

## USB-to-SPI Protocol Converter with GPIO (Master Mode)

### Features:

#### Universal Serial Bus (USB)

- Supports Full-Speed USB (12 Mb/s)
- Human Interface Device (HID) device
- 128-Byte Buffer to Handle Data Throughput:
  - 64-byte transmit
  - 64-byte receive
- Fully Configurable VID, PID Assignments and String Descriptor (factory programming also available)
- Bus Powered (factory default) or Self-Powered (can be selected through special USB commands)
- USB 2.0 Compliant

#### USB Driver and Software Support

- Uses Standard HID Drivers (built-in support on Windows® XP, Vista, 7, Linux and Mac OS®)
- Configuration Utility for Device's Power-up Configuration
- Utility for USB-SPI Communication, GPIO Manipulation and Miscellaneous Features Usage

#### SPI Master Peripheral

- Supports all Four SPI modes (Mode 0, 1, 2, 3)
- Bit Rates from 1500 bps up to 12 Mbps
- Configurable Delays for SPI Transactions:
  - Chip Select (assert) to 1<sup>st</sup> byte of data delay
  - Data to data delay
  - Data to Chip Select (de-assert) delay
- SPI Transactions Lengths of up to 65535 Bytes Long
- Up to 9 Chip Select lines – to be used in any combination for a given SPI transaction (the Chip Select lines are shared between GPIOs and alternate function pins; certain GPs – up to 9 of them – can be assigned with the Chip Select functionality)

#### General Purpose Input/Output (GPIO) Pins

- Nine General Purpose I/O Pins

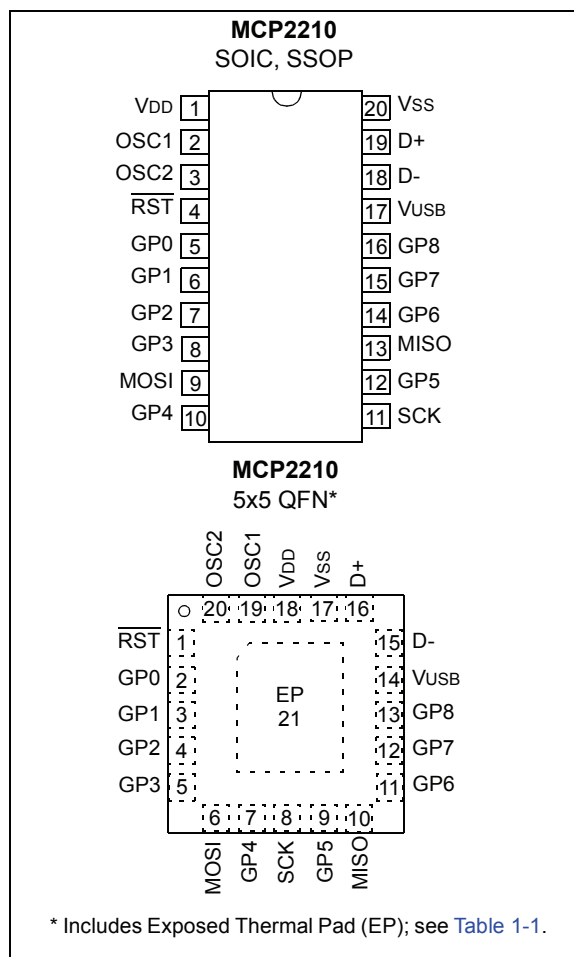
#### EEPROM

- 256 Bytes of User EEPROM (accessible through certain USB commands)

### Package Types:

The device will be offered in the following packages:

- 20-lead QFN (5 x 5 mm)
- 20-lead SOIC
- 20-lead SSOP

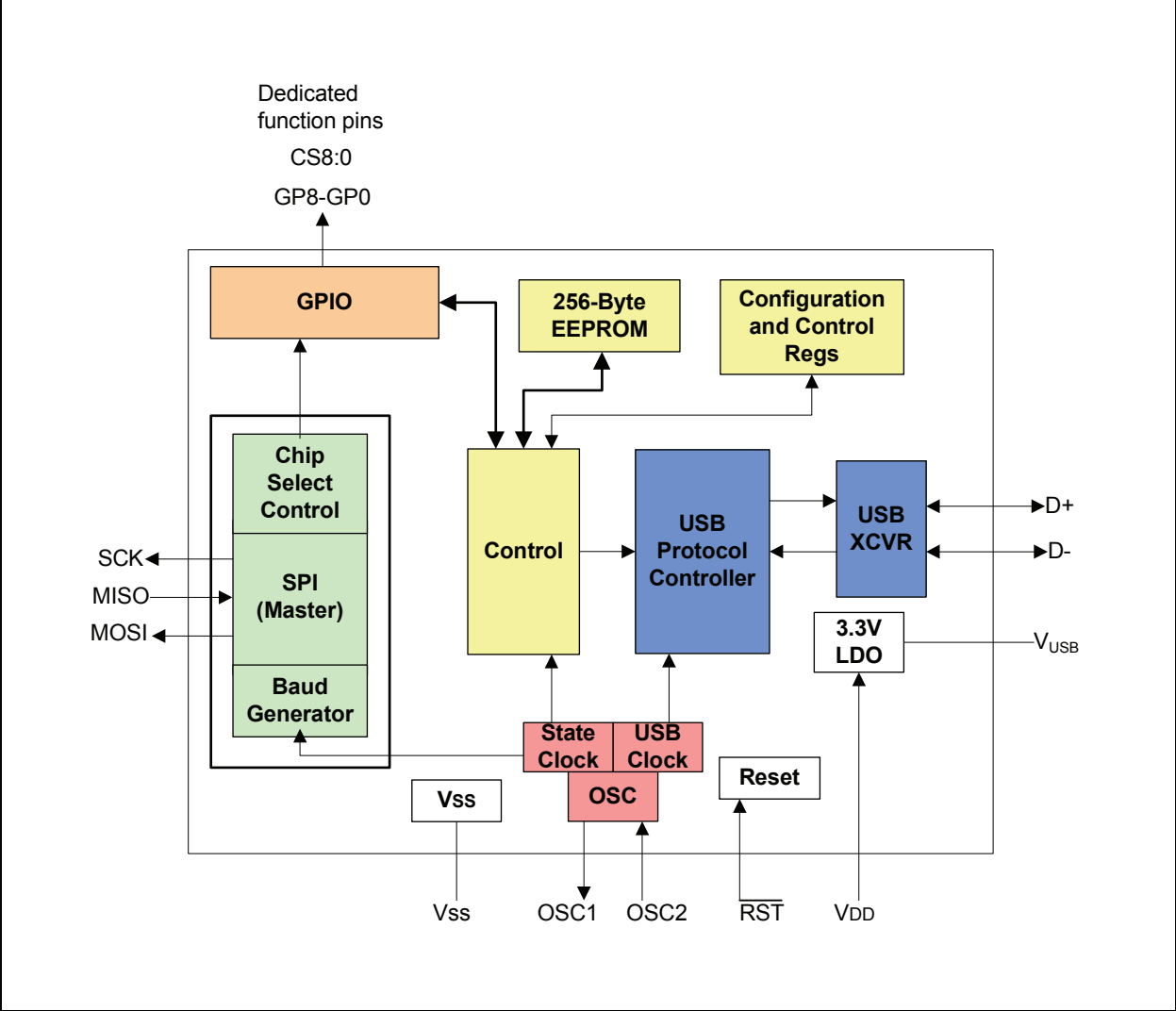


### Other

- USB Activity LED Output
- SSPND Output Pin (to signal USB Suspend state)
- USBCFG Output Pin (indicates when the enumeration is completed)
- Operating Voltage: 3.3-5.5V
- Oscillator Input: 12 MHz
- Industrial Operating Temperature: -40°C to +85°C

# MCP2210

## Block Diagram



## 1.0 FUNCTIONAL DESCRIPTION

The MCP2210 device is a USB-to-SPI Master converter which enables USB connectivity in applications that have an SPI interface. The device reduces external components by integrating the USB termination resistors.

The MCP2210 also has 256 bytes of integrated user EEPROM.

The MCP2210 has nine general purpose input/output pins. Seven pins have alternate functions to indicate USB and communication status. See [Table 1-1](#) and [Section 1.6 “GP Module”](#) for details about the pin functions.

**TABLE 1-1: PINOUT DESCRIPTION**

| MCP2210 |            | Symbol | Type | Standard Function (GPIO) | Alternate Function 1 (Chip Selects) | Alternate Function 2 (dedicated functions) | Description          |
|---------|------------|--------|------|--------------------------|-------------------------------------|--|----------------------|
| QFN     | SOIC, SSOP |        |      |                          |                                     |  |                      |
| 1       | 4          | RST    | I    | —                        | —                                   | —  | Reset input          |
| 2       | 5          | GP0    | I/O  | GPIO0                    | CS0                                 | —  | General Purpose I/O  |
| 3       | 6          | GP1    | I/O  | GPIO1                    | CS1                                 | —  | General Purpose I/O  |
| 4       | 7          | GP2    | I/O  | GPIO2                    | CS2                                 | USB Suspend                                | General Purpose I/O  |
| 5       | 8          | GP3    | I/O  | GPIO3                    | CS3                                 | SPI Transfer Traffic LED                   | General Purpose I/O  |
| 6       | 9          | MOSI   | O    | —                        | —                                   | —  | SPI Master output    |
| 7       | 10         | GP4    | I/O  | GPIO4                    | CS4                                 | USB Low Power                              | General Purpose I/O  |
| 8       | 11         | SCK    | O    | —                        | —                                   | —  | SPI Clock output     |
| 9       | 12         | GP5    | I/O  | GPIO5                    | CS5                                 | USB Configured                             | General Purpose I/O  |
| 10      | 13         | MISO   | I    | —                        | —                                   | —  | SPI Master input     |
| 11      | 14         | GP6    | I/O  | GPIO6                    | CS6                                 | External Interrupt                         | General Purpose I/O  |
| 12      | 15         | GP7    | I/O  | GPIO7                    | CS7                                 | SPI Bus Release ACK                        | General Purpose I/O  |
| 13      | 16         | GP8    | I/O  | GPIO8                    | CS8                                 | SPI Bus Release REQ                        | General Purpose I/O  |
| 14      | 17         | VUSB   | USB  | —                        | —                                   | —  | USB Regulator output |
| 15      | 18         | D-     | USB  | —                        | —                                   | —  | USB D-               |
| 16      | 19         | D+     | USB  | —                        | —                                   | —  | USB D+               |
| 17      | 20         | VSS    | GND  | —                        | —                                   | —  | Ground               |
| 18      | 1          | VDD    | P    | —                        | —                                   | —  | Power                |
| 19      | 2          | OSC1   | I    | —                        | —                                   | —  | Oscillator input     |
| 20      | 3          | OSC2   | O    | —                        | —                                   | —  | Oscillator output    |

# MCP2210

---

## 1.1 Supported Operating Systems

The following operating systems are supported:

- Windows XP/Vista/7
- Linux
- Mac OS

### 1.1.1 ENUMERATION

The MCP2210 will enumerate as a USB device after Power-on Reset (POR). The device enumerates as a Human Interface Device (HID) only.

#### 1.1.1.1 Human Interface Device (HID)

The MCP2210 enumerates as an HID, so the device can be configured and all the other functionalities can be controlled. A DLL package that facilitates I/O control through a custom interface is supplied by Microchip and is available on the product landing page.

## 1.2 Control Module

The control module is the heart of the MCP2210. All other modules are tied together and controlled via the control module. The control module manages the data transfers between the USB and the SPI, as well as command requests generated by the USB host controller, and commands for controlling the function of the SPI and I/O.

### 1.2.1 SPI INTERFACE

The control module interfaces to the SPI and USB modules.

### 1.2.2 INTERFACING TO THE DEVICE

The MCP2210 can be accessed for reading and writing via USB host commands. The device cannot be accessed and controlled via the SPI interface.

## 1.3 SPI Module

The MCP2210 SPI module provides the MOSI, MISO and SCK signals to the outside world. The module has the ability to control the GP pins (as Chip Select) only if these pins are configured for Chip Select operation.

### 1.3.1 SPI MODULE FEATURES

The SPI module has the following configurable features:

- Bit rates
- Delays
- Chip Select pin assignments (up to 9 Chip Select lines)

All the above features are available for customization using certain USB commands.

### 1.3.2 SPI MODULE POWER-UP CONFIGURATION

Default parameters:

- 1 Mbit
- 4 bytes to transfer per SPI transaction
- GP1 as Chip Select line

## 1.4 USB Protocol Controller

The USB controller in the MCP2210 is full-speed USB 2.0 compliant.

- HID only device used for:
  - SPI transfers
  - I/O control
  - EEPROM access
  - Chip configuration manipulation
- 128-byte buffer to handle data for SPI transfers
  - 64-byte transmit
  - 64-byte receive
- Fully configurable VID, PID assignments, string descriptors (stored on-chip) and chip power-up settings (default chip settings and SPI transfer parameters)
- Bus powered or self-powered

### 1.4.1 DESCRIPTORS

The string descriptors are stored internally in the MCP2210 and they can be changed so when the chip enumerates, the host gets the customer's own product and manufacturer names. They can be customized to the user's needs by using the Microchip provided configuration utility or a custom built application that will send the proper USB commands for storing the new descriptors into the chip.

### 1.4.2 USB EVENTS

The MCP2210 provides support for signaling important USB-related events such as:

- USB Suspend and Resume – these states are signaled on the GP2, if the pin is configured for its dedicated function
  - USB Suspend mode is entered when a suspend signaling event is detected on the USB bus
  - USB Resume is signaled when one of the following events is occurring:
    - a) Resume signaling is detected or generated
    - b) A USB Reset signal is detected
    - c) A device Reset occurs
- USB device enumerated successfully (this state is signaled if the GP4 is configured for its dedicated function)
- USB Low-Power mode

## 1.5 USB Transceiver

The MCP2210 has a built-in, USB 2.0, full-speed transceiver internally connected to the USB module.

The USB transceiver obtains power from the V<sub>USB</sub> pin, which is internally connected to a 3.3V internal regulator. The best electrical signal quality is obtained when V<sub>USB</sub> is locally bypassed with a high-quality ceramic capacitor.

The internal 3.3V regulator draws power from the V<sub>DD</sub> pin. In certain scenarios, where V<sub>DD</sub> is lower than 3.3V+ internal LDO dropout, the V<sub>USB</sub> pin must be tied to an external regulated 3.3V. This will allow the USB transceiver to work correctly, while the I/O voltage in the rest of the system can be lower than 3.3V. As an example, in a system where the MCP2210 is used and the I/O required is of 2.2V, the V<sub>DD</sub> of the chip will be tied to the 2.2V digital power rail, while the V<sub>USB</sub> pin must be connected to a regulated 3.3V power supply.

### 1.5.1 INTERNAL PULL-UP RESISTORS

The MCP2210 device has built-in pull-up resistors designed to meet the requirements for full-speed USB.

### 1.5.2 MCP2210 POWER OPTIONS

The following are the main power options for the MCP2210:

- USB Bus Powered (5V)
- Self Powered (from 3.3V to 5V), while the V<sub>USB</sub> pin is supplied with 3.3V (regulated). If the V<sub>DD</sub> is powered with 5V, then the V<sub>USB</sub> will be powered by the internal regulator and the V<sub>USB</sub> pin will need only a decoupling capacitor

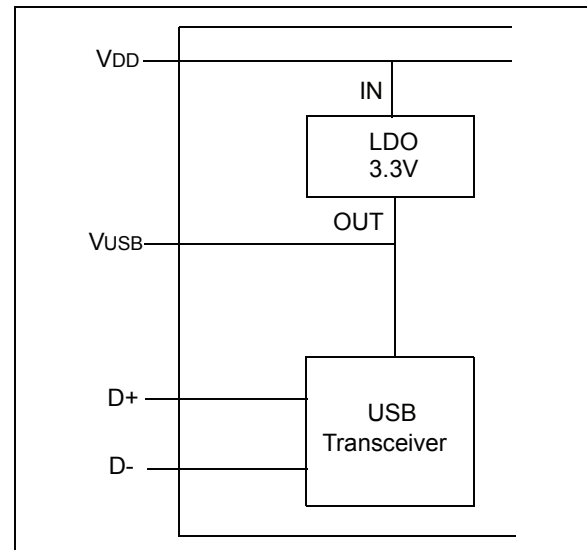
#### 1.5.2.1 Internal Power Supply Details

MCP2210 offers various options for power supply. To meet the required USB signaling levels, MCP2210 device incorporates an internal LDO used solely by the USB transceiver, in order to present the correct D+ / D- voltage levels.

Figure 1-1 shows the internal connections of the USB transceiver LDO in relation with the V<sub>DD</sub> power supply rail. The output of the USB transceiver LDO is tied to the V<sub>USB</sub> line.

A capacitor connected to the V<sub>USB</sub> pin is required if the USB transceiver LDO provides the 3.3V supply to the transceiver.

**FIGURE 1-1: MCP2210 INTERNAL POWER SUPPLY DETAILS**



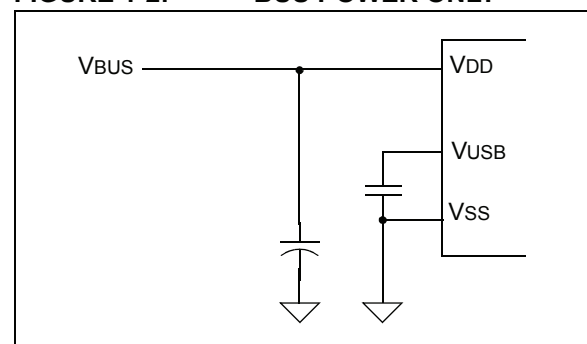
The provided V<sub>DD</sub> voltage has a direct influence on the voltage levels present on the GPIO and SPI module pins (GP8-GP0, MOSI, MISO and SCK). When V<sub>DD</sub> is 5V, all of these pins will have a logical '1' around 5V with the variations specified in [Section 4.1 "DC Characteristics"](#).

For applications that require a 3.3V logical '1' level, V<sub>DD</sub> must be connected to a power supply providing the 3.3V voltage. In this case, the internal USB transceiver LDO cannot provide the required 3.3V power. It is necessary to also connect the V<sub>USB</sub> pin of the MCP2210 to the 3.3V power supply rail. This way, the USB transceiver is powered up directly from the 3.3V power supply.

#### 1.5.2.2 USB Bus Powered (5V)

In Bus Power Only mode, the entire power for the application is drawn from the USB (see [Figure 1-2](#)). This is effectively the simplest power method for the device.

**FIGURE 1-2: BUS POWER ONLY**



# MCP2210

In order to meet the inrush current requirements of the USB 2.0 specifications, the total effective capacitance appearing across VBUS and ground must be no more than 10  $\mu\text{F}$ . If it is more than 10  $\mu\text{F}$ , some kind of inrush limiting is required. For more details on Inrush Current Limiting, see the current *Universal Serial Bus Specification*.

According to the USB 2.0 specification, all USB devices must also support a Low-Power Suspend mode. In the USB Suspend mode, devices must consume no more than 500  $\mu\text{A}$  (or 2.5 mA for high powered devices that are remote wake-up capable) from the 5V VBUS line of the USB cable.

The host signals the USB device to enter Suspend mode by stopping all USB traffic to that device for more than 3 ms.

The USB bus provides a 5V voltage. However, the USB transceiver requires 3.3V for the signaling (on D+ and D- lines).

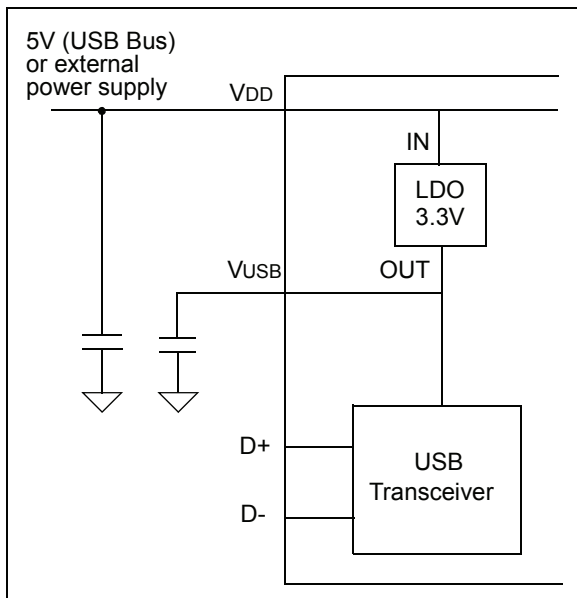
During USB Suspend mode, the D+ or D- pull-up resistor must remain active, which will consume some of the allowed suspend current budget (500  $\mu\text{A}$ /2.5 mA).

The VUSB pin is required to have an external bypass capacitor. It is recommended that the capacitor be a ceramic cap, between 0.22 and 0.47  $\mu\text{F}$ .

Figure 1-3 shows a circuit where the MCP2210 internal LDO is used to provide 3.3V to the USB transceiver.

The voltage on the VDD affects the voltage levels onto the GP and SPI module pins (GP8-GP0, MOSI, MISO and SCK). With VDD at 5V, these pins will have a logic '1' of 5V with the variations specified in Section 4.1 "DC Characteristics".

**FIGURE 1-3: TYPICAL POWER SUPPLY OPTION USING THE 5V PROVIDED BY THE USB**

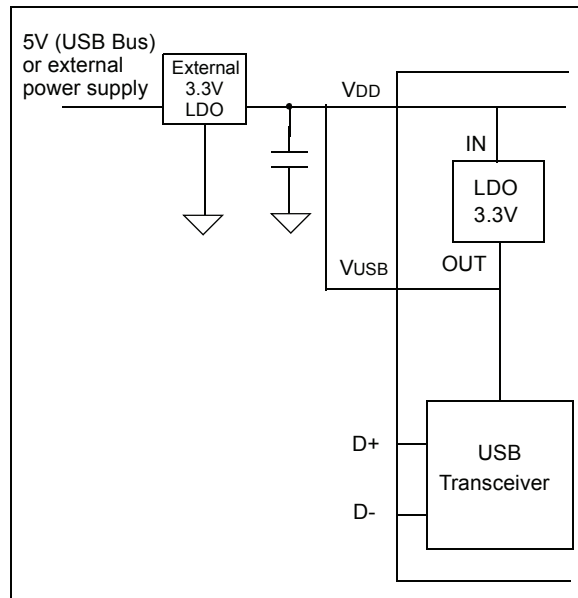


## 1.5.2.3 3.3V – Self Powered

Typically, many embedded applications are using 3.3V or lower power supplies. When such an option is available in the target system, MCP2210 can be powered up (VDD) from the existing power supply rail. The typical connections for MCP2210 powered from 3.3V rail are shown in Figure 1-4.

In this example MCP2210 has both VDD and VUSB lines tied to the 3.3V rail. These tied connections disable the internal USB transceiver LDO of the MCP2210 to regulate the power supply on VUSB pin. Another consequence is that the '1' logical level on the GP and SPI pins will be at the 3.3V level, in accordance with the variations specified in Section 4.1 "DC Characteristics".

**FIGURE 1-4: USING AN EXTERNALLY PROVIDED 3.3V POWER SUPPLY**



## 1.6 GP Module

The GP module features nine I/O lines.

### 1.6.1 CONFIGURABLE PIN FUNCTIONS

The pins can be configured as:

- GPIO – individually configurable, general purpose input or output
- Chip Select pins – used by the SPI module
- Alternate function pins – used for miscellaneous features such as:
  - $\overline{\text{SSPND}}$  – USB Suspend and Resume states
  - $\overline{\text{USBCFG}}$  – indicates USB configuration status
  - $\overline{\text{LOWPWR}}$  – signals when the host does not accept the requirements (presented during enumeration) and the chip is not configured. In this mode, the whole system powered from the USB host should draw up to 100 mA.
  - External Interrupt Input – used to count external events
  - SPI bus Release Request – used to request SPI bus access from the MCP2210
  - SPI bus Release Acknowledge – used to acknowledge when the MCP2210 has released the SPI bus
  - LED – indicates SPI traffic led

#### 1.6.1.1 GPIO Pins Function

The GP pins (if enabled for GPIO functionality) can be used as digital inputs/outputs.

These pins can be read (both inputs and outputs) and written (only the outputs).

#### 1.6.1.2 Chip Select Pins Function

The GP pins (if enabled for the Chip Select functionality) are controlled by the SPI module. Their Idle/Active value is determined by the SPI transfer parameters.

#### 1.6.1.3 $\overline{\text{SSPND}}$ Pin Function

The GP2 pin (if enabled for this functionality) reflects the USB state (Suspend/Resume). The pin is active 'low' when the Suspend state has been issued by the USB host.

Likewise, the pin drives 'high' after the Resume state is achieved.

This pin allows the application to go into Low-Power mode when USB communication is suspended, and switches to a full active state when USB activity is resumed.

#### 1.6.1.4 $\overline{\text{USBCFG}}$ Pin Function

The GP5 pin (if enabled for this functionality) starts out 'high' during power-up or after Reset, and goes 'low' after the device successfully configures to the USB. The pin will go 'high' when in Suspend mode and 'low' when the USB resumes.

#### 1.6.1.5 $\overline{\text{LOWPWR}}$ Pin Function

The GP4 pin (if enabled for this functionality) starts out 'low' during power-up or after Reset, and goes 'high' after the device successfully configures to the USB. The pin will go 'low' when in Suspend mode and 'high' when the USB resumes.

#### 1.6.1.6 External Interrupt Input Pin Function

The GP4 pin (if enabled for this functionality) is used as an interrupt input pin and it will count interrupt events such as:

- Falling edges
- Rising edges
- Low-logic pulses
- High-logic pulses

#### 1.6.1.7 SPI Bus Release Request Pin Function

The GP8 pin (if enabled for this functionality) is used by an external device to request the MCP2210 to release the SPI bus. This way, more than one SPI master can have access to the SPI slave chips on the bus. When this pin is driven 'low', the MCP2210 will examine the request and, based on the conditions and internal logic, it might release the SPI bus. If there is an ongoing SPI transfer taking place at the moment when an external device requests the bus, MCP2210 will release it after the transfer is completed or if the USB host cancels the current SPI transfer.

#### 1.6.1.8 SPI Bus Release Acknowledge Pin Function

The GP7 pin (if enabled for this functionality) is used by the MCP2210 to signal back if the SPI bus was released. When a SPI bus release request is registered by the MCP2210, based on the condition and internal logic, the chip might release the bus. The bus is released immediately if there is no SPI transfer taking place, or it will do so after the current SPI transfer is finished or cancelled by the USB host.

#### 1.6.1.9 LED Pin Function

The GP3 pin (if enabled for this functionality) is used as an SPI traffic indication. When an SPI transfer is taking place (active state for this pin), this pin will be driven 'low'. When there is no SPI traffic taking place, the pin is in its inactive state or logic 'high'.



# MCP2210

## 1.7 EEPROM Module

The EEPROM module is a 256-byte array of nonvolatile memory. The memory locations are accessed for read/write operations solely via USB host commands. The memory cells for data EEPROM are rated to endure thousands of erase/write cycles, up to 100K for EEPROM.

Data retention without refresh is conservatively estimated to be greater than 40 years.

## 1.8 Reset/POR

### 1.8.1 RESET PIN

The  $\overline{\text{RST}}$  pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. MCP2210 has a noise filter in the Reset path which detects and ignores small pulses.

### 1.8.2 POR

A POR pulse is generated on-chip whenever  $V_{DD}$  rises above a certain threshold. This allows the device to start in the initialized state when  $V_{DD}$  is adequate for operation.

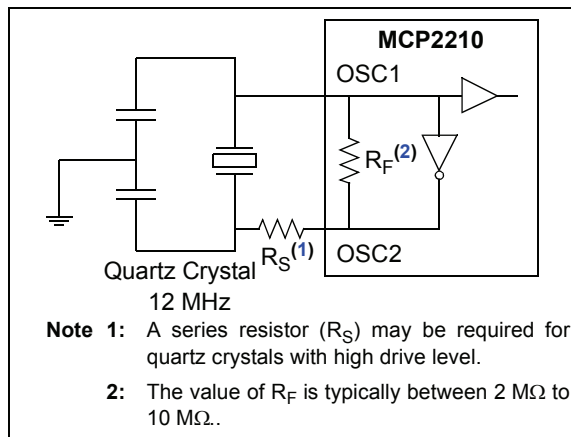
To take advantage of the POR circuitry, tie the  $\overline{\text{RST}}$  pin through a resistor (1 k $\Omega$  to 10 k $\Omega$ ) to  $V_{DD}$ . This will eliminate external RC components usually needed to create a POR delay.

When the device starts normal operation (i.e., exits the Reset condition), the device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not achieved, the device must be held in Reset until the operating conditions are met.

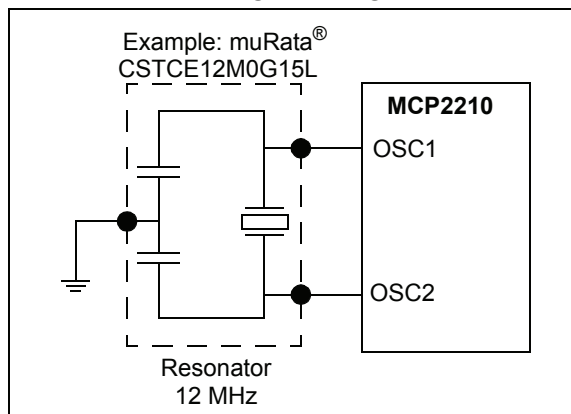
## 1.9 Oscillator

The input clock must be 12 MHz to provide the proper frequency for the USB module. USB full-speed is nominally 12 Mb/s. The clock input accuracy is  $\pm 0.25\%$  (2,500 ppm maximum).

**FIGURE 1-5: QUARTZ CRYSTAL OPERATION**



**FIGURE 1-6: CERAMIC RESONATOR OPERATION**





## 2.0 MCP2210 FUNCTIONAL DESCRIPTION

The MCP2210 uses NVRAM to store relevant chip settings. These settings are loaded by the chip during the power-up process and they are used for GP designation and SPI transfers.

The NVRAM settings at power-up (or Reset) are loaded into the RAM portion of the chip and they can be altered through certain USB commands. This is very useful since it allows dynamic reconfiguring of the GPs or SPI transfer parameters. A practical example to illustrate this mechanism is a system which uses at least two SPI slave chips and the GPs in the MCP2210 for various GPIO purposes. The default SPI settings might be ok for one of the SPI slave chips, but not for the 2<sup>nd</sup>. At first, the PC application will make an SPI transfer to the first chip, using the NVRAM copy of the SPI settings. Then, by sending a certain USB command, the SPI transfer settings residing in RAM will be altered in order to fit the SPI transfer requirements of the second chip.

Also, if the altered SPI transfer settings are needed to be the default power-up (or Reset) settings for SPI, the user can send a series of USB commands in order to store the current (RAM) SPI settings into NVRAM. In this way, these new settings will be the power-up default SPI settings.

The NVRAM settings and EEPROM contents can be protected by password access means, or they can be permanently locked without any possible further modification.

### 2.1 MCP2210 NVRAM Settings

The chip settings that can be stored in the NVRAM area are as follows:

- SPI transfer parameters:
  - SPI bit rate
  - SPI mode
  - Idle Chip Select values
  - Active Chip Select values
  - SPI transfer configurable delays
  - Number of bytes to read/write for the given SPI transfer
- GP designation:
  - GPIO
  - Chip Select
  - Dedicated function
- GPIO default direction (applies only to those GPs designated as GPIOs)
- GPIO default output value (applies only to those GPs designated as output GPIOs)

- Chip mode flags:
  - Remote wake-up capability
  - External Interrupt Pin mode (applies only when GP6 is designated for this function)
  - SPI bus release enable/disable – enable/disable the release of the SPI bus when there is no SPI transfer (useful when more than one SPI master on the bus)
- NVRAM Access mode:
  - Full access (no protection – factory default)
  - Password protection
  - Permanently locked
- Password (relevant when password protection mechanism is active)

The specified settings are loaded at power-up or Reset moments, and they can be altered through certain USB commands.

When a NVRAM conditional access method is already in place, such as password protection, the NVRAM settings modification is permitted only when the user has supplied the correct password for the chip. The RAM settings can be altered even when a password protection or permanent lock mechanism are in place. This allows the user to communicate with various SPI slave chips without knowing the password, but it will not allow the modification of the power-up default settings in NVRAM.

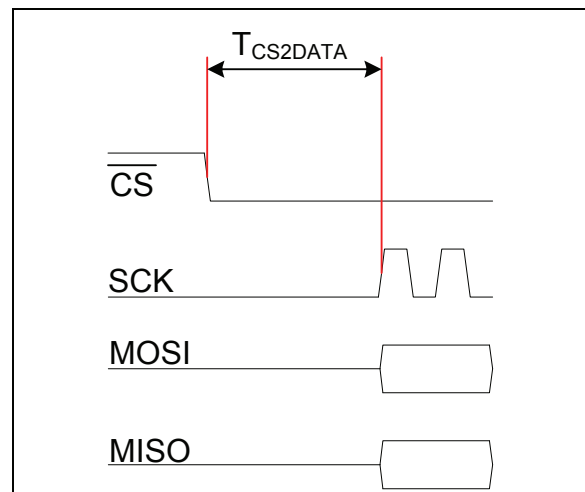
### 2.2 SPI Transfers

The MCP2210 device provides advanced SPI communication features such as configurable delays and multiple Chip Select support.

The configurable delays are related to certain aspects of the SPI transfer:

- The delay between the assertion of Chip Select(s) and the first data byte (Figure 2-1)

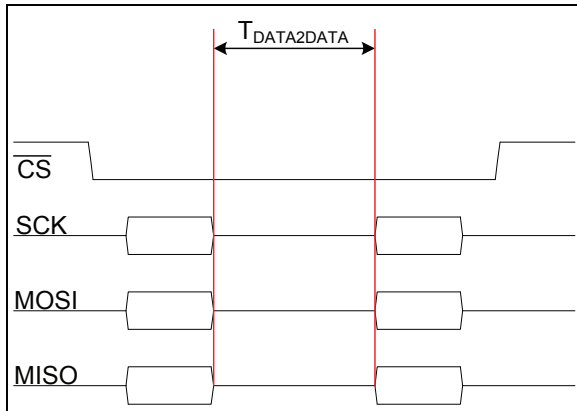
**FIGURE 2-1: CHIP SELECT TO DATA DELAY**



# MCP2210

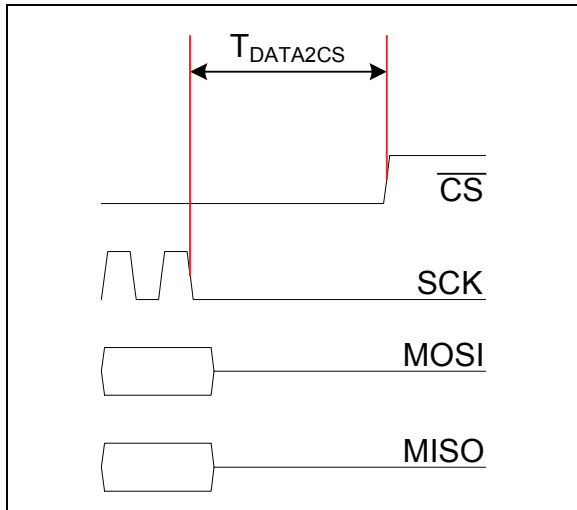
- The delay between subsequent data bytes  
(Figure 2-2)

**FIGURE 2-2: DATA-TO-DATA DELAY**



- The delay between the end of the last byte (of the SPI transfer) and the de-assertion of the Chip Select(s)

**FIGURE 2-3: DATA TO CHIP SELECT DELAY**



For a particular SPI transfer, the user can choose any number (out of the available ones) of Chip Select pins. The SPI transfer parameters contain two fields where the user will specify the Chip Select values when the SPI transfer is active/idle. This mechanism allows the user to specify any combination of Chip Select values for the Idle mode and some other combination for the Active mode (SPI transfer active).

## 3.0 USB COMMANDS/RESPONSES DESCRIPTION

MCP2210 implements the HID interface for all the device-provided functionalities. The chip uses a command/response mechanism for the USB engine. This means that for every USB command sent (by the USB host) to the MCP2210, it will always reply with a response packet.

The MCP2210 USB commands can be grouped by their provided features as follows:

- [NVRAM Settings](#)
  - Read/Write NVRAM related parameters
  - Send access password
- [Read/Write RAM Settings](#) (copied from NVRAM at power-up or Reset):
  - Read/Write (volatile – RAM stored settings) SPI transfer settings
  - Read/Write (volatile – RAM stored settings) chip settings
  - Read/Write (volatile – RAM stored settings) GPIO direction
  - Read/Write (volatile – RAM stored settings) GPIO output values
- [Read/Write EEPROM Memory](#)
- [External Interrupt Pin \(GP6\) Event Status](#)
- [SPI Data Transfer](#):
  - Read/Write SPI transfer data
  - Cancels the ongoing SPI transfer
  - SPI bus release manipulation
- [Chip Status](#) and Unsupported commands

## 3.1 NVRAM Settings

The commands in this category are related to the NVRAM settings manipulation.

### 3.1.1 SET CHIP SETTINGS POWER-UP DEFAULT

**TABLE 3-1: COMMAND STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – command code   |
| 1          | 0x20 – <b>Set Chip Settings Power-up Default</b> – sub-command code  |
| 2          | 0x00 – <b>Reserved</b>   |
| 3          | 0x00 – <b>Reserved</b>   |
| 4          | <b>GP0 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 5          | <b>GP1 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 6          | <b>GP2 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 7          | <b>GP3 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 8          | <b>GP4 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |

# MCP2210

**TABLE 3-1: COMMAND STRUCTURE (CONTINUED)**

| Byte Index | Meaning  |
|------------|--|
| 9          | <b>GP5 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 10         | <b>GP6 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 11         | <b>GP7 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 12         | <b>GP8 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 13         | <b>Default GPIO Output – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB        –        –        –        –        –        –        LSB</li> <li>          GP7VAL GP6VAL GP5VAL GP4VAL GP3VAL GP2VAL GP1VAL GP0VAL</li> </ul>  |
| 14         | <b>Default GPIO Output – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB        –        –        –        –        –        –        LSB</li> <li>          x        x        x        x        x        x        x        GP8VAL</li> </ul> <p>where x = Don't Care</p> |
| 15         | <b>Default GPIO Direction – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB        –        –        –        –        –        –        LSB</li> <li>          GP7DIR GP6DIR GP5DIR GP4DIR GP3DIR GP2DIR GP1DIR GP0DIR</li> </ul>   |
| 16         | <b>Default GPIO Direction – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB        –        –        –        –        –        –        LSB</li> <li>          x        x        x        x        x        x        x        GP8DIR</li> </ul>                          |

**TABLE 3-1: COMMAND STRUCTURE (CONTINUED)**

| Byte Index   | Meaning   |
|--|---|
| 17   | <b>Other Chip Settings – Enable/Disable Wake-up, Interrupt Counting, SPI Bus Release Options</b> <ul style="list-style-type: none"> <li>• Bit 7 – Don't Care</li> <li>• Bit 6 – Don't Care</li> <li>• Bit 5 – Don't Care</li> <li>• Bit 4 – Remote Wake-up Enabled/Disabled                             <ul style="list-style-type: none"> <li>- 0 – Remote Wake-up Disabled</li> <li>- 1 – Remote Wake-up Enabled</li> </ul> </li> <li>• Bit 3 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 2 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 1 – Dedicated Function – Interrupt Pin mode                             <ul style="list-style-type: none"> <li>- b111 – Reserved</li> <li>- b110 – Reserved</li> <li>- b101 – Reserved</li> <li>- b100 – Count High Pulses</li> <li>- b011 – Count Low Pulses</li> <li>- b010 – Count Rising Edges</li> <li>- b001 – Count Falling Edges</li> <li>- b000 – No Interrupt Counting</li> </ul> </li> <li>• Bit 0 – SPI Bus Release Enable                             <ul style="list-style-type: none"> <li>- 0 = SPI Bus is Released Between Transfer</li> <li>- 1 = SPI Bus is Not Released by the MCP2210 between transfers</li> </ul> </li> </ul> |
| 18   | <b>NVRAM Chip Parameters Access Control</b> <ul style="list-style-type: none"> <li>• 0x00 – Chip settings not protected</li> <li>• 0x40 – Chip settings protected by password access</li> <li>• 0x80 – Chip settings permanently locked</li> </ul>  |
| 19   | <b>New Password Character 0 (Note 1)</b>  |
| 20   | <b>New Password Character 1 (Note 1)</b>  |
| 21   | <b>New Password Character 2 (Note 1)</b>  |
| 22   | <b>New Password Character 3 (Note 1)</b>  |
| 23   | <b>New Password Character 4 (Note 1)</b>  |
| 24   | <b>New Password Character 5 (Note 1)</b>  |
| 25   | <b>New Password Character 6 (Note 1)</b>  |
| 26   | <b>New Password Character 7 (Note 1)</b>  |
| 27-63  | <b>Reserved</b> (fill with 0x00)  |
| <b>Note 1:</b> When the password does not need to change, this field must be filled with 0 (it applies to (byte index 19 to 26). |   |

### 3.1.1.1 Responses

**TABLE 3-2: RESPONSE 1 STRUCTURE**

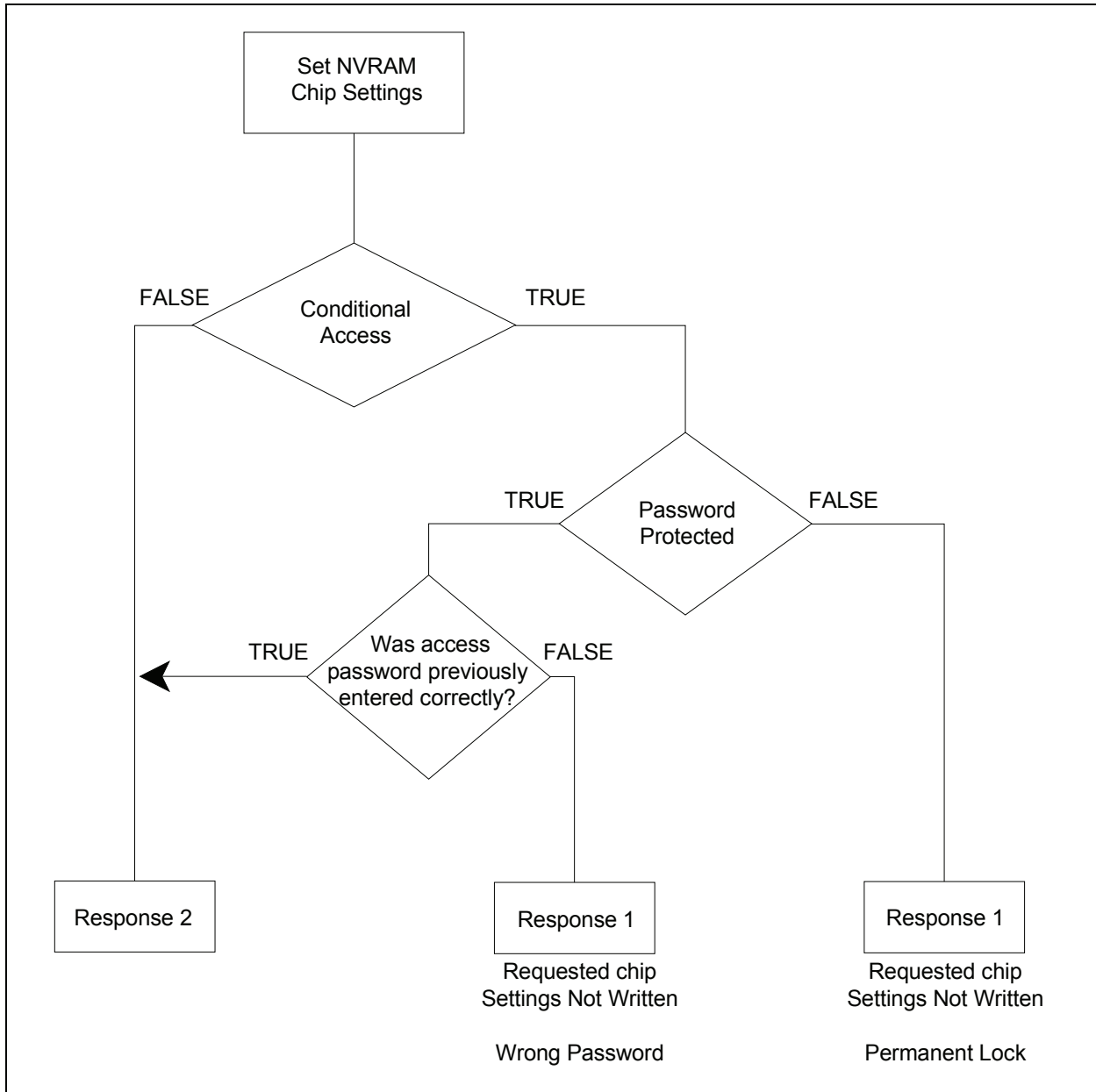
| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code  |
| 1          | 0xFB – <b>Blocked Access</b> – The provided password is not matching the one stored in the chip, or the settings are permanently locked. |
| 2-63       | <b>Don't Care</b>  |

# MCP2210

**TABLE 3-3: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code       |
| 1          | 0x00 – <b>Command Completed Successfully</b> – settings written                   |
| 2          | 0x20 – <b>Sub-command Echoed Back</b> for Set Chip Settings Power-up Default code |
| 3-63       | <b>Don't Care</b>   |

**FIGURE 3-1: SET CHIP SETTINGS POWER-UP DEFAULT LOGIC FLOW**



## 3.1.2 SET SPI POWER-UP TRANSFER SETTINGS

TABLE 3-4: COMMAND STRUCTURE

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – command code   |
| 1          | 0x10 – <b>Set SPI Power-up Transfer Settings</b> – sub-command code  |
| 2          | 0x00 – <b>Reserved</b>   |
| 3          | 0x00 – <b>Reserved</b>   |
| 4          | <b>Bit Rate (Byte 3)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte = 0x00  |
| 5          | <b>Bit Rate (Byte 2)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte = 0x1B  |
| 6          | <b>Bit Rate (Byte 1)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte = 0xB7  |
| 7          | <b>Bit Rate (Byte 0)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte = 0x00  |
| 8          | <b>Idle Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0   |
| 9          | <b>Idle Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – – – LSB<br>x x x x x x x CS8  |
| 10         | <b>Active Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0   |
| 11         | <b>Active Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – – – LSB<br>x x x x x x x CS8  |
| 12         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the CS being asserted and the first byte of data is required, the value will be 0x0005.<br>- Fill this byte position with: 0x05                     |
| 13         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the CS being asserted and the first byte of data is required, the value will be 0x0005.<br>- Fill this byte position with: 0x00                    |
| 14         | <b>Last Data Byte to CS</b> (de-asserted) delay (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted is required, the value will be 0x0005.<br>- Fill this byte position with: 0x05  |
| 15         | <b>Last Data Byte to CS</b> (de-asserted) delay (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted is required, the value will be 0x0005.<br>- Fill this byte position with: 0x00 |
| 16         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- Fill this byte position with: 0x05                                 |



# MCP2210

**TABLE 3-4: COMMAND STRUCTURE (CONTINUED)**

| Byte Index | Meaning   |
|------------|---|
| 17         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- Fill this byte position with: 0x00 |
| 18         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (low byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- Fill this byte position with: 0xE2                     |
| 19         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (high byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- Fill this byte position with: 0x04                    |
| 20         | <b>SPI Mode</b> <ul style="list-style-type: none"> <li>• 0x00 – SPI mode 0</li> <li>• 0x01 – SPI mode 1</li> <li>• 0x02 – SPI mode 2</li> <li>• 0x03 – SPI mode 3</li> </ul>  |
| 21 - 63    | <b>Reserved</b> – fill with 0x00  |

### 3.1.2.1 Responses

**TABLE 3-5: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code                                  |
| 1          | 0xFB – <b>Blocked Access</b> – Access password has not been provided or the settings are permanently locked. |
| 2-63       | <b>Don't Care</b>  |

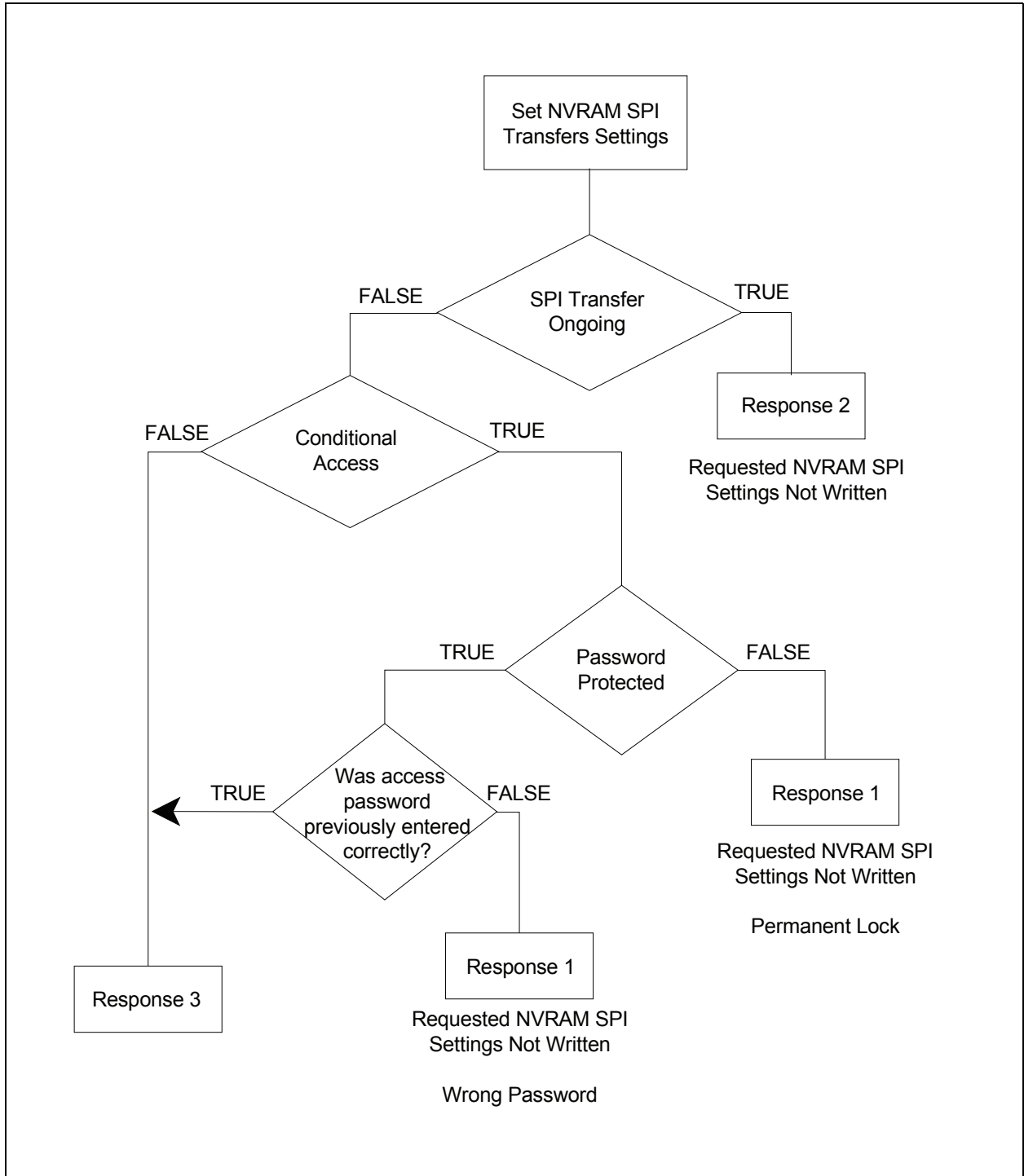
**TABLE 3-6: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code |
| 1          | 0xF8 – <b>USB Transfer in Progress</b> – settings not written               |
| 2          | 0x10 – <b>Sub-command Echoed Back</b> – set SPI power-up transfer settings  |
| 3-63       | <b>Don't Care</b>   |

**TABLE 3-7: RESPONSE 3 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code       |
| 1          | 0x00 – <b>Command Completed Successfully</b> – settings written                   |
| 2          | 0x10 – <b>Sub-command Echoed Back</b> for Set SPI Power-up Transfer Settings code |
| 3-63       | <b>Don't Care</b>   |

**FIGURE 3-2: SET SPI POWER-UP TRANSFER SETTINGS LOGIC FLOW**



# MCP2210

## 3.1.3 SET USB POWER-UP KEY PARAMETERS

**TABLE 3-8: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – command code  |
| 1          | 0x30 – <b>Set USB Power-up Key Parameters</b> – sub-command code  |
| 2          | 0x00 – <b>Reserved</b>  |
| 3          | 0x00 – <b>Reserved</b>  |
| 4          | <b>VID</b> – 16-bit value (low byte)  |
| 5          | <b>VID</b> – 16-bit value (high byte)   |
| 6          | <b>PID</b> – 16-bit value (low byte)  |
| 7          | <b>VID</b> – 16-bit value (high byte)   |
| 8          | <b>Chip Power Option</b> (as per USB specs – Chapter 9) <ul style="list-style-type: none"><li>• Bit 7 – Host Powered (1 = yes; 0 = no)</li><li>• Bit 6 – Self Powered (1 = yes; 0 = no)</li><li>• Bit 5 – Remote Wake-up Capable</li><li>• Bit 4 – Reserved – fill with 0</li><li>• Bit 3 – Reserved – fill with 0</li><li>• Bit 2 – Reserved – fill with 0</li><li>• Bit 1 – Reserved – fill with 0</li><li>• Bit 0 – Reserved – fill with 0</li></ul> <b>Note:</b> Only bit 6 or bit 7 should be set, not both. |
| 9          | <b>Requested Current Amount from USB Host</b> (quanta of 2 mA)<br><u>Example:</u> For 100 mA fill this byte index with 50 (in decimal) or 0x32.   |
| 10-63      | <b>Reserved</b> – fill with 0x00  |

### 3.1.3.1 Responses

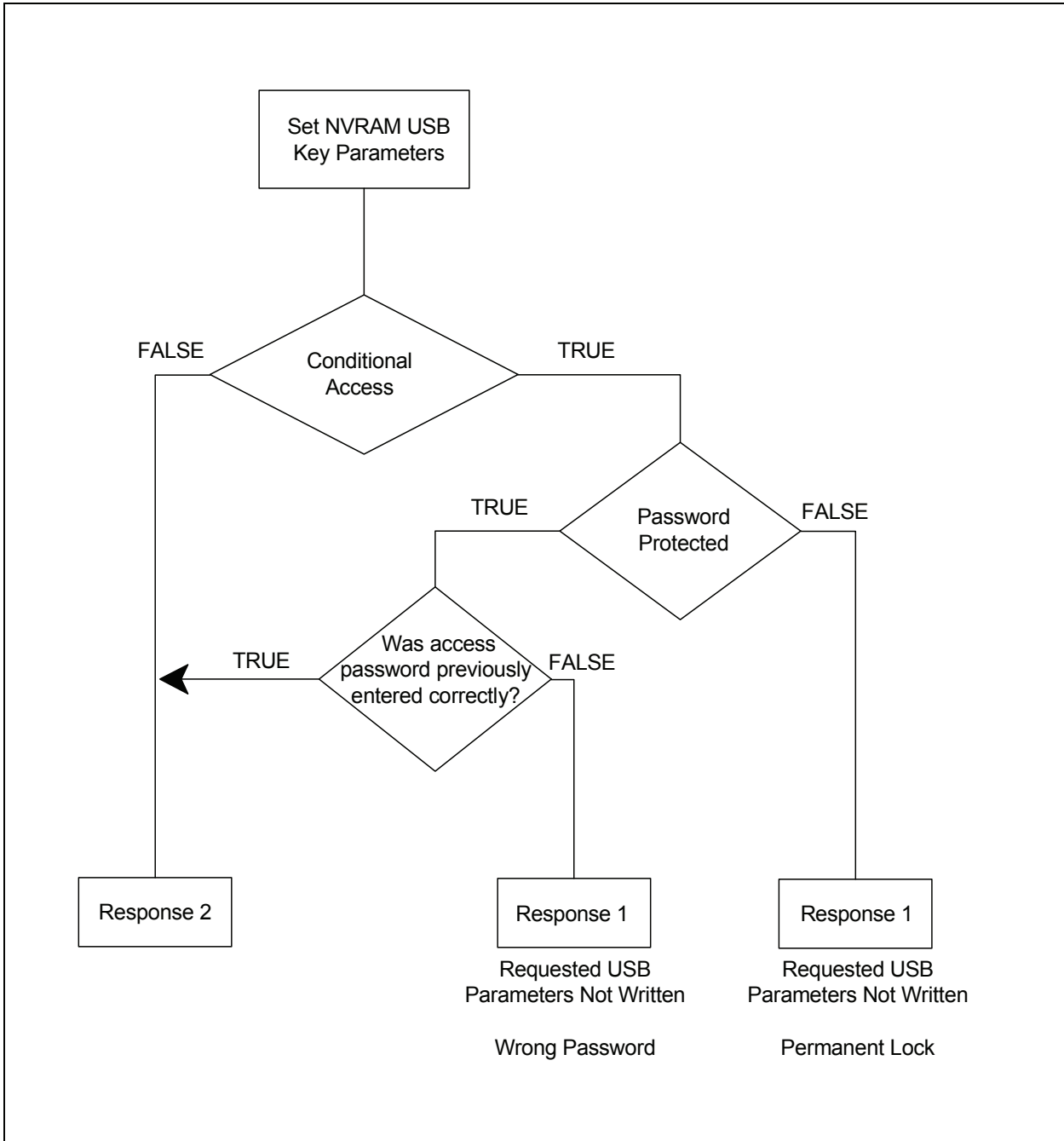
**TABLE 3-9: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echo back the given command code  |
| 1          | 0xFB – <b>Blocked Access</b> – The provided password is not matching the one stored in the chip or the settings are permanently locked. |
| 2-63       | <b>Don't Care</b>   |

**TABLE 3-10: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echo back the given command code     |
| 1          | 0x00 – <b>Command Completed Successfully</b> – Settings written                |
| 2          | 0x30 – <b>Sub-command Echoed Back</b> for Set USB Power-up Key Parameters code |
| 3-63       | <b>Don't Care</b>  |

**FIGURE 3-3: SET USB POWER-UP KEY PARAMETERS LOGIC FLOW**



# MCP2210

## 3.1.4 SET USB MANUFACTURER NAME

**TABLE 3-11: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – command code  |
| 1          | 0x50 – <b>Set USB Manufacturer Name</b> – sub-command code  |
| 2          | 0x00 – <b>Reserved</b>  |
| 3          | 0x00 – <b>Reserved</b>  |
| 4          | <b>Total USB String Descriptor Length</b> (this is the length of the Manufacturer string, multiplied by 2 + 2)<br><u>Example:</u> “Microchip Technology Inc.” has 25 Unicode characters.<br>- The value to be filled in is: $(25 \times 2) + 2 = 52$ (decimal) = 0x34 |
| 5          | <b>USB String Descriptor ID</b> – always fill with 0x03   |
| 6          | <b>Unicode Character Low Byte</b><br><u>Example:</u> For the “Microchip Technology Inc.” Unicode string, place here the low byte of the Unicode for character “M”.<br>- Fill this index with 0x4D   |
| 7          | <b>Unicode Character High Byte</b><br><u>Example:</u> For the “Microchip Technology Inc.” Unicode string, place here the high byte of the Unicode for character “M”.<br>- Fill this index with 0x00   |
| 8-63       | Fill in the remaining Unicode characters in the string  |

### 3.1.4.1 Responses

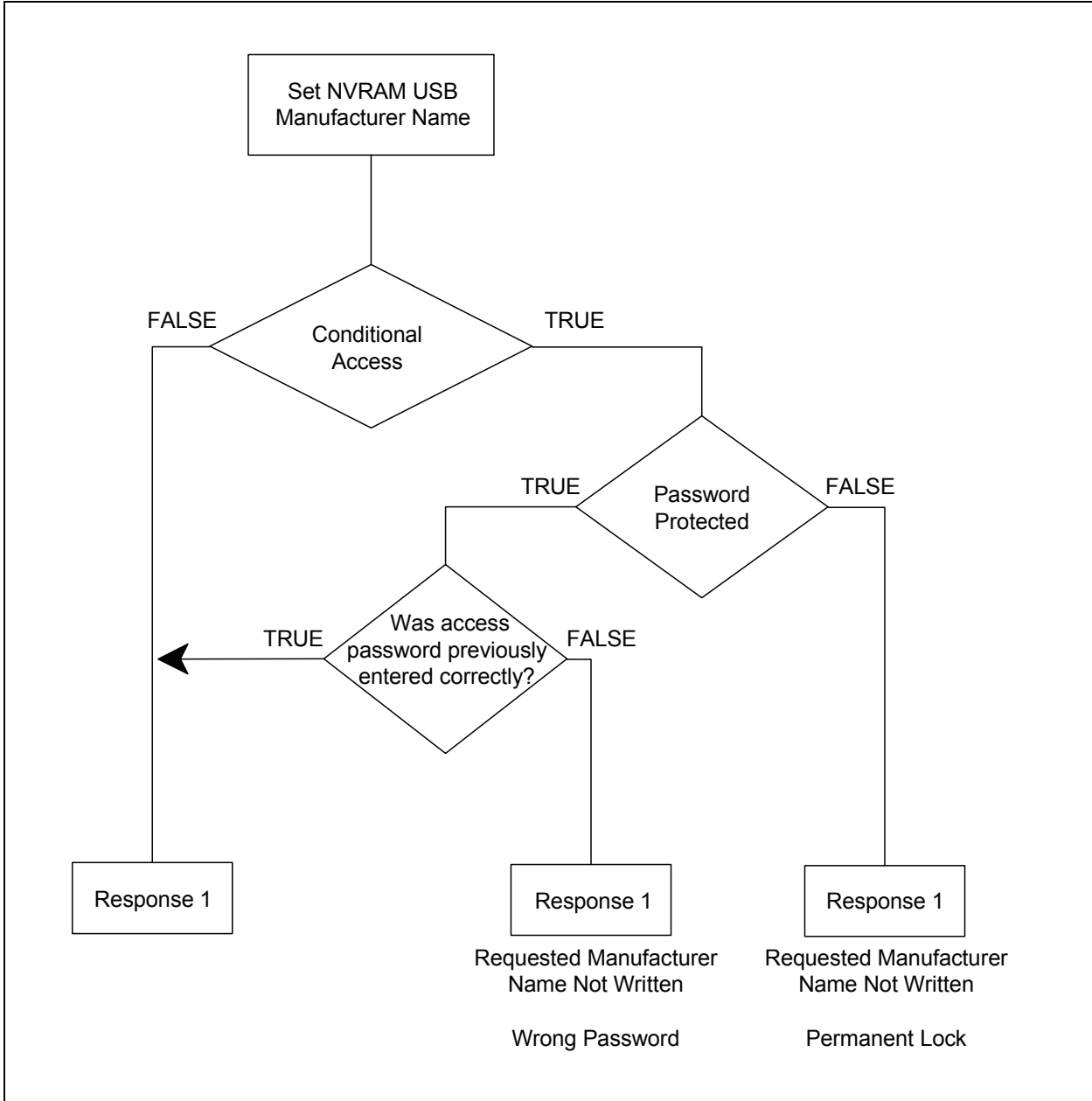
**TABLE 3-12: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code   |
| 1          | 0xFB – <b>Blocked Access</b> – The provided password is not matching the one stored in the chip or the settings are permanently locked. |
| 2-63       | <b>Don't Care</b>   |

**TABLE 3-13: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b> – settings written             |
| 2          | 0x50 – <b>Sub-command Echoed Back</b> for Set USB Manufacturer Name code    |
| 3-63       | <b>Don't Care</b>   |

**FIGURE 3-4: SET USB MANUFACTURER LOGIC FLOW**



# MCP2210

## 3.1.5 SET USB PRODUCT NAME

**TABLE 3-14: COMMAND STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – command code   |
| 1          | 0x40 – <b>Set USB Product Name</b> – sub-command code  |
| 2          | 0x00 – <b>Reserved</b>   |
| 3          | 0x00 – <b>Reserved</b>   |
| 4          | <b>Total USB String Descriptor Length</b> (this is the length of the Product string multiplied by 2 + 2)<br><u>Example:</u> “MCP2210 USB to SPI Master” has 25 Unicode characters.<br>- The value to be filled in is: $(25 * 2) + 2 = 52$ (decimal) = 0x34 |
| 5          | <b>USB String Descriptor ID</b> – always fill with 0x03  |
| 6          | <b>Unicode Character Low Byte</b><br><u>Example:</u> For the “MCP2210 USB to SPI Master” Unicode string, place here the low byte of the Unicode for character “M”.<br>- Fill this index with 0x4D  |
| 7          | <b>Unicode Character High Byte</b><br><u>Example:</u> For the “MCP2210 USB to SPI Master” Unicode string, place here the high byte of the Unicode for character “M”.<br>- Fill this index with 0x00  |
| 8-63       | Fill in the remaining Unicode characters in the string   |

### 3.1.5.1 Responses

**TABLE 3-15: RESPONSE 1 STRUCTURE**

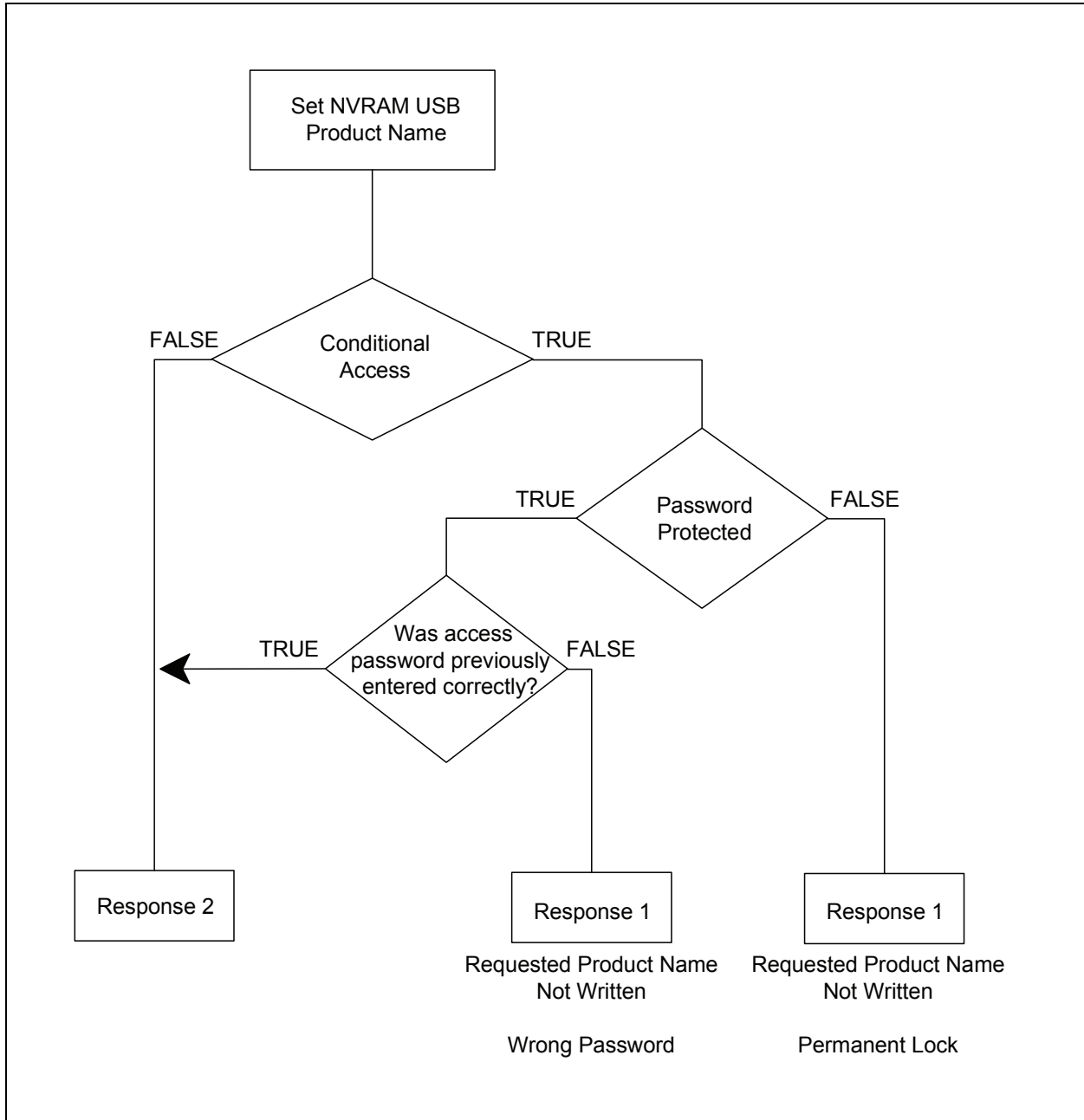
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code   |
| 1          | 0xFB – <b>Blocked Access</b> – The provided password is not matching the one stored in the chip or the settings are permanently locked. |
| 2-63       | <b>Don't Care</b>   |

**TABLE 3-16: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x60 – <b>Set Chip NVRAM Parameters</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b> – settings written             |
| 2          | 0x40 – <b>Sub-command Echoed Back for Set USB Product Name code</b>         |
| 3-63       | <b>Don't Care</b>   |



FIGURE 3-5: SET USB PRODUCT NAME LOGIC FLOW



# MCP2210

## 3.1.6 GET SPI POWER-UP TRANSFER SETTINGS

**TABLE 3-17: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – command code                     |
| 1          | 0x10 – <b>Get SPI Power-up Transfer Settings</b> – sub-command code |
| 2          | 0x00 – <b>Reserved</b>  |
| 3-63       | 0x00 – <b>Reserved</b>  |

### 3.1.6.1 Responses

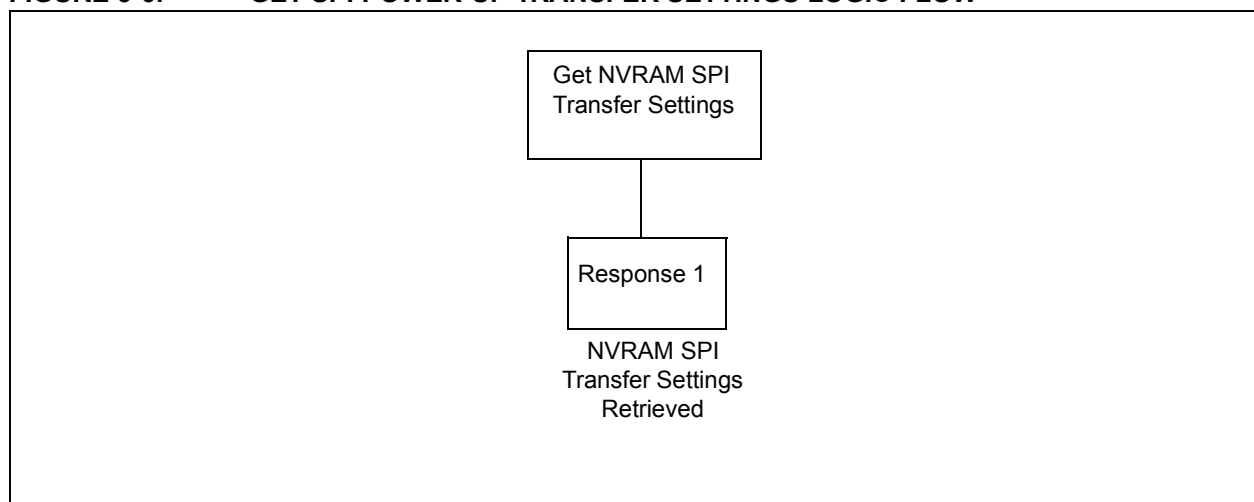
**TABLE 3-18: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | 0x10 – <b>Sub-command Echoed Back</b> for Get SPI Power-up Transfer Settings code  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Bit Rate (Byte 3)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00  |
| 5          | <b>Bit Rate (Byte 2)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x1B  |
| 6          | <b>Bit Rate (Byte 1)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0xB7  |
| 7          | <b>Bit Rate (Byte 0)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00  |
| 8          | <b>Idle Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0   |
| 9          | <b>Idle Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – LSB<br>x x x x x x x CS8  |
| 10         | <b>Active Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0   |
| 11         | <b>Active Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – LSB<br>x x x x x x x CS8  |
| 12         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the CS being asserted and the first byte of data is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x05  |
| 13         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the CS being asserted and the first byte of data is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00 |

**TABLE 3-18: RESPONSE 1 STRUCTURE (CONTINUED)**

| Byte Index | Meaning   |
|------------|---|
| 14         | <b>Last Data Byte to CS (De-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x05  |
| 15         | <b>Last Data Byte to CS (De-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00 |
| 16         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x05                                 |
| 17         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00                                |
| 18         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (low byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- This byte position will have a value of: 0xE2  |
| 19         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (high byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2<br>- This byte position will have a value of: 0x04  |
| 20         | <b>SPI Mode</b> <ul style="list-style-type: none"> <li>• 0x00 – SPI mode 0</li> <li>• 0x01 – SPI mode 1</li> <li>• 0x02 – SPI mode 2</li> <li>• 0x03 – SPI mode 3</li> </ul>  |
| 21 - 63    | <b>Don't care</b>   |

**FIGURE 3-6: GET SPI POWER-UP TRANSFER SETTINGS LOGIC FLOW**



# MCP2210

## 3.1.7 GET POWER-UP CHIP SETTINGS

**TABLE 3-19: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – command code             |
| 1          | 0x20 – <b>Get Power-up Chip Settings</b> – sub-command code |
| 2          | 0x00 – <b>Reserved</b>                                      |
| 3-63       | 0x00 – <b>Reserved</b>                                      |

### 3.1.7.1 Responses

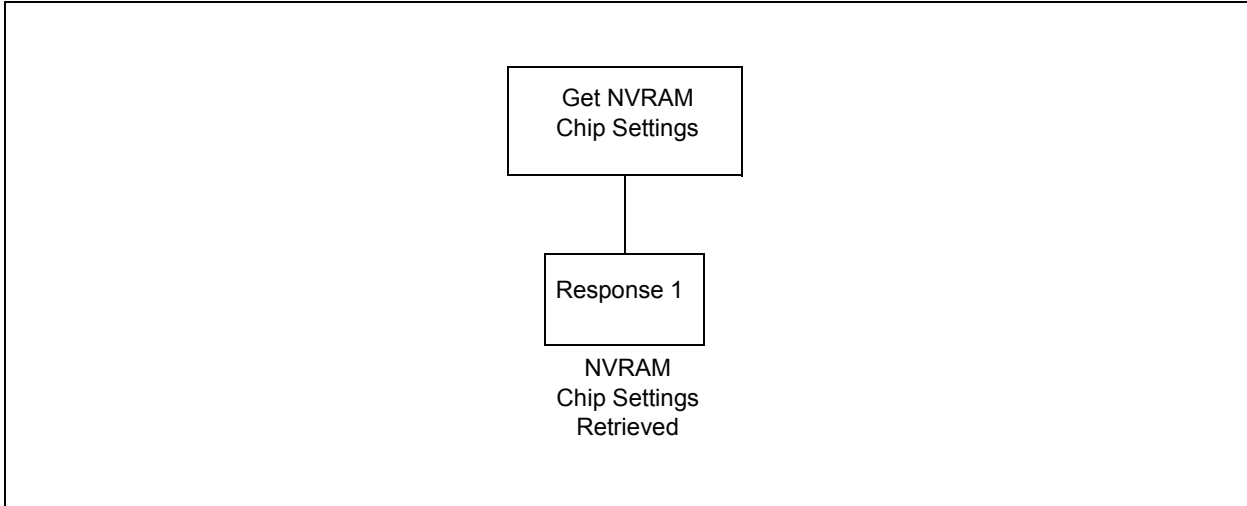
**TABLE 3-20: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | 0x20 – <b>Sub-command Echoed Back</b> for Get Power-up Chip Settings code  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>GP0 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 5          | <b>GP1 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 6          | <b>GP2 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 7          | <b>GP3 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 8          | <b>GP4 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 9          | <b>GP5 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |
| 10         | <b>GP6 Pin Designation</b> <ul style="list-style-type: none"><li>• GPIO = 0x00</li><li>• Chip Selects = 0x01</li><li>• Dedicated Function pin = 0x02</li></ul> |

**TABLE 3-20: RESPONSE 1 STRUCTURE (CONTINUED)**

| Byte Index | Meaning   |
|------------|---|
| 11         | <b>GP7 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>  |
| 12         | <b>GP8 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>  |
| 13         | <b>Default GPIO Output – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      GP7 GP6 GP5 GP4 GP3 GP2 GP1 GP0</li> </ul>   |
| 14         | <b>Default GPIO Output – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8</li> </ul> <p>where x = Don't Care</p>   |
| 15         | <b>Default GPIO Direction – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    –    LSB</li> <li>      GP7DIR GP6DIR GP5DIR GP4DIR GP3DIR GP2DIR GP1DIR GP0DIR</li> </ul>   |
| 16         | <b>Default GPIO Direction – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8DIR</li> </ul>   |
| 17         | <b>Other Chip Settings – Enable/Disable Wake-up, Interrupt Counting, SPI Bus Release Options</b> <ul style="list-style-type: none"> <li>• Bit 7 – Don't Care</li> <li>• Bit 6 – Don't Care</li> <li>• Bit 5 – Don't Care</li> <li>• Bit 4 – Remote Wake-up Enabled/Disabled <ul style="list-style-type: none"> <li>- 0 – Remote Wake-up Disabled</li> <li>- 1 – Remote Wake-up Enabled</li> </ul> </li> <li>• Bit 3 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 2 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 1 – Dedicated Function – Interrupt Pin mode <ul style="list-style-type: none"> <li>- b111 – Reserved</li> <li>- b110 – Reserved</li> <li>- b101 – Reserved</li> <li>- b100 – Count High Pulses</li> <li>- b011 – Count Low Pulses</li> <li>- b010 – Count Rising Edges</li> <li>- b001 – Count Falling Edges</li> <li>- b000 – No Interrupt Counting</li> </ul> </li> <li>• Bit 0 – SPI Bus Release Enable <ul style="list-style-type: none"> <li>- 0 = SPI Bus is Released Between Transfer</li> <li>- 1 = SPI Bus is not released by the MCP2210 between transfers</li> </ul> </li> </ul> |
| 18         | <b>NVRAM Chip Parameters Access Control</b> <ul style="list-style-type: none"> <li>• 0x00 – Chip Settings Not Protected</li> <li>• 0x40 – Chip Settings Protected By Password Access</li> <li>• 0x80 – Chip Settings Permanently Locked</li> </ul>  |
| 19 - 63    | <b>Don't Care</b>   |

**FIGURE 3-7: GET POWER-UP CHIP SETTINGS LOGIC FLOW**



## 3.1.8 GET USB KEY PARAMETERS

**TABLE 3-21: COMMAND STRUCTURE**

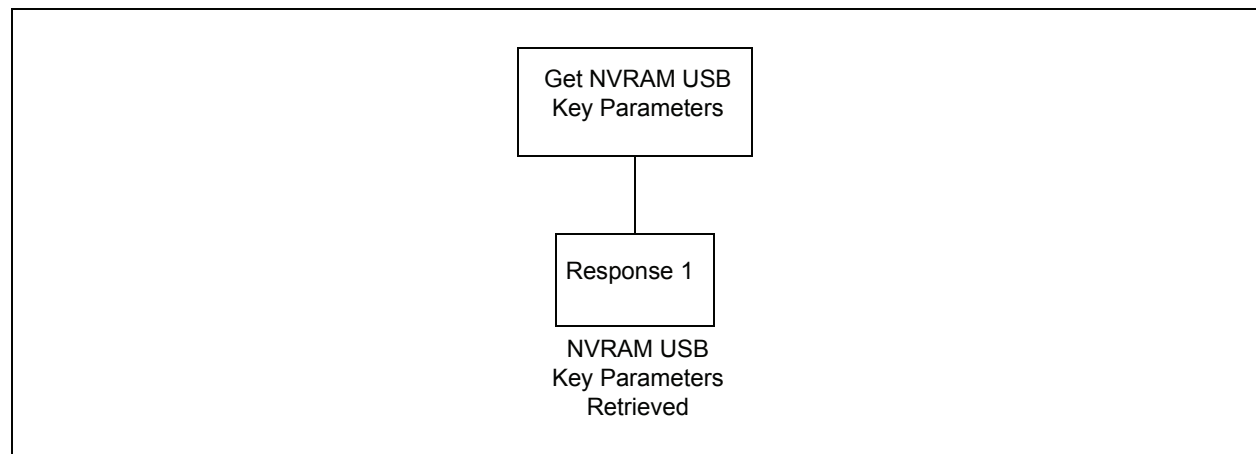
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – command code         |
| 1          | 0x30 – <b>Get USB Key Parameters</b> – sub-command code |
| 2          | 0x00 – <b>Reserved</b>                                  |
| 3-63       | 0x00 – <b>Reserved</b>                                  |

### 3.1.8.1 Responses

**TABLE 3-22: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | 0x30 – <b>Sub-command Echoed Back</b> for Get USB Key Parameters code  |
| 3-11       | <b>Don't care</b>  |
| 12         | <b>VID</b> low byte  |
| 13         | <b>VID</b> high byte   |
| 14         | <b>PID</b> low byte  |
| 15         | <b>PID</b> high byte   |
| 16-28      | <b>Don't care</b>  |
| 29         | <b>Chip Power Option</b> (as per USB specs – Chapter 9) <ul style="list-style-type: none"> <li>• Bit 7 – Host Powered</li> <li>• Bit 6 – Self Powered</li> <li>• Bit 5 – Remote Wake-up Capable</li> <li>• Bit 4 – Don't Care</li> <li>• Bit 3 – Don't Care</li> <li>• Bit 2 – Don't Care</li> <li>• Bit 1 – Don't Care</li> <li>• Bit 0 – Don't Care</li> </ul> |
| 30         | <b>Requested Current Amount from USB Host</b> (quanta of 2 mA)<br><u>Example:</u> For 100 mA this byte index will have a value of 50 (in decimal) or 0x32.   |
| 31-63      | <b>Don't Care</b>  |

**FIGURE 3-8: GET USB KEY PARAMETERS LOGIC FLOW**





# MCP2210

## 3.1.9 GET USB MANUFACTURER NAME

**TABLE 3-23: COMMAND STRUCTURE**

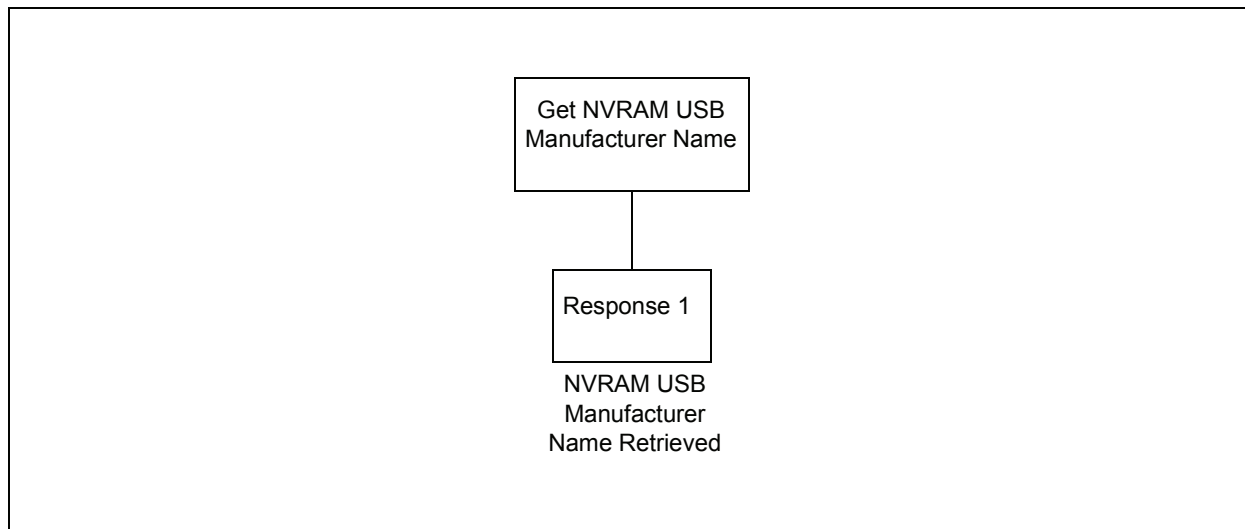
| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – command code            |
| 1          | 0x50 – <b>Get USB Manufacturer Name</b> – sub-command code |
| 2          | 0x00 – <b>Reserved</b>                                     |
| 3-63       | 0x00 – <b>Reserved</b>                                     |

### 3.1.9.1 Responses

**TABLE 3-24: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | 0x50 – <b>Sub-command Echoed Back</b> for Get USB Manufacturer Name code   |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Total USB String Descriptor Length</b> (this is the length of the Manufacturer string multiplied by 2 + 2)<br><u>Example:</u> "Microchip Technology Inc." has 25 Unicode characters.<br>- The retrieved value is: $(25 \times 2) + 2 = 52$ (decimal) = 0x34 |
| 5          | <b>USB String Descriptor ID</b> – always 0x03  |
| 6          | <b>Unicode Character Low Byte</b><br><u>Example:</u> For the "Microchip Technology Inc." Unicode string, there will be the low byte of the Unicode for character "M".<br>- This byte index will have a value of 0x4D   |
| 7          | <b>Unicode Character High Byte</b><br><u>Example:</u> For the "Microchip Technology Inc." Unicode string, there will be the high byte of the Unicode for character "M".<br>- This byte index will have a value of 0x00   |
| 8-63       | Remaining Unicode Characters   |

**FIGURE 3-9: GET USB MANUFACTURER NAME LOGIC FLOW**



## 3.1.10 GET USB PRODUCT NAME

**TABLE 3-25: COMMAND STRUCTURE**

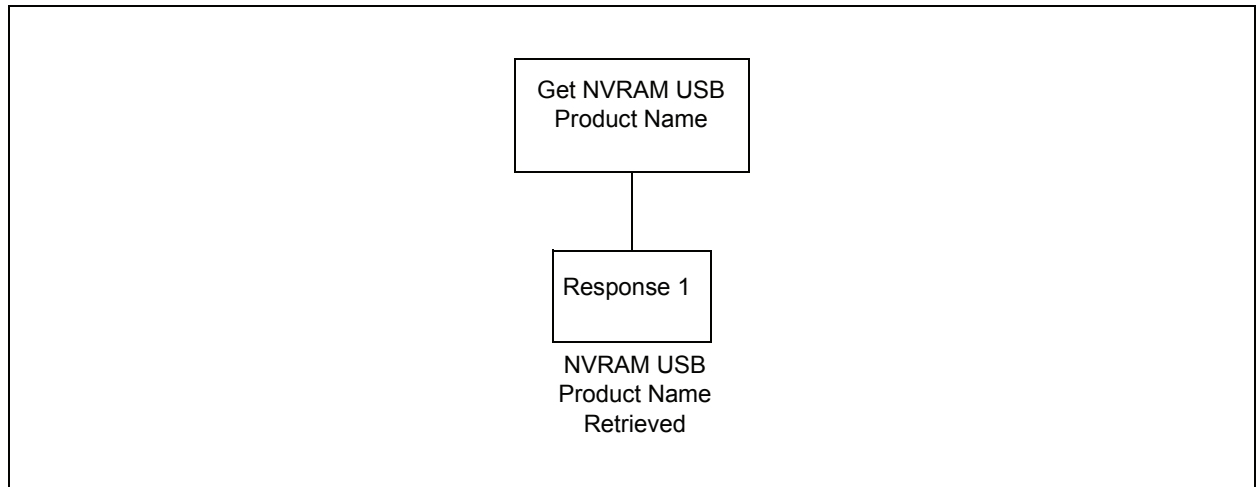
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – command code       |
| 1          | 0x40 – <b>Get USB Product Name</b> – sub-command code |
| 2          | 0x00 – <b>Reserved</b>                                |
| 3-63       | 0x00 – <b>Reserved</b>                                |

### 3.1.10.1 Responses

**TABLE 3-26: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x61 – <b>Get NVRAM Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | 0x40 – <b>Sub-command Echoed Back</b> for Get USB Product Name code  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Total USB String Descriptor Length</b> (this is the length of the Product string multiplied by 2 + 2)<br><u>Example:</u> "MCP2210 USB to SPI Master" has 25 Unicode characters<br>- The retrieved value is: $(25 \times 2) + 2 = 52$ (decimal) = 0x34 |
| 5          | <b>USB String Descriptor ID</b> – always 0x03  |
| 6          | <b>Unicode Character Low byte</b><br><u>Example:</u> For the "MCP2210 USB to SPI Master" Unicode string, there will be the low byte of the Unicode for character "M".<br>- This byte index will have a value of 0x4D                                     |
| 7          | <b>Unicode Character High byte</b><br><u>Example:</u> For the "MCP2210 USB to SPI Master" Unicode string, there will be the high byte of the Unicode for character "M".<br>- This byte index will have a value of 0x00                                   |
| 8-63       | Remaining Unicode Characters   |

**FIGURE 3-10: GET USB PRODUCT NAME LOGIC FLOW**



# MCP2210

---

## 3.1.11 SEND ACCESS PASSWORD

**TABLE 3-27: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – command code |
| 1          | 0x00 – <b>Reserved</b>                            |
| 2          | 0x00 – <b>Reserved</b>                            |
| 3          | 0x00 – <b>Reserved</b>                            |
| 4          | <b>Password Character 0</b>                       |
| 5          | <b>Password Character 1</b>                       |
| 6          | <b>Password Character 2</b>                       |
| 7          | <b>Password Character 3</b>                       |
| 8          | <b>Password Character 4</b>                       |
| 9          | <b>Password Character 5</b>                       |
| 10         | <b>Password Character 6</b>                       |
| 11         | <b>Password Character 7</b>                       |
| 12-63      | 0x00 – <b>Reserved</b>                            |

### 3.1.11.1 Responses

**TABLE 3-28: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – echos back the given command code     |
| 1          | 0x00 – <b>Command Completed Successfully</b> – chip settings not protected |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**TABLE 3-29: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – echos back the given command code |
| 1          | 0xFC – <b>Access Not Allowed</b> – access rejected                     |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**TABLE 3-30: RESPONSE 3 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – echos back the given command code   |
| 1          | 0xFD – <b>Access Not Allowed</b> – Chip conditional access is on, the password does not match and the number of attempts is less than the accepted threshold of 5. |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

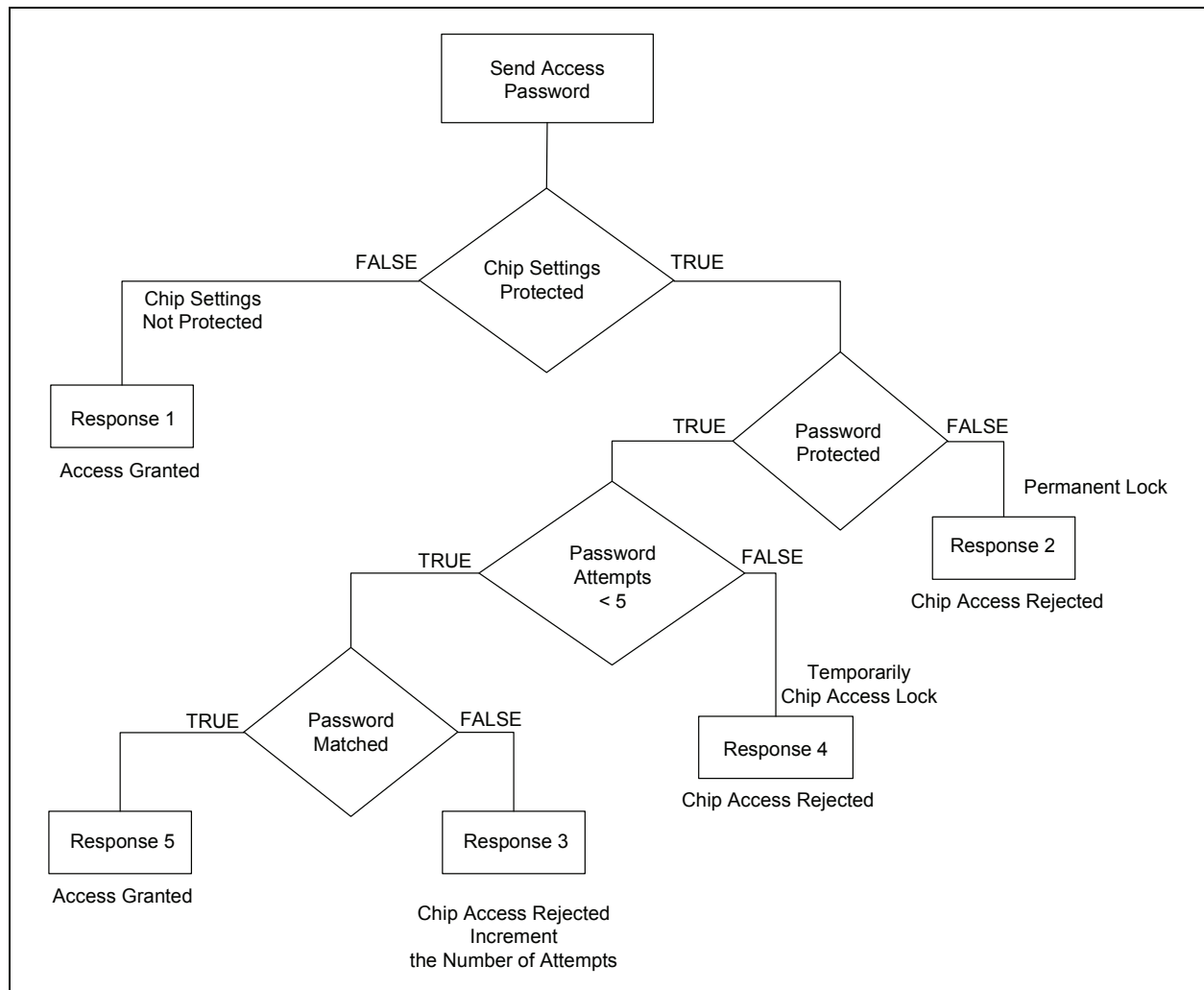
**TABLE 3-31: RESPONSE 4 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – echos back the given command code   |
| 1          | 0xFB – <b>Access Not Allowed</b> – Chip conditional access is on, the password does not match and the number of attempts is above the accepted threshold of 5. The Access Password mechanism is temporarily blocked and no further password access will be accepted until the next power-up. |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**TABLE 3-32: RESPONSE 5 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x70 – <b>SEND ACCESS Password</b> – echos back the given command code  |
| 1          | 0x00 – <b>Command Completed Successfully</b> – Chip conditional access is on, the supplied password is matching the one stored in the chip's NVRAM. |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

**FIGURE 3-11: SEND ACCESS PASSWORD LOGIC FLOW**



# MCP2210

## 3.2 Read/Write RAM Settings

The set of commands/responses described in this section relates to the manipulation of the RAM settings (volatile).

### 3.2.1 GET (VM) SPI TRANSFER SETTINGS

**TABLE 3-33: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x41 – <b>Get (VM) SPI Transfer Settings</b> – command code |
| 1          | 0x00 – <b>Reserved</b>                                      |
| 2          | 0x00 – <b>Reserved</b>                                      |
| 3-63       | 0x00 – <b>Reserved</b>                                      |

#### 3.2.1.1 Responses

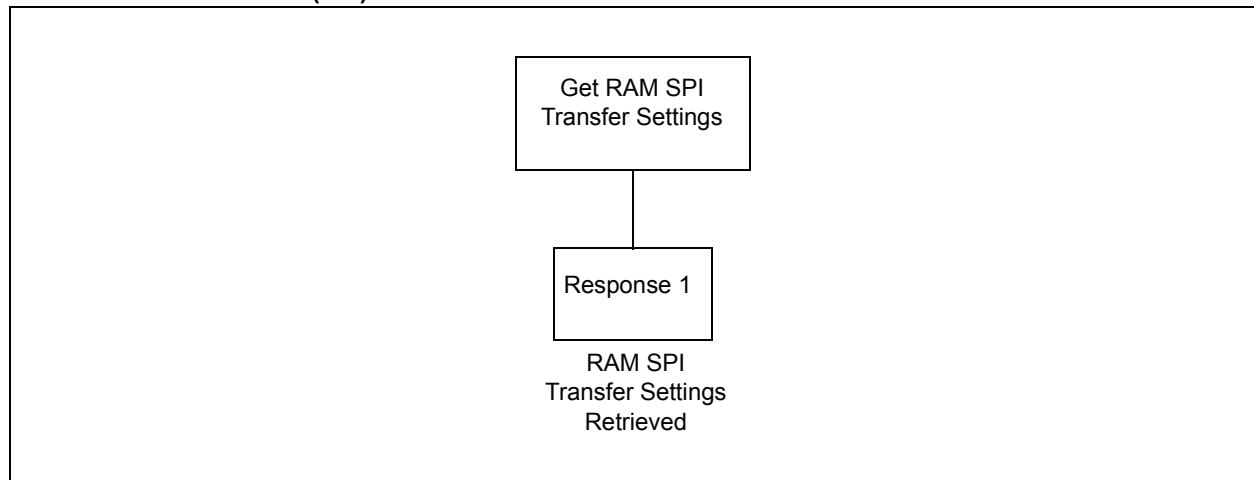
**TABLE 3-34: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x41 – <b>Get SPI Transfer Settings</b> (volatile memory)   |
| 1          | 0x00 – <b>Command Completed Successfully</b>  |
| 2          | <b>Size in Bytes of the SPI Transfer Structure:</b> 17 (in decimal) = 0x11  |
| 3          | <b>Don't Care</b>   |
| 4          | <b>Bit Rate (Byte 3)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00   |
| 5          | <b>Bit Rate (Byte 2)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x1B   |
| 6          | <b>Bit Rate (Byte 1)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0xB7   |
| 7          | <b>Bit Rate (Byte 0)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00   |
| 8          | <b>Idle Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0  |
| 9          | <b>Idle Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – – – – LSB<br>x x x x x x x CS8   |
| 10         | <b>Active Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0  |
| 11         | <b>Active Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – – – – LSB<br>x x x x x x x CS8   |
| 12         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the CS being asserted and the first byte of data, the value will be 0x0005.<br>- This byte position will have a value of: 0x05 |

**TABLE 3-34: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 13         | <p><b>Chip Select to Data Delay</b> (quanta of 100 <math>\mu</math>s) – 16-bit value (high byte)<br/> <u>Example:</u> If we have 500 <math>\mu</math>s delay between the CS being asserted and the first byte of data, the value will be 0x0005.<br/>                     - This byte position will have a value of: 0x00</p>                    |
| 14         | <p><b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 <math>\mu</math>s) – 16-bit value (low byte)<br/> <u>Example:</u> If we have 500 <math>\mu</math>s delay between the last data byte sent and the CS being de-asserted, the value will be 0x0005.<br/>                     - This byte position will have a value of: 0x05</p>  |
| 15         | <p><b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 <math>\mu</math>s) – 16-bit value (high byte)<br/> <u>Example:</u> If we have 500 <math>\mu</math>s delay between the last data byte sent and the CS being de-asserted, the value will be 0x0005.<br/>                     - This byte position will have a value of: 0x00</p> |
| 16         | <p><b>Delay Between Subsequent Data Bytes</b> (quanta of 100 <math>\mu</math>s) – 16-bit value (low byte)<br/> <u>Example:</u> If we have 500 <math>\mu</math>s delay between two consecutive data bytes, the value will be 0x0005.<br/>                     - This byte position will have a value of: 0x05</p>                                 |
| 17         | <p><b>Delay Between Subsequent Data Bytes</b> (quanta of 100 <math>\mu</math>s) – 16-bit value (high byte)<br/> <u>Example:</u> If we have 500 <math>\mu</math>s delay between two consecutive data bytes, the value will be 0x0005.<br/>                     - This byte position will have a value of: 0x00</p>                                |
| 18         | <p><b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (low byte)<br/> <u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br/>                     - This byte position will have a value of: 0xE2</p>  |
| 19         | <p><b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (high byte)<br/> <u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br/>                     - This byte position will have a value of: 0x04</p>   |
| 20         | <p><b>SPI Mode</b></p> <ul style="list-style-type: none"> <li>• 0x00 – SPI mode 0</li> <li>• 0x01 – SPI mode 1</li> <li>• 0x02 – SPI mode 2</li> <li>• 0x03 – SPI mode 3</li> </ul>  |
| 21 - 63    | <b>Don't Care</b>  |

**FIGURE 3-12: GET (VM) SPI TRANSFER SETTINGS LOGIC FLOW**



# MCP2210

## 3.2.2 SET (VM) SPI TRANSFER SETTINGS

**TABLE 3-35: COMMAND 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x40 – <b>Set (VM) SPI Transfer Settings</b> (volatile memory)  |
| 1          | 0x00 – <b>Reserved</b>  |
| 2          | 0x00 – <b>Reserved</b>  |
| 3          | 0x00 – <b>Reserved</b>  |
| 4          | <b>Bit Rate (Byte 3)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00   |
| 5          | <b>Bit Rate (Byte 2)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x1B   |
| 6          | <b>Bit Rate (Byte 1)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0xB7   |
| 7          | <b>Bit Rate (Byte 0)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00   |
| 8          | <b>Idle Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0  |
| 9          | <b>Idle Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – LSB<br>x x x x x x x CS8   |
| 10         | <b>Active Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0  |
| 11         | <b>Active Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – LSB<br>x x x x x x x CS8   |
| 12         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the CS being asserted and the first byte of data, the value will be 0x0005.<br>- This byte position will have a value of: 0x05                           |
| 13         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the CS being asserted and the first byte of data is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00                    |
| 14         | <b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being asserted is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x05     |
| 15         | <b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00 |

TABLE 3-35: COMMAND 1 STRUCTURE (CONTINUED)

| Byte Index | Meaning  |
|------------|--|
| 16         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If a 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x05  |
| 17         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If a 500 $\mu$ s delay between two consecutive data bytes is required, the value will be 0x0005.<br>- This byte position will have a value of: 0x00 |
| 18         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (low byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- This byte position will have a value of: 0xE2                       |
| 19         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (high byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- This byte position will have a value of: 0x04                      |
| 20         | <b>SPI Mode</b> <ul style="list-style-type: none"> <li>• 0x00 – SPI mode 0</li> <li>• 0x01 – SPI mode 1</li> <li>• 0x02 – SPI mode 2</li> <li>• 0x03 – SPI mode 3</li> </ul>   |
| 21-63      | <b>Don't care</b>  |

## 3.2.2.1 Responses

TABLE 3-36: RESPONSE 1 STRUCTURE

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x40 – Echoes back the completed command for Set (VM) SPI Transfer Settings code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | <b>Don't Care</b>  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Bit Rate (Byte 3)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00                        |
| 5          | <b>Bit Rate (Byte 2)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x1B                        |
| 6          | <b>Bit Rate (Byte 1)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0xB7                        |
| 7          | <b>Bit Rate (Byte 0)</b> – 32-bit value (Byte 0, Byte 1, Byte 2, Byte 3)<br><u>Example:</u> Bit rate = 12,000,000 bps = 00B7 1B00<br>- This byte position will have a value of = 0x00                        |
| 8          | <b>Idle Chip Select Value</b> – 16-bit value (low byte):<br><ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>   CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0</li> </ul>         |
| 9          | <b>Idle Chip Select Value</b> – 16-bit value (high byte):<br><ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>   x    x    x    x    x    x    x    CS8</li> </ul> |



# MCP2210

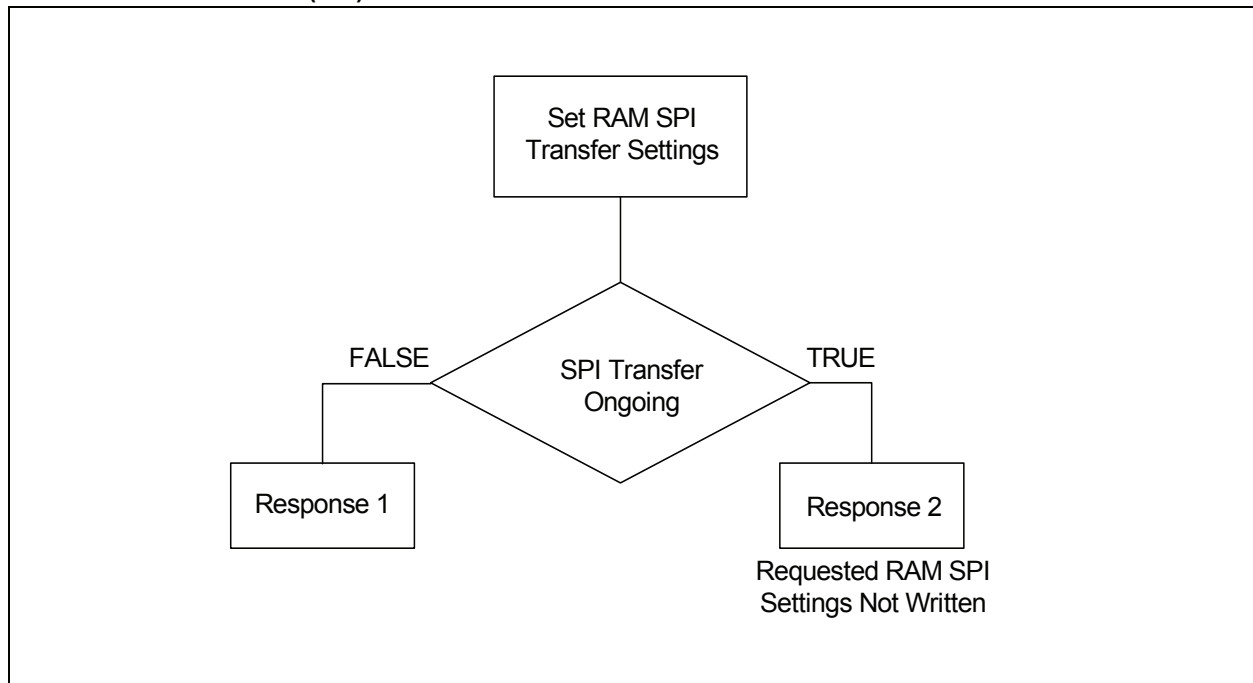
**TABLE 3-36: RESPONSE 1 STRUCTURE (CONTINUED)**

| Byte Index | Meaning   |
|------------|---|
| 10         | <b>Active Chip Select Value</b> – 16-bit value (low byte):<br>• MSB – – – – – LSB<br>CS7 CS6 CS5 CS4 CS3 CS2 CS1 CS0  |
| 11         | <b>Active Chip Select Value</b> – 16-bit value (high byte):<br>• MSB – – – – – LSB<br>x x x x x x x CS8   |
| 12         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the CS being asserted and the first byte of data, the value will be 0x0005.<br>- This byte position will have a value of: 0x05                     |
| 13         | <b>Chip Select to Data Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the CS being asserted and the first byte of data, the value will be 0x0005.<br>- This byte position will have a value of: 0x00                    |
| 14         | <b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted, the value will be 0x0005.<br>- This byte position will have a value of: 0x05  |
| 15         | <b>Last Data Byte to CS (de-asserted) Delay</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between the last data byte sent and the CS being de-asserted, the value will be 0x0005.<br>- This byte position will have a value of: 0x00 |
| 16         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (low byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between two consecutive data bytes, the value will be 0x0005.<br>- This byte position will have a value of: 0x05                                 |
| 17         | <b>Delay Between Subsequent Data Bytes</b> (quanta of 100 $\mu$ s) – 16-bit value (high byte)<br><u>Example:</u> If we have 500 $\mu$ s delay between two consecutive data bytes, the value will be 0x0005.<br>- This byte position will have a value of: 0x00                                |
| 18         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (low byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- This byte position will have a value of: 0xE2  |
| 19         | <b>Bytes to Transfer per SPI Transaction</b> – 16-bit value (high byte)<br><u>Example:</u> If an SPI transaction of 1250 bytes long is required, the corresponding hex value will be 0x04E2.<br>- This byte position will have a value of: 0x04   |
| 20         | <b>SPI Mode</b><br>• 0x00 – SPI mode 0<br>• 0x01 – SPI mode 1<br>• 0x02 – SPI mode 2<br>• 0x03 – SPI mode 3   |
| 21 - 63    | <b>Don't Care</b>   |

**TABLE 3-37: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x40 – <b>Set (VM) SPI Transfer Settings</b> – echos back the given command code |
| 1          | 0xF8 – USB transfer in progress – <b>Settings not written</b>                    |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**FIGURE 3-13: SET (VM) SPI TRANSFER SETTINGS LOGIC FLOW**



### 3.2.3 GET (VM) CURRENT CHIP SETTINGS

**TABLE 3-38: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x20 – <b>Get (VM) GPIO Current Chip Settings</b> |
| 1          | 0x00 – <b>Reserved</b>                            |
| 2          | 0x00 – <b>Reserved</b>                            |
| 3-63       | 0x00 – <b>Reserved</b>                            |

# MCP2210

## 3.2.3.1 Responses

**TABLE 3-39: RESPONSE 1 STRUCTURE**

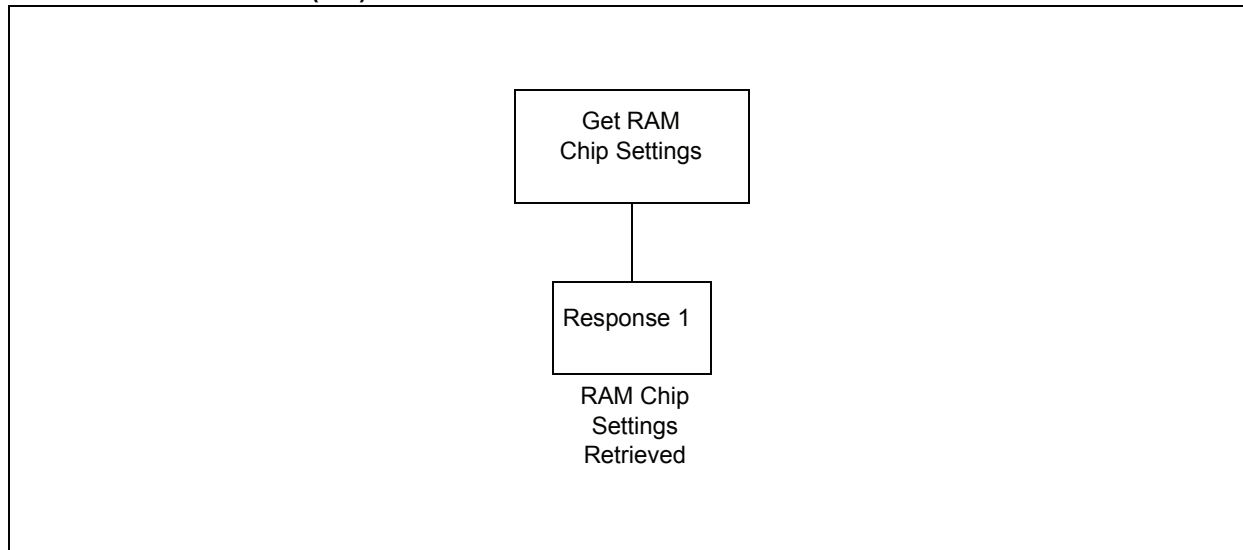
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x20 – <b>Get (VM) GPIO Current Chip Settings</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>  |
| 2          | <b>Don't Care</b>   |
| 3          | <b>Don't Care</b>   |
| 4          | <b>GP0 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 5          | <b>GP1 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 6          | <b>GP2 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 7          | <b>GP3 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 8          | <b>GP4 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 9          | <b>GP5 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 10         | <b>GP6 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 11         | <b>GP7 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 12         | <b>GP8 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>                                |
| 13         | <b>Default GPIO Output</b> – 16-bit value (low byte): <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      GP7 GP6 GP5 GP4 GP3 GP2 GP1 GP0</li> </ul> |

**TABLE 3-39: RESPONSE 1 STRUCTURE (CONTINUED)**

| Byte Index | Meaning   |
|------------|---|
| 14         | <p><b>Default GPIO Output</b> – 16-bit value (high byte):</p> <ul style="list-style-type: none"> <li>MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8</li> </ul> <p>where x = <b>Don't Care</b></p>   |
| 15         | <p><b>Default GPIO Direction</b> – 16-bit value (low byte):</p> <ul style="list-style-type: none"> <li>MSB            –            –            –            –            –            LSB</li> <li>GP7DIR GP6DIR GP5DIR GP4DIR GP3DIR GP2DIR GP1DIR GP0DIR</li> </ul>  |
| 16         | <p><b>Default GPIO Direction</b> – 16-bit value (high byte):</p> <ul style="list-style-type: none"> <li>MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8DIR</li> </ul>  |
| 17         | <p><b>Other Chip Settings – Enable/Disable Wake-up, Interrupt Counting, SPI Bus Release Options</b></p> <ul style="list-style-type: none"> <li>• Bit 7 – Don't Care</li> <li>• Bit 6 – Don't Care</li> <li>• Bit 5 – Don't Care</li> <li>• Bit 4 – Remote Wake-up Enabled/Disabled <ul style="list-style-type: none"> <li>- 0 – Remote Wake-up Disabled</li> <li>- 1 – Remote Wake-up Enabled</li> </ul> </li> <li>• Bit 3 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 2 – Dedicated Function – Interrupt Pin mode</li> <li>• Bit 1 – Dedicated Function – Interrupt Pin mode <ul style="list-style-type: none"> <li>- b111 – Reserved</li> <li>- b110 – Reserved</li> <li>- b101 – Reserved</li> <li>- b100 – Count High Pulses</li> <li>- b011 – Count Low Pulses</li> <li>- b100 – Count High Pulses</li> <li>- b011 – Count Low Pulses</li> <li>- b010 – Count Rising Edges</li> <li>- b001 – Count Falling Edges</li> <li>- b000 – No Interrupt Counting</li> </ul> </li> <li>• Bit 0 – SPI Bus Release Enable <ul style="list-style-type: none"> <li>- 0 = SPI Bus is Released between transfer</li> <li>- 1 = SPI Bus is Not Released by the MCP2210 between transfers</li> </ul> </li> </ul> |
| 18         | <p><b>NVRAM Chip Parameters Access Control</b></p> <ul style="list-style-type: none"> <li>• 0x00 – Chip settings not protected</li> <li>• 0x40 – Chip settings protected by password access</li> <li>• 0x80 – Chip settings permanently locked</li> </ul>   |
| 19-63      | <b>Don't Care</b>   |

# MCP2210

**FIGURE 3-14: GET (VM) CURRENT CHIP SETTINGS LOGIC FLOW**



## 3.2.4 SET (VM) CURRENT CHIP SETTINGS

**TABLE 3-40: COMMAND STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x21 – <b>Set (VM) Current Chip Settings</b>   |
| 1          | 0x00 – <b>Reserved</b>   |
| 2          | 0x00 – <b>Reserved</b>   |
| 3          | 0x00 – <b>Reserved</b>   |
| 4          | <b>GP0 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 5          | <b>GP1 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 6          | <b>GP2 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 7          | <b>GP3 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |
| 8          | <b>GP4 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul> |

**TABLE 3-40: COMMAND STRUCTURE (CONTINUED)**

| Byte Index | Meaning  |
|------------|--|
| 9          | <b>GP5 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 10         | <b>GP6 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 11         | <b>GP7 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 12         | <b>GP8 Pin Designation</b> <ul style="list-style-type: none"> <li>• GPIO = 0x00</li> <li>• Chip Selects = 0x01</li> <li>• Dedicated Function pin = 0x02</li> </ul>   |
| 13         | <b>Default GPIO Output – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      GP7 GP6 GP5 GP4 GP3 GP2 GP1 GP0</li> </ul>  |
| 14         | <b>Default GPIO Output – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8</li> </ul> <p>where x = <b>Don't Care</b></p> |
| 15         | <b>Default GPIO Direction – 16-bit value (low byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    –    LSB</li> <li>      GP7DIR GP6DIR GP5DIR GP4DIR GP3DIR GP2DIR GP1DIR GP0DIR</li> </ul>            |
| 16         | <b>Default GPIO Direction – 16-bit value (high byte):</b> <ul style="list-style-type: none"> <li>• MSB    –    –    –    –    –    –    LSB</li> <li>      x    x    x    x    x    x    x    GP8DIR</li> </ul>                              |

# MCP2210

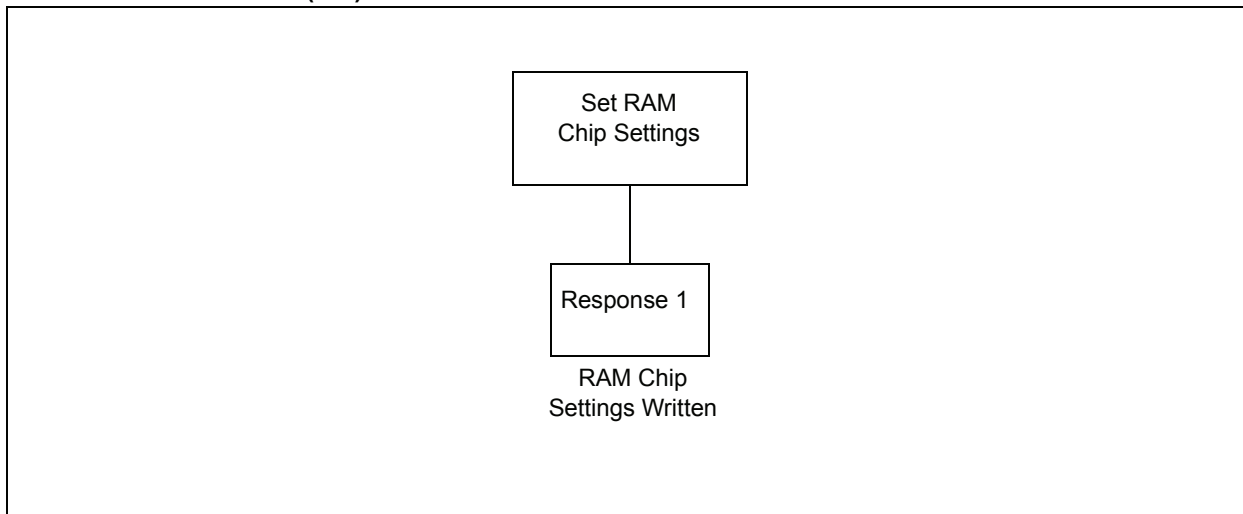
TABLE 3-40: COMMAND STRUCTURE (CONTINUED)

| Byte Index | Meaning   |
|------------|---|
| 17         | <p><b>Other Chip Settings – Enable/Disable Wake-up, Interrupt Counting, SPI Bus Release Options</b></p> <ul style="list-style-type: none"><li>• Bit 7 – Don't Care</li><li>• Bit 6 – Don't Care</li><li>• Bit 5 – Don't Care</li><li>• Bit 4 – Remote Wake-up Enabled/Disabled<ul style="list-style-type: none"><li>- 0 – Remote Wake-up Disabled</li><li>- 1 – Remote Wake-up Enabled</li></ul></li><li>• Bit 3 – Dedicated Function – Interrupt Pin mode</li><li>• Bit 2 – Dedicated Function – Interrupt Pin mode</li><li>• Bit 1 – Dedicated Function – Interrupt Pin mode<ul style="list-style-type: none"><li>- b111 – Reserved</li><li>- b110 – Reserved</li><li>- b101 – Reserved</li><li>- b100 – Count High Pulses</li><li>- b011 – Count Low Pulses</li><li>- b100 – Count High Pulses</li><li>- b011 – Count Low Pulses</li><li>- b010 – Count Rising Edges</li><li>- b001 – Count Falling Edges</li><li>- b000 – No Interrupt Counting</li></ul></li><li>• Bit 0 – SPI Bus Release Enable<ul style="list-style-type: none"><li>- 0 = SPI Bus is Released between transfer</li><li>- 1 = SPI Bus is Not Released by the MCP2210 between transfers</li></ul></li></ul> |
| 18-63      | <b>Reserved</b> (fill in with 0x00)   |

**TABLE 3-41: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x21 – <b>Set (VM) Current Chip Settings</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b>                                     |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**FIGURE 3-15: SET (VM) CURRENT CHIP SETTINGS LOGIC FLOW**





# MCP2210

## 3.2.5 GET (VM) GPIO CURRENT PIN DIRECTION

**TABLE 3-42: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x33 – <b>Get (VM) GPIO Current Pin Direction</b> |
| 1          | 0x00 – <b>Reserved</b>                            |
| 2          | 0x00 – <b>Reserved</b>                            |
| 3-63       | 0x00 – <b>Reserved</b>                            |

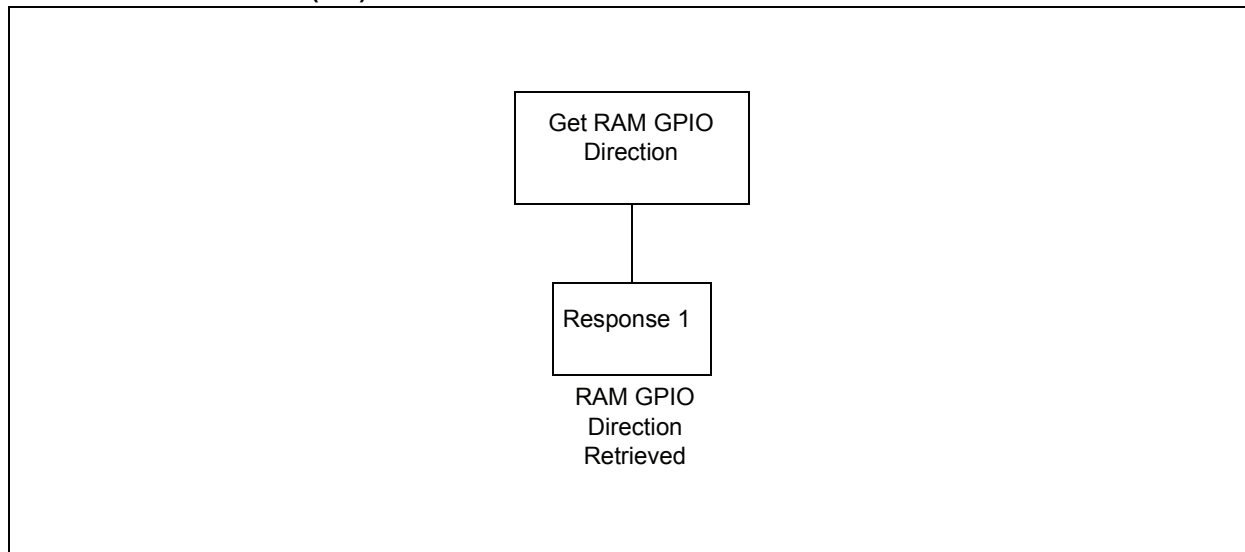
### 3.2.5.1 Responses

**TABLE 3-43: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x33 – <b>Get (VM) GPIO Current Pin Direction</b> – echos back the given command code  |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | <b>Don't Care</b>  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>GPIO Direction</b> – 16-bit value (low byte):<br><ul style="list-style-type: none"> <li>• MSB      –      –      –      –      –      –      –      LSB</li> <li>    GP7DIR GP6DIR GP5DIR GP4DIR GP3DIR GP2DIR GP1DIR GP0DIR</li> </ul> |
| 5          | <b>GPIO Direction</b> – 16-bit value (high byte):<br><ul style="list-style-type: none"> <li>• MSB      –      –      –      –      –      –      LSB</li> <li>    x      x      x      x      x      x      x      GP8DIR</li> </ul>       |
| 6-63       | <b>Don't Care</b>  |

**Note 1:** This command will only have an effect on those GPs previously configured as GPIOs.

**FIGURE 3-16: GET (VM) GPIO CURRENT PIN DIRECTION LOGIC FLOW**





# MCP2210

## 3.2.7 GET GPIO CURRENT PIN VALUE

**TABLE 3-46: COMMAND STRUCTURE**

| Byte Index | Meaning                                       |
|------------|---|
| 0          | 0x31 – <b>Get (VM) GPIO Current Pin Value</b> |
| 1          | 0x00 – <b>Reserved</b>                        |
| 2          | 0x00 – <b>Reserved</b>                        |
| 3-63       | 0x00 – <b>Reserved</b>                        |

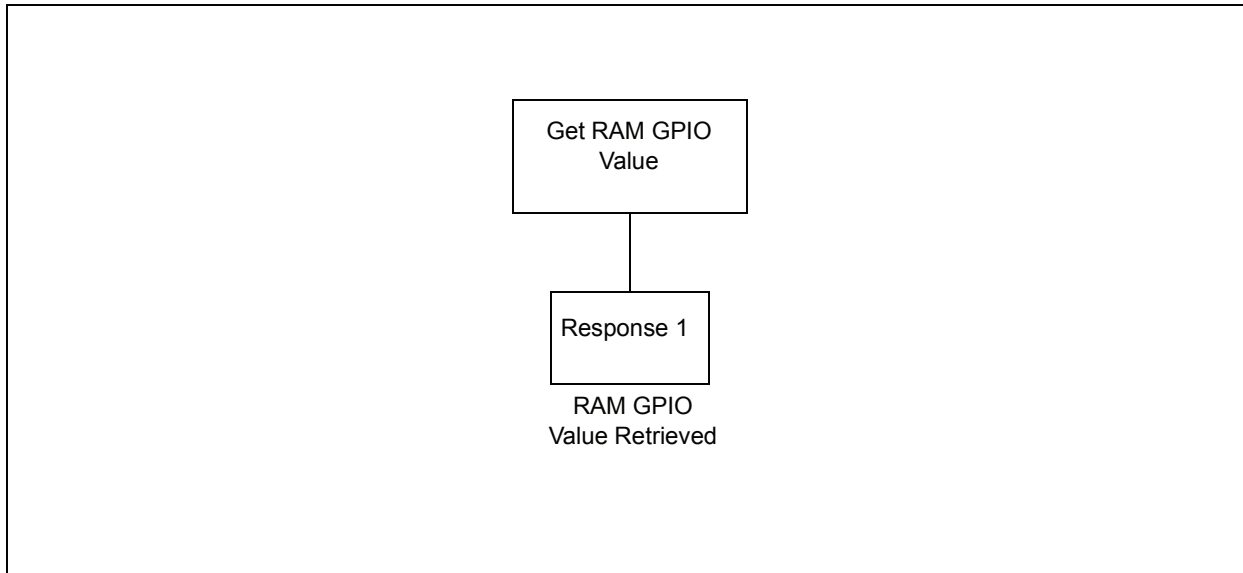
### 3.2.7.1 Responses

**TABLE 3-47: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x31 – <b>Get (VM) GPIO Current Pin Value</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>  |
| 2          | <b>Don't Care</b>   |
| 3          | <b>Don't Care</b>   |
| 4          | <b>GPIO Pin Value</b> – 16-bit value (low byte):<br><ul style="list-style-type: none"> <li>MSB      –      –      –      –      –      –      LSB</li> <li>GP7VAL GP6VAL GP5VAL GP4VAL GP3VAL GP2VAL GP1VAL GP0VAL</li> </ul> |
| 5          | <b>GPIO Pin Value</b> – 16-bit value (high byte):<br><ul style="list-style-type: none"> <li>MSB      –      –      –      –      –      –      LSB</li> <li>      x    x    x    x    x    x    x    GP8VAL</li> </ul>        |
| 6-63       | <b>Don't Care</b>   |

**Note 1:** This command will only have an effect on those GPs previously assigned to a GPIO functionality.

**FIGURE 3-18: GET GPIO CURRENT PIN VALUE LOGIC FLOW**



## 3.2.8 SET GPIO CURRENT PIN VALUE

**TABLE 3-48: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x30 – <b>Set (VM) GPIO Current Pin Value</b>   |
| 1          | 0x00 – <b>Reserved</b>  |
| 2          | 0x00 – <b>Reserved</b>  |
| 3          | 0x00 – <b>Reserved</b>  |
| 4          | <b>GPIO Pin Value</b> – 16-bit value (low byte):<br>• MSB        –        –        –        –        –        –        LSB<br>GP7VAL GP6VAL GP5VAL GP4VAL GP3VAL GP2VAL GP1VAL GP0VAL |
| 5          | <b>GPIO Pin Value</b> – 16-bit value (high byte):<br>• MSB        –        –        –        –        –        LSB<br>x    x    x    x    x    x    x    GP8VAL                       |
| 6-63       | 0x00 – <b>Reserved</b>  |

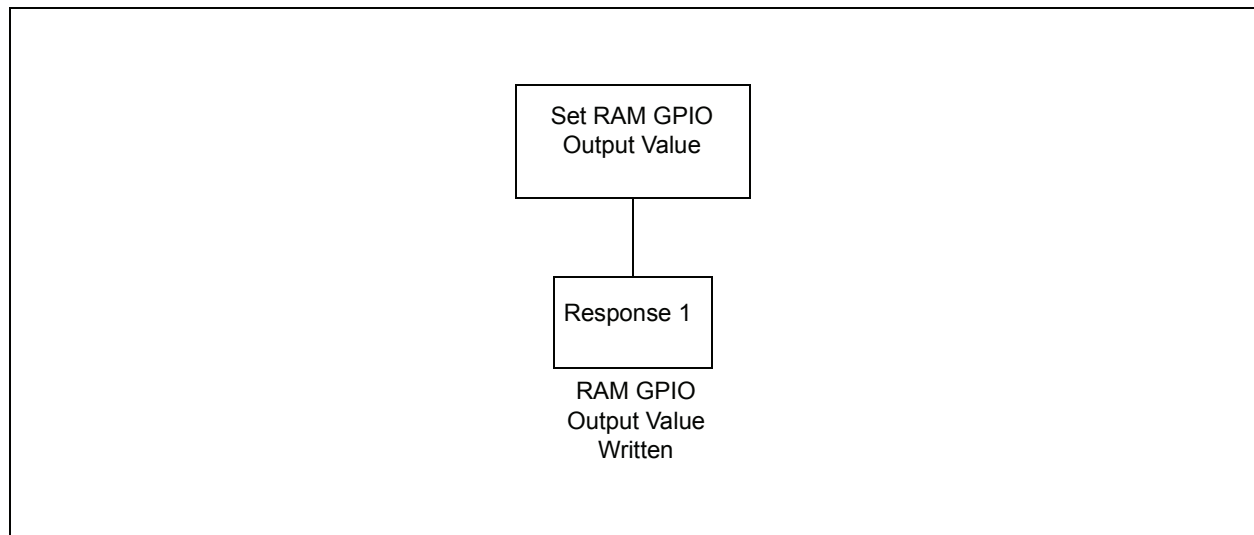
**Note 1:** The GPIO pin value will have an effect only on those GPs previously configured as GPIOs.

### 3.2.8.1 Responses

**TABLE 3-49: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x30 – <b>Set (VM) GPIO Current Pin Value</b> – echos back the given command code  |
| 1          | 0x00 - <b>Command Completed Successfully</b>   |
| 2          | <b>Don't Care</b>  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Read Back Actual GPIO Pin Value</b> – 16-bit value (low byte):<br>• MSB        –        –        –        –        –        –        LSB<br>GP7VAL GP6VAL GP5VAL GP4VAL GP3VAL GP2VAL GP1VAL GP0VAL |
| 5          | <b>Read Back Actual GPIO Pin Value</b> – 16-bit value (high byte):<br>• MSB        –        –        –        –        –        LSB<br>x    x    x    x    x    x    x    GP8VAL                       |
| 6-63       | <b>Don't Care</b>  |

**FIGURE 3-19: SET GPIO CURRENT PIN VALUE LOGIC FLOW**



# MCP2210

## 3.3 Read/Write EEPROM Memory

This set of commands/responses described in this section relates to the manipulation of the EEPROM memory.

### 3.3.1 READ EEPROM MEMORY

**TABLE 3-50: COMMAND STRUCTURE**

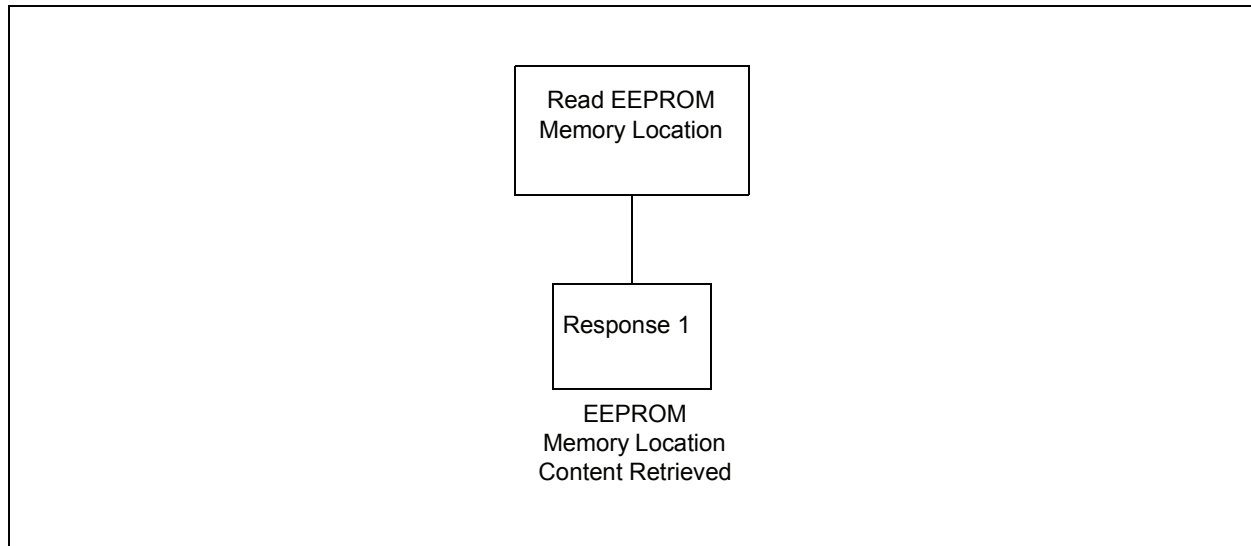
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x50 – <b>READ EEPROM Memory</b> – command code |
| 1          | <b>EEPROM Memory Address</b> to be read         |
| 2          | 0x00 – <b>Reserved</b>                          |
| 3-63       | 0x00 – <b>Reserved</b>                          |

#### 3.3.1.1 Responses

**TABLE 3-51: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x50 – <b>READ EEPROM Memory</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b>                         |
| 2          | <b>EEPROM Memory Address</b>   |
| 3          | <b>EEPROM Memory content at the requested address</b>                |
| 4-63       | <b>Don't Care</b>  |

**FIGURE 3-20: READ EEPROM MEMORY LOGIC FLOW**



## 3.3.2 WRITE EEPROM MEMORY

**TABLE 3-52: COMMAND STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x51 – <b>WRITE EEPROM Memory</b> – command code |
| 1          | <b>EEPROM Memory Address</b> to be written       |
| 2          | The value to be written to at the given address  |
| 3-63       | 0x00 – <b>Reserved</b>                           |

### 3.3.2.1 Responses

**TABLE 3-53: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x51 – <b>WRITE EEPROM Memory</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b>                          |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

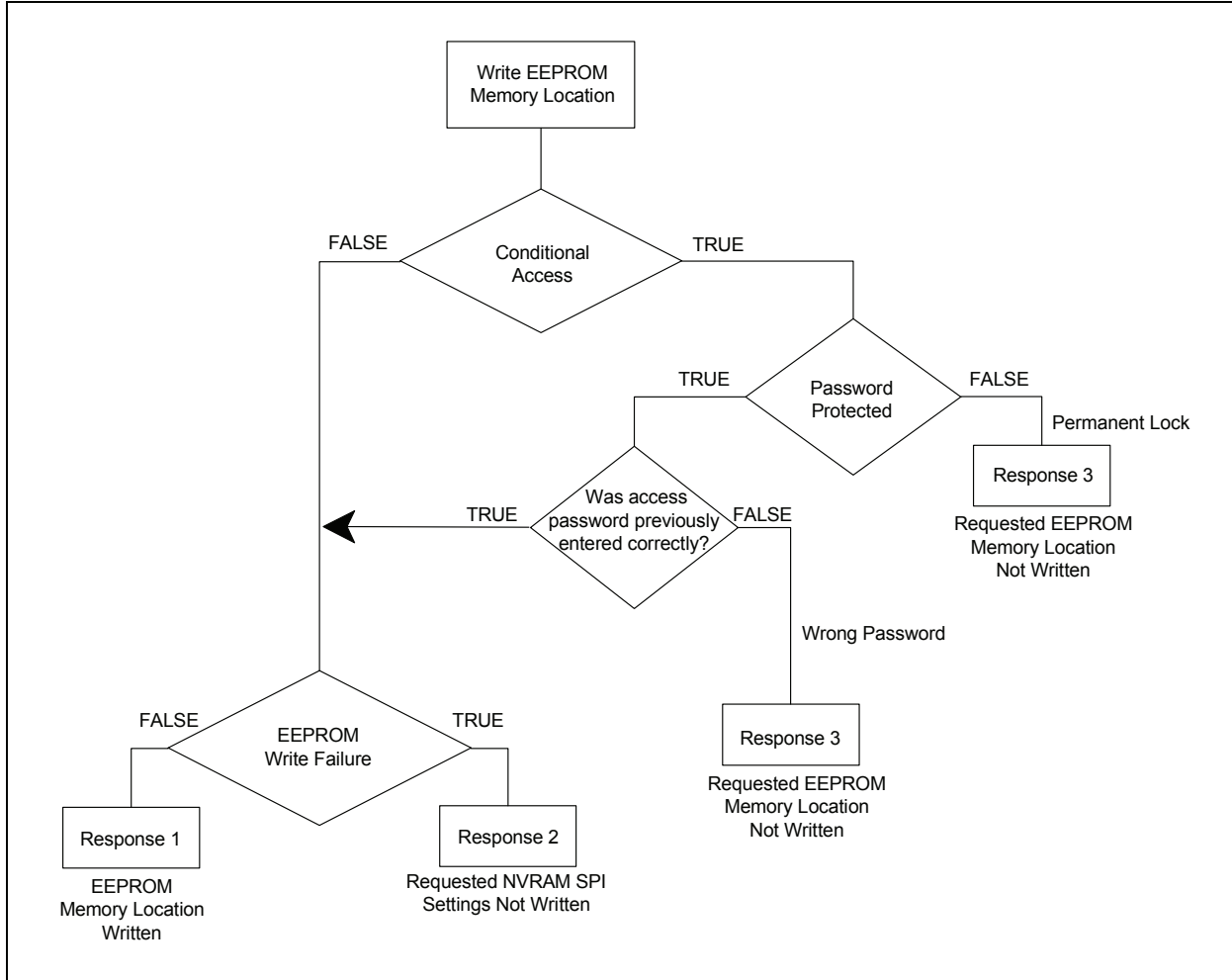
**TABLE 3-54: RESPONSE 2 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x51 – <b>WRITE EEPROM Memory</b> – echos back the given command code |
| 1          | 0xFA – <b>EEPROM Write Failure</b>                                    |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

**TABLE 3-55: RESPONSE 3 STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x51 – <b>WRITE EEPROM Memory</b> – echos back the given command code |
| 1          | 0xFB – <b>EEPROM is password protected or permanently locked</b>      |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

FIGURE 3-21: WRITE EEPROM MEMORY LOGIC FLOW



## 3.4 External Interrupt Pin (GP6) Event Status

The External Interrupt pin event status command is used by the USB host to query the external interrupt events recorded by the MCP2210. In order to have the MCP2210 record the number of external interrupt events, GP6 must be configured to have its dedicated function active.

### 3.4.1 GET (VM) THE CURRENT NUMBER OF EVENTS FROM THE INTERRUPT PIN

**TABLE 3-56: COMMAND STRUCTURE**

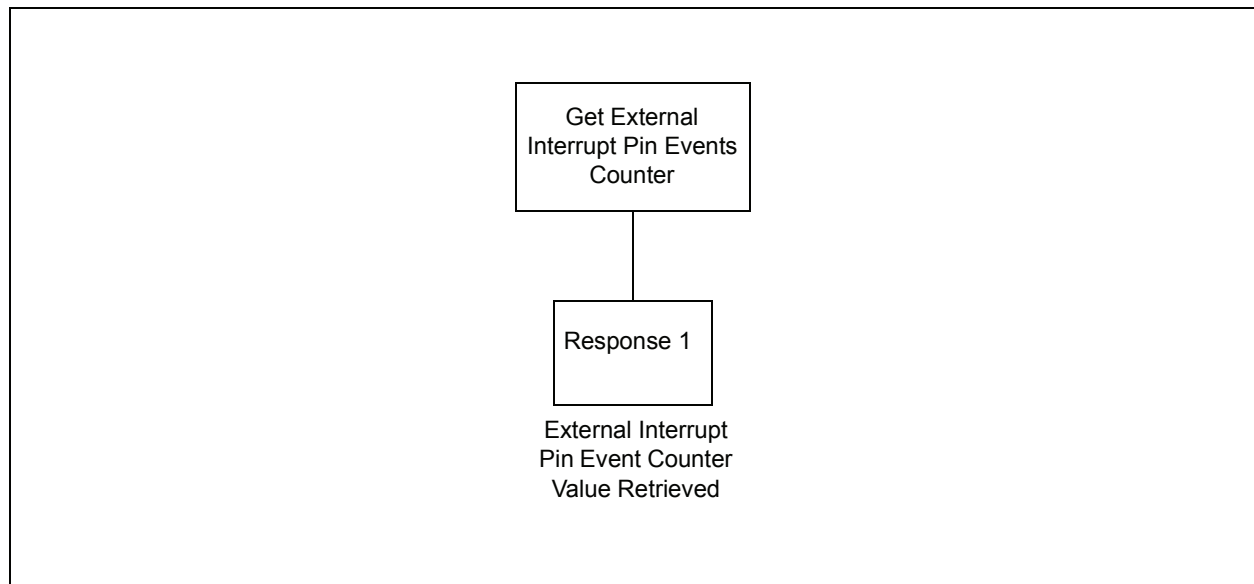
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x12 – <b>Get (VM) the Current Number of Events From the Interrupt Pin</b>  |
| 1          | <b>Reset or Not the Event Counter</b> <ul style="list-style-type: none"> <li>• 0x00 – reads, then resets the event counter</li> <li>• Any other value – the event counter is read, however, the counter is not reset</li> </ul> |
| 2-63       | 0x00 - <b>Reserved</b>  |

#### 3.4.1.1 Responses

**TABLE 3-57: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x12 – <b>Get (VM) the Current Number of Events from the Interrupt Pin</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | <b>Don't Care</b>  |
| 3          | <b>Don't Care</b>  |
| 4          | <b>Interrupt Event Counter</b> – 16-bit value (low byte)   |
| 5          | <b>Interrupt Event Counter</b> – 16-bit value (high byte)  |
| 63-63      | <b>Don't Care</b>  |

**FIGURE 3-22: GET (VM) THE CURRENT NUMBER OF EVENTS FROM THE INTERRUPT PIN LOGIC FLOW**





# MCP2210

---

## 3.5 SPI Data Transfer

The set of commands/responses described in this section relates to the SPI data transfer functionality.

### 3.5.1 TRANSFER SPI DATA

**TABLE 3-58: COMMAND STRUCTURE**

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x42 – <b>Transfer SPI Data</b> – command code                                  |
| 1          | The number of bytes to be transferred in this packet (from 0 to 60 inclusively) |
| 2          | 0x00 – <b>Reserved</b>  |
| 3          | 0x00 – <b>Reserved</b>  |
| 4-63       | The SPI Data to be sent on the data transfer                                    |

#### 3.5.1.1 Responses

**TABLE 3-59: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x42 – <b>Transfer SPI Data</b> – echos back the given command code                                  |
| 1          | 0xF7 – <b>SPI Data Not Accepted</b> – SPI bus not available (the external owner has control over it) |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

**TABLE 3-60: RESPONSES 2 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x42 – <b>Transfer SPI Data</b> – echos back the given command code  |
| 1          | 0x00 – <b>SPI Data accepted – Command Completed Successfully</b> – SPI data accepted   |
| 2          | How many SPI received data bytes the chip is sending back to the host  |
| 3          | <b>SPI Transfer Engine Status</b> <ul style="list-style-type: none"><li>• 0x20 – SPI transfer started – no data to receive</li></ul> |
| 4-63       | <b>SPI Received Data Bytes.</b> The number of data bytes is specified at byte index 2  |

**TABLE 3-61: RESPONSE 3 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x42 – <b>Transfer SPI Data</b> – echos back the given command code                                    |
| 1          | 0xF8 – <b>SPI Data Not Accepted</b> – SPI transfer in progress – cannot accept any data for the moment |
| 2          | <b>Don't Care</b>  |
| 3-63       | <b>Don't Care</b>  |

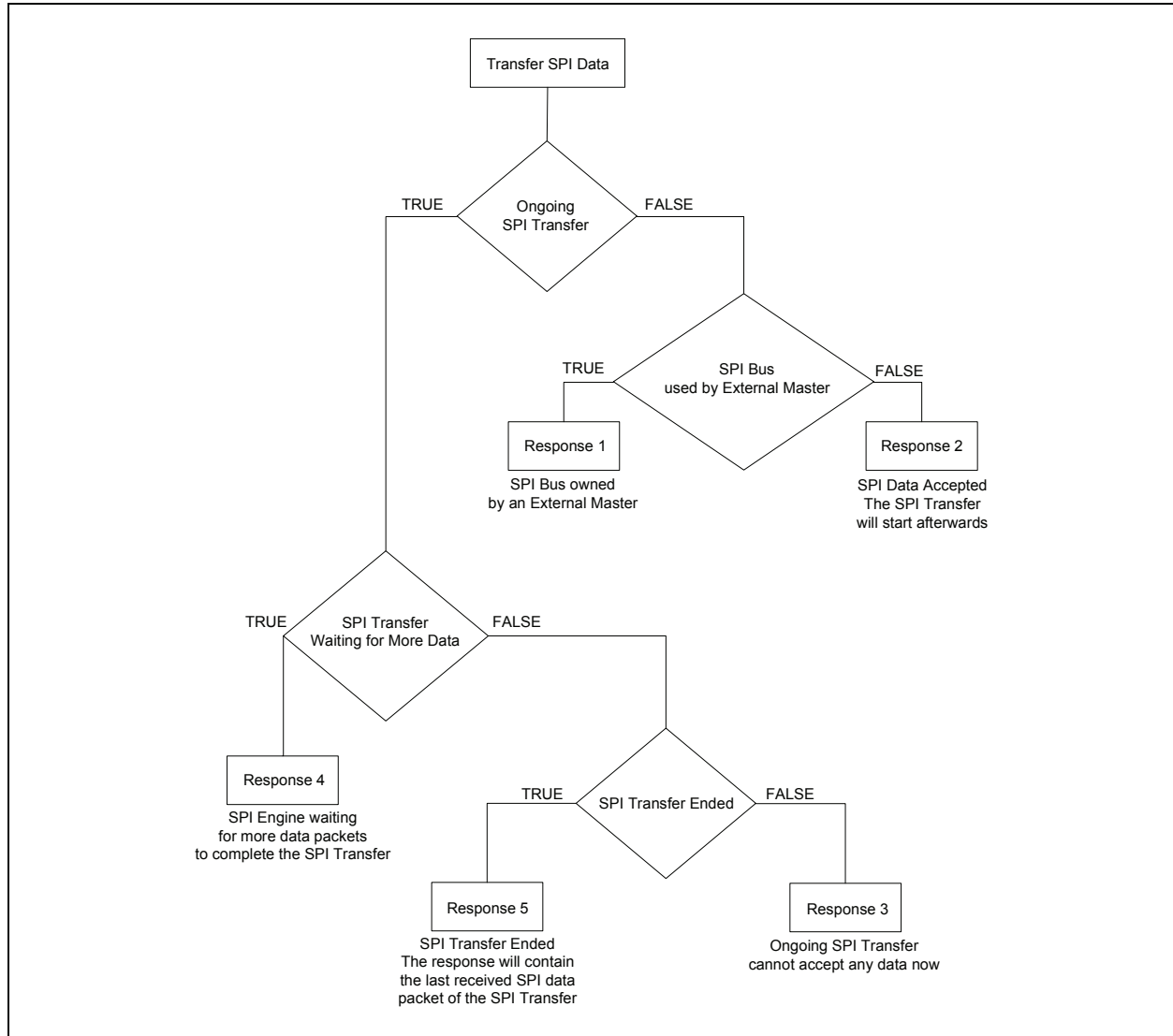
**TABLE 3-62: RESPONSE 4 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x42 – <b>Transfer SPI Data</b> – echos back the given command code                          |
| 1          | 0x00 – <b>SPI Data accepted – Command Completed Successfully</b> – SPI data accepted         |
| 2          | How many SPI received data bytes the chip is sending back to the host                        |
| 3          | 0x30 – <b>SPI Transfer Engine Status:</b> SPI transfer not finished; received data available |
| 4-63       | <b>SPI received data bytes.</b> The number of data bytes is specified at byte index 2        |

**TABLE 3-63: RESPONSE 5 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x42 – <b>Transfer SPI Data</b> – echos back the given command code                    |
| 1          | 0x00 – <b>SPI Data accepted – Command Completed Successfully</b> – SPI data accepted   |
| 2          | How many SPI received data bytes the chip is sending back to the host                  |
| 3          | 0x10 – <b>SPI Transfer Engine Status:</b> SPI transfer finished – no more data to send |
| 4-63       | <b>SPI received data bytes.</b> The number of data bytes is specified at byte index 2  |

**FIGURE 3-23: TRANSFER SPI DATA LOGIC FLOW**



# MCP2210

## 3.5.2 CANCEL THE CURRENT SPI TRANSFER

**TABLE 3-64: COMMAND STRUCTURE**

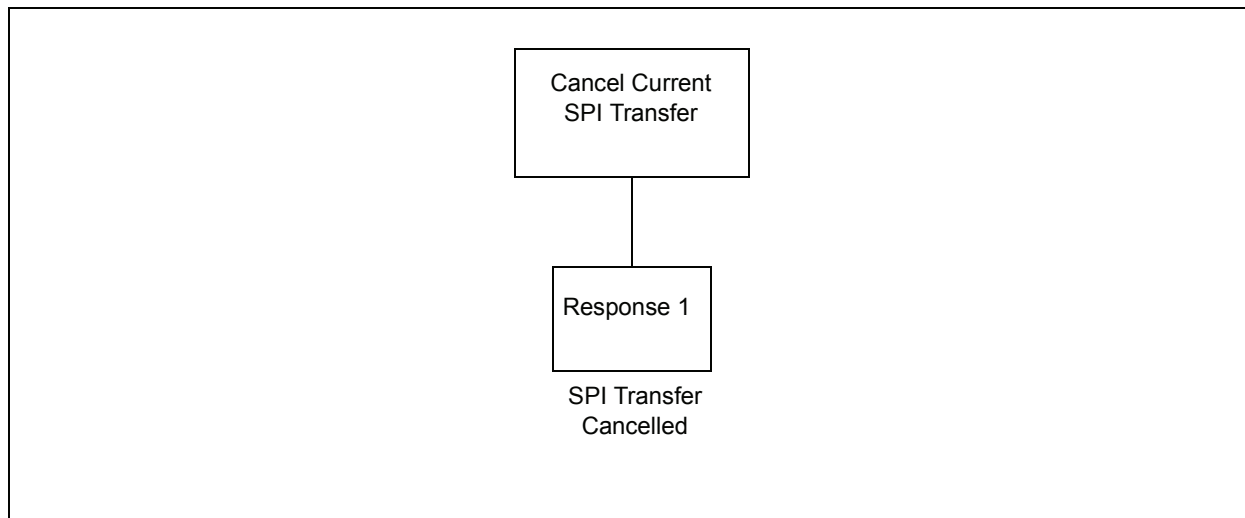
| Byte Index | Meaning  |
|------------|--|
| 0          | 0x11 – <b>CANCEL the current SPI transfer</b> – command code |
| 1          | 0x00 – <b>Reserved</b>                                       |
| 2          | 0x00 – <b>Reserved</b>                                       |
| 3-63       | 0x00 – <b>Reserved</b>                                       |

### 3.5.2.1 Responses

**TABLE 3-65: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x11 – <b>CANCEL the current SPI transfer</b> – echos back the given command code  |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | <b>SPI Bus Release External Request Status</b> <ul style="list-style-type: none"> <li>• 0x01 – No External Request for SPI Bus Release</li> <li>• 0x00 – Pending External Request for SPI Bus Release</li> </ul> |
| 3          | <b>SPI Bus Current Owner</b> <ul style="list-style-type: none"> <li>• 0x00 – No Owner</li> <li>• 0x01 – USB Bridge</li> <li>• 0x02 – External Master</li> </ul>  |
| 4          | <b>Attempted Password Accesses</b> – informs the USB host on how many times the NVRAM password was tried   |
| 5          | <b>Password Guessed</b> <ul style="list-style-type: none"> <li>• 0x00 – Password Not Guessed</li> <li>• 0x01 – Password Guessed</li> </ul>   |
| 6-63       | <b>Don't Care</b>  |

**FIGURE 3-24: CANCEL THE CURRENT SPI TRANSFER LOGIC FLOW**



## 3.5.3 REQUEST SPI BUS RELEASE

TABLE 3-66: COMMAND STRUCTURE

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x80 – <b>Request SPI bus Release</b> – command code  |
| 1          | The value of the SPI Bus Release ACK pin (only if GP7 is assigned to this dedicated function) |
| 2          | 0x00 – <b>Reserved</b>  |
| 3-63       | 0x00 – <b>Reserved</b>  |

## 3.5.3.1 Responses

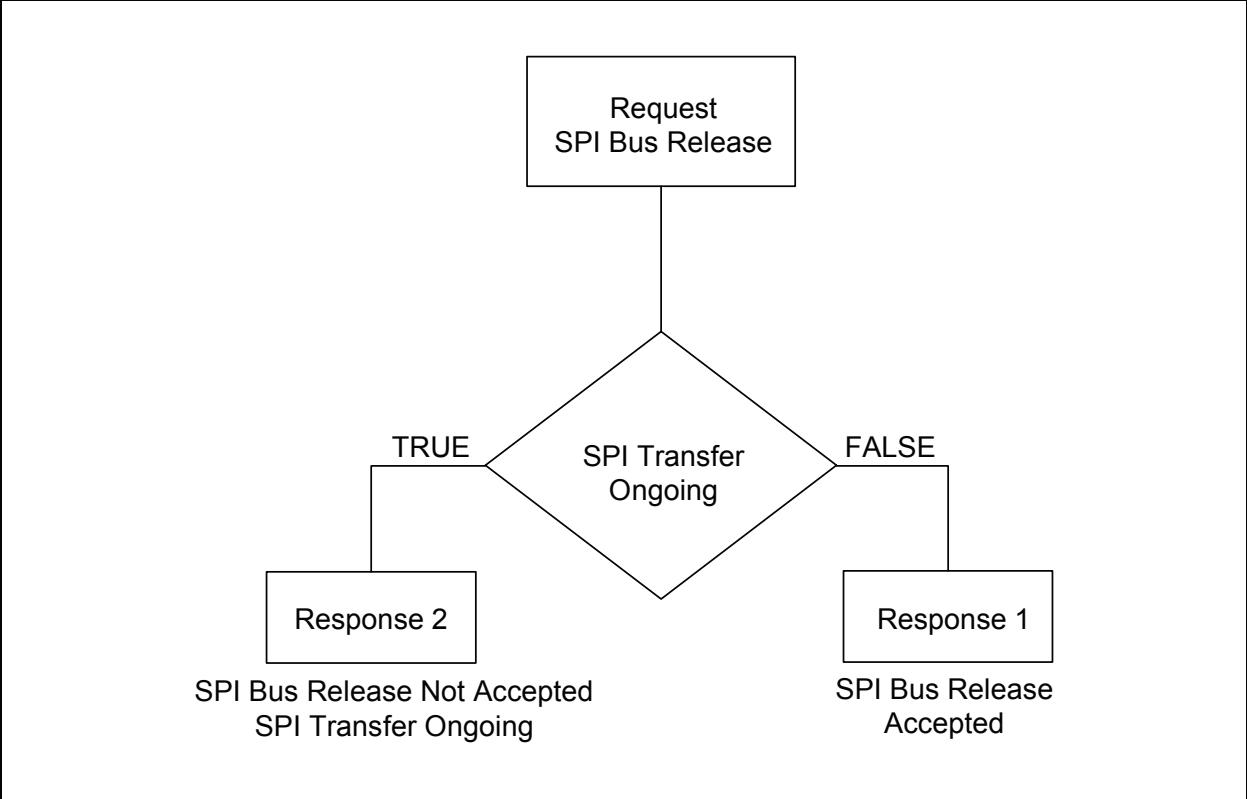
TABLE 3-67: RESPONSE 1 STRUCTURE

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x80 – <b>Request SPI bus Release</b> – echos back the given command code |
| 1          | 0x00 – <b>Command Completed Successfully</b> – SPI bus released           |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

TABLE 3-68: RESPONSES 2 STRUCTURE

| Byte Index | Meaning   |
|------------|---|
| 0          | 0x80 – <b>Request SPI bus Release</b> – echos back the given command code |
| 1          | 0xF8 – <b>SPI Bus Not Released</b> – SPI transfer in process              |
| 2          | <b>Don't Care</b>   |
| 3-63       | <b>Don't Care</b>   |

FIGURE 3-25: REQUEST SPI BUS RELEASE LOGIC FLOW



## 3.6 Chip Status

The chip status command is used to retrieve status information regarding the state of the SPI transfer engine.

### 3.6.1 GET MCP2210 STATUS

**TABLE 3-69: COMMAND STRUCTURE**

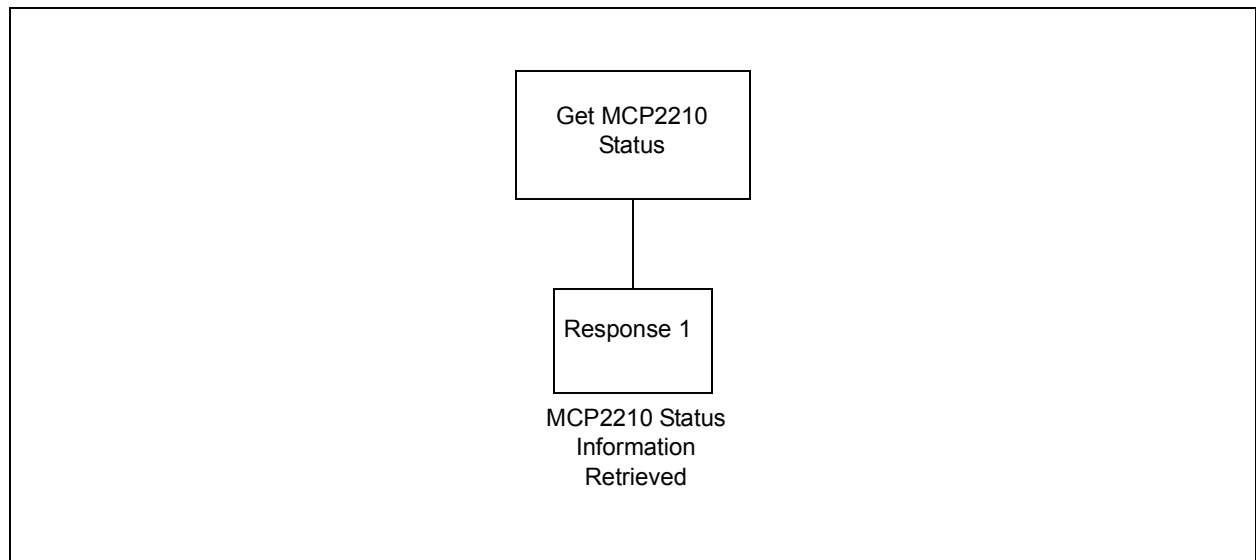
| Byte Index | Meaning   |
|------------|---|
| 0          | 0x10 – <b>Get MCP2210 Status</b> – command code |
| 1          | 0x00 – <b>Reserved</b>                          |
| 2          | 0x00 – <b>Reserved</b>                          |
| 3-63       | 0x00 – <b>Reserved</b>                          |

#### 3.6.1.1 Responses

**TABLE 3-70: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | 0x10 – <b>Get MCP2210 Status</b> – echos back the given command code   |
| 1          | 0x00 – <b>Command Completed Successfully</b>   |
| 2          | <b>SPI Bus Release External Request Status</b> <ul style="list-style-type: none"> <li>• 0x01 – No External Request for SPI Bus Release</li> <li>• 0x00 – Pending External Request for SPI Bus Release</li> </ul> |
| 3          | <b>SPI Bus Current Owner</b> <ul style="list-style-type: none"> <li>• 0x00 – No Owner</li> <li>• 0x01 – USB Bridge</li> <li>• 0x02 – External Master</li> </ul>  |
| 4          | <b>Attempted Password Accesses</b> – informs the USB host on how many times the NVRAM password was tried   |
| 5          | <b>Password Guessed</b> <ul style="list-style-type: none"> <li>• 0x00 – Password Not Guessed</li> <li>• 0x01 – Password Guessed</li> </ul>   |
| 6-63       | <b>Don't Care</b>  |

**FIGURE 3-26: GET MCP2210 STATUS LOGIC FLOW**



# MCP2210

---

## 3.6.2 UNSUPPORTED COMMAND CODES

**TABLE 3-71: COMMAND STRUCTURE**

| Byte Index | Meaning                         |
|------------|---------------------------------|
| 0          | <b>Unsupported Command Code</b> |
| 1          | <b>Don't Care</b>               |
| 2-63       | <b>Don't Care</b>               |

### 3.6.2.1 Responses

**TABLE 3-72: RESPONSE 1 STRUCTURE**

| Byte Index | Meaning  |
|------------|--|
| 0          | <b>Unsupported Command Code Sent</b> – echos back the given command code |
| 1          | 0xF9 – <b>Unknown Command</b> – No effect                                |
| 2-63       | <b>Don't Care</b>  |

NOTES:



# MCP2210

## 4.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

|   |                        |
|---|------------------------|
| Ambient temperature under bias .....  | -40°C to +85°C         |
| Storage temperature .....   | -65°C to +150°C        |
| Voltage on VDD with respect to VSS .....  | -0.3V to +6.0V         |
| Voltage on $\overline{\text{MCLR}}$ with respect to VSS .....                       | -0.3V to +9.0V         |
| Voltage on VUSB pin with respect to VSS .....                                       | -0.3V to +4.0V         |
| Voltage on D+ and D- pins with respect to VSS .....                                 | -0.3V to (VUSB + 0.3V) |
| Voltage on all other pins with respect to VSS .....                                 | -0.3V to (VDD + 0.3V)  |
| Total power dissipation <sup>(1)</sup> .....  | 800 mW                 |
| Maximum current out of VSS pin .....  | 95 mA                  |
| Maximum current into VDD pin .....  | 95 mA                  |
| Clamp current, I <sub>K</sub> (V <sub>PIN</sub> < 0 or V <sub>PIN</sub> > VDD)..... | ± 20 mA                |
| Maximum output current sunk by any I/O pin.....                                     | 25 mA                  |
| Maximum output current sourced by any I/O pin.....                                  | 25 mA                  |
| Maximum current sunk by all ports.....  | 90 mA                  |
| Maximum current sourced by all ports .....  | 90 mA                  |

**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ .

**2:** V<sub>USB</sub> must always be  $\leq V_{DD} + 0.3V$

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

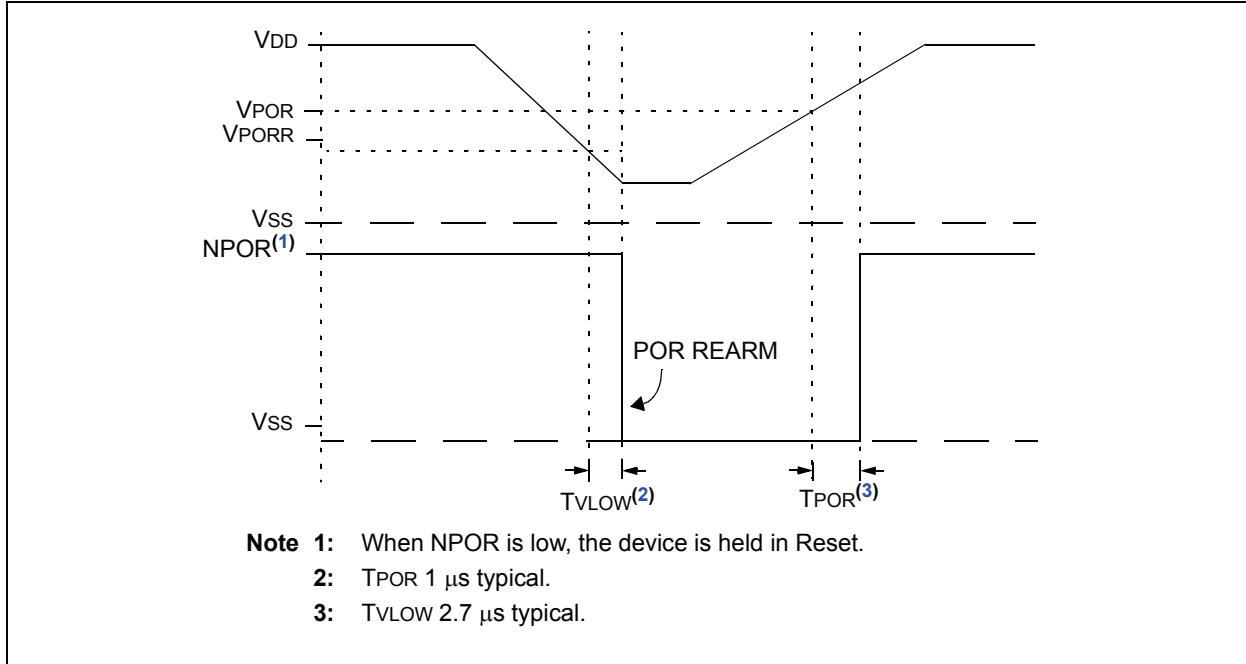
## 4.1 DC CHARACTERISTICS

| DC Characteristics                             |   | Operating Conditions (unless otherwise indicated):<br>3.0V ≤ VDD ≤ 5.5V at -40°C ≤ TA ≤ +85°C (I-Temp) |           |     |         |       |                                    |
|--|---|--|-----------|-----|---------|-------|------------------------------------|
| Param No.                                      | Characteristic                                | Sym  | Min       | Typ | Max     | Units | Conditions                         |
| D001   | Supply Voltage                                | VDD  | 3.3       | —   | 5.5     | V     |                                    |
|  | Power-on Reset Release Voltage                | VPOR   |           | 1.6 |         | V     |                                    |
|  | Power-on Reset Rearm Voltage                  |  |           | 0.8 |         | V     |                                    |
| D003   | VDD Rise Rate to Ensure Power-on Reset        | SVDD   | 0.05      | —   | —       | V/ms  | Design guidance only<br>Not tested |
| D004   | Supply Current                                | IDD  |           |     |         |       |                                    |
|  | VDD = 3.0V                                    |  | —         | 10  | 12      | mA    | FOSC = 12 MHz,<br>(330 nF on VUSB) |
|  | VDD = 5.0V                                    |  | —         | 13  | 15      | mA    |                                    |
| D005   | Standby current                               | IDDS   | —         | 9   | —       | µA    |                                    |
| <b>Input Low Voltage</b>                       |   |  |           |     |         |       |                                    |
| D031   | Schmitt Trigger (GP0–3, 6–8, MOSI, MISO, SCK) | VIL  | —         | —   | 0.2 VDD | V     | 3.0V ≤ VDD ≤ 5.5V                  |
|  | TTL (GP4, GP5)                                |  | —         | —   | 0.8     |       | 4.5V ≤ VDD ≤ 5.5V                  |
| <b>Input High Voltage</b>                      |   |  |           |     |         |       |                                    |
| D041   | Schmitt Trigger (GP0–3, 6–8, MOSI, MISO, SCK) | VIH  | 0.8 VDD   | —   | VDD     | V     | 3.0V ≤ VDD ≤ 5.5V                  |
|  | TTL (GP4, GP5)                                |  | 2.0       | —   | VDD     |       | 4.5V ≤ VDD ≤ 5.5V                  |
| <b>Input Leakage Current</b>                   |   |  |           |     |         |       |                                    |
| D060   | GP0–8, MOSI, MISO, SCK                        | IIL  | —         | ±50 | ±100    | nA    | VSS ≤ VPIN ≤ VDD,<br>pin at Hi-Z   |
|  | RST   |  |           | ±50 | ±200    |       |                                    |
|  | OSC1  |  |           | ±50 | ±100    |       |                                    |
| <b>Output Low Voltage</b>                      |   |  |           |     |         |       |                                    |
| D080   | GP0–8, MOSI, MISO, SCK                        | VOL  | —         | —   | 0.6     | V     | IOI = 8.0 mA, VDD = 5.0V           |
|  |   |  | —         | —   | 0.6     |       | IOI = 6.0 mA, VDD = 3.3V           |
| <b>Output High Voltage</b>                     |   |  |           |     |         |       |                                    |
| D090   | GP0–8, MOSI, MISO, SCK                        | VOH  | VDD – 0.7 | —   | —       | V     | IOH = -3.5 mA, VDD = 5.0V          |
|  |   |  | VDD – 0.7 | —   | —       |       | IOH = -3.0 mA, VDD = 3.3V          |
| <b>Capacitive Loading Specs on Output Pins</b> |   |  |           |     |         |       |                                    |
| D101   | OSC2  | COSC2  | —         | —   | 15      | pF    | <a href="#">Note 1</a>             |
| D102   | GP0–8, MOSI, MISO, SCK                        | CIO  | —         | —   | 50      | pF    | <a href="#">Note 1</a>             |

**Note 1:** This parameter is characterized, but not tested.

# MCP2210

**FIGURE 4-1: POR AND POR REARM WITH SLOW RISING V<sub>DD</sub>**



**TABLE 4-1: USB MODULE SPECIFICATIONS**

| DC Characteristics |  | Operating Conditions (unless otherwise indicated):<br>3.0V $\leq$ V <sub>DD</sub> $\leq$ 5.5V at -40°C $\leq$ T <sub>A</sub> $\leq$ +85°C (I-Temp) |     |     |         |          |  |
|--------------------|--|--|-----|-----|---------|----------|--|
| Param No.          | Characteristic                         | Sym  | Min | Typ | Max     | Units    | Conditions   |
| D313               | USB Voltage                            | V <sub>USB</sub>   | 3.0 | —   | 3.6     | V        | Voltage on V <sub>USB</sub> pin must be in this range for proper USB operation       |
| D314               | Input Leakage on Pin                   | I <sub>IL</sub>  | —   | —   | $\pm 1$ | $\mu$ A  | V <sub>SS</sub> $\leq$ V <sub>PIN</sub> $\leq$ V <sub>DD</sub> pin at high-impedance |
| D315               | Input Low Voltage for USB Buffer       | V <sub>ILUSB</sub>   | —   | —   | 0.8     | V        | For V <sub>USB</sub> range   |
| D316               | Input High Voltage for USB Buffer      | V <sub>IHUSB</sub>   | 2.0 | —   | —       | V        | For V <sub>USB</sub> range   |
| D318               | Differential Input Sensitivity         | V <sub>DIFS</sub>  | —   | —   | 0.2     | V        | The difference between D+ and D- must exceed this value while V <sub>CM</sub> is met |
| D319               | Differential Common Mode Range         | V <sub>CM</sub>  | 0.8 | —   | 2.5     | V        |  |
| D320               | Driver Output Impedance <sup>(1)</sup> | Z <sub>OUT</sub>   | 28  | —   | 44      | $\Omega$ |  |
| D321               | Voltage Output Low                     | V <sub>OL</sub>  | 0.0 | —   | 0.3     | V        | 1.5 k $\Omega$ load connected to 3.6V  |
| D322               | Voltage Output High                    | V <sub>OH</sub>  | 2.8 | —   | 3.6     | V        | 1.5 k $\Omega$ load connected to ground  |

**Note 1:** The D+ and D- signal lines have been built-in impedance matching resistors. No external resistors, capacitors or magnetic components are necessary on the D+/D- signal paths between the MCP2210 family device and the USB cable.

**TABLE 4-2: THERMAL CONSIDERATIONS**

| Standard Operating Conditions (unless otherwise stated)                                |                |  |       |                             |  |
|--|----------------|--|-------|-----------------------------|--|
| Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (I-Temp) |                |  |       |                             |  |
| Param No.  | Sym            | Characteristic                         | Typ   | Units                       | Conditions   |
| TH01   | $\theta_{JA}$  | Thermal Resistance Junction to Ambient | 85.2  | $^{\circ}\text{C}/\text{W}$ | 20-pin SOIC package  |
|  |                |  | 108.1 | $^{\circ}\text{C}/\text{W}$ | 20-pin SSOP package  |
|  |                |  | 36.1  | $^{\circ}\text{C}/\text{W}$ | 20-pin QFN 5x5 mm package  |
| TH02   | $\theta_{JC}$  | Thermal Resistance Junction to Case    | 24    | $^{\circ}\text{C}/\text{W}$ | 20-pin SOIC package  |
|  |                |  | 24    | $^{\circ}\text{C}/\text{W}$ | 20-pin SSOP package  |
|  |                |  | 1.7   | $^{\circ}\text{C}/\text{W}$ | 20-pin QFN 5x5 mm package  |
| TH03   | $T_{JMAX}$     | Maximum Junction Temperature           | 150   | $^{\circ}\text{C}$          |  |
| TH04   | PD             | Power Dissipation                      | —     | W                           | $PD = P_{INTERNAL} + P_{I/O}$  |
| TH05   | $P_{INTERNAL}$ | Internal Power Dissipation             | —     | W                           | $P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$                            |
| TH06   | $P_{I/O}$      | I/O Power Dissipation                  | —     | W                           | $P_{I/O} = \sum (I_{OL} * V_{OL}) + \sum (I_{OH} * (V_{DD} - V_{OH}))$ |
| TH07   | $P_{DER}$      | Derated Power                          | —     | W                           | $P_{DER} = P_{D_{MAX}} (T_J - T_A) / \theta_{JA}^{(2,3)}$              |

**Note 1:**  $I_{DD}$  is the current to run the chip alone without driving any load on the output pins.

**2:**  $T_A$  = Ambient Temperature.

**3:**  $T_J$  = Junction Temperature.

# MCP2210

## 4.2 AC Characteristics

### 4.2.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created in one of the following formats:

1. TppS2ppS

2. TppS

|  |                    |
|--|--------------------|
| <p>T</p> <p>F      Frequency</p> <p>E      Error</p> | <p>T      Time</p> |
|--|--------------------|

Lowercase letters (pp) and their meanings:

|   |  |
|---|--|
| <p>pp</p> <p>io      Input or Output pin</p> <p>rx      Receive</p> <p>bitclk   RX/TX BITCLK</p> <p>drt      Device Reset Timer</p> | <p>osc      Oscillator</p> <p>tx      Transmit</p> <p>RST      Reset</p> |
|---|--|

Uppercase letters and their meanings:

|   |  |
|---|--|
| <p>S</p> <p>F      Fall</p> <p>H      High</p> <p>I      Invalid (high-impedance)</p> <p>L      Low</p> | <p>P      Period</p> <p>R      Rise</p> <p>V      Valid</p> <p>Z      High-impedance</p> |
|---|--|

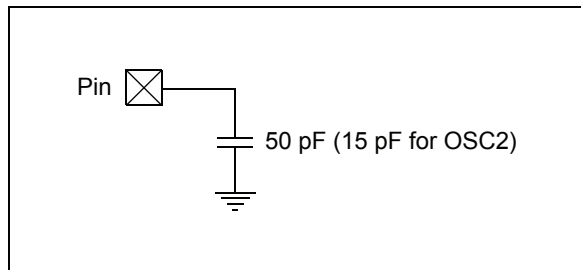
### 4.2.2 TIMING CONDITIONS

The operating temperature and voltage specified in [Table 4-3](#) apply to all timing specifications unless otherwise noted. [Figure 4-2](#) specifies the load conditions for the timing specifications.

**TABLE 4-3: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC**

|                           |   |
|---------------------------|---|
| <b>AC CHARACTERISTICS</b> | <p><b>Standard Operating Conditions (unless otherwise stated)</b></p> <p>Operating temperature <math>-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}</math></p> <p>Operating voltage <math>V_{DD}</math> range as described in DC spec, <a href="#">Section 4.1 “DC Characteristics”</a>.</p> |
|---------------------------|---|

**FIGURE 4-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS**



## 4.2.3 TIMING DIAGRAMS AND SPECIFICATIONS

**TABLE 4-4: RESET, OSCILLATOR START-UP TIMER AND POWER-UP TIMER PARAMETERS**

| Standard Operating Conditions (unless otherwise stated)                       |       |                          |     |                  |     |       |            |
|---|-------|--------------------------|-----|------------------|-----|-------|------------|
| Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ |       |                          |     |                  |     |       |            |
| Param No.   | Sym   | Characteristic           | Min | Typ <sup>†</sup> | Max | Units | Conditions |
| 30  | TRST  | MCLR Pulse Width (low)   | 2   | —                | —   | μs    |            |
| 31  | TPWRT | Power-up timer           | 40  | 65               | 140 | ms    |            |
| 32  | TOST  | Oscillator start-up time | —   | 1024             | —   | TOST  |            |

\* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

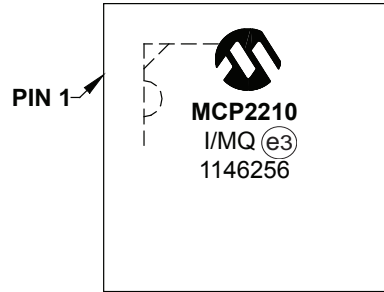
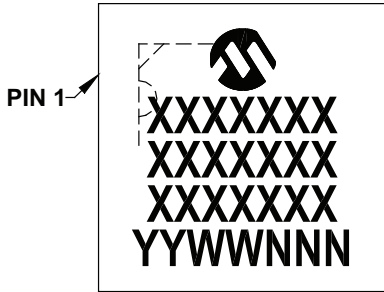
# MCP2210

## 5.0 PACKAGING INFORMATION

### 5.1 Package Marking Information

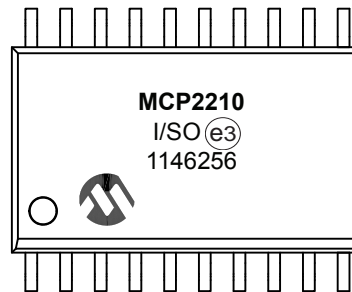
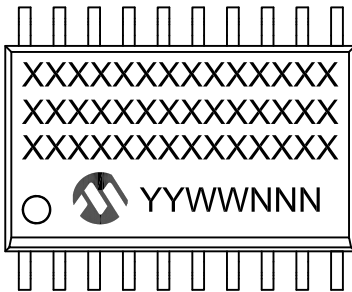
20-Lead 5x5 QFN

Example



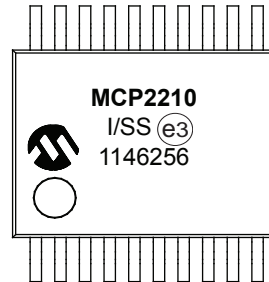
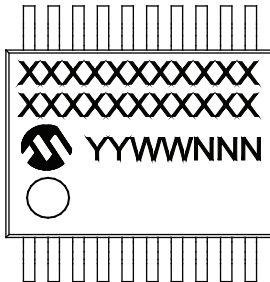
20-Lead SOIC

Example



20-Lead SSOP

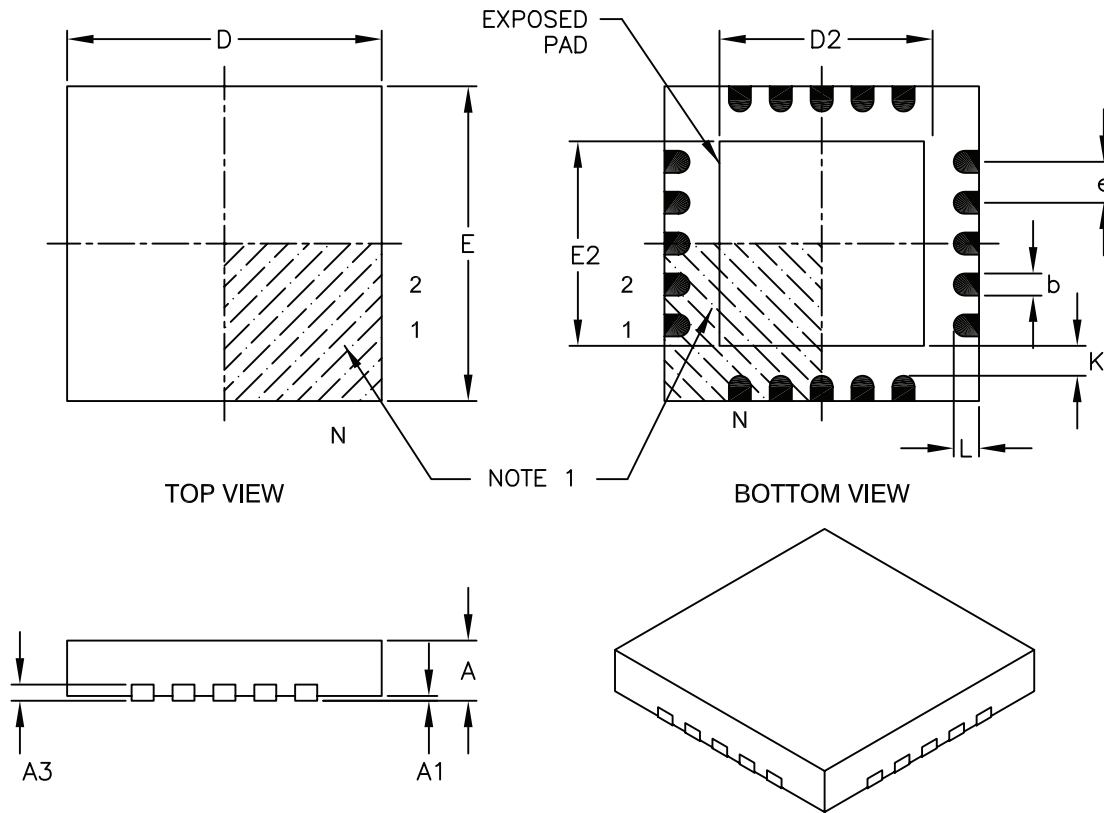
Example



|                |   |  |
|----------------|---|--|
| <b>Legend:</b> | XX...X  | Customer-specific information  |
|                | Y   | Year code (last digit of calendar year)  |
|                | YY  | Year code (last 2 digits of calendar year)   |
|                | WW  | Week code (week of January 1 is week '01')   |
|                | NNN   | Alphanumeric traceability code   |
|                | (e3)  | Pb-free JEDEC designator for Matte Tin (Sn)  |
|                | *   | This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package. |
| <b>Note:</b>   | In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information. |  |

## 20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits       | Units | MILLIMETERS |      |      |
|------------------------|-------|-------------|------|------|
|                        |       | MIN         | NOM  | MAX  |
| Number of Pins         | N     | 20          |      |      |
| Pitch                  | e     | 0.65 BSC    |      |      |
| Overall Height         | A     | 0.80        | 0.90 | 1.00 |
| Standoff               | A1    | 0.00        | 0.02 | 0.05 |
| Contact Thickness      | A3    | 0.20 REF    |      |      |
| Overall Width          | E     | 5.00 BSC    |      |      |
| Exposed Pad Width      | E2    | 3.15        | 3.25 | 3.35 |
| Overall Length         | D     | 5.00 BSC    |      |      |
| Exposed Pad Length     | D2    | 3.15        | 3.25 | 3.35 |
| Contact Width          | b     | 0.25        | 0.30 | 0.35 |
| Contact Length         | L     | 0.35        | 0.40 | 0.45 |
| Contact-to-Exposed Pad | K     | 0.20        | -    | -    |

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

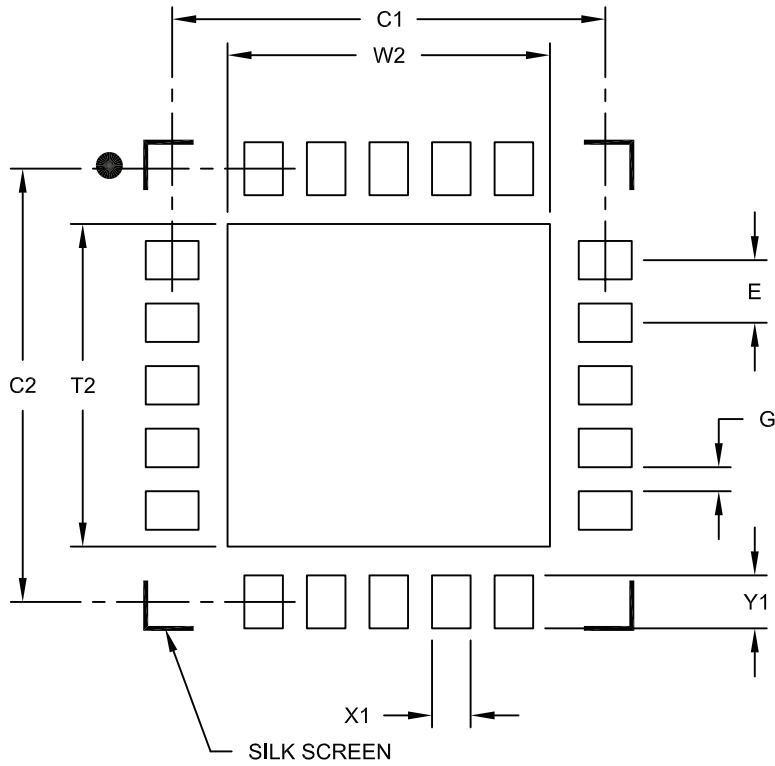
Microchip Technology Drawing C04-139B



# MCP2210

20-Lead Plastic Quad Flat, No Lead Package (MQ) - 5x5 mm Body [QFN]  
 With 0.40mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits           | Units | MILLIMETERS |          |      |
|----------------------------|-------|-------------|----------|------|
|                            |       | MIN         | NOM      | MAX  |
| Contact Pitch              | E     |             | 0.65 BSC |      |
| Optional Center Pad Width  | W2    |             |          | 3.35 |
| Optional Center Pad Length | T2    |             |          | 3.35 |
| Contact Pad Spacing        | C1    |             | 4.50     |      |
| Contact Pad Spacing        | C2    |             | 4.50     |      |
| Contact Pad Width (X20)    | X1    |             |          | 0.40 |
| Contact Pad Length (X20)   | Y1    |             |          | 0.55 |
| Distance Between Pads      | G     | 0.20        |          |      |

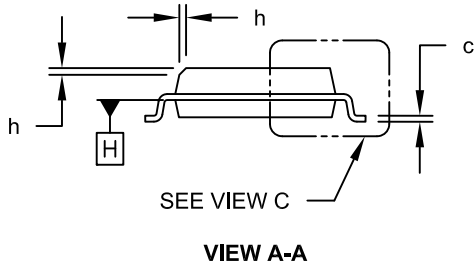
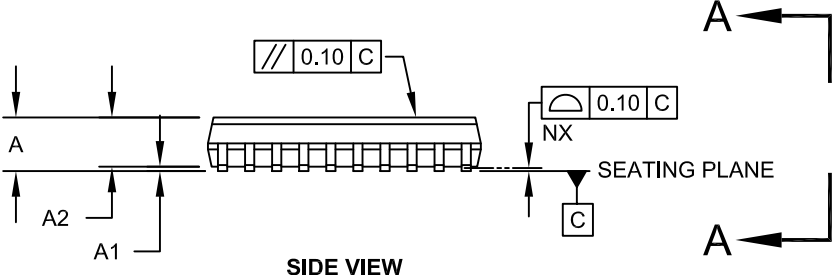
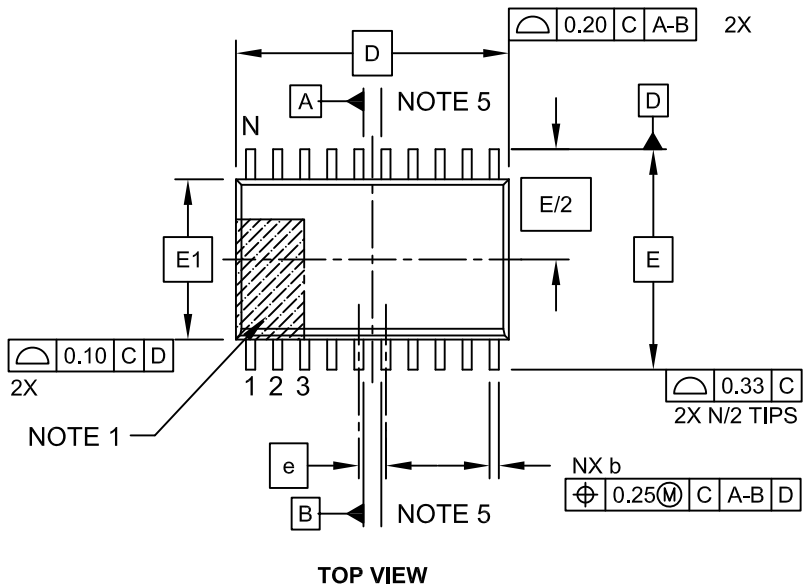
**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2139A

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

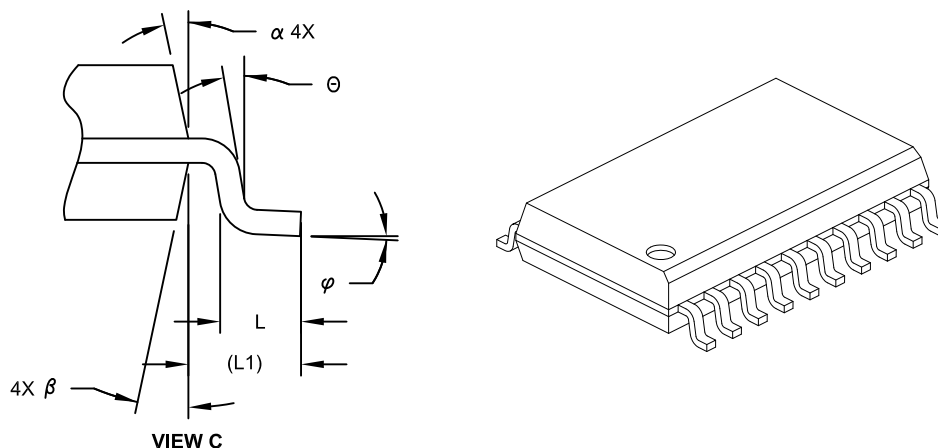
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



# MCP2210

## 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits         | Units     | MILLIMETERS |     |      |
|--------------------------|-----------|-------------|-----|------|
|                          |           | MIN         | NOM | MAX  |
| Number of Pins           | N         | 20          |     |      |
| Pitch                    | e         | 1.27 BSC    |     |      |
| Overall Height           | A         | -           | -   | 2.65 |
| Molded Package Thickness | A2        | 2.05        | -   | -    |
| Standoff §               | A1        | 0.10        | -   | 0.30 |
| Overall Width            | E         | 10.30 BSC   |     |      |
| Molded Package Width     | E1        | 7.50 BSC    |     |      |
| Overall Length           | D         | 12.80 BSC   |     |      |
| Chamfer (Optional)       | h         | 0.25        | -   | 0.75 |
| Foot Length              | L         | 0.40        | -   | 1.27 |
| Footprint                | L1        | 1.40 REF    |     |      |
| Lead Angle               | $\Theta$  | 0°          | -   | -    |
| Foot Angle               | $\varphi$ | 0°          | -   | 8°   |
| Lead Thickness           | c         | 0.20        | -   | 0.33 |
| Lead Width               | b         | 0.31        | -   | 0.51 |
| Mold Draft Angle Top     | $\alpha$  | 5°          | -   | 15°  |
| Mold Draft Angle Bottom  | $\beta$   | 5°          | -   | 15°  |

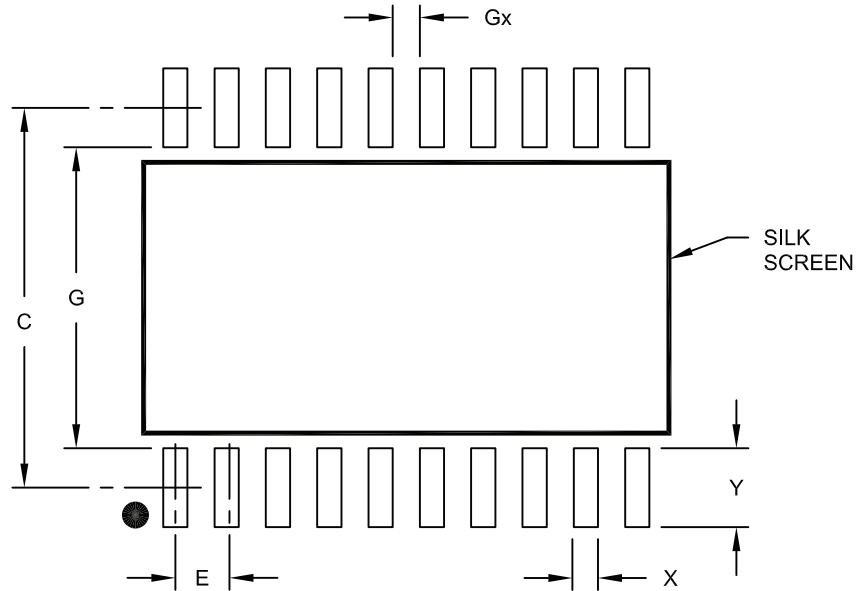
### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

## 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

| Dimension Limits         | Units | MILLIMETERS |     |      |
|--------------------------|-------|-------------|-----|------|
|                          |       | MIN         | NOM | MAX  |
| Contact Pitch            | E     | 1.27 BSC    |     |      |
| Contact Pad Spacing      | C     | 9.40        |     |      |
| Contact Pad Width (X20)  | X     |             |     | 0.60 |
| Contact Pad Length (X20) | Y     |             |     | 1.95 |
| Distance Between Pads    | Gx    | 0.67        |     |      |
| Distance Between Pads    | G     | 7.45        |     |      |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

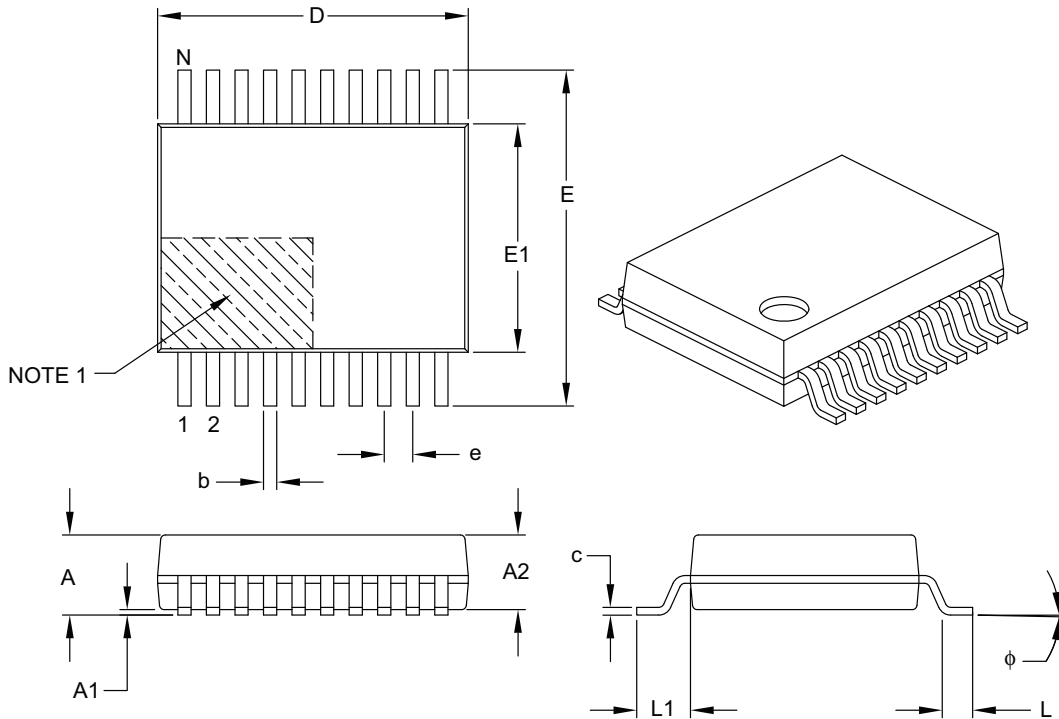
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

# MCP2210

## 20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits         | Units  | MILLIMETERS |      |      |
|--------------------------|--------|-------------|------|------|
|                          |        | MIN         | NOM  | MAX  |
| Number of Pins           | N      | 20          |      |      |
| Pitch                    | e      | 0.65 BSC    |      |      |
| Overall Height           | A      | –           | –    | 2.00 |
| Molded Package Thickness | A2     | 1.65        | 1.75 | 1.85 |
| Standoff                 | A1     | 0.05        | –    | –    |
| Overall Width            | E      | 7.40        | 7.80 | 8.20 |
| Molded Package Width     | E1     | 5.00        | 5.30 | 5.60 |
| Overall Length           | D      | 6.90        | 7.20 | 7.50 |
| Foot Length              | L      | 0.55        | 0.75 | 0.95 |
| Footprint                | L1     | 1.25 REF    |      |      |
| Lead Thickness           | c      | 0.09        | –    | 0.25 |
| Foot Angle               | $\phi$ | 0°          | 4°   | 8°   |
| Lead Width               | b      | 0.22        | –    | 0.38 |

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

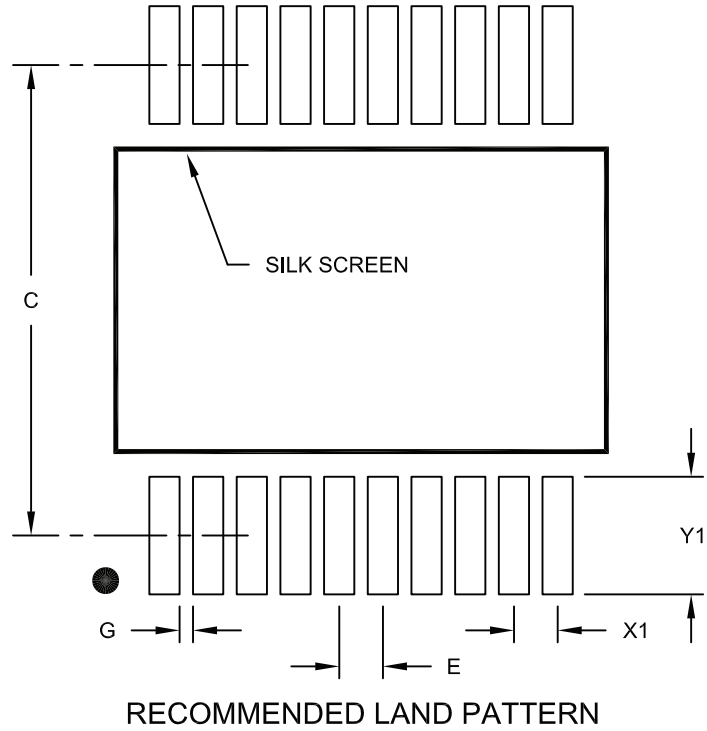
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

## 20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



|                          |    | Units | MILLIMETERS |      |      |
|--------------------------|----|-------|-------------|------|------|
| Dimension Limits         |    |       | MIN         | NOM  | MAX  |
| Contact Pitch            | E  |       | 0.65 BSC    |      |      |
| Contact Pad Spacing      | C  |       |             | 7.20 |      |
| Contact Pad Width (X20)  | X1 |       |             |      | 0.45 |
| Contact Pad Length (X20) | Y1 |       |             |      | 1.75 |
| Distance Between Pads    | G  | 0.20  |             |      |      |

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

# MCP2210

---

NOTES:

## APPENDIX A: REVISION HISTORY

### Revision A (December, 2011)

- Original Release of this Document.



# MCP2210

---

NOTES:

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| <u>PART NO.</u>           | <u>X</u>          | <u>/XX</u>   |
|---------------------------|-------------------|--|
| Device                    | Temperature Range | Package  |
| <b>Device:</b>            | MCP2210:          | USB to SPI Protocol Converter with GPIO                                |
|                           | MCP2210T:         | USB to SPI Protocol Converter with GPIO (Tape and Reel)                |
| <b>Temperature Range:</b> | I                 | = -40°C to +85°C (Industrial)  |
| <b>Package:</b>           | MQ                | = Plastic Quad Flat, No Lead Package<br>5x5x0.9 mm Body (QFN), 20-Lead |
|                           | SO                | = Plastic Small Outline - Wide, 7.50 mm Body (SO),<br>20-Lead          |
|                           | SS                | = Plastic Shrink Small Outline - 5.30 mm Body (SS)<br>20-Lead          |

**Examples:**

a) MCP2210- I/MQ: Industrial temperature, 20LD QFN Package.

b) MCP2210T- I/MQ: Tape and Reel, Industrial temperature, 20LD QFN Package.

a) MCP2210- I/SO: Industrial temperature, 20LD SOIC Package.

b) MCP2210T- I/SO: Tape and Reel, Industrial temperature, 20LD SOIC Package.

a) MCP2210- I/SS: Industrial temperature, 20LD SSOP Package.

b) MCP2210T- I/SS: Tape and Reel, Industrial temperature, 20LD SSOP Package.

# MCP2210

---

NOTES:

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2011, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-61341-902-1

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2009 ==**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*



# MICROCHIP

## Worldwide Sales and Service

### AMERICAS

**Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://www.microchip.com/support>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

**Atlanta**  
Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

**Boston**  
Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

**Chicago**  
Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

**Cleveland**  
Independence, OH  
Tel: 216-447-0464  
Fax: 216-447-0643

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

**Indianapolis**  
Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**China - Beijing**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Chongqing**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**China - Hangzhou**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**China - Nanjing**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**China - Qingdao**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**China - Shanghai**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**China - Shenyang**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

**China - Wuhan**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**China - Xian**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**China - Xiamen**  
Tel: 86-592-2388138  
Fax: 86-592-2388130

**China - Zhuhai**  
Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**India - Pune**  
Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

**Japan - Osaka**  
Tel: 81-66-152-7160  
Fax: 81-66-152-9310

**Japan - Yokohama**  
Tel: 81-45-471-6166  
Fax: 81-45-471-6122

**Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

**Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**Taiwan - Hsin Chu**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**  
Tel: 886-7-536-4818  
Fax: 886-7-330-9305

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

11/29/11

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «JONHON», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «FORSTAR».



## JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон: 8 (812) 309-75-97 (многоканальный)

Факс: 8 (812) 320-03-32

Электронная почта: [ocean@oceanchips.ru](mailto:ocean@oceanchips.ru)

Web: <http://oceanchips.ru/>

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А