User's Guide

TPS65941310-Q1 and TPS65941111-Q1 PMIC User's Guide for Jacinto™ 7 J721E, PDN-0A

TEXAS INSTRUMENTS

ABSTRACT

This user's guide can be used as a guide for integrating the TPS6594-Q1 power management integrated circuit (PMIC) into a system powering the Automotive Jacinto 7 DRA829 or TDA4VM processor.

Table of Contents

Trademarks

Jacinto™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

This user’s guide describes a power distribution network (PDN), PDN-0A, using two TPS6594-Q1 devices to supply either the DRA829V or the TDA4VM processor with independent MCU and Main power rails. DRA829/ TDA4VM Dual PMIC PDN-0C enables board level isolation of the MCU safety island and main voltage resources as required for implementing two desirable features of the processor:

1. MCU processor acts as independent safety monitor (MCU Safety Island) over the Main processing resources to ensure safe system operations.
2. MCU processor maintains minimum system operations (MCU Only) to significantly reduce processor power dissipation thereby extending battery life during stand-by use cases and reducing component temperature.

The following topics are described to clarify platform system operation:

1. PDN power resource connections
2. PDN digital control connections
3. Primary and secondary PMIC static NVM contents
4. PMIC sequencing settings to support different PDN power state transitions for an advanced processor system

PMIC and processor data manuals provide recommended operating conditions, electrical characteristics, recommended external components, package details, register maps, and overall component functionality. In the event of any inconsistency between any user's guide, application report, or other referenced material, the data sheet specification is the definitive source.

2 Device Versions

There are different orderable part numbers (OPNs) of the TPS6594-Q1 device available with unique NVM settings to support different end product use cases and processor types. The unique NVM settings for each PMIC device are optimized per PDN design to support different processors, processing loads, SDRAM types, system functional safety levels, and end product features (such as low power modes, processor voltages, and memory subsystems). The NVM settings can be identified by both NVM_ID and NVM_REV registers. Each PMIC device is distinguished by the part number, NVM_ID, and NVM_REV values listed in Table 2-1.

Table 2-1. Dual TPS6594-Q1 Orderable Part Numbers for Independent MCU and Main PDN System

PDN USE CASE	PDN	Orderable Part Number	TI_NVM_ID (TI_NVM_REV)	Orderable Part Number	TI_NVM_ID (TI_NVM_REV)	Error Signal Monitoring
• Up to 6 A[1] on the Primary PMIC 2-phase CPU rail • Up to 12 A[1] on the Secondary PMIC 4-phase CORE rail • Up to 3.4 A[1] on the SDRAM, with support for LPDDR4 • Supports Processor 2 GHz maximum clock with high-speed SERDES operations • Supports 32 Gb of LPDDR4 SDRAM with 4266MTs data rate • Supports Functional Safety up to ASIL-D level with MCU Safety Island • Supports MCU-only and Retention[2] low power modes • Supports I/O level of 3.3 V or 1.8 V • Supports optional end product features: – Compliant high-speed SD Card memory – Compliant USB 2.0 Interface – On-board Efuse programming of high security processors	0A	TPS65941310 RWERQ1	0x10 (0x04)	TPS65941111 RWERQ1	0x11 (0x03)	Combined MCU and SOC

[1] TI recommends having 15% margin between the maximum expected load current and the maximum current allowed per each PMIC output rail.

[2] Retention, either GPIO or RAM, is configured by the processor.

3 Processor Connections

This section details how the dual TPS6594-Q1 power resources and GPIO signals are connected to the processor and other peripheral components.

3.1 Power Mapping

Figure 3-1 shows the power mapping between the dual TPS6594-Q1 PMIC power resources and processor voltage domains required to support independent MCU and Main power rails. In this configuration, both PMICs use a 3.3 V input voltage. For Functional Safety applications, there is a protection FET before VCCA that connects to the OVPGDRV pin of the primary PMIC, allowing voltage monitoring of the input supply to the PMICs.

For SD card dual-voltage I/O support (3.3 V and 1.8 V), LDO1 of the TPS65941111-Q1 device can be used. A processor GPIO control signal with a logic high default value is used to set SD VIO to 3.3 V initially. During processor power up, the boot loader SW can set GPIO signal low to select 1.8 V level as needed for high-speed card operation per SD specification. The GPIO allows control of the LDO1 voltage without the need for the MCU processor to establish I2C communication with the PMICs during boot from SD card operations.

This PDN uses three discrete power components. Two components are required and one is optional depending upon end product features. The two TPS22965-Q1 Load Switches connect VCCA_3V3 power rail to supply OV protected 3.3 V to processor I/O domains. Two load switches are required in order to enable isolation between MCU and Main processor sub-sections for MCU Safety Island or MCU Only low power operations. The last discrete power component is an optional TLV73318-Q1 LDO that can be used if an end product uses a high security processor type and desires the capability to program Efuse values on-board. If this feature is not desired, then this LDO can be omitted and processor pins terminated per data manual recommendations.

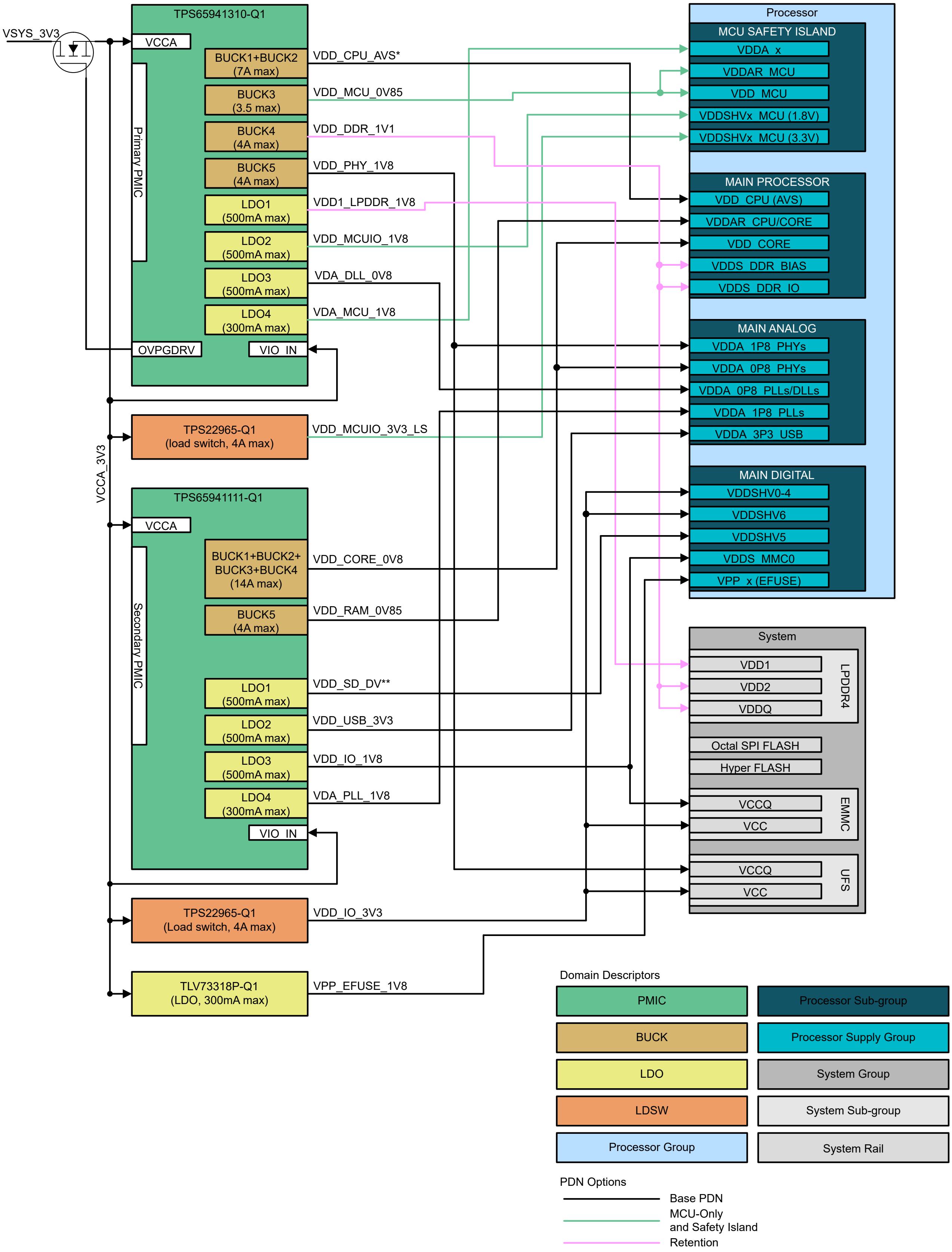


Figure 3-1. Power Connections

- * VDD_CPU_AVS, boot voltage of 0.8 V then software sets device specific AVS; 0.68 V – 0.72 V.
- ** VDD_SD_DV, 3.3 V then software changes to 1.8 V per HS-SD.

Table 3-1 identifies which power resources are required to support different system features. In the Active SoC column, there is an additional option for including or excluding the VPP_x(EFUSE) rail. LDO1 and LDO2 of TPS65941111, which support optional SD CARD and USB Interface features, are enabled as part of the power on sequence. Even if these System Features are not used, the regulators are energized as part of the power up sequence.

Table 3-1. PDN Power Mapping and System Features

Power Mapping				System Features[1]				
Device	**Power Resource**	**Power Rails**	**Processor and Memory Domains**	**Active SoC**	**MCU - only**	**DDR Retention**	**SD Card**	**USB Interface**
TPS65941310-Q1	BUCK12	VDD_CPU_AVS	VDD_CPU	R				
	BUCK3	VDD_MCU_0V85	VDDAR_MCU, VDD_MCU	R	R			
	BUCK4	VDD_DDR_1V1	VDDS_DDR_BIAS, VDDS_DDR_IO Mem: VDD2	R	O[3]	R[3]		
	BUCK5	VDD_PHY_1V8	VDDA_1P8_PHYs	R				
	LDO1	VDD1_DDR_1V8	Mem: VDD1	R	O[2]	R[2]		
	LDO2	VDD_MCUIO_1V8	VDDSHVx_MCU (1.8 V) Mem: VCC	R	R			
	LDO3	VDA_DLL_0V8	VDDA_0P8_PLLs/DLLs	R				
	LDO4	VDA_MCU_1V8	VDDA_x	R	R			
TPS65941111-Q1	BUCK1234	VDD_CORE_0V8	VDD_CORE, VDDA_0P8_PHYs	R				
	BUCK5	VDD_RAM_0V85	VDDAR_CPU/CORE	R				
	LDO1	VDD_SD_DV	VDDSHV5				R	
	LDO2	VDD_USB_3V3	VDDA_3P3_USB					R
	LDO3	VDD_IO_1V8	VDDS_MMC0 Mem: VCCQ	R				
	LDO4	VDA_PLL_1V8	VDDA_1P8_PLLs	R				
TPS22965-Q1	Load Switch	VDD_MCUIO_3V3	VDDSHVx_MCU (3.3 V)	R	R			
TPS22965-Q1	Load Switch	VDD_IO_3V3	VDDSHV0-4,VDDSHV6 (3.3 V)	R				
TLV73318P-Q1	LDO	VPP_EFUSE_1V8	VPP_x(EFUSE)	O				

[1] 'R' is required and 'O' is optional.
[2] LDO1 of the TPS65941310-Q1 remains on when TRIGGER_I2C_7, in FSM_I2C_TRIGGERS Register, is set.
[3] BUCK4 of the TPS65941310-Q1 remains on when TRIGGER_I2C_7, in FSM_I2C_TRIGGERS Register, is set.

3.2 Control Mapping

Figure 3-2 shows the digital control signal mapping between processor and PMIC devices. For the two PMIC devices to work together, the primary PMIC and secondary PMIC must establish an SPMI communication channel. This allows the two TPS6594-Q1 to synchronize their internal Pre-Configurable State Machines (PFSM) so that they operate as one PFSM across all power and digital resources. The GPIO_5 and GPIO_6 pins on the TPS6594-Q1 are assigned for this functionality. In addition, the LDOVINT pin of the primary PMIC is connected to the ENABLE pin of the secondary PMIC in order to correctly initiate the PFSM.

Other digital connections from the TPS6594-Q1 PMICs to the processor provide error monitoring, processor reset, processor wake up, and system low-power modes. Specific GPIO pins have been assigned to key signals in order to ensure proper operation during low power modes when only a few GPIO pins remain operational.

The digital connections shown in Figure 3-2 allow system features including 'MCU-only, MCU Safety Island' and DDR Retention modes, and compliant dual voltage SD card operation.

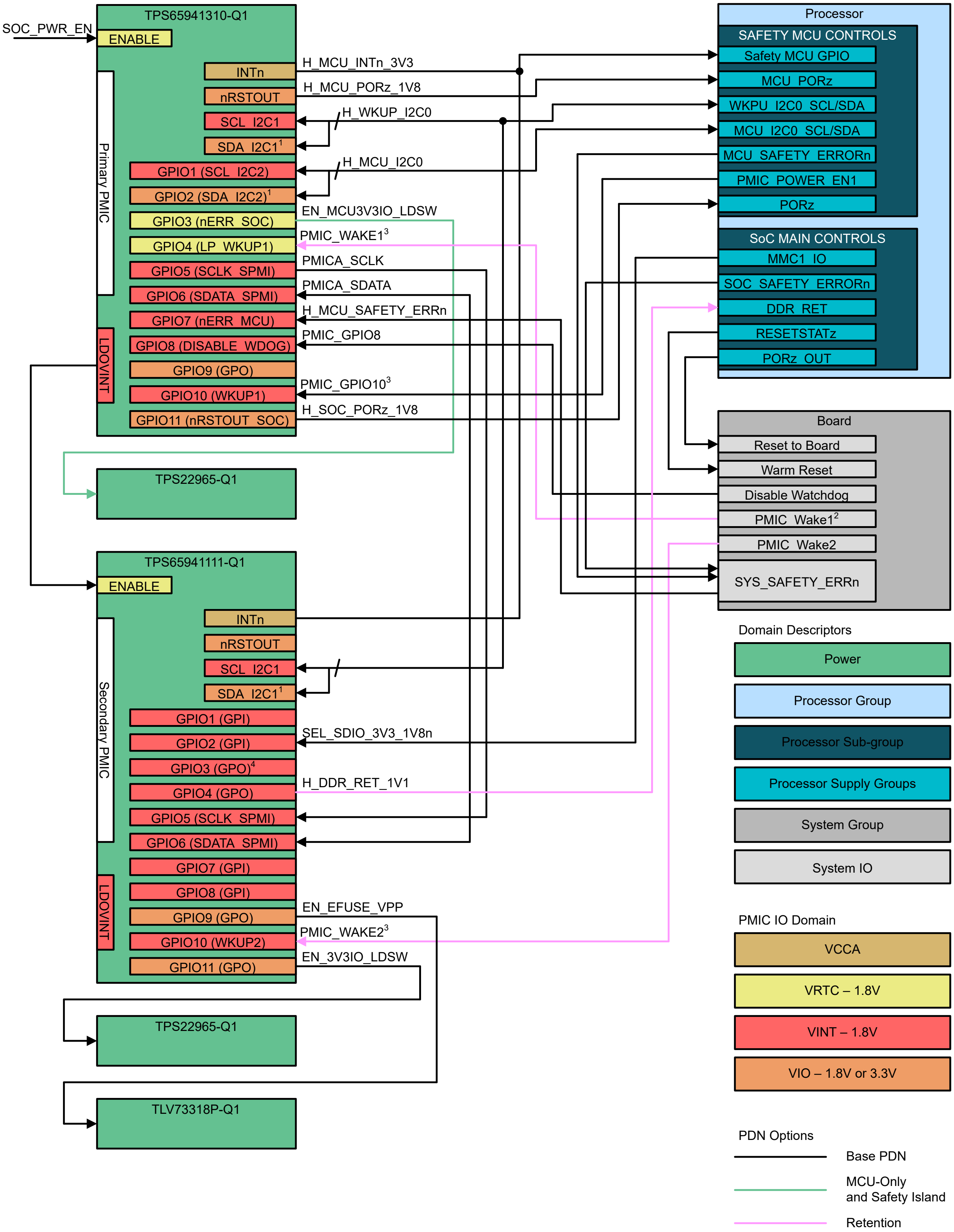


Figure 3-2. TPS6594-Q1 Digital Connections

1. PMIC IO can have distinct power domains for input and output functionality. The SDA function for I2C1 and I2C2 use the VINT voltage domain as an input and the VIO voltage domain as an output. Please refer to the

device datasheet for a complete description. The PMIC voltage domains indicated are for the PDN-0A NVM configuration.
2. PMIC_Wake1 is typically a CAN PHY INH output.
3. WKUP2 triggers a transition to MCU Only state. LP_WKUP1 and WKUP1 transition to the ACTIVE state. Figure 6-1
4. GPIO3 is part of the power sequencing but unused in this PDN.

Note

The PMIC voltage domain of an IO can be different depending upon configuration. When configured as an input GPIO3 and GPIO4 are in the VRTC domain. When configured as an output, GPIO3 and GPIO4 are in the VINT domain.

Note

In addition to the I2C signals, four additional signals are open-drain outputs and require a pullup to a specific power rail. Please refer to Table 3-2 for a list of the signals and the specific power rail.

Table 3-2. Open-drain signals and Power Rail

PDN Signal	Pullup Power Rail
H_MCU_INTn	VDD_MCUIO_3V3
H_MCU_PORz_1V8	VDA_MCU_1V8
H_SOC_PORz_1V8	VDA_MCU_1V8
H_DDR_RET_1V1	VDD_DDR_1V1_REG
H_WKUP_I2C0	VDD_MCUIO_3V3
H_MCU_I2C0_SCL/SDA	VDD_MCUIO_3V3

Please use Table 3-3 as a guide to understand GPIO assignments required for each PDN system feature. If the feature listed is not required, the digital connection can be removed; however, the GPIO pin is still configured per NVM defined default function shown. After the processor has booted up, the processor can reconfigure unused GPIOs to support new functions. The reconfiguration of unused GPIO is possible as long as that function is only needed after boot and default function does not cause any conflicts with normal operations (for example, two outputs driving same net). For details on how functional safety related connections help achieve functional safety system-level goals, see Section 4.

Table 3-3. Digital Connections by System Feature

Device	GPIO Mapping			System Features[1]				
	PMIC Pin	NVM Function	PDN Signals	Active SoC	Functional Safety	MCU - only MCU-Safety Island	DDR Retention	SD Card
TPS65941310-Q1	nPWRON/ ENABLE	Enable	SOC_PWR_ON	R				
	INT	INT	H_MCU_INTn		R			
	nRSTOUT	nRSTOUT	H_MCU_PORz_1V8	R		R		
	SCL_I2C1	SCL_I2C1	H_WKUP_I2C0	R				
	SDA_I2C1	SDA_I2C1	H_WKUP_I2C0	R				
	GPIO_1	SCL_I2C2	H_MCU_I2C0_SCL		R			
	GPIO_2	SDA_I2C2	H_MCU_I2C0_SDA		R			
	GPIO_3	GPO	EN_MCU3V3IO_LDSW	R		R		
	GPIO_4	LP_WKUP1[2]	PMIC_WAKE1				R	
	GPIO_5	SCLK_SPMI	LEOA_SCLK	R				
	GPIO_6	SDATA_SPMI	LEOA_SDATA	R				
	GPIO_7	nERR_MCU	H_MCU_SAFETY_ERRn		R			
	GPIO_8	DISABLE_WDOG	PMICA_GPIO8	[4]	[4]			
	GPIO_9	GPI						
	GPIO_10	WKUP1	PMICA_GPIO10/ H_PMIC_PWR_EN1	R				
	GPIO_11	nRSTOUT_SOC	H_SOC_PORz_1V8			R		
TPS65941111-Q1	nPWRON/ ENABLE	ENABLE	VINT_LEOA_1V8		R			
	nINT	nINT	H_MCU_INTn					
	nRSTOUT	nRSTOUT	Unused					
	SCL_I2C1	SCL_I2C1	H_WKUP_I2C0	R				
	SDA_I2C1	SCL_I2C1	H_WKUP_I2C0	R				
	GPIO_1	GPI	Unused[5]					
	GPIO_2	GPI	SEL_SDIO_3V3_1V8n[3]					R
	GPIO_3	GPO						
	GPIO_4[6]	GPO	H_DDR_RET_1V1				R	
	GPIO_5	SCLK_SPMI	LEOA_SCLK	R				
	GPIO_6	SDATA_SPMI	LEOA_SDATA	R				
	GPIO_7	GPI	Unused[5]					
	GPIO_8	GPI	Unused[5]					
	GPIO_9	GPO	EN_EFUSE_LDO[5]					
	GPIO_10	WKUP2	Unused[5]					
	GPIO_11	GPO	EN_3V3IO_LDSW	R				

[1] R is Required. O is optional.

[2] LP_WKUP1 function is masked in the static settings. Instructions for unmasking the function are provided in Section 7.1.3, Section 7.2 and Section 7.3.

(3) This pin is an input with an internal pulldown enabled. A rising edge on this GPI initiates the FSM trigger and associated sequence. The sequence configures LDO1 to bypass mode, supplying 3.3 V. A falling edge triggers an alternate sequence which configures LDO1 to LDO mode, supplying 1.8 V. See also Table 6-1

(4) If it is desired to disable the watchdog through hardware, GPIO_8 is required and must be set high by the time nRSTOUT goes high. After nRSTOUT is high, the watchdog state is latched and the pin can be configured for other functions through software.

(5) This GPIO is not required for power sequencing or PMIC functionality and can be configured by software for a different purpose if desired.

(6) GPIO4 of the TPS65941111 is not part of the power sequences, Section 6.3. This GPIO must be configured by the SOC at runtime.

4 Supporting Functional Safety Systems

By using the dual TPS6594-Q1 solution to power the DRA829V or TDA4VM processor, the system can leverage the following PMIC functional safety features:

- Independent Power Control of MCU and Main Rails
- Independent Monitoring and Reset for MCU and Main Rails
- Input Supply Monitoring
- Output Voltage and Current Monitoring
- Question and Answer Watchdog
- Fault Reporting Interrupts
- Enable Drive Pin that provides an independent path to disable system actuators
- Error Pin Monitoring
- Internal Diagnostics including voltage monitoring, temperature monitoring, and Built-In Self-Test

Refer to the Safety Manual of the TPS6594-Q1 device for full descriptions and analysis of the PMIC functional safety features. These functional safety features can assist in achieving up to ASIL-D rating for a system. Additionally, these features help in achieving the functional safety assumptions utilized by the processor to achieve up to ASIL-D rating. See the DRA829/TDA4VM Safety Manual for Jacinto™ 7 Processors for a complete list of functional safety system assumptions.

5 Static NVM Settings

The TPS6594-Q1 devices consist of user register space and an NVM. The settings in NVM, which are loaded into the user registers during the transition from INIT to BOOT BIST, are provided in this section. Note: The user registers can be changed during state transitions, such as moving from STANDBY to ACTIVE mode. The user register map is described in the TPS6594-Q1 datasheet.

5.1 Application-Based Configuration Settings

In the TPS6594-Q1 datasheet, there are seven application-based configurations for each BUCK to operate within. The following list includes the different configurations available:

- 2.2 MHz Single Phase for DDR Termination
- 4.4 MHz VOUT Less than 1.9 V, Multiphase or High COUT Single Phase
- 4.4 MHz VOUT Less than 1.9 V, Low COUT, Single Phase Only
- 4.4 MHz VOUT Greater than 1.7 V, Single Phase Only
- 2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase
- 2.2 MHz Full VOUT Range and VIN Greater than 4.5 V, Single Phase Only
- 2.2 MHz Full VOUT and Full VIN Range, Single Phase Only

The seven configurations also have optimal output inductance values that optimize the performance of each buck under these various conditions. Table 5-1 shows the default configurations for the BUCKs. These settings cannot be changed after device startup.

Table 5-1. Application Use Case Settings

Device	BUCK Rail	Default Application Use Case	Recommended Inductor Value
TPS65941310-Q1	BUCK1	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK2	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK3	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK4	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK5	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH

Table 5-1. Application Use Case Settings (continued)

Device	BUCK Rail	Default Application Use Case	Recommended Inductor Value
TPS65941111-Q1	BUCK1	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK2	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK3	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK4	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH
	BUCK5	2.2 MHz VOUT Less than 1.9 V Multiphase or Single Phase	470 nH

5.2 Device Identification Settings

These settings are used to distinguish which device is detected in a system. These settings cannot be changed after device startup.

Table 5-2. Device Identification NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
DEV_REV	DEVICE_ID	0x82		0x82	
NVM_CODE_1	TI_NVM_ID	0x10		0x11	
NVM_CODE_2	TI_NVM_REV	0x4		0x3	
PHASE_CONFIG	MP_CONFIG	0x2	2+1+1+1	0x0	4+1

5.3 BUCK Settings

These settings detail the voltages, configurations, and monitoring of the BUCK rails stored in the NVM. All these settings can be changed though I^{2}C after startup. Some settings, typically the enable bits, are also changed by the PFSM, as described in Section 6.3.

After the Section 6.3.8 sequence has completed, the BUCKx_EN bit is set for BUCK1, BUCK3, BUCK4, and BUCK5 in the TPS65941310. BUCKx_EN is set for the BUCK1 and BUCK5 in the TPS65941111. The BUCKx_VMON_EN bit is set for BUCK1, BUCK3, BUCK4 and BUCK5 in the TPS65941310. The BUCKx_VMON_EN bit is set for BUCK1 and BUCK5 in the TPS65941111. The BUCKx_RV_SEL bit is cleared for all BUCKs. The other bits remain unchanged, but are still accessible via I^{2}C.

Table 5-3. BUCK NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
BUCK1_CTRL	BUCK1_EN	0x0	Disabled; BUCK1 regulator	0x0	Disabled; BUCK1 regulator
	BUCK1_FPWM	0x0	PFM and PWM operation (AUTO mode).	0x0	PFM and PWM operation (AUTO mode).
	BUCK1_FPWM_MP	0x0	Automatic phase adding and shedding.	0x0	Automatic phase adding and shedding.
	BUCK1_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK1_VSEL	0x0	BUCK1_VOUT_1	0x0	BUCK1_VOUT_1
	BUCK1_PLDN	0x1	Enabled; Pull-down resistor	0x1	Enabled; Pull-down resistor
	BUCK1_RV_SEL	0x1	Enabled	0x1	Enabled
BUCK1_CONF	BUCK1_SLEW_RATE	0x3	5.0 mV/µs	0x3	5.0 mV/µs
	BUCK1_ILIM	0x5	5.5 A	0x5	5.5 A
BUCK2_CTRL	BUCK2_EN	0x0	Disabled; BUCK2 regulator	0x0	Disabled; BUCK2 regulator
	BUCK2_FPWM	0x0	PFM and PWM operation (AUTO mode).	0x0	PFM and PWM operation (AUTO mode).
	BUCK2_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK2_VSEL	0x0	BUCK2_VOUT_1	0x0	BUCK2_VOUT_1
	BUCK2_PLDN	0x1	Enabled; Pull-down resistor	0x1	Enabled; Pull-down resistor
	BUCK2_RV_SEL	0x1	Enabled	0x1	Enabled
BUCK2_CONF	BUCK2_SLEW_RATE	0x3	5.0 mV/µs	0x3	5.0 mV/µs
	BUCK2_ILIM	0x5	5.5 A	0x5	5.5 A

Table 5-3. BUCK NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
BUCK3_CTRL	BUCK3_EN	0x0	Disabled; BUCK3 regulator	0x0	Disabled; BUCK3 regulator
	BUCK3_FPWM	0x0	PFM and PWM operation (AUTO mode).	0x0	PFM and PWM operation (AUTO mode).
	BUCK3_FPWM_MP	0x0	Automatic phase adding and shedding.	0x0	Automatic phase adding and shedding.
	BUCK3_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK3_VSEL	0x0	BUCK3_VOUT_1	0x0	BUCK3_VOUT_1
	BUCK3_PLDN	0x1	Enabled; Pull-down resistor	0x1	Enabled; Pull-down resistor
	BUCK3_RV_SEL	0x1	Enabled	0x0	Disabled
BUCK3_CONF	BUCK3_SLEW_RATE	0x3	5.0 mV/µs	0x2	10 mV/µs
	BUCK3_ILIM	0x5	5.5 A	0x4	4.5 A
BUCK4_CTRL	BUCK4_EN	0x0	Disabled; BUCK4 regulator	0x0	Disabled; BUCK4 regulator
	BUCK4_FPWM	0x0	PFM and PWM operation (AUTO mode).	0x0	PFM and PWM operation (AUTO mode).
	BUCK4_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK4_VSEL	0x0	BUCK4_VOUT_1	0x0	BUCK4_VOUT_1
	BUCK4_PLDN	0x1	Enabled; Pull-down resistor	0x1	Enabled; Pull-down resistor
	BUCK4_RV_SEL	0x1	Enabled	0x0	Disabled
BUCK4_CONF	BUCK4_SLEW_RATE	0x3	5.0 mV/µs	0x2	10 mV/µs
	BUCK4_ILIM	0x5	5.5 A	0x4	4.5 A
BUCK5_CTRL	BUCK5_EN	0x0	Disabled; BUCK5 regulator	0x0	Disabled; BUCK5 regulator
	BUCK5_FPWM	0x0	PFM and PWM operation (AUTO mode).	0x0	PFM and PWM operation (AUTO mode).
	BUCK5_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK5_VSEL	0x0	BUCK5_VOUT_1	0x0	BUCK5_VOUT_1
	BUCK5_PLDN	0x1	Enable Pull-down resistor	0x1	Enable Pull-down resistor
	BUCK5_RV_SEL	0x1	Enabled	0x1	Enabled
BUCK5_CONF	BUCK5_SLEW_RATE	0x3	5.0 mV/µs	0x3	5.0 mV/µs
	BUCK5_ILIM	0x3	3.5 A	0x3	3.5 A
BUCK1_VOUT_1	BUCK1_VSET1	0x37	0.800 V	0x37	0.800 V
BUCK1_VOUT_2	BUCK1_VSET2	0x37	0.800 V	0x0	0.3 V
BUCK2_VOUT_1	BUCK2_VSET1	0x37	0.800 V	0x37	0.800 V
BUCK2_VOUT_2	BUCK2_VSET2	0x37	0.800 V	0x0	0.3 V
BUCK3_VOUT_1	BUCK3_VSET1	0x41	0.850 V	0x0	0.3 V
BUCK3_VOUT_2	BUCK3_VSET2	0x41	0.850 V	0x0	0.3 V
BUCK4_VOUT_1	BUCK4_VSET1	0x73	1.10 V	0x0	0.3 V
BUCK4_VOUT_2	BUCK4_VSET2	0x73	1.10 V	0x0	0.3 V
BUCK5_VOUT_1	BUCK5_VSET1	0xb2	1.80 V	0x41	0.850 V
BUCK5_VOUT_2	BUCK5_VSET2	0x0	0.3 V	0x0	0.3 V
BUCK1_PG_WINDOW	BUCK1_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	BUCK1_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV
BUCK2_PG_WINDOW	BUCK2_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	BUCK2_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV
BUCK3_PG_WINDOW	BUCK3_OV_THR	0x3	+5% / +50 mV	0x0	+3% / +30mV
	BUCK3_UV_THR	0x3	-5% / -50 mV	0x0	-3% / -30mV

Table 5-3. BUCK NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
BUCK4_PG_WINDOW	BUCK4_OV_THR	0x3	+5% / +50 mV	0x0	+3% / +30mV
	BUCK4_UV_THR	0x3	-5% / -50 mV	0x0	-3% / -30mV
BUCK5_PG_WINDOW	BUCK5_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	BUCK5_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV

5.4 LDO Settings

These settings detail the voltages, configurations, and monitoring of the LDO rails stored in the NVM. All these settings can be changed though I^2C after startup. Some settings, typically the enable bits, are also changed by the PFSM, as described in Section 6.3.

After the Section 6.3.8 sequence has completed, the LDOx_EN and LDOx_VMON_EN bits are set and the LDOx_RV_SEL bit is cleared for all LDOs. The other bits remain unchanged, but are still accessible via I^2C.

Table 5-4. LDO NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
LDO1_CTRL	LDO1_EN	0x0	Disabled; LDO1 regulator.	0x0	Disabled; LDO1 regulator.
	LDO1_SLOW_RAMP	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET
	LDO1_PLDN	0x1	125 Ohm	0x1	125 Ohm
	LDO1_VMON_EN	0x0	Disable OV and UV comparators.	0x0	Disable OV and UV comparators.
	LDO1_RV_SEL	0x1	Enabled	0x1	Enabled
LDO2_CTRL	LDO2_EN	0x0	Disabled; LDO2 regulator.	0x0	Disabled; LDO2 regulator.
	LDO2_SLOW_RAMP	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET
	LDO2_PLDN	0x1	125 Ohm	0x1	125 Ohm
	LDO2_VMON_EN	0x0	Disabled; OV and UV comparators.	0x0	Disabled; OV and UV comparators.
	LDO2_RV_SEL	0x1	Enabled	0x1	Enabled
LDO3_CTRL	LDO3_EN	0x0	Disabled; LDO3 regulator.	0x0	Disabled; LDO3 regulator.
	LDO3_SLOW_RAMP	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET
	LDO3_PLDN	0x1	125 Ohm	0x1	125 Ohm
	LDO3_VMON_EN	0x0	Disabled; OV and UV comparators.	0x0	Disabled; OV and UV comparators.
	LDO3_RV_SEL	0x1	Enabled	0x1	Enabled
LDO4_CTRL	LDO4_EN	0x0	Disabled; LDO4 regulator.	0x0	Disabled; LDO4 regulator.
	LDO4_SLOW_RAMP	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET	0x0	25mV/us max ramp up slew rate for LDO output from 0.3V to 90% of LDOn_VSET
	LDO4_PLDN	0x1	125 Ohm	0x1	125 Ohm
	LDO4_VMON_EN	0x0	Disabled; OV and UV comparators.	0x0	Disabled; OV and UV comparators.
	LDO4_RV_SEL	0x1	Enabled	0x1	Enabled
LDO1_VOUT	LDO1_VSET	0x1c	1.80 V	0x3a	3.30 V
	LDO1_BYPASS	0x0	Linear regulator mode.	0x1	Bypass mode.
LDO2_VOUT	LDO2_VSET	0x1c	1.80 V	0x3a	3.30 V
	LDO2_BYPASS	0x0	Linear regulator mode.	0x1	Bypass mode.
LDO3_VOUT	LDO3_VSET	0x8	0.80 V	0x1c	1.80 V
	LDO3_BYPASS	0x0	Linear regulator mode.	0x0	Linear regulator mode.
LDO4_VOUT	LDO4_VSET	0x38	1.800 V	0x38	1.800 V
LDO1_PG_WINDOW	LDO1_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	LDO1_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV
LDO2_PG_WINDOW	LDO2_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	LDO2_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV
LDO3_PG_WINDOW	LDO3_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	LDO3_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV

Table 5-4. LDO NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
LDO4_PG_WINDOW	LDO4_OV_THR	0x3	+5% / +50 mV	0x3	+5% / +50 mV
	LDO4_UV_THR	0x3	-5% / -50 mV	0x3	-5% / -50 mV

5.5 VCCA Settings

These settings detail the default monitoring enabled on VCCA. All these settings can be changed though I^2C after startup.

Table 5-5. VCCA NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
VCCA_VMON_CTRL	VMON_DEGLITCH_SEL	0x1	20 us	0x1	20 us
	VCCA_VMON_EN	0x1	Enabled; OV and UV comparators.	0x1	Enabled; OV and UV comparators.
VCCA_PG_WINDOW	VCCA_OV_THR	0x7	+10%	0x7	+10%
	VCCA_UV_THR	0x7	-10%	0x7	-10%
	VCCA_PG_SET	0x0	3.3 V	0x0	3.3 V
GENERAL_REG_1	FAST_VCCA_OVP	0x0	slow, 4us deglitch filter enabled	0x0	slow, 4us deglitch filter enabled
GENERAL_REG_3	LPM_EN_DISABLES_VCCA_VMON	0x1	VCCA_VMON enabled if VCCA_VMON_EN=1 and LPM_EN=0	0x1	VCCA_VMON enabled if VCCA_VMON_EN=1 and LPM_EN=0

5.6 GPIO Settings

These settings detail the default configurations of the GPIO rails. All these settings can be changed though I^2C after startup. Note that the contents of the GPIOx_SEL field determine which other fields in the GPIOx_CONF and GPIO_OUT_x registers are applicable. To understand which NVM fields apply to each GPIOx_SEL option, see the *Digital Signal Descriptions* section in TPS6594-Q1 data sheet.

Table 5-6. GPIO NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
GPIO1_CONF	GPIO1_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO1_DIR	0x0	Input	0x0	Input
	GPIO1_SEL	0x1	SCL_I2C2/CS_SPI	0x0	GPIO1
	GPIO1_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO1_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO1_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO2_CONF	GPIO2_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO2_DIR	0x0	Input	0x0	Input
	GPIO2_SEL	0x2	SDA_I2C2/SDO_SPI	0x0	GPIO2
	GPIO2_PU_SEL	0x0	Pull-down resistor selected	0x1	Pull-up resistor selected
	GPIO2_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO2_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x1	8 us deglitch time.

Table 5-6. GPIO NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
GPIO3_CONF	GPIO3_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO3_DIR	0x1	Output	0x1	Output
	GPIO3_SEL	0x0	GPIO3	0x0	GPIO3
	GPIO3_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO3_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO3_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO4_CONF	GPIO4_OD	0x0	Push-pull output	0x1	Open-drain output
	GPIO4_DIR	0x0	Input	0x1	Output
	GPIO4_SEL	0x6	LP_WKUP1	0x0	GPIO4
	GPIO4_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO4_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO4_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO5_CONF	GPIO5_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO5_DIR	0x1	Output	0x0	Input
	GPIO5_SEL	0x1	SCLK_SPMI	0x1	SCLK_SPMI
	GPIO5_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO5_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO5_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO6_CONF	GPIO6_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO6_DIR	0x0	Input	0x0	Input
	GPIO6_SEL	0x1	SDATA_SPMI	0x1	SDATA_SPMI
	GPIO6_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO6_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO6_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO7_CONF	GPIO7_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO7_DIR	0x0	Input	0x0	Input
	GPIO7_SEL	0x1	NERR_MCU	0x0	GPIO7
	GPIO7_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO7_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO7_DEGLITCH_EN	0x1	8 us deglitch time.	0x1	8 us deglitch time.
GPIO8_CONF	GPIO8_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO8_DIR	0x0	Input	0x0	Input
	GPIO8_SEL	0x3	DISABLE_WDOG	0x0	GPIO8
	GPIO8_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO8_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO8_DEGLITCH_EN	0x1	8 us deglitch time.	0x0	No deglitch, only synchronization.

Table 5-6. GPIO NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
GPIO9_CONF	GPIO9_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO9_DIR	0x0	Input	0x1	Output
	GPIO9_SEL	0x0	GPIO9	0x0	GPIO9
	GPIO9_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO9_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO9_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
GPIO10_CONF	GPIO10_OD	0x0	Push-pull output	0x0	Push-pull output
	GPIO10_DIR	0x0	Input	0x0	Input
	GPIO10_SEL	0x6	WKUP1	0x7	WKUP2
	GPIO10_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO10_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO10_DEGLITCH_EN	0x1	8 us deglitch time.	0x1	8 us deglitch time.
GPIO11_CONF	GPIO11_OD	0x1	Open-drain output	0x0	Push-pull output
	GPIO11_DIR	0x1	Output	0x1	Output
	GPIO11_SEL	0x2	NRSTOUT_SOC	0x0	GPIO11
	GPIO11_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	GPIO11_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO11_DEGLITCH_EN	0x0	No deglitch, only synchronization.	0x0	No deglitch, only synchronization.
NPWRON_CONF	NPWRON_SEL	0x0	ENABLE	0x0	ENABLE
	ENABLE_PU_SEL	0x0	Pull-down resistor selected	0x0	Pull-down resistor selected
	ENABLE_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.	0x1	Enabled; Pull-up/pull-down resistor.
	ENABLE_DEGLITCH_EN	0x1	8 us deglitch time when ENABLE, 50 ms deglitch time when NPWRON.	0x1	8 us deglitch time when ENABLE, 50 ms deglitch time when NPWRON.
	ENABLE_POL	0x0	Active high	0x0	Active high
	NRSTOUT_OD	0x1	Open-drain output	0x1	Open-drain output
GPIO_OUT_1	GPIO1_OUT	0x0	Low	0x0	Low
	GPIO2_OUT	0x0	Low	0x0	Low
	GPIO3_OUT	0x0	Low	0x0	Low
	GPIO4_OUT	0x0	Low	0x0	Low
	GPIO5_OUT	0x0	Low	0x0	Low
	GPIO6_OUT	0x0	Low	0x0	Low
	GPIO7_OUT	0x0	Low	0x0	Low
	GPIO8_OUT	0x0	Low	0x0	Low
GPIO_OUT_2	GPIO9_OUT	0x0	Low	0x0	Low
	GPIO10_OUT	0x0	Low	0x0	Low
	GPIO11_OUT	0x0	Low	0x0	Low

5.7 Finite State Machine (FSM) Settings

These settings describe how the PMIC output rails are assigned to various system-level states. Also, the default trigger for each system-level state is described. All these settings can be changed though I^{2}C after startup.

Table 5-7. FSM NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
RAIL_SEL_1	BUCK1_GRP_SEL	0x2	SOC rail group	0x2	SOC rail group
	BUCK2_GRP_SEL	0x2	SOC rail group	0x2	SOC rail group
	BUCK3_GRP_SEL	0x1	MCU rail group	0x0	No group assigned
	BUCK4_GRP_SEL	0x1	MCU rail group	0x0	No group assigned
RAIL_SEL_2	BUCK5_GRP_SEL	0x2	SOC rail group	0x2	SOC rail group
	LDO1_GRP_SEL	0x1	MCU rail group	0x0	No group assigned
	LDO2_GRP_SEL	0x1	MCU rail group	0x2	SOC rail group
	LDO3_GRP_SEL	0x2	SOC rail group	0x2	SOC rail group
RAIL_SEL_3	LDO4_GRP_SEL	0x1	MCU rail group	0x2	SOC rail group
	VCCA_GRP_SEL	0x1	MCU rail group	0x1	MCU rail group
FSM_TRIG_SEL_1	MCU_RAIL_TRIG	0x2	MCU power error	0x2	MCU power error
	SOC_RAIL_TRIG	0x3	SOC power error	0x3	SOC power error
	OTHER_RAIL_TRIG	0x1	Orderly shutdown	0x1	Orderly shutdown
	SEVERE_ERR_TRIG	0x0	Immediate shutdown	0x0	Immediate shutdown
FSM_TRIG_SEL_2	MODERATE_ERR_TRIG	0x1	Orderly shutdown	0x1	Orderly shutdown

5.8 Interrupt Settings

These settings detail the default configurations for what is monitored by nINT pin. All these settings can be changed though I^{2}C after startup.

Table 5-8. Interrupt NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
FSM_TRIG_MASK_1	GPIO1_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO1_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO2_FSM_MASK	0x1	Masked	0x0	Not masked
	GPIO2_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO3_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO3_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO4_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO4_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
FSM_TRIG_MASK_2	GPIO5_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO5_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO6_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO6_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO7_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO7_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO8_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO8_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'

Table 5-8. Interrupt NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
FSM_TRIG_MASK_3	GPIO9_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO9_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO10_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO10_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
	GPIO11_FSM_MASK	0x1	Masked	0x1	Masked
	GPIO11_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'	0x0	Low; Masking sets signal value to '0'
MASK_BUCK1_2	BUCK1_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK1_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK1_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK2_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK2_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK2_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_BUCK3_4	BUCK3_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK3_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK3_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK4_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK4_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK4_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_BUCK5	BUCK5_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK5_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BUCK5_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_LDO1_2	LDO1_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO1_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO2_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO2_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO1_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO2_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_LDO3_4	LDO3_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO3_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO4_OV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO4_UV_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO3_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	LDO4_ILIM_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_VMON	VCCA_OV_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	VCCA_UV_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
MASK_GPIO1_8_FALL	GPIO1_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO2_FALL_MASK	0x1	Interrupt not generated.	0x0	Interrupt generated
	GPIO3_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO4_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO5_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO6_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO7_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO8_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.

Table 5-8. Interrupt NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
MASK_GPIO1_8_RISE	GPIO1_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO2_RISE_MASK	0x1	Interrupt not generated.	0x0	Interrupt generated
	GPIO3_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO4_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO5_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO6_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO7_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO8_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
MASK_GPIO9_11 / MASK_GPIO9_10	GPIO9_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO9_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO10_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO11_FALL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO10_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	GPIO11_RISE_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
MASK_STARTUP	NPWRON_START_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ENABLE_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	FSD_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	SOFT_REBOOT_MASK	0x0	Interrupt generated	0x0	Interrupt generated
MASK_MISC	TWARN_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	BIST_PASS_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	EXT_CLK_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
MASK_MODERATE_ERR	BIST_FAIL_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	REG_CRC_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	SPMI_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	NPWRON_LONG_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	NINT_READBACK_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	NRSTOUT_READBACK_ MASK	0x0	Interrupt generated	0x1	Interrupt not generated.
MASK_FSM_ERR	IMM_SHUTDOWN_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	MCU_PWR_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	SOC_PWR_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	ORD_SHUTDOWN_MASK	0x0	Interrupt generated	0x0	Interrupt generated

Table 5-8. Interrupt NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
MASK_COMM_ERR	COMM_FRM_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	COMM_CRC_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	COMM_ADR_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated
	I2C2_CRC_ERR_MASK	0x0	Interrupt generated	0x1	Interrupt not generated.
	I2C2_ADR_ERR_MASK	0x0	Interrupt generated	0x1	Interrupt not generated.
MASK_READBACK_ERR	EN_DRV_READBACK_MASK	0x0	Interrupt generated	0x1	Interrupt not generated.
	NRSTOUT_SOC_READBACK_MASK	0x0	Interrupt generated	0x1	Interrupt not generated.
MASK_ESM	ESM_SOC_PIN_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ESM_SOC_RST_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ESM_SOC_FAIL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ESM_MCU_PIN_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ESM_MCU_RST_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
	ESM_MCU_FAIL_MASK	0x1	Interrupt not generated.	0x1	Interrupt not generated.
GENERAL_REG_1	PFSM_ERR_MASK	0x0	Interrupt generated	0x0	Interrupt generated

1. The VCCA_OV_MASK and VCCA_UV_MASK are cleared in both PMICs after the completing BOOT_BIST but before starting the sequence, Section 6.3.8.

5.9 POWERGOOD Settings

These settings detail the default configurations for what is monitored by PGOOD pin. All these settings can be changed though I^{2}C after startup.

Table 5-9. POWERGOOD NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
PGOOD_SEL_1	PGOOD_SEL_BUCK1	0x0	Masked	0x0	Masked
	PGOOD_SEL_BUCK2	0x0	Masked	0x0	Masked
	PGOOD_SEL_BUCK3	0x0	Masked	0x0	Masked
	PGOOD_SEL_BUCK4	0x0	Masked	0x0	Masked
PGOOD_SEL_2	PGOOD_SEL_BUCK5	0x0	Masked	0x0	Masked
PGOOD_SEL_3	PGOOD_SEL_LDO1	0x0	Masked	0x0	Masked
	PGOOD_SEL_LDO2	0x0	Masked	0x0	Masked
	PGOOD_SEL_LDO3	0x0	Masked	0x0	Masked
	PGOOD_SEL_LDO4	0x0	Masked	0x0	Masked

Table 5-9. POWERGOOD NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
PGOOD_SEL_4	PGOOD_SEL_VCCA	0x0	Masked	0x0	Masked
	PGOOD_SEL_TDIE_WARN	0x0	Masked	0x0	Masked
	PGOOD_SEL_NRSTOUT	0x0	Masked	0x0	Masked
	PGOOD_SEL_NRSTOUT_ SOC	0x0	Masked	0x0	Masked
	PGOOD_POL	0x0	PGOOD signal is high when monitored inputs are valid	0x0	PGOOD signal is high when monitored inputs are valid
	PGOOD_WINDOW	0x0	Only undervoltage is monitored	0x0	Only undervoltage is monitored

5.10 Miscellaneous Settings

These settings detail the default configurations of additional settings, such as spread spectrum, BUCK frequency, and LDO timeout. All these settings can be changed though I^{2}C after startup.

Table 5-10. Miscellaneous NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
PLL_CTRL	EXT_CLK_FREQ	0x0	1.1 MHz	0x0	1.1 MHz
CONFIG_1	TWARN_LEVEL	0x0	130C	0x0	130C
	I2C1_HS	0x0	Standard, fast or fast+ by default, can be set to Hs-mode by Hs-mode primary code.	0x0	Standard, fast or fast+ by default, can be set to Hs-mode by Hs-mode primary code.
	I2C2_HS	0x0	Standard, fast or fast+ by default, can be set to Hs-mode by Hs-mode primary code.	0x0	Standard, fast or fast+ by default, can be set to Hs-mode by Hs-mode primary code.
	EN_ILIM_FSM_CTRL	0x0	Buck/LDO regulators ILIM interrupts do not affect FSM triggers.	0x0	Buck/LDO regulators ILIM interrupts do not affect FSM triggers.
	NSLEEP1_MASK	0x0	NSLEEP1(B) affects FSM state transitions.	0x0	NSLEEP1(B) affects FSM state transitions.
	NSLEEP2_MASK	0x0	NSLEEP2(B) affects FSM state transitions.	0x0	NSLEEP2(B) affects FSM state transitions.
CONFIG_2	BB_CHARGER_EN	0x0	Disabled	0x0	Disabled
	BB_VEOC	0x0	2.5V	0x0	2.5V
	BB_ICHR	0x0	100uA	0x0	100uA
RECOV_CNT_REG_2	RECOV_CNT_THR	0xf	0xf	0xf	0xf
BUCK_RESET_REG	BUCK1_RESET	0x0	0x0	0x0	0x0
	BUCK2_RESET	0x0	0x0	0x0	0x0
	BUCK3_RESET	0x0	0x0	0x0	0x0
	BUCK4_RESET	0x0	0x0	0x0	0x0
	BUCK5_RESET	0x0	0x0	0x0	0x0
SPREAD_SPECTRUM_1	SS_EN	0x0	Spread spectrum disabled	0x0	Spread spectrum disabled
	SS_MODE	0x1	Mixed dwell	0x1	Mixed dwell
	SS_DEPTH	0x0	No modulation	0x0	No modulation
SPREAD_SPECTRUM_2	SS_PARAM1	0x7	0x7	0x7	0x7
	SS_PARAM2	0xc	0xc	0xc	0xc

Table 5-10. Miscellaneous NVM Settings (continued)

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
FREQ_SEL	BUCK1_FREQ_SEL	0x0	2.2 MHz	0x0	2.2 MHz
	BUCK2_FREQ_SEL	0x0	2.2 MHz	0x0	2.2 MHz
	BUCK3_FREQ_SEL	0x0	2.2 MHz	0x0	2.2 MHz
	BUCK4_FREQ_SEL	0x0	2.2 MHz	0x0	2.2 MHz
	BUCK5_FREQ_SEL	0x0	2.2 MHz	0x0	2.2 MHz
FSM_STEP_SIZE	PFSM_DELAY_STEP	0xb	0xb	0xb	0xb
LDO_RV_TIMEOUT_REG_1	LDO1_RV_TIMEOUT	0xf	16ms	0xf	16ms
	LDO2_RV_TIMEOUT	0xf	16ms	0xf	16ms
LDO_RV_TIMEOUT_REG_2	LDO3_RV_TIMEOUT	0xf	16ms	0xf	16ms
	LDO4_RV_TIMEOUT	0xf	16ms	0xf	16ms
USER_SPARE_REGS	USER_SPARE_1	0x0	0x0	0x0	0x0
	USER_SPARE_2	0x0	0x0	0x0	0x0
	USER_SPARE_3	0x0	0x0	0x0	0x0
	USER_SPARE_4	0x0	0x0	0x0	0x0
ESM_MCU_MODE_CFG	ESM_MCU_EN	0x0	ESM_MCU disabled.	0x0	ESM_MCU disabled.
ESM_SOC_MODE_CFG	ESM_SOC_EN	0x0	ESM_SoC disabled.	0x0	ESM_SoC disabled.
CUSTOMER_NVM_ID_REG	CUSTOMER_NVM_ID	0x0	0x0	0x0	0x0
RTC_CTRL_2	XTAL_EN	0x0	Crystal oscillator is disabled	0x0	Crystal oscillator is disabled
	LP_STANDBY_SEL	0x1	Low power standby state is used as standby state (LDOINT is disabled).	0x1	Low power standby state is used as standby state (LDOINT is disabled).
	FAST_BIST	0x0	Logic and analog BIST is run at BOOT BIST.	0x0	Logic and analog BIST is run at BOOT BIST.
	STARTUP_DEST	0x3	ACTIVE	0x3	ACTIVE
	XTAL_SEL	0x0	6 pF	0x0	6 pF
PFSM_DELAY_REG_1	PFSM_DELAY1	0x58	0x58	0x0	0x0
PFSM_DELAY_REG_2	PFSM_DELAY2	0x9d	0x9d	0x1d	0x1d
PFSM_DELAY_REG_3	PFSM_DELAY3	0x0	0x0	0x0	0x0
PFSM_DELAY_REG_4	PFSM_DELAY4	0x0	0x0	0x0	0x0
GENERAL_REG_0	FAST_BOOT_BIST	0x0	LBIST is run during boot BIST	0x0	LBIST is run during boot BIST
GENERAL_REG_1	REG_CRC_EN	0x1	Register CRC enabled	0x1	Register CRC enabled

5.11 Interface Settings

These settings detail the default interface, interface configurations, and device addresses. These settings cannot be changed after device startup.

Table 5-11. Interface NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
SERIAL_IF_CONFIG	I2C_SPI_SEL	0x0	I2C	0x0	I2C
	I2C1_SPI_CRC_EN	0x0	CRC disabled	0x0	CRC disabled
	I2C2_CRC_EN	0x0	CRC disabled	0x0	CRC disabled
I2C1_ID_REG	I2C1_ID	0x48	0x48	0x4c	0x4C
I2C2_ID_REG	I2C2_ID	0x12	0x12	0x13	0x13

5.12 Multi-Device Settings

These settings detail whether the device is a operating as a primary or secondary in the system. These settings cannot be changed after device startup.

Table 5-12. Multi-Device NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
SPMI_CONFIG_1	SPMI_CRC_EN	0x1	SPMI CRC check enabled	0x1	SPMI CRC check enabled
	BIT 1	0x1	Primary mode	0x0	Secondary mode
	SPMI_CLK_SEL	0x2	5MHz	0x2	5MHz
SPMI_CONFIG_2	SPMI_IF_SEL	0x0	Debug feature and uses primary logic to implement logical secondary.	0x0	Debug feature and uses primary logic to implement logical secondary.
	SPMI_RETRY_LIMIT	0x3	Three retries in case of error detected	0x3	Three retries in case of error detected
	SPMI_WD_AUTO_BOOT	0x1	SPMI auto boot enabled	0x1	SPMI auto boot enabled
	SPMI_EN	0x1	SPMI enabled	0x1	SPMI enabled
	SPMI_WD_EN	0x1	SPMI WD enabled	0x1	SPMI WD enabled
SPMI_CONFIG_3	SPMI_WD_BOOT_ INTERVAL	0x8	0x8	0x8	0x8
	SPMI_WD_RUNTIME_ INTERVAL	0x8	0x8	0x8	0x8
SPMI_CONFIG_4	SPMI_WD_RESPONSE_ TIMEOUT	0x8	0x8	0x8	0x8
	SPMI_PFSM_RESPONSE_ TIMEOUT	0x8	0x8	0x8	0x8
SPMI_CONFIG_5	SPMI_WD_RUNTIME_ BIST_ TIMEOUT	0x8	0x8	0x8	0x8
	SPMI_WD_BOOT_BIST_ TIMEOUT	0x8	0x8	0x8	0x8
SPMI_CONFIG_6	BOOT_DELAY	0x0	0x0	0x0	0x0
SPMI_ID	SPMI_SID	0x5	0x5	0x3	0x3
	SPMI_MID	0x0	0x0	0x0	0x0

5.13 Watchdog Settings

These settings detail the default watchdog addresses. These settings can be changed though I^{2}C after startup.

Table 5-13. Watchdog NVM Settings

Register Name	Field Name	TPS65941310-Q1		TPS65941111-Q1	
		Value	Description	Value	Description
WD_LONGWIN_CFG	WD_LONGWIN	0xff	0xff	0xff	0xff
WD_THR_CFG	WD_EN	0x1	Watchdog enabled.	0x0	Watchdog disabled.

6 Pre-Configurable Finite State Machine (PFSM) Settings

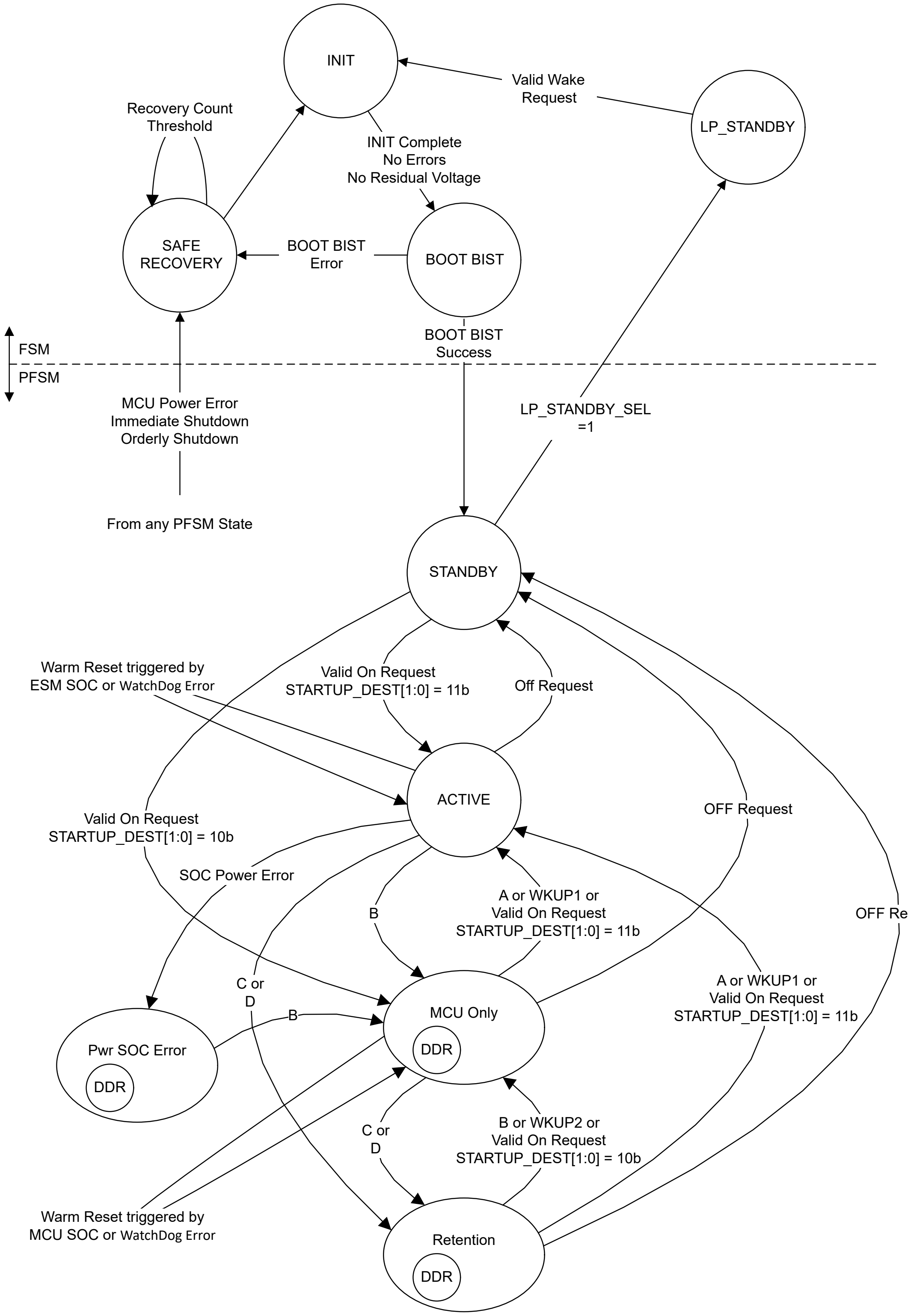
This section describes the default PFSM settings of the TPS6594-Q1 devices. These settings cannot be changed after device startup.

6.1 Configured States

In this PDN, the PMIC devices have the following four configured power states:

- Standby
- Active
- MCU Only
- Pwr SoC Error
- DDR Retention

In Figure 6-1, the configured PDN power states are shown, along with the transition conditions to move between the states. Additionally, the transitions to hardware states, such as SAFE RECOVERY and LP_STANDBY are shown. The hardware states are part of the fixed device power Finite State Machine (FSM) and described in the TPS6594-Q1 data sheet, see Section 8.



Trigger	NSLEEP2	NSLEEP1
A	1	1
B	1	0
C	0	1
D	0	0

Figure 6-1. Pre-Configurable Finite State Machine (PFSM) Mission States and Transitions

When the PMICs transition from the FSM to the PFSM, several initialization instructions are performed to disable the residual voltage checks on both the BUCK and LDO regulators, set the FIRST_STARTUP_DONE bit and clear the VCCA OV and UV masks which are set in the static configurations, Table 5-8. After these instructions are executed the PMICs wait for a valid ON Request before entering the ACTIVE state. The definition for each power state is described below:

STANDBY The PMICs are powered by a valid supply on the system power rail (VCCA > VCCA_UV). All device resources are powered down in the STANDBY state. EN_DRV is forced low in this state. The processor is in the Off state, no voltage domains are energized. Refer to the Section 6.3.2 sequence description.

The STANDBY state is also entered when an error occurs and the PMIC transitions out of the PFSM mission states and into the FSM states. When the device returns from the FSM state the to PFSM the first state is represented by STANDBY with all of the resources powered down and EN_DRV forced low. The sequence Section 6.3.1 is performed before the PMIC leaves the PFSM and enters the FSM state SAFE_RECOVERY.

ACTIVE The PMICs are powered by a valid supply. The PMICs are fully functional and supply power to all PDN loads. The processor has completed a recommended power up sequence with all voltage domains energized in both MCU and Main processor sections. Refer to the Section 6.3.8 sequence description.

MCU_ONLY The PMICs are powered by a valid supply. Only the power resources assigned to the MCU Safety Island are on. Refer to the Section 6.3.7 sequence description.

Pwr SoC Error The PMICs are powered by a valid supply. Only the power resources assigned to the MCU Safety Island are on. Refer to the Section 6.3.5 sequence description. The only active trigger is 'B', requiring the PMICs to return to the MCU_ONLY mode. The return to MCU_ONLY mode and eventually ACTIVE mode is only recommended after the interrupts which caused the SOC_PWR_ERROR have been cleared.

Retention The PMICs are powered by a valid supply. When the PMICs I2C_7 triggers are set (DDR Retention), only 3 SoC voltage domains (vdds_ddr_bias, vdds_ddr, and vdds_ddr_c) remain energized while all other domains are off to minimize total system power. EN_DRV is forced low in this state. Refer to the Section 6.3.9 sequence description.

6.2 PFSM Triggers

As shown in Figure 6-1, there are various triggers that can enable a state transition between configured states. Table 6-1 describes each trigger and its associated state transition from highest priority (Immediate Shutdown) to lowest priority (I2C_3). Active triggers of higher priority block triggers of lower priority and the associated sequence.

Table 6-1. State Transition Triggers

Trigger	Priority (ID)	Immediate (IMM)	REENTERANT	PFSM Current State	PFSM Destination State	Power Sequence or Function Executed
Immediate Shutdown[9]	0	True	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	SAFE[1]	TO_SAFE_SEVERE
MCU Power Error	1	True	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	SAFE[1]	TO_SAFE
Orderly Shutdown[9]	2	True	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	SAFE[1]	TO_SAFE_ORDERLY
OFF Request	4[11]	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	STANDBY[2]	TO_STANDBY
WDOG Error	5	False	True	ACTIVE	ACTIVE	ACTIVE_TO_WARM
ESM MCU Error	6	False	True	ACTIVE	ACTIVE	
ESM SOC Error	7	False	True	ACTIVE	ACTIVE	ESM_SOC_ERROR
WDOG Error	8	False	True	MCU ONLY	MCU ONLY	MCU_TO_WARM
ESM MCU Error	9	False	True	MCU ONLY	MCU ONLY	

Table 6-1. State Transition Triggers (continued)

Trigger	Priority (ID)	Immediate (IMM)	REENTERANT	PFSM Current State	PFSM Destination State	Power Sequence or Function Executed
SOC Power Error[9]	10	False	False	ACTIVE	MCU ONLY	PWR_SOC_ERR
I2C_1 bit is high[3]	11	False	True	ACTIVE, MCU ONLY	No State Change	Execute RUNTIME BIST
I2C_2 bit is high[3]	12	False	True	ACTIVE, MCU ONLY	No State Change	Enable I^2C CRC on I^2C1 and I^2C2 on all devices.[4]
GPIO2 Falling Edge[7]	13	False	False	ACTIVE	No State Change	TPS65941111-Q1 LDO1 output is 3.3 V in BYPASS mode
GPIO2 Rising Edge[7]	14	False	False	ACTIVE	No State Change	TPS65941111-Q1 LDO1 output is 1.8 V in LDO mode
ON Request	15	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	TO_ACTIVE
WKUP1 goes high	16	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	
NSLEEP1 and NSLEEP2 are high[5]	17	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	
MCU ON Request	18	False	False	STANDBY, ACTIVE[8], MCU ONLY, Suspend-to-RAM	MCU ONLY	TO_MCU
WKUP2 goes high	19	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	MCU ONLY	
NSLEEP1 goes low and NSLEEP2 goes high[5]	20	False	False	ACTIVE, MCU ONLY, Suspend-to-RAM	MCU ONLY	
NSLEEP1 goes low and NSLEEP2 goes low[5]	21	False	False	ACTIVE, MCU ONLY	Suspend-to-RAM	TO_RETENTION
NSLEEP1 goes high and NSLEEP2 goes low[5]	22	False	False	ACTIVE, MCU ONLY	Suspend-to-RAM	
I2C_0 bit goes high[3]	23[10]	False	False	STANDBY, ACTIVE, MCU ONLY	LP_STANDBY[2]	TO_STANDBY
I2C_3 bit goes high[3]	24[10]	False	False	ACTIVE, MCU ONLY	No State Change	Devices are prepared for OTA NVM update.[6]

[1] From the SAFE state, the PFSM automatically transitions to the hardware FSM state of SAFE_RECOVERY. From the SAFE_RECOVERY state, the recovery counter is incremented and compared to the recovery count threshold (see RECOV_CNT_REG_2, in Table 5-10). If the recovery count threshold is reached, then the PMICs halt recovery attempts and require a power cycle. Refer to the datasheet for more details.

[2] If the LP_STANDBY_SEL bit is set in the TPS65941310-Q1 (see RTC_CTRL_2, in Table 5-10), then the PFSM transitions to the hardware FSM state of LP_STANDBY. When LP_STANDBY is entered, then please use the appropriate mechanism to wake up the device as determined by the means of entering LP_STANDBY. Refer to the datasheet for more details. LP_STANDBY_SEL in the TPS65941111-Q1 is not applicable to the PFSM triggers.

[3] I2C_0, I2C_1, I2C_2 and I2C_3 are self-clearing triggers.

[4] Enabling the I^2C CRC, enables the CRC on both I2C1 and I2C2, however, the I2C2 is disabled for 2ms after the CRC is enabled. Be aware when using the watchdog Q&A before enabling I^2C CRC. The recommendation is to enable the I^2C CRC first, and then after 2ms, start the watchdog Q&A.

[5] NSLEEP1 and NSLEEP2 of the primary PMIC can be accessed through the GPIO pin or through a register bit. If either the register bit or the GPIO pin is pulled high, the NSLEEPx value is read as a *high* logic level.

(6) After completion of an OTA update, the processor is required to initiate a reset of the PMICs to apply the new NVM settings.
(7) The GPIO2 triggers are referring to GPIO2 on the TPS65941111.
(8) When in the ACTIVE mode, the ON Request to MCU ONLY trigger cannot be accessed while other higher priority triggers, like NSLEEP1=NSLEEP2=HIGH, are still active.
(9) These triggers can originate from either the TPS65941310 or the TPS65941111. All other triggers except for the GPIO2 triggers originate from the TPS65941310.
(10) Trigger IDs 23 and 24 are not available until the NSLEEP bits are masked: NSLEEP2_MASK=NSLEEP1_MASK=1.
(11) Trigger IDs 3, 25, and 26 are enabled and activated by the power sequences. These triggers are used to manage the transition between the PFSM and the FSM.

6.3 Power Sequences

6.3.1 TO_SAFE_SEVERE and TO_SAFE

The TO_SAFE_SEVERE and TO_SAFE power sequences shut down all rails without delay and the resulting state after the sequence is the SAFE_RECOVERY state. The TO_SAFE_SEVERE sequence immediately ceases BUCK switching and enables the pulldown resistors of the BUCKs and LDOs. This reset of the BUCK controllers is to prevent any damage of the PMICs in case of over voltage on VCCA or thermal shutdown.
In contrast, the TO_SAFE sequence resets the BUCK controllers at the end of the sequence. The timing is described in the following code blocks.

```
//TPS65941310 TO_SAFE_SEVERE
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
SET_DELAY 26ms
DELAY_IMM 500ms
TRIG_MASK 0xBFFFFFF
END
```

```
//TPS65941111 TO_SAFE_SEVERE
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x10 MASK=0xEF
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x20 MASK=0x4E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x20 MASK=0x0E DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
TRIG_MASK 0xBFFFFFF
END
```

```
//TPS65941310 TO_SAFE
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=MATCH_EN DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
TRIG_MASK 0xBFFFFFF
END
```

```
//TPS65941111 TO_SAFE
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x10 MASK=0xEF
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
TRIG_MASK 0xBFFFFFF
END
```

6.3.2 TO_SAFE_ORDERLY and TO_STANDBY

If a moderate error occurs, an orderly shutdown trigger is generated. This trigger shuts down the PMIC outputs using the recommended power down sequence and proceed to the SAFE state.

If an OFF request occurs, such as the ENABLE pin of the primary TPS6594-Q1 device being pulled low, the same power down sequence occurs, except that the PMICs go to STANDBY (LP_STANDBY_SEL=0) or LP_STANDBY (LP_STANDBY_SEL=1) states, rather than going to the SAFE state.

At the end of the TO_SAFE_ORDERLY both PMICs wait approximately 16 ms before executing the following instructions:

```
//TPS65941310 TO_SAFE_ORDERLY
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 500us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
TRIG_MASK 0xBFFFFFF
END
```

```
//TPS65941111 TO_SAFE_ORDERLY
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x10 MASK=0xEF
DELAY_IMM 500us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
```

```
TRIG_MASK 0xBFFFFFF
END
```

```
// TPS65941310 TO_STANDBY
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 500us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
WAIT COND=LP_STANDBY_SEL TYPE=HIGH TIMEOUT=0us DEST=stayinstandby
TRIG_MASK 0xFFFFFF0
END
stayinstandby:
TRIG_MASK 0xFF27FE8
END
enter_lp_standby:
REG_WRITE_MASK_IMM ADDR=0x21 DATA=0x01 MASK=0xFE
END
```

```
// TPS65941111 TO_STANDBY
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x10 MASK=0xEF
DELAY_IMM 500us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
TRIG_MASK 0xFF27FF8
END
enter_lp_standby:
REG_WRITE_MASK_IMM ADDR=0x21 DATA=0x21 MASK=0xCE
END
```

6.3.3 ACTIVE_TO_WARM

The ACTIVE_TO_WARM sequence can be triggered by either a watchdog or ESM_MCU error. In the event of a trigger, the nRSTOUT and nRSTOUT_SOC signals are driven low and the recovery count (register RECOV_CNT_REG_1) increments. Then, all BUCKs and LDOs are reset to their default voltages. The PMICs remain in the ACTIVE state.

```
//TPS65941310 ACTIVE_TO_WARM
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x08 MASK=0xF7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0xa5 DATA=0x01 MASK=0xFE
REG_WRITE_VOUT_IMM REGULATOR=LDO1 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=LDO4 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=LDO2 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=BUCK5 SEL=2 VOUT=1800mV DELAY=0us
REG_WRITE_VOUT_IMM REGULATOR=BUCK3 SEL=2 VOUT=850mV DELAY=0us
REG_WRITE_VOUT_IMM REGULATOR=BUCK1 SEL=2 VOUT=800mV DELAY=0us
REG_WRITE_VOUT_IMM REGULATOR=LDO3 SEL=0 VOUT=800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=BUCK4 SEL=2 VOUT=1100mV DELAY=0us
DELAY_IMM 2ms
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xF7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x03 MASK=0xFC
END
```

```
//TPS65941111 ACTIVE_TO_WARM
REG_WRITE_VOUT_IMM REGULATOR=LDO2 SEL=0 VOUT=3300mV DELAY=0us BYPASS=1
REG_WRITE_VOUT_IMM REGULATOR=LDO1 SEL=0 VOUT=3300mV DELAY=0us BYPASS=1
REG_WRITE_VOUT_IMM REGULATOR=LDO4 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=BUCK1 SEL=2 VOUT=800mV DELAY=0us
REG_WRITE_VOUT_IMM REGULATOR=LDO3 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=BUCK5 SEL=2 VOUT=850mV DELAY=0us
END
```

Note

The watchdog or ESM error is an indication of a significant error which has taken place outside of the PMIC. The PMIC does not actually transition through the safe recovery as with an MCU_POWER_ERR, however, in order to maintain consistency all of the regulators are returned to the values stored in NVM and the recovery counter is incremented. If the recovery counter exceeds the recovery count threshold the PMICs stay in the safe recovery state.

Note

After the ACTIVE_TO_WARM sequence the MCU is responsible for managing the EN_DRV and recovery counter. At the end of the sequence the 'FORCE_EN_DRV_LOW' bit is cleared so that the MCU can set the ENABLE_DRV bit.

6.3.4 ESM_SOC_ERROR

In the event of an ESM_SOC error, the nRSTOUT_SOC signal is driven low and then driven high again after 200 µs. There is no change to the power rails and no action is performed on the TPS65941111.

```
// TPS65941310 ESM_SOC_ERROR
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFD
DELAY_IMM 200us
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x02 MASK=0xFD
END
```

```
// TPS65941111 ESM_SOC_ERROR
END
```

6.3.5 PWR_SOC_ERROR

In the event of an error on any of the power rails which are part of the SOC power rail group, the PWR_SOC_ERROR sequence is performed. The nRSTOUT_SOC pin is pulled low and the SOC power rails execute a normal processor power down sequence except the MCU power group remains energized. The state of the I2C_7 trigger in both PMICs determines whether the DDR supplies remain energized (I2C_7=1) or disabled (I2C_7=0).

In the start of the sequence the following instructions are executed:

```

// TPS65941310 PWR_SOC_ERROR
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE1
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_socpwr_off1
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_socpwr_off1:
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_socpwr_off2
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_socpwr_off2:
TRIG_MASK 0xE6FE4E8
SREG_READ_REG ADDR=0xCE REG=R2
```

```
DELAY_SREG R2
END
```

```
// TPS65941111 PWR_SOC_ERROR
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_socpwr_off1
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
ddr_socpwr_off1:
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
TRIG_MASK 0xFFFFFF8
END
```

6.3.6 MCU_TO_WARM

The MCU_TO_WARM sequence is triggered by a WATCHDOG or ESM_MCU error. The MCU_TO_WARM, similar to the ACTIVE_TO_WARM sequence does not result in a state change. The event and sequence originate from the MCU_ONLY state and stays in the MCU_ONLY state. In the sequence, the recover counter (found in register, RECOV_CNT_REG_1) is incremented and the nRSTOUT (MCU_PORz) signal is driven low. The MCU relevant BUCK and LDOs are reset to their default voltages, and the MCU_PORz signal is set high after 2ms.

Note

GPIOs do not reset during the MCU warm reset event. There is no action on the TPS65941111.

```
// TPS65941310 MCU_TO_WARM
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x08 MASK=0xF7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFE
REG_WRITE_MASK_IMM ADDR=0xa5 DATA=0x01 MASK=0xFE
REG_WRITE_VOUT_IMM REGULATOR=LDO1 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=LDO4 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=LDO2 SEL=0 VOUT=1800mV DELAY=0us BYPASS=0
REG_WRITE_VOUT_IMM REGULATOR=BUCK3 SEL=2 VOUT=850mV DELAY=0us
REG_WRITE_VOUT_IMM REGULATOR=BUCK4 SEL=2 VOUT=1100mV DELAY=0us
DELAY_IMM 2ms
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xF7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x01 MASK=0xFE
END
```

```
// TPS65941111 MCU_TO_WARM
END
```

Note

The watchdog or MCU error is an indication of a significant error which has taken place outside of the PMIC. The PMIC does not actually transition through the safe recovery as with an MCU_POWER_ERR, however, in order to maintain consistency all of the regulators are returned to the values stored in NVM and the recovery counter is incremented. If the recovery counter exceeds the recovery count threshold the PMICs stay in the safe recovery state.

Note

After the MCU_TO_WARM sequence the MCU is responsible for managing the EN_DRV and recovery counter. At the end of the sequence the 'FORCE_EN_DRV_LOW' bit is cleared so that the MCU can set the ENABLE_DRV bit.

6.3.7 TO_MCU

The TO_MCU sequence first turns off rails and GPIOs which are assigned to the SOC power group. The sequence enables the MCU rails, in the event that they are not already active (when transitioning from STANDBY to MCU_ONLY for example). There are two cases for this sequence, based off the value stored in the I2C_7 register bit of primary TPS65941310-Q1 and secondary TPS65941111-Q1. The I2C_7 setting must be the same in each PMIC before triggering the sequence. If the bits are low, then VDD1_DDR_1V8 and VDD_DDR_1V1 are disabled. If the I2C_7 bit is high, then VDD1_DDR_1V8 and VDD_DDR_1V1 are enabled.

```
// TPS65941310 TO_MCU
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE1
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_mcu_off1
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_mcu_off1:
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_mcu_off2
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_mcu_off2:
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x04 MASK=0xFB
DELAY_IMM 1700us
SREG_READ_REG ADDR=0x1D REG=R0
WAIT COND=SREG0_0 TYPE=LOW TIMEOUT=0us DEST=no_mask_ddr_active
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x03 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x30 MASK=0xCF
no_mask_ddr_active:
WAIT COND=I2C_7 TYPE=HIGH TIMEOUT=0us DEST=ddr_mcu_on3
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_mcu_on3:
SREG_READ_REG ADDR=0x20 REG=R0
WAIT COND=SREG0_0 TYPE=LOW TIMEOUT=0us DEST=nomask_mcuonly
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x30 MASK=0xCF
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x30 MASK=0xCF
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x03 MASK=0xFC
nomask_mcuonly:
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1000us
DELAY_IMM 1000us
WAIT COND=I2C_7 TYPE=HIGH TIMEOUT=0us DEST=ddr_mcu_on4
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_mcu_on4:
SREG_READ_REG ADDR=0xCD REG=R1
DELAY_SREG R1
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xE7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x01 MASK=0xFE
TRIG_MASK 0xE0064E8
END
```

```
// TPS65941111 TO_MCU
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_mcu_off1
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
ddr_mcu_off1:
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
DELAY_IMM 3700us
WAIT COND=I2C_7 TYPE=HIGH TIMEOUT=0us DEST=ddr_mcu_on3
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x04 MASK=0xFB
```

```
ddr_mcu_on3:
TRIG_MASK 0xFF27FF8
END
```

Note

After the TO_MCU sequence the MCU is responsible for managing the EN_DRV.

6.3.8 TO_ACTIVE

When a trigger causes the TO_ACTIVE sequence to execute, all rails power up in the recommended power up sequence.

```
// TPS65941310 TO_ACTIVE
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x04 MASK=0xFB
SREG_READ_REG ADDR=0x20 REG=R0
WAIT COND=SREG0_0 TYPE=LOW TIMEOUT=0us DEST=nomask_mcurails
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x30 MASK=0xCF
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x30 MASK=0xCF
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x03 MASK=0xFC
nomask_mcurails:
SREG_READ_REG ADDR=0x1D REG=R0
WAIT COND=SREG0_0 TYPE=LOW TIMEOUT=0us DEST=nomask_ddrrails
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x03 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x30 MASK=0xCF
nomask_ddrrails:
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1700us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x4B DATA=0x03 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x49 DATA=0x33 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x03 MASK=0xFC
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1000us
SREG_READ_REG ADDR=0xCD REG=R1
DELAY_SREG R1
REG_WRITE_MASK_IMM ADDR=0x49 DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4A DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4B DATA=0x00 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xF7
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x03 MASK=0xFC
TRIG_MASK 0xE000308
END
```

```
//TPS65941111 TO_ACTIVE
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE3
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x04 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x49 DATA=0x33 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4B DATA=0x03 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x33 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x33 MASK=0xCC
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1700us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1000us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=1000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x11 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x04 MASK=0xFB
DELAY_IMM 5000us
REG_WRITE_MASK_IMM ADDR=0x49 DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4B DATA=0x00 MASK=0xFC
REG_WRITE_MASK_IMM ADDR=0x4C DATA=0x00 MASK=0xCC
REG_WRITE_MASK_IMM ADDR=0x4D DATA=0x00 MASK=0xCC
TRIG_MASK 0xFF21BF8
END
```

Note

After the TO_ACTIVE sequence the MCU is responsible for managing the EN_DRV.

6.3.9 TO_RETENTION

The C and D triggers, defined by the NSLEEPx bits or pins, trigger the TO_RETENTION sequence. This sequence disables all power rails and GPIOs that are not supplying the retention rails, as described in Figure 3-1. The sequence can be modified using the I2C_7 bit found in register FSM_I2C_TRIGGERS. These bits need to be set by I^{2}C in both PMICs before a trigger for the retention state occurs. If the I2C_7 bit is set high in both PMICs, they enter the DDR retention state. If I2C_7 is set low, the components associated with DDR do not remain active.

Note

The I2C_7 bits need to be set or cleared by I^{2}C in both PMICs before a trigger to the retention state occurs. The I2C_7 trigger is not self-clearing and must be maintained during operation.

In addition to the I2C_7, the processor must also configure the H_DDR_RET_1V1 signal on GPIO4 of the TPS65941111 device. This signal is included in the Section 3.2 but is not part of the power sequence.

```
//TPS65941310 TO_RETENTION
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xF8
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_s2r_off1
REG_WRITE_VCTRL_IMM REGULATOR=BUCK4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_s2r_off1:
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_s2r_off2
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
ddr_s2r_off2:
DELAY_IMM 500us
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
TRIG_MASK 0xFE07FE8
SREG_READ_REG ADDR=0xCE REG=R2
DELAY_SREG R2
END
```

```
//TPS65941111 TO_RETENTION
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x10 MASK=0xEF
DELAY_IMM 500us
WAIT COND=I2C_7 TYPE=LOW TIMEOUT=0us DEST=ddr_s2r_off1
REG_WRITE_MASK_IMM ADDR=0x3D DATA=0x00 MASK=0xFB
ddr_s2r_off1:
REG_WRITE_VCTRL_IMM REGULATOR=BUCK5 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO3 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
DELAY_IMM 2000us
REG_WRITE_VCTRL_IMM REGULATOR=BUCK1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_VCTRL_IMM REGULATOR=LDO4 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO1 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=500us
REG_WRITE_VCTRL_IMM REGULATOR=LDO2 VCTRL=0x00 MASK=0xEE DELAY_MODE=ALWAYS DELAY=0us
REG_WRITE_MASK_IMM ADDR=0x3E DATA=0x00 MASK=0xFB
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
TRIG_MASK 0xFF27FF8
END
```

7 Application Examples

This section provides examples of how to interact with the PMICs from the perspective of the MCU and over I^{2}C. Table 7-1 shows how the I^{2}C commands are presented in the following sections. These examples, when used in conjunction with the datasheet, can be generalized and applied to other use cases.

Table 7-1. I^{2}C Instruction Format

I2C Address	Register Address	Data	Mask
0x48 or 0x4C	0x00 - 0xFF	0x00 - 0xFF	0x00 - 0xFF

7.1 Moving Between States; ACTIVE, MCU ONLY, and RETENTION

The default configuration of the NVM transitions the PMICs to the ACTIVE state when the ENABLE pin on the TPS65941310 goes high (rising edge triggered). The nINT pin goes high to indicate to the MCU that interrupts have occurred in the PMICs. After a normal power up sequence the interrupts are the ENABLE_INT and BIST_PASS_INT. The ENABLE_INT prohibits the PMICs from processing any lower priority triggers below the 'ON Request' in Table 6-1. The higher priority of the SU_ACTIVE trigger is why the PMICs are in the ACTIVE state even though the NSLEEP1 and NSLEEP2 bits are both cleared. Once the ENABLE_INT is cleared the state is defined by Table 7-2. The following sections describe the I^{2}C commands for transitioning between the different states.

Table 7-2. State Table

NSLEEP1	NSLEEP2	I2C_7	I2C_6	State
1	1	NA	NA	ACTIVE
0	1	1	NA	MCU Only with DDR Retention
0	1	0	NA	MCU Only without DDR Retention
Do not Care	0	1	NA	DDR Retention
	0	0	NA	Retention

7.1.1 ACTIVE

In this example the, PMIC is already in the ACTIVE state after a normal power up event. The PMIC is kept in the ACTIVE state by setting the NSLEEP1 and NSLEEP2 bits before clearing the ENABLE_INT.

```
Write 0x48:0x86:0x03:0xFC    // Set NSLEEP1 and NSLEEP2 in TPS65951310
Write 0x48:0x66:0x01:0xFE    // Clear BIST_PASS_INT
Write 0x48:0x65:0x26:0xD9    // Clear all potential sources of the On Request
```

7.1.2 MCU ONLY

Transitioning to the MCU ONLY state from the ACTIVE state, requires configuring the I2C_7 trigger before changing the NSLEEP bits. The configuration must be consistent between both PMICs.

```
Write 0x48:0x85:0x80:0x7F    // Set I2C_7 Trigger on TPS65941310
Write 0x4C:0x85:0x80:0x7F    // Set I2C_7 Trigger on TPS65941111
Write 0x48:0x86:0x02:0xFC    // Set NSLEEP2 to trigger TO_MCU power sequence
```

Instead of writing to the NSLEEP bits to return to the ACTIVE state, it is also possible to use the WKUP1 pin on GPIO4 or GPIO10 to return the PMIC to the ACTIVE state. Because of the similarity this is shown in the context of the RETENTION state.

7.1.3 RETENTION

As shown in Section 6.3.9, the MCU is powered off and therefore the transition out of the RETENTION to the MCU ONLY or the ACTIVE states must be configured before entering RETENTION. Similar to the MCU ONLY state the I2C_7 triggers must be set for both PMICs. Additionally, the TPS65941111 GPIO4

(H_DDR_RET_1V1), must be set before entering RETENTION. In this example GPIO4 on the TPS65941310 is used to wake the device from RETENTION to ACTIVE.

```
Write 0x48:0x85:0x80:0x7F  // I2C_7 is high
Write 0x4C:0x85:0x80:0x7F
Write 0x48:0x34:0xC0;0x3F  // Set GPIO4 to WKUP1 (goes to ACTIVE state)
Write 0x48:0x64:0x08:0xF7  // clear interrupt of gpio4, write to clear
Write 0x48:0x4F:0x00:0xF7  // unmask interrupt for GPIO4 falling edge
Write 0x4C:0x3D:0x08:0xF7  // set PMICB:GPIO4, H_DDR_RET_1V1
Write 0x48:0x86:0x00:0xFC  // trigger the TO_RETENTION power sequence
After the GPIO4 has gone low and the PMICs have returned to the ACTIVE state
Write 0x48:0x86:0x03:0xFC  // Set NSLEEPx bits for ACTIVE state
Write 0x48:0x64:0x08:0xF7  // clear interrupt of gpio4
Write 0x4C:0x3D:0x00:0xF7  // clear PMICB:GPIO4, DDR_RET
```

In this example the TPS65941310 RTC Timer is used to wake the device from RETENTION to ACTIVE.

```
Write 0x48:0x85:0x80:0x7F  // I2C_7 is high
Write 0x4C:0x85:0x80:0x7F
Write 0x48:0xC3:0x01;0xFE  // Enable Crystal
Write 0x48:0xC5:0x05:0xF8  // minute timer, enable TIMER interrupts
Write 0x48:0xC2:0x01:0xFE  // start timer, if the timer values are non-zero clear before starting
Write 0x4C:0x3D:0x08:0xF7  // set PMICB:GPIO4, H_DDR_RET_1V1
Write 0x48:0x86:0x00:0xFC  // trigger the TO_RETENTION power sequence
After the RTC Timer interrupt has occurred and the PMICs have returned to the ACTIVE state
Write 0x48:0x86:0x03:0xFC  // Set NSLEEPx bits for ACTIVE state
Write 0x48:0xC5:0x00:0xFB  // disable timer interrupt, clear bit 2
Write 0x48:0xC4:0x00:0xDF  // clear timer interrupt, clear bit 5
Write 0x4C:0x3D:0x00:0xF7  // clear PMICB:GPIO4, DDR_RET
```

7.2 Entering and Exiting Standby

STANDBY can be entered from ACTIVE, MCU ONLY, or the RETENTION states. In order to stay in the mission state of STANDBY and not enter the hardware state LP_STANDBY the LP_STANDBY_SEL bit must be cleared.

Similar to the RETENTION state the STANDBY state turns off all regulators which power the MCU. Therefore, it is required to select the state, MCU ONLY or ACTIVE, that the STANDBY state returns to.

When the ENABLE pin goes low, the TO_STANDBY sequence is triggered. When the ENABLE pin goes high again, the destination state is dependent upon the STARTUP_DEST bits. The TO_STANDBY sequence is also triggered by the I2C_0 trigger. When triggered from I2C_0 the PMIC can be triggered to return to either the ACTIVE or MCU ONLY states by GPIO4, GPIO10, or and RTC timer or alarm. In this example, I2C_0 trigger is used to enter the STANDBY state and the GPIO4 is used to enter the ACTIVE state.

```
Write 0x48:0xC3:0x00:0xF7  // LP_STANDBY_SEL=0
Write 0x48:0x7D:0xC0:0x3F  // Mask NSLEEP bits
Write 0x48:0x34:0xC0;0x3F  // Set GPIO4 to WKUP1 (goes to ACTIVE state)
Write 0x48:0x64:0x08:0xF7  // clear interrupt of GPIO4
Write 0x48:0x4F:0x00:0xF7  // unmask interrupt for GPIO4 falling edge
Write 0x48:0x85:0x01:0xFE  // set I2C_0 trigger, trigger TO_STANDBY sequence
After the GPIO4 has gone low and the PMICs have returned to the ACTIVE state
Write 0x48:0x7D:0x00:0x3F  // unmask NSLEEP bits
Write 0x48:0x86:0x03:0xFC  // Set NSLEEPx bits for ACTIVE state
Write 0x48:0x64:0x08:0xF7  // clear interrupt of GPIO4
```

7.3 Entering and Existing LP_STANDBY

Entering the LP_STANDBY hardware state is the same as entering STANDBY. Exiting LP_STANDBY is different and requires different initializations before entering LP_STANDBY. Also, when the PMICs return from LP_STANDBY the PFSM triggers are gated by the ENABLE_INT while in STANDBY the triggers were gated by the GPIO interrupt.

```
Write 0x48:0xC3:0x08:0xF7  // LP_STANDBY_SEL=1
Write 0x48:0x7D:0xC0:0x3F  // Mask NSLEEP bits
Write 0x48:0x34:0xC0;0x3F  // Set GPIO4 to WKUP1 (goes to ACTIVE state)
Write 0x48:0xC3:0x60;0x9F  // Set the STARTUP_DEST=ACTIVE
Write 0x48:0x64:0x08:0xF7  // clear interrupt of GPIO4
Write 0x48:0x4F:0x00:0xF7  // unmask interrupt for GPIO4 falling edge
Write 0x48:0x85:0x01:0xFE  // set I2C_0 trigger, trigger TO_STANDBY sequence
After the GPIO4 has gone low and the PMICs have returned to the ACTIVE state
Write 0x48:0x7D:0x00:0x3F  // unmask NSLEEP bits
Write 0x48:0x86:0x03:0xFC  // Set NSLEEPx bits for ACTIVE state
Write 0x48:0x64:0x08:0xF7  // clear interrupt of GPIO4
Write 0x48:0x65:0x02:0xFD  // clear ENABLE_INT
```

7.4 Runtime Customization

The TPS65941310 GPIO8 is configured as an input to disable the watchdog. Typically, during development this pin is tied high, so that when the nRSTOUT bit is set WD_PWRHOLD is also set. The configuration of this pin can be utilized for other features or functions but this requires servicing the watchdog before it expires. The watchdog long window is 772 seconds, Table 5-13.

```
Write 0x12:0x09:0x00:0xBF  // Disable Watchdog
Write 0x48:0x38:0x01:0x00  // configure GPIO8 as a pushpull output
```

When it is time to enable and configure the watchdog, then in addition to enabling the watchdog the WD_PWR_HOLD must be cleared.

```
Write 0x12:0x09:0x00:0xFB  // Clear WD_PWRHOLD
Write 0x12:0x09:0x40:0xBF  // Enable Watchdog
```

In addition to the GPIO8 of the TPS65941310 there are also the feedback pins for BUCK3 and BUCK4 on the TPS65941111. These monitors can be used independently since the BUCK3 and BUCK4 regulators are multiphased with BUCKs 1 and 2. When enabling a monitor, the built in self-test is performed. Please refer to the datasheet for an explanation of the monitor self-test. If the self-test fails this results in a Moderate error which triggers the TO_SAFE_ORDERLY power sequence.

Unlike the GPIO, the BUCK monitor can become part of the PFSM by assigning a group to the BUCK regulator and unmasking the OV/UV interrupts. Per the Table 5-7 the BUCK3_GRP_SEL and BUCK4_GRP_SEL are not assigned a group.

Table 7-3. Rail Group Associations

Selected Rail group Selection	PFSM Trigger	Description
No Group Assigned	None	OV/UV can set nINT pin for MCU interrogation.
MCU Rail Group	MCU Power Error	OV/UV can trigger TO_SAFE
Soc Rail Group	SoC Power Error	OV/UV can trigger PWR_SOC_ERROR
Other Rail Group	Orderly Shutdown	OV/UV can trigger TO_SAFE_ORDERLY

In this example BUCK3 is used to monitor a 1.1V supply and BUCK4 is used to monitor a 0.8V supply. The wait statement ensures that the built in self-test of the monitors is completed before the OV and UV monitors are unmasked. Refer to the *TPS6594-Q1Power Management IC (PMIC) with 5 Bucks and 4 LDOs for Safety-Relevant Automotive Applications* datasheet for more details.

```
Write 0x4C:0x12:0x73:0x00  // Set to 1.1V
Write 0x4C:0x14:0x37:0x00  // Set to 0.8V
Write 0x4C:0x09:0x07:0xF1  // Set slew rate to 0.31mV/us
Write 0x4C:0x0B:0x07:0xF1
Write 0x4C:0x41:0xA0:0x0F  // SOC rail group
Write 0x4C:0x4A:0x33:0xCC  // Mask OV/UV
Write 0x4C:0x08:0x10:0xEF  // Enable BUCK3 Monitor
Write 0x4C:0x0A:0x10:0xEF  // Enable BUCK4 Monitor
// Startup = 220us, ramp = 42us, settling = 105us, OV/UV test=50us
wait 500us
Write 0x4C:0x4A:0x00:0xCC  // Unmask OV/UV
```

With the TO_SAFE and TO_SAFE_ORDERLY sequences the PMICs transition through the SAFE RECOVERY state as well as hardware states INIT and BOOT BIST. Through this transition the user registers are restored with the NVM settings. For both the GPIO and BUCK monitor customizations, these customizations are not preserved and must be re-applied with every power cycle and transition through the hardware states.

8 References

For additional information regarding the PMIC or processor devices, use the following:

- Texas Instruments, *DRA829 Jacinto™ Processors Silicon Revisions 1.0 and 1.1* data sheet
- Texas Instruments, DRA829 Safety Manual Jacinto™ 7 Processors (request through mySecure)
- Texas Instruments, *DRA829/TDA4VM/AM752x Technical Reference Manual (Rev. B)* reference model
- Texas Instruments, *TPS6594-Q1 Power Management IC (PMIC) with 5 Bucks and 4 LDOs for Safety-Relevant Automotive Applications* data sheet
- Texas Instruments, TPS6594-Q1 Safety Manual (request through mySecure)
- Texas Instruments, *TPS6594-Q1 Schematic PCB Checklist* application note

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