Application Note

Scalable PMIC NVM Update Guide

Chris Sterzik *Integrated Power Management*

ABSTRACT

The TPS6594-Q1, TPS6593-Q1, and LP8764-Q1 family of power management integrated circuits (PMICs) include a configurable non-volatile memory (NVM) space. The Scalable PMIC GUI provides the ability to generate a binary image either from an NVM configuration (assembly file) generated from the GUI or generated manually. This document details the hardware setup and steps for uploading the binary image to the NVM of the PMIC.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The configuration process described in this document writes to the NVM space and is intended to be used in a production line or development setting. This mechanism is not intended to be used in applications since the process impacts the regulator outputs and GPIO functions.

The Scalable PMIC GUI provides a mechanism to generate a binary image which can, in turn, be uploaded to the NVM of the PMIC. The binary is generated from the NVM programming page by selecting *Save as Binary Code*, as shown in Figure 1-1. See the *Scalable PMIC's GUI User's Guide* for instructions on configuring the PMIC and generating an image from that configuration.

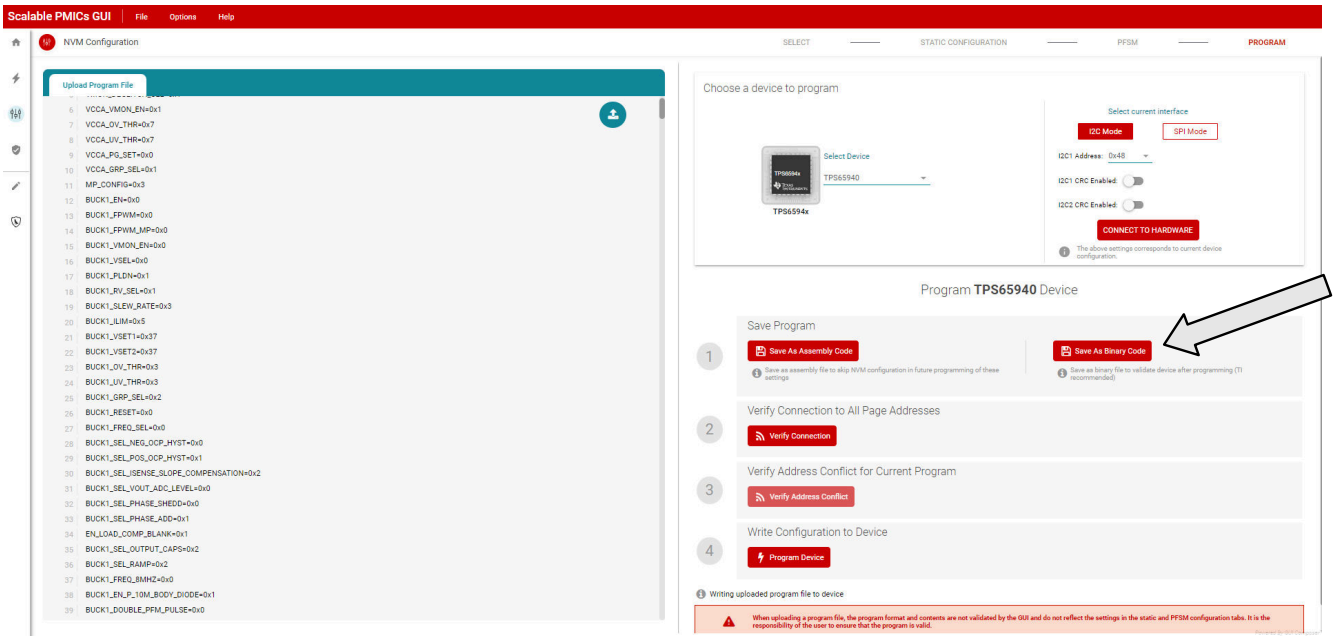

Figure 1-1. Generating a Binary File From the Scalable PMIC GUI

Table 1-1 shows examples of entries in the binary file. Each row consists of the page information, the register address, and the data.

Table 1-1. GUI Binary Output Format

Binary Row	Description
0x0004 = 0xa0	Page 0 entries
0x00d1 = 0x00	
0x0100 = 0x02	Page 1 entries
0x014A = 0x06	
0x3000 = 0x0A	Page 3 entries
0x32ff = 0x00	
0x405 = 0xff	Page 4 entries
0x409 = 0xff	

2 Hardware and PMIC Setup

The hardware connections are described in the schematic checklist (see Reference 6). Table 2-1 and Table 2-2 show the required connections associated with NVM configuration.

Table 2-1. Subset of Schematic Checklist for TPS6594-Q1 and TPS6593-Q1

Pin	Name	Purpose	Connection for Functional Use
2	VOUT_LDOVINT	Output pins of internal LDOs for noise decoupling capacitor	Capacitor: Ctyp = 2.2 µF; Vcap > 6.3 V
3	VOUT_LDOVRTC		
4	VCCA	Analog input voltage for the internal LDOs and other internal blocks	Capacitor: Cmin = 0.47 µF; Ctyp = 1 µF; Vcap>6.3 V
5	REFGND1	System reference ground	Connect to solid ground plane but not the thermal pad on the top layer.
6	REFGND2		
30	SDA_I2C1, SDI_SPI	I²C or SPI data	Connect to data line of controller. For I²C use resistor value depending upon speed and PCB.
31	SCL_I2C1/SCK_SPI	I²C or SPI clock	Connect to clock line of controller. For I²C use resistor value depending upon speed and PCB.
32	CS_SPI(1)	SPI Chip select	Connect to CS of SPI controller
33	SDO_SPI(1)	SPI SDO	Connect to SDO of SPI controller
48	VIO_IN	Digital supply input for GPIOs and I/O supply voltage	Capacitor: Cmin = 0.47 µF; Ctyp = 1 µF; Vcap > 6.3 V
57	Thermal Pad	Power ground, which is also the thermal pad of the package.	Connect to top layer power ground polygon

(1) Pins 32 and 33 are only needed for SPI communication. With I²C only pins 30 and 31 are needed.

Note

Refer to the schematic checklist for the latest HW guidance.

Table 2-2. Subset of Schematic Checklist for LP8764-Q1

Pin	Name	Purpose	Connection for functional use
20	VOUT_LDO	Output pin of internal LDO for noise decoupling capacitor	Capacitor: Ctyp = 2.2 µF; Vcap > 6.3 V
18	VCCA	Analog input voltage for the internal LDOs and other internal blocks	Capacitor: Cmin = 0.1 µF; Ctyp = 0.47 µF; Vcap > 6.3 V
19	AGND1	System reference ground	Connect to solid ground plane but not the thermal pad on the top layer.
21	AGND2		
5	SDA_I2C1, SDI_SPI	I²C or SPI data	Connect to data line of controller. For I²C use resistor value depending upon speed and PCB.
4	SCL_I2C1/SCK_SPI	I²C or SPI clock	Connect to clock line of controller. For I²C use resistor value depending upon speed and PCB.
2	CS_SPI(1)	SPI Chip select	Connect to CS of SPI controller
3	SDO_SPI(1)	SPI SDO	Connect to SDO of SPI controller
24	VIO	Digital supply input for GPIOs and I/O supply voltage	Capacitor: Cmin = 0.1µF; Ctyp = 0.47 µF; Vcap > 6.3 V
13/29	PGND	Power Ground	Connect to top layer power ground polygon

(1) Pins 2 and 3 are only needed for SPI communication. With I²C only pins 4 and 5 are needed.

VCCA and VIO must be applied and the serial interface must be accessible to update the NVM. The VIO must not be connected to or dependent upon any GPIO or regulator from the PMIC. Similarly, in the case of an I^2C serial interface, the pullup voltage must also be independent of the PMIC. When configuring the NVM via I^2C only the I2C1 interface is needed. The I2C2 interface is only used in the application when enabled and utilized solely for the watchdog communication.

The initial PMIC state must also be understood before attempting to configure the NVM. Generally, the PMIC must be in a static or idle state. In some NVM configurations, the PMIC does not power up until the ENABLE pin is activated. Simply holding the ENABLE pin low can be an effective means to hold the PMIC in a known static state.

Also, if the NVM is configured to loop continuously between states polling for a certain condition, this can interfere with the initial steps in the configuration which unlocks the NVM.

Finally, the PMIC does allow for the NVM to be configured while in the safe recovery hardware state. This provides a means to change the NVM in the event that the NVM was erroneously configured and results in an error or shutdown. When defining an NVM, a transition to the safe recovery state is important and must not be omitted. Examples which include the transition to safe recovery are provided in the Scalable PMIC GUI as templates. If a transition is not provided, then the associated interrupts must be serviced for the NVM to be unlocked successfully.

3 Configuration Overview

The PMIC has two memory spaces, the register map space and the NVM space. As Figure 3-1 shows, configuring the NVM is done by first writing to the register map through the serial interface and then copying the contents to the NVM. Because the configuration first involves writing to the register map, which controls the regulators and GPIO, there must be no dependency or need to use the PMIC resources. For example, in the PDN-0B implementation(4) the VIO is supplied by a load switch which is controlled by the PMIC GPIO. The dependency upon the GPIO output prohibits NVM configuration since the GPIO controls the VIO, impacting the I^2C and corrupting the configuration. In contrast, the TPS6594 EVM(5) provides an independent supply for VIO.

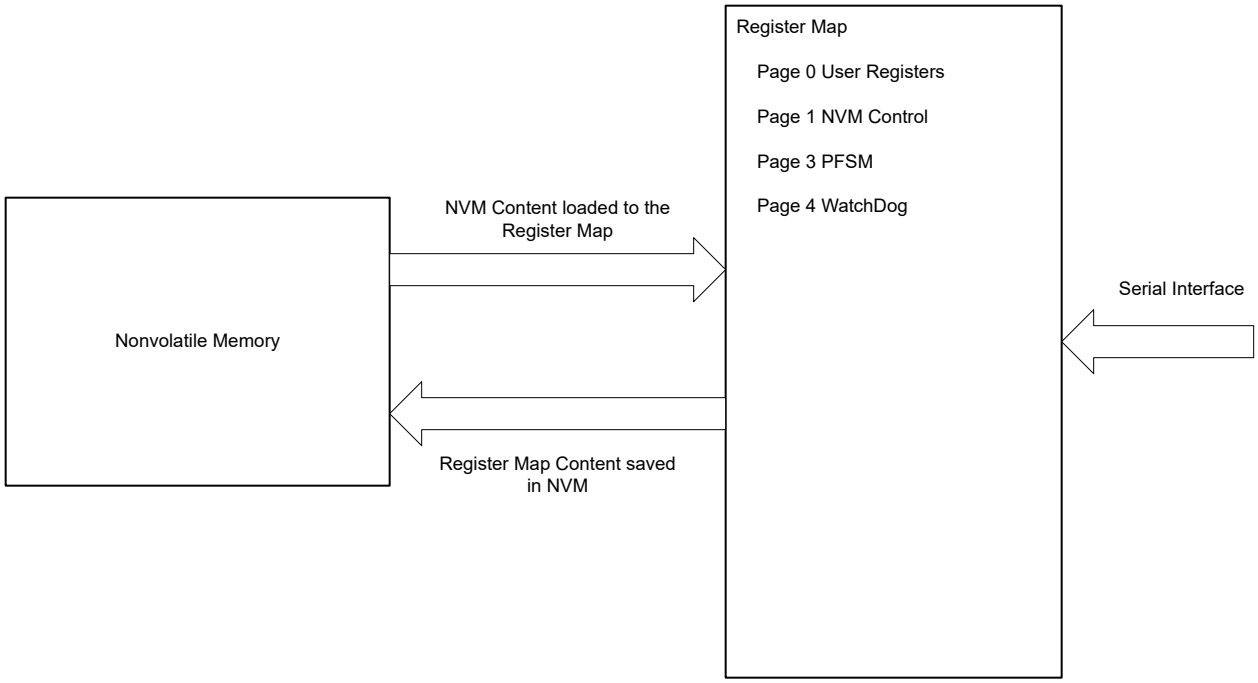

Figure 3-1. Register Map Memory and NVM Spaces

4 Instructions

These instructions refer to pages and register addresses. When an I^2C serial interface is used, the pages are delineated by distinct I^2C addresses while in SPI the page information is included in the second byte of the transmission payload. The serial interface can be changed as part of the NVM configuration as discussed in Section 5. See the device data sheet regarding page implementation.

1. Unlocking the NVM
   Unlocking the NVM requires a series of write accesses to the USER_EE_CTRL_1 register located at register address 0xA2 in page 0. Write '0x00' to the register first to reset the unlock mechanism. The next four writes must occur in order and no register accesses can be performed in between the writes. First, 0x98, second, 0xB8, third, 0x13, and fourth 0x7D. When the NVM has been successfully unlocked, both bits 6 and 7 of register address 0xA3 in page 0 are set.

   Note

   Register accesses include the PFSM as well as the serial interface. During the unlocking of the NVM, the PMIC must be in an idle state and the PFSM is not attempting to access the register map. The MCU performing the unlock must also limit register accesses to only those writes associated with the unlock sequence. In some MCU serial communications an automatic readback of each register is performed after each write. Any such mechanism must be turned off.
2. Disable the PFSM
   Once the NVM has been unlocked the next step is to disable the PFSM. This is done by setting bit 0 of register address 0xA3 in page 0. Since this register also contains the contents of the unlock, write 0xC1 to the register.
3. Special Considerations for Buck Frequency and Serial Interface Changes
   After disabling the PFSM and before writing the new content to the user registers any special considerations for buck frequency or interface changes must be applied. If the buck frequency remains the same and the serial interface is unchanged then no additional instructions are required. See the applicable special considerations as well as the example. Details are provided in Section 5.1 and Section 5.2
4. Writing Content to pages 0, 1, and 4
   Writing to pages 0, 1, and 4 must only include the register addresses described in Appendix A. In the event that the register CRC is enabled, then non-NVM registers must also be returned to their default value as described in Appendix B.
5. Writing Content to page 3
   After completion of pages 0,1, and 4, write to page 3. Page 3 is a special use case and requires additional handling to access the *sub-pages* for the PFSM memory space. The page and sub-page delineations can also be identified in the binary file, where page 0 is in the address space 0x00-0xFF, page 1 is 0x100-0x1FF, page 4 is 0x400-0x4FF, and page 3, sub-page 0, is 0x3000-0x30FF, sub-page 1 is 0x3100-0x31FF, and sub-page 2 is 0x3200-0x32FF. A detailed description of the sub-pages is found in the PFSM section. Selecting the Page 3 sub-pages is done through register 0xA4 on Page 0.

   CAUTION

   The TPS6594-Q1 and TPS6593-Q1 family of devices have three subpages for page 3. The LP8764-Q1 device only has two subpages for page 3 and sub-page 2 does not exist. Attempting to write to sub-page 2 of the LP8764-Q1 results in writing to sub-page 1, corrupting the memory space.

Table 4-1. PFSM Address Control, Register Address 0xA4, Page 0

Bit	Field	Type	Reset	Description
7-2	Reserved	R/W	0h	
1-0	PFSM_PAGE_SEL	R/W	0h	Select the Page 3 address space that can be addressed. 0: 0x000-0x0FF 1: 0x100-0x1FF 2: 0x200-0x2FF

6. Lock the NVM to prevent future programming (optional)
   If locking the NVM to prevent future updates is desired, then the value of register 0x41 on page 1 must be something other than 0xA5. Locking the NVM also prevents the ability to access page 3 for validation purposes.

   Note

   Locking the NVM is a permanent decision and prohibits any future changes.
7. Update register CRC
   After all of the content has been updated in the register map, then the register CRC can be calculated and applied. This is only necessary if the register CRC is enabled in the NVM being configured.
8. Move content from register map to NVM
   Now that all of the user register content has been updated and the register CRC updated, the contents can be copied into the NVM space. To initiate the transfer of content, set bit 1 of register 0xEF in page 1. The transfer is not instantaneous and the status of the transfer can be observed in bit 1 of register address 0xF3 in page 1. When this bit is cleared then the transfer activity is complete.

5 Special Considerations

5.1 Changing the Serial Control Interface

The serial control interface registers are 0x11A, 0x122, and 0x123. As shown in Table 5-1, 0x11A is the serial interface register and indicates if the serial interface is either I2C or SPI and if the CRC is enabled on the interfaces, I2C1 or SPI and I2C2. The I2C1 address is stored in 0x122 and the I2C2 address is stored in 0x123.

Table 5-1. Serial Interface Register

Page 1, Register Address 0x1A Bit Field	Description
Bit 2	• 0 = I2C2 CRC Disabled • 1 = I2C2 CRC Enabled
Bit 1	• 0 = I2C1 or SPI CRC Disabled • 1 = I2C1 or SPI CRC Enabled
Bit 0	• 0 = I2C Mode • 1 = SPI Mode

Once the NVM is successfully unlocked, the serial interface can be changed. Once the change is made, the previous serial interface no longer applies and the interface must change appropriately on the host side. In the context of an I2C address change, the necessary updates must be made for the pages. For example, if the I2C1 address is changed from 0x48 to 0x28, then pages before and after the change are represented in Table 5-2.

Table 5-2. I2C Address and Page Relationship

Page(1)	• I2C1 Address = 0x48 • I2C2 Address = 0x12	• I2C1 Address = 0x28 • I2C2 Address = 0x12
Page 0	0x48	0x28
Page 1	0x49	0x29
Page 3	0x4B	0x2B
Page 4	0x12	0x12

(1) Page 4 is always directly associated with I2C2 and page 0 is directly associated with I2C1. Pages 1 and 3 are always offset from page 0.

For a change from SPI to I^2C or I^2C to SPI, the necessary GPIO must be available to support both interfaces. Provide a wait or delay so that the hardware is correctly configured to the desired serial interface before proceeding with the NVM instructions. See the device data sheet for an explanation of the I^2C interface and SPI.

Note

While the NVM is unlocked, the I2C2 physical interface is not used and all pages are accessed from I2C1 of the PMIC(SDA_I2C1, SCL_I2C1).

5.2 Updating the Frequency Selection

The buck frequency selection is protected by bit 3 in register address 0x18 of page 1. The buck frequency cannot be updated until this bit is set and the bit cannot be set unless the NVM is unlocked. Once the frequency update is made the bit must be cleared. Typically this is accomplished in the normal program flow when register 0x18 of page 1 is written.

5.3 PFSM

The PFSM is divided into sub-pages. The TPS659x devices have 3 sub-pages while the LP876x family of devices have 2. Control of the sub-pages is managed in bits 0 and 1 of register address 0xa4 of page 0. The contents of the PFSM which are stored in binary file are to be moved to the page 3 user registers as indicated in Table 5-3.

CAUTION

The TPS6594-Q1 and TPS6593-Q1 family of devices have three subpages for page 3. The LP8764-Q1 device only has two subpages for page 3 and sub-page 2 does not exist. Attempting to write to sub-page 2 of the LP8764-Q1 results in writing to sub-page 1, corrupting the memory space.

Table 5-3. PFSM Sub-page Control

Page, Register Address	Page 0, Register Address 0xA4		Binary File
	Bit 1	Bit 0	
3, 0x00-0xFF	0	0	0x3000-0x30FF
	0	1	0x3100-0x31FF
	1	0	0x3200-0x32FF

5.4 Permanently Locking the NVM

To permanently lock the NVM, the register address 0x41 in page 1 must be changed to some value other than 0xA5. 0xA5 is the default value and therefore no action is needed to keep the NVM in the unlocked state. When the NVM is permanently locked the PMIC functionality remains unchanged with the exception that the NVM can no longer be updated. Access to the register map, shown in Figure 3-1, is preserved while the NVM is locked.

5.5 Updating the Register CRC

Note

The Register CRC update is only required if the Register CRC is enabled in the NVM being programmed. Bit 6, of register 0x18, of page 1 is set.

The process for updating the register CRC for the Page 3 and Page 1 contents uses the CRC update feature within the PMIC. Before running the CRC update, clear the register CRC contents in address 0xF0 through 0xFB. The CRC update is performed when bit 1 of register 0xEF in page 0 is set. This bit is self-clearing. Once the update is complete, the content of address 0xFB, page 0, becomes a non-zero value. This check of register 0xFB changing from a 0 to non-zero number is recommended to confirm that the CRC is complete before attempting move on to the next step.

The PMIC CRC update feature calculates the CRC for pages 0,1,3, and 4. Page 2 is protected memory space as well as the associated CRC and is not described in this document. In most applications the calculation for the user registers, the combination of pages 1 and 4, is incorrect. The value of register 0x82 on page 0 does not match the default value, see Appendix B, and the bits SPMI_LPM_EN and FORCE_EN_DRV_LOW, found in the register are read only from the serial interface.

Note

If the data value of Page 0 register 0x82, ENABLE_DRV_STAT, is not 0x08, then the register CRC for Page 0 and Page 4 must be computed manually and the results written to the PMIC.

The Page 0 and Page 4 register CRC has two components: include persist and exclude persist. As shown in Table 5-4, the include persist CRC information is stored in registers 0xF0 and 0xF1 of page 0 and the exclude persist is stored in registers 0xF2 and 0xF3 of page 0.

Table 5-4. User Registers (Page 0 and Page 4) Register CRC registers

Register	Address, Page 0	Bit7-Bit0
CRC_1	0xF0	REGMAP_USER_INCLUDE_PERSIST_CRC16_LOW
CRC_2	0xF1	REGMAP_USER_INCLUDE_PERSIST_CRC16_HIGH
CRC_3	0xF2	REGMAP_USER_EXCLUDE_PERSIST_CRC16_LOW
CRC_4	0xF3	REGMAP_USER_EXCLUDE_PERSIST_CRC16_HIGH

The CRC calculation shown in the following code-snippet is taken from the Scalable PMIC GUI (see Reference 3). Appendix C is provided as an array of the register information. The information associated with the watchdog is mapped to the appropriate locations and registers are populated with default settings if not backed by NVM and with 0x00, if undefined.

```
const CRC_POLYNOMIAL = 0x755b;
var calculate_register_crc = function(registers, regmap_json, include_persist, page) {
    crc = 0xffff;
    for (var address = 0; address < 0x100; address++) {
        var data;
        var json_address;
        if (address >= 0xf0 && page === 0) {
            // Watchdog registers is mapped from 0x0fX in array to 0x40X in register map
            // Offset by 1 since 0x0f0 is mapped to 0x401
            json_address = address + 0x310 + 1;
        } else {
            json_address = address + (page * 0x100);
        }
        if (registers[address] === undefined) {
            if (json_address in regmap_json) {
                // Non-NVM register uses reset value
                data = regmap_json[json_address].reset;
            } else {
                // Register does not exist, so reads 0s
                data = 0;
            }
        } else {
            // Regsiter is NVM-backed, use value from device
            data = registers[address];
        }
        var crc_mask = 0x00;
        if (json_address in regmap_json) {
            if (include_persist) {
                crc_mask = regmap_json[json_address].crc_mask_include_persist;
            } else {
                crc_mask = regmap_json[json_address].crc_mask_exclude_persist;
            }
        }
        data = data & crc_mask;
        crc = crc_d8(crc, data);
    }
    return crc;
};
// Calculate CRC based on look-up table
var crc_d8 = function(crc, data) {
    var table_index = data ^ ((crc & 0xff00) >> 8);
    crc = (crc << 8) ^ lookup_table[table_index];
    return crc & 0xffff;
};
// Compute lookup-table used for CRC calculation
crc = 0x8000;
for (var i = 1; i < 256; i <<= 1) {
    if ((crc & 0x8000) !== 0) {
        crc = ((crc << 1) ^ CRC_POLYNOMIAL) & 0xffff;
    } else {
        crc = (crc << 1) & 0xffff;
    }
    for (var j = 0; j < i; j++) {
        lookup_table[i+j] = crc ^ lookup_table[j];
    }
}
```

6 NVM Validation

After the NVM has been updated and the PMIC power cycled, the NVM contents can be validated by simply reading out the user register map. Consideration is required if the PFSM instructions overwrite register settings during power up.

Alternatively, the NVM can be unlocked, the PFSM halted, and the NVM content directly transferred to the user registers. Once the NVM content is transferred to the user registers the values are again accessible through the serial interface. This alternative method is described in Table 6-1.

Note

If the NVM is locked, then the alternative mode described in Table 6-1 is not possible. Only user registers, Page 0 and Page 4, can be verified.

Table 6-1. NVM Validation Example

Instruction	I2C Address (Page)	Read/ Write	Register Address	Data	Description
1	0x28 (0)	Write	0xA2	0x00	Reset unlock logic
2	0x28 (0)	Write	0xA2	0x98	NVM Unlock Sequence
3	0x28 (0)	Write	0xA2	0xB8	
4	0x28 (0)	Write	0xA2	0x13	
5	0x28 (0)	Write	0xA2	0x7D	
6	0x28 (0)	Read	0xA3	0xC0	Confirm that the NVM was successfully unlocked; bit 6 is set
7	0x28 (0)	Write	0xA3	0xC1	Halt the PFSM
8	0x29 (1)	Write	0xEF	0x01	Transfer configuration from NVM to user registers
9	0x29 (1)	Read	0xF3	0x04	Confirm that the transfer is complete; bit 1 is cleared. Bit 2 is a don't care.
10	0x29 (1)	Read	0x23	0x12	Determine the I2C2 address to read from
11	0x28 (0)	Read	0x01-0xFF	Array	Read content from page 0. See Appendix A for valid address ranges.
12	0x29 (1)	Read	0x00-0xFF	Array	Read content from page 1. See Appendix A for valid address ranges.
13	0x12 (4)	Read	0x00-0x0A	Array	Read content from page 4. See Appendix A for valid address ranges.
14	0x28 (0)	Write	0xA4	0x00	Set PFSM to sub-page 0
15	0x2B (3)	Read	0x00-0xFF	Array	Read content from page 3, sub-page 0
16	0x28 (0)	Write	0xA4	0x01	Set PFSM to sub-page 1
17	0x2B (3)	Read	0x00-0xFF	Array	Read content from page 3, sub-page 1
18	0x28 (0)	Write	0xA4	0x02	Set PFSM to sub-page 2
19	0x2B (3)	Read	0x00-0xFF	Array	Read content from page 3, sub-page 2
20	0x28 (0)	Write	0xA4	0x00	Set PFSM to sub-page 0
21	0x28 (0)	Write	0xA2	0x00	Reset unlock logic

7 References

1. Texas Instruments, *TPS6594-Q1 Power Management IC (PMIC) with 5 Bucks and 4 LDOs for Safety-Relevant Automotive Applications* data sheet
2. Texas Instruments, *LP8764-Q1 Four-Phase, 20-A Buck Converter With Integrated Switches* data sheet
3. Texas Instruments, *Scalable PMIC's GUI* user's guide
4. Texas Instruments, *Dual TPS6594-Q1 PMIC User Guide for Jacinto 7 DRA829 and TDA4VM Automotive PDN-0B* user's guide
5. Texas Instruments, *TPS6594x-Q1 Evaluation Module* user's guide
6. Texas Instruments, *TPS6594-Q1 Schematic PCB Checklist (Rev. A)* application note

A Registers Backed by NVM

While the NVM space is continuous, write to only the registers described in Table A-1 for NVM updates.

Table A-1. TPS6594-Q1 and TPS6593-Q1 NVM Backed Registers

Page 0	Page 1[4]	Page3[2]	Page 4
0x04-0x20	0x01	0x00-0xFF	0x05
0x23-0x2C	0x03- 0x0A	0x00-0xFF	0x09
0x31-0x3E	0x0C- 0x14	0x00-0xFF	
0x41-0x54	0x16- 0x28[1]		
0x56-0x59	0x33 - 0x35		
0x78-0x7E	0x3F - 0x40[5]		
0x84	0x42 - 0x43		
0x87-0x88	0x45 - 0x4A		
0x8A- 0x8E			
0x92			
0x9B			
0xA7			
0xC3			
0xCD-0xD1			
0xF0 - 0xFB[3]			

(1) When the serial interface is changed, the recommendation is to handle page 1 addresses 0x18, 0x22, and 0x23 immediately after the NVM is unlocked.
(2) Page 3 sub-page address control is described in PFSM.
(3) The content stored in address 0xF0 through 0xFB is related to the register CRC and is described in Updating the Register CRC.
(4) Page 1 registers 0x00, 0x02, 0x0B, 0x15, 0xA8, 0xB1, 0xBA, 0xC3, and 0xCC are not to be changed from the factory settings.
(5) Page 1 register 0x41 is described in Section 5.4

Table A-2. LP876x-Q1 NVM Backed Registers

Page 0	Page 1[4]	Page 3[2]	Page 4
0x04-0x0B	0x01	0x00-0xFF	0x05
0x0E-0x15	0x03-0x0A	0x00-0xFF	0x09
0x18-0x1B	0x0C-0x14		
0x2B-0x3A	0x16-0x28[1]		
0x3C-0x3E	0x33-0x37		
0x41	0x3F-0x40[5]		
0x43-0x4A	0x42-0x43		
0x4E-0x54			
0x56-0x59			
0x78			
0x7B-0x7D			
0x84			
0x87-0x88			
0x8A-0x8B			
0x8E			

Table A-2. LP876x-Q1 NVM Backed Registers (continued)

Page 0	Page 1(4)	Page 3(2)	Page 4
0x92			
0xA7-0xA8			
0xC3			
0xCD-0xD0			
0xF0-0xF9(3)			

(1) When the serial interface is changed, the recommendation is to handle page 1 addresses 0x18, 0x22, and 0x23 immediately after the NVM is unlocked.
(2) Page 3 sub-page address control is described in PFSM.
(3) The content stored in address 0xF0 through 0xF9 is related to the register CRC and is described in Updating the Register CRC.
(4) Page 1 registers 0x00, 0x02, 0x0B, 0x15, 0xA8, 0xB1, 0xBA, and 0xC3 are not to be changed from the factory settings.
(5) Page 1 register 0x41 is described in Section 5.4

B Non-NVM Registers Which are Part of the Register CRC

The registers in Table B-1 and Table B-2 must be restored to their default values before the register CRC is updated. This process is only applicable when the register CRC is enabled.

Table B-1. TPS6594 and TPS6593 Registers Included Register CRC and Default Values

Page 0		Page 4	
Address	**Default**	**Address**	**Default**
0x22	0x00	0x01	0x00
0x80	0x00	0x03	0x7F
0x81	0x00	0x04	0x7F
0x82	0x08(1)	0x07	0x0A
0x85	0x00		
0x86	0x00		
0x8F	0x00		
0x90	0x00		
0x91	0x00		
0x92	0x00		
0x93	0x00		
0x94	0x00		
0x96	0x00		
0x98	0x00		
0x99	0x00		
0x9A	0x00		
0x9B	0x00		
0x9C	0x00		
0x9D	0x00		
0x9E	0x00		
0x9F	0x00		
0xAB	0x00		
0xC9	0x00		
0xCA	0x00		
0xCB	0x00		
0xCC	0x00		

(1) This register is read-only from the serial interface. If the value of the register does not match the default value then the Register CRC for Page 0 and Page 4 must be calculated externally and the results programmed into the PMIC.

Table B-2. LP876x Registers Included Register CRC and Default Values

Page 0		Page 4	
Address	**Default**	**Register**	**Default**
0x80	0x00	0x01	0x00
0x81	0x00	0x03	0x7F
0x82	0x08(1)	0x04	0x7F
0x85	0x00	0x07	0x0A
0x86	0x00		
0x8F	0x00		
0x90	0x00		
0x91	0x00		
0x92	0x00		
0x93	0x00		
0x94	0x00		
0x96	0x00		
0xAB	0x00		
0xC9	0x00		
0xCA	0x00		
0xCB	0x00		
0xCC	0x00		

(1) This register is read-only from the serial interface. If the value of the register does not match the default value then the Register CRC for Page 0 and Page 4 must be calculated externally and the results programmed into the PMIC.

C CRC for User Registers, Page 0 and Page 4

Table C-1 includes the data or data references and the appropriate masks as described in Section 5.5.

Table C-1. Data and Masks for User Registers CRC calculation

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0x00	0x00	0x00	NA(3)	NA(3)	NA(3)	NA(3)
0x01(1)	Read from Device	Read from Device	0xFF	0xFF	0xFF	0xFF
0x02(1)	Read from Device	Read from Device	0xFF	0xFF	0xFF	0xFF
0x03(1)	Read from Device	Read from Device	0xFF	0xFF	0xFF	0xFF
0x04	NVM(2)	NVM	0xFF	0xBF	0xFF	0xBF
0x05	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x06	NVM	NVM	0xFF	0xBB	0xFF	0xBB
0x07	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x08	NVM	NVM	0xFF	0xBF	0xFF	0xBF
0x09	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x0A	NVM	NVM	0xFF	0xBB	0xFF	0xBB
0x0B	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x0C	NVM	0x00	0xFF	0xBB	NA	NA
0x0D	NVM	0x00	0xFF	0x3F	NA	NA
0x0E	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x0F	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x10	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x11	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x12	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x13	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x14	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x15	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x16	NVM	0x00	0xFF	0xFF	NA	NA
0x17	NVM	0x00	0xFF	0xFF	NA	NA
0x18	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x19	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x1A	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x1B	NVM	NVM	0xFF	0x3F	0xFF	0x3F
0x1C	NVM	0x00	0xFF	0x3F	NA	NA
0x1D	NVM	0x00	0xFF	0xF3	NA	NA
0x1E	NVM	0x00	0xFF	0xF3	NA	NA
0x1F	NVM	0x00	0xFF	0xF3	NA	NA
0x20	NVM	0x00	0xFF	0xF3	NA	NA
0x21	0x00	0x00	0x00	0x00	0x00	0x00
0x22	0x00	0x00	0xFF	0x01	NA	NA
0x23	NVM	0x00	0xFF	0xFE	NA	NA
0x24	NVM	0x00	0xFF	0xFE	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0x25	NVM	0x00	0xFF	0xFE	NA	NA
0x26	NVM	0x00	0xFF	0x7F	NA	NA
0x27	NVM	0x00	0xFF	0x3F	NA	NA
0x28	NVM	0x00	0xFF	0x3F	NA	NA
0x29	NVM	0x00	0xFF	0x3F	NA	NA
0x2A	NVM	0x00	0xFF	0x3F	NA	NA
0x2B	NVM	NVM	0xFF	0x21	0xFF	0xFF
0x2C	NVM	NVM	0xFF	0x7F	0xFF	0x7F
0x2D	0x00	NVM	NA(3)	NA(3)	0xFF	0x7F
0x2E	0x00	NVM	NA	NA	0xFF	0xFF
0x2F	0x00	NVM	NA	NA	0xFF	0x7F
0x30	0x00	NVM	NA	NA	0xFF	0xFF
0x31	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x32	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x33	NVM	NVM	0xFF	0x10	0xFF	0xFF
0x34	NVM	NVM	0xFF	0x10	0xFF	0xFF
0x35	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x36	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x37	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x38	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x39	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x3A	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x3B	NVM	0x00	0xFF	0xFF	NA	NA
0x3C	NVM	NVM	0xFF	0x11	0xFF	0x00
0x3D	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x3E	NVM	NVM	0xFF	0x07	0xFF	0x03
0x3F	0x00	0x00	NA	NA	NA	NA
0x40	0x00	0x00	NA	NA	NA	NA
0x41	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x42	NVM	0x00	0xFF	0xFF	NA	NA
0x43	NVM	NVM	0xFF	0x0F	0xFF	0xFC
0x44	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x45	NVM	NVM	0xFF	0x03	0xFF	0x03
0x46	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x47	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x48	NVM	NVM	0xFF	0x3F	0xFF	0x0F
0x49	NVM	NVM	0xFF	0xBB	0xFF	0xBB
0x4A	NVM	NVM	0xFF	0xBB	0xFF	0xBB
0x4B	NVM	0x00	0xFF	0x0B	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0x4C	NVM	0x00	0xFF	0xBB	NA	NA
0x4D	NVM	0x00	0xFF	0xBB	NA	NA
0x4E	NVM	NVM	0xFF	0x03	0xFF	0x6F
0x4F	NVM	NVM	0xFF	0xF3	0xFF	0xFF
0x50	NVM	NVM	0xFF	0xF3	0xFF	0xFF
0x51	NVM	NVM	0xFF	0x3F	0xFF	0x1B
0x52	NVM	NVM	0xFF	0x20	0xFF	0x00
0x53	NVM	NVM	0xFF	0x0B	0xFF	0x0B
0x54	NVM	NVM	0xFF	0xD6	0xFF	0xD6
0x55	0x00	0x00	NA	NA	NA	NA
0x56	NVM	NVM	0xFF	0x0F	0xFF	0x8F
0x57	NVM	NVM	0xFF	0xAB	0xFF	0xAB
0x58	NVM	NVM	0xFF	0x09	0xFF	0x09
0x59	NVM	NVM	0xFF	0x3F	0xFF	0x38
0x5A	0x00	0x00	NA	NA	NA	NA
0x5B	0x00	0x00	NA	NA	NA	NA
0x5C	0x00	0x00	NA	NA	NA	NA
0x5D	0x00	0x00	NA	NA	NA	NA
0x5E	0x00	0x00	NA	NA	NA	NA
0x5F	0x00	0x00	NA	NA	NA	NA
0x60	0x00	0x00	NA	NA	NA	NA
0x61	0x00	0x00	NA	NA	NA	NA
0x62	0x00	0x00	NA	NA	NA	NA
0x63	0x00	0x00	NA	NA	NA	NA
0x64	0x00	0x00	NA	NA	NA	NA
0x65	0x00	0x00	NA	NA	NA	NA
0x66	0x00	0x00	NA	NA	NA	NA
0x67	0x00	0x00	NA	NA	NA	NA
0x68	0x00	0x00	NA	NA	NA	NA
0x69	0x00	0x00	NA	NA	NA	NA
0x6A	0x00	0x00	NA	NA	NA	NA
0x6B	0x00	0x00	NA	NA	NA	NA
0x6C	0x00	0x00	NA	NA	NA	NA
0x6D	0x00	0x00	NA	NA	NA	NA
0x6E	0x00	0x00	NA	NA	NA	NA
0x6F	0x00	0x00	NA	NA	NA	NA
0x70	0x00	0x00	NA	NA	NA	NA
0x71	0x00	0x00	NA	NA	NA	NA
0x72	0x00	0x00	NA	NA	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0x73	0x00	0x00	NA	NA	NA	NA
0x74	0x00	0x00	NA	NA	NA	NA
0x75	0x00	0x00	NA	NA	NA	NA
0x76	0x00	0x00	NA	NA	NA	NA
0x77	0x00	0x00	NA	NA	NA	NA
0x78	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0x79	NVM	0x00	0xFF	0x03	NA	NA
0x7A	NVM	0x00	0xFF	0xFF	NA	NA
0x7B	NVM	NVM	0xFF	0xF9	0xFF	0xFF
0x7C	NVM	NVM	0xFF	0x03	0xFF	0x03
0x7D	NVM	NVM	0xFF	0xFB	0xFF	0xFB
0x7E	NVM	0x00	0x7F	0x0F	NA	NA
0x7F	0x00	0x00	NA	NA	NA	NA
0x80	0x00	0x00	0xFF	0x01	0xFF	0x01
0x81	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x82	0x08	0x08	0xF8	0x18	0xF8	0x18
0x83	0x00	0x00	NA	NA	NA	NA
0x84	NVM	NVM	0xEF	0x0F	0xEF	0x0F
0x85	0x00	0x00	0xF0	0xF0	0xF0	0xF0
0x86	0x00	0x00	0xFF	0x03	0xFF	0x03
0x87	NVM	NVM	0xFF	0x1F	0xFF	0x0F
0x88	NVM	NVM	0xFF	0x07	0xFF	0x07
0x89	0x00	0x00	NA	NA	NA	NA
0x8A	NVM	NVM	0xFF	0x1F	0xFF	0xFF
0x8B	NVM	NVM	0xFF	0x1F	0xFF	0x1F
0x8C	NVM	0x00	0xFF	0xFF	NA	NA
0x8D	NVM	0x00	0xFF	0xFF	NA	NA
0x8E	NVM	NVM	0xFF	0x0F	0xFF	0x0F
0x8F	0x00	0x00	0xFF	0x01	0xFF	0x01
0x90	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x91	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x92(4)	0x00	0x00	0xFF	0xEF	0xFF	0xEF
0x93	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x94	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x95	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x96	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x97	0x00	0x00	NA	NA	NA	NA
0x98	0x00	0x00	0xFF	0x01	NA	NA
0x99	0x00	0x00	0xFF	0xFF	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0x9A	0x00	0x00	0xFF	0xFF	NA	NA
0x9B(4)	0x00	0x00	0xFF	0xEF	NA	NA
0x9C	0x00	0x00	0xFF	0xFF	NA	NA
0x9D	0x00	0x00	0xFF	0xFF	NA	NA
0x9E	0x00	0x00	0xFF	0xFF	NA	NA
0x9F	0x00	0x00	0xFF	0xFF	NA	NA
0xA0	0x00	0x00	NA	NA	NA	NA
0xA1	0x00	0x00	NA	NA	NA	NA
0xA2	0x00	0x00	NA	NA	NA	NA
0xA3	0x00	0x00	NA	NA	NA	NA
0xA4	0x00	0x00	NA	NA	NA	NA
0xA5	0x00	0x00	NA	NA	NA	NA
0xA6	0x00	0x00	NA	NA	NA	NA
0xA7	NVM	NVM	0xFF	0xFF	0xFF	0xFF
0xA8	0x00	NVM	NA	NA	0xFF	0x3F
0xA9	0x00	0x00	NA	NA	NA	NA
0xAA	0x00	0x00	NA	NA	NA	NA
0xAB	0x00	0x00	NA	NA	NA	NA
0xAC	0x00	0x00	NA	NA	NA	NA
0xAD	0x00	0x00	NA	NA	NA	NA
0xAE	0x00	0x00	NA	NA	NA	NA
0xAF	0x00	0x00	NA	NA	NA	NA
0xB0	0x00	0x00	NA	NA	NA	NA
0xB1	0x00	0x00	NA	NA	NA	NA
0xB2	0x00	0x00	NA	NA	NA	NA
0xB3	0x00	0x00	NA	NA	NA	NA
0xB4	0x00	0x00	NA	NA	NA	NA
0xB5	0x00	0x00	NA	NA	NA	NA
0xB6	0x00	0x00	NA	NA	NA	NA
0xB7	0x00	0x00	NA	NA	NA	NA
0xB8	0x00	0x00	NA	NA	NA	NA
0xB9	0x00	0x00	NA	NA	NA	NA
0xBA	0x00	0x00	NA	NA	NA	NA
0xBB	0x00	0x00	NA	NA	NA	NA
0xBC	0x00	0x00	NA	NA	NA	NA
0xBD	0x00	0x00	NA	NA	NA	NA
0xBE	0x00	0x00	NA	NA	NA	NA
0xBF	0x00	0x00	NA	NA	NA	NA
0xC0	0x00	0x00	NA	NA	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0xC1	0x00	0x00	NA	NA	NA	NA
0xC2	0x00	0x00	NA	NA	NA	NA
0xC3(4)	NVM	NVM	0xFF	0x00	0xFF	0x00
0xC4	0x00	0x00	NA	NA	NA	NA
0xC5	0x00	0x00	NA	NA	NA	NA
0xC6	0x00	0x00	NA	NA	NA	NA
0xC7	0x00	0x00	NA	NA	NA	NA
0xC8	0x00	0x00	NA	NA	NA	NA
0xC9	0x00	0x00	0xFF	0x00	0xFF	0x00
0xCA	0x00	0x00	0xFF	0x00	0xFF	0x00
0xCB	0x00	0x00	0xFF	0x00	0xFF	0x00
0xCC	0x00	0x00	0xFF	0x00	0xFF	0x00
0xCD	NVM	NVM	0xFF	0x00	0xFF	0x00
0xCE	NVM	NVM	0xFF	0x00	0xFF	0x00
0xCF	NVM	NVM	0xFF	0x00	0xFF	0x00
0xD0	NVM	NVM	0xFF	0x00	0xFF	0x00
0xD1	NVM	0x00	0xFF	0x00	NA	NA
0xD2	0x00	0x00	NA	NA	NA	NA
0xD3	0x00	0x00	NA	NA	NA	NA
0xD4	0x00	0x00	NA	NA	NA	NA
0xD5	0x00	0x00	NA	NA	NA	NA
0xD6	0x00	0x00	NA	NA	NA	NA
0xD7	0x00	0x00	NA	NA	NA	NA
0xD8	0x00	0x00	NA	NA	NA	NA
0xD9	0x00	0x00	NA	NA	NA	NA
0xDA	0x00	0x00	NA	NA	NA	NA
0xDB	0x00	0x00	NA	NA	NA	NA
0xDC	0x00	0x00	NA	NA	NA	NA
0xDD	0x00	0x00	NA	NA	NA	NA
0xDE	0x00	0x00	NA	NA	NA	NA
0xDF	0x00	0x00	NA	NA	NA	NA
0xE0	0x00	0x00	NA	NA	NA	NA
0xE1	0x00	0x00	NA	NA	NA	NA
0xE2	0x00	0x00	NA	NA	NA	NA
0xE3	0x00	0x00	NA	NA	NA	NA
0xE4	0x00	0x00	NA	NA	NA	NA
0xE5	0x00	0x00	NA	NA	NA	NA
0xE6	0x00	0x00	NA	NA	NA	NA
0xE7	0x00	0x00	NA	NA	NA	NA

Table C-1. Data and Masks for User Registers CRC calculation (continued)

Address	TP659x-Q1 Data	LP876x-Q1 Data	TPS659x-Q1 Include Mask	TPS659x-Q1 Exclude Mask	LP876x-Q1 Include Mask	LP876x-Q1 Exclude Mask
0xE8	0x00	0x00	NA	NA	NA	NA
0xE9	0x00	0x00	NA	NA	NA	NA
0xEA	0x00	0x00	NA	NA	NA	NA
0xEB	0x00	0x00	NA	NA	NA	NA
0xEC	0x00	0x00	NA	NA	NA	NA
0xED	0x00	0x00	NA	NA	NA	NA
0xEE	0x00	0x00	NA	NA	NA	NA
0xEF	0x00	0x00	NA	NA	NA	NA
0xF0	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0xF1	0x00	0x00	NA	NA	NA	NA
0xF2	0x7F	0x7F	0xFF	0x7F	0xFF	0x7F
0xF3	0x7F	0x7F	0xFF	0x7F	0xFF	0x7F
0xF4	Page 4, Register 0x05(5)	NVM Page 4, Register 0x05(5)	0xFF	0xFF	0xFF	0xFF
0xF5	0x00	0x00	NA	NA	NA	NA
0xF6	0x0A	0x0A	0xFF	0xFF	0xFF	0xFF
0xF7	0x00	0x00	NA	NA	NA	NA
0xF8	Page 4, Register 0x09(6)	NVM Page 4, Register 0x09(6)	0xFF	0xFF	0xFF	0xFF
0xF9	0x00	0x00	NA	NA	NA	NA
0xFA	0x00	0x00	NA	NA	NA	NA
0xFB	0x00	0x00	NA	NA	NA	NA
0xFC	0x00	0x00	NA	NA	NA	NA
0xFD	0x00	0x00	NA	NA	NA	NA
0xFE	0x00	0x00	NA	NA	NA	NA
0xFF	0x00	0x00	NA	NA	NA	NA

(1) Register addresses 0x01, 0x02, and 0x03 of Page 0 (DEV_REV, NVM_CODE_1, and NVM_CODE_2) are programmed in the factory and cannot be changed. These values are part of the register CRC calculation and must be read from the device.
(2) NVM means that this register is backed by NVM and the value is be read from the .BIN file generated from the GUI (see Reference 3).
(3) NA means that the CRC mask is not applicable. In the algorithm the value used to update the CRC is the logical 'AND' of the data and the mask. If either the data or the mask is '0x00' then the result is 0x00. From the algorithm this can be interpreted as an undefined register.
(4) Register addresses 0x92, 0x9B, and 0xC3 are a mix of bits which are backed by NVM and bits which are not backed by NVM. The NVM backed bits, ESM_MCU_EN and ESM_SOC_EN, found in registers 0x92 and 0x9B, respectively, are typically '0'. Enabling the ESM immediately upon power up is not recommended. FIRST_STARTUP_DONE, found in register 0xC3 is not backed by NVM and is '0' by default.
(5) Page 4, Register 0x05 is the WD_LONGWIN register. This value is taken from address 0x405 found in the GUI generated .BIN file.
(6) Page 4, Register 0x09 is the WD_THR_CFG register. This value is taken from address 0x405 found in the GUI generated .BIN file. Only bit 6, WD_EN, is backed by NVM but the other bits are set by default. Therefore this value is either 0xFF or 0xBF.

D Example With I2C Serial Interface

The TPS6594-Q1 example in Table D-1 is taken from the Scalable PMIC GUI implementation. In this example, the initial I2C address of the target device is 0x48 and the updated I2C address is 0x28.

Table D-1. I2C Example With Updated I2C1 Address

Instruction	I2C Address (Page)	Read/Write	Register Address	Data	Description
1	0x48 (0)	Write	0xA2	0x00	Reset unlock logic
2	0x48 (0)	Write	0xA2	0x98	NVM unlock
3	0x48 (0)	Write	0xA2	0xB8	
4	0x48 (0)	Write	0xA2	0x13	
5	0x48 (0)	Write	0xA2	0x7D	
6	0x48 (0)	Read	0xA3	0xC0	Confirm that the NVM was successfully unlocked; bit 6 is set.
7	0x48 (0)	Write	0xA3	0xC1	Halt the PFSM
8	0x49 (1)	Write	0x18	0x0D	This instruction unlocks the frequency selection so that the BUCK frequency (register 0x8A of page 0) can be changed along with other updates to page 0. This register is set to the appropriate application value when page 1 is updated.
9	0x48 (0)	Write	0x31	0x20	Update GPIO1, GPIO2, and GPIO3 (LP876x) to the desired final serial interface settings.
10	0x48 (0)	Write	0x32	0x40	
11	0x48 (0)	Write	0x33	0x10	
12	0x49 (1)	Write	0x22	0x28	Update I2C1 address to 0x28. All of the following register accesses are based upon this address.
13	0x29 (1)	Write	0x23	0x12	Update the I2C2 address to 0x12
14	0x29 (1)	Write	0x1A	0x00	Update the Serial Interface mode. At this point if the serial interface was changed to SPI or the CRC enabled, then the associated changes must be made before proceeding to the next instruction.
15	0x28 (0)	Write	0x04-0xD1	Array	Write content to page 0 register map based upon Table A-1. Return registers found in Table B-1 to default values.
16	0x29 (1)	Write	0x01-0x43	Array	Write content to page 1 register map based upon Table A-1
17	0x12 (4)	Write	0x05,0x09	Array	Write contents to page 4
18	0x28 (0)	Write	0xA4	0x00	Set PFSM control to sub-page 0
19	0x2B (3)	Write	0x00-0xFF	Array	Write content to page 3 sub-page 0
20	0x28 (0)	Write	0xA4	0x01	Set PFSM control to sub-page 1
21	0x2B (3)	Write	0x00-0xFF	Array	Write content to page 3 sub-page 1
22	0x28 (0)	Write	0xA4	0x02	Set PFSM control to sub-page 2
23	0x2B (3)	Write	0x00-0xFF	Array	Write content to page 3 sub-page 2
24	0x28 (0)	Write	0xA4	0x00	Set PFSM control to sub-page 0
25	0x28 (0)	Write	0xF0-0xFB	0x00	Clear register CRC content
26	0x28 (0)	Write	0xEF	0x02	Run CRC BIST and update the register CRC values
27	0x28 (0)	Read	0xFB	Non-zero value	This is a simple check to see if the crc is complete. The check is simply looking to see if the value was updated from 0x00 (cleared in step 24).
28	0x28(0)	Write	0xF0-0xF3	Array	Compute the 16-bit User register CRC values and update the REGMAP_USER_INCLUDE_PERSIST_CRC16 and REGMAP_USER_INCLUDE_EXCLUDE_CRC16 registers.
29	0x29 (1)	Write	0xE1	0x00	Prepare NVM to receive update from register map. If the desire is to lock the EEPROM so that future updates cannot be made, then before the transfer command, write any value other than 0xA5 to address 0x141: Page 1 register address 0x41.
30	0x29 (1)	Write	0xEF	0x02	Start the transfer from the register map to the NVM.
31	0x29 (1)	Read	0xF3	0x04	Poll bit 1. When bit 1 is cleared, the transfer is complete. Bit 2 can be either '0' or '1'.
32	0x28 (0)	Write	0xA2	0x00	Reset unlock logic

E Revision History

Changes from Revision * (October 2022) to Revision A (April 2023)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1
• Added *Caution* note to highlight difference between LP8764-Q1 and other devices	6
• Added caution to highlight difference between LP8764-Q1 and other devices	9
• Changed to clarify what is locked when updating register 0x141	9
• Deleted addresses from table that are to remain unchanged	14
• Changed start address for page 1 update	25

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265