read** | | | | | | | | | | | | | |
| READ | 03 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | Yes | NA |
| FAST_ READ | 0B | Yes | NA | | | | Yes | Yes | Yes | NA | NA | Yes | Yes |
| DOR | 3B | NA | Yes | NA | | | | | Yes | NA | NA | Yes | Yes |
| DIOR | BB | NA | | | Yes | NA | | | Yes | NA | NA | Yes | Yes |
| QOR | 6B | NA | | Yes | NA | | | | Yes | NA | NA | Yes | Yes |
| QIOR | EB | NA | | | | Yes | NA | Yes | Yes | NA | NA | Yes | Yes |
| DDRFR | 0D | NA | | | | | | Yes | NA | Yes | NA | Yes | Yes |
| DDRQIOR | ED | NA | | | | Yes | NA | Yes | NA | Yes | NA | Yes | Yes |
| **Memory write** | | | | | | | | | | | | | |
| WRITE | 02 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| FAST_ WRITE | DA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | Yes |
| DIW | A2 | NA | Yes | NA | | | | | Yes | NA | NA | NA | Yes |
| DIOW | A1 | NA | | | Yes | NA | | | Yes | NA | NA | NA | Yes |
| QIW | 32 | NA | | Yes | NA | | | | Yes | NA | NA | NA | Yes |
| QIOW | D2 | NA | | | | Yes | NA | | Yes | NA | NA | NA | Yes |
| DDR_FAST _WRITE | DD | NA | | | | | | Yes | NA | Yes | NA | NA | Yes |
| DDRWRITE | DE | NA | | | | | | Yes | NA | Yes | NA | NA | NA |
| DDRQIOW | D1 | NA | | | | Yes | NA | | NA | Yes | NA | NA | Yes |
| **Special sector memory access** | | | | | | | | | | | | | |
| SSWR | 42 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| SSRD | 4B | Yes | NA | | | | Yes | Yes | Yes | NA | NA | Yes | NA |
| **ECC and CRC** | | | | | | | | | | | | | |
| CLECC | 1B | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| ECCRD | 19 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | Yes | NA |
| CRCC | 5B | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| EPCS | 75 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| EPCR | 7A | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| **Identification & serial number** | | | | | | | | | | | | | |
| RUID | 4C | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDID | 9F | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| WRSN | C2 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |
| RDSN | C3 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | NA |

**Table 32 Opcode Commands** (continued)

| Command | | SPI bus interface | | | | | | | Data transfer | | Latency | | XIP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Command** | **Opcode (HEX)** | **SPI** | **Dual data** | **Quad data** | **Dual I/O** | **Quad I/O** | **DPI** | **QPI** | **SDR** | **DDR** | **Register latency** | **Memory latency** | **Execute -in-place** |
| **Power modes and reset** | | | | | | | | | | | | | |
| DPD | B9 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| HBN | BA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| RSTEN | 66 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |
| RST | 99 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | NA |

## 5.1.1 Write enable control commands

These commands set or clear the write enable latch bit in the status register 1 (SR1[1]).

**Table 33 Write enable control commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| WREN | 06 | Write enable – sets the WEL bit of status register 1 to '1' |
| WRDI | 04 | Write disable – clears the WEL bit of status register 1 to '0' |

**Table 34 Write enable control command details**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | XIP | Latency | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **SPI** | **Dual data** | **Quad data** | **Dual I/O** | **Quad IO** | **DPI** | **QPI** | **SDR** | **DDR** | **Execute -in-place** | **Dummy cycles** | |
| 06 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 04 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |

### 5.1.1.1 Set write enable latch (WREN, 06h)

The WREN command sets the WEL bit of status register 1 (SR1[1]) to a '1'. CY15X102QSN requires WEL bit set to a '1' prior to issuing any write command. The CY15X102QSN commands requiring WEL set to '1' prior to their execution are WRSR, WRAR, WRITE, FAST_WRITE, DIW, DIOW, QIW, QIOW, DDR_FAST_WRITE, DDRWRITE, DDRQIOW, SSWR, and WRSN.

$\overline{CS}$ must be driven to the logic HIGH state after the eighth bit of the instruction byte has been latched in on SI. CY15X102QSN executes the WREN command and sets the WEL bit (SR1[1]) to '1' after $\overline{CS}$ is driven HIGH after 8-bit WREN opcode is successfully latched in.

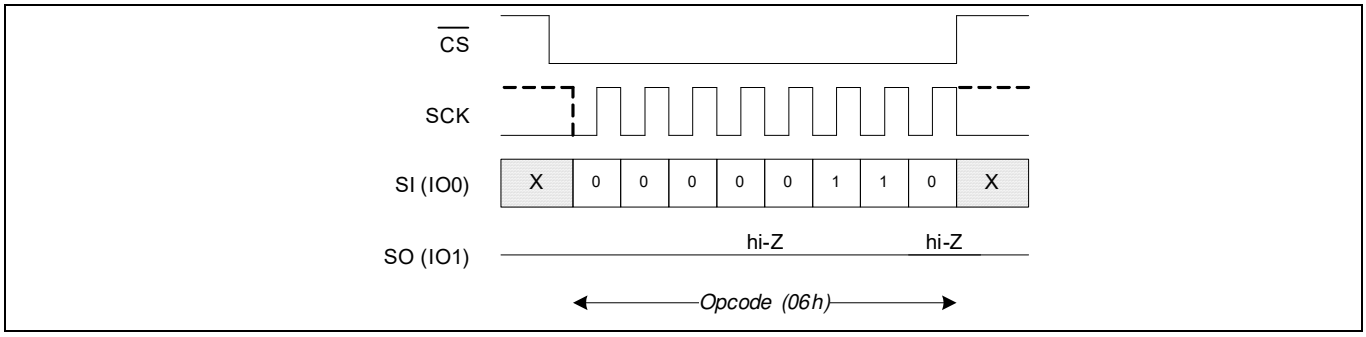



**Figure 6 WREN bus configuration in SPI mode**

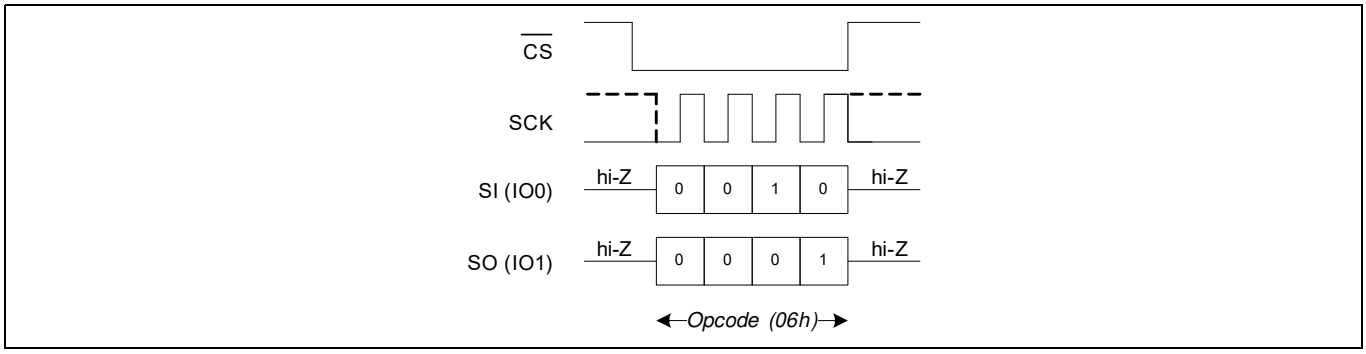



**Figure 7 WREN bus configuration in DPI mode**

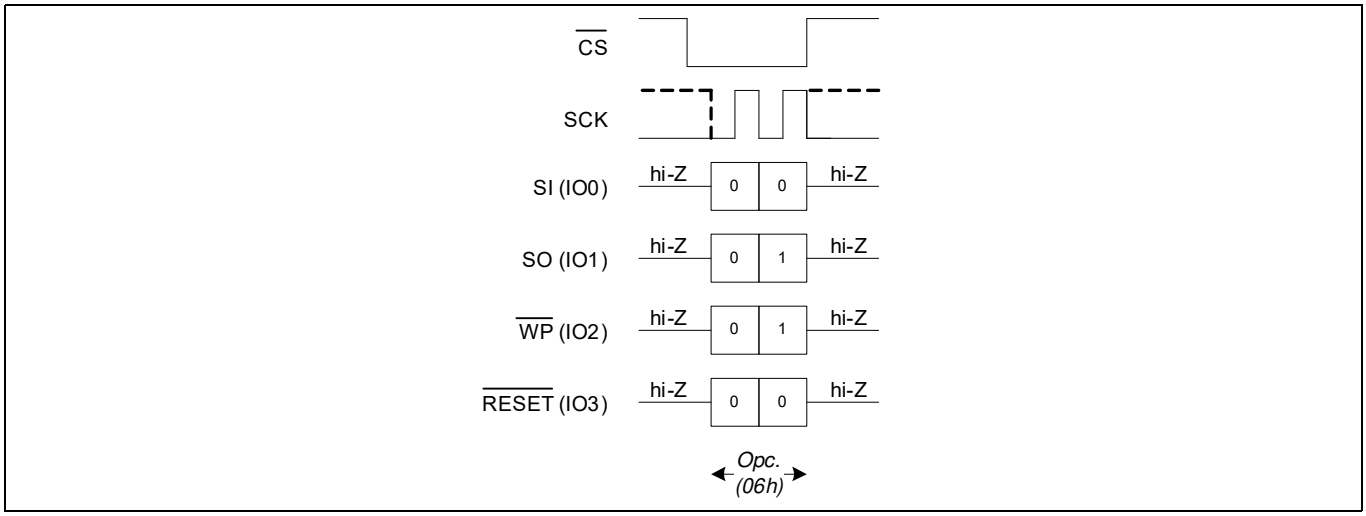



**Figure 8 WREN bus configuration in QPI mode**

### 5.1.1.2 Reset write enable latch (WRDI, 04h)

The WRDI instruction clears the write enable latch (WEL) bit of the status register 1 (SR1[1]) to a '0'. This disables write status register (WRSR), write any register (WRAR), special sector write (SSWR), and other instructions that require WEL to be set to '1' prior to the execution. The WRDI instruction can be used to protect the memory and the SPI registers against inadvertent writes. The WRDI command is ignored during an embedded operation while WIP bit = '1'.

$\overline{CS}$ must be driven to the logic HIGH state after the eighth bit of the instruction byte has been latched in on SI. CY15X102QSN executes the WRDI command and clears the WEL bit (SR1[1]) to '0' after $\overline{CS}$ is driven HIGH after 8-bit WRDI opcode is successfully latched in.

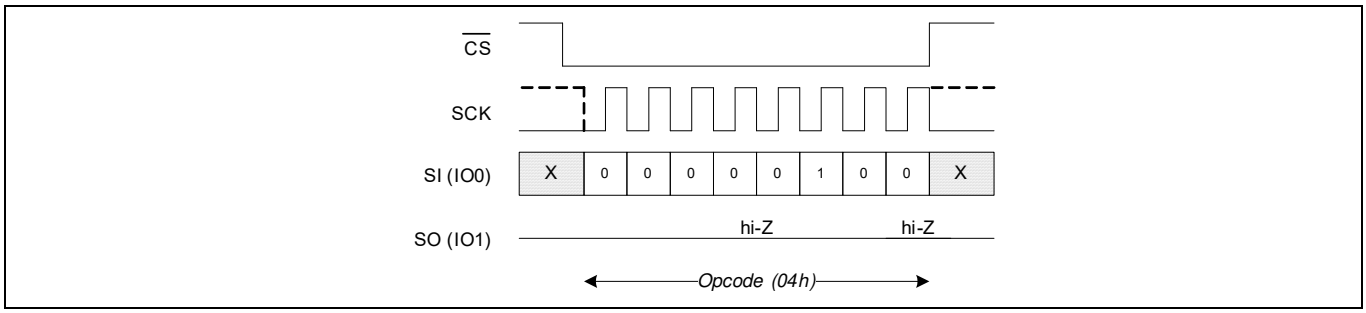



**Figure 9 WRDI bus configuration in SPI mode**

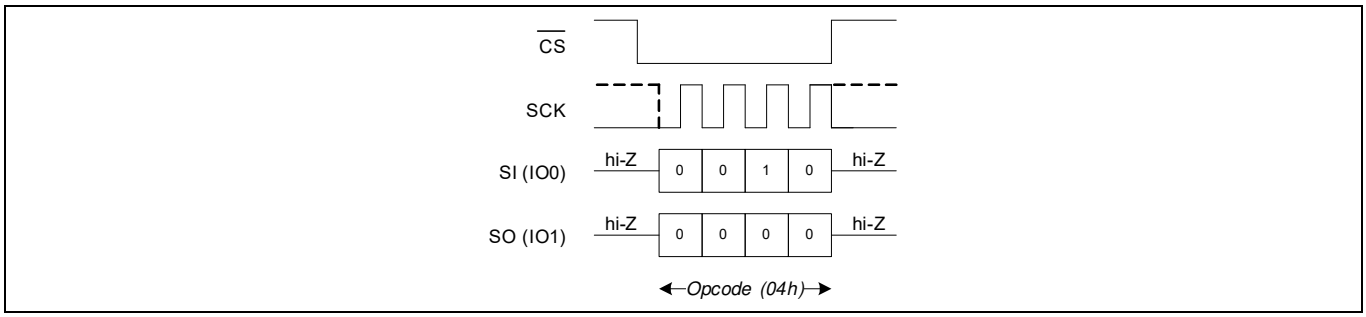



**Figure 10 WRDI bus configuration in DPI mode**

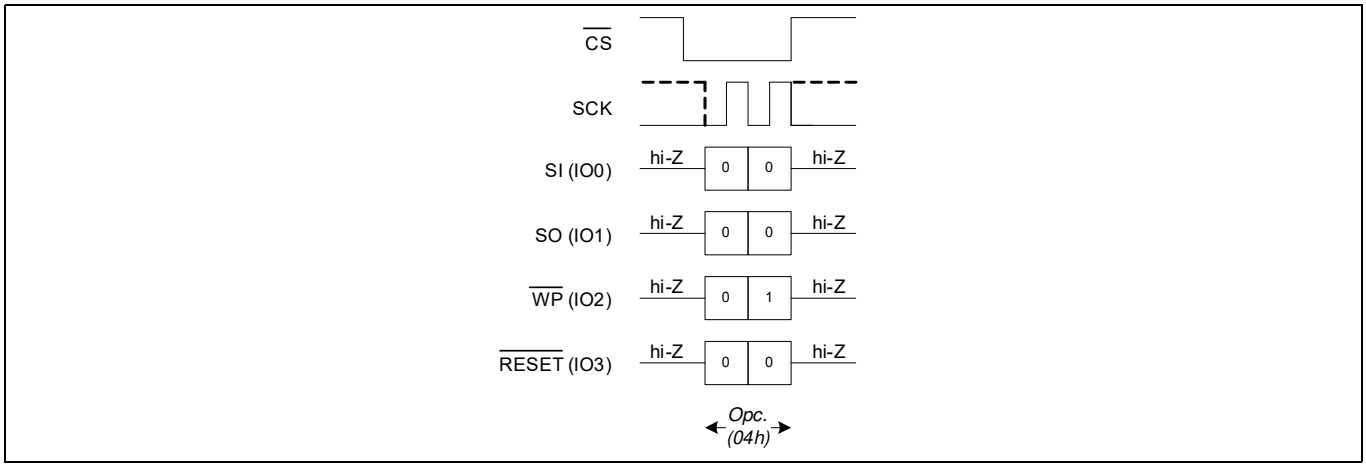



**Figure 11 WRDI bus configuration in QPI mode**

### 5.1.2 Register access commands

CY15X102QSN provides various configuration and status registers. These registers are user-writable, which can be programmed to enable or disable certain configurations/features in the part as well as can be polled to know the device status. These registers are accessed by specific commands, called opcodes.

The individual register bits can be one of multiple types: write/read, read only, or reserved for future use (RFU). The specific type of each bit is specified in their respective register section. Register bits can be either volatile or nonvolatile in nature. All volatile (V) bits are set to their default values after power-on reset (POR), or any reset event (via hardware or software resets); while all nonvolatile (NV) bits resume to user configured values after power-on reset (POR), or any reset event (via hardware or software resets).

**Table 35 Register access commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| WRSR | 01 | Write status register 1 |
| RDSR1 | 05 | Read status register 1 |
| RDSR2 | 07 | Read status register 2 |
| RDCR1 | 35 | Read configuration register 1 |
| RDCR2 | 3F | Read configuration register 2 |
| RDCR4 | 45 | Read configuration register 4 |
| RDCR5 | 5E | Read configuration register 5 |
| WRAR | 71 | Write any register - including status registers, configurations registers, serial number registers |
| RDAR | 65 | Read any register - including status registers, configurations registers, CRC registers, ECC registers, serial number registers, and ID registers |

**Table 36 Register access command details**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | Register latency | Max clock frequency | Register latency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Dummy cycle | | |
| 01 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | NA | 108 MHz | NA |
| 05 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 07 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 35 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 3F | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 45 | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 5E | 0 | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |
| 71 | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | NA | 108 MHz | NA |
| 65 | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz | Yes |

### 5.1.2.1 Write status register (WRSR, 01h)

The write status register (WRSR) instruction allows new values to be programmed in status register 1 (SR1). This instruction writes to the nonvolatile SR1, thus survives the power cycle. The WRSR command is ignored when the SRWD bit in SR1 (SR1[7]) is set '1' and the $\overline{WP}$ pin is asserted LOW.

**Notes**

- The WRSR instruction executes only when WEL bit in SR1 is set to '1'; otherwise, the WRSR instruction will be ignored.
- The WEL bit of the status register 1 (SR1[1]) is automatically cleared to '0' after WRSR command is terminated (at the rising edge of $\overline{CS}$).

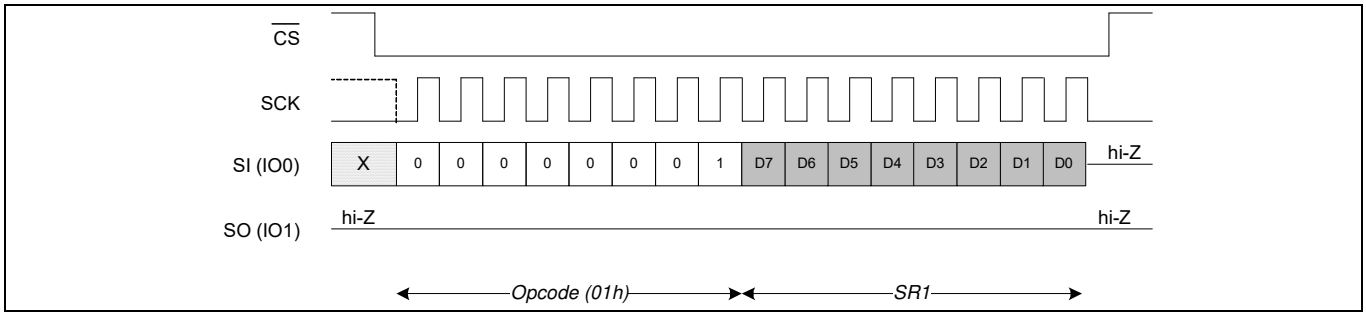



**Figure 12 WRSR in SPI mode (WREN not shown)**

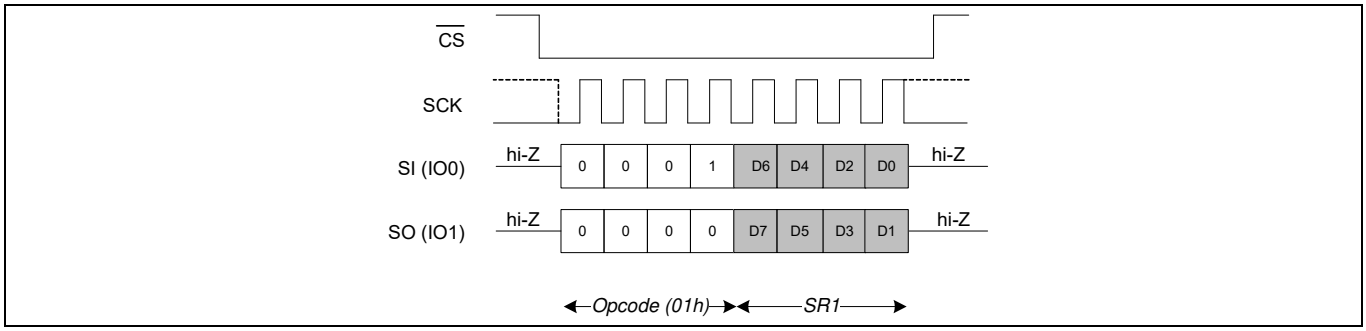



**Figure 13 WRSR in DPI mode (WREN not shown)**

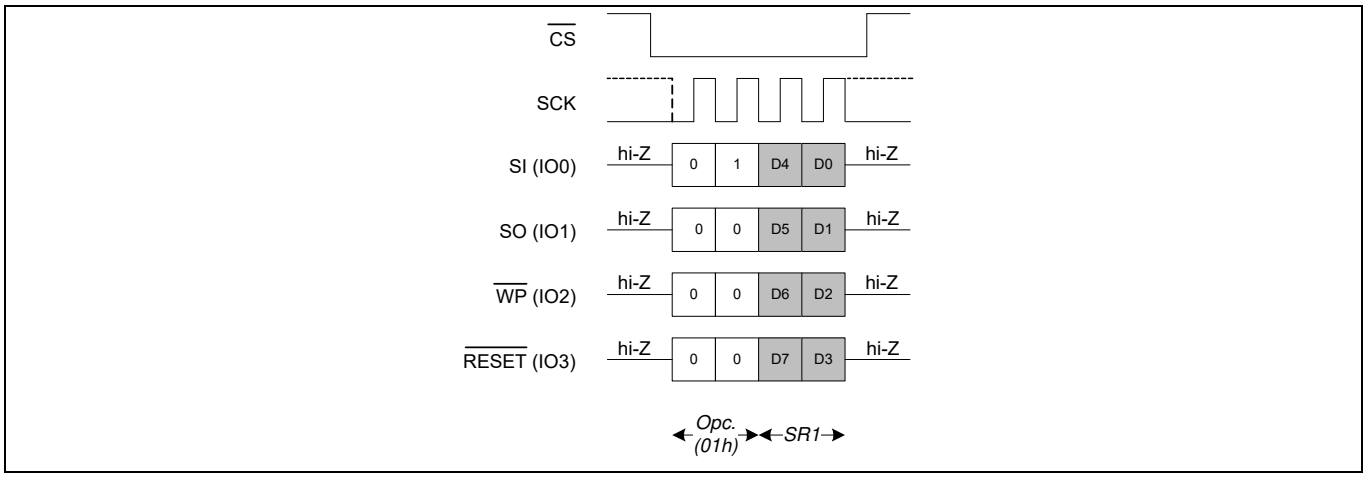



**Figure 14 WRSR in QPI mode (WREN not shown)**

### 5.1.2.2 Read status register 1 (RDSR1, 05h)

The RDSR1 command allows the bus master to verify the contents of the status register 1 (SR1). Reading SR1 provides information about the current state of the write-protection features, WEL, and WIP status. Following the RDSR1 opcode, the CY15X102QSN will return one byte SR1 content.

**Notes**

- The RDSR1 returns the volatile content of SR1.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

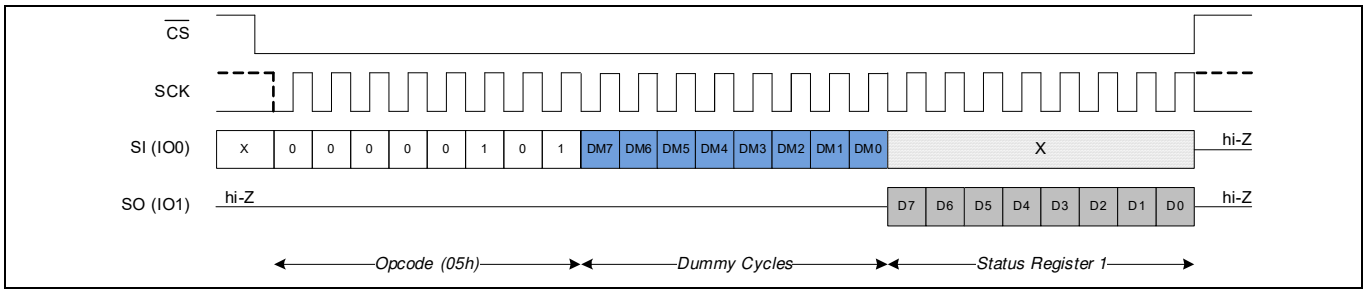



**Figure 15 Read SR1 (RDSR1) in SPI mode**

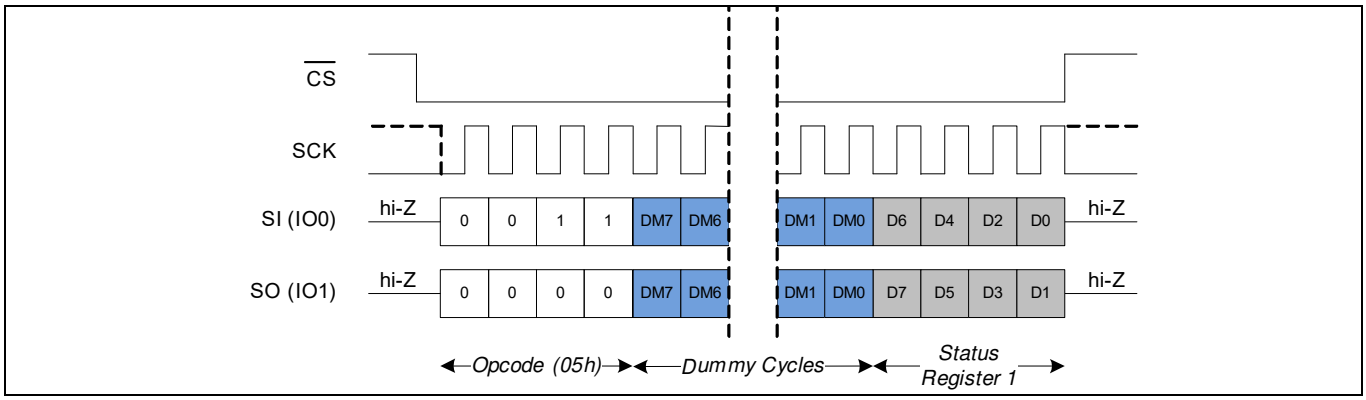



**Figure 16 Read SR1 (RDSR1) in DPI mode**

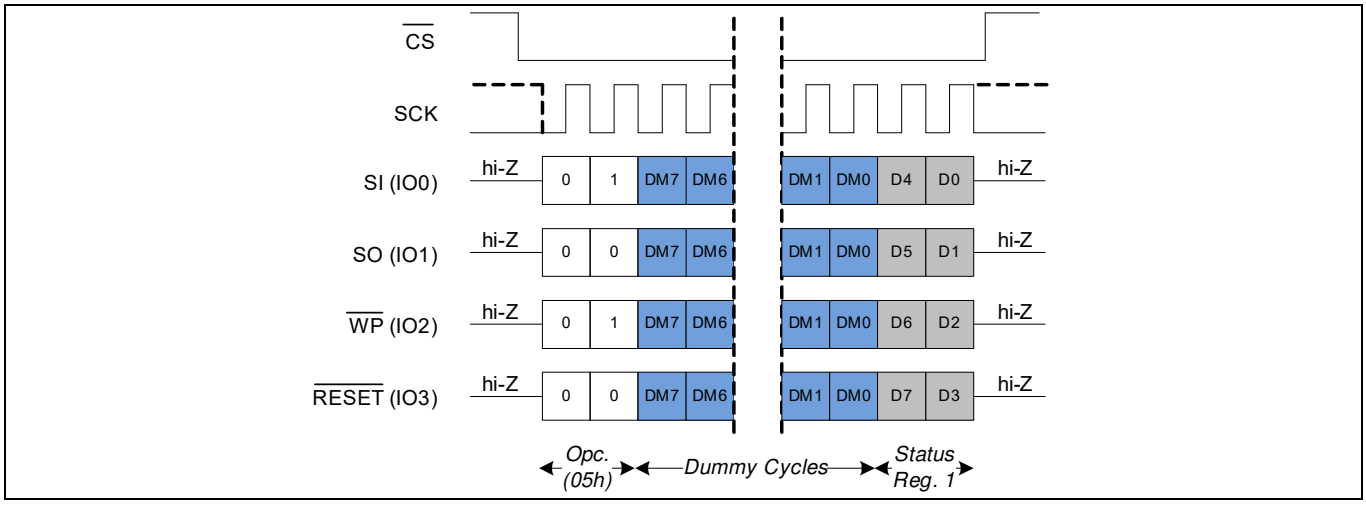



**Figure 17 Read SR1 (RDSR1) in QPI mode**

### 5.1.2.3 Read status register 2 (RDSR2, 07h)

The RDSR2 command allows the bus master to verify the contents of the status register 2 (SR2). This is a read only register and provides information about the CRC suspend and CRC abort status. The SR2 bits indicate the correct status (CRCS and CRCA) only when the WIP bit of SR1 is '0'. Reading SR2 while WIP is '1' will return an undetermined status.

**Notes**

- The RDSR2 returns the volatile content of SR2.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

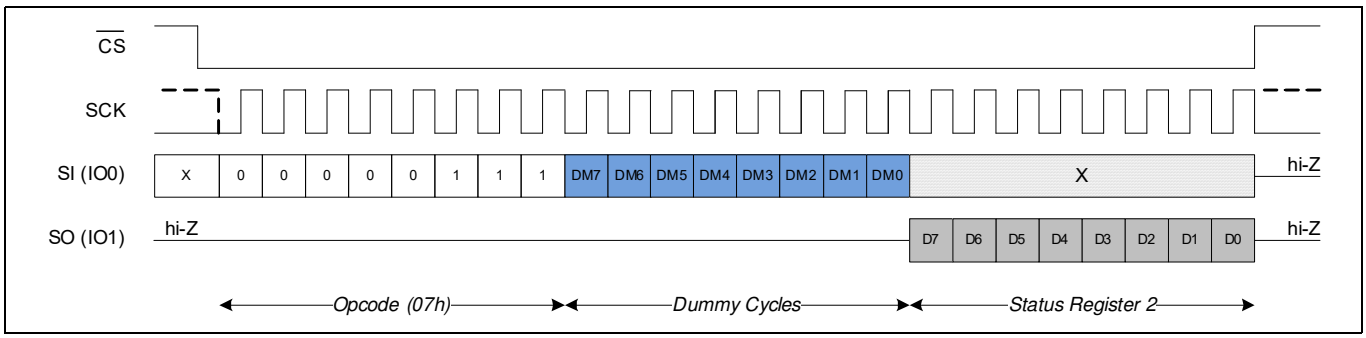



**Figure 18 Read SR2 (RDSR2) in SPI mode**

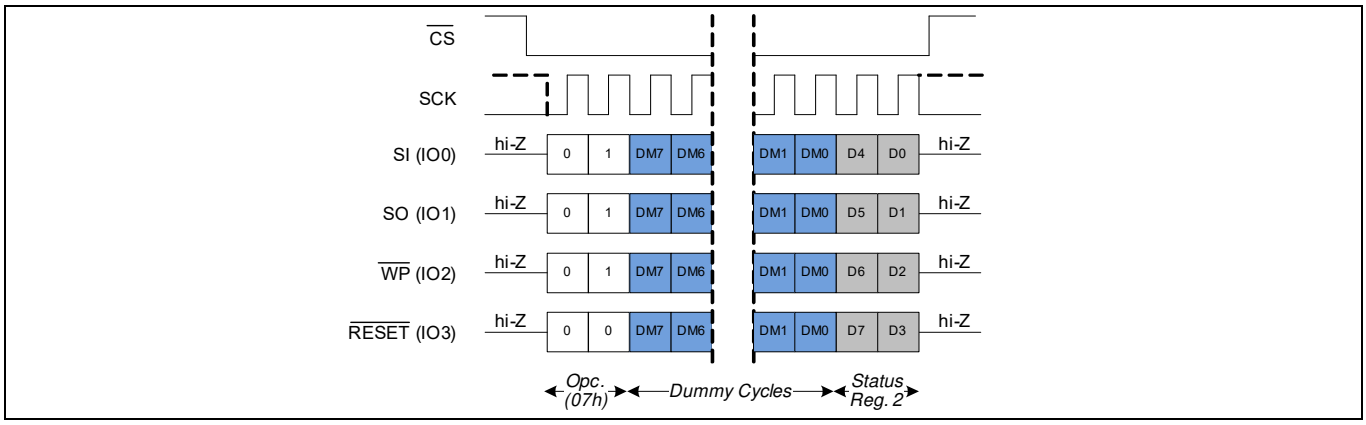



**Figure 19 Read SR2 (RDSR2) in DPI mode**

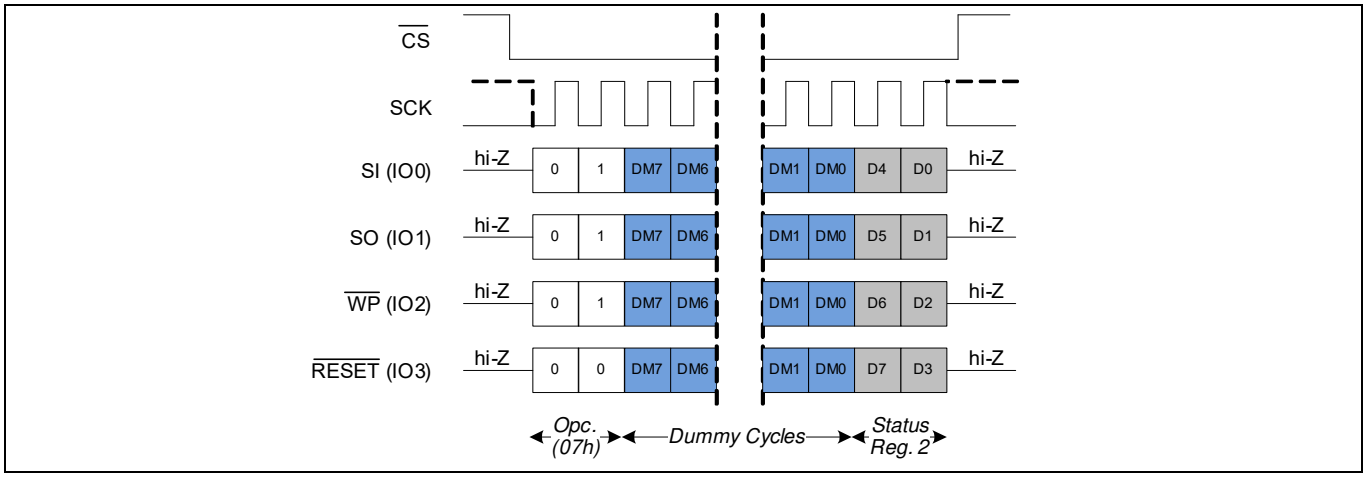



**Figure 20 Read SR2 (RDSR2) in QPI mode**

### 5.1.2.4 Read configuration register 1 (RDCR1, 35h)

The RDCR1 command allows the bus master to verify the contents of the configuration register 1 (CR1). Reading CR1 provides information about the current state of the memory latency code and QUAD bit status. Following the RDCR1 opcode, CY15X102QSN will return one byte content of CR1.

**Notes**

- The RDCR1 returns the volatile content of CR1.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

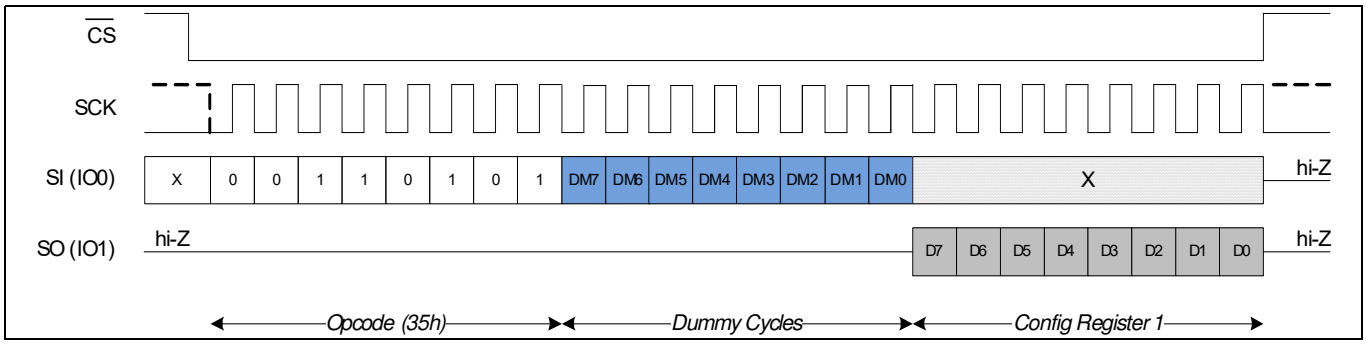



**Figure 21 Read CR1 (RDCR1) in SPI mode**

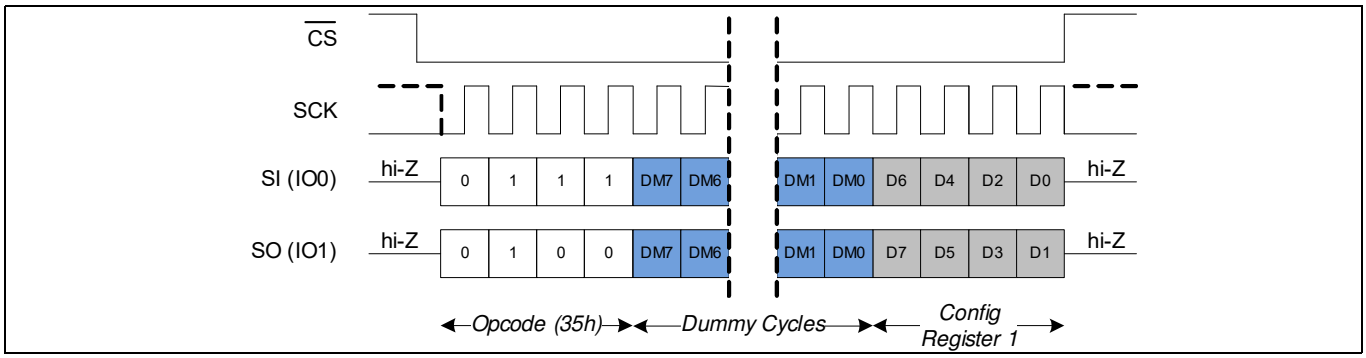



**Figure 22 Read CR1 (RDCR1) in DPI mode**

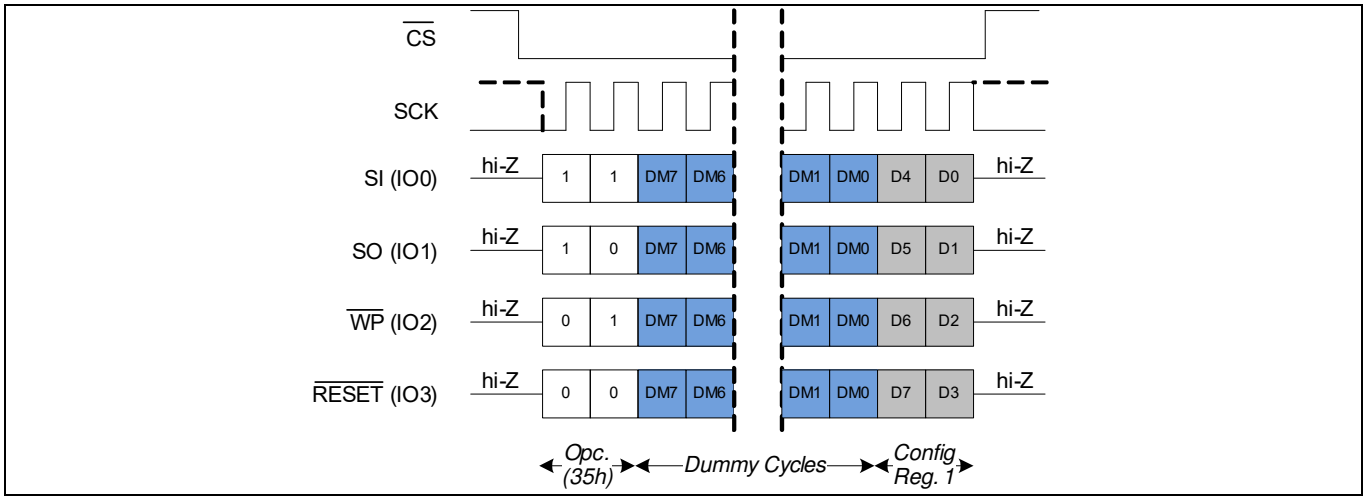



**Figure 23 Read CR1 (RDCR1) in QPI mode**

### 5.1.2.5 Read configuration register 2 (RDCR2, 3Fh)

The RDCR2 command allows the bus master to verify the contents of the configuration register 2 (CR2). Reading CR2 provides information about the current SPI interface option (SPI vs DPI vs QPI) and RESET / (I/O3) status. Following the RDCR2 opcode, the CY15X102QSN will return one byte content of CR2.

**Notes**

- The RDCR2 returns the volatile content of CR2.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

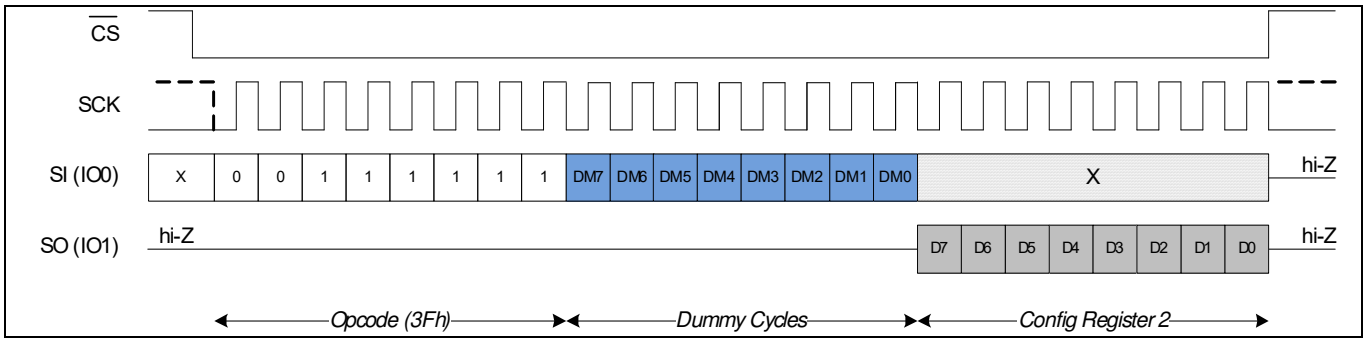



**Figure 24 Read CR2 (RDCR2) in SPI mode**

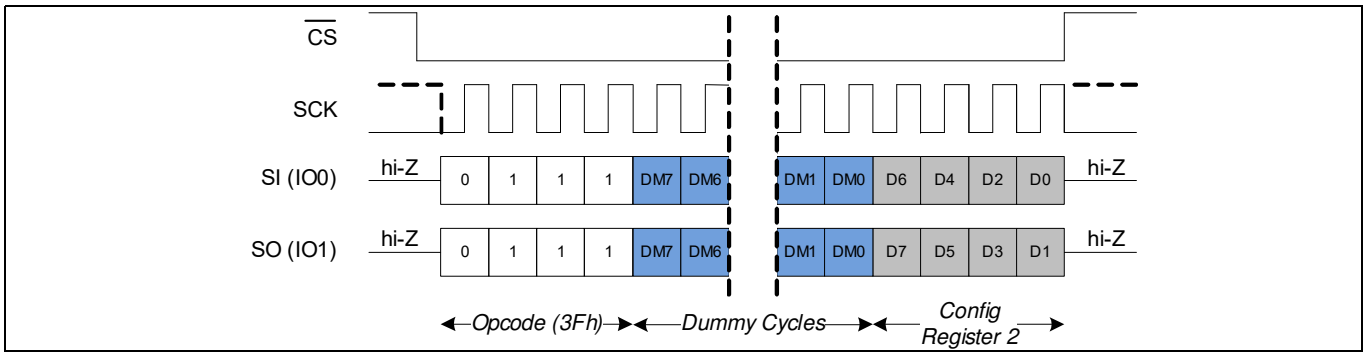



**Figure 25 Read CR2 (RDCR2) in DPI mode**

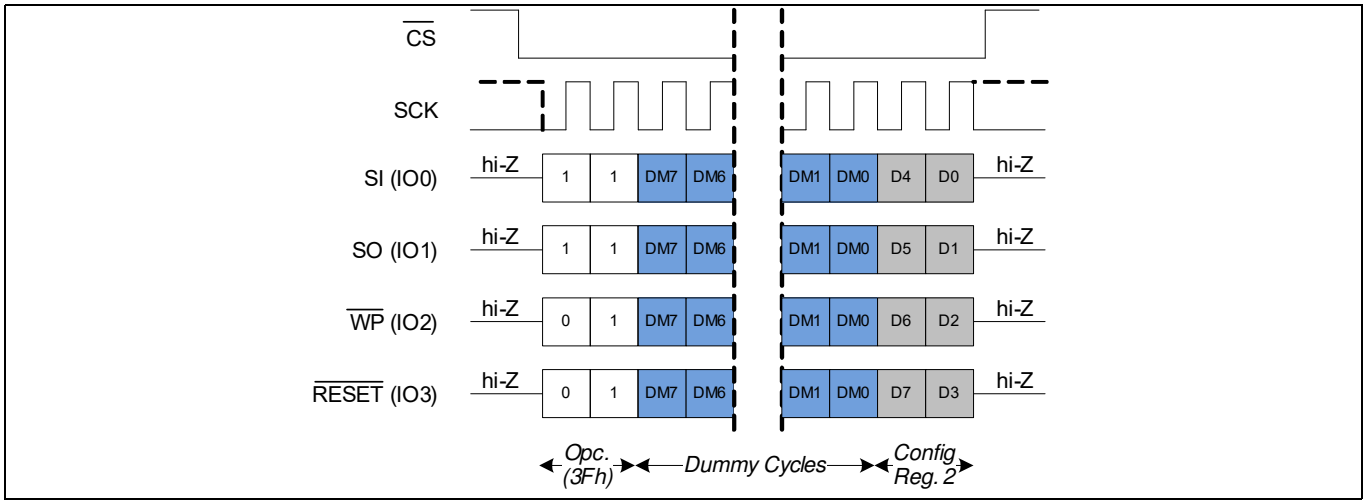



**Figure 26 Read CR2 (RDCR2) in QPI mode**

### 5.1.2.6 Read configuration register 4 (RDCR4, 45h)

The RDCR4 command allows the bus master to verify the contents of the configuration register 4 (CR4). Reading CR4 provides information about the output impedance setting and device power mode status after POR (deep-power-down vs standby). Following the RDCR4 opcode, the CY15X102QSN will return one byte content of CR4.

**Notes**

- The RDCR4 returns the volatile content of CR4.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

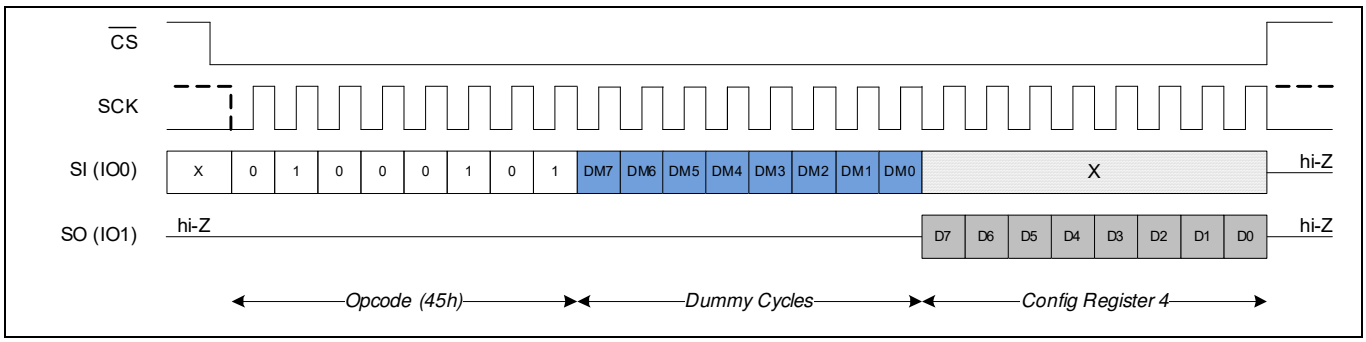



**Figure 27 Read CR4 (RDCR4) in SPI mode**

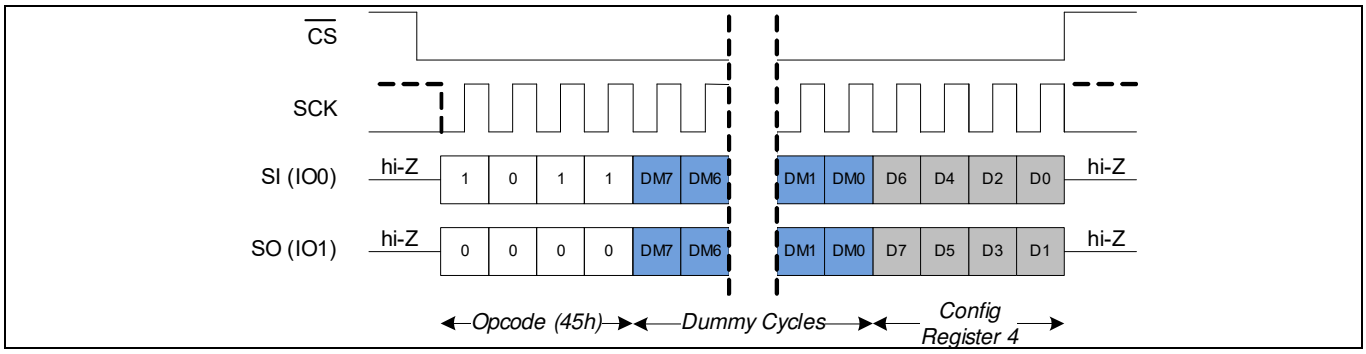



**Figure 28 Read CR4 (RDCR4) in DPI mode**

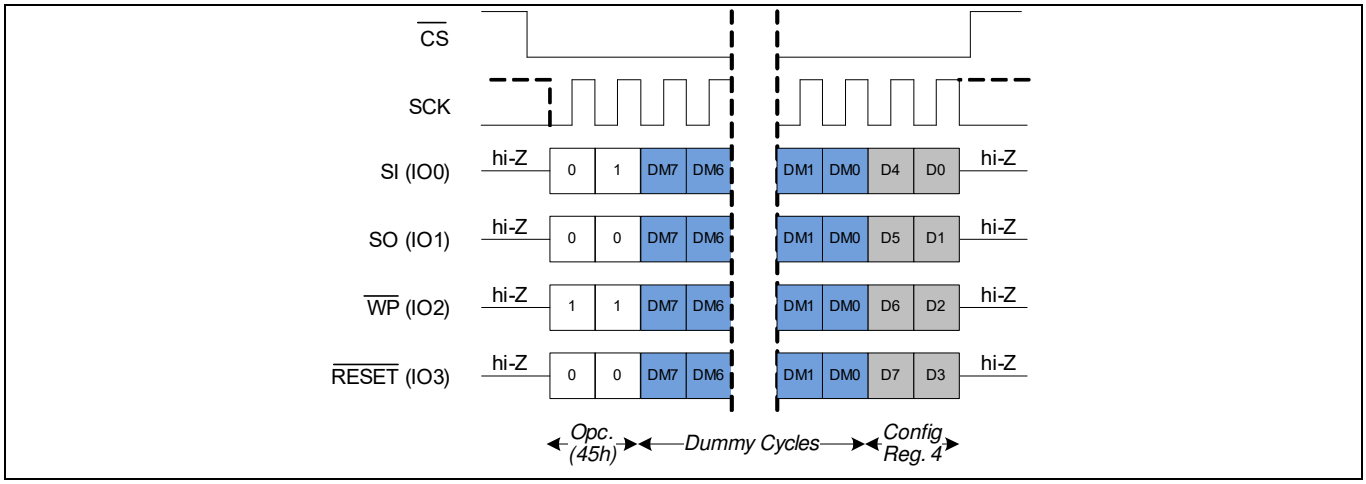



**Figure 29 Read CR4 (RDCR4) in QPI mode**

### 5.1.2.7 Read configuration register 5 (RDCR5, 5Eh)

The RDCR5 command allows the bus master to verify the contents of the configuration register 5 (CR5). Reading CR5 provides information about the register read latency cycles (RLC0, RLC1) setting. Following the RDCR5 opcode, the CY15X102QSN will return one byte content of CR5.

**Notes**

- The RDCR5 returns the volatile content of CR5.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

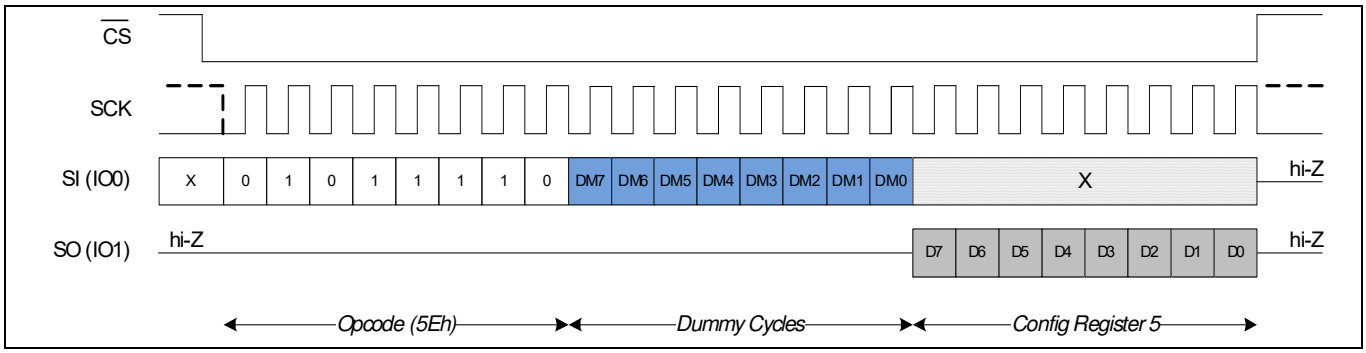



**Figure 30 Read CR5 (RDCR5) in SPI mode**

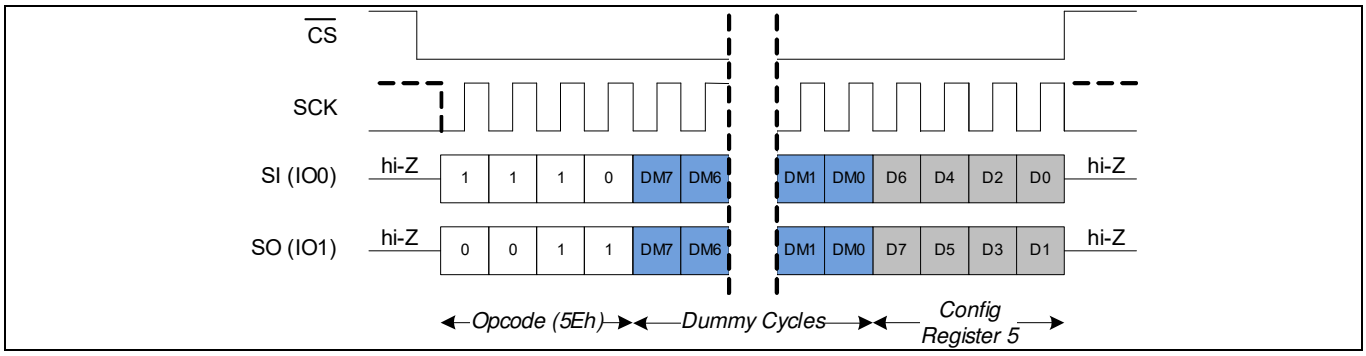



**Figure 31 Read CR5 (RDCR5) in DPI mode**

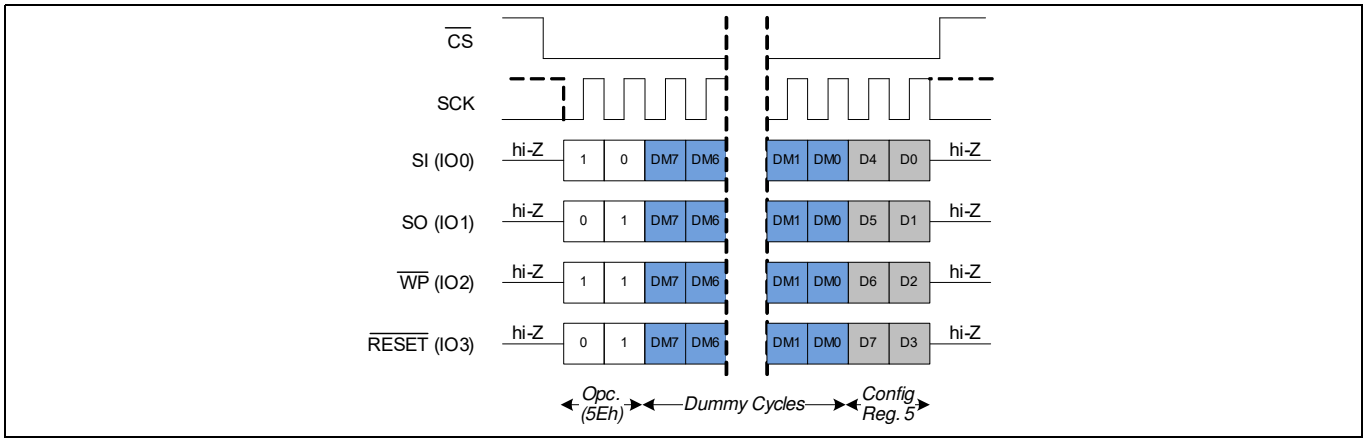



**Figure 32 Read CR5 (RDCR5) in QPI mode**

### 5.1.2.8 Write any register (WRAR, 71h)

The WRAR instruction allows writing into CY15X102QSN registers, one register at a time, addressable by their 3-byte addressing. The WRAR opcode is followed by the three-byte address of the register, as shown in **Table 38**, followed by one byte register data to be written. The WREN command precedes the WRAR command to set the WEL bit '1' prior to WRAR. The WEL bit is automatically cleared to '1' after WRAR command is terminated (at the rising edge of $\overline{CS}$). The WRAR command is ignored when the SRWD bit in SR1 (SR1[7]) is set to '1' and the $\overline{WP}$ pin is driven LOW.

**Notes**

- The WRAR command supports only one byte write per WRAR command at the given register address. The WRAR command format is shown in **Table 37**.
- The register address sent in 3-byte address field after the WRAR opcode determines whether new configuration will be programmed into the volatile status/configuration register only, or will be programmed into both volatile and nonvolatile status/configuration register. **Table 38** shows register addresses for both volatile and nonvolatile registers.

**Table 37 Registers with generic write instructions**

| Instruction name | Instruction description | Opcode | Address bytes | Data bytes |
|---|---|---|---|---|
| WRAR | Write any register | 71h | 3 | 1 |

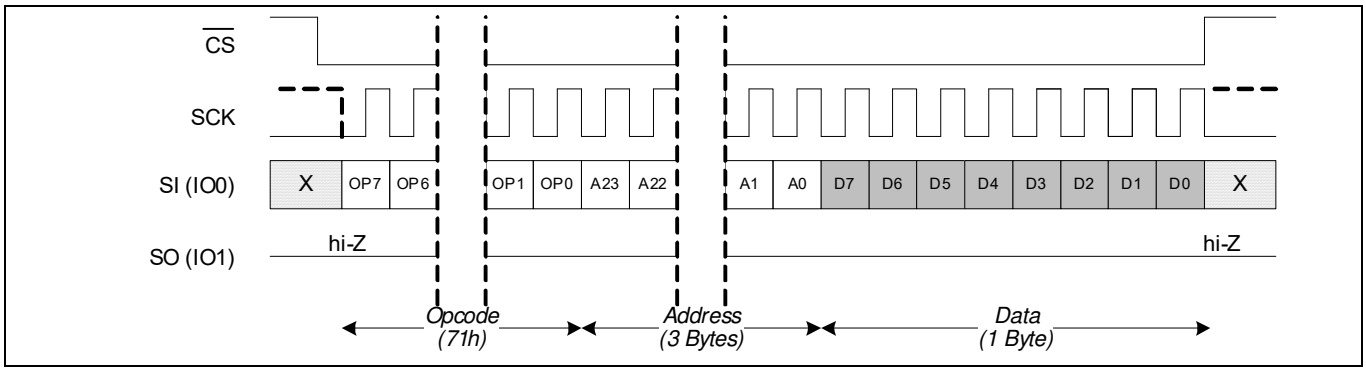



**Figure 33 Write any register (WRAR) in SPI mode**

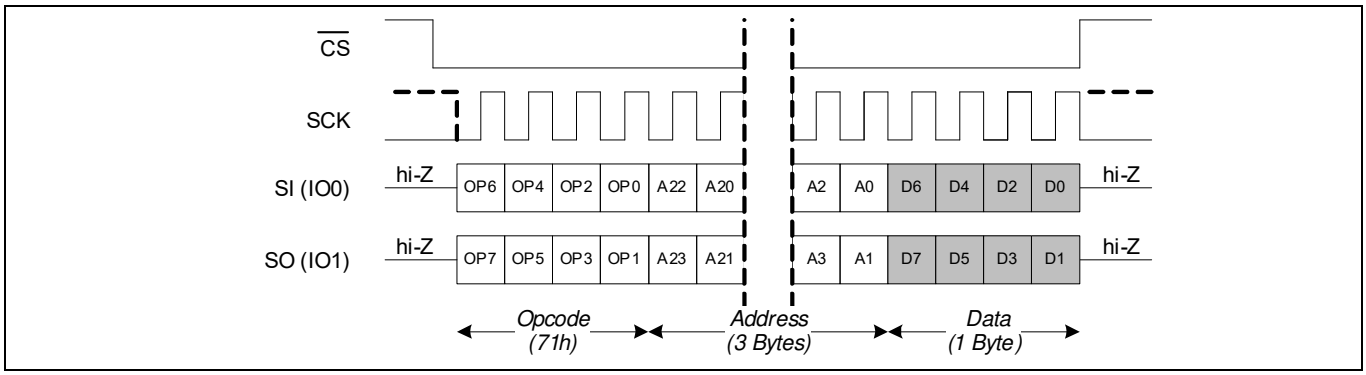



**Figure 34 Write any register (WRAR) in DPI mode**

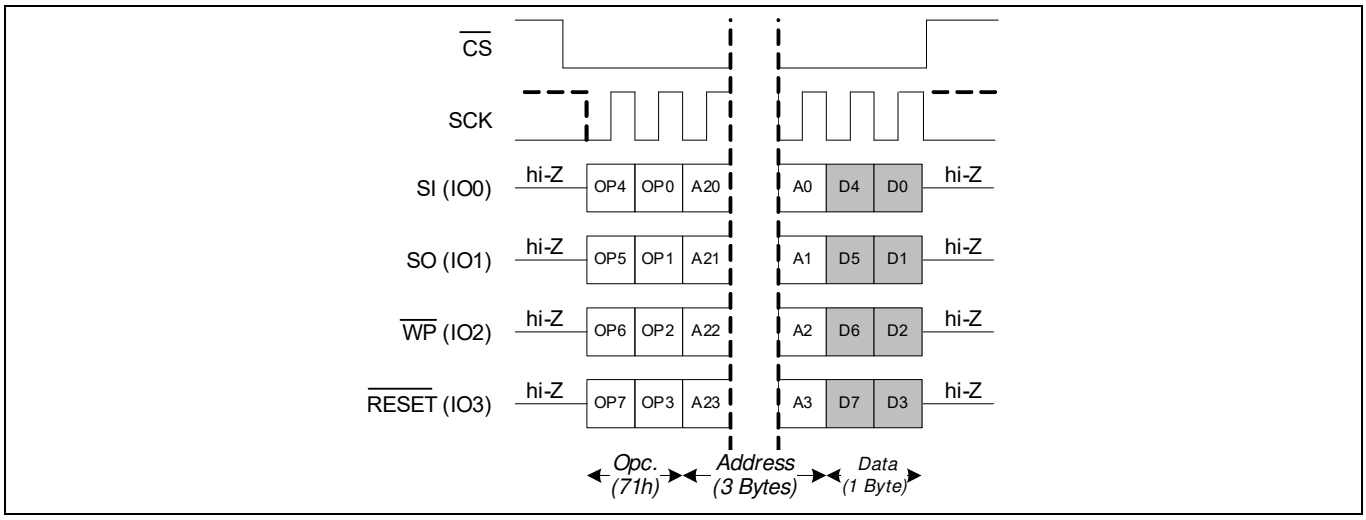



**Figure 35 Write any register (WRAR) in QPI mode**

**Table 38 Register address for generic register access**

| Function | Register type | Register content[17] | WRAR | RDAR[17] | Register address[16] | |
|---|---|---|---|---|---|---|
| | | | | | Volatile | Nonvolatile |
| Device status | Status register 1 | Volatile and nonvolatile | Yes | Yes | 0x070000 | 0x000000 |
| | Status register 2 | Volatile only | NA | Yes | 0x070001 or 0x000001 | |
| Device configuration | Configuration register 1 | Volatile and nonvolatile | Yes | Yes | 0x070002 | 0x000002 |
| | Configuration register 2 | Volatile and nonvolatile | Yes | Yes | 0x070003 | 0x000003 |
| | Configuration register 4 | Volatile and nonvolatile | Yes | Yes | 0x070005 | 0x000005 |
| | Configuration register 5 | Volatile and nonvolatile | Yes | Yes | 0x070006 | 0x000006 |
| Error correction | ECC status register | Volatile only | NA | Yes | 0x070089 or 0x000089 | |
| | ECC count register [7:0] | Volatile only | NA | Yes | 0x07008A or 0x00008A | |
| | ECC count register [15:8] | Volatile only | NA | Yes | 0x07008B or 0x00008B | |
| | ECC address trap register [7:0] | Volatile only | NA | Yes | 0x07008E or 0x00008E | |
| | ECC address trap register [15:8] | Volatile only | NA | Yes | 0x07008F or 0x00008F | |
| | ECC address trap register [23:16] | Volatile only | NA | Yes | 0x070040 or 0x000040 | |
| | ECC address trap register [31:24] | Volatile only | NA | Yes | 0x070041 or 0x000041 | |
| Cyclic redundancy check | CRC register [7:0] | Volatile only | NA | Yes | 0x070095 or 0x000095 | |
| | CRC register [15:8] | Volatile only | NA | Yes | 0x070096 or 0x000096 | |
| | CRC register [23:16] | Volatile only | NA | Yes | 0x070097 or 0x000097 | |
| | CRC register [31:24] | Volatile only | NA | Yes | 0x070098 or 0x000098 | |

**Notes**
16. The volatile registers return to their default state after POR or hardware reset. Refer to **Table 59** for the volatile register status after any POR or Reset event.
17. The RDAR command always returns content from the volatile register. Therefore, RDAR followed by either volatile register address or nonvolatile register address will return identical values (from respective volatile register only). The volatile only register doesn't have associated nonvolatile register.

### 5.1.2.9 Read any register (RDAR, 65h)

The RDAR instruction allows reading CY15X102QSN registers, one register at a time, addressable by their 3-byte addressing. The RDAR opcode is followed by the three-byte address of the register and dummy cycle (per register latency set in CR5), after which CY15X102QSN returns one byte register content on its output bus. The host should terminate the RDAR command by pulling CS HIGH after one register byte is received. Keeping CS LOW after the first data byte received will return undefined data byte(s). The RDAR instruction timing diagrams are shown in **Figure 36** to **Figure 38**.

**Notes**

- Since the status and configuration register read always returns the register content from its volatile space, hence the 3-byte address following the WRAR opcode can be the register address of either volatile register or its associated nonvolatile register.
  **Table 38** shows register addresses for both volatile and nonvolatile registers.
- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.

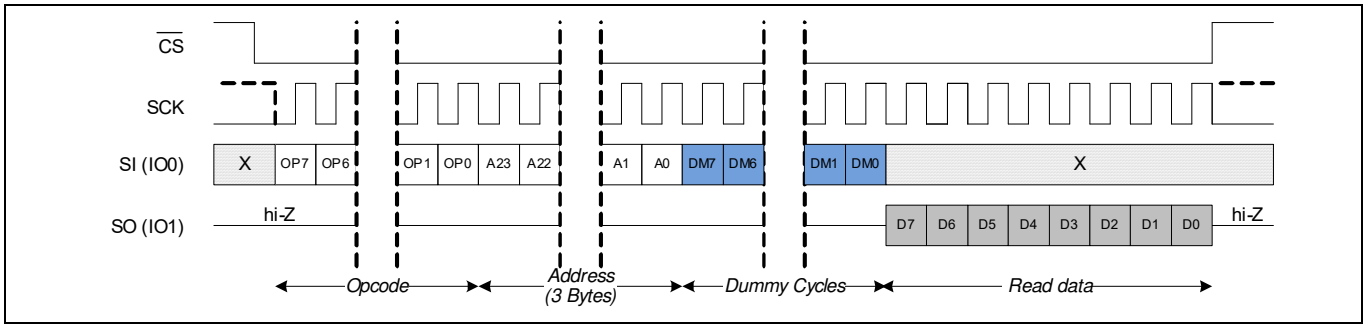



**Figure 36 Read any register (RDAR) in SPI mode**

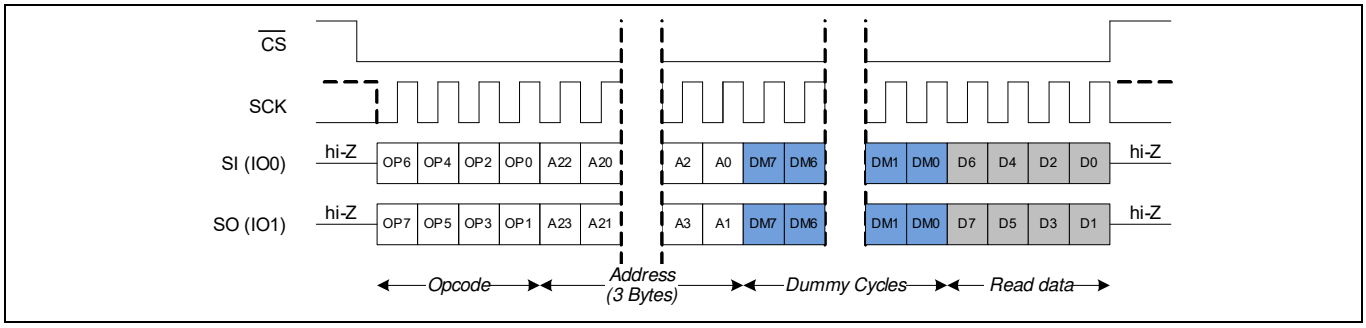



**Figure 37 Read any register (RDAR) in DPI mode**

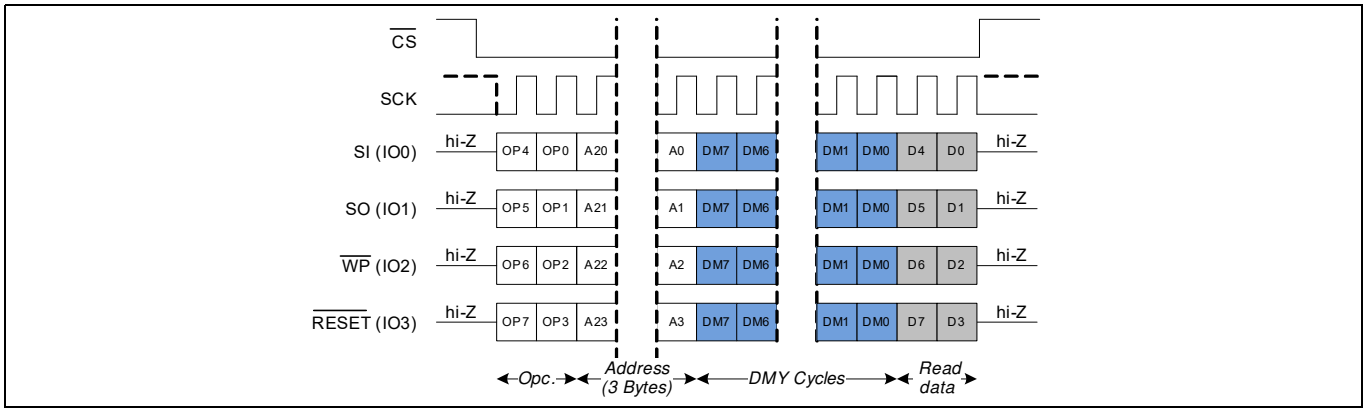



**Figure 38 Read any register (RDAR) in QPI mode**

### 5.1.3 Memory operation

The SPI interface, which is capable of a high clock frequency, highlights the fast write capability of the F-RAM technology. Unlike serial flash, the CY15X102QSN can perform sequential writes at bus speed. No page register is needed and any number of sequential writes can be performed.

### 5.1.4 Memory write operation commands

The memory write instruction is sent after the $\overline{CS}$ pin is pulled LOW. The write opcode is followed by a three-byte address and mode byte for XIP (as applicable). The CY15X102QSN has a 18-bit address space for 2-Mbit (256K × 8) density. The most significant address byte contains A15, A16, and A17 active bits while the remaining A[23:18] bits are considered 'don't care'. Address bits A17 to A0 are transmitted in three bytes over the SPI bus, following the (XIP) mode byte, if supported. Immediately after the last address bit or the last mode bit (if XIP is supported) is transmitted, the data byte(s) ([D7:0]) is (are) transmitted through the input line (s). The memory write operations are allowed in SPI, extended SPI, DPI or QPI Modes in SDR and DDR bus interfaces and some of them support execute-in-place (XIP). **Table 39** shows the list of memory write commands supported in CY15X102QSN in various SPI bus interface and data transfer modes.

**Notes**

- When a burst write reaches a protected block address, it continues incrementing the address into the protected space but does not write any data to the protected memory. If the address rolls over and takes the burst write to unprotected space, it resumes writes. The same operation is true if a burst write is initiated within a write-protected block.
- If the power is lost in the middle of the byte transfer during the write operation, only the last completed byte will be written.

**Table 39 Memory write commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| WRITE | 02 | Memory write - write to F-RAM array |
| DDRWRITE | DE | DDR write - memory write in QPI DDR mode |
| FAST_WRITE | DA | Memory fast write - memory write with execute-in-place |
| DDR_FAST_WRITE | DD | DDR fast write - memory fast write in DDR mode |
| DIW | A2 | Dual input write - command, address and mode byte are sent on single SI line, data bytes are sent on dual input lines I/O1 (SO), I/O0 (SI) |
| DIOW | A1 | DDR dual I/O write - command is sent on single SI line, address and mode byte and data bytes are sent on dual input lines I/O1 (SO), I/O0 (SI) |
| QIW | 32 | Quad input write - command, address and mode bytes are sent on single SI line, data bytes are sent on quad input lines I/O3 ($\overline{RESET}$), I/O2 ($\overline{WP}$), I/O1 (SO), I/O0 (SI) |
| QIOW | D2 | Quad I/O write - command is sent on single SI line, address and mode byte and data bytes are sent on quad input lines I/O3 ($\overline{RESET}$), I/O2 ($\overline{WP}$), I/O1 (SO), I/O0 (SI) |
| DDRQIOW | D1 | DDR quad I/O write - quad I/O write in DDR mode |

**Table 40 Memory write command details**

| Command | | | SPI bus interface | | | | | | | Data transfer | | XIP | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Command | Opcode (Hex) | Address length | SPI | Dual data | Quad data | Dual I/O | Quad IO | DPI | QPI | SDR | DDR | Execute-in-place (mode byte) | |
| WRITE | 02 | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | NA | 108 MHz |
| DDRWRITE | DE | 3 bytes | NA | | | | | | Yes | NA | Yes | NA | 54 MHz |
| FAST_ WRITE | DA | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | 108 MHz |
| DDR_FAST _WRITE | DD | 3 bytes | NA | | | | | | Yes | NA | Yes | Yes | 54 MHz |
| DIW | A2 | 3 bytes | NA | Yes | NA | | | | | Yes | NA | Yes | 108 MHz |
| DIOW | A1 | 3 bytes | NA | | | Yes | NA | | | Yes | NA | Yes | 108 MHz |
| QIW | 32 | 3 bytes | NA | | Yes | NA | | | | Yes | NA | Yes | 108 MHz |
| QIOW | D2 | 3 bytes | NA | | | | Yes | NA | | Yes | NA | Yes | 108 MHz |
| DDRQIOW | D1 | 3 bytes | NA | | | | Yes | NA | | NA | Yes | Yes | 54 MHz |

### 5.1.4.1 Write (WRITE, 02h)

Write operations are preformed when the WRITE opcode, along with write data, are transmitted on the SI pin for SPI mode, or I/O1 and I/O0 pins for DPI mode, or I/O3, I/O2, I/O1, and I/O0 pins for QPI mode. The burst writes can be used to write consecutive addresses without issuing a new WRITE instruction. If only one byte is to be written, the $\overline{CS}$ pin must be driven HIGH after the D0 (LSb of data) is transmitted. However, if more bytes are to be written, $\overline{CS}$ pin must be held LOW and the address is incremented automatically. The data bytes on the input pin(s) are written in successive addresses. When the internal address counter reaches to 0x3FFFF, the address rolls over to 0x00000 and the device continues to write.

**Notes**

- The WRITE instruction will only execute if the WEL bit (SR1[1]) is set to '1'.
- The WEL bit (SR1[1]) does not clear to '0' on completion of the WRITE operation. Therefore, any write command following the WRITE operation doesn't require preceding WREN command to set the WEL bit to '1'.

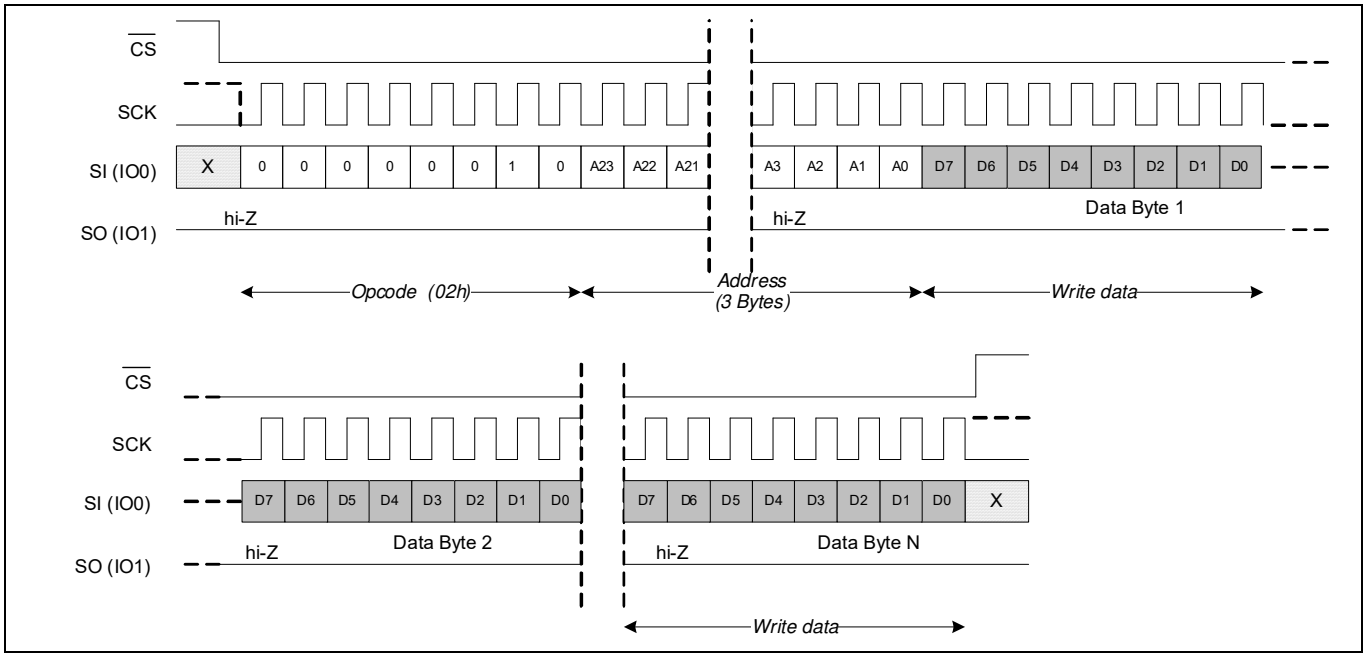



**Figure 39 Memory write (WRITE) in SPI mode**

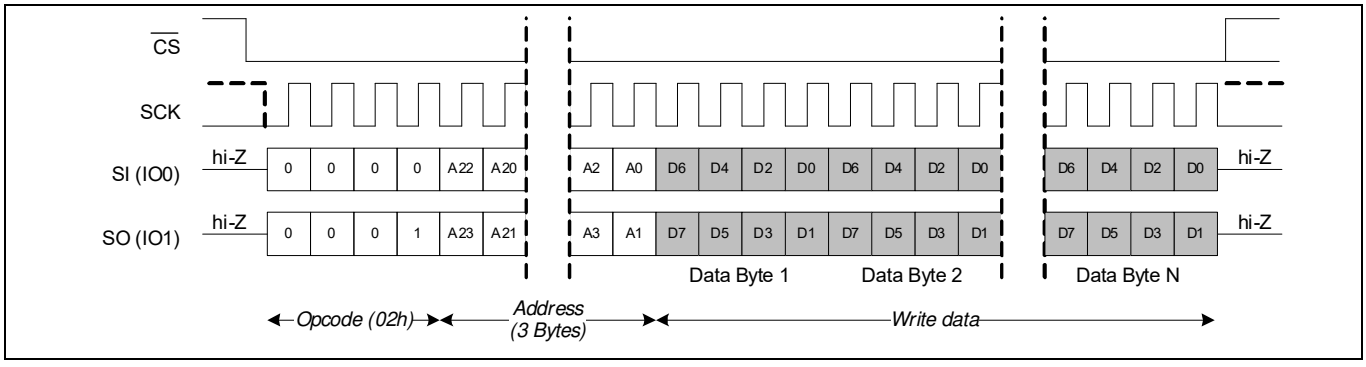



**Figure 40 Memory write (WRITE) in DPI mode**

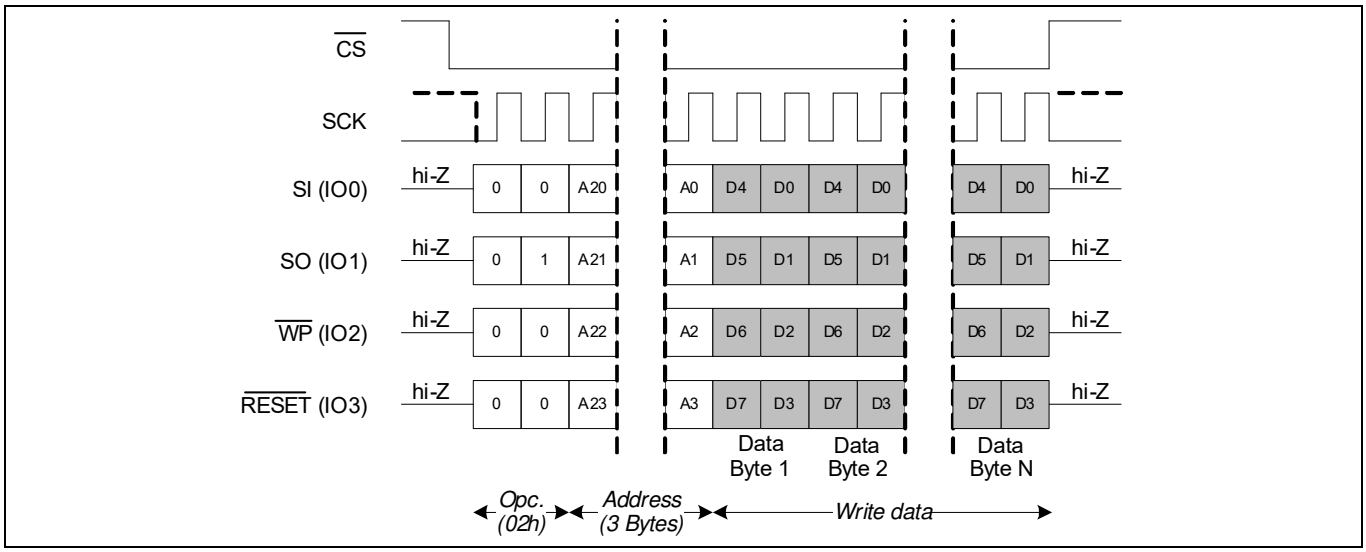



**Figure 41 Memory write (WRITE) in QPI mode**

## 5.1.4.2 DDR write (DDRWRITE, DEh)

The DDRWRITE instruction improves bandwidth by transferring address and data bits on both edges of SCK. The address can start at any byte location of the memory array. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can therefore be written with one single write opcode and the start address provided. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to be continued indefinitely. This opcode does not support SPI Mode 3.

**Notes**

- DDRWRITE instruction can only be executed by the device if the WEL bit is set to '1' to enable write operations.
- The WEL bit does not reset to '0' on completion of the DDRWRITE operation.

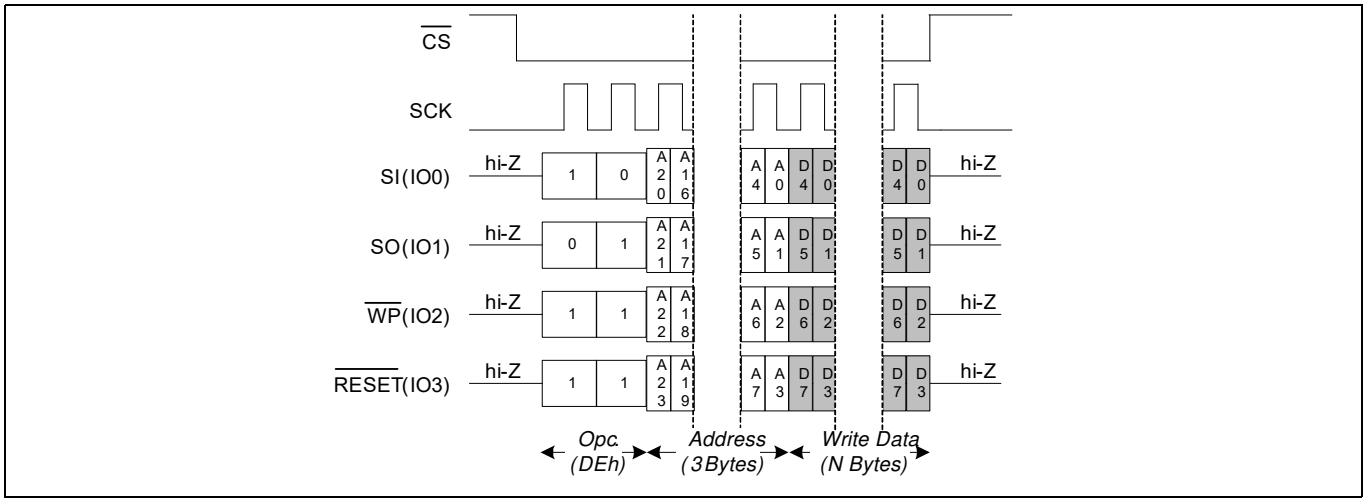



**Figure 42 DDR write (DDRWRITE) in QPI mode**

### 5.1.4.3 Fast write (FAST_WRITE, DAh)

The FAST_WRITE instruction is similar to WRITE instruction except for it allows for XIP operation set through mode byte. Mode bits allow a series of fast write instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), reduces initial access times (improves performance). The mode bits control the length of the next fast write operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous fast write mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the DAh opcode thus eliminating 8-cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the FAST_WRITE XIP mode.
- FAST_WRITE instruction can only be executed by the device if the write enable latch (WEL) in the status register is set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the FAST_WRITE operation.

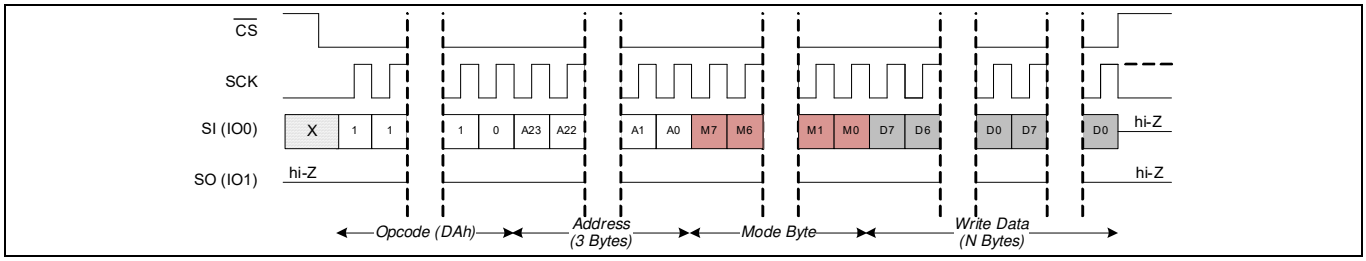



**Figure 43 Fast write (FAST_WRITE) in SPI mode**

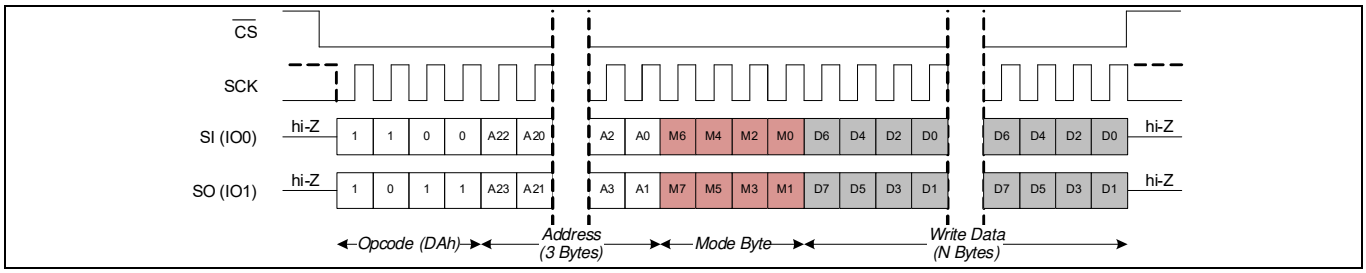



**Figure 44 Fast write (FAST_WRITE) in DPI mode**

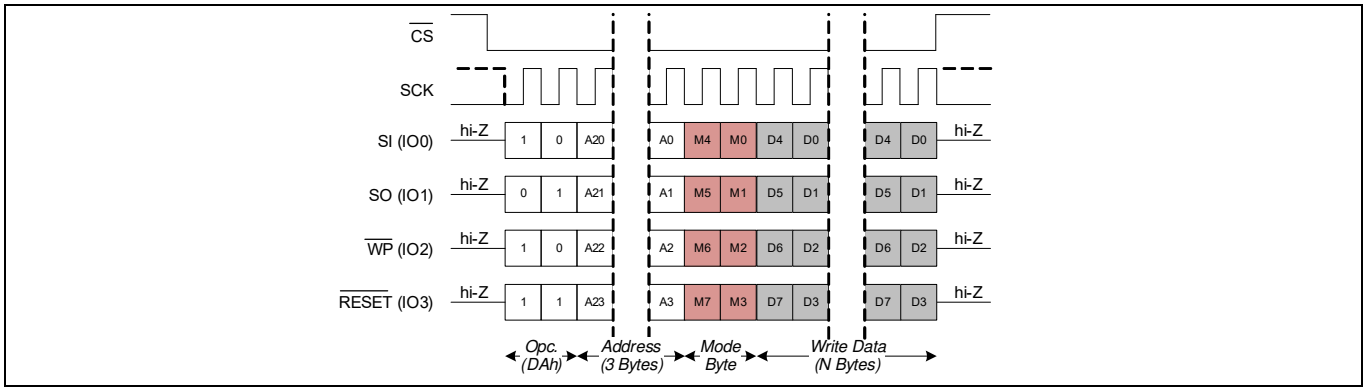



**Figure 45 Fast write (FAST_WRITE) in QPI mode**

### 5.1.4.4 DDR fast write (DDR_FAST_WRITE, DDh)

The DDR_FAST_WRITE instruction is similar to the DDRWRITE instruction, except that it allows XIP operation.

Mode bits allow a series of DDR_FAST_WRITE instructions to eliminate the 8-bit opcode after the first instruction sends an A5h mode bit (“10100101”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves performance). The mode bits control the length of the next DDR_FAST_WRITE operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are A5h, the device transitions to continuous DDR_FAST_WRITE mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the DDh opcode, thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW. This opcode doesn’t support SPI mode 3.

**Notes**

- Mode bits with !A5h (logical NOT of A5h byte) will exit the DDR_FAST_WRITE XIP mode.
- DDR_FAST_WRITE instruction can only be executed by the device if the WEL bit is set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the DDR_FAST_WRITE operation.

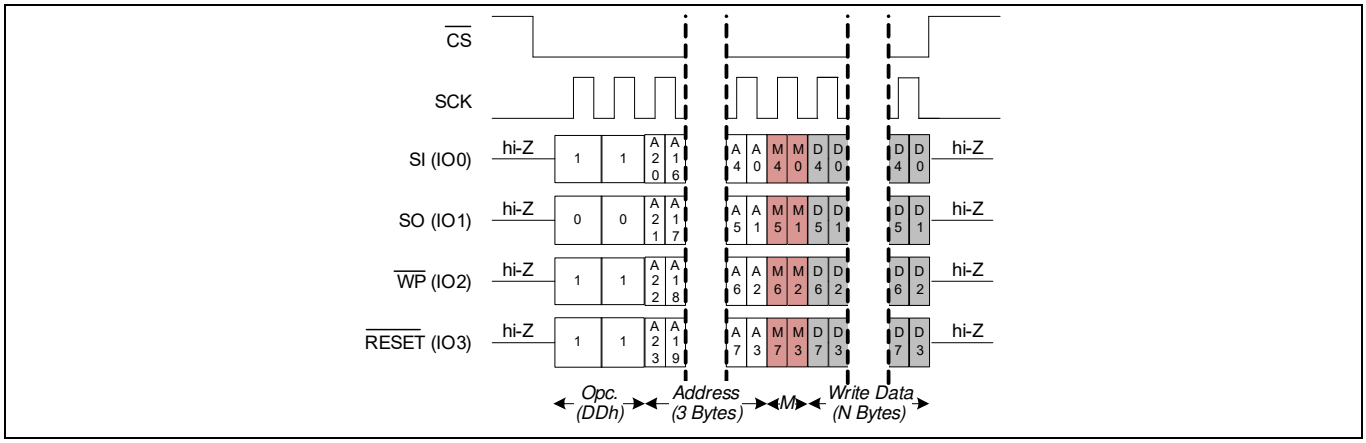



**Figure 46 DDR fast write (DDR_FAST_WRITE) in QPI mode**

### 5.1.4.5 Dual input write (DIW, A2h)

The DIW instruction can be used in dual data mode which is part of the extended SPI write instructions. In dual data mode, opcode, address and mode bytes are transmitted through SI pin, one bit per clock cycle. Immediately after the last address bit is transmitted, the pins are reconfigured as SO becoming I/O1, and SI becoming I/O0, and the data (D[7:0]) is transmitted into the I/O1, and I/O0 pins, 2 bits per clock cycle, starting with D7 on I/O1 and D6 on I/O0.

Mode bits allow a series of DIW instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), and reduces initial access times (improves performance). The mode bits control the length of the next DIW operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous DIW mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the A2h opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the DIW XIP mode.
- DIW instruction can only be executed by the device when the WEL bit is set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the DIW operation.

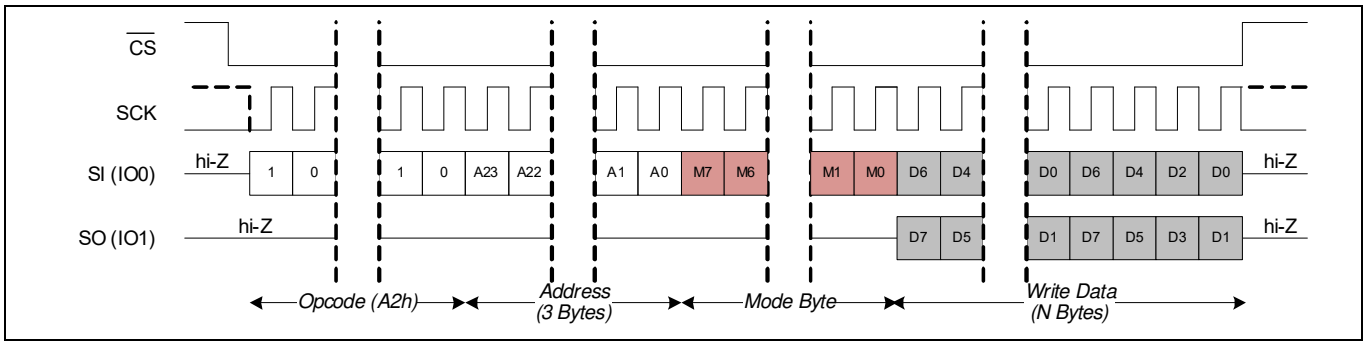



**Figure 47 Dual input write (DIW)**

### 5.1.4.6 Dual I/O write (DIOW, A1h)

The DIOW instruction can be used in dual addr/data mode, which is part of extended SPI write instructions. In dual addr/data mode, the opcode is transmitted through the SI pin, one bit per clock cycle. Immediately after the last opcode bit is transmitted, the pins are reconfigured as SO becoming I/O1, and SI becoming I/O0, and the address along with the mode byte are transmitted into the part through I/O1 and I/O0 pins, 2 bits per clock cycle, starting with address A23 on I/O1, A22 on I/O0, until the three-byte address is input. After the last address bits are transmitted, the data (D[7:0]) is transmitted into the part through I/O1 and I/O0 two bits per clock cycle starting with D7 on I/O1 and D6 on I/O0.

Mode bits allow a series of DIOW instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves performance). The mode bits control the length of the next DIOW operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous DIOW mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the A1h opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the DIOW XIP mode.
- The DIOW instruction can only be executed by the device when the WEL bit set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the DIOW operation.

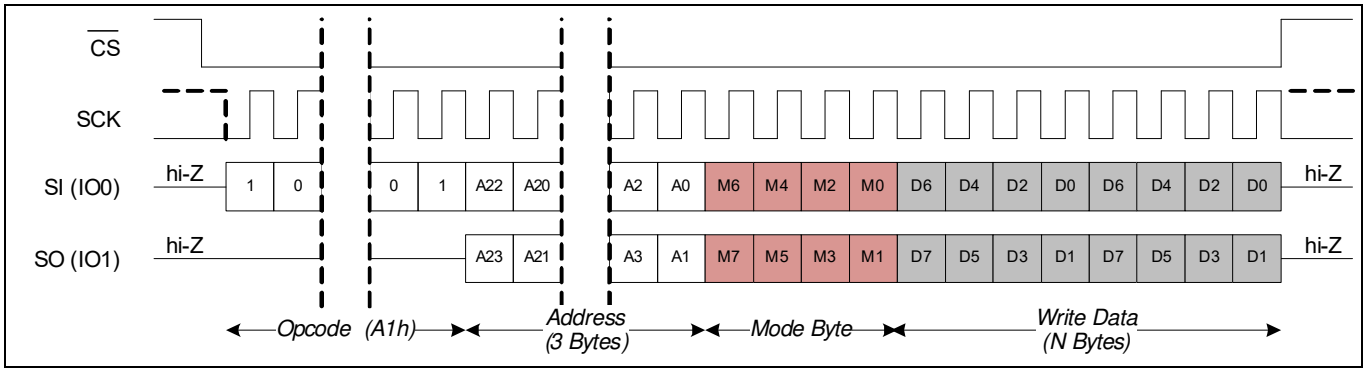



**Figure 48 Dual I/O write (DIOW)**

### 5.1.4.7 Quad input write (QIW, 32h)

The QIW instruction is used in quad data mode which is part of extended SPI write instructions. In quad data mode, opcode, address, and mode bytes are transmitted through the SI pin, one bit per clock cycle. Immediately after the last address bit is transmitted, the pins are reconfigured as RESET becoming I/O3, WP becoming I/O2, SO becoming I/O1, and SI becoming I/O0, and the data (D7–D0) is transmitted into the I/O3 I/O2, I/O1, and I/O0 pins, 4 bits per clock cycle, starting with D7 on I/O3 and D6 on I/O2, D5 on I/O1, and D4 on I/O0.

Mode bits allow a series of QIW instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves performance). The mode bits control the length of the next QIW operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous QIW mode and the next address can be entered (after CS is raised HIGH and then asserted LOW) without requiring the 32h opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once CS transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the QIW XIP mode.
- The QIW instruction can only be executed by the device if the write enable latch (WEL) in the status register is set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the QIW operation.

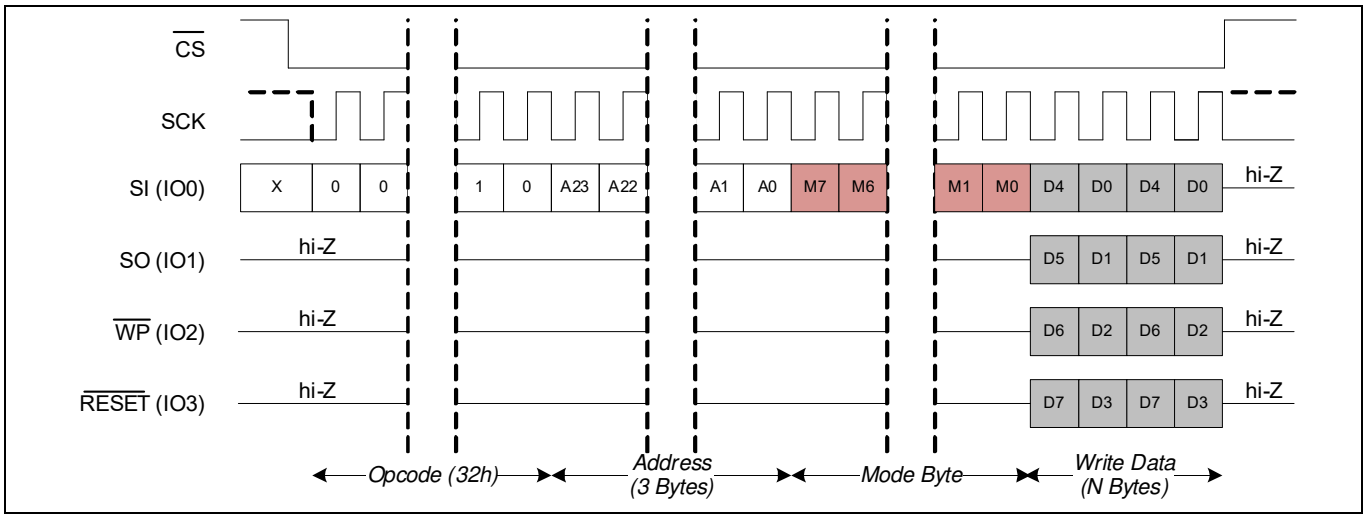



**Figure 49 Quad input write (QIW)**

### 5.1.4.8 Quad I/O write (QIOW, D2h)

The QIOW instruction can be used in quad addr/data mode which is part of extended SPI write instructions. In quad addr/data mode, opcode is transmitted through SI pin, one bit per clock cycle. Immediately after the last opcode bit is transmitted, the pins are reconfigured as RESET becoming I/O3, WP becoming I/O2, SO becoming I/O1, and SI becoming I/O0, and the address is transmitted into the part through I/O3, I/O2, I/O1 and I/O0 pins, 4 bits per clock cycle, starting with address A23 on I/O3, A22 in I/O2, A21 on I/O1 and A20 on I/O0, until the three-byte address is input. After the last address bits are transmitted, the data (D7–D0) is transmitted into the part through I/O3, I/O2, I/O1, and I/O0 four bits per clock cycle starting with D7 on I/O3, D6 on I/O2, D5 on I/O1 and D4 on I/O0.

Mode bits allow a series of QIOW instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves performance). The mode bits control the length of the next QIOW operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous DIOW mode and the next address can be entered (after CS is raised HIGH and then asserted LOW) without requiring the D2h opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once CS transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the QIOW XIP mode.
- The QIOW instruction can only be executed by the device if the write enable latch (WEL) in the status register is set to ‘1’ to enable write operations.
- The WEL bit does not reset to ‘0’ on completion of the QIOW operation.

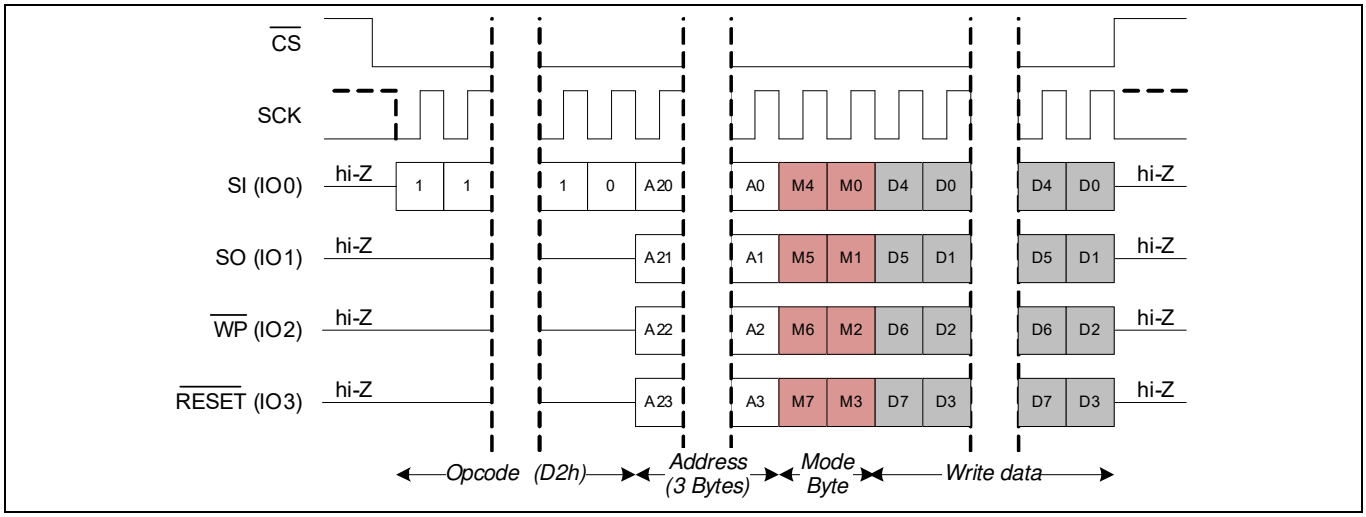



**Figure 50 Quad I/O write (QIOW)**

### 5.1.4.9 DDR quad I/O write (DDRQIOW, D1h)

Double data rate quad I/O write is similar to quad I/O except that the address and data are sent on every edge of the clock and the mode bit pattern in DDRQIOW is A5h (“10100101”). This opcode does not support SPI mode 3.

**Note** Mode bits with !A5h (logical NOT of A5h byte) will exit the DDRQIOW XIP mode.

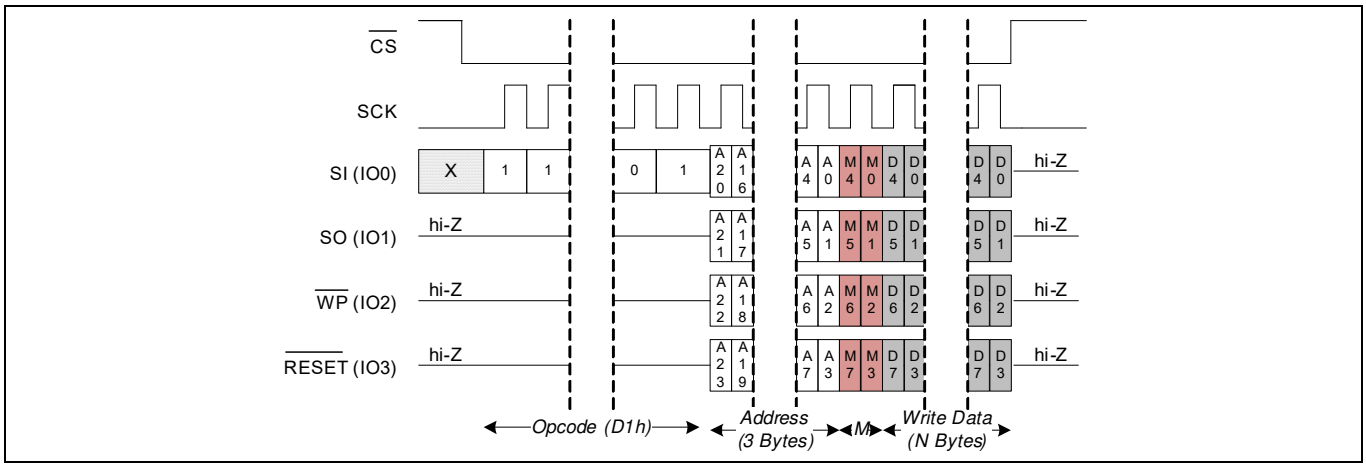



**Figure 51 Quad I/O write (QIOW)**

### 5.1.5 Memory read operation commands

The memory read instruction is sent after the $\overline{CS}$ pin is pulled LOW to select a device. The read opcode is followed by a three-byte address and mode byte for XIP (as applicable). The CY15X102QSN has a 18-bit address space for 2-Mbit (256K × 8) density. The most significant address byte contains A15, A16, and A17 active bits while the remaining bits are considered 'don't care'. Address bits A17 to A0 are transmitted as three bytes over the SPI bus followed by the mode byte and Dummy cycles as applicable.

The memory read supports SPI, extended SPI, DPI, or QPI modes in SDR and DDR bus interface and includes execute-in-place (XIP) support. **Table 41** shows the list of memory read commands supported in CY15X102QSN in various SPI bus interface and data transfer modes.

**Table 41 Memory read commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| READ | 03 | Memory read - reads up to 50 MHz without memory latency cycle in SPI SDR mode and up to 108 MHz with memory latency cycles in SPI, DPI, QPI SDR modes |
| FAST_READ | 0B | Memory fast read - reads up to 108 MHz with memory latency cycles in SPI, DPI, QPI SDR modes |
| DDRFR | 0D | DDR fast read - fast read instruction in QPI DDR mode |
| DOR | 3B | Dual output read - command and address bytes are sent on single SI line and data on dual output lines I/O1 (SO), I/O0 (SI) |
| DIOR | BB | Dual I/O read - command sent on single SI line, address input and data output on dual output lines I/O1 (SO), I/O0 (SI) |
| QOR | 6B | Quad output read - command and address sent on single SI line, data on quad output lines I/O3 ($\overline{RESET}$), I/O2 ($\overline{WP}$), I/O1 (SO), I/O0 (SI) |
| QIOR | EB | Quad I/O read - command sent on single SI line, address input and data output on quad output lines I/O3 ($\overline{RESET}$), I/O2 ($\overline{WP}$), I/O1 (SO), I/O0 (SI). This opcode executes in extended SPI (quad I/O) SDR and in QPI SDR mode |
| DDRQIOR | ED | Quad I/O read in SDR and DDR modes. This opcode executes in extended SPI (quad I/O) SDR and in QPI DDR mode. |

**Table 42 Memory read command details**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | XIP | Memory latency | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Execute-in-place | Dummy cycles | |
| 03 | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | NA | Yes | 108 MHz |
| 0B | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | Yes | 108 MHz |
| 0D | 3 bytes | NA | | | | | | Yes | NA | Yes | Yes | Yes | 54 MHz |
| 3B | 3 bytes | NA | Yes | NA | | | | | Yes | NA | Yes | Yes | 108 MHz |
| BB | 3 bytes | NA | | | Yes | NA | | | Yes | NA | Yes | Yes | 108 MHz |
| 6B | 3 bytes | NA | | Yes | NA | | | | Yes | NA | Yes | Yes | 108 MHz |
| EB | 3 bytes | NA | | | | Yes | NA | Yes | Yes | NA | Yes | Yes | 108 MHz |
| ED | 3 bytes | NA | | | | Yes | NA | Yes | NA | Yes | Yes | Yes | 54 MHz |

### 5.1.5.1 Memory read (READ, 03h)

The READ instruction reads out the memory contents at the given address. The address can start at any byte location of the 2-Mbit memory array determined by the three-byte address. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire 2-Mbit memory can therefore be read out with one single read opcode and address provided. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely. This command executes in SPI, DPI, or QPI modes.

**Note** The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

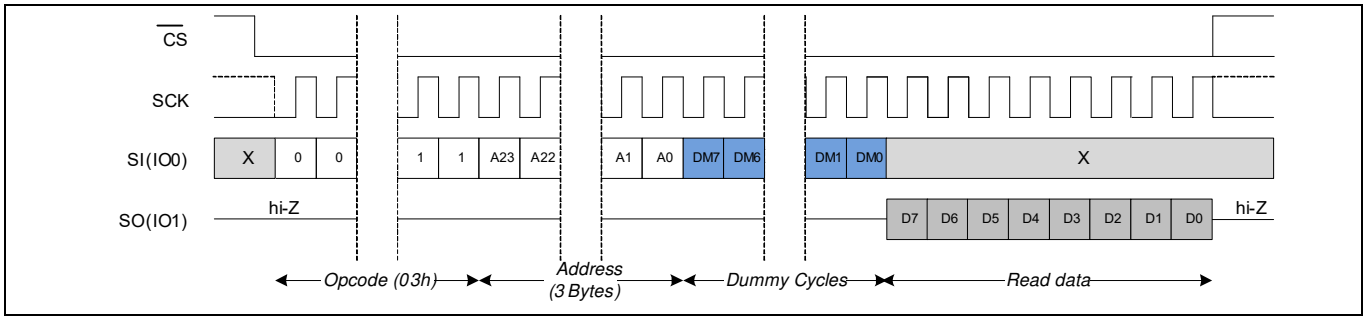



**Figure 52 READ in SPI mode**

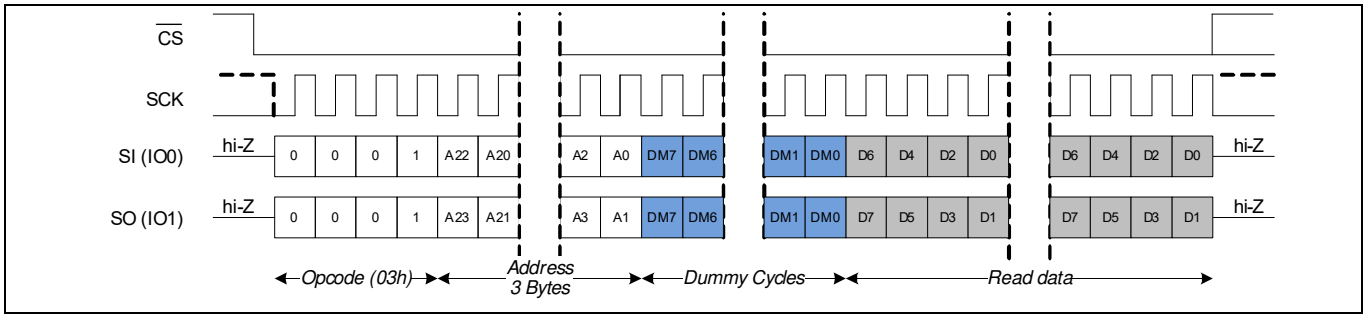



**Figure 53 READ in DPI mode**

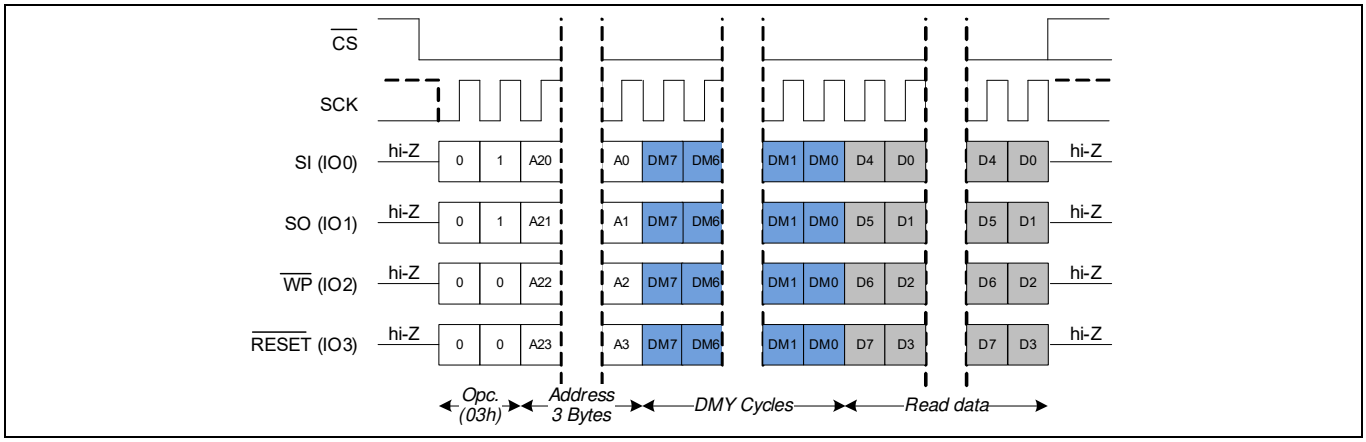



**Figure 54 READ in QPI mode**

### 5.1.5.2 Fast read (FAST_READ, 0Bh)

The FAST_READ instruction reads out the memory contents at the given address. The address can start at any byte location of the 2-Mbit memory array determined by the three-byte address. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can therefore be read out with one single read opcode and address provided. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely. This command executes in SPI, DPI or QPI modes.

Mode bits allow a series of fast read instructions to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves performance). The mode bits control the length of the next FAST_READ operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous FAST_READ mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the 0Bh opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the FAST_READ XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

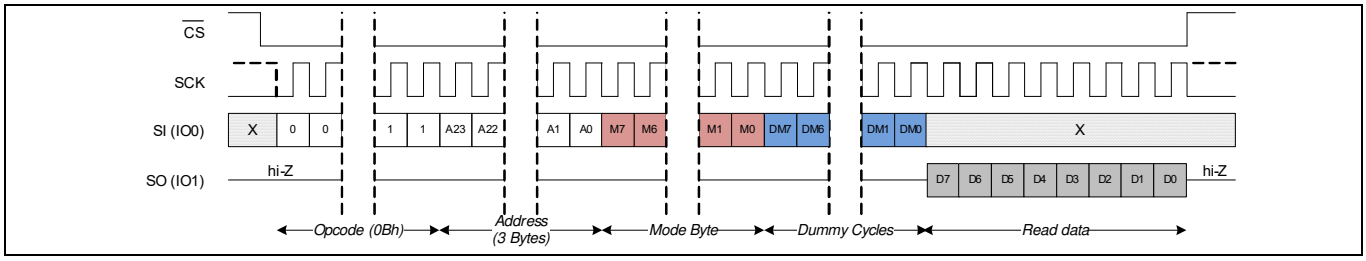



**Figure 55 FAST_READ in SPI mode**

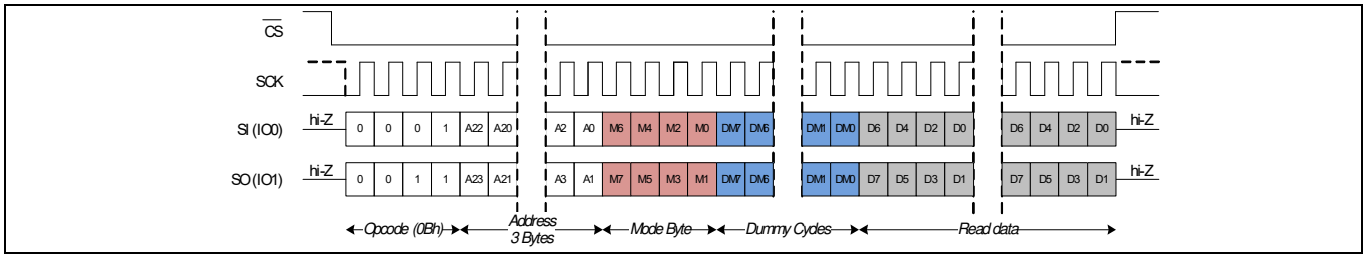



**Figure 56 FAST_READ in DPI mode**

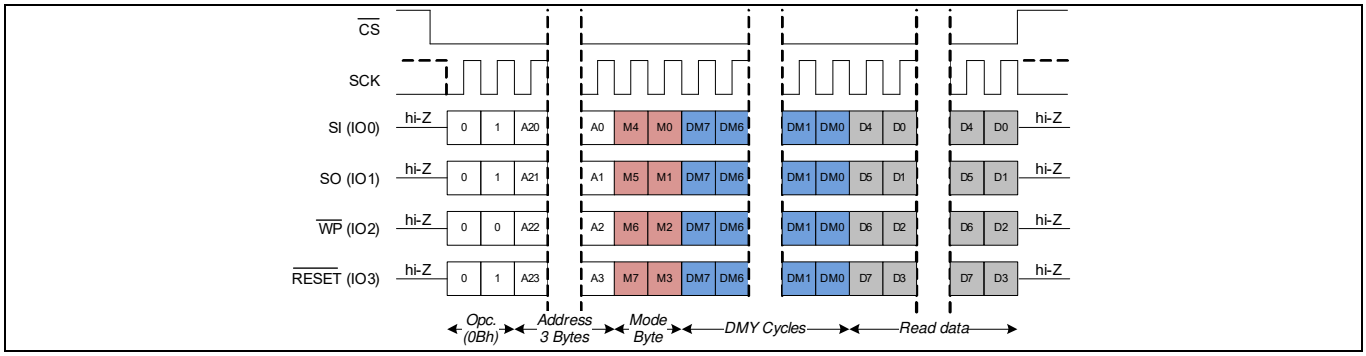



**Figure 57 FAST_READ in QPI mode**

### 5.1.5.3 DDR fast read (DDRFR, 0Dh)

The DDRFR instruction improves bandwidth by transferring address, dummy bits and data bits on every edge of the clock. The address can start at any byte location of the 2-Mbit memory array determined by the three-byte address. The address is automatically incremented to the next address in sequential order after each byte of data is shifted out. The entire memory can therefore be read out with one single read opcode and the start address provided. When the highest address 0x3FFFF is reached, the address counter wraps around and rolls back to 0x000000, allowing the read sequence to continue indefinitely. $\overline{CS}$ should remain LOW during the dummy cycle(s). This command executes in QPI mode.

Mode bits allow a series of fast read DDR instructions to eliminate the 8-bit opcode after the first instruction sends an A5h mode bit (“10100101”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next DDRFR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are A5h, the device transitions to continuous DDR fast read mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the 0Dh opcode thus eliminating eight cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW. This opcode doesn’t support SPI mode 3.

**Notes**

- Mode bits with !A5h (logical NOT of A5h byte) will exit the DDRFR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

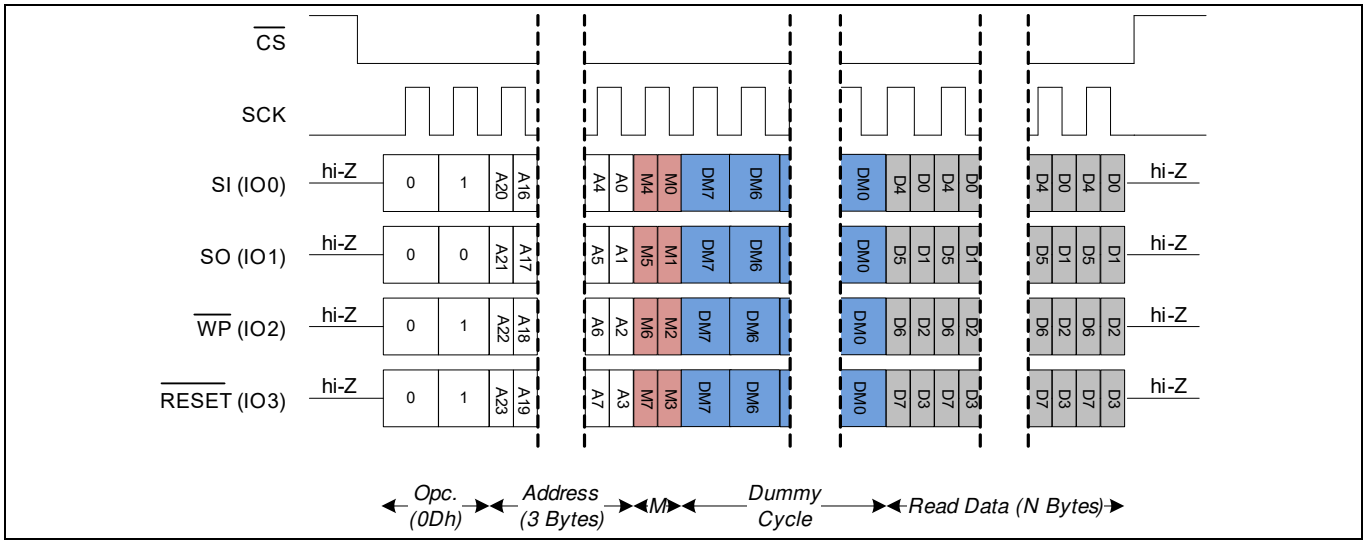



**Figure 58 DDRFR in QPI mode**

### 5.1.5.4 Dual output read (DOR, 3Bh)

The DOR instruction is used in dual data mode which is the part of extended SPI read instructions. In dual data mode, opcode, address, and mode byte (Axh) and dummy cycles are transmitted through SI pin, one bit per clock cycle. At the falling edge of SCK of the last dummy cycle, the pins are reconfigured as SO becoming I/O1, and SI becoming I/O0. The data (D7–D0) from the specified address is shifted out on I/O1, and I/O0 pins two bits per clock cycle starting with D7 on I/O1, and D6 on I/O. The address can start at any byte location of the memory array. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can therefore be read out. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely.

Mode bits allow a series of DOR instruction to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next DOR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh, the device transitions to continuous DOR Mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the 3Bh opcode thus eliminating eight cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the DOR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

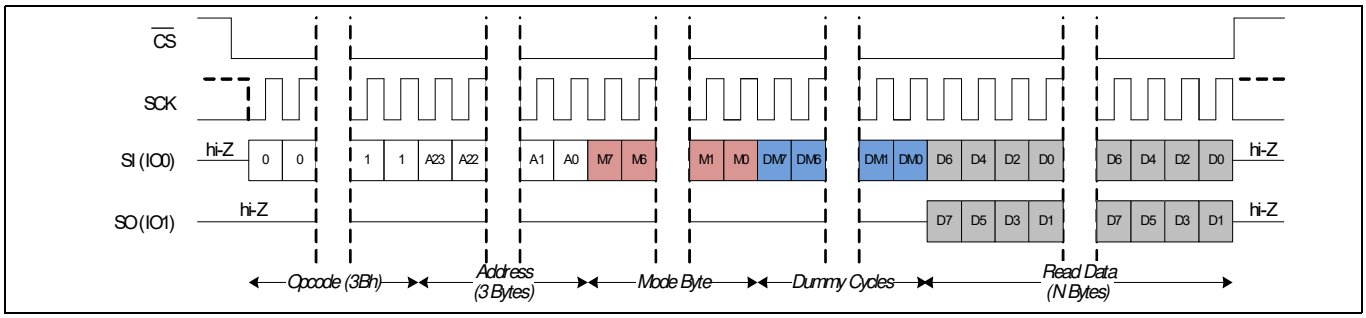



**Figure 59 Double output read (DOR)**

### 5.1.5.5 Dual I/O read (DIOR, BBh)

The DIOR instruction is used in dual addr/data mode which is part of extended SPI read instructions. In dual addr/data mode, opcode is transmitted through SI pin, one bit per clock cycle. After the last bit of the opcode, the pins are reconfigured as SO becoming I/O1, and SI becoming I/O0. The address is then transmitted into the part through I/O1 and I/O0 pins, 2 bits per clock cycle, starting with address A23 on I/O1 and A22 on I/O0, until the three-byte address is input. The data (D7–D0) at the specific address is shifted out on I/O1, and I/O0 pins two bits per clock cycle starting with D7 on I/O1, and D6 on I/O0. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can, therefore, be read out. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely.

Mode bits allow a series of DIOR instruction to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next DIOR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous DIOR mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the BBh opcode thus eliminating eight cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ transitions from HIGH to LOW.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the FAST_READ XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

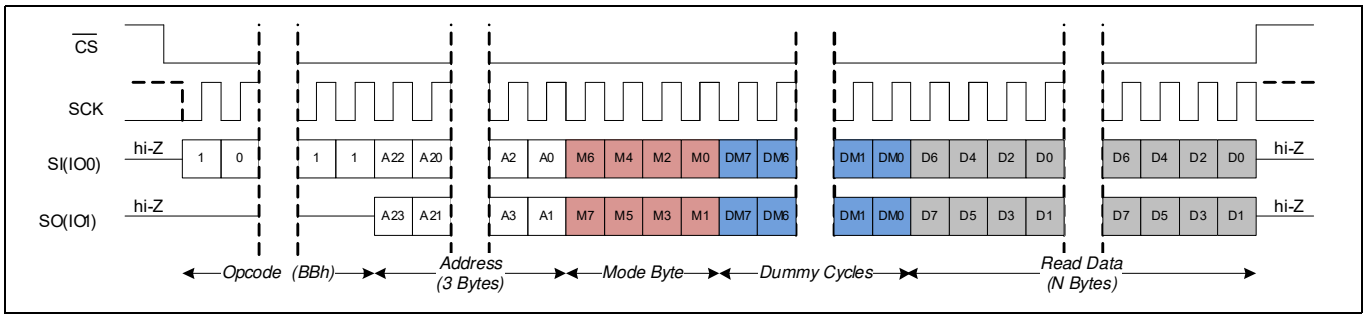



**Figure 60 Double I/O read (DIOR)**

### 5.1.5.6 Quad output read (QOR, 6Bh)

The QOR instruction is used in quad data mode which is the part of extended SPI read instructions. In quad data mode, opcode, address, mode byte (Axh) and dummy cycles are transmitted through SI pin, one bit per clock cycle. At the falling edge of SCK of the last mode cycle, the pins are reconfigured as RESET becoming I/O3, WP becoming I/O2, SO becoming I/O1, and SI becoming I/O0. The data (D7–D0) from the specified address is shifted out on I/O3, I/O2, I/O1, and I/O0 pins four bits per clock cycle starting with D7 on I/O3 and D6 on I/O2, D5 on I/O1, and D4 on I/O0. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can, therefore, be read out. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely.

Mode bits allow a series of DOR instruction to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next QOR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous QOR Mode and the next address can be entered (after CS is raised HIGH and then asserted LOW) without requiring the 6Bh opcode thus eliminating eight cycles from the instruction sequence. Otherwise, opcode is required once CS transitions from HIGH to LOW.

**Notes**

- The QUAD bit CR1[1] must be set to ‘1’ in the configuration register 1.
- Mode bits with !Axh (logical NOT of Axh byte) will exit the DOR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

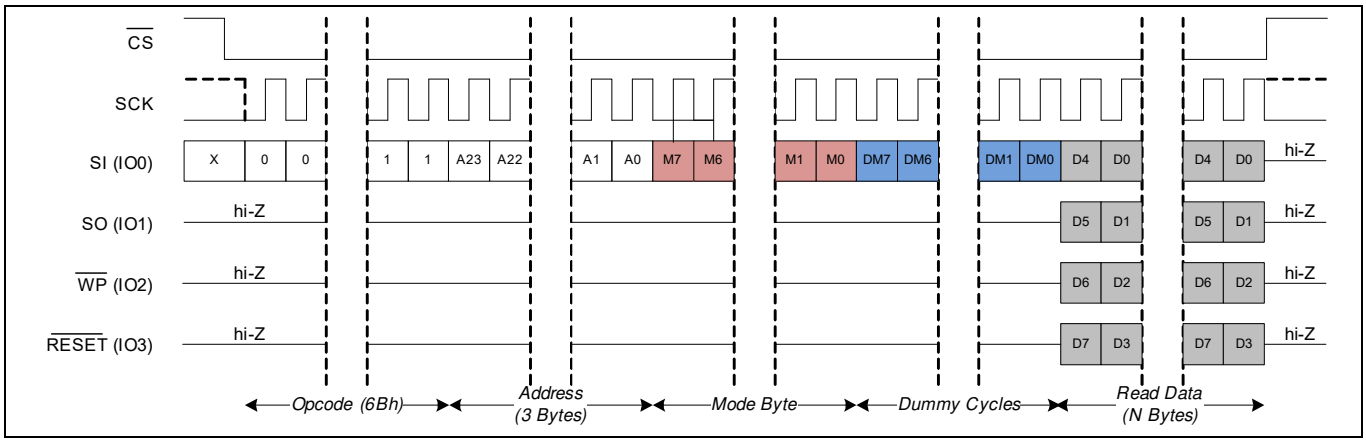



**Figure 61 Quad output read (QOR)**

### 5.1.5.7 Quad I/O read (QIOR, EBh) – in extended SPI mode

The QIOR instruction is used in quad addr/data mode which is part of extended SPI read instructions. In quad addr/data mode, opcode is transmitted through SI pin, one bit per clock cycle. After the last bit of the opcode, the pins are reconfigured as RESET becoming I/O3, WP becoming I/O2, SO becoming I/O1, and SI becoming I/O0. The address is then transmitted into the part through I/O3, I/O2, I/O1 and I/O0 pins, 4 bits per clock cycle, starting with address A23 on I/O3, A22 on I/O2, A21 on I/O1 and A20 on I/O0, until the three-byte address is input. The data (D7–D0) at the specific address is shifted out on I/O3, I/O2, I/O1, and I/O0 pins four bits per clock cycle starting with D7 on I/O3 and D6 on I/O2, D5 on I/O1, and D4 on I/O0. The entire memory can therefore be read out. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to continue indefinitely.

Mode bits allow a series of QIOR instruction to eliminate the 8-bit opcode after the first instruction sends an Axh mode bit (“1010XXXX”) pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next QIOR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous QIOR mode and the next address can be entered (after CS is raised HIGH and then asserted LOW) without requiring the EBh opcode thus eliminating 8 cycles from the instruction sequence. Otherwise, opcode is required once CS is raised HIGH and then asserted LOW.

**Notes**

- The QUAD bit CR1[1] must be set to ‘1’ in configuration register 1.
- Mode bits with !Axh (logical NOT of Axh byte) will exit the QIOR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

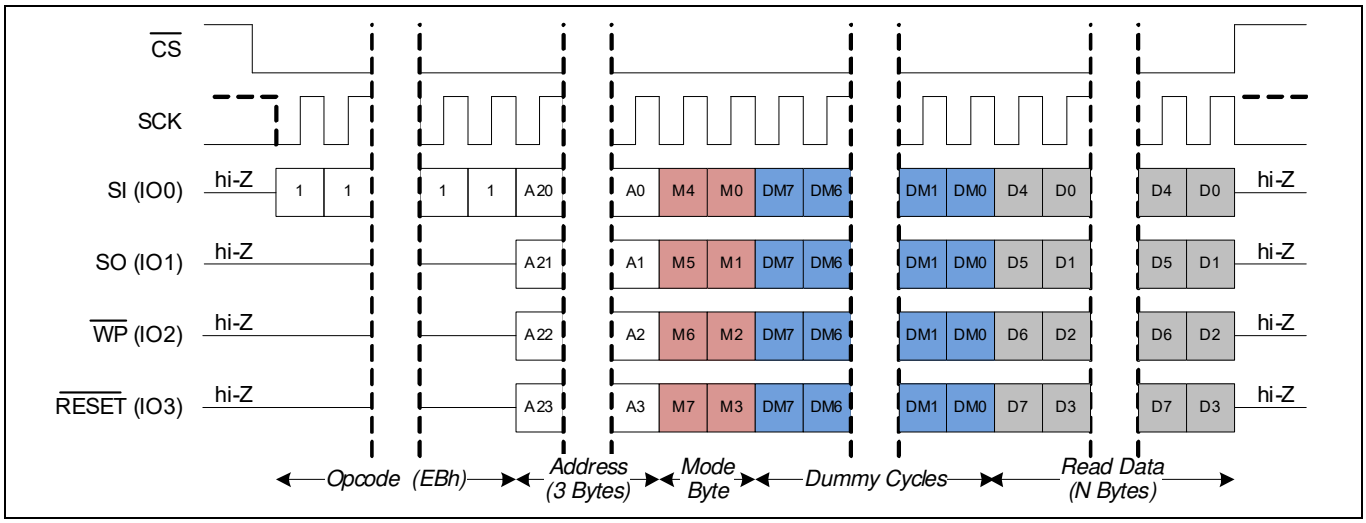



**Figure 62 Quad I/O read (QIOR) in extended SPI mode**

### 5.1.5.8 Quad I/O read (QIOR, EBh) – in QPI mode

The opcode for QIOR can be executed in the QSPI mode as well. As the device is in QSPI mode, the opcode, address, and mode bytes are transmitted over all four I/Os. The data (D7–D0) at the specific address is shifted out on I/O3, I/O2, I/O1, and I/O0 pins four bits per clock cycle starting with D7 on I/O3 and D6 on I/O2, D5 on I/O1, and D4 on I/O0.

**Notes**

- Mode bits with !Axh (logical NOT of Axh byte) will exit the QIOR mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

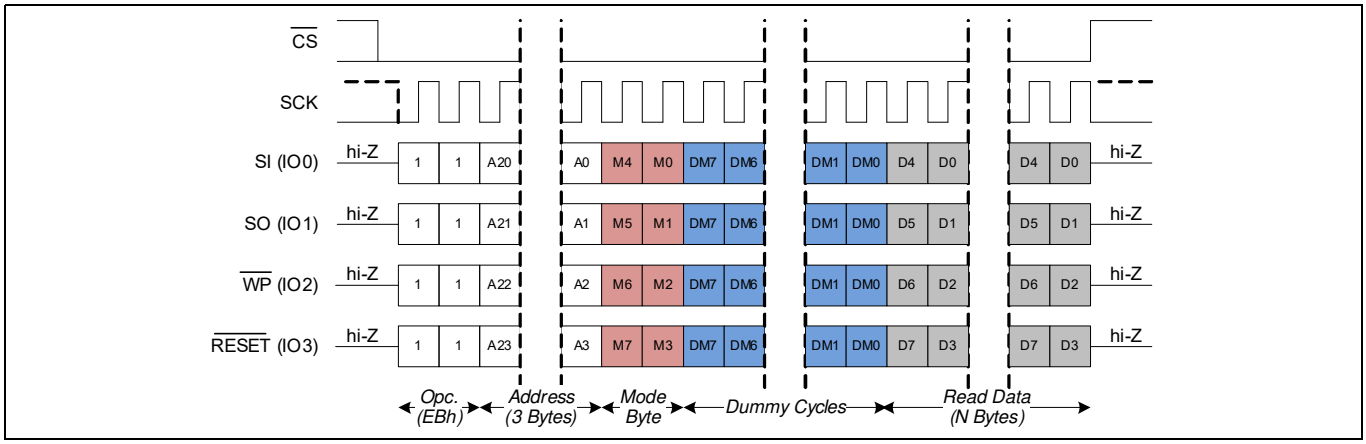



**Figure 63 Quad I/O read (QIOR) in QPI mode**

### 5.1.5.9 DDR quad I/O read (DDRQIOR EDh) – in extended SPI mode

The DDRQIOR instruction improves bandwidth with four I/O signals SI (I/O0), SO (I/O1), $\overline{WP}$ (I/O2) and $\overline{RESET}$ (I/O3). It is similar to the quad I/O read instruction but transfers address, mode, dummy or data bits on every edge of the clock. The address can start at any byte location of the memory array. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire memory can therefore be read out with one single read opcode and address provided. When the highest address 0x3FFFF is reached, the address counter will wrap around and roll back to 0x000000, allowing the read sequence to be continued indefinitely. $\overline{CS}$ should not be driven HIGH during dummy bits as this may make the bits indeterminate.

Mode bits allow a series of QIOR DDR instructions to eliminate the 8-bit opcode after the first instruction sends an A5h mode bit pattern. This feature, called execute-in-place (XIP), significantly reduces initial access times (improves XIP performance). The mode bits control the length of the next DDR QIOR operation through the inclusion or exclusion of the first byte instruction opcode. If the mode bits are Axh the device transitions to continuous QIOR DDR mode and the next address can be entered (after $\overline{CS}$ is raised HIGH and then asserted LOW) without requiring the EDh opcode thus eliminating eight cycles from the instruction sequence. Otherwise, opcode is required once $\overline{CS}$ is raised HIGH and then asserted LOW. This opcode doesn't support SPI mode 3.

**Notes**

- The QUAD bit CR1[1] must be set to '1' in configuration register 1.
- Mode bits with !A5h (logical NOT of A5h byte) will exit the DDRQIOR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

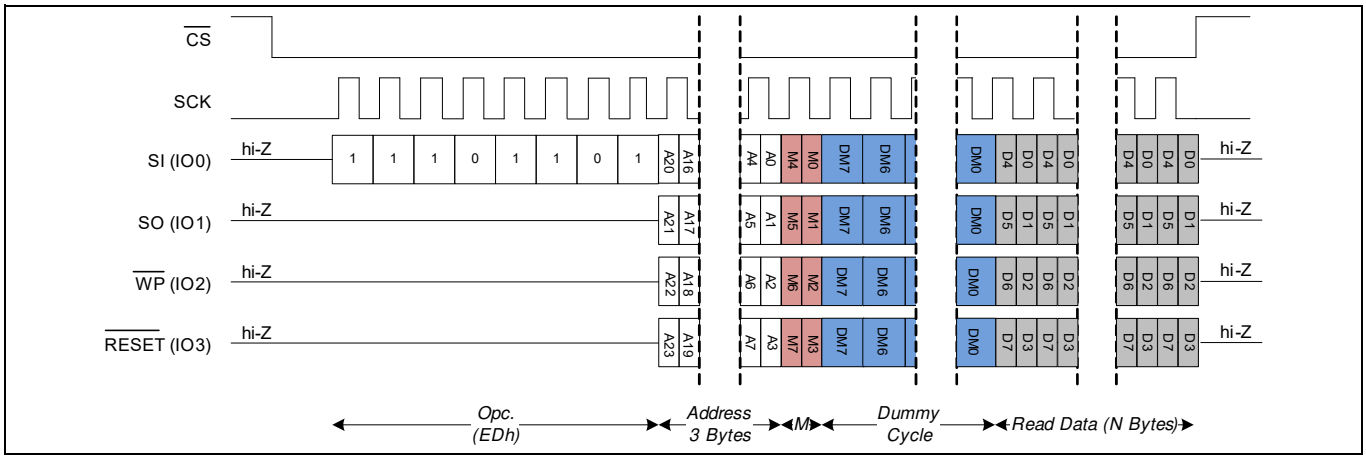



**Figure 64 Quad I/O read in DDR (DDRQIOR) – in extended SPI mode**

### 5.1.5.10 DDR quad I/O read (DDRQIOR EDh) – in QPI mode

The opcode for DDRQIOR can be executed in QSPI mode as well. For DDR quad in/out read (DDRQIOR) in QPI mode, the data is read over (I/O0, I/O1, I/O2, I/O3) in DDR and address and mode bits are also sent over (I/O0, I/O1, I/O2, I/O3) in DDR while the opcode is sent over (I/O0, I/O1, I/O2, I/O3) in SDR.

**Notes**

- Mode bits with !A5h (logical NOT of A5h byte) will exit the DDRQIOR XIP mode.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

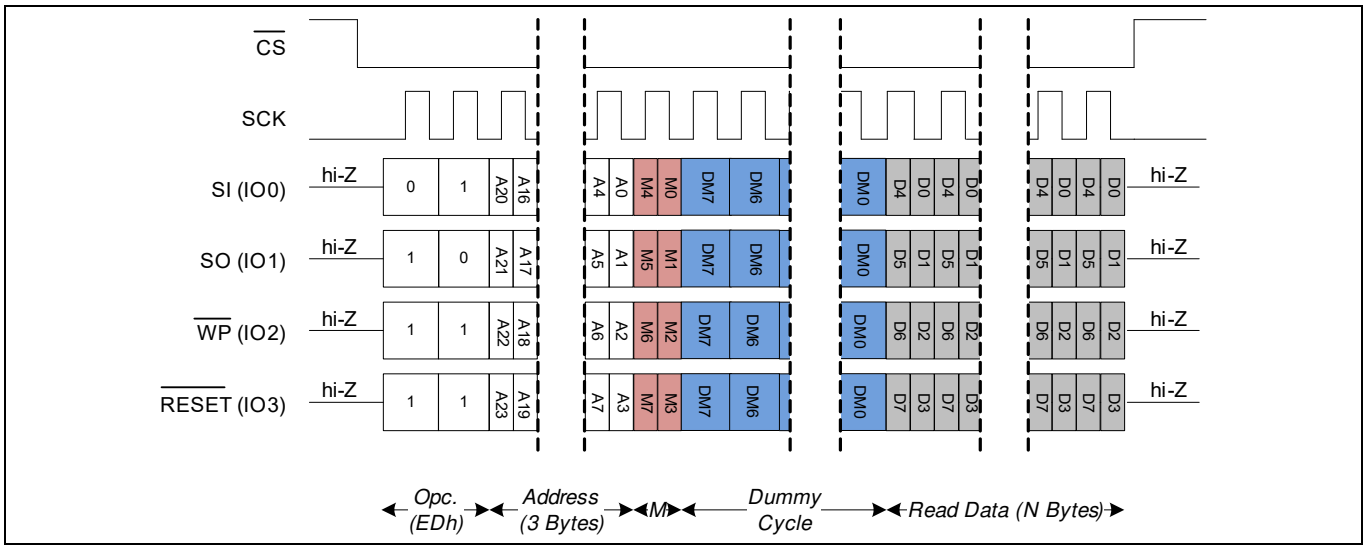



**Figure 65 Quad I/O read in DDR (DDRQIOR) – in QPI mode**

### 5.1.6 Special sector memory access commands

The CY15X102QSN also provides an additional special sector memory region that is 256 bytes in length. This special sector region design for a higher thermal reliability for stored content. Data stored into this special sector can survive up to three standard reflow cycles. This special sector location can be used to store the PCB module details, serial number details, and so on. The special sector memory access commands support the SPI, DPI, and QPI modes of operation.

**Table 43 Special sector memory access commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| SSWR | 42 | Special sector write - dedicated command to write 256 bytes special sector memory |
| SSRD | 4B | Special sector read - dedicated command to read 256 bytes from the special sector memory |

**Table 44 Special sector memory access command details**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | Memory latency | XIP | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Dummy cycles | Execute-in-place | |
| 42 | | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 4B | | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | 108 MHz |

#### 5.1.6.1 Special sector write (SSWR, 42h)

The special sector write operation is preformed when the SSWR opcodes along with write data are given on the SI pin for SPI Mode or the I/O1, I/O0 pins for dual mode (DPI) or the I/O3, I/O2, I/O1, and I/O0 pins for quad mode (QPI). Burst writes can be used to write consecutive addresses without issuing a new SSWR instruction. If only one byte is to be written, the $\overline{CS}$ pin must be driven HIGH after the D0 (LSb of data) is transmitted. However, if more bytes are to be written, the $\overline{CS}$ pin can be held LOW and the address is incremented automatically. The data bytes on the input pin(s) are written in successive addresses. Once the internal address counter auto increments to 0xFF, $\overline{CS}$ should toggle HIGH to terminate the ongoing SSWR operation. Data is written MSb first. The rising edge of $\overline{CS}$ terminates a write operation.

**Notes**

- The three-byte address contains the lower 8-bit for sector address (A7–A0). While the remaining 16 most significant bits of the three-byte address should be set to '0'.
- SSWR instruction can only be executed by the device if the write enable latch (WEL) in SR1 is set to '1' to enable write operations.
- The WEL bit of SR1 (SR1[1]) is automatically cleared to '0' after SSWR command is terminated (at the rising edge of $\overline{CS}$).

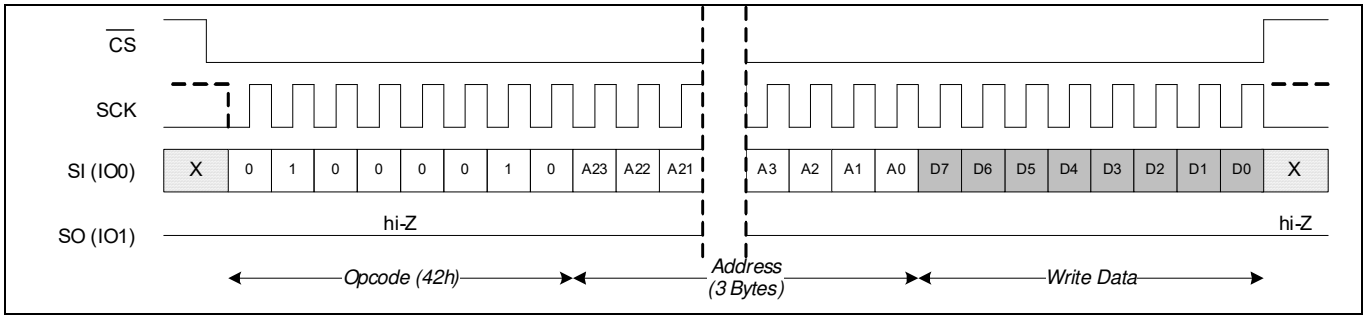



**Figure 66 Special sector write (SSWR) in SPI mode (WREN is not shown)**

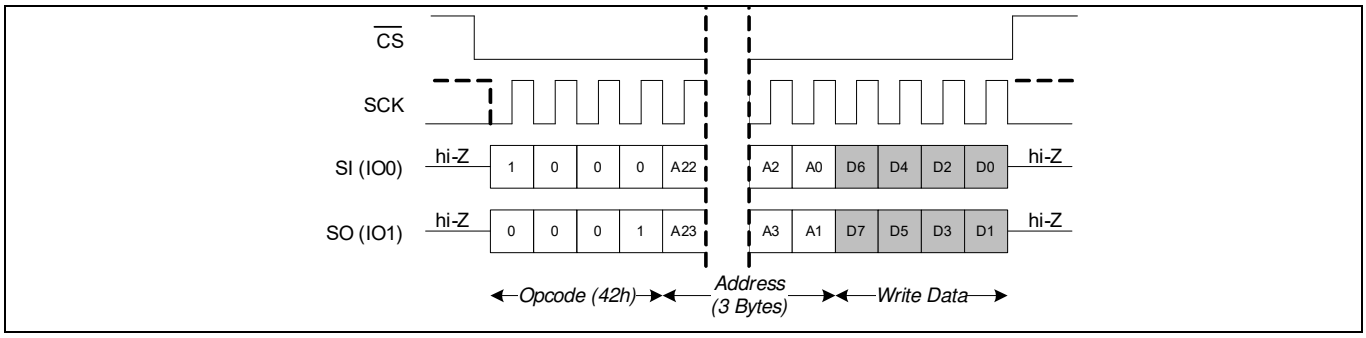



**Figure 67 Special sector write (SSWR) in DPI mode (WREN is not shown)**

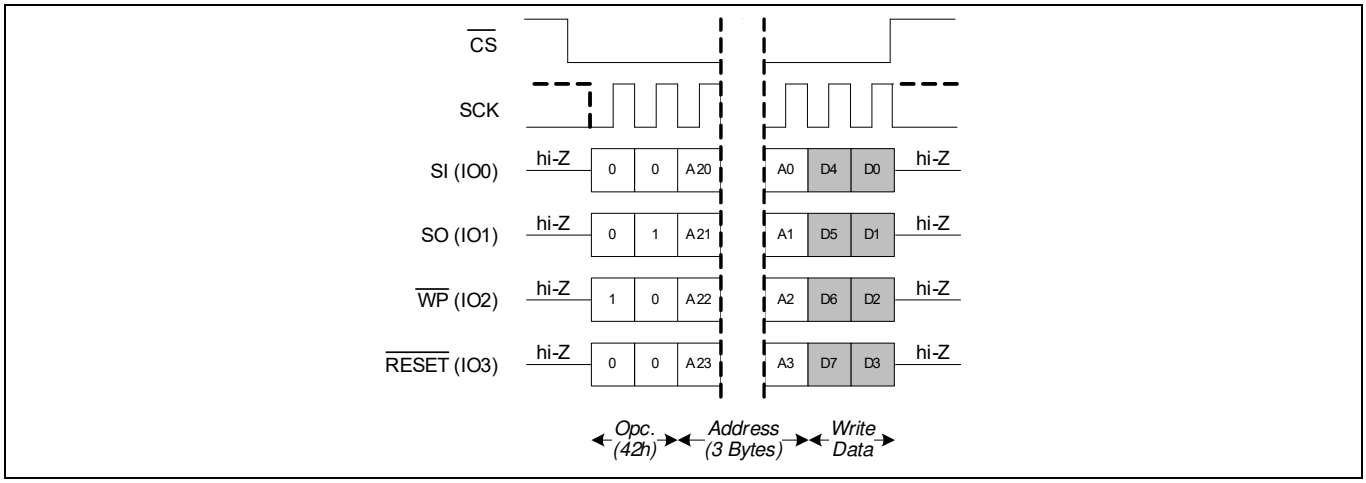



**Figure 68 Special sector write (SSWR) in QPI mode (WREN is not shown)**

### 5.1.6.2 Special sector read (SSRD, 4Bh)

The SSRD instruction reads out the memory contents at the given address. The address can start at any byte location of the 256-byte special sector memory determined by the three-byte address. The address is automatically incremented to the next higher address in sequential order after each byte of data is shifted out. The entire 256-byte special sector can therefore be read out with one single special sector read opcode and address provided. Once the internal address counter auto increments to 0xFF and if the host continues clocking on SCK, the device will return undefined data byte(s).

**Notes**

- The three-byte address contains the lower 8-bit for sector address (A7–A0). While the remaining 16 most significant bits of the three-byte address should be set to '0'.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.
- The special sector F-RAM guarantees to retain user data up to three cycles of standard reflow soldering.

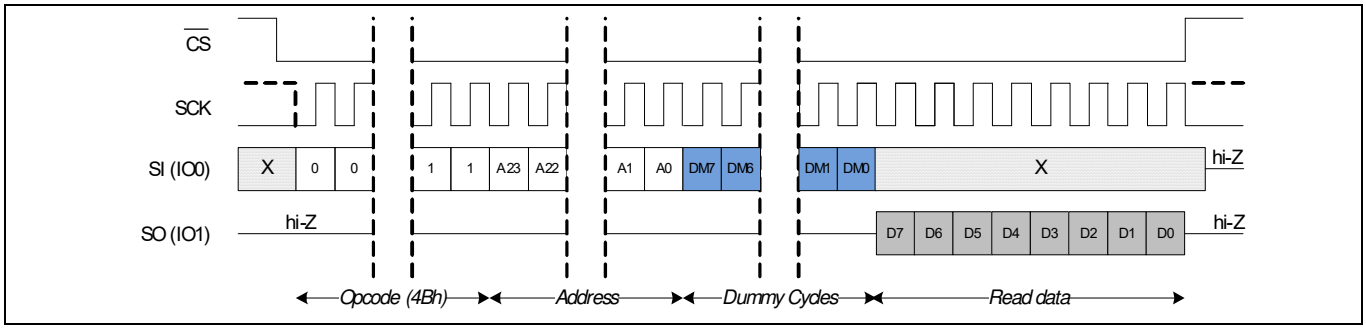



**Figure 69 Special sector read (SSRD) in SPI mode**

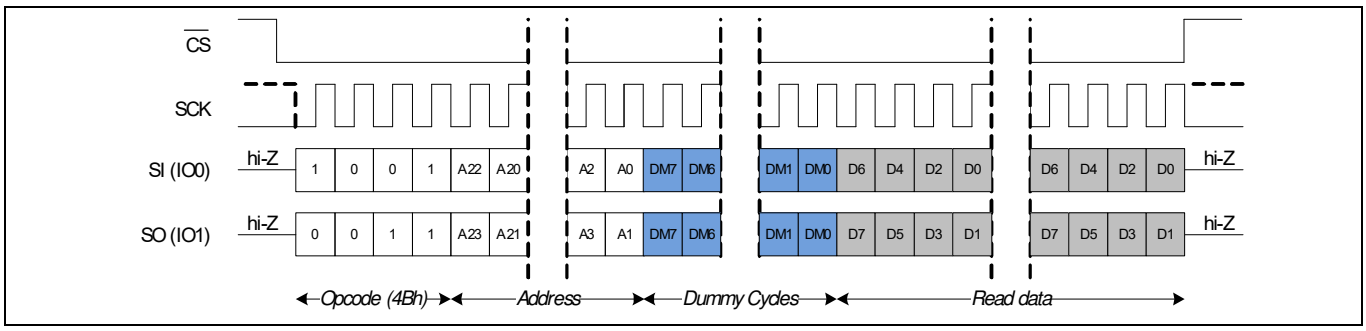



**Figure 70 Special sector read (SSRD) in DPI mode**

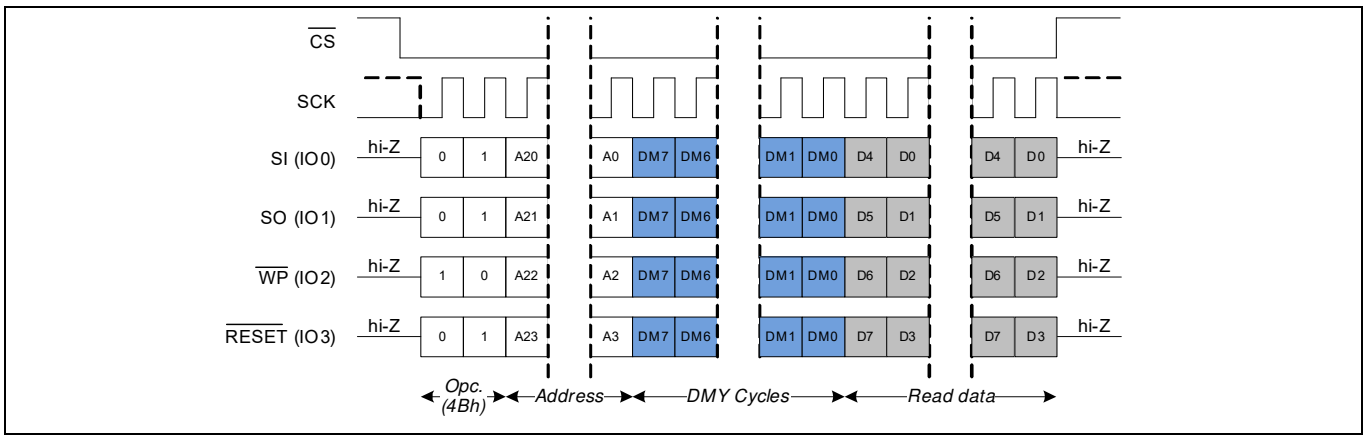



**Figure 71 Special sector read (SSRD) in QPI mode**

## 5.1.7 Error correction code (ECC) and cyclic redundancy check commands

### 5.1.7.1 Error correction code (ECC)

The CY15X102QSN provides an in-built hardware error correction code (ECC) with 2-bit error detection and reporting on an 8-byte (64 bits) unit data. Since every F-RAM read follows a write cycle (refresh cycle), the 1-bit error detected is automatically corrected and written back to the F-RAM array during the refresh cycle. Hence, CY15X102QSN does not report 1-bit error detection because the subsequent ECC check on the same data unit will not reproduce the same 1-bit error. CY15X102QSN ECC is always enabled and observes the following behavior in run time:

- Whenever there is a 2-bit error detected during F-RAM read, CY15X102QSN will set the ECC status register (ECCSR) '2BD' flag bit to '1' (ECCSR is cleared after POR, reset, or CLECC) and also captures the corresponding unit data address in the 4-byte ADDRTRAP register.
- The first three least significant bytes of ADDRTRAP register will hold the 3-byte unit data address of the very first 2-bit error detected in an 8-byte unit data after POR, reset, or CLECC. Any subsequent occurrence of a 2-bit error will not overwrite the ADDRTRAP register with the most recent data unit address.
- CY15X102QSN provides a 2-byte ECC detection count (ECCDC) register which increments by '1' every time a 2-bit error is detected. The ECCDC register is cleared after POR, any reset event, or after CLECC command execution.
- User can read either ADDRTRAP register for its non-zero value (with an exception to where the 2-bit error detected at address 0x00000) or read '2BD' flag bit of ECCSR register, or read the non-zero value in the ECCDC register to determine the occurrence of a 2-bit error detection.
- In addition, CY15X102QSN also supports the ECCRD (19h) command which returns the 2-bit error detection status in 8-byte unit data by setting the '2BD' error flag to '1' in the ECCSR at the unit address sent with the ECCRD command.

ECC is not supported on the 256-byte special sector memory, status and configuration registers.

### 5.1.7.2 ECC status register

The status of ECC is presented in the ECC status register (ECCSR). The ECCSR details are shown in **Table 46**. The ECCSR content can be read only by using the RDAR commands as described in **"Read any register (RDAR, 65h)" on page 44**. The ECCRD command returns the ECCSR status for the unit data. The unit data is defined as the number of bytes over which the ECC is calculated. CY15X102QSN has 8-bytes unit data.

**Table 45 ECC status register**

| ECCSR[7] | ECCSR[6] | ECCSR[5] | ECCSR[4] | ECCSR[3] | ECCSR[2] | ECCSR[1] | ECCSR[0] |
|---|---|---|---|---|---|---|---|
| RFU (0) | RFU (0) | RFU (0) | 2BD (0) | RFU (0) | RFU (0) | RFU (0) | RFU (0) |

**Table 46 ECC status register - volatile only**

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| ECCSR[7] | RFU | Reserved (0) | | | Reserved for future use |
| ECCSR[6] | RFU | Reserved (0) | | | Reserved for future use |
| ECCSR[5] | RFU | Reserved (0) | | | Reserved for future use |
| ECCSR[4] | 2BD | 2-bit ECC detection | V | R | 1 = 2-bit error detection occurred since last ECCSR clear command (CLECC)<br>0 = 2-bit error detection has not occurred since last ECCSR clear command (CLECC) |
| ECCSR[3] | RFU | Reserved (0) | | | Reserved for future use |
| ECCSR[2] | RFU | Reserved (0) | | | Reserved for future use |
| ECCSR[1] | RFU | Reserved (0) | | | Reserved for future use |

**Table 46 ECC status register - volatile only** (continued)

| Bit | Bit name | Bit function | Type | Read/write | Description |
|---|---|---|---|---|---|
| ECCSR[0] | RFU | Reserved (0) | | | Reserved for future use |

V - volatile

### 5.1.7.3 2-bit ECC detection (2BD) ECCSR [4]

This bit indicates that a 2-bit ECC detection has occurred on the read data since the last clear ECC status register. The CLECC instruction resets 2BD bit to '0'.

### 5.1.7.4 ECC detection counter (ECCDC)

The ECC detection counter (ECCDC) register is a 2-byte volatile register, which stores the number of times 2-bit error detections have occurred during the memory read operations since the last POR, any reset event, or after CLECC command. The ECCDC register content can be read by using RDAR commands as described in **"Read any register (RDAR, 65h)"** on page 44.

**Notes**

- Once the ECCDC count reaches 0xFFFF, the ECCDC will stop incrementing.
- The ECCDC loses its content when in deep power-down (DPD) mode and returns with 0x0000 upon DPD exit.

**Table 47 ECC detection counter register (ECCDC)**

| Bits | Name | Function | Type | Read/write | Default state | Description |
|---|---|---|---|---|---|---|
| 15:0 | ECCDC | ECC 2-bit error detection count | V | R | 0x0000 | Total count of 2-bit ECC detections since the last POR or any reset event. CLECC command does not clear this register. |

V - Volatile

### 5.1.7.5 Address trap register (ADDTRAP)

The address trap register (ADDTRAP) is a 4-byte volatile register which stores the ECC unit data address where a 2-bit error detection has occurred during a read operation. The ADDTRAP register stores the address of very first ECC data unit in which 2-bit error detected since the last clear ECC instruction (CLECC), POR, or any reset event. The address of subsequent data unit with 2-bit error detected will not be captured into ADDTRAP. In this case only ECCDC count will increment. The ADDTRAP register content can be read by using the RDAR command as described in **"Read any register (RDAR, 65h)"** on page 44.

**Note:** The ADDTRAP register loses its content when in deep power down (DPD) mode and returns with 0x00000000 upon DPD exit.

**Table 48 Address trap register**

| Bits | Name | Function | Type | Read/write | Default state | Description |
|---|---|---|---|---|---|---|
| 31:0 | ADDTRAP | Stores ECC address | V | R | 0x00000000 | Store address of unit data where 2-bit ECC detection occurred |

V - Volatile

### 5.1.7.6 ECC commands

The CY15X102QSN ECC commands are described in the following section.

**Table 49 ECC commands**

| Command (Hex) | Opcode | Command description |
|---|---|---|
| ECCRD | 19 | ECC status read - Determines the ECC status of the addressed unit data |
| CLECC | 1B | Clear ECC register(s) - ECC flags and address trap registers |

**Table 50 ECC command details**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | Memory latency | XIP | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Dummy cycles | Execute -in-place | |
| 19 | 3 bytes | Yes | NA | | | | Yes | Yes | Yes | NA | Yes | NA | 108 MHz |
| 1B | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |

### 5.1.7.7 ECC status read (ECCRD, 19h)

The ECCRD instruction is used to determine the 2-bit error detection status of the addressed unit data. To do so, $\overline{CS}$ is pulled LOW and the ECCRD instruction is followed by the ECC data unit address in which the three least significant bits (LSb) of address should be set to zero. Even if the least three significant bits (LSb) of address are not set to zero, they will be ignored internally and the start address for the data unit is determined by the rest of the MS bits.

The address bytes are followed by the number of dummy cycles selected by the read latency value for the memory read. The 8-bit ECC status is shifted out on output lines. $\overline{CS}$ must be pulled high after 8-bit ECC status is read out.

**Notes**

- If $\overline{CS}$ remains LOW after 8-bit ECC status is read out, the subsequent ECC status data will be indeterminate. It is necessary to send the new ECCRD command with next unit address to read the ECC status of next data unit.
- The dummy cycles are a configuration option through the memory latency code bits (MLC0 to MLC3) in CR1.

**Table 51 Unit data ECC status byte details**

| Bits | Name | Function | Read/write | Default state | Description |
|---|---|---|---|---|---|
| 7 | RFU | Reserved | – | 0 | Reserved for future use |
| 6 | RFU | Reserved | – | 0 | Reserved for future use |
| 5 | RFU | Reserved | – | 0 | Reserved for future use |
| 4 | RFU | Reserved | – | 0 | Reserved for future use |
| 3 | EECC2D | 2-bit error in ECC unit | R | 0 | 1 = 2-bit error detected in ECC unit<br>0 = no error |
| 2 | RFU | Reserved | – | 0 | Reserved for future use |
| 1 | RFU | Reserved | – | 0 | Reserved for future use |
| 0 | RFU | Reserved | – | 0 | Reserved for future use |

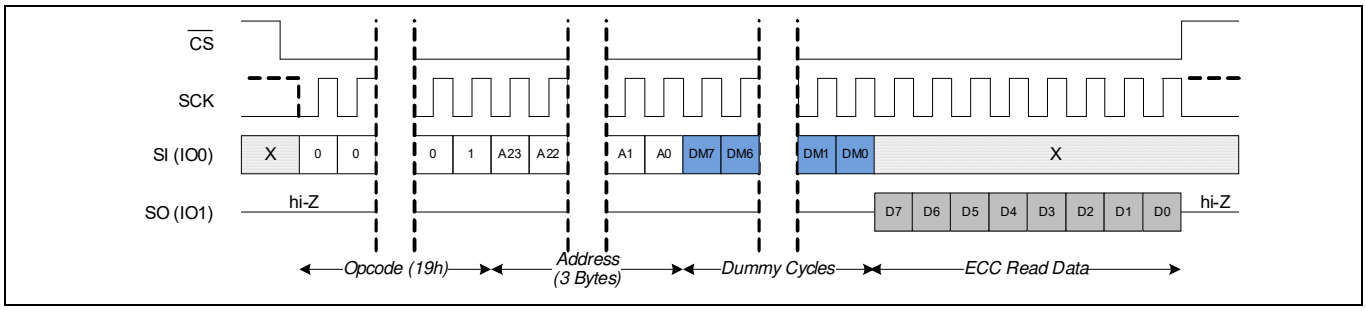



**Figure 72 ECC read (ECCRD) in SPI mode**

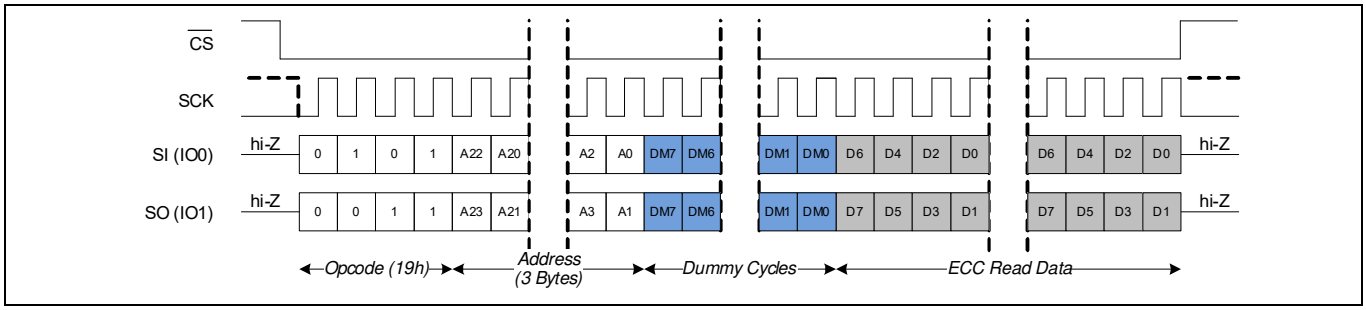



**Figure 73 ECC read (ECCRD) in DPI mode**

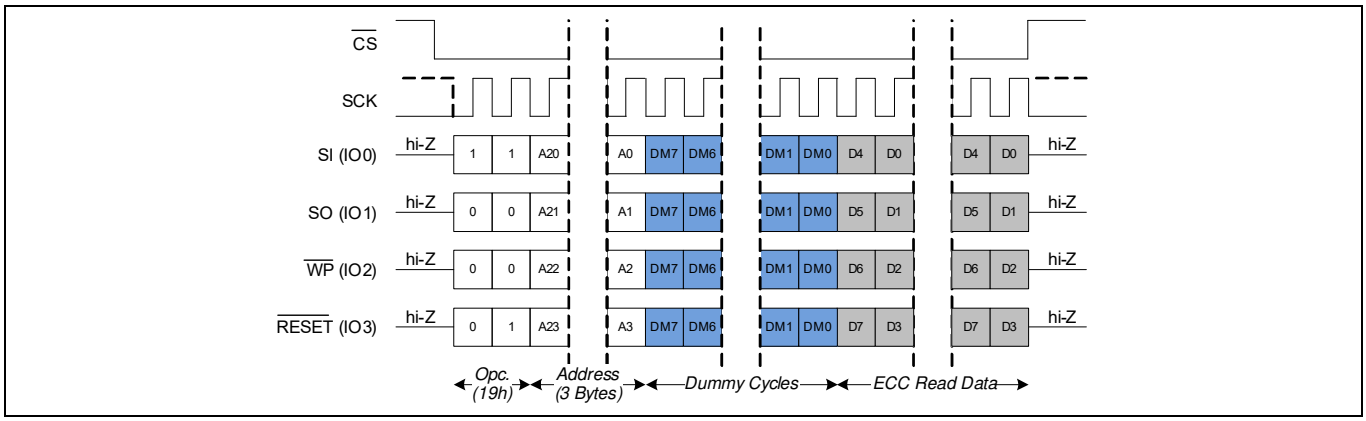



**Figure 74 ECC read (ECCRD) in QPI mode**

### 5.1.7.8 Clear ECC (CLECC, 1Bh)

The CLECC instruction clears all ECC flags, ADDTRAP, and ECCDC registers. It is not necessary to set the WEL bit before a CLECC instruction is executed.

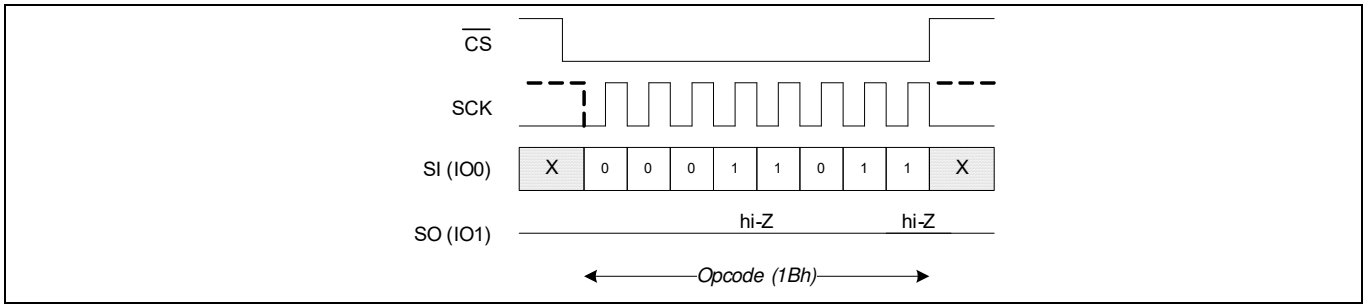



**Figure 75 Clear ECC (CLECC) in SPI mode**

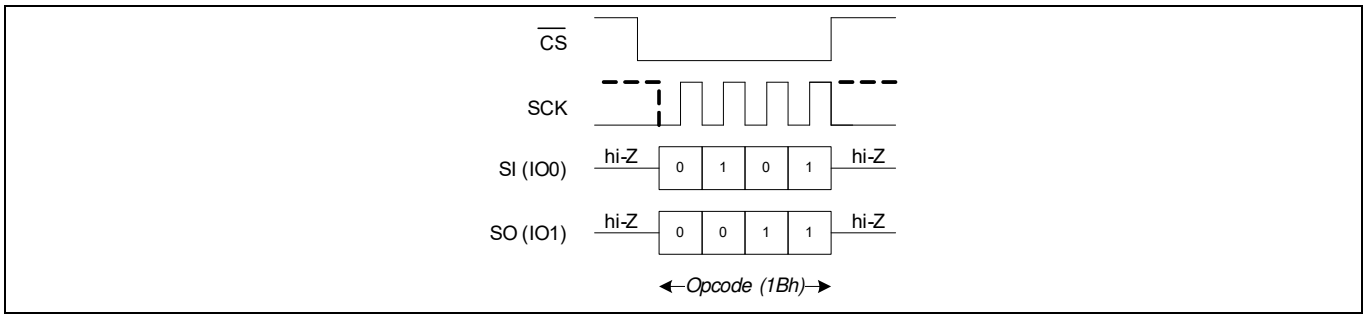



**Figure 76 Clear ECC (CLECC) in DPI mode**

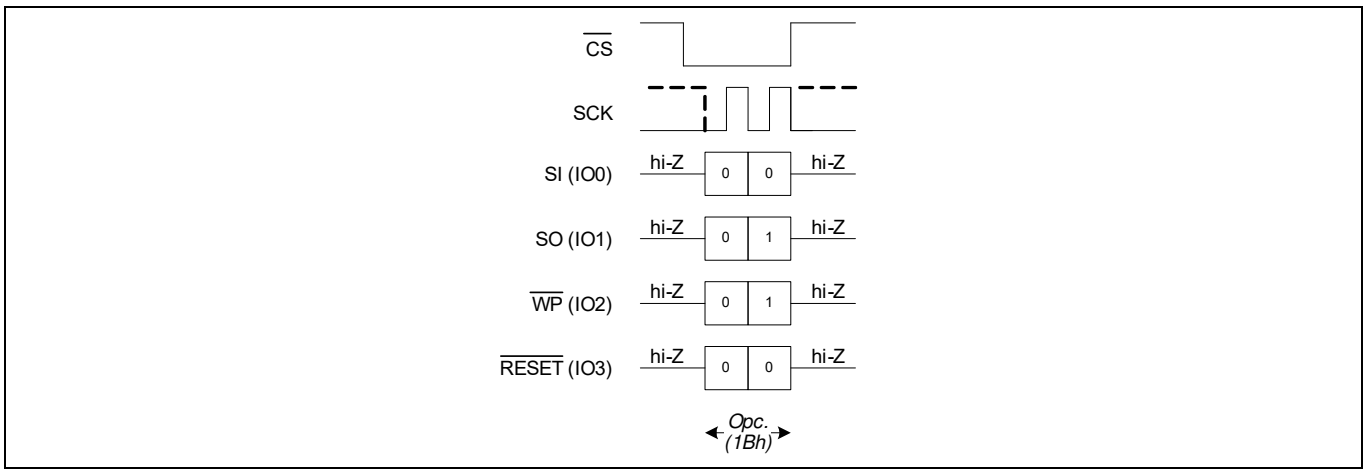



**Figure 77 Clear ECC (CLECC) in QPI mode**

### 5.1.7.9 Cyclic redundancy check (CRC)

CY15X102QSN provides an in-built cyclic redundancy check (CRC) engine that computes the check sequence on the stored data in the memory array. CRC is not supported on 256-byte special sector memory, status and configurations registers.

The CY15X102QSN supports CRC with the following opcodes.

**Table 52 CRC access commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| CRCC | 5B | CRC calculation - performs a CRC calculation over a user defined address range |
| EPCS | 75 | CRC suspend - interrupts the CRCC operation and allow other accesses |
| EPCR1 | 7A | CRC resume - resumes suspended CRCC operation |

**Table 53 CRC access command description**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | Memory latency | XIP | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Dummy cycle | Execute -in-place | |
| 5B | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 75 | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 7A | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |

### 5.1.7.10 Data CRC calculation (CRCC, 5Bh)

The CRCC instruction sequence causes CY15X102QSN to perform a cyclic redundancy check calculation (CRCC) over a user-defined address range. A data CRC-enabled CY15X102QSN device calculates a fixed-length binary sequence, known as the CRC checksum, for each block of data and sends them both together to the host. When the host device receives the data block, it recalculates the CRC checksum. If the new CRC checksum does not match the original checksum sent with the data, then the block contains a data error and the host device may take corrective action such as requesting the data block to be sent again.

The CRCC process calculates the check-value on the data contained at the starting address through the ending address.

The CRC calculation instruction starts by entering the opcode followed by the starting address and ending address. $\overline{CS}$ must be driven HIGH after the ending address has been latched in. This will initiate the beginning of internal CRC process that calculates the check-value on the data contained at the starting address through the ending address. If $\overline{CS}$ is not driven HIGH after the last bit of address, the CRC calculation operation will not be executed. The CRCC command does not check the WEL status. However, if the WEL is set '1' prior to the CRC command, the WEL gets cleared to '0' after the CRC operation is complete.

The ending address (EA) should be at least a 32-bit aligned word higher than the starting address (SA). If EA < SA + 3, the CRC Calculation will abort and the device will return to the standby mode. The CRC abort (CRCA) bit (SR2[3] = '1') is set to indicate the aborted condition and the CRC register (CRCR) will hold indeterminate data.

When the CRC calculation is in progress, CY15X102QSN sets the WIP bit of SR1 (SR1[0]) to '1'. User can poll the WIP status to determine when the ongoing CRCC operation is complete and device is ready for access. The WIP bit will be '1' when the CRC calculation is in progress and a '0' when it has been completed. The CRC register (CRCR) stores the results of the CRC process that calculates the check-value on the data contained at the starting address through the ending address. The details of the CRC register is described in **Table 54**. The CRC check-value bits 0–31 can be read by reading the CRC register using read any register (RDAR) command as described in **"Read any register (RDAR, 65h)"** on page 44.

The CRC register bits are initialized with all 0s (0x00000000) every time CRC calculation is initiated. A POR or any reset event will also initialize the CRC register value to all 0s.

The check-value calculation can be suspended with the CRC suspend command (EPCS, B0h) to read data from the memory array or registers. During the suspended state, the CRC suspend (CRCS) status bit in status register-2 will be set (SR2[4] = '1'). Once suspended, the host can read the status register, read data from the array and can resume the CRC calculation by using the CRC resume command (EPCR, 30h). CY15X102QSN takes $t_{CRCC}$ to calculate the CRC checksum on data between the SA and EA (including data at SA and EA).

The 32-bit CRC (CRC-32C) polynomial (0x1EDC6F41) is defined as follows:

32X + 28X + 27X + 26X + 25X + 23X + 22X + 20X+ 19X + 18X + 14X + 13X + 11X + 10X + 9X + 8X + 6X + 1X

**Note:** 4-byte memory data are internally read as {data[7:0], data[15:8], data[23:16], data[31:24]} and are assigned to CRC[31:0] for the CRC calculation.

**Table 54 CRC register description**

| Bits | Name | Function | Default state | Description |
|---|---|---|---|---|
| 31:0 | CRCR | Check CRC value | 0x00000000 | Volatile register to store the CRC checksum value resulted after the CRC calculation (CRCC command). |

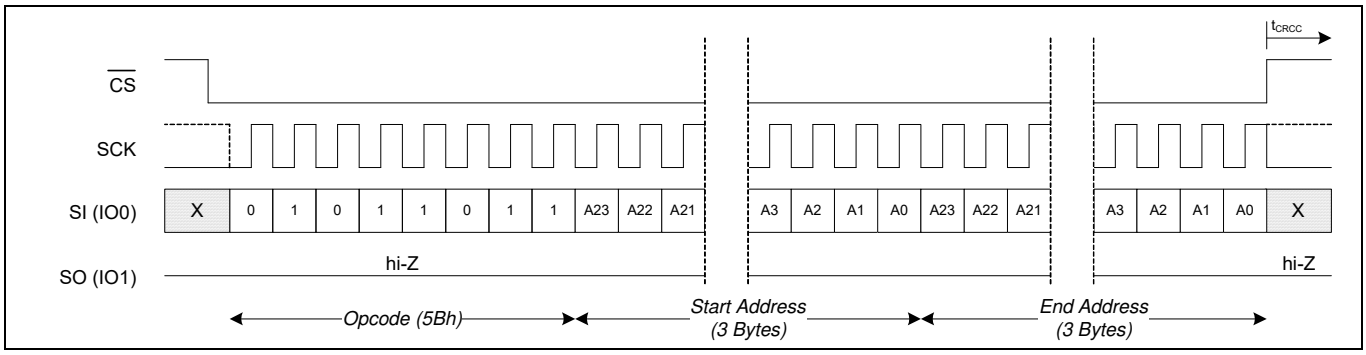



**Figure 78 CRC calculation (CRCC) in SPI mode**

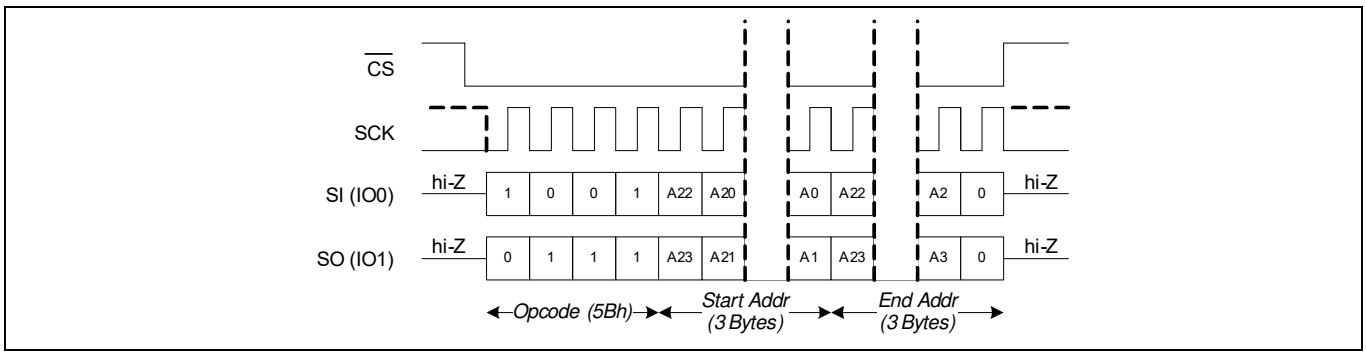



**Figure 79 CRC calculation (CRCC) in DPI mode**

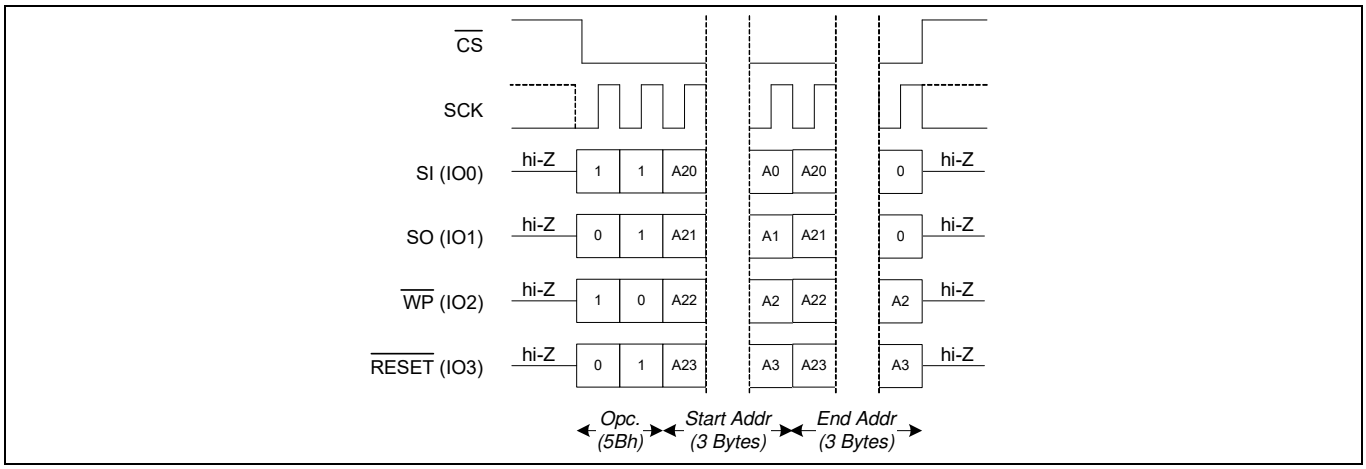



**Figure 80 CRC calculation (CRCC) in QPI mode**

## 5.1.7.11 CRC suspend (EPCS, 75h)

EPCS allows the system to interrupt the ongoing CRCC operation and allow other accesses while the current CRC operation is suspended. Commands which can execute while CRC is suspended are: READ, RDSR1, RDSR2, FAST_READ, DDRFR, ECCRD, CLECC, RDCR1, DOR, RDCR2, RDCR4, SSRD, RDCR5, RDAR, RSTEN, QOR, EPCR, RST, RDID, DIOR, RDSN, QIOR, DDRQIOR.

The CRC suspend is valid only during a CRC calculation operation. The status register 2 (SR2) can be checked to determine if the CRCC operation has been suspended or completed. The CRC status bit shows if a CRCC operation is suspended or was completed at the time WIP status bit in status register 1 changes to '0'. EPCS takes $t_{CRCS}$ time to process the CRC suspend operation and keeps the WIP bit status '1'. In the case CRCC calculation completes before the EPCS command is fully processed, the CRCS bit in SR2 (SR2 [4]) will not set to '1', indicating EPCS did not execute.

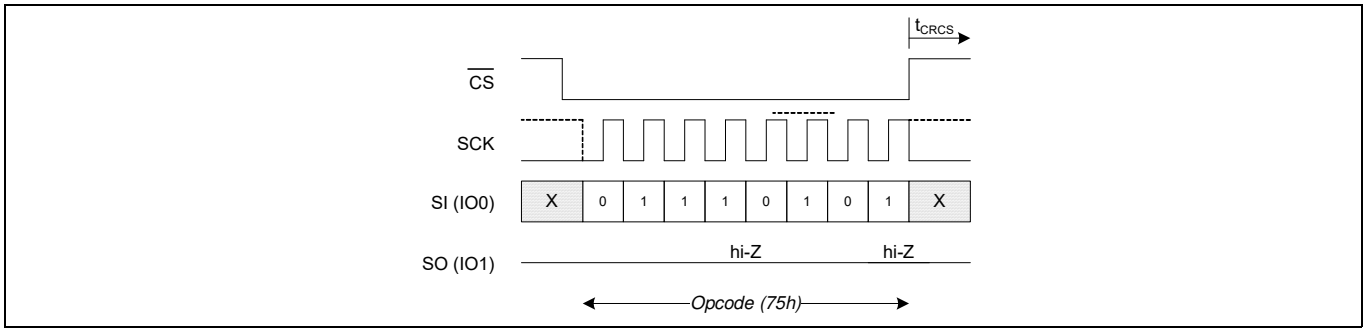



**Figure 81 CRC suspend (EPCS) in SPI mode**

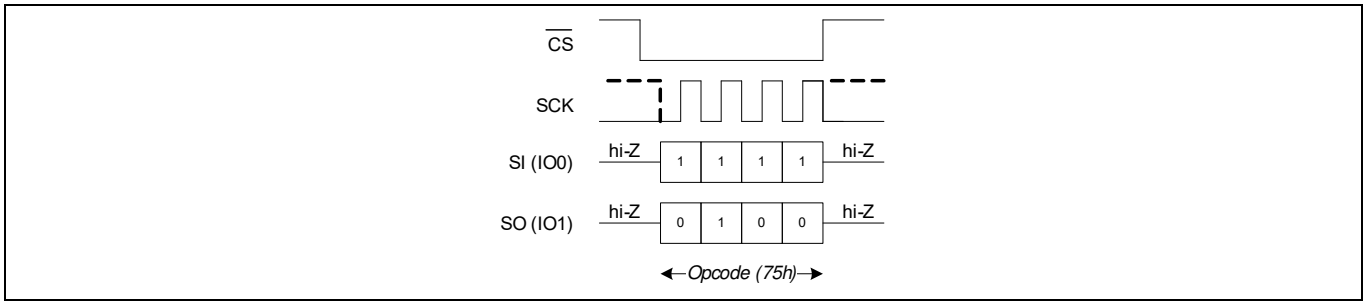



**Figure 82 CRC suspend (EPCS) in DPI mode**

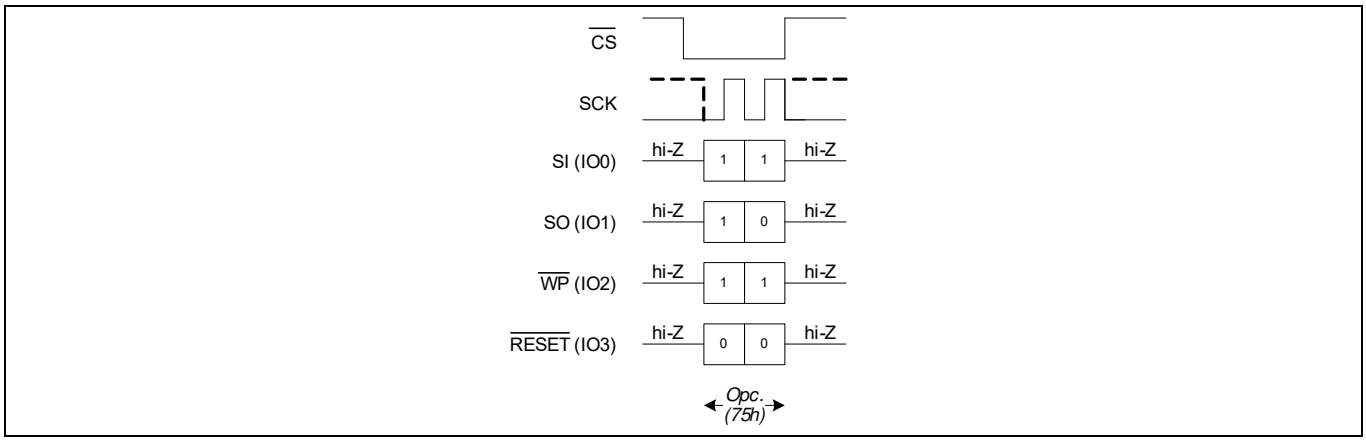



**Figure 83 Suspend (EPCS) in QPI mode**

## 5.1.7.12 CRC resume (EPCR, 7Ah)

EPCR resumes a suspended CRCC operation. After the CRC resume instruction is issued, the WIP bit is set to '1'. The CRCC operation can be interrupted as often as necessary. The EPCR resumes a suspended CRCC operation only when CRCS bit of SR2 (SR2[4]) is set '1', otherwise EPCR command will be ignored. After the EPCR instruction is issued, the WIP bit is set to '1'. The CRCC operation can be interrupted and resumed as often as necessary.

EPCR takes $t_{CRCR}$ time to process the command and resumes the CRC calculation on the remaining data bytes, until the end address (EA) reaches.

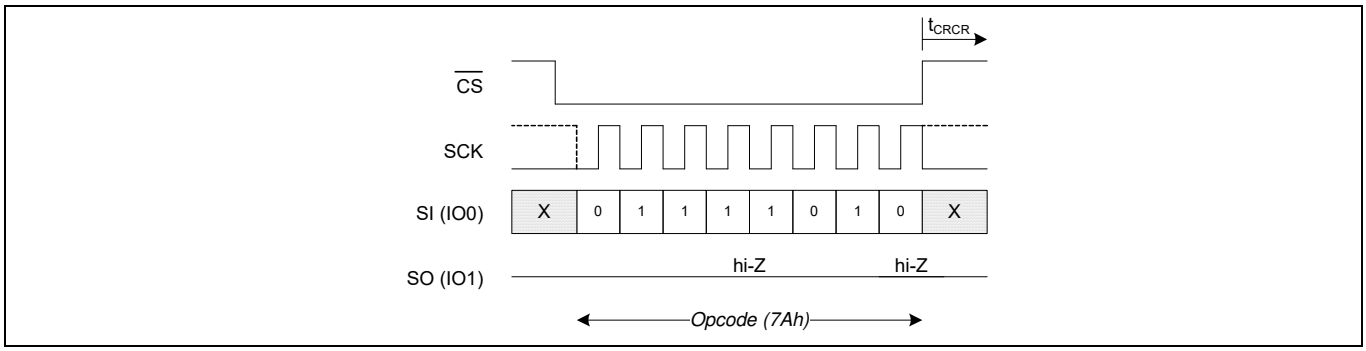



**Figure 84 CRC resume (EPCR) in SPI mode**

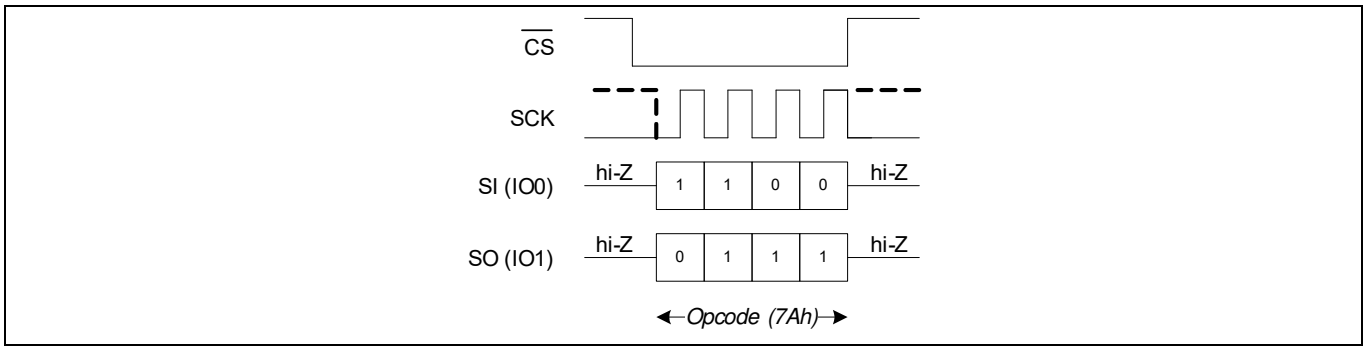



**Figure 85 CRC resume (EPCR) in DPI mode**

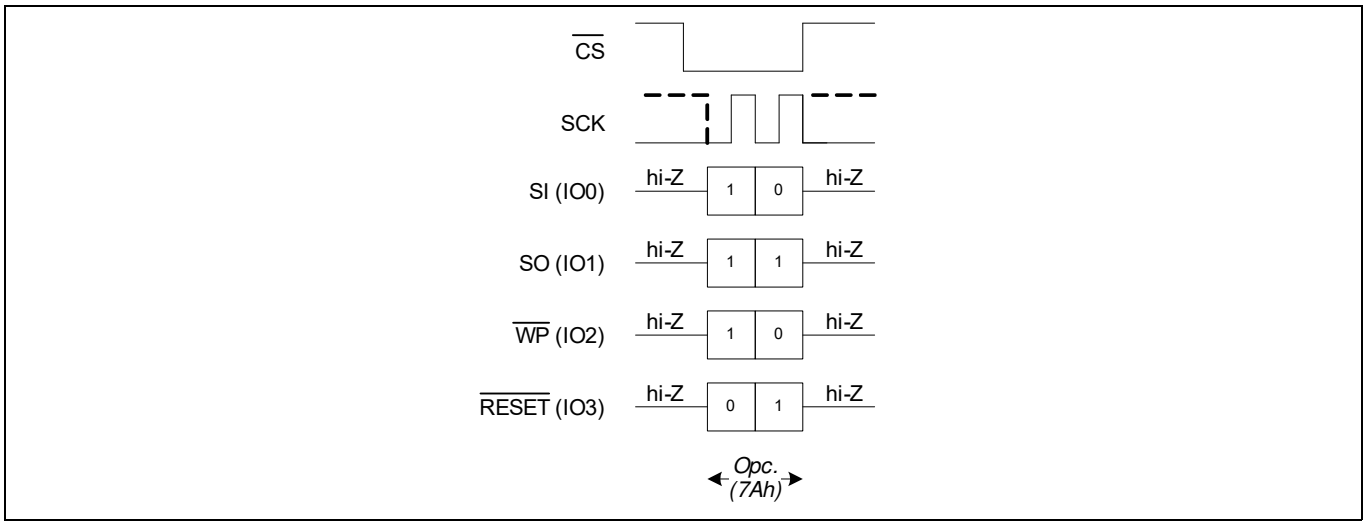



**Figure 86 CRC resume (EPCR) in QPI mode**

## 5.1.8 Identification and serial number commands

The CY15X102QSN device offers three different types of identification features that include device ID and unique ID which are 8-byte read only registers and 8-byte writable serial number registers. Details of each is described in the following section.

### 5.1.8.1 Read device ID (RDID, 9Fh)

The CY15X102QSN device can be interrogated for its manufacturer, product identification, and die revision. The RDID opcode 9Fh allows the user to read the 8-byte manufacturer ID and product ID, both of which are read-only bytes. The device ID field is described in the device ID Field register table. The device ID of the corresponding part number is shown in the **"Ordering information"** on page 109.

**Notes**

- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.
- RDID data preference - LSb shifts out first, MSb shifts out last. No wrap is allowed for the RDID command. After the 8th byte, if the host continues to clock the device will return undefined data byte/s.

**Table 55 Device ID field**

| Bits (number of bits) | 63–32 (32 bits) | 31–21 (11 bits) | 20–8 (13 bits) | 7–3 (5 bits) | 2–0 (3 bits) |
|---|---|---|---|---|---|
| Description | 00000000000000000000000000000000 (Reserved) | 00000110100 (Manufacturer ID) | Product ID | Density ID | Die rev |

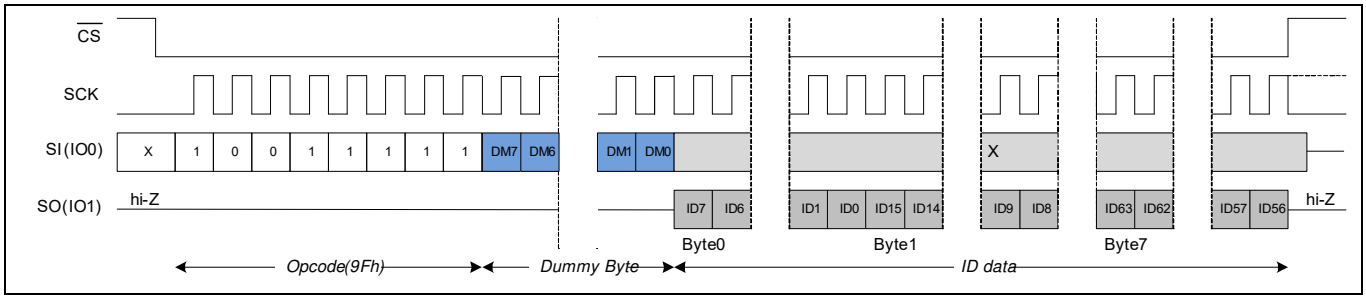



**Figure 87 Read device ID (RDID) in SPI mode**

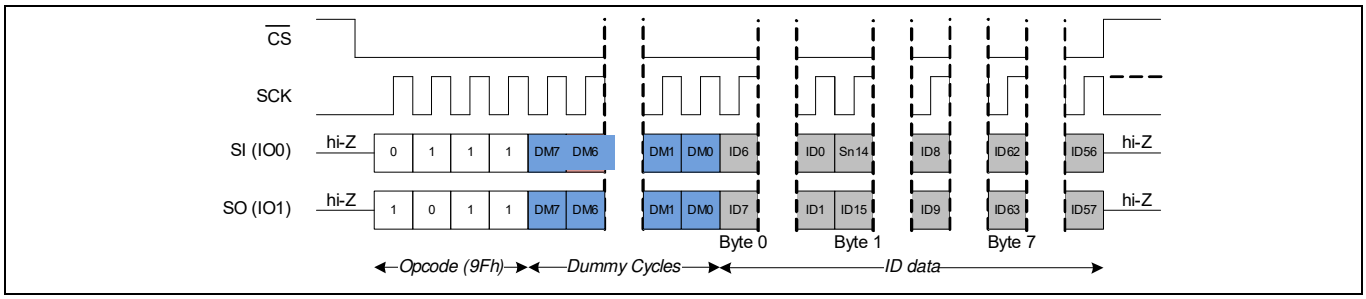



**Figure 88 Read device ID (RDID) in DPI mode**

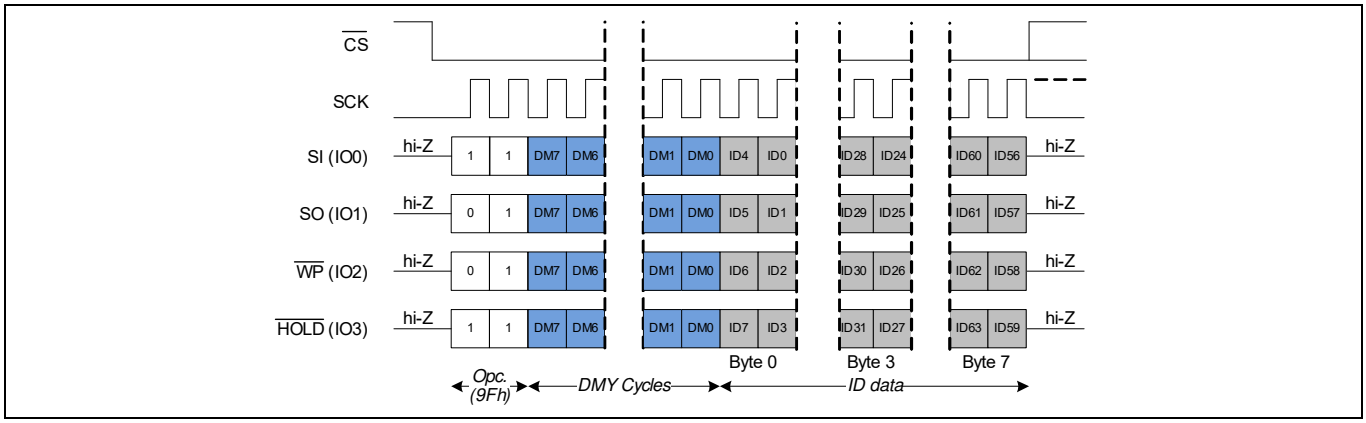



**Figure 89 Read device ID (RDID) in QPI mode**

## 5.1.8.2 Read unique ID (RUID, 4Ch)

The CY15X102QSN device can be interrogated for unique ID which is a factory programmed, 64-bit number unique to each device. The RUID opcode, 4Ch allows to read the 8-byte, read only unique ID.

**Notes**

- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.
- RUID data preference - LSb shifts out first, MSb shifts out last. No wrap is allowed for the RDID command. After the 8th byte, if the host continues to clock, the device will return undefined data byte(s).
- The unique ID registers guarantee to retain user data up to three cycles of standard reflow soldering.

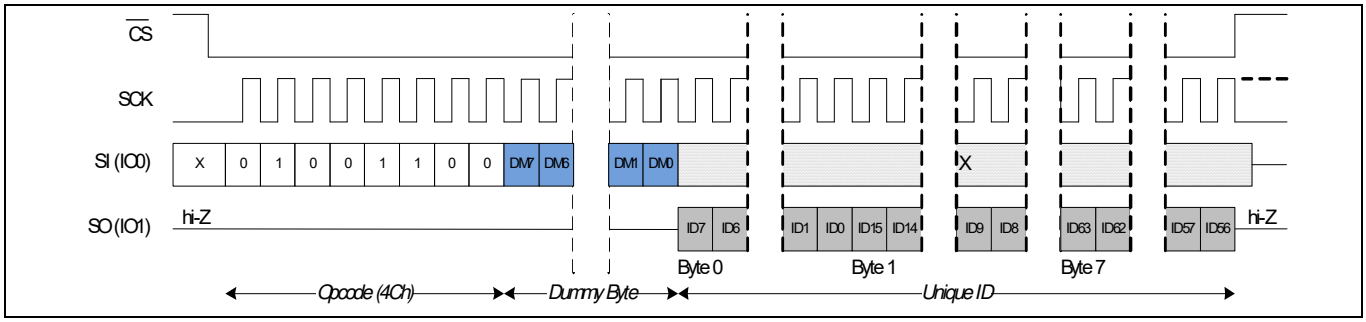



**Figure 90 Read unique ID in SPI mode**

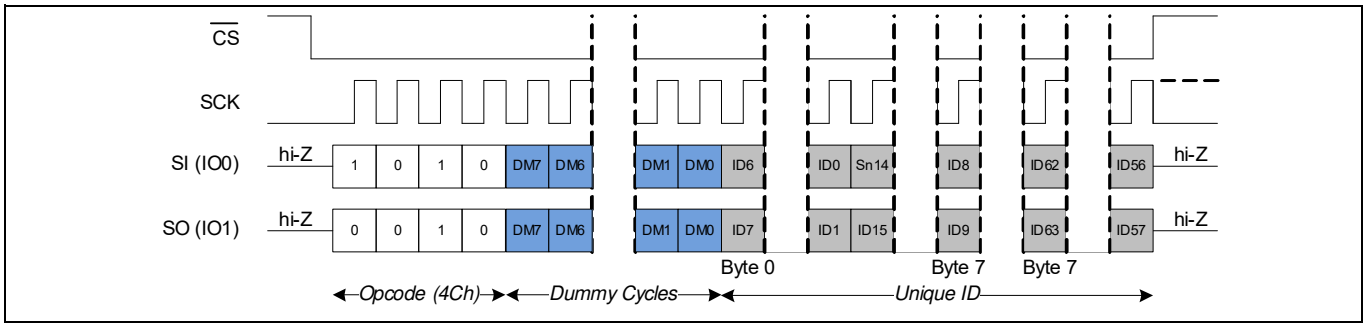



**Figure 91 Read unique ID in DPI mode**

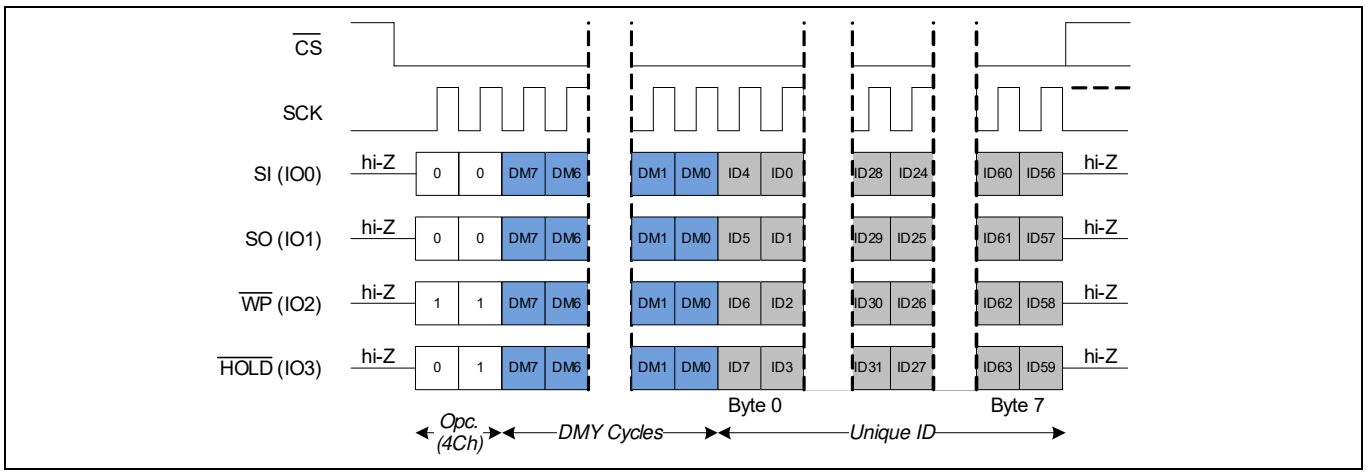



**Figure 92 Read unique ID in QPI mode**

### 5.1.8.3 Write serial number (WRSN, C2h)

The serial number is an 8-byte programmable memory space provided to the user to uniquely identify a pc board or a system. A serial number typically consists of a two byte customer ID, followed by five bytes of unique serial number and one byte of CRC check. However, end application can define their own format for 8-byte serial number. All writes to the serial number register begin with a WREN opcode with $\overline{CS}$ being asserted and de-asserted. The next opcode is WRSN. The WRSN instruction can be used in burst mode to write all the 8 bytes of serial number. After the last byte of serial number is shifted in, $\overline{CS}$ must be driven HIGH to complete the WRSN operation.

**Notes**

- The WRSN instruction can only be executed by the device if the write enable latch (WEL) in the status register is set to '1' to enable write operations. When the WRSN operation is completed, the write enable latch (WEL) is reset to a '0'.
- WRSN data preference - LSb shifts in first, MSb shifts in last.
- The CRC checksum on the 7-byte ID is not calculated by the device. The system firmware must calculate the CRC checksum and append the checksum to the 7-byte user defined serial number before programming the entire 8-byte serial number into the serial number register. Factory default value for the 8-byte serial number is '0x0000000000000000'.
- The WEL bit is automatically cleared to '0' after WRSN command is terminated (at the rising edge of $\overline{CS}$).
- Exactly 8 bytes must be entered, otherwise the serial number write (WRSN) will not execute.

**Table 56 8-byte serial number**

| 16-bit customer identifier | | 40-bit unique number | | | | | 8-bit CRC |
|---|---|---|---|---|---|---|---|
| SN[63:56] | SN[55:48] | SN[47:40] | SN[39:32] | SN[31:24] | SN[23:16] | SN[15:8] | SN[7:0] |

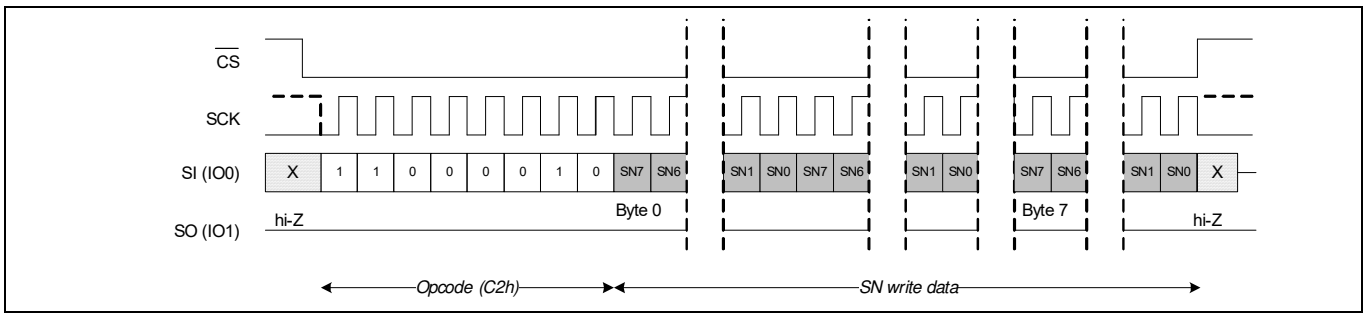



**Figure 93 Write serial number in SPI mode (WREN not shown)**

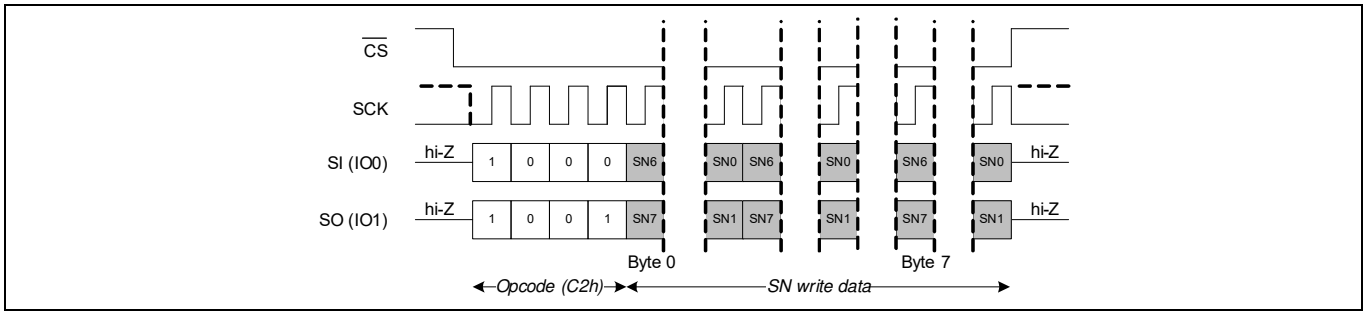



**Figure 94 Write serial number in DPI mode (WREN not shown)**

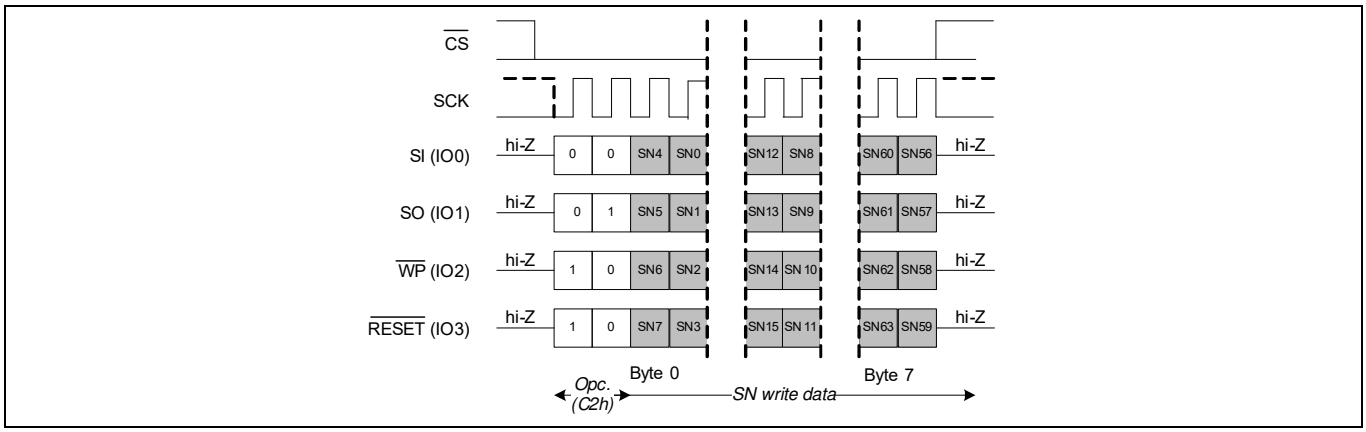



**Figure 95 Write serial number in QPI mode (WREN not shown)**

### 5.1.8.4 Read serial number (RDSN, C3h)

The CY15X102QSN device incorporates an 8-byte serial space provided to the user to uniquely identify the device. The serial number is read using the RDSN instruction. A serial number read may be performed in burst mode to read all the eight bytes at once. After the last byte of serial number is read, the host must stop clocking and drive the $\overline{CS}$ HIGH to terminate the RDSN command. An RDSN instruction can be issued by shifting the opcode for RDSN after $\overline{CS}$ goes LOW.

**Notes**

- The dummy cycles shown are a configuration option through register latency code bits (RLC0, RLC1) in CR5.
- LSb shifts out first, MSb shifts out last. If the host continues clocking after 8th byte, the device may return undefined data byte/s.

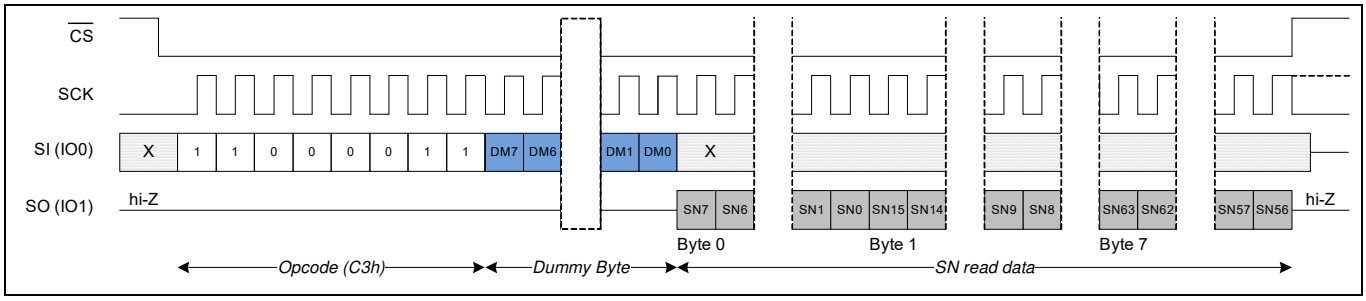



**Figure 96 Read serial number (RDSN) in SPI mode**

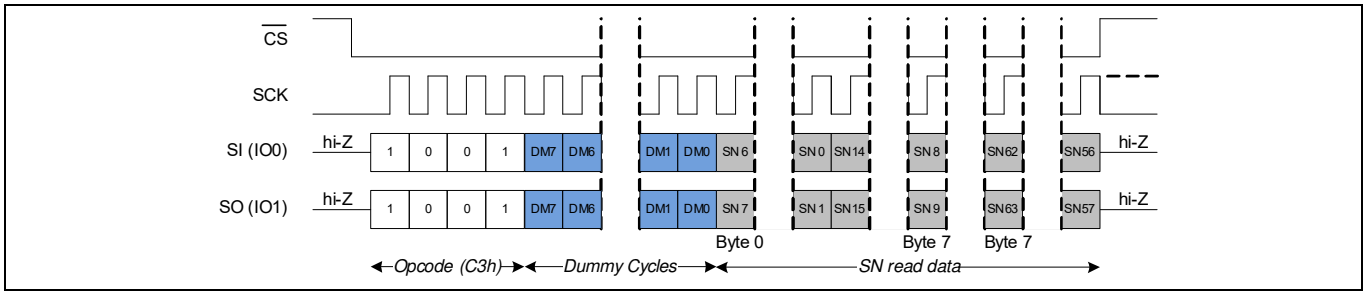



**Figure 97 Read serial number (RDSN) in DPI mode**

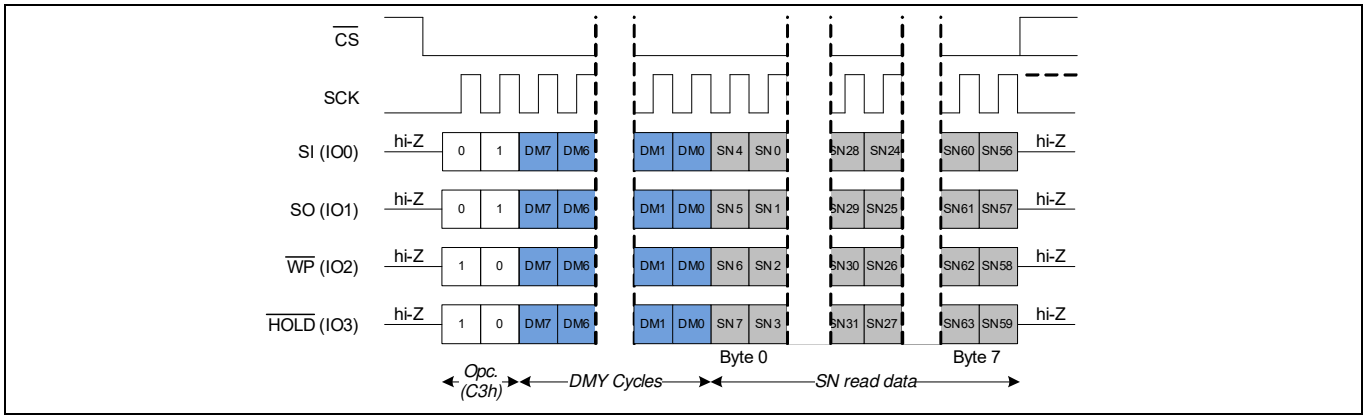



**Figure 98 Read serial number (RDSN) in QPI mode**

### 5.1.9 Low power modes and resets

**Table 57 Low power mode and reset commands**

| Command | Opcode (Hex) | Command description |
|---|---|---|
| DPD | B9 | Deep power down - enters deep-power-down power mode |
| HBN | BA | Hibernate mode - enters hibernate power mode |
| RSTEN | 66 | Reset enable - pre command to enable software reset |
| RST | 99 | Software reset - command to initiate software reset |

**Table 58 Low power mode and reset command description**

| Opcode (Hex) | Address length | SPI bus interface | | | | | | | Data transfer | | Latency (none) | XIP | Max clock frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SPI | Dual data | Quad data | Dual I/O | Quad I/O | DPI | QPI | SDR | DDR | Dummy cycles | Execute -in-place | |
| B9 | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| BA | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 66 | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |
| 99 | NA | Yes | NA | | | | Yes | Yes | Yes | NA | NA | NA | 108 MHz |

#### 5.1.9.1 Deep power-down mode (DPD, B9h)

The device enters deep power down mode when the DPD opcode B9 is clocked in and a rising edge of $\overline{CS}$ is applied. When in deep power-down mode, the SCK and SI pins are ignored and SO goes to hi-Z, but the device continues to monitor the $\overline{CS}$ pin.

A $\overline{CS}$ pulse-width of $t_{CSDPD}$ or hardware reset exits the DPD mode after $t_{EXTDPD}$ time. The $\overline{CS}$ pulse-width can be generated either by sending a dummy command cycle or toggling $\overline{CS}$ alone while SCK and I/Os are don't care. The I/Os remain in hi-Z state during the wakeup from deep power down. Refer to **Figure 99** for DPD entry and **Figure 102** for DPD exit timing.

**Notes**

- The timing details shown in the **Figure 99** are applicable as is in DPI and QPI modes.
- CRC register (CRCR) and ECC registers (ECCDC and ADDRTRAP) will lose their content in the DPD mode and will return to their default values, 0x00.
- The WEL bit (SR0[1]) status is not retained in the DPD mode. If the WEL status was '1' before entering DPD, it will clear to '0' after the DPD mode exits.

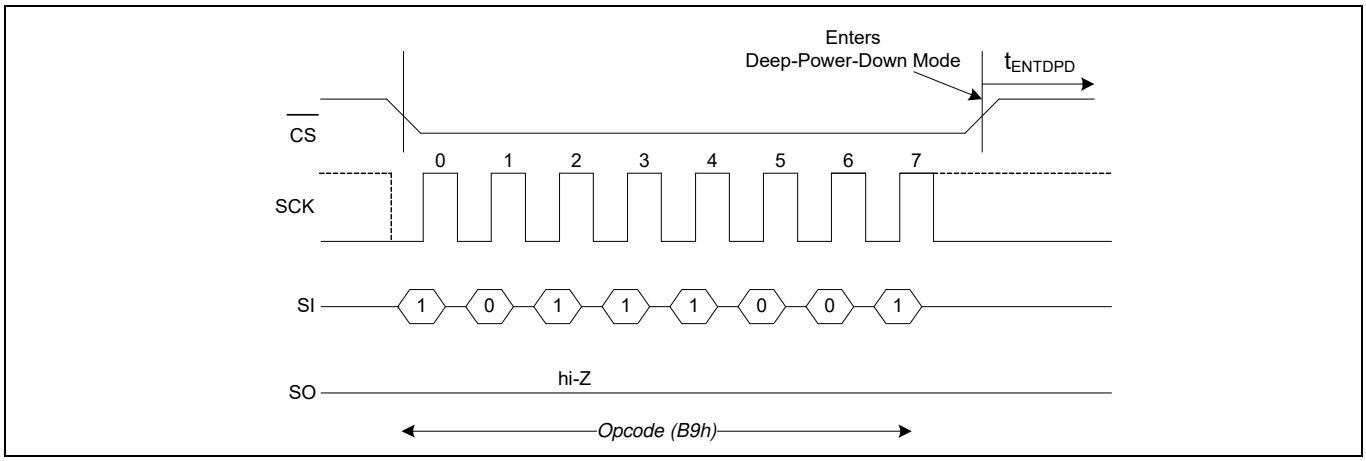



**Figure 99 DPD entry in SPI mode**

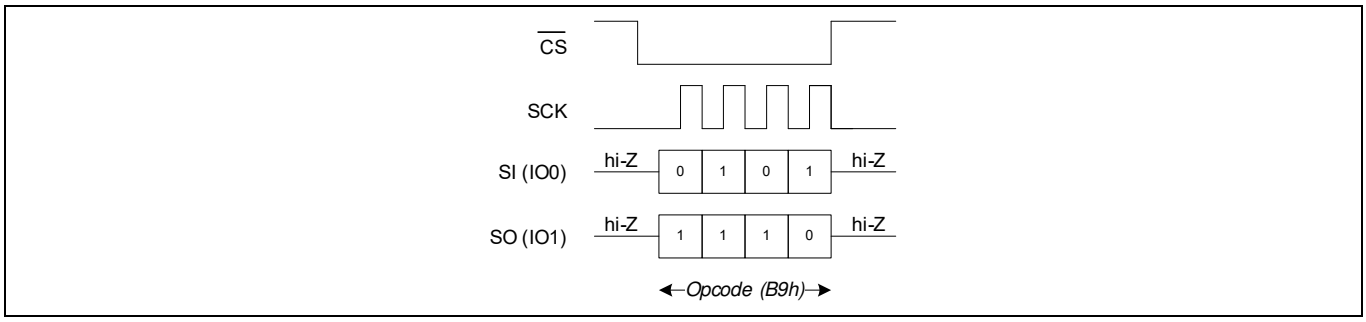



**Figure 100 Deep power-down mode operation in DPI mode**

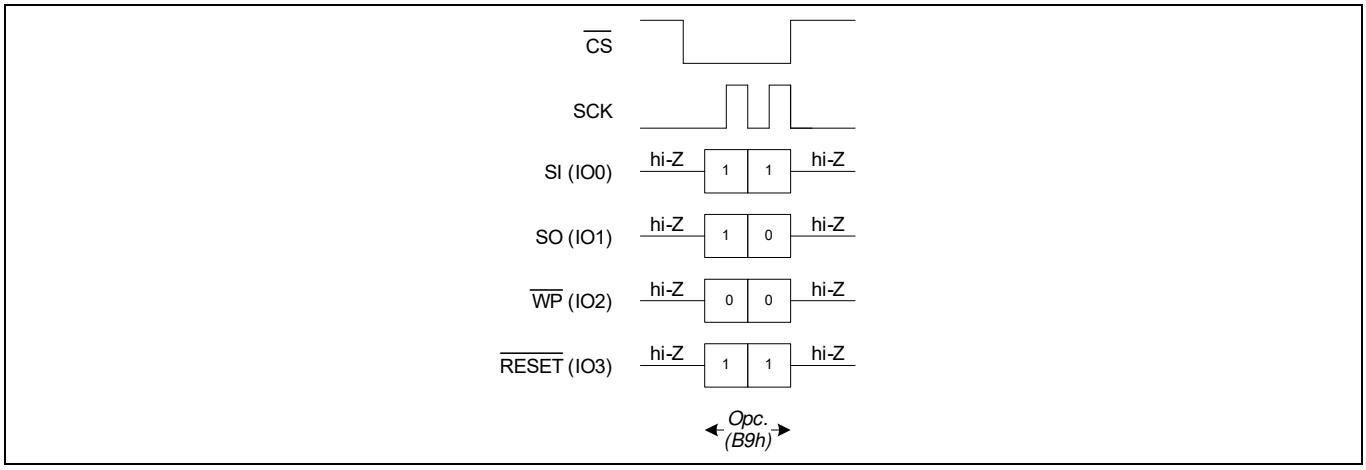



**Figure 101 Deep power-down mode operation in QPI mode**

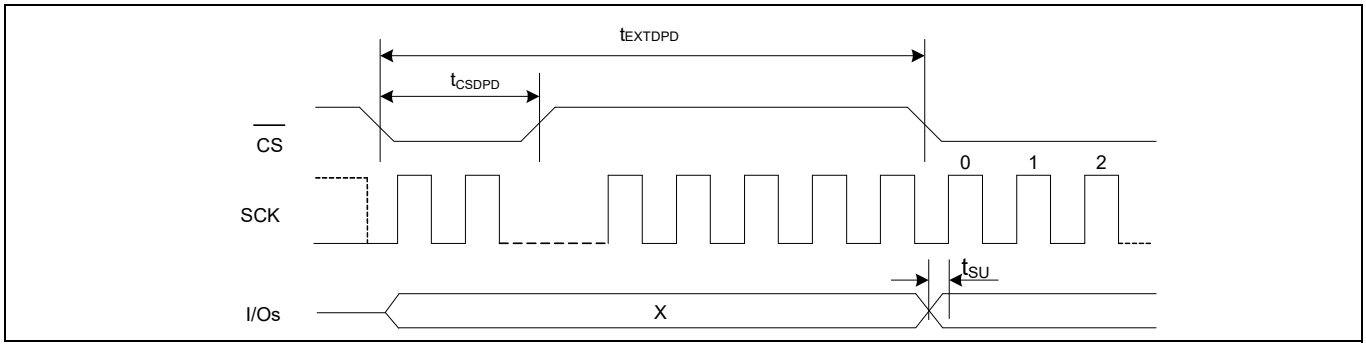



**Figure 102 DPD exit in SPI mode**

### 5.1.9.2 Hibernate mode (HBN, BAh)

The device enters hibernate mode when the HBN opcode BAh is clocked in and a rising edge of $\overline{CS}$ is applied. When in hibernate mode, the SCK and SI pins are ignored and SO will be hi-Z, but the device continues to monitor the $\overline{CS}$ pin. On the next falling edge of $\overline{CS}$, the device will return to normal operation within $t_{EXTHIB}$ time. The SO pin remains in a hi-Z state during the wakeup from hibernate period. The device does not necessarily respond to an opcode within the wakeup period. To exit the hibernate mode, the controller may send a “dummy” read, for example, and wait for the remaining $t_{EXTHIB}$ time.

**Notes**

- The timing details shown in the SPI mode timing diagram are applicable as is in the DPI and QPI modes.
- Return from hibernate reloads all registers to their default POR values. Refer to **Table 3** for details on registers default values after POR.

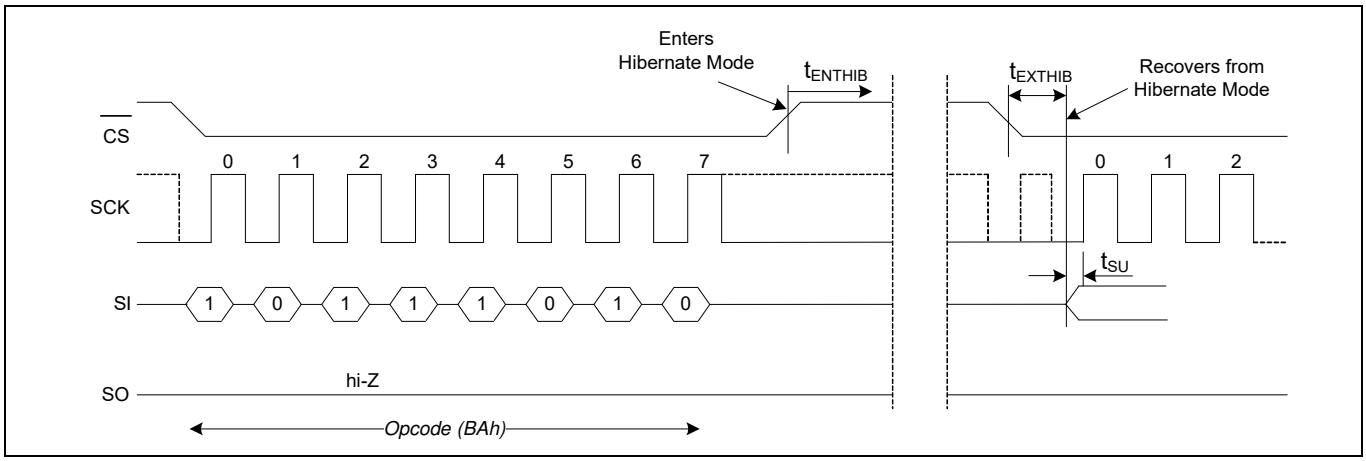



**Figure 103 Hibernate mode operation in SPI mode**

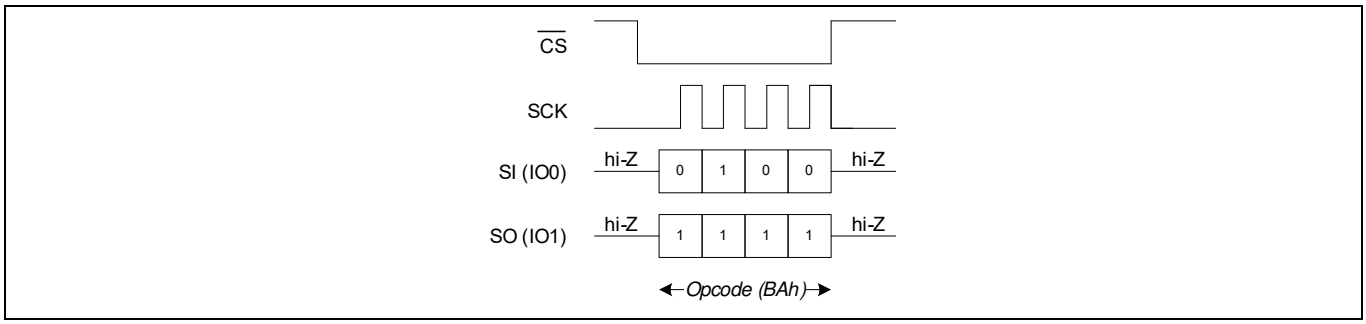



**Figure 104 Hibernate mode operation in DPI mode**

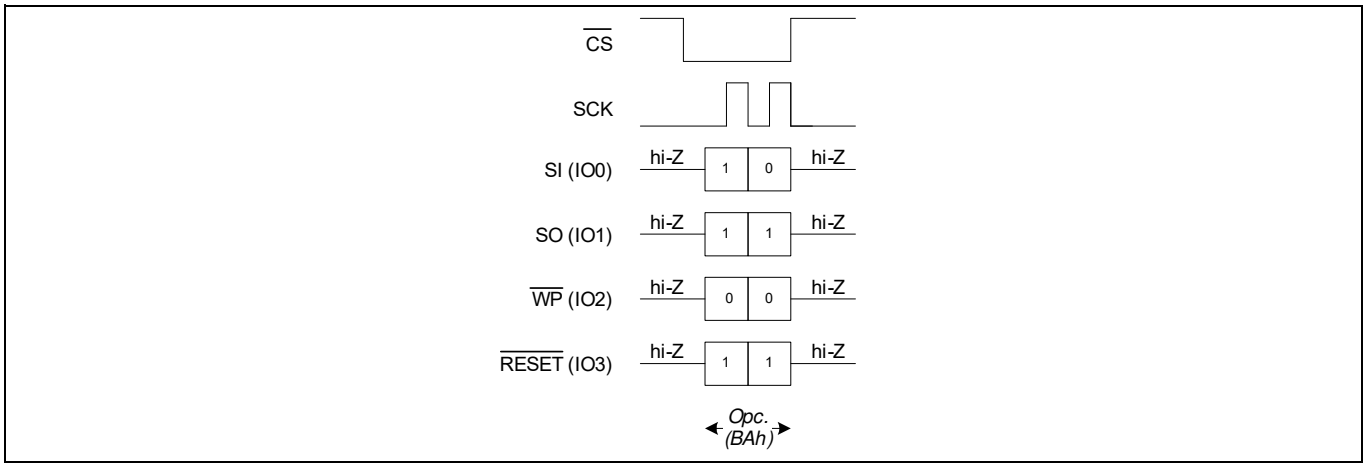



**Figure 105 Hibernate mode operation in QPI mode**

### 5.1.9.3 Software reset

The software reset operation combines two instructions: reset-enable (RSTEN) instruction followed by a reset (RST) instruction. It resets the whole device and makes it ready to receive instructions only after $t_{SRESET}$ time.

**Notes**

- Any instruction other than RST following the RSTEN instruction will clear the reset enable condition and prevent a later RST instruction from being recognized.
- During software reset, only RDSR1 and RDAR (to access RDSR1) commands are supported. Other commands will be ignored.
- The timing details shown in the SPI mode timing diagram are applicable as is in the DPI and QPI modes.

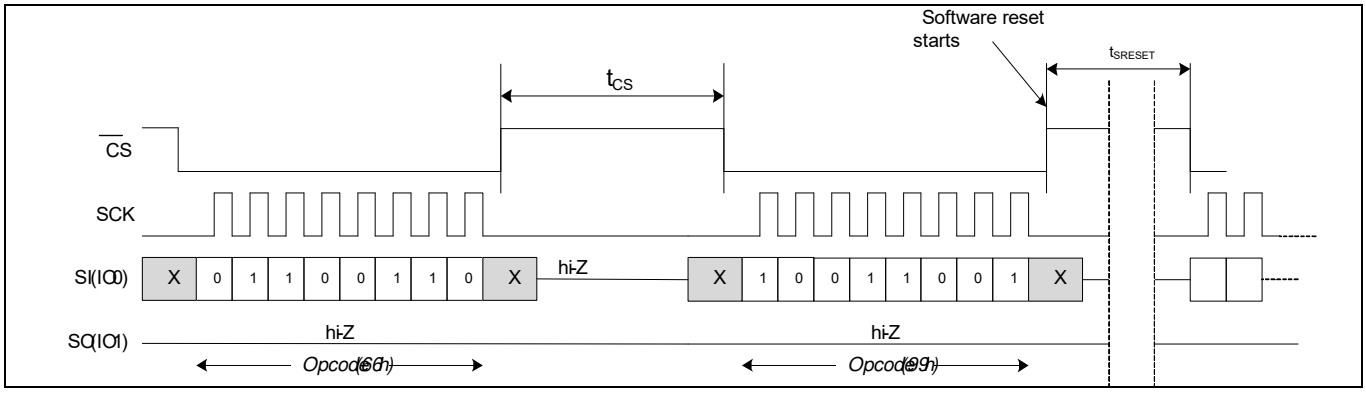



**Figure 106 Software reset timing in SPI mode**

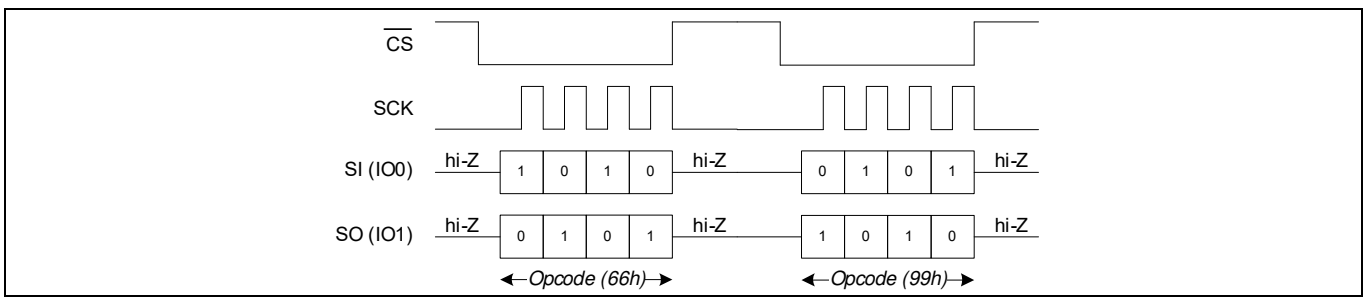



**Figure 107 Software reset timing in DPI mode**

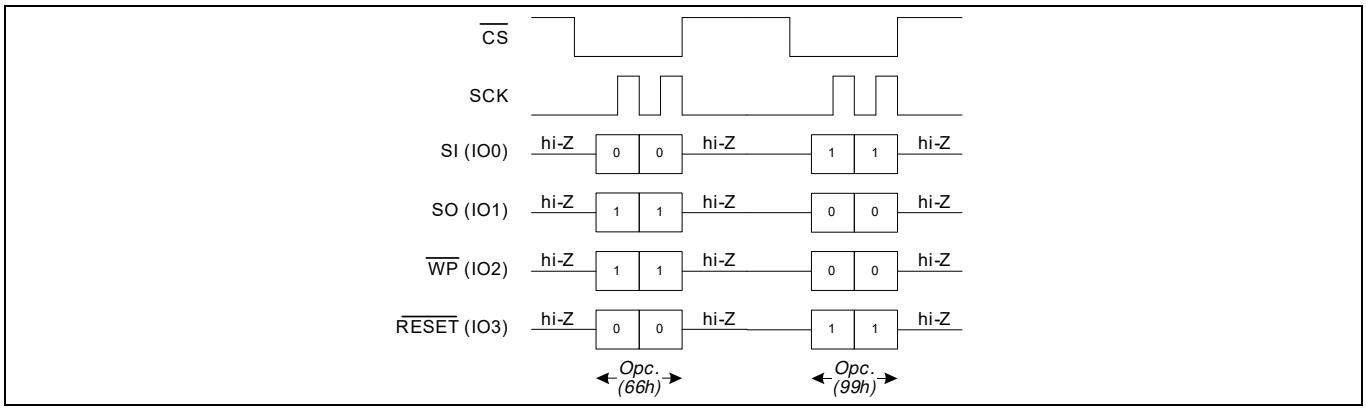



**Figure 108 Software reset timing in QPI mode**

### 5.1.9.4 Hardware reset (RESET)

The hardware reset input (RESET) is multiplexed on (RESET / (I/O3) and is an active LOW signal in CY15X102QSN device. Refer to **Table 21** for hardware reset (RESET) pin configurations across various SPI interfaces. When RESET is pulled LOW, CY15X102QSN self initializes and brings its configuration back to the power up status. Refer to **Table 59** for different registers configuration after RESET cycle. Once RESET is issued, CY15X102QSN takes $t_{RPH}$/$t_{HRESET}$ time from RESET rising edge to complete the reset cycle. CY15X102QSN becomes inaccessible during $t_{RPH}$ time. **Figure 109** to **Figure 111** show the RESET timings in different reset mode.

**Notes**

- The RESET pin is multiplexed on I/O3 in the QPI mode. When using the hardware (RESET) in QPI mode, the CR2 [5] bit must be set to '1' to enable to use I/O3 as RESET input when CS is HIGH. **Figure 109** shows the RESET / (I/O3) timing in QPI mode
- QUAD bit CR1 [1] in configuration register 1 must be set to '0' to enable the hardware reset feature on the RESET pin.
- In a shared bus configuration in QPI mode, if the RESET function is enabled, the device will reset every time (RESET / (I/O3)) toggles due to any ongoing communication between the master another QSPI slave on the same bus. Hence, it is recommended to disable RESET pin functionality in a shared bus configuration.

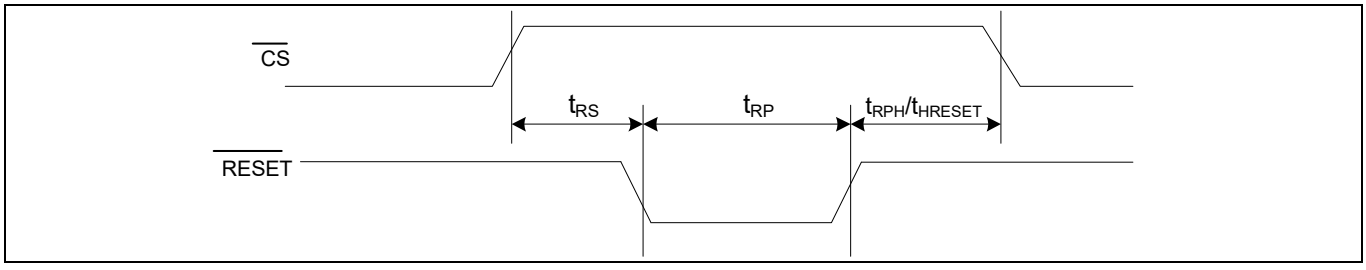



**Figure 109 RESET timing - SPI with QUAD set (CR1[1] = '1') or QPI enabled (CR2[6] = '1')**

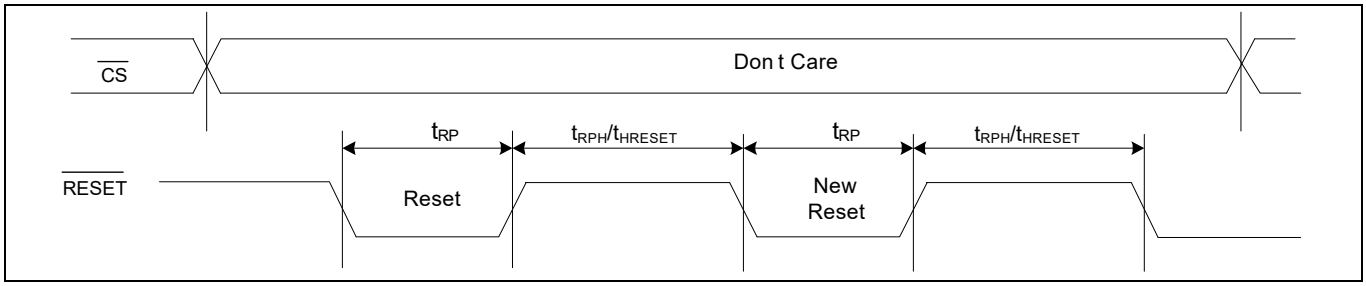



**Figure 110 RESET timing - SPI with QUAD clear (CR1[1] = '0') and QPI disabled (CR2[6] = '0')**

### 5.1.9.5 JEDEC SPI reset

JEDEC SPI reset is a signaling protocol which initiates a hardware reset independent of the device's operating I/O mode. It brings the device to its default mode as selected in the status and configuration registers. **Table 59** shows the device status after the default recovery is initiated.

The default recovery steps are as follows:

1. $\overline{CS}$ toggles active LOW to select the SPI slave.
2. SCK remains stable either in a HIGH or in a LOW state.
3. SI (I/O0) toggles HIGH to LOW, simultaneously with $\overline{CS}$ going LOW. Other I/Os (I/O1, I/O2, and I/O3) remain don't care.
4. $\overline{CS}$ is driven HIGH while I/O0 remain LOW.
5. Repeat the above steps 1 to 4 each time alternating the state of SI (I/O0) at the falling edge of $\overline{CS}$ for a total of four times.
6. Reset occurs after the 4$^{th}$ $\overline{CS}$ goes HIGH (inactive).

Refer to **Figure 111** for timing details.

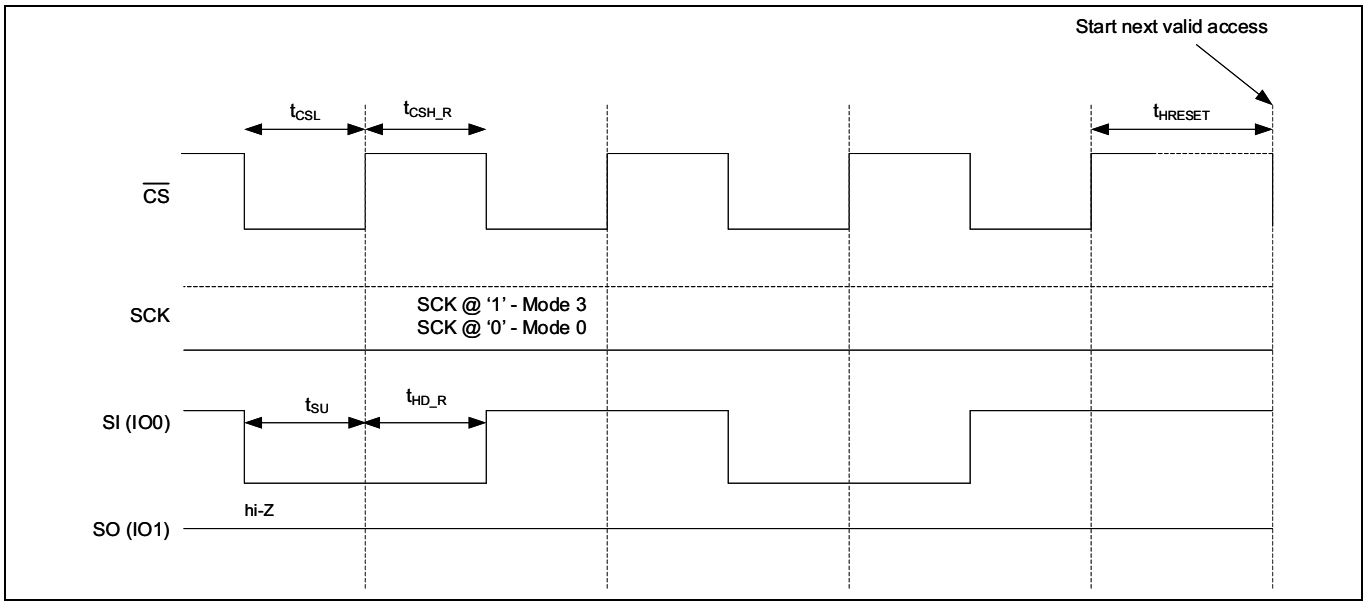



**Figure 111 JEDEC SPI reset**

**Table 59 Status of registers after various types of reset**

| Reset function | I/O requirements | Status registers (SRx) | Configuration register (CRx) | ECC status | CRC reg | ECC count reg (ECCDC) | ADDR trap reg (ADDTRAP) | I/O modes |
|---|---|---|---|---|---|---|---|---|
| **Power-on reset** | $\overline{CS}$ = '1' Other inputs - ignored All outputs - tristated | SR1 - load default values SR2 - 0x00 | CR1, CR2, CR4, CR5 Load default values | Load - 0x00 | Load - 0x00 | Load - 0x00 | Load - 0x00 | No change |
| **Hardware reset** | $\overline{CS}$ = '1' Other inputs - ignored All outputs - tristated | SR1 - load default values SR2 - 0x00 | CR1, CR2, CR4, CR5 Load default values | Load - 0x00 | Load - 0x00 | Load - 0x00 | Load - 0x00 | No change |
| **Software reset** | Instruction (RSTEN, RST) | SR1 - no change except the WEL bit which will clear to '0' if set '1'. SR2 - 0x00 | CR1, CR2, CR4, CR5 - no change | Load - 0x00 | Load - 0x00 | Load - 0x00 | Load - 0x00 | No change |
| **JEDEC reset (default recovery)** | $\overline{CS}$ and SI (IO0) = toggle Other inputs - ignored All outputs - tristated | SR1 - load default values SR2 - 0x00 | CR1, CR2, CR4, CR5 Load default values | Load - 0x00 | Load - 0x00 | Load - 0x00 | Load - 0x00 | No change |

The SPI host can issue hardware $\overline{RESET}$ or JEDEC SPI reset if CY15X102QSN goes into an undefined state and stops responding to any SPI command. The CY15X102QSN enters into an internal test mode or any undefined mode either due to wrong opcode or any glitch on the SPI signals which can internally cause latching of a wrong opcode, or part didn't boot up successfully (keep showing busy status WIP = '1' after $t_{PU}$).

**Note** ECC (ECCDC and ADDRTRAP) registers lose their content while in DPD and return to their default values 0x00 for ECC registers. Return from hibernate reloads all registers to their default values at power up as shown in **Table 3**.

# 6 Electrical characteristics

## 6.1 Maximum ratings

Exceeding the maximum ratings may impair the useful life of the device. User guidelines are not tested.

Storage temperature ..... –65°C to +125°C

Maximum accumulated storage time
At 125 °C ambient temperature ..... 1000 h
At 85 °C ambient temperature ..... 10 Years

Maximum junction temperature ..... 125°C

Supply voltage on $V_{DD}$ relative to $V_{SS}$:
CY15V102QSN: ..... –0.5 V to +2.4 V
CY15B102QSN: ..... –0.5 V to +4.1 V

Input voltage ..... $V_{IN} \leq V_{DD} + 0.5$ V

DC voltage applied to outputs in High-Z state ..... –0.5 V to $V_{DD}$ + 0.5 V

Transient voltage (< 20 ns) on any pin to ground potential ..... –2.0 V to $V_{DD}$ + 2.0 V

Package power dissipation capability ($T_A$ = 25°C) ..... 1.0 W

Surface mount lead soldering temperature (3 seconds) ..... +260°C

DC output current
(1 output at a time, 1s duration) ..... 15 mA

Electrostatic discharge voltage
Human Body Model (JEDEC Std JESD22-A114-B) ..... 2 kV
Charged Device Model (JEDEC Std JESD22-C101-A) ..... 500 V

Latch-up current ..... >140 mA

# 7 Operating range

**Table 60 Operating range**

| Device | Ambient temperature | $V_{DD}$ |
|---|---|---|
| CY15V102QSN | Industrial, –40°C to +85°C | 1.71 V to 1.89 V |
| CY15B102QSN | | 1.8 V to 3.6 V |

# 8 DC electrical characteristics

**Table 61 DC electrical characteristics**

Over the **Operating range**

| Parameter | Description | Test Conditions | | Min | Typ[18] | Max | Unit |
|---|---|---|---|---|---|---|---|
| $V_{DD}$ | Power supply | CY15V102QSN | | 1.71 | 1.8 | 1.89 | V |
| | | CY15B102QSN | | 1.8 | 3.0 | 3.6 | V |
| $I_{DD1}$ | $V_{DD}$ supply current in SPI SDR mode | $V_{DD}$ = 1.71 V to 1.89 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 50 MHz | – | 4.9 | 6.0 | mA |
| | | | $f_{SCK}$ = 108 MHz | – | 10 | 12 | mA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 50 MHz | – | 5.6 | 7.2 | mA |
| | | | $f_{SCK}$ = 108 MHz | – | 11 | 14 | mA |
| $I_{DD2}$ | VDD supply current in DPI SDR mode | $V_{DD}$ = 1.71 V to 1.89 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 108 MHz | – | 12 | 14 | mA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 108 MHz | – | 13 | 16 | mA |
| $I_{DD3}$ | $V_{DD}$ supply current in QPI SDR mode | $V_{DD}$ = 1.71 V to 1.89 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 108 MHz | – | 16 | 19 | mA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 108 MHz | – | 17 | 21 | mA |

**Note**

18. Typical values are at 25°C, $V_{DD}$ = $V_{DD}$ (Typ). Not 100% tested.

**Table 61 DC electrical characteristics** (continued)

Over the **Operating range**

| Parameter | Description | Test Conditions | | Min | Typ[18] | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{DD3}$ | $V_{DD}$ supply current in QPI DDR mode | $V_{DD}$ = 1.71 V to 1.89 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 54 MHz | – | 16 | 19 | mA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; SCK toggling between $V_{DD}$ – 0.2 V and $V_{SS}$, other inputs $V_{SS}$ or $V_{DD}$ – 0.2 V. No output loads. | $f_{SCK}$ = 54 MHz | – | 17 | 21 | mA |
| $I_{SB}$ | $V_{DD}$ standby current | $V_{DD}$ = 1.71 V to 1.89 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 110 | – | µA |
| | | | $T_A$ = 85°C | – | – | 209 | µA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 200 | – | µA |
| | | | $T_A$ = 85°C | – | – | 350 | µA |
| $I_{DPD}$ | Deep power-down current | $V_{DD}$ = 1.71 V to 1.89 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 0.8 | – | µA |
| | | | $T_A$ = 85°C | – | – | 15 | µA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 1.0 | – | µA |
| | | | $T_A$ = 85°C | – | – | 17 | µA |
| $I_{HBN}$ | Hibernate mode current | $V_{DD}$ = 1.71 V to 1.89 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 0.1 | – | µA |
| | | | $T_A$ = 85°C | – | – | 0.9 | µA |
| | | $V_{DD}$ = 1.8 V to 3.6 V; $\overline{CS}$ = $V_{DD}$. All other inputs $V_{SS}$ or $V_{DD}$. | $T_A$ = 25°C | – | 0.1 | – | µA |
| | | | $T_A$ = 85°C | – | – | 1.6 | µA |
| $I_{LI}$ | Input leakage current on I/O pins | $V_{SS}$ < $V_{IN}$ < $V_{DD}$ | | –1 | – | 1 | µA |
| | Input leakage current on $\overline{WP}$ and $\overline{RESET}$ (when I/O2 and I/O3 functions disabled) | | | –100 | – | 1 | µA |

**Note**
18. Typical values are at 25°C, $V_{DD}$ = $V_{DD}$ (Typ). Not 100% tested.

**Table 61 DC electrical characteristics** (continued)

Over the **Operating range**

| Parameter | Description | Test Conditions | Min | Typ[18] | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{LO}$ | Output leakage current | $V_{SS} < V_{OUT} < V_{DD}$ | –1 | – | 1 | µA |
| $V_{IH}$ | Input HIGH voltage | – | $0.7 \times V_{DD}$ | – | $V_{DD} + 0.3$ | V |
| $V_{IL}$ | Input LOW voltage | – | –0.3 | – | $0.3 \times V_{DD}$ | V |
| $V_{OH1}$ | Output HIGH voltage | $I_{OH} = -1$ mA, $V_{DD} = 2.7$ V. | 2.4 | – | – | V |
| $V_{OH2}$ | Output HIGH voltage | $I_{OH} = -100$ µA | $V_{DD} - 0.2$ | – | – | V |
| $V_{OL1}$ | Output LOW voltage | $I_{OL} = 2$ mA, $V_{DD} = 2.7$ V | – | – | 0.4 | V |
| $V_{OL2}$ | Output LOW voltage | $I_{OL} = 150$ µA | – | – | 0.2 | V |

**Note**

18. Typical values are at 25°C, $V_{DD} = V_{DD}$ (Typ). Not 100% tested.

# 9 Data retention and endurance

**Table 62 Data retention and endurance**

| Parameter | Description | Test condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $T_{DR}$ | Data retention | $T_A$ = 85°C | 10 | – | Years |
| | | $T_A$ = 75°C | 38 | – | Years |
| | | $T_A$ = 65°C | 151 | – | Years |
| $NV_C$ | Endurance | Over operating temperature | $10^{14}$ | – | Cycles |

# 10 Capacitance

**Table 63 Capacitance**

| Parameter[19] | Description | Test conditions | Max | Unit |
|---|---|---|---|---|
| $C_O$ | Output pin capacitance (SO) | $T_A = 25°C, f = 1\ MHz, V_{DD} = V_{DD}(Typ)$ | 6 | pF |
| $C_I$ | Input pin capacitance | | 5 | pF |

**Note**
19.This parameter is periodically sampled and not 100% tested.

# 11 Thermal resistance

**Table 64 Thermal resistance**

| Parameter[20] | Description | Test conditions | 8-pin SOIC package | Unit |
|---|---|---|---|---|
| $\Theta_{JA}$ | Thermal resistance (junction to ambient) | Test conditions follow standard test methods and procedures for measuring thermal impedance, per EIA/JESD51. | 44.9 | °C/W |
| $\Theta_{JC}$ | Thermal resistance (junction to case) | | 20.2 | °C/W |

**Note**
20.This parameter is periodically sampled and not 100% tested.

# 12 AC test conditions

**Table 65 AC test conditions**

| Parameter | Value | |
|---|---|---|
| | CY15V102QSN | CY15B102QSN |
| Input pulse levels (0 V to $V_{DD}$) | 0 V to $V_{DD}$ | 0 V to $V_{DD}$ |
| Input rise and fall times (10% to 90%) | ≤ 1.8 ns | ≤ 2.0 ns |
| Input timing reference voltages | 0.3 × $V_{DD}$ to 0.7 × $V_{DD}$ | 0.3 × $V_{DD}$ to 0.7 × $V_{DD}$ |
| Output timing reference voltages ($V_T$) | $V_{DD}$/2 | $V_{DD}$/2 |
| Load capacitance ($C_L$) | 30 pF | 30 pF |

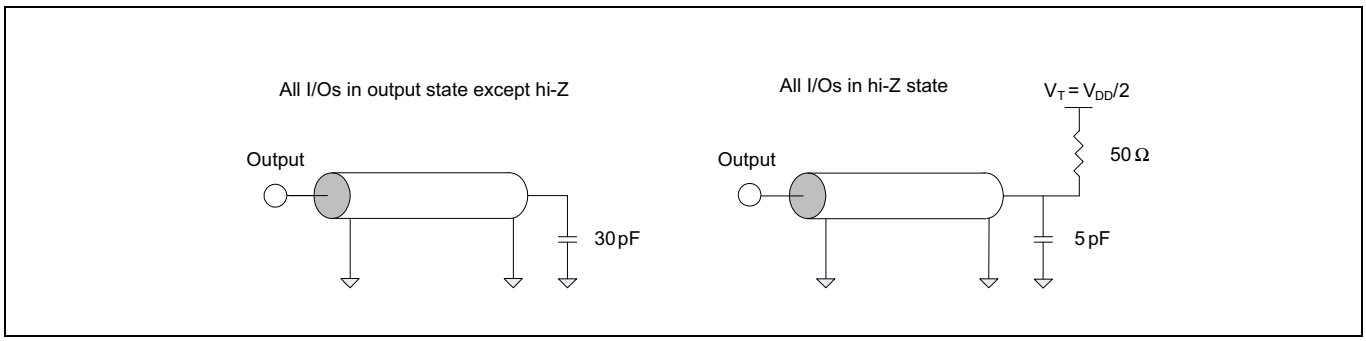



**Figure 112 AC test loads**

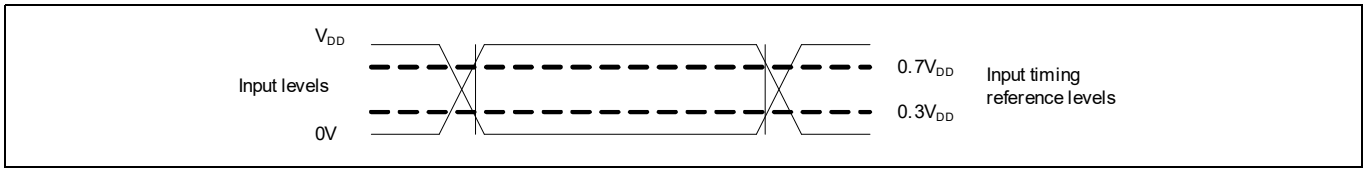



**Figure 113 AC timing input voltage reference levels**

# 13 SDR AC switching characteristics

**Table 66 SDR AC switching characteristics**

| Parameters[21] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| Parameter | Alt. parameter | | | | |
| $f_{SCK}$ | – | SCK clock frequency | 0 | 108 | MHz |
| $t_{CH}$ | – | Clock HIGH time | 0.45 × 1/$f_{SCK}$ | – | ns |
| $t_{CL}$ | – | Clock LOW time | 0.45 × 1/$f_{SCK}$ | – | ns |
| $t_{CSS}$ | $t_{CSU}$ | Chip select ($\overline{CS}$) setup time | 5 | – | ns |
| $t_{CSH}$ | $t_{CSH}$ | Chip select ($\overline{CS}$) hold time - SPI mode 0 | 4 | – | ns |
| $t_{CSH1}$ | – | Chip select ($\overline{CS}$) hold time - SPI mode 3 | 9 | – | ns |
| $t_{HZCS}$ | $t_{OD}$[22, 23] | Output disable time – CY15B102QSN | – | 10 | ns |
| | | Output disable time – CY15V102QSN | – | 11 | ns |
| $t_{CO}$ | | Output data valid time with 15-pF load (Output driver set to 45 Ω. Over the **"Operating range"** on page 92) | – | 7 | ns |
| | | Clock low to output valid – 15-pF load (Output driver set to 45 Ω. For $V_{DD}$ = 2.7 V to 3.6 V; over the **"Operating range"** on page 92) | – | 6.7 | ns |
| | | Clock low to output valid – 30-pF load (Output driver set to 45 Ω. For $V_{DD}$ = 2.7 V to 3.6 V; over the **"Operating range"** on page 92) | – | 7 | ns |
| | | Clock low to output valid – 30-pF load (Output driver set to default 30 Ω. Over the **"Operating range"** on page 92) | – | 7 | ns |
| $t_{OH}$ | – | Output hold time | 1 | – | ns |

**Notes**

21. These parameters are tested per **"AC test conditions"** on page 99.
22. $t_{OD}$ and $t_{HZ}$ are specified with a load capacitance of 5 pF. Transition is measured when the outputs enter a high impedance state.
23. Characterized but not 100% tested in production.
24. $t_{CS}$ is the minimum chip deselect ($\overline{CS}$ HIGH) time before the new command cycle starts in a specific SPI mode (SPI, DPI or QPI). This parameter ensures that previous operation is successfully completed before the host starts a new command cycle. Refer to **Figure 116**.

**Table 66 SDR AC switching characteristics** (continued)

| Parameters[21] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| **Parameter** | **Alt. parameter** | | | | |
| $t_{CS}$[24] | $t_D$ | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in SPI mode; all accesses (memory array and registers) | 40 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in DPI mode; all accesses except memory array access | 75 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in DPI mode (including dual mode in extended SPI); memory array access (non XIP mode) | 40 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in DPI mode (including dual mode in extended SPI); memory array access (XIP mode) | 55 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode; all accesses except memory array access | 110 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode (including quad mode in extended SPI); memory array access (non XIP mode) | 90 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode (including quad mode in extended SPI); memory array access (XIP mode) | 110 | – | ns |
| $t_{SD}$ | $t_{SU}$ | Data in setup time (with respect to SCK) | 2 | – | ns |
| $t_{HD}$ | $t_H$ | Data in hold time (with respect to SCK) | 3 | – | ns |
| $t_{CLZ}$ | – | Clock Low to Output low-Z | 0 | – | ns |
| $t_{CRCC}$ | – | CRC calculation time (100 µs + (0.8 µs/Byte of data)) | 0.10 | 440 | ms |
| $t_{CRCS}$ | – | $\overline{CS}$ high to CRC calculation suspends | – | 100 | µs |
| $t_{CRCR}$ | – | $\overline{CS}$ high to CRC calculation resumes | – | 100 | µs |

**Notes**
21. These parameters are tested per **"AC test conditions"** on page 99.
22. $t_{OD}$ and $t_{HZ}$ are specified with a load capacitance of 5 pF. Transition is measured when the outputs enter a high impedance state.
23. Characterized but not 100% tested in production.
24. $t_{CS}$ is the minimum chip deselect ($\overline{CS}$ HIGH) time before the new command cycle starts in a specific SPI mode (SPI, DPI or QPI). This parameter ensures that previous operation is successfully completed before the host starts a new command cycle. Refer to **Figure 116**.

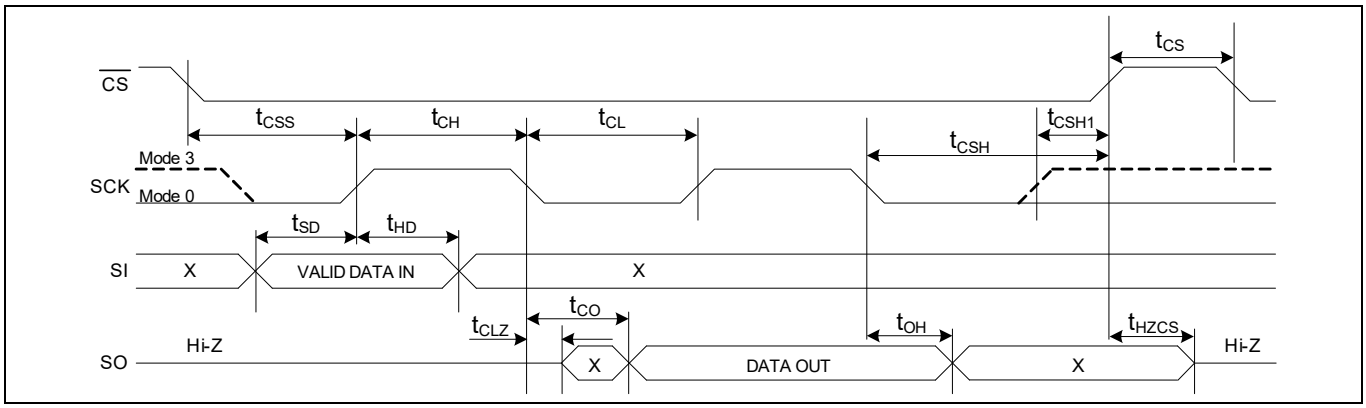



**Figure 114 SPI switching timing - single IO, SDR (mode 0 and mode 3)**

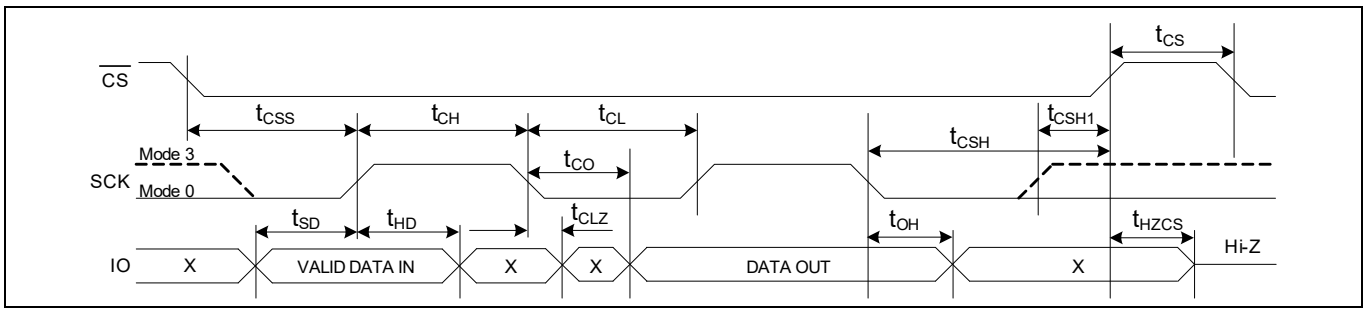



**Figure 115 SPI switching timing - multiple I/O, SDR (mode 0 and mode 3)**

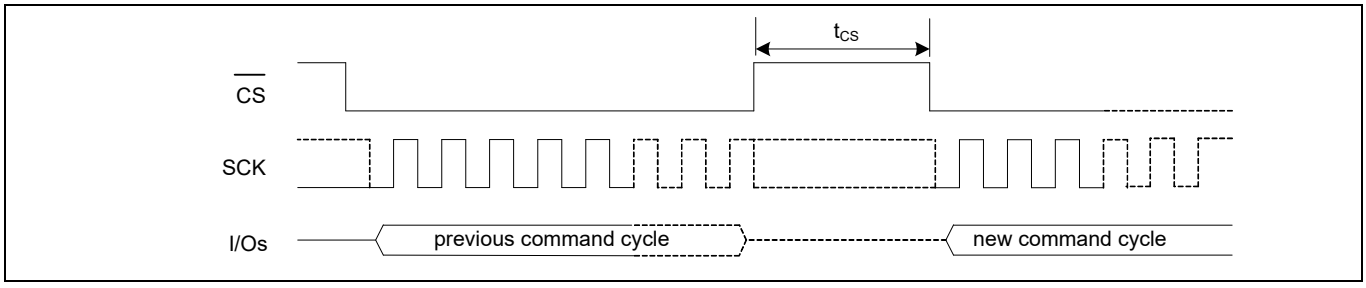



**Figure 116 Chip deselect ($\overline{CS}$ HIGH) - $t_{CS}$ timing**

# 14 DDR AC switching characteristics

**Table 67 DDR AC switching characteristics**

Over the **Operating range**

| Parameters[25] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| **Parameter** | **Alt. parameter** | | | | |
| $f_{SCK}$ | – | SCK clock frequency | 0 | 54 | MHz |
| $t_{CH}$ | – | Clock HIGH time | 0.45 × 1/$f_{SCK}$ | – | ns |
| $t_{CL}$ | – | Clock LOW time | 0.45 × 1/$f_{SCK}$ | – | ns |
| $t_{CSS}$ | $t_{CSU}$ | Chip select ($\overline{CS}$) setup time | 5 | – | ns |
| $t_{CSH}$ | $t_{CSH}$ | Chip select ($\overline{CS}$) hold time | 5 | – | ns |
| $t_{HZCS}$ | $t_{OD}$[26, 27] | Output disable time – CY15B102QSN | – | 10 | ns |
| | | Output disable time – CY15V102QSN | – | 11 | ns |
| $t_{CO}$ | – | Output data valid time with 15-pF load (Output driver set to 45 Ω. Over the **"Operating range"** on page 92) | – | 7 | ns |
| | | Clock low to output valid – 15-pF load (Output driver set to 45 Ω. For $V_{DD}$ = 2.7 V to 3.6 V; over the **"Operating range"** on page 92) | – | 6.7 | ns |
| | | Clock low to output valid – 30-pF load (Output driver set to 45 Ω. For $V_{DD}$ = 2.7 V to 3.6 V; over the **"Operating range"** on page 92) | – | 7 | ns |
| | | Clock low to output valid – 30-pF load (Output driver set to default 30 Ω. Over the **"Operating range"** on page 92) | – | 7 | ns |
| $t_{OH}$ | – | Output hold time | 1 | – | ns |
| $t_{CS}$[28] | $t_D$ | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in SPI mode; all accesses (memory array and registers) | 40 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode; all accesses except memory array access | 110 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode (including quad mode in extended SPI); memory array access (non XIP mode) | 90 | – | ns |
| | | Chip deselect ($\overline{CS}$ HIGH) time before the command cycle in QPI mode (including quad mode in extended SPI); memory array access (XIP mode) | 110 | – | ns |
| $t_{SD}$ | $t_{SU}$ | Data in setup time (with respect to SCK) | 4 | – | ns |

**Notes**

25. These parameters are tested per **"AC test conditions"** on page 99.
26. $t_{OD}$ and $t_{HZ}$ are specified with a load capacitance of 5 pF. Transition is measured when the outputs enter a high impedance state.
27. Characterized but not 100% tested in production.
28. $t_{CS}$ is the minimum chip deselect ($\overline{CS}$ HIGH) time before the new command cycle starts in a specific SPI mode (SPI or QPI). This parameter ensures that previous operation is successfully completed before the host starts a new command cycle. Refer to **Figure 116**.

**Table 67 DDR AC switching characteristics** (continued)

Over the **Operating range**

| Parameters[25] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| **Parameter** | **Alt. parameter** | | | | |
| $t_{HD}$ | $t_H$ | Data in hold time (with respect to SCK) | 4 | – | ns |
| $t_{CLZ}$ | – | Clock low to output Low-Z | 0 | – | ns |

**Notes**

25. These parameters are tested per **"AC test conditions"** on page 99.
26. $t_{OD}$ and $t_{HZ}$ are specified with a load capacitance of 5 pF. Transition is measured when the outputs enter a high impedance state.
27. Characterized but not 100% tested in production.
28. $t_{CS}$ is the minimum chip deselect ($\overline{CS}$ HIGH) time before the new command cycle starts in a specific SPI mode (SPI or QPI). This parameter ensures that previous operation is successfully completed before the host starts a new command cycle. Refer to **Figure 116**.

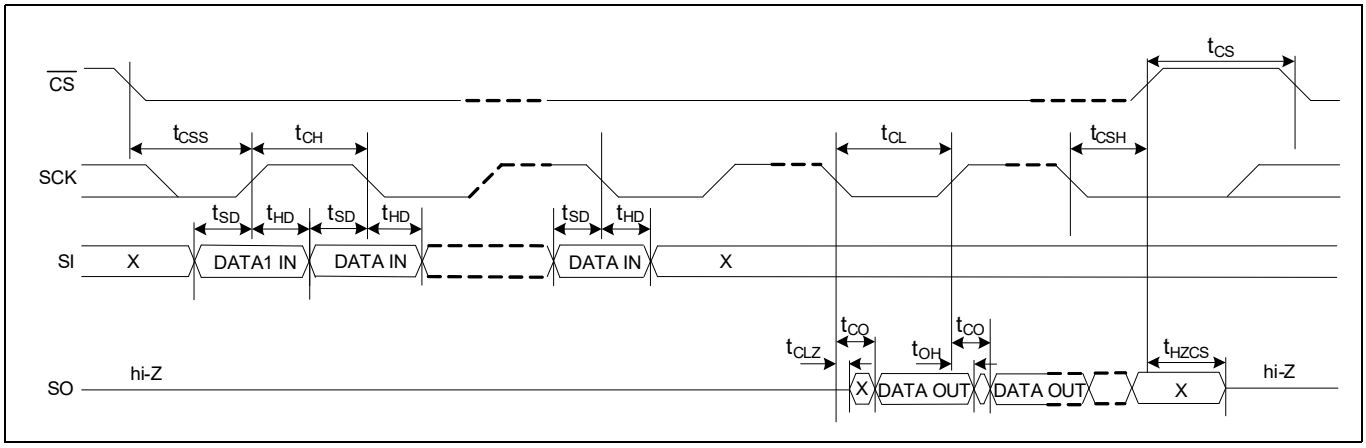



**Figure 117 SPI switching timing - single I/O, DDR**

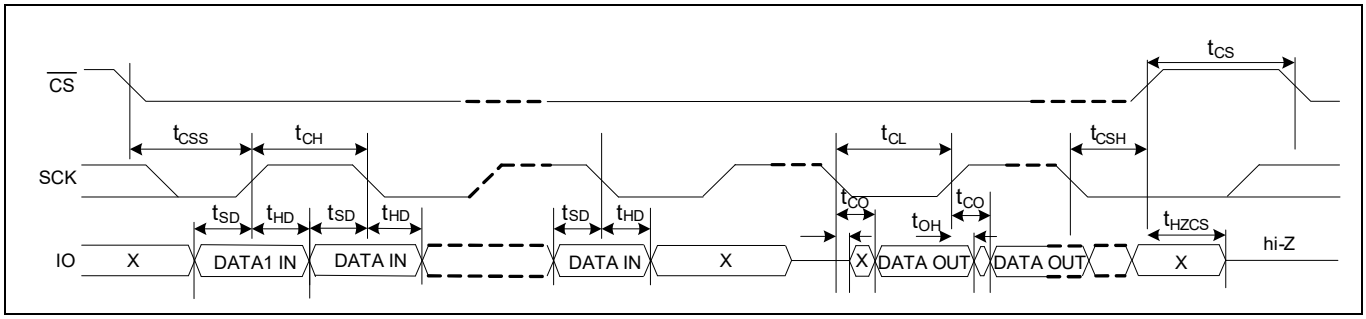



**Figure 118 SPI switching timing - multiple I/O, DDR**[29]

**Note**

29. The DDR mode input timing, capturing data input on both the clock edge, is applicable to address and data input cycles only. The DDR opcodes are always transmitted in SDR mode during the opcode cycle.

# 15 Write protect ($\overline{WP}$) timing parameters

**Table 68 Write protect ($\overline{WP}$) timing parameters**

Over the **Operating range**

| Parameters[30] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| Parameter | Alt. parameter | | | | |
| $t_{WPS}$ | $t_{SW}$ | $\overline{WP}$ setup time (with respect to $\overline{CS}$) | 20 | – | ns |
| $t_{WPH}$ | $t_{HW}$ | $\overline{WP}$ hold time (with respect to $\overline{CS}$) | 20 | – | ns |

**Note**
30.These parameters are tested per **"AC test conditions"** on page 99.

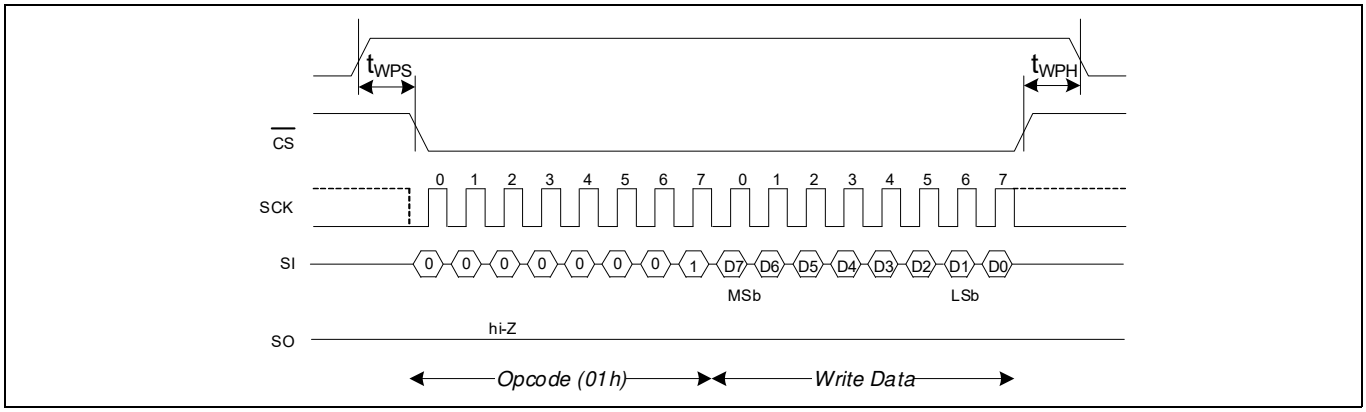



**Figure 119 Write protect setup and hold timing**

# 16 Reset ($\overline{\text{RESET}}$) timing parameters

**Table 69 Reset ($\overline{\text{RESET}}$) timing parameters**

Over the **Operating range**

| Parameters[31] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| **Parameter** | **Alt. parameter** | | | | |
| $t_{RS}$ | – | Hardware $\overline{\text{RESET}}$ setup time | 50 | – | ns |
| $t_{RPH}$ | $t_{RHSL}$, $t_{RH}$ | Hardware $\overline{\text{RESET}}$ hold time | 450 | – | µs |
| $t_{RP}$ | $t_{RLRH}$ | Hardware $\overline{\text{RESET}}$ pulse width | 200 | – | ns |
| $t_{HRESET}$ | – | Hardware $\overline{\text{RESET}}$ time | – | 450 | µs |
| $t_{SRESET}$ | – | Software $\overline{\text{RESET}}$ time | – | 100 | µs |
| $t_{CSL}$ | – | Chip Select ($\overline{\text{CS}}$) LOW time for JEDEC Reset | 500 | – | ns |
| $t_{CSH\_R}$ | – | Chip Select ($\overline{\text{CS}}$) HIGH time for JEDEC Reset | 500 | – | ns |
| $t_{SU}$ | – | SI (I/O0) setup time (with respect to $\overline{\text{CS}}$ HIGH) for JEDEC reset | 5 | – | ns |
| $t_{HD\_R}$ | – | SI (I/O0) hold time (with respect to $\overline{\text{CS}}$ HIGH) for JEDEC reset | 5 | – | ns |

**Note**
31.These parameters are tested per **"AC test conditions"** on page 99.

# 17 Power cycle timing

**Table 70 Power cycle timing**

Over the **Operating range**

| Parameter[32] | | Description | Min | Max | Unit |
|---|---|---|---|---|---|
| **Parameter** | **Alt. parameter** | | | | |
| $t_{PU}$ | – | Power-up $V_{DD}$(min) to first access ($\overline{CS}$ LOW) | 450 | – | µs |
| $t_{VR}$[33] | – | $V_{DD}$ power-up ramp rate | 30 | – | µs/V |
| $t_{VF}$[33] | – | $V_{DD}$ power-down ramp rate | 20 | – | µs/V |
| $t_{ENTDPD}$[34] | $t_{DP}$ | $\overline{CS}$ HIGH to enter deep power-down ($\overline{CS}$ HIGH to hibernate mode current) | – | 3 | µs |
| $t_{CSDPD}$[34] | – | $\overline{CS}$ pulse width to wake up from deep power-down mode | 0.015 | $4 \times 1/f_{SCK}$ | µs |
| $t_{EXTDPD}$[35] | $t_{RDP}$ | Recovery time from deep power-down mode ($\overline{CS}$ LOW to ready for access) | – | 10 | µs |
| $t_{ENTHIB}$ | $t_{HBN}$ | Time to enter hibernate ($\overline{CS}$ HIGH to hibernate mode current) | – | 3 | µs |
| $t_{EXTHIB}$[36] | $t_{REC}$ | Recovery time from hibernate mode ($\overline{CS}$ LOW to ready for access) | – | 450 | µs |
| $V_{DD}$(low) | – | Low $V_{DD}$ where initialization must occur | 0.6 | – | V |
| $t_{PD}$ | – | $V_{DD}$(low) time when $V_{DD}$(low) at 0.6 V | 130 | – | µs |
| | | $V_{DD}$(low) time when $V_{DD}$(low) at $V_{SS}$ | 70 | – | |

**Notes**
32. These parameters are tested per **"AC test conditions"** on page 99.
33. Slope measured at any point on the $V_{DD}$ waveform.
34. Guaranteed by design. Refer to **Figure 99** and **Figure 102** for deep sleep mode timing.
35. Guaranteed by design. Refer to **Figure 103** for hibernate mode timing.
36. Characterized but not 100% tested in production.

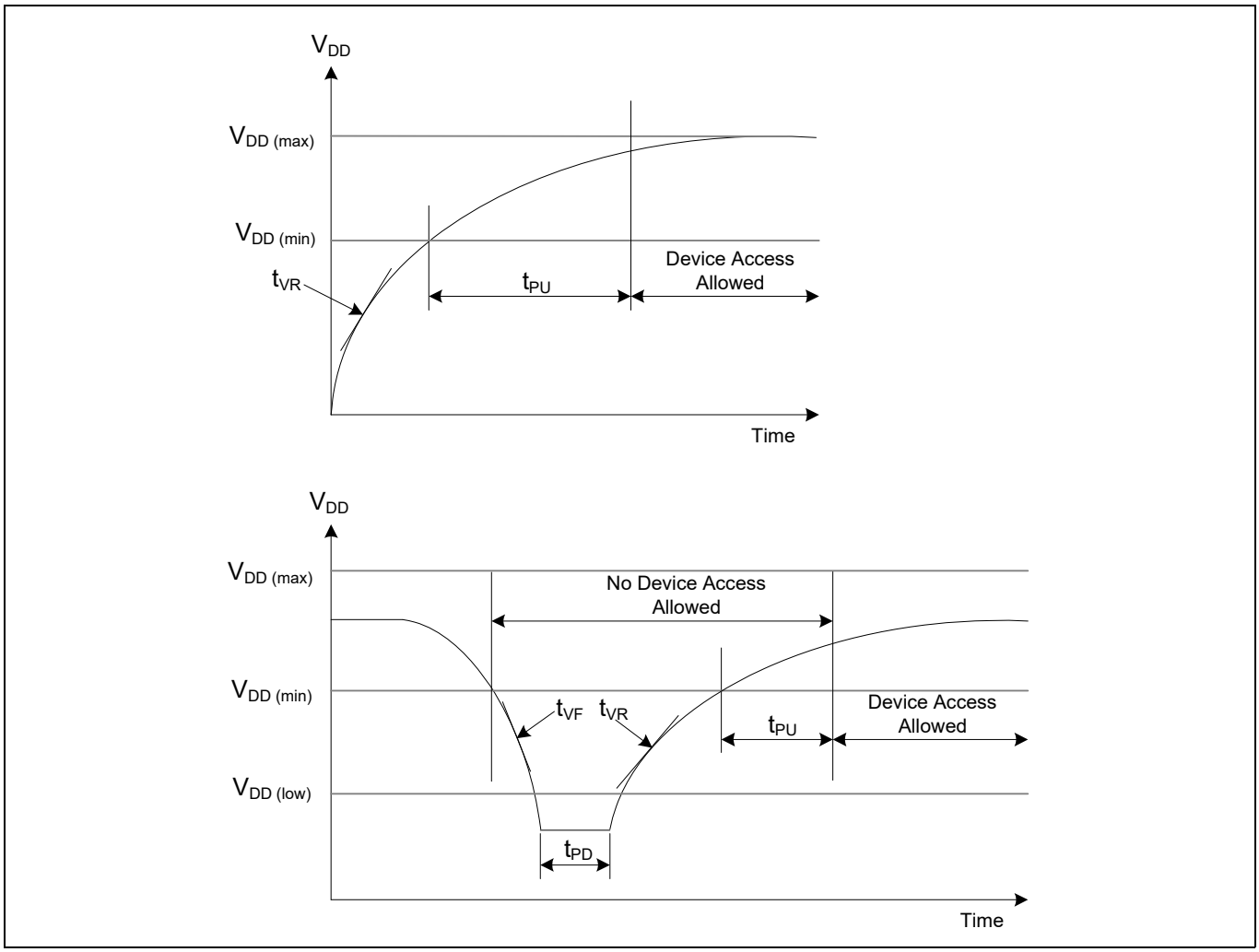



**Figure 120 Power cycle timing**

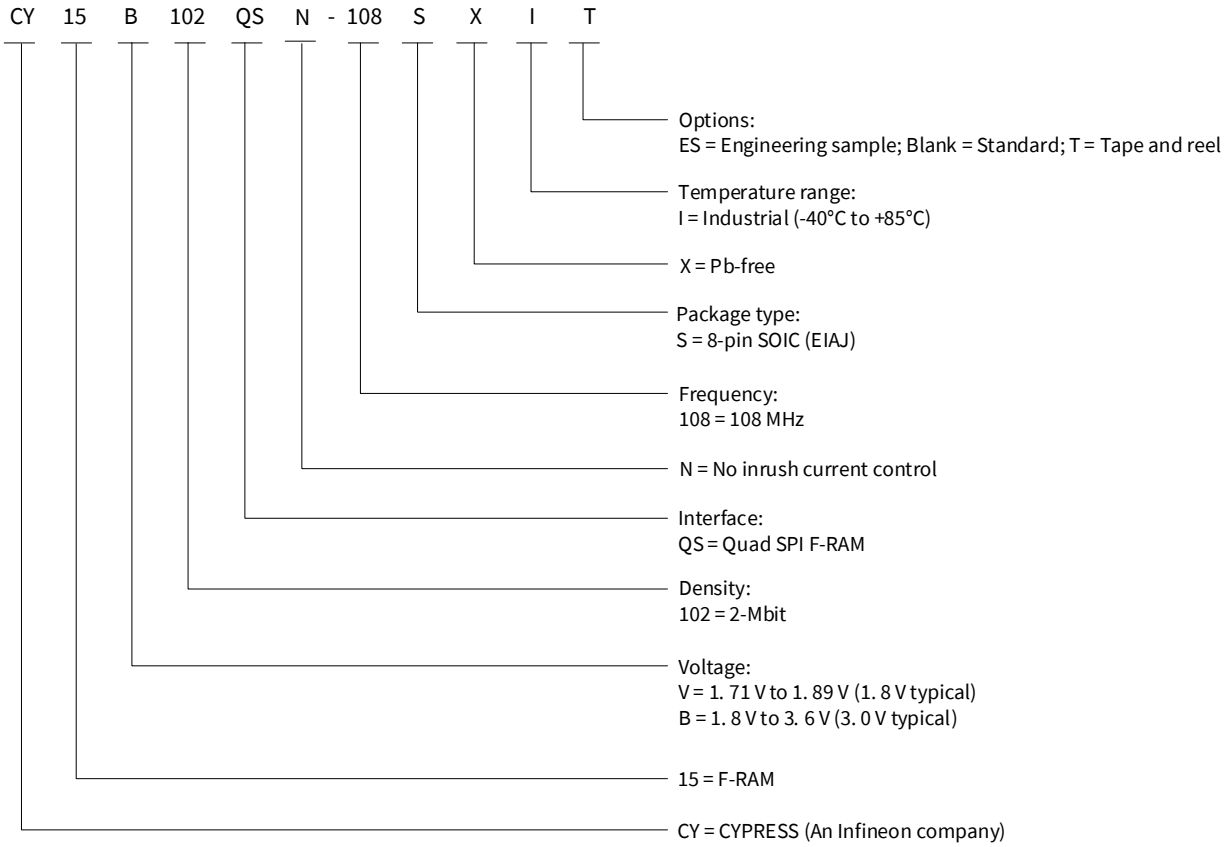
# 18 Ordering information

**Table 71 Ordering information**

| Ordering code | Device ID | Package diagram | Package type | Operating range |
|---|---|---|---|---|
| CY15B102QSN-108SXI | 0000000006825148 | 001-85261 | 8-pin SOIC (EIAJ) | Industrial |
| CY15V102QSN-108SXI | 0000000006805148 | 001-85261 | 8-pin SOIC (EIAJ) | Industrial |

All these parts are Pb-free. Contact your local sales representative for availability of these parts.

## 18.1 Ordering code definitions



# 19 Package diagram

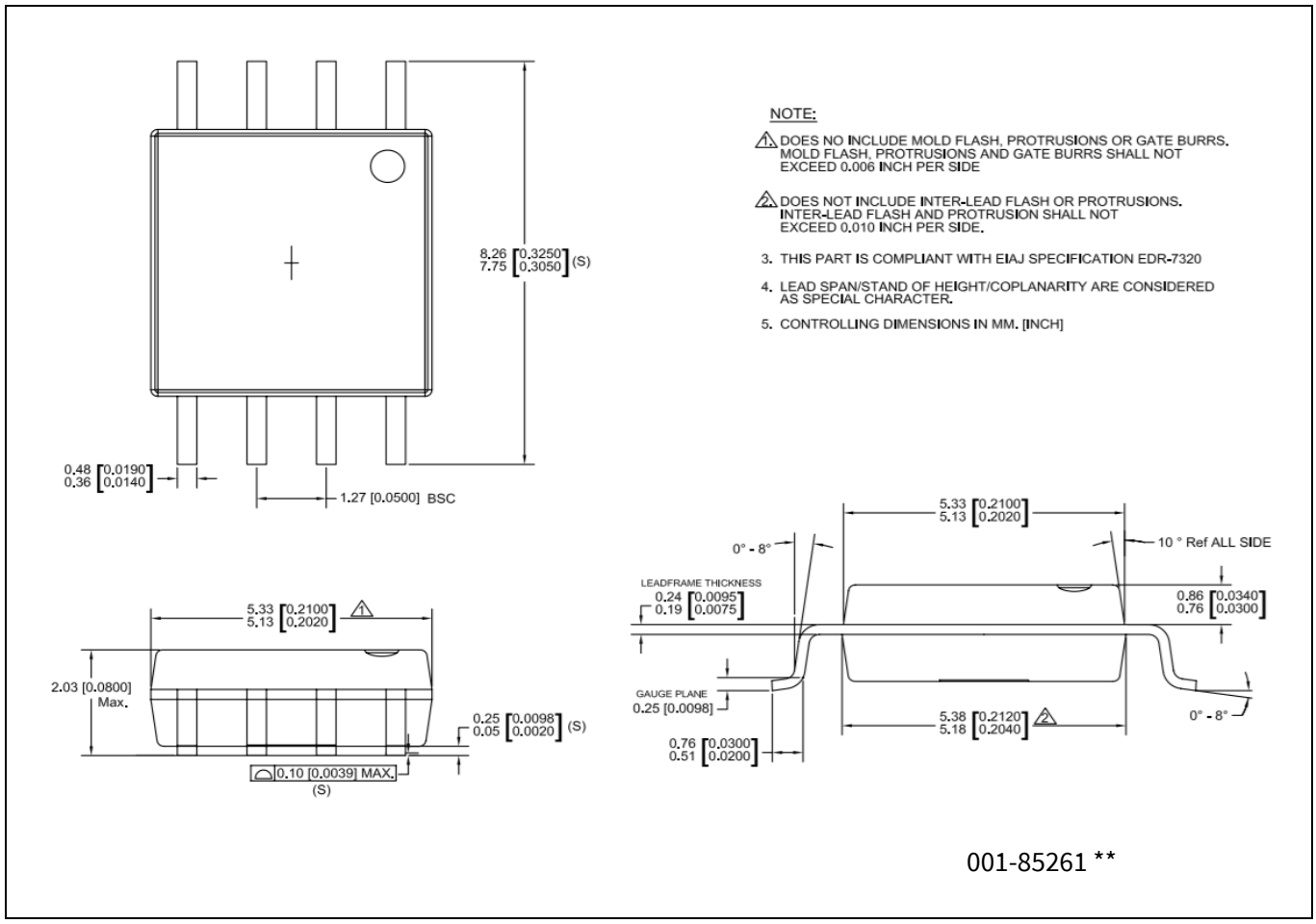



**Figure 121 8-pin SOIC (208 Mils) package outline, 001-85261**

# 20 Acronyms

**Table 72 Acronyms used in this document**

| Acronym | Description |
|---|---|
| CPHA | clock phase |
| CPOL | clock polarity |
| CRC | cyclic redundancy check |
| DDR | double data rate |
| DPI | dual SPI |
| ECC | Error Correction Code |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EIA | electronic industries alliance |
| F-RAM | ferroelectric random access memory |
| I/O | input/output |
| JEDEC | Joint Electron Devices Engineering Council |
| JESD | JEDEC standards |
| LSb | least significant bit |
| MSb | most significant bit |
| RoHS | Restriction of Hazardous Substances |
| SDR | single data rate |
| SPI | serial peripheral interface |
| SOIC | small outline integrated circuit |

# 21 Document conventions

## 21.1 Units of measure

**Table 73 Units of measure**

| Symbol | Unit of measure |
|---|---|
| °C | degree Celsius |
| Hz | hertz |
| kHz | kilohertz |
| kΩ | kilohm |
| Mbit | megabit |
| MHz | megahertz |
| µA | microampere |
| µF | microfarad |
| µs | microsecond |
| mA | milliampere |
| ms | millisecond |
| ns | nanosecond |
| Ω | ohm |
| % | percent |
| pF | picofarad |
| V | volt |
| W | watt |

# Revision history

| Document version | Date of release | Description of changes |
|---|---|---|
| *B | 2020-03-19 | Post to external web. |
| *C | 2022-01-24 | Migrated to Infineon template. |
| *D | 2022-07-26 | Updated **Features**:<br>Updated description.<br>Updated **Pin definitions**:<br>Updated **Table 1**.<br>Updated **Hardware reset (RESET)**:<br>Updated description.<br>Updated to new template. |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2022-07-26**
**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**

**© 2022 Infineon Technologies AG. All Rights Reserved.**

**Do you have a question about this document?**
**Go to www.infineon.com/support**

**Document reference**
**002-26767 Rev. *D**

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

**WARNINGS**
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.