User's Guide

TPS65941319-Q1 PMIC User Guide for AM65 Sitara™ Processors

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 industrial AM65x Sitara™ processor.

Table of Contents

Trademarks

Sitara™ 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) using a TPS6594-Q1 device to supply the AM65x processor with independent MCU and Main power rails. This PDN enables board level isolation of the MCU safety island and main voltage resources as required for implementing two desirable features of the processor:

- MCU processor acts as independent safety monitor (MCU Safety Island) over the Main processing resources to maintain safe system operations.
- 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:

- PDN power resource connections
- PDN digital control connections
- Primary and secondary PMIC static NVM contents
- 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. TPS6594-Q1 Orderable Part Number for the AM65x Processor

Features Supported	Orderable Part Number	TI_NVM_ID	TI_NVM_REV
• AM65Superset Use Case with 1.1GHz (Turbo) clock & Peak Power est. • Supports AM65x Junction Temperature (Tj) up to 105C. • Functional Safety up to max SIL-3 level with SoC rails monitored. • (Optional) System Low Power Modes: MCU Only, Suspend-to-RAM (S2R) • Support DDR3L, DDR4, and LPDDR4 memory types • (Optional) Provides Low Frequency Clock Source (32KHz) • Supports UHS-I SD Cards (Dual-Voltage IO) • Supports both 3.3v and 1.8V IO	TPS65941319RWERQ1	0x19	0x01

Table 2-1. TPS6594-Q1 Orderable Part Number for the AM65x Processor (continued)

Features Supported	Orderable Part Number	TI_NVM_ID	TI_NVM_REV
• Has capacity on IO rails to support variety of peripheral (Octal SPI, eMMC, and so on) • (Optional) Supports Efuse in-system programming • (Optional) External components (Step-down converters, LDO, and so on) can be replaced with customer's own selections	TPS65941319RWERQ1	0x19	0x01

3 Processor Connections

This section details how the 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 TPS6594-Q1 PMIC power resources and processor voltage domains required to support independent MCU and Main power rails. In this configuration, the PMIC uses a 3.3V input voltage. For safety applications, there is a protection FET before VCCA that connects to the OVPGDRV pin of the primary PMIC, allowing voltage monitoring and control of the input supply to the PMICs.

This PDN uses five discrete power components with four being required and one is optional depending if the efuse feature is needed. The three load switches in addition to the BUCK4 regulator create independent IO power rails with the following benefits:

- Sequencing SoC power domains in desired order by using PMIC GPIO control signals with desired start-up & shut-down timing delays shown in the Section 6.3 section.
- PMIC monitoring of VCCA Over Voltage (see Section 5.5) allows PMIC GPIOs to disconnect these 3.3V power rails from SoC if OV is detected.
- Required for low power modes (MCU Only and S2R Retention) since disabling/monitoring the IO domain of the SoC independently is required.

Note

The load switches selected (TPS22919) are for light loads (< 500mA). For larger loading, it is recommended to use TPS22965 or similar.

The fourth discrete device is a TPS628502 Buck Converter which supplies the LPDDR4 SDRAM component with required 1.1V supply. The last discrete power component is an optional TLV70018-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.

Note

The processor support multiple DDR memory types including DDR3L, DDR4, and LPDDR4. Each of these memory technologies requires different voltages to operate. For this reason, the regulator for the DDR voltage was purposely not included within the PMIC. The PMIC does include control signal to enable/disable an external DDR regulator in the proper sequence.

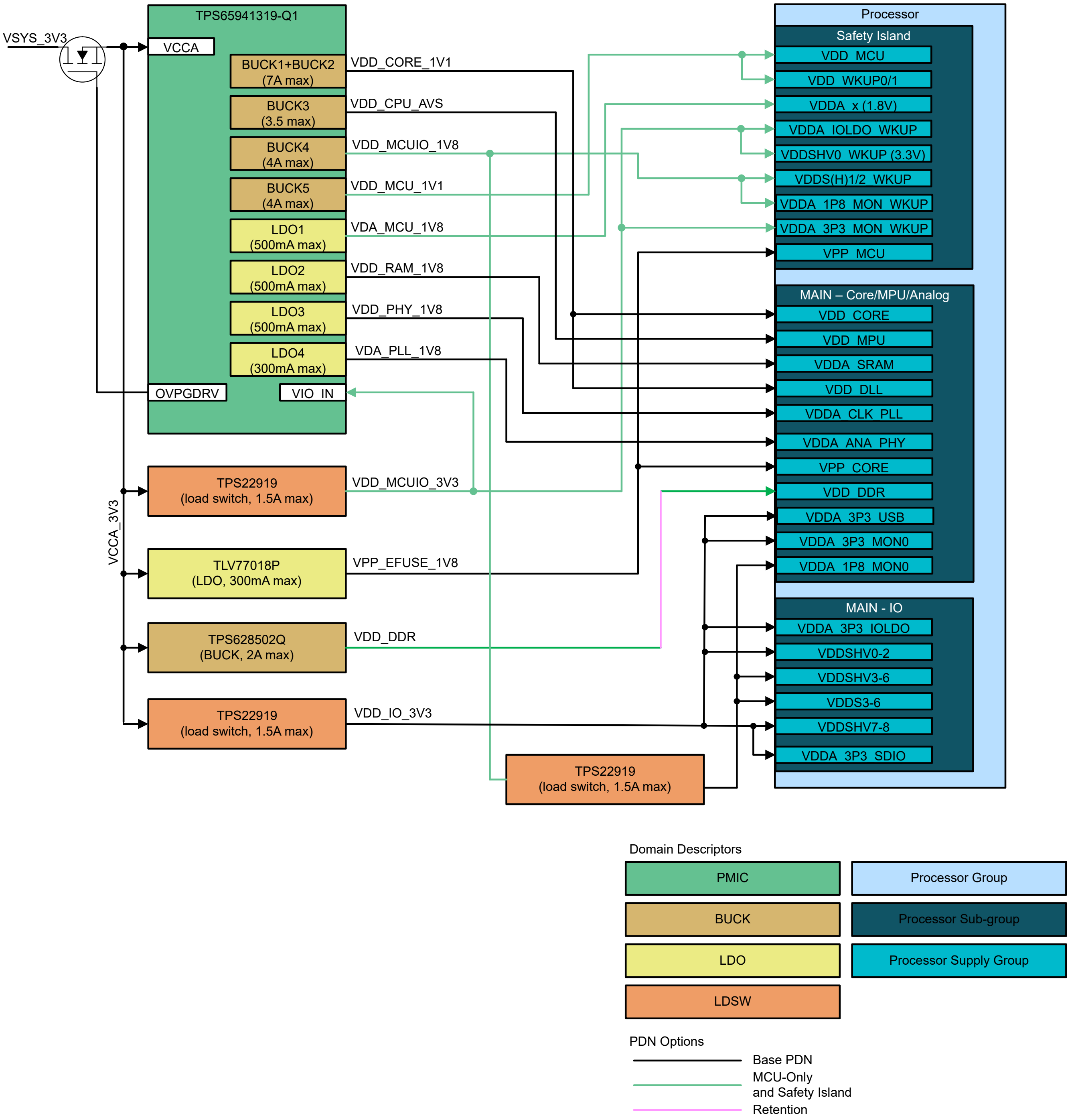


Figure 3-1. Power Connections

- VDD_CPU_AVS, boot voltage of 1.1V then software sets device specific AVS.

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.

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**
TPS65941319-Q1	BUCK12	VDD_CORE_1V1	VDD_CORE,VDD_DLL_MMC0/1	R		
	BUCK3	VDD_CPU	VDD_MPU0/1	R		
	BUCK4	VDD_MCUIO_1V8	VDDS1/2_WKUP,VCCSHV1/2_WKUP,VDDA_1P8_MON_WKUP	R	R	
	BUCK5	VDD_MCU_1V1	VDD_MCU, VDD_WKUP0/1	R	R	
	LDO1	VDA_MCU_1V8	VDDA_LDO_WKUP, VDDA_MCU/WKUP, VDDA_ADC_MCU, VDDA_POR_WKUP	R	R	
	LDO2	VDD_RAM_1V8	VDDA_SRAM_CORE0/1,VDDA_SRAM_MPU0/1, VDDA_1P8_OLDI0	R		
	LDO3	VDD_PHY_1V8	VDDA_1P8_CSI0, VDDA_1P8_SERDES0	R		
	LDO4	VDA_PLL_1V8	VDDA_PLL_CORE, VDDA_PLL0/1_DDR, VDDA_PLL_MPU0/1, VDDA_PLL_DSS, VDDA_PLL_PER0, VDDS_OSC1	R		
TPS22919	Load Switch	VDD_MCUIO_3V3	VDDA_3P3_IOLDO_WKUP, VDDSHV0_WKUP, VDDA_3P3_MON_WKUP	R[2]	R	
TPS22919	Load Switch	VDD_IO_3V3	VDDA_3P3_USB, VDDA_3P3_MON0, VDDA_3P3_IOLDO0/1, VDDSHV0-2,VDDSHV7-8, VDDA_3P3_SDIO	R[3]		
TPS22919	Load Switch	VDD_IO_1V8	VDDA_1P8_MON0, VDDS3-6,VDDSHV3-6,	R[4]		
TLV70018	LDO	VPP_EFUSE_1V8	VPP_x(EFUSE)	O		
TPS628502Q	BUCK	VDD_DDR	VDDS_DDR	R	R[5]	

[1] 'R' is required and 'O' is optional. If left 'blank' then the regulator is not enabled during the mode.
[2] The TPS22919 supplying VDD_MCUIO_3V3 is controlled by TPS65941319-Q1 GPIO3.
[3] The TPS22919 supplying VDD_IO_3V3 is controlled by TPS65941319-Q1 GPIO5.
[4] The TPS22919 supplying VDD_IO_1V8 is controlled by TPS65941319-Q1 GPIO6.
[5] The TPS628502Q is controlled by the TPS65941319-Q1 GPIO4 and remains active while TRIGGER_I2C_7, in FSM_I2C_TRIGGERS, is set.

3.2 Control Mapping

Figure 3-2 shows the digital control signal mapping between processor and PMIC devices. Connections from the TPS6594-Q1 PMICs to the processor provide error monitoring, processor reset, processor wake up, and system low-power modes.

The digital connections shown in Figure 3-2 allow system features including 'MCU-only, MCU Safety Island' and Retention modes, and functional safety up to SIL-3.

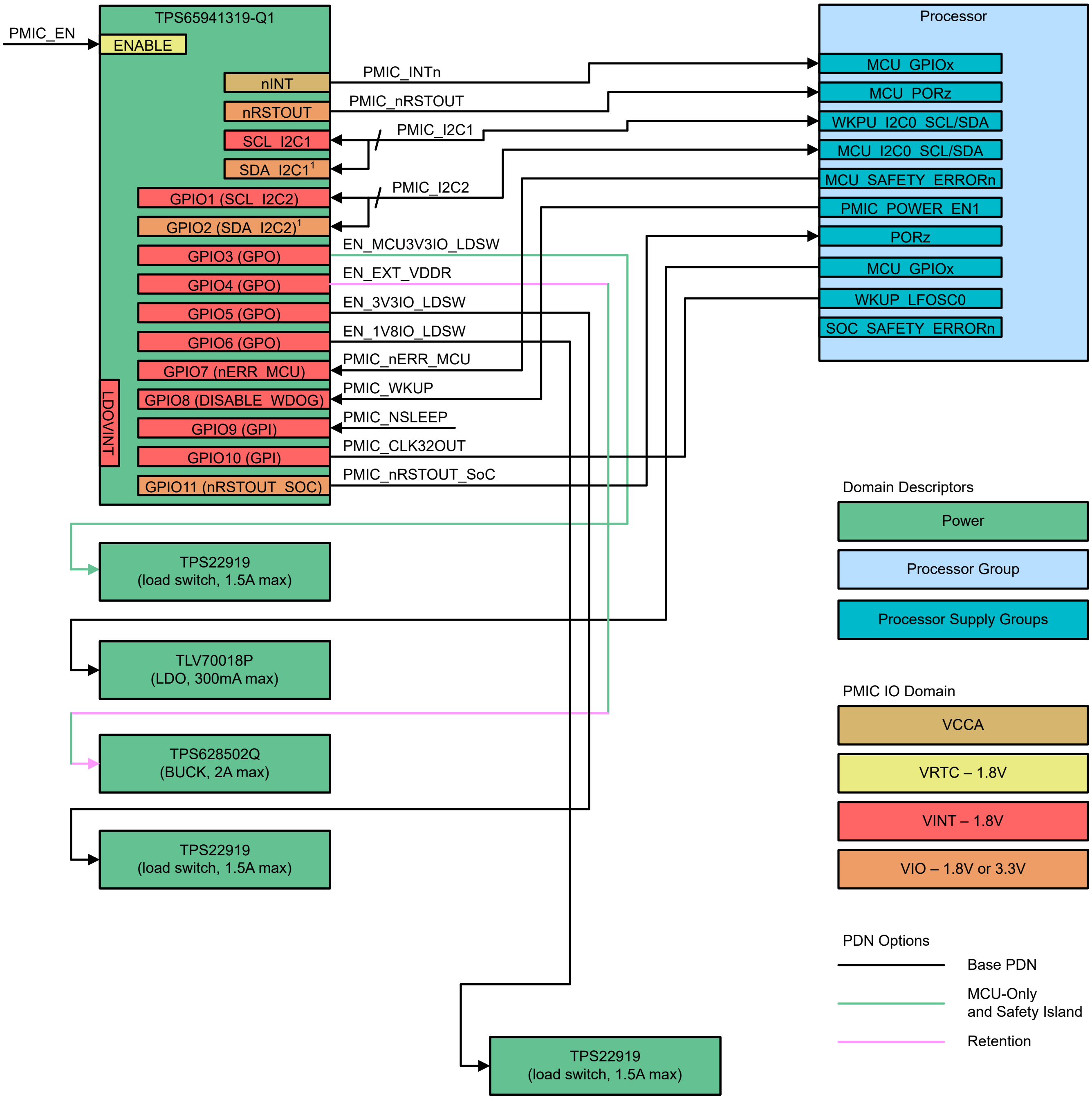


Figure 3-2. TPS6594-Q1 Digital Connections

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. Refer to the device data sheet for a complete description.
2. GPIO8 is configured as the DISABLE_WDOG pin. When the PMIC sets nRSTOUT, the logic level of GPIO8 is latched into the WD_PWRHOLD bit. If low, then the watchdog enters the long window and the processor must service the WDOG before the long window expires or the PMIC performs a warm reset of the processor. Once the WDOG is serviced, then the control of the WDOG can be maintained through the I2C2 and the GPIO8 can be repurposed for WKUP1 or WKUP2. WKUP1 and WKUP2 are not functional from LP_STANDBY.
3. GPIO9, NSLEEP1 or NSLEEP2, is not connected to the processor and is not part of the PDN. If the customer chooses to use this GPIO, the function must be defined at runtime.

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. 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
PMIC_INTn	VDD_MCUIO_3V3
PMIC_nRSTOUT	VDA_MCU_1V8
PMIC_nRSTOUT_SOC	VDA_MCU_1V8
PMIC_I2C1	VDD_MCUIO_3V3
PMIC_I2C2	VDD_MCUIO_3V3

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. Reconfiguration of the GPIO function 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).

Table 3-3. Digital Connections by System Feature

Device	GPIO Mapping			System Features[1]			
	PMIC Pin	NVM Function	PDN Signals	Functional Safety	Active SoC	MCU - Only MCU-Safety Island	Retention
TPS65941319-Q1	nPWRON/ ENABLE	Enable	PMIC_ENABLE		R		R
	INT	INT	PMIC_INTn	R			
	nRSTOUT	nRSTOUT	PMIC_nRSTOUT	R	R	R	
	SCL_I2C1	SCL_I2C1	PMIC_I2C1[2]	R			
	SDA_I2C1	SDA_I2C1					
	GPIO_1	SCL_I2C2	PMIC_I2C2	R			
	GPIO_2	SDA_I2C2					
	GPIO_3	GPO	EN_MCU3V3IO_LDSW		R		
	GPIO_4	GPO	EN_EXT_VDDR		R	O	
	GPIO_5	GPO	EN_3V3IO_LDSW		R		
	GPIO_6	GPO	EN_1V8IO_LDSW		R		
	GPIO_7	nERR_MCU	PMIC_nERR_MCU	R			
	GPIO_8	DISABLE_WDOG[3]	PMIC_WKUP	R			
	GPIO_9	GPI	PMIC_nSLEEP				
	GPIO_10	GPI	PMIC_CLK32OUT				
	GPIO_11	nRSTOUT_SOC	PMIC_nRSTOUT_SoC	R			

(1) R is Required. O is optional. If left blank then the resource is not used to support the feature.
(2) I2C1 is the primary I^{2}C communication and is required for functional safety. This communication channel is used to interrogate and clear interrupts as well as interface with the error signal monitor (ESM). The I^{2}C is also used to transition between ACTIVE, MCU_ONLY, and RETENTION states. Alternatively, GPIO9, PMIC_nSLEEP is used to transition between states.
(3) If 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.

4 Supporting Functional Safety Systems

The TPS65941319 offers the following functional safety features:

- Independent Power Control of MCU and Main Rails
- Independent Monitoring and Reset for MCU (nRSTOUT) and Main Rails (nRSTOUT_SOC)
- 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.

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 data sheet.

5.1 Application-Based Configuration Settings

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

- 2.2MHz Single Phase for DDR Termination
- 4.4MHz VOUT Less than 1.9V, Multiphase or High COUT Single Phase
- 4.4MHz VOUT Less than 1.9V, Low COUT, Single Phase Only
- 4.4MHz VOUT Greater than 1.7V, Single Phase Only
- 2.2MHz VOUT Less than 1.9V Multiphase or Single Phase
- 2.2MHz Full VOUT Range and VIN Greater than 4.5V, Single Phase Only
- 2.2MHz 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
TPS65941319-Q1	BUCK1	2.2MHz VOUT Less than 1.9V Multiphase or Single Phase	470nH
	BUCK2	2.2MHz VOUT Less than 1.9V Multiphase or Single Phase	470nH
	BUCK3	4.4MHz VOUT Less than 1.9V, Multiphase or High COUT Single Phase	220nH
	BUCK4	4.4MHz VOUT Less than 1.9V, Multiphase or High COUT Single Phase	220nH
	BUCK5	4.4MHz VOUT Less than 1.9V, Multiphase or High COUT Single Phase	220nH

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

		TPS65941319-Q1	
Register Name	Field Name	Value	Description
DEV_REV	DEVICE_ID	0x2	
NVM_CODE_1	TI_NVM_ID	0x19	
NVM_CODE_2	TI_NVM_REV	0x1	
PHASE_CONFIG	MP_CONFIG	0x2	2+1+1+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 I2C 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. The BUCKx_VMON_EN bit is set for BUCK1, BUCK3, BUCK4 and BUCK5. The BUCKx_RV_SEL bit is cleared for all BUCKs. The other bits remain unchanged, but are still accessible via I2C.

Table 5-3. BUCK NVM Settings

		TPS65941319-Q1	
Register Name	**Field Name**	**Value**	**Description**
BUCK1_CTRL	BUCK1_EN	0x0	Disabled; BUCK1 regulator
	BUCK1_FPWM	0x0	PFM and PWM operation (AUTO mode).
	BUCK1_FPWM_MP	0x0	Automatic phase adding and shedding.
	BUCK1_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK1_VSEL	0x0	BUCK1_VOUT_1
	BUCK1_PLDN	0x1	Enabled; Pull-down resistor
	BUCK1_RV_SEL	0x1	Enabled
BUCK1_CONF	BUCK1_SLEW_RATE	0x3	5.0 mV/µs
	BUCK1_ILIM	0x5	5.5 A
BUCK2_CTRL	BUCK2_EN	0x0	Disabled; BUCK2 regulator
	BUCK2_FPWM	0x0	PFM and PWM operation (AUTO mode).
	BUCK2_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK2_VSEL	0x0	BUCK2_VOUT_1
	BUCK2_PLDN	0x1	Enabled; Pull-down resistor
	BUCK2_RV_SEL	0x1	Enabled
BUCK2_CONF	BUCK2_SLEW_RATE	0x3	5.0 mV/µs
	BUCK2_ILIM	0x5	5.5 A
BUCK3_CTRL	BUCK3_EN	0x0	Disabled; BUCK3 regulator
	BUCK3_FPWM	0x0	PFM and PWM operation (AUTO mode).
	BUCK3_FPWM_MP	0x0	Automatic phase adding and shedding.
	BUCK3_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK3_VSEL	0x0	BUCK3_VOUT_1
	BUCK3_PLDN	0x1	Enabled; Pull-down resistor
	BUCK3_RV_SEL	0x1	Enabled
BUCK3_CONF	BUCK3_SLEW_RATE	0x3	5.0 mV/µs
	BUCK3_ILIM	0x5	5.5 A
BUCK4_CTRL	BUCK4_EN	0x0	Disabled; BUCK4 regulator
	BUCK4_FPWM	0x0	PFM and PWM operation (AUTO mode).
	BUCK4_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK4_VSEL	0x0	BUCK4_VOUT_1
	BUCK4_PLDN	0x1	Enabled; Pull-down resistor
	BUCK4_RV_SEL	0x1	Enabled
BUCK4_CONF	BUCK4_SLEW_RATE	0x3	5.0 mV/µs
	BUCK4_ILIM	0x5	5.5 A

Table 5-3. BUCK NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
BUCK5_CTRL	BUCK5_EN	0x0	Disabled; BUCK5 regulator
	BUCK5_FPWM	0x0	PFM and PWM operation (AUTO mode).
	BUCK5_VMON_EN	0x0	Disabled; OV, UV, SC and ILIM comparators.
	BUCK5_VSEL	0x0	BUCK5_VOUT_1
	BUCK5_PLDN	0x1	Enable Pull-down resistor
	BUCK5_RV_SEL	0x1	Enabled
BUCK5_CONF	BUCK5_SLEW_RATE	0x3	5.0 mV/µs
	BUCK5_ILIM	0x3	3.5 A
BUCK1_VOUT_1	BUCK1_VSET1	0x73	1.10 V
BUCK1_VOUT_2	BUCK1_VSET2	0x0	0.3 V
BUCK2_VOUT_1	BUCK2_VSET1	0x73	1.10 V
BUCK2_VOUT_2	BUCK2_VSET2	0x0	0.3 V
BUCK3_VOUT_1	BUCK3_VSET1	0x73	1.10 V
BUCK3_VOUT_2	BUCK3_VSET2	0x0	0.3 V
BUCK4_VOUT_1	BUCK4_VSET1	0xb2	1.80 V
BUCK4_VOUT_2	BUCK4_VSET2	0x0	0.3 V
BUCK5_VOUT_1	BUCK5_VSET1	0x73	1.10 V
BUCK5_VOUT_2	BUCK5_VSET2	0x0	0.3 V
BUCK1_PG_WINDOW	BUCK1_OV_THR	0x3	+5% / +50 mV
	BUCK1_UV_THR	0x3	-5% / -50 mV
BUCK2_PG_WINDOW	BUCK2_OV_THR	0x3	+5% / +50 mV
	BUCK2_UV_THR	0x3	-5% / -50 mV
BUCK3_PG_WINDOW	BUCK3_OV_THR	0x3	+5% / +50 mV
	BUCK3_UV_THR	0x3	-5% / -50 mV
BUCK4_PG_WINDOW	BUCK4_OV_THR	0x3	+5% / +50 mV
	BUCK4_UV_THR	0x3	-5% / -50 mV
BUCK5_PG_WINDOW	BUCK5_OV_THR	0x3	+5% / +50 mV
	BUCK5_UV_THR	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	TPS65941319-Q1	
		Value	**Description**
LDO1_CTRL	LDO1_EN	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
	LDO1_PLDN	0x1	125 Ohm
	LDO1_VMON_EN	0x0	Disable OV and UV comparators.
	LDO1_RV_SEL	0x1	Enabled
LDO2_CTRL	LDO2_EN	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
	LDO2_PLDN	0x1	125 Ohm
	LDO2_VMON_EN	0x0	Disabled; OV and UV comparators.
	LDO2_RV_SEL	0x1	Enabled
LDO3_CTRL	LDO3_EN	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
	LDO3_PLDN	0x1	125 Ohm
	LDO3_VMON_EN	0x0	Disabled; OV and UV comparators.
	LDO3_RV_SEL	0x1	Enabled
LDO4_CTRL	LDO4_EN	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
	LDO4_PLDN	0x1	125 Ohm
	LDO4_VMON_EN	0x0	Disabled; OV and UV comparators.
	LDO4_RV_SEL	0x1	Enabled
LDO1_VOUT	LDO1_VSET	0x1c	1.80 V
	LDO1_BYPASS	0x0	Linear regulator mode.
LDO2_VOUT	LDO2_VSET	0x1c	1.80 V
	LDO2_BYPASS	0x0	Linear regulator mode.
LDO3_VOUT	LDO3_VSET	0x1c	1.80 V
	LDO3_BYPASS	0x0	Linear regulator mode.
LDO4_VOUT	LDO4_VSET	0x38	1.800 V
LDO1_PG_WINDOW	LDO1_OV_THR	0x3	+5% / +50 mV
	LDO1_UV_THR	0x3	-5% / -50 mV
LDO2_PG_WINDOW	LDO2_OV_THR	0x3	+5% / +50 mV
	LDO2_UV_THR	0x3	-5% / -50 mV
LDO3_PG_WINDOW	LDO3_OV_THR	0x3	+5% / +50 mV
	LDO3_UV_THR	0x3	-5% / -50 mV
LDO4_PG_WINDOW	LDO4_OV_THR	0x3	+5% / +50 mV
	LDO4_UV_THR	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	TPS65941319-Q1	
		Value	Description
VCCA_VMON_CTRL	VMON_DEGLITCH_SEL	0x1	20 us
	VCCA_VMON_EN	0x1	Enabled; OV and UV comparators.
VCCA_PG_WINDOW	VCCA_OV_THR	0x7	+10%
	VCCA_UV_THR	0x7	-10%
	VCCA_PG_SET	0x0	3.3 V
GENERAL_REG_1	FAST_VCCA_OVP	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

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	TPS65941319-Q1	
		Value	Description
GPIO1_CONF	GPIO1_OD	0x0	Push-pull output
	GPIO1_DIR	0x0	Input
	GPIO1_SEL	0x1	SCL_I2C2/CS_SPI
	GPIO1_PU_SEL	0x0	Pull-down resistor selected
	GPIO1_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO1_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO2_CONF	GPIO2_OD	0x0	Push-pull output
	GPIO2_DIR	0x0	Input
	GPIO2_SEL	0x2	SDA_I2C2/SDO_SPI
	GPIO2_PU_SEL	0x0	Pull-down resistor selected
	GPIO2_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO2_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO3_CONF	GPIO3_OD	0x0	Push-pull output
	GPIO3_DIR	0x1	Output
	GPIO3_SEL	0x0	GPIO3
	GPIO3_PU_SEL	0x0	Pull-down resistor selected
	GPIO3_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO3_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO4_CONF	GPIO4_OD	0x0	Push-pull output
	GPIO4_DIR	0x1	Output
	GPIO4_SEL	0x0	GPIO4
	GPIO4_PU_SEL	0x0	Pull-down resistor selected
	GPIO4_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO4_DEGLITCH_EN	0x0	No deglitch, only synchronization.

Table 5-6. GPIO NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
GPIO5_CONF	GPIO5_OD	0x0	Push-pull output
	GPIO5_DIR	0x1	Output
	GPIO5_SEL	0x0	GPIO5
	GPIO5_PU_SEL	0x0	Pull-down resistor selected
	GPIO5_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO5_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO6_CONF	GPIO6_OD	0x0	Push-pull output
	GPIO6_DIR	0x1	Output
	GPIO6_SEL	0x0	GPIO6
	GPIO6_PU_SEL	0x0	Pull-down resistor selected
	GPIO6_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO6_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO7_CONF	GPIO7_OD	0x0	Push-pull output
	GPIO7_DIR	0x0	Input
	GPIO7_SEL	0x1	NERR_MCU
	GPIO7_PU_SEL	0x0	Pull-down resistor selected
	GPIO7_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO7_DEGLITCH_EN	0x1	8 us deglitch time.
GPIO8_CONF	GPIO8_OD	0x0	Push-pull output
	GPIO8_DIR	0x0	Input
	GPIO8_SEL	0x3	DISABLE_WDOG
	GPIO8_PU_SEL	0x0	Pull-down resistor selected
	GPIO8_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO8_DEGLITCH_EN	0x1	8 us deglitch time.
GPIO9_CONF	GPIO9_OD	0x0	Push-pull output
	GPIO9_DIR	0x0	Input
	GPIO9_SEL	0x0	GPIO9
	GPIO9_PU_SEL	0x0	Pull-down resistor selected
	GPIO9_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO9_DEGLITCH_EN	0x1	8 us deglitch time.
GPIO10_CONF	GPIO10_OD	0x0	Push-pull output
	GPIO10_DIR	0x0	Input
	GPIO10_SEL	0x0	GPIO10
	GPIO10_PU_SEL	0x0	Pull-down resistor selected
	GPIO10_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.
	GPIO10_DEGLITCH_EN	0x0	No deglitch, only synchronization.
GPIO11_CONF	GPIO11_OD	0x1	Open-drain output
	GPIO11_DIR	0x1	Output
	GPIO11_SEL	0x2	NRSTOUT_SOC
	GPIO11_PU_SEL	0x0	Pull-down resistor selected
	GPIO11_PU_PD_EN	0x0	Disabled; Pull-up/pull-down resistor.
	GPIO11_DEGLITCH_EN	0x0	No deglitch, only synchronization.

Table 5-6. GPIO NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	**Description**
NPWRON_CONF	NPWRON_SEL	0x0	ENABLE
	ENABLE_PU_SEL	0x0	Pull-down resistor selected
	ENABLE_PU_PD_EN	0x1	Enabled; Pull-up/pull-down resistor.
	ENABLE_DEGLITCH_EN	0x1	8 us deglitch time when ENABLE, 50 ms deglitch time when NPWRON.
	ENABLE_POL	0x0	Active high
	NRSTOUT_OD	0x1	Open-drain output
GPIO_OUT_1	GPIO1_OUT	0x0	Low
	GPIO2_OUT	0x0	Low
	GPIO3_OUT	0x0	Low
	GPIO4_OUT	0x0	Low
	GPIO5_OUT	0x0	Low
	GPIO6_OUT	0x0	Low
	GPIO7_OUT	0x0	Low
	GPIO8_OUT	0x0	Low
GPIO_OUT_2	GPIO9_OUT	0x0	Low
	GPIO10_OUT	0x0	Low
	GPIO11_OUT	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	TPS65941319-Q1	
		Value	**Description**
RAIL_SEL_1	BUCK1_GRP_SEL	0x2	SOC rail group
	BUCK2_GRP_SEL	0x2	SOC rail group
	BUCK3_GRP_SEL	0x2	SOC rail group
	BUCK4_GRP_SEL	0x1	MCU rail group
RAIL_SEL_2	BUCK5_GRP_SEL	0x1	MCU rail group
	LDO1_GRP_SEL	0x1	MCU rail group
	LDO2_GRP_SEL	0x2	SOC rail group
	LDO3_GRP_SEL	0x2	SOC rail group
RAIL_SEL_3	LDO4_GRP_SEL	0x2	SOC rail group
	VCCA_GRP_SEL	0x1	MCU rail group
FSM_TRIG_SEL_1	MCU_RAIL_TRIG	0x2	MCU power error
	SOC_RAIL_TRIG	0x3	SOC power error
	OTHER_RAIL_TRIG	0x1	Orderly shutdown
	SEVERE_ERR_TRIG	0x0	Immediate shutdown
FSM_TRIG_SEL_2	MODERATE_ERR_TRIG	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 I2C after startup.

Table 5-8. Interrupt NVM Settings

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
FSM_TRIG_MASK_1	GPIO1_FSM_MASK	0x1	Masked
	GPIO1_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO2_FSM_MASK	0x1	Masked
	GPIO2_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO3_FSM_MASK	0x1	Masked
	GPIO3_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO4_FSM_MASK	0x1	Masked
	GPIO4_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
FSM_TRIG_MASK_2	GPIO5_FSM_MASK	0x1	Masked
	GPIO5_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO6_FSM_MASK	0x1	Masked
	GPIO6_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO7_FSM_MASK	0x1	Masked
	GPIO7_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO8_FSM_MASK	0x1	Masked
	GPIO8_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
FSM_TRIG_MASK_3	GPIO9_FSM_MASK	0x1	Masked
	GPIO9_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO10_FSM_MASK	0x1	Masked
	GPIO10_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
	GPIO11_FSM_MASK	0x1	Masked
	GPIO11_FSM_MASK_POL	0x0	Low; Masking sets signal value to '0'
MASK_BUCK1_2	BUCK1_ILIM_MASK	0x0	Interrupt generated
	BUCK1_OV_MASK	0x0	Interrupt generated
	BUCK1_UV_MASK	0x0	Interrupt generated
	BUCK2_ILIM_MASK	0x0	Interrupt generated
	BUCK2_OV_MASK	0x0	Interrupt generated
	BUCK2_UV_MASK	0x0	Interrupt generated
MASK_BUCK3_4	BUCK3_ILIM_MASK	0x0	Interrupt generated
	BUCK3_OV_MASK	0x0	Interrupt generated
	BUCK3_UV_MASK	0x0	Interrupt generated
	BUCK4_OV_MASK	0x0	Interrupt generated
	BUCK4_UV_MASK	0x0	Interrupt generated
	BUCK4_ILIM_MASK	0x0	Interrupt generated
MASK_BUCK5	BUCK5_ILIM_MASK	0x0	Interrupt generated
	BUCK5_OV_MASK	0x0	Interrupt generated
	BUCK5_UV_MASK	0x0	Interrupt generated

Table 5-8. Interrupt NVM Settings (continued)

		TPS65941319-Q1	
Register Name	**Field Name**	**Value**	**Description**
MASK_LDO1_2	LDO1_OV_MASK	0x0	Interrupt generated
	LDO1_UV_MASK	0x0	Interrupt generated
	LDO2_OV_MASK	0x0	Interrupt generated
	LDO2_UV_MASK	0x0	Interrupt generated
	LDO1_ILIM_MASK	0x0	Interrupt generated
	LDO2_ILIM_MASK	0x0	Interrupt generated
MASK_LDO3_4	LDO3_OV_MASK	0x0	Interrupt generated
	LDO3_UV_MASK	0x0	Interrupt generated
	LDO4_OV_MASK	0x0	Interrupt generated
	LDO4_UV_MASK	0x0	Interrupt generated
	LDO3_ILIM_MASK	0x0	Interrupt generated
	LDO4_ILIM_MASK	0x0	Interrupt generated
MASK_VMON	VCCA_OV_MASK	0x1	Interrupt not generated.
	VCCA_UV_MASK	0x1	Interrupt not generated.
MASK_GPIO1_8_FALL	GPIO1_FALL_MASK	0x1	Interrupt not generated.
	GPIO2_FALL_MASK	0x1	Interrupt not generated.
	GPIO3_FALL_MASK	0x1	Interrupt not generated.
	GPIO4_FALL_MASK	0x1	Interrupt not generated.
	GPIO5_FALL_MASK	0x1	Interrupt not generated.
	GPIO6_FALL_MASK	0x1	Interrupt not generated.
	GPIO7_FALL_MASK	0x1	Interrupt not generated.
	GPIO8_FALL_MASK	0x1	Interrupt not generated.
MASK_GPIO1_8_RISE	GPIO1_RISE_MASK	0x1	Interrupt not generated.
	GPIO2_RISE_MASK	0x1	Interrupt not generated.
	GPIO3_RISE_MASK	0x1	Interrupt not generated.
	GPIO4_RISE_MASK	0x1	Interrupt not generated.
	GPIO5_RISE_MASK	0x1	Interrupt not generated.
	GPIO6_RISE_MASK	0x1	Interrupt not generated.
	GPIO7_RISE_MASK	0x1	Interrupt not generated.
	GPIO8_RISE_MASK	0x1	Interrupt not generated.
MASK_GPIO9_11 / MASK_GPIO9_10	GPIO9_FALL_MASK	0x0	Interrupt generated
	GPIO9_RISE_MASK	0x0	Interrupt generated
	GPIO10_FALL_MASK	0x1	Interrupt not generated.
	GPIO11_FALL_MASK	0x1	Interrupt not generated.
	GPIO10_RISE_MASK	0x1	Interrupt not generated.
	GPIO11_RISE_MASK	0x1	Interrupt not generated.
MASK_STARTUP	NPWRON_START_MASK	0x1	Interrupt not generated.
	ENABLE_MASK	0x0	Interrupt generated
	FSD_MASK	0x1	Interrupt not generated.
	SOFT_REBOOT_MASK	0x0	Interrupt generated
MASK_MISC	TWARN_MASK	0x0	Interrupt generated
	BIST_PASS_MASK	0x0	Interrupt generated
	EXT_CLK_MASK	0x1	Interrupt not generated.

Table 5-8. Interrupt NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
MASK_MODERATE_ERR	BIST_FAIL_MASK	0x0	Interrupt generated
	REG_CRC_ERR_MASK	0x0	Interrupt generated
	SPMI_ERR_MASK	0x1	Interrupt not generated.
	NPWRON_LONG_MASK	0x1	Interrupt not generated.
	NINT_READBACK_MASK	0x0	Interrupt generated
	NRSTOUT_READBACK_ MASK	0x0	Interrupt generated
MASK_FSM_ERR	IMM_SHUTDOWN_MASK	0x0	Interrupt generated
	MCU_PWR_ERR_MASK	0x0	Interrupt generated
	SOC_PWR_ERR_MASK	0x0	Interrupt generated
	ORD_SHUTDOWN_MASK	0x0	Interrupt generated
MASK_COMM_ERR	COMM_FRM_ERR_MASK	0x0	Interrupt generated
	COMM_CRC_ERR_MASK	0x0	Interrupt generated
	COMM_ADR_ERR_MASK	0x0	Interrupt generated
	I2C2_CRC_ERR_MASK	0x0	Interrupt generated
	I2C2_ADR_ERR_MASK	0x0	Interrupt generated
MASK_READBACK_ERR	EN_DRV_READBACK_ MASK	0x0	Interrupt generated
	NRSTOUT_SOC_ READBACK_MASK	0x0	Interrupt generated
MASK_ESM	ESM_SOC_PIN_MASK	0x0	Interrupt generated
	ESM_SOC_RST_MASK	0x0	Interrupt generated
	ESM_SOC_FAIL_MASK	0x0	Interrupt generated
	ESM_MCU_PIN_MASK	0x0	Interrupt generated
	ESM_MCU_RST_MASK	0x0	Interrupt generated
	ESM_MCU_FAIL_MASK	0x0	Interrupt generated
GENERAL_REG_1	PFSM_ERR_MASK	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	TPS65941319-Q1	
		Value	Description
PGOOD_SEL_1	PGOOD_SEL_BUCK1	0x0	Masked
	PGOOD_SEL_BUCK2	0x0	Masked
	PGOOD_SEL_BUCK3	0x0	Masked
	PGOOD_SEL_BUCK4	0x0	Masked
PGOOD_SEL_2	PGOOD_SEL_BUCK5	0x0	Masked
PGOOD_SEL_3	PGOOD_SEL_LDO1	0x0	Masked
	PGOOD_SEL_LDO2	0x0	Masked
	PGOOD_SEL_LDO3	0x0	Masked
	PGOOD_SEL_LDO4	0x0	Masked

Table 5-9. POWERGOOD NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
PGOOD_SEL_4	PGOOD_SEL_VCCA	0x0	Masked
	PGOOD_SEL_TDIE_WARN	0x0	Masked
	PGOOD_SEL_NRSTOUT	0x0	Masked
	PGOOD_SEL_NRSTOUT_ SOC	0x0	Masked
	PGOOD_POL	0x0	PGOOD signal is high when monitored inputs are valid
	PGOOD_WINDOW	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	TPS65941319-Q1	
		Value	Description
PLL_CTRL	EXT_CLK_FREQ	0x0	1.1 MHz
CONFIG_1	TWARN_LEVEL	0x0	130C
	I2C1_HS	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.
	EN_ILIM_FSM_CTRL	0x0	Buck/LDO regulators ILIM interrupts do not affect FSM triggers.
	NSLEEP1_MASK	0x0	NSLEEP1(B) affects FSM state transitions.
	NSLEEP2_MASK	0x0	NSLEEP2(B) affects FSM state transitions.
CONFIG_2	BB_CHARGER_EN	0x0	Disabled
	BB_VEOC	0x0	2.5V
	BB_ICHR	0x0	100uA
RECOV_CNT_REG_2	RECOV_CNT_THR	0xf	0xf
BUCK_RESET_REG	BUCK1_RESET	0x0	0x0
	BUCK2_RESET	0x0	0x0
	BUCK3_RESET	0x0	0x0
	BUCK4_RESET	0x0	0x0
	BUCK5_RESET	0x0	0x0
SPREAD_SPECTRUM_1	SS_EN	0x0	Spread spectrum disabled
	SS_MODE	0x1	Mixed dwell
	SS_DEPTH	0x0	No modulation
SPREAD_SPECTRUM_2	SS_PARAM1	0x7	0x7
	SS_PARAM2	0xc	0xc
FREQ_SEL	BUCK1_FREQ_SEL	0x0	2.2 MHz
	BUCK2_FREQ_SEL	0x0	2.2 MHz
	BUCK3_FREQ_SEL	0x1	4.4 MHz
	BUCK4_FREQ_SEL	0x1	4.4 MHz
	BUCK5_FREQ_SEL	0x1	4.4 MHz
FSM_STEP_SIZE	PFSM_DELAY_STEP	0xb	0xb
LDO_RV_TIMEOUT_ REG_1	LDO1_RV_TIMEOUT	0xf	16ms
	LDO2_RV_TIMEOUT	0xf	16ms

Table 5-10. Miscellaneous NVM Settings (continued)

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
LDO_RV_TIMEOUT_ REG_2	LDO3_RV_TIMEOUT	0xf	16ms
	LDO4_RV_TIMEOUT	0xf	16ms
USER_SPARE_REGS	USER_SPARE_1	0x0	0x0
	USER_SPARE_2	0x0	0x0
	USER_SPARE_3	0x0	0x0
	USER_SPARE_4	0x0	0x0
ESM_MCU_MODE_ CFG	ESM_MCU_EN	0x0	ESM_MCU disabled.
ESM_SOC_MODE_ CFG	ESM_SOC_EN	0x0	ESM_SoC disabled.
CUSTOMER_NVM_ID_REG	CUSTOMER_NVM_ID	0x0	0x0
RTC_CTRL_2	XTAL_EN	0x1	Crystal oscillator is enabled
	LP_STANDBY_SEL	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.
	STARTUP_DEST	0x3	ACTIVE
	XTAL_SEL	0x1	9 pF
PFSM_DELAY_REG_1	PFSM_DELAY1	0x58	0x58
PFSM_DELAY_REG_2	PFSM_DELAY2	0x9d	0x9d
PFSM_DELAY_REG_3	PFSM_DELAY3	0x0	0x0
PFSM_DELAY_REG_4	PFSM_DELAY4	0x0	0x0
GENERAL_REG_0	FAST_BOOT_BIST	0x0	LBIST is run during boot BIST
GENERAL_REG_1	REG_CRC_EN	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	TPS65941319-Q1	
		Value	Description
SERIAL_IF_CONFIG	I2C_SPI_SEL	0x0	I2C
	I2C1_SPI_CRC_EN	0x0	CRC disabled
	I2C2_CRC_EN	0x0	CRC disabled
I2C1_ID_REG	I2C1_ID	0x48	0x48
I2C2_ID_REG	I2C2_ID	0x12	0x12

5.12 Watchdog Settings

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

Table 5-12. Watchdog NVM Settings

Register Name	Field Name	TPS65941319-Q1	
		Value	Description
WD_LONGWIN_CFG	WD_LONGWIN	0xff	0xff
WD_THR_CFG	WD_EN	0x1	Watchdog enabled.

6 Pre-Configurable Finite State Machine (PFSM) Settings

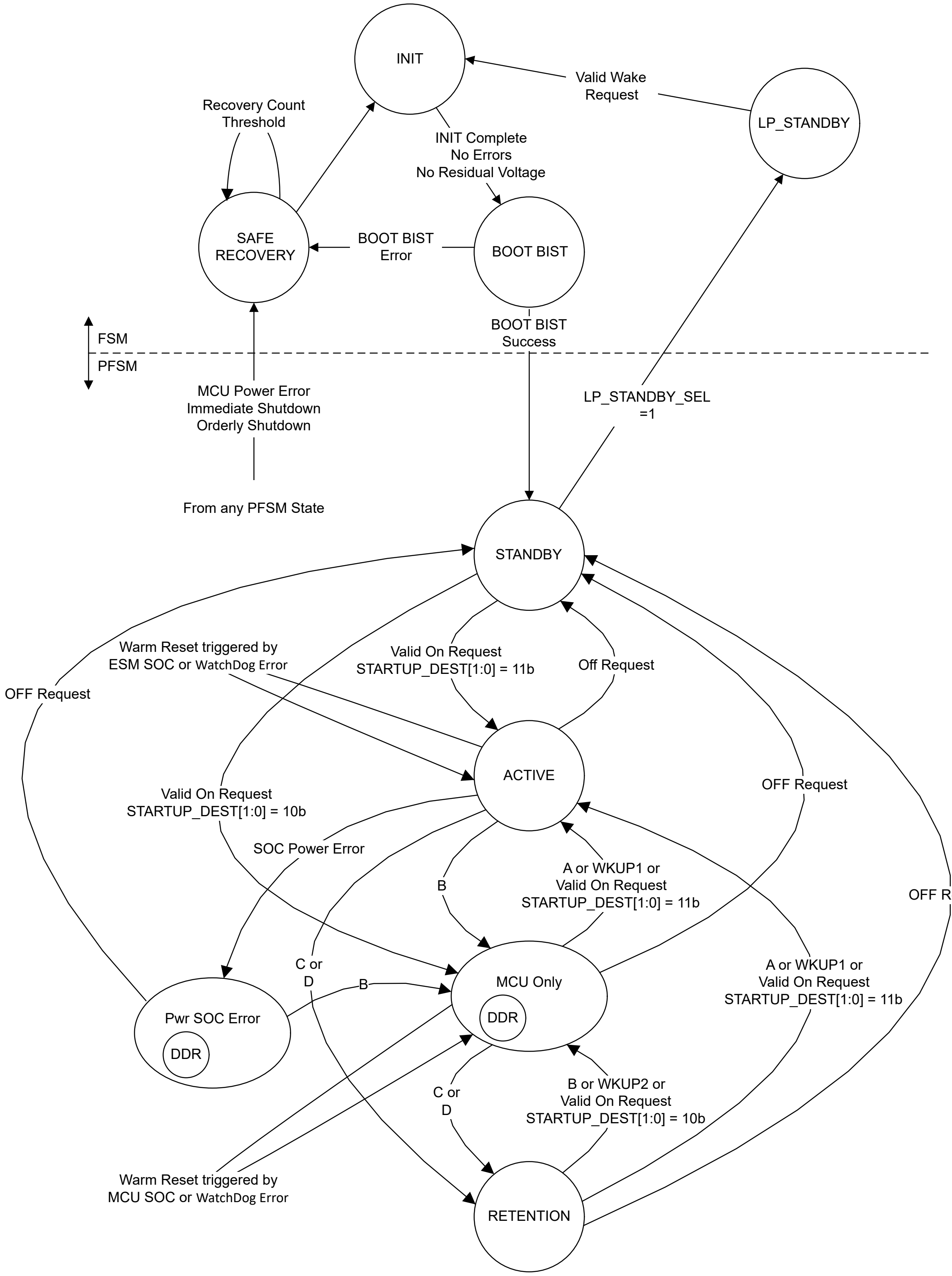
This section describes the default PFSM settings of the TPS6594-Q1 device. These settings cannot be changed after device startup.

6.1 Configured States

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

- Standby
- Active
- MCU Only
- Pwr SoC Error
- 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 Power Management IC (PMIC) with 5 Bucks and 4 LDOs for Safety-Relevant Automotive Applications 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. VDD_DDR, which is enabled via GPIO4 remains on when trigger I2C_7 is high and is turned off during the transition to the MCU_ONLY state when I2C_7 is low. 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. Only GPIO4 remains high, which enables VDD_DDR, 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	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	2	True	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	SAFE[1]	TO_SAFE_ORDERLY
OFF Request	4[9]	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	

Table 6-1. State Transition Triggers (continued)

Trigger	Priority (ID)	Immediate (IMM)	REENTERANT	PFSM Current State	PFSM Destination State	Power Sequence or Function Executed
ESM SOC Error	7	False	True	ACTIVE	Pwr SOC Error	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	
SOC Power Error	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]
ON Request	13	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	TO_ACTIVE
WKUP1 goes high	14	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	
NSLEEP1 and NSLEEP2 are high[5]	15	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	ACTIVE	
MCU ON Request	16	False	False	STANDBY, ACTIVE[7], MCU ONLY, Suspend-to-RAM	MCU ONLY	TO_MCU
WKUP2 goes high	17	False	False	STANDBY, ACTIVE, MCU ONLY, Suspend-to-RAM	MCU ONLY	
NSLEEP1 goes low and NSLEEP2 goes high[5]	18	False	False	ACTIVE, MCU ONLY, Suspend-to-RAM	MCU ONLY	
NSLEEP1 goes low and NSLEEP2 goes low[5]	19	False	False	ACTIVE, MCU ONLY	Suspend-to-RAM	TO_RETENTION
NSLEEP1 goes high and NSLEEP2 goes low[5]	20	False	False	ACTIVE, MCU ONLY	Suspend-to-RAM	
I2C_0 bit goes high[3]	21[8]	False	False	STANDBY, ACTIVE, MCU ONLY	LP_STANDBY[2]	TO_STANDBY
I2C_3 bit goes high[3]	22[8]	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 data sheet for more details.

[2] If the LP_STANDBY_SEL bit is set (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 wakeup the device as determined by the means of entering LP_STANDBY. Refer to the data sheet for more details.

[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] 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.

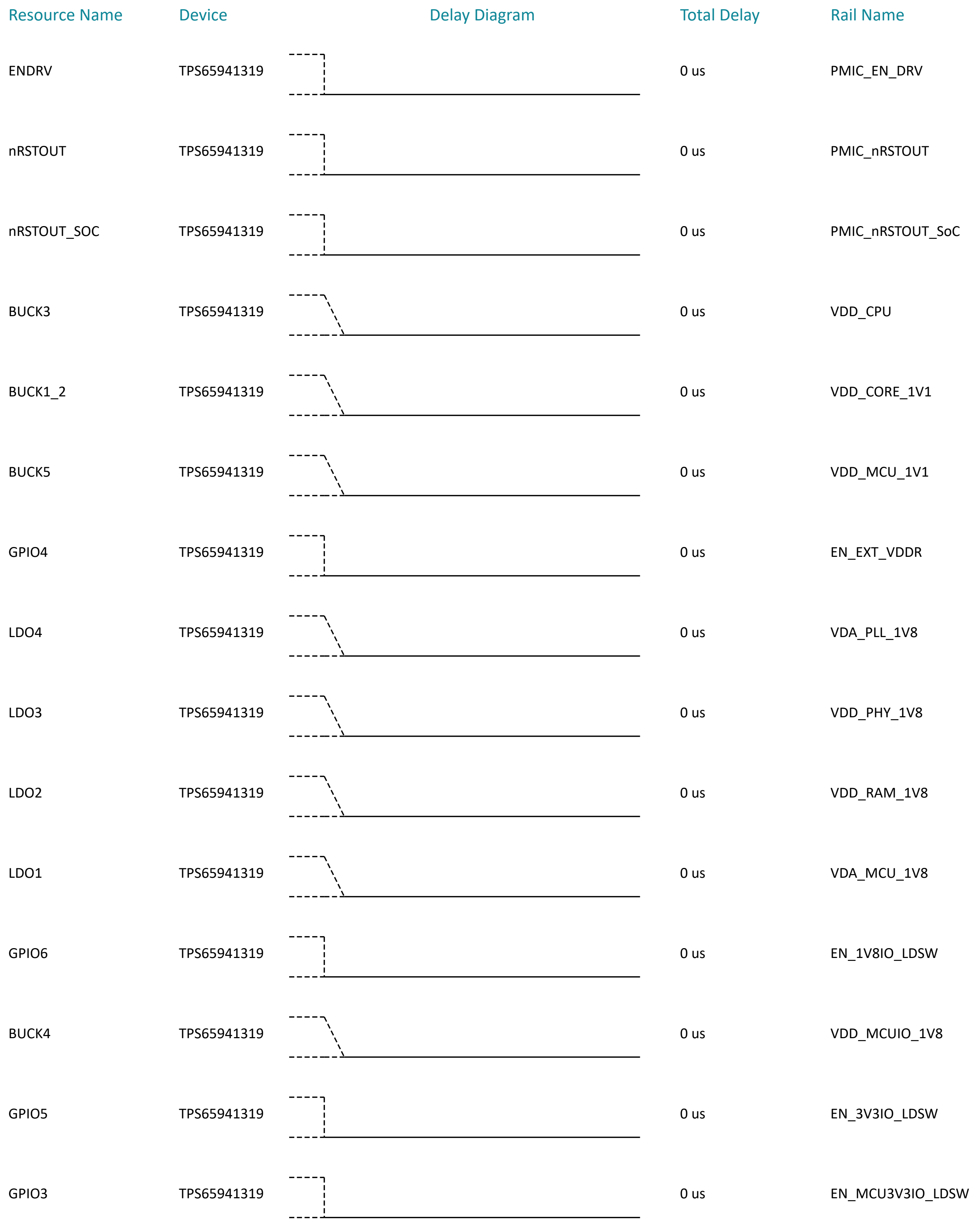
[8] Trigger IDs 21 and 22 are not available until the NSLEEP bits are masked: NSLEEP2_MASK=NSLEEP1_MASK=1.

[9] Trigger IDs 3, 23, and 24 are enabled and activated by the power sequences and are not shown. 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 are distinct sequences which occur when transition to the SAFE state. Both sequences shut down all rails without delay. The TO_SAFE_SEVERE sequence immediately ceases BUCK switching and enables the pulldown resistors of the BUCKs and LDOs. The cessation of BUCK switching is to prevent any damage of the PMICs in case of over voltage on VCCA or thermal shutdown. The timing is illustrated in Figure 6-2. The TO_SAFE sequence does not reset the BUCK regulators until after the regulators are turned off.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
ENDRV	TPS65941319		0 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		0 us	PMIC_nRSTOUT
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
BUCK3	TPS65941319		0 us	VDD_CPU
BUCK1_2	TPS65941319		0 us	VDD_CORE_1V1
BUCK5	TPS65941319		0 us	VDD_MCU_1V1
GPIO4	TPS65941319		0 us	EN_EXT_VDDR
LDO4	TPS65941319		0 us	VDA_PLL_1V8
LDO3	TPS65941319		0 us	VDD_PHY_1V8
LDO2	TPS65941319		0 us	VDD_RAM_1V8
LDO1	TPS65941319		0 us	VDA_MCU_1V8
GPIO6	TPS65941319		0 us	EN_1V8IO_LDSW
BUCK4	TPS65941319		0 us	VDD_MCUIO_1V8
GPIO5	TPS65941319		0 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		0 us	EN_MCU3V3IO_LDSW

Figure 6-2. TO_SAFE_SEVERE and TO_SAFE Power Sequences

After the power sequence shown in Figure 6-2, the TO_SAFE sequence delays the sequence by 16 ms. After this delays, the following instructions are executed:

```
// Clear AMUXOUT_EN, CLKMON_EN, set LPM_EN
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
// Reset all BUCK regulators
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
// Make GPIO9 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x39 DATA=0x18 MASK=0x00
// Make GPIO10 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x3A DATA=0x08 MASK=0x00
```

The TO_SAFE_SEVERE sequence executes the following instruction after the power sequence:

```
// Clear AMUXOUT_EN, CLKMON_EN, set LPM_EN
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
// Make GPIO9 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x39 DATA=0x18 MASK=0x00
// Make GPIO10 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x3A DATA=0x08 MASK=0x00
```

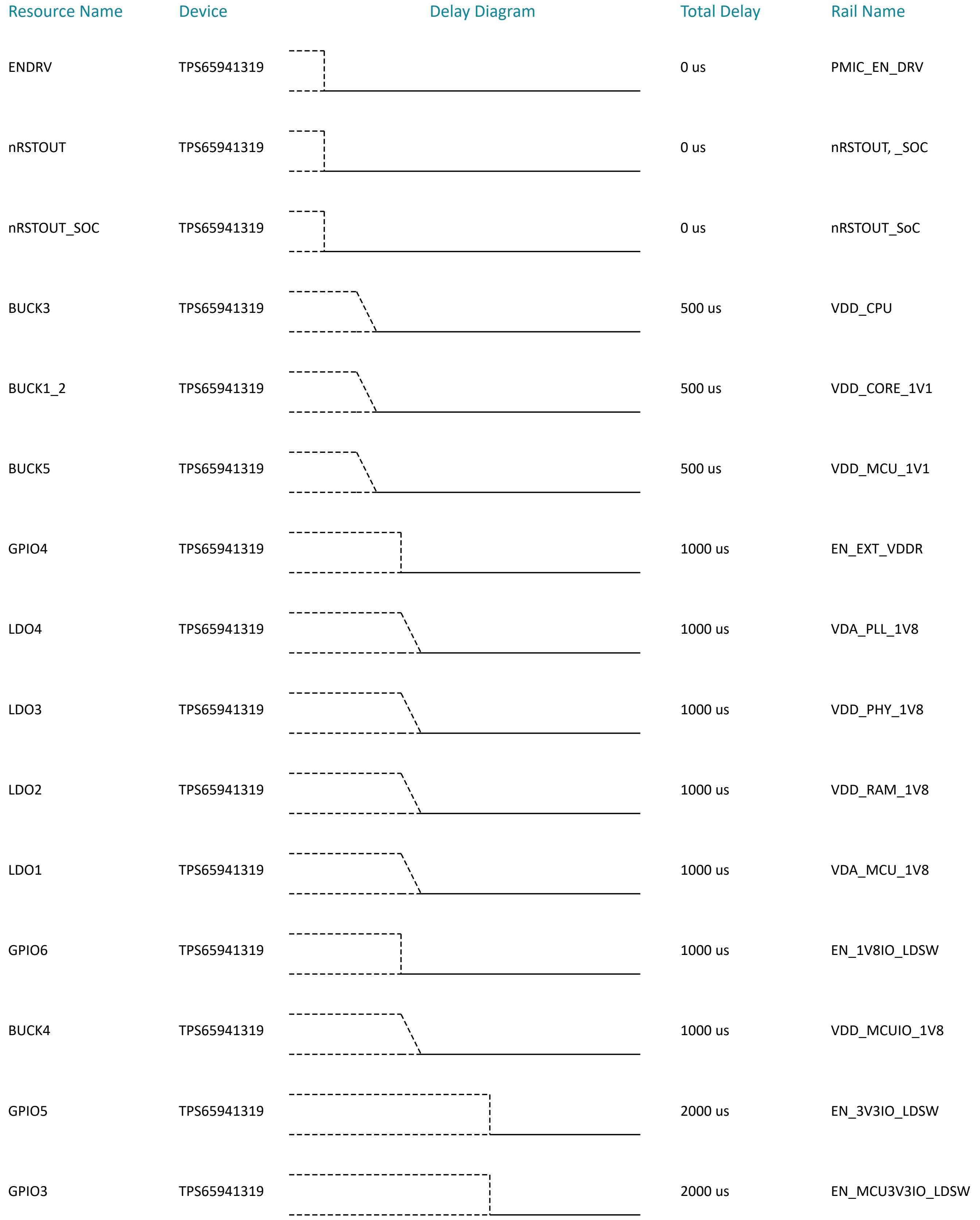
A delay of 500 ms is at the end of the TO_SAFE_SEVERE sequence. The recovery is not attempted until after the sequence delay is complete.

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 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. The power sequence for both of these events is shown in Figure 6-3.

Both the TO_SAFE_ORDERLY and TO_STANDBY sequences set the FORCE_EN_DRV_LOW bit.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
ENDRV	TPS65941319		0 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		0 us	nRSTOUT, _SOC
nRSTOUT_SOC	TPS65941319		0 us	nRSTOUT_SoC
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
BUCK5	TPS65941319		500 us	VDD_MCU_1V1
GPIO4	TPS65941319		1000 us	EN_EXT_VDDR
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
LDO1	TPS65941319		1000 us	VDA_MCU_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
BUCK4	TPS65941319		1000 us	VDD_MCUIO_1V8
GPIO5	TPS65941319		2000 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		2000 us	EN_MCU3V3IO_LDSW

Figure 6-3. TO_SAFE_ORDERLY and TO_STANDBY Power Sequence

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

```
// Clear AMUXOUT_EN and CLKMON_EN and set LPM_EN
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
// Reset all BUCKs (not performed in the TO_STANDBY sequence)
REG_WRITE_MASK_IMM ADDR=0x87 DATA=0x1F MASK=0xE0
// Make GPIO9 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x39 DATA=0x18 MASK=0x00
// Make GPIO10 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x3A DATA=0x08 MASK=0x00
```

The resetting of the BUCK regulators is done in preparation to transitioning to the SAFE_RECOVERY state, meaning that the PMIC leaves the mission state. The SAFE_RECOVERY state is where the recovery mechanism increments the recovery counter and determines if the recovery count threshold (see Table 5-10) is reached before attempting to recover.

At the end of the TO_STANDBY sequence, the same 16 ms delay and instructions are with the exception of the BUCK_RESET. After these instructions, the PMIC performs an additional check to determine if the LP_STANDBY_SEL (see Table 5-10) is true. If true then the PMICs enter the LP_STANDBY state and leave the mission state. If the LP_STANDBY_SEL is false, then the PMICs remain in the mission state defined by STANDBY in Section 6.1.

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.

Note

GPIOs do not reset during the sequence as shown in Figure 6-4

At the beginning of the sequence the following instructions are executed:

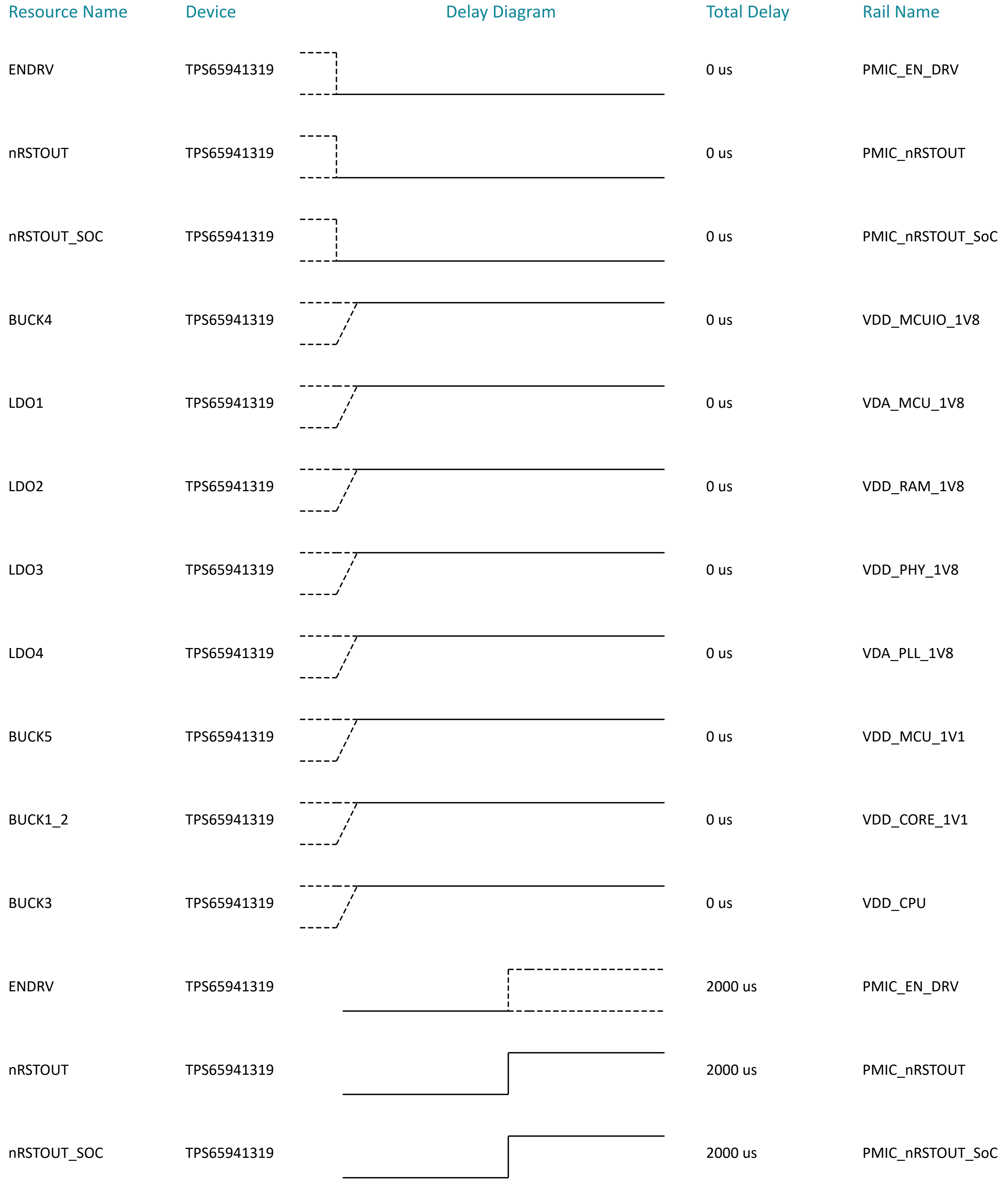
```
// Set FORCE_EN_DRV_LOW
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x08 MASK=0xF7
// Clear nRSTOUT and nRSTOUT_SOC
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFC
// Increment the recovery counter
REG_WRITE_MASK_IMM ADDR=0xa5 DATA=0x01 MASK=0xFE
```

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.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
ENDRV	TPS65941319		0 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		0 us	PMIC_nRSTOUT
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
BUCK4	TPS65941319		0 us	VDD_MCUIO_1V8
LDO1	TPS65941319		0 us	VDA_MCU_1V8
LDO2	TPS65941319		0 us	VDD_RAM_1V8
LDO3	TPS65941319		0 us	VDD_PHY_1V8
LDO4	TPS65941319		0 us	VDA_PLL_1V8
BUCK5	TPS65941319		0 us	VDD_MCU_1V1
BUCK1_2	TPS65941319		0 us	VDD_CORE_1V1
BUCK3	TPS65941319		0 us	VDD_CPU
ENDRV	TPS65941319		2000 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		2000 us	PMIC_nRSTOUT
nRSTOUT_SOC	TPS65941319		2000 us	PMIC_nRSTOUT_SoC

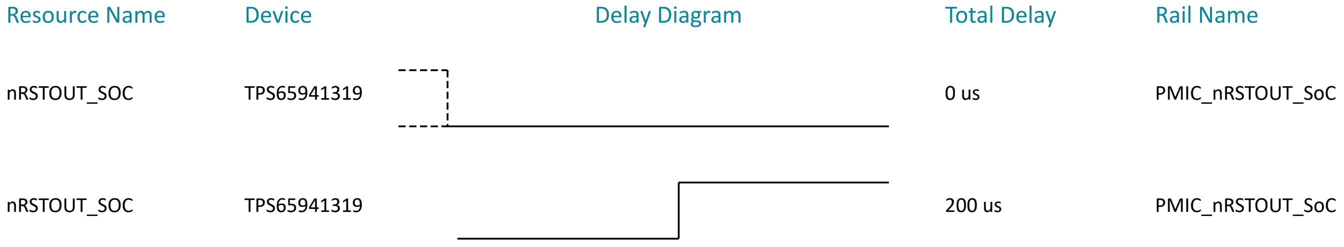
Figure 6-4. ACTIVE_TO_WARM Power Sequence

Note

The regulator transitions do not represent enabling of the regulators but the time at which the voltages are restored to their default values. Since this sequence originates from the ACTIVE state all of the regulators are on.

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. The sequence is shown in Figure 6-5.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
nRSTOUT_SOC	TPS65941319		200 us	PMIC_nRSTOUT_SoC

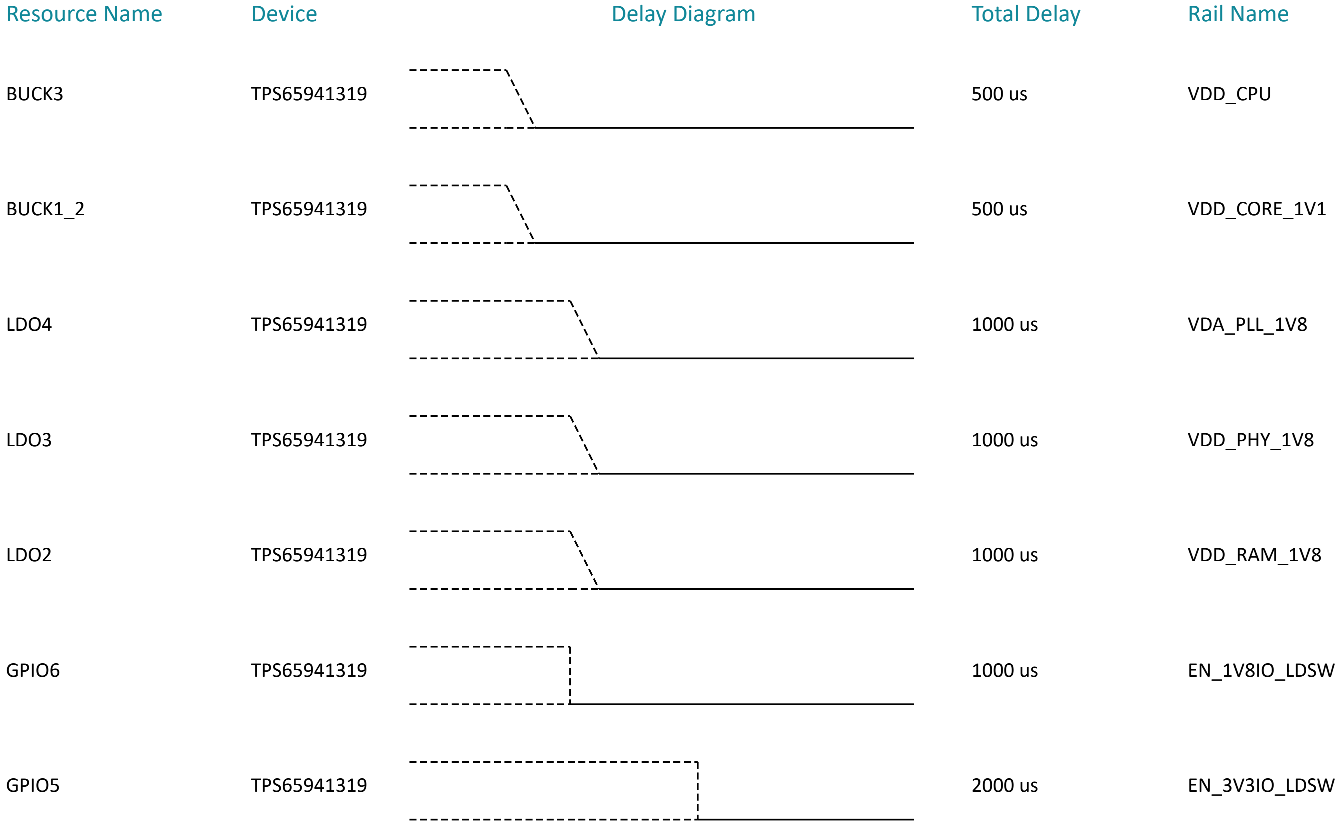
Figure 6-5. ESM_SOC_ERROR Sequence

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 as shown in Figure 6-6. The state of the I2C_7 trigger determines whether the control signal, GPIO4, remain energized (I2C_7=1) or disabled (I2C_7=0), as shown in Figure 6-7.

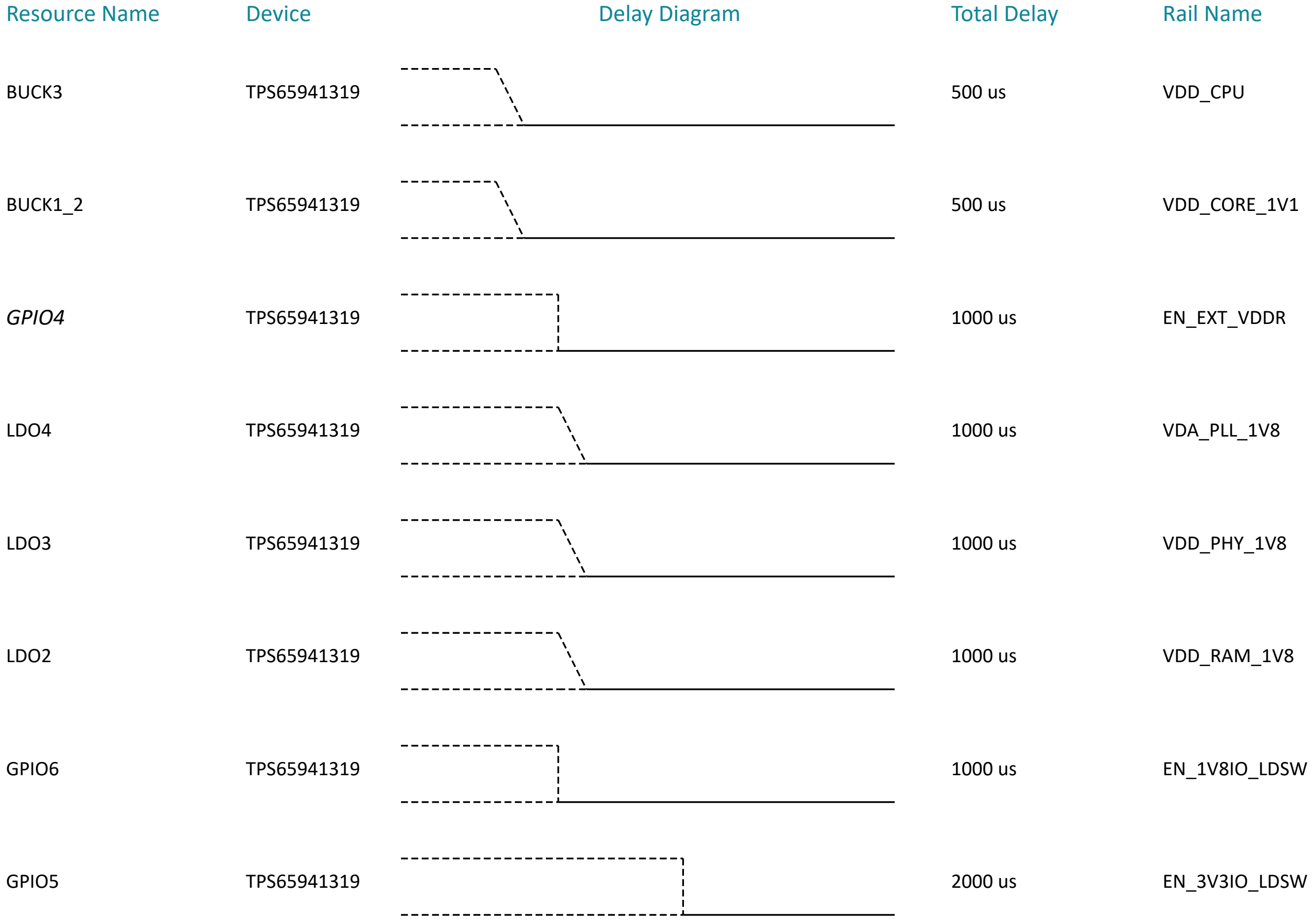
In the start of the sequence the following instructions are executed:

```
// Set AMUXOUT_EN and CLKMON_EN, clear LPM_EN and nRSTOUT_SOC
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE1
// Clear SPMI_LPM_EN
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
```



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
GPIO5	TPS65941319		2000 us	EN_3V3IO_LDSW

Figure 6-6. PWR_SOC_ERROR with I2C_7 High in both PMICs



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
GPIO4	TPS65941319		1000 us	EN_EXT_VDDR
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
GPIO5	TPS65941319		2000 us	EN_3V3IO_LDSW

Figure 6-7. PWR_SOC_ERROR with I2C_7 low in both PMICs

Note

NRSTOUT_SOC is not shown in the diagram but goes low at time 0us.

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 at the time indicated in Figure 6-8, and finally the MCU_PORz signal is set high after 2ms.

Note

GPIOs do not reset during the MCU warm reset event.

Also, at the beginning of the sequence the following instructions are executed to increment the recovery counter and configure the PMICs:

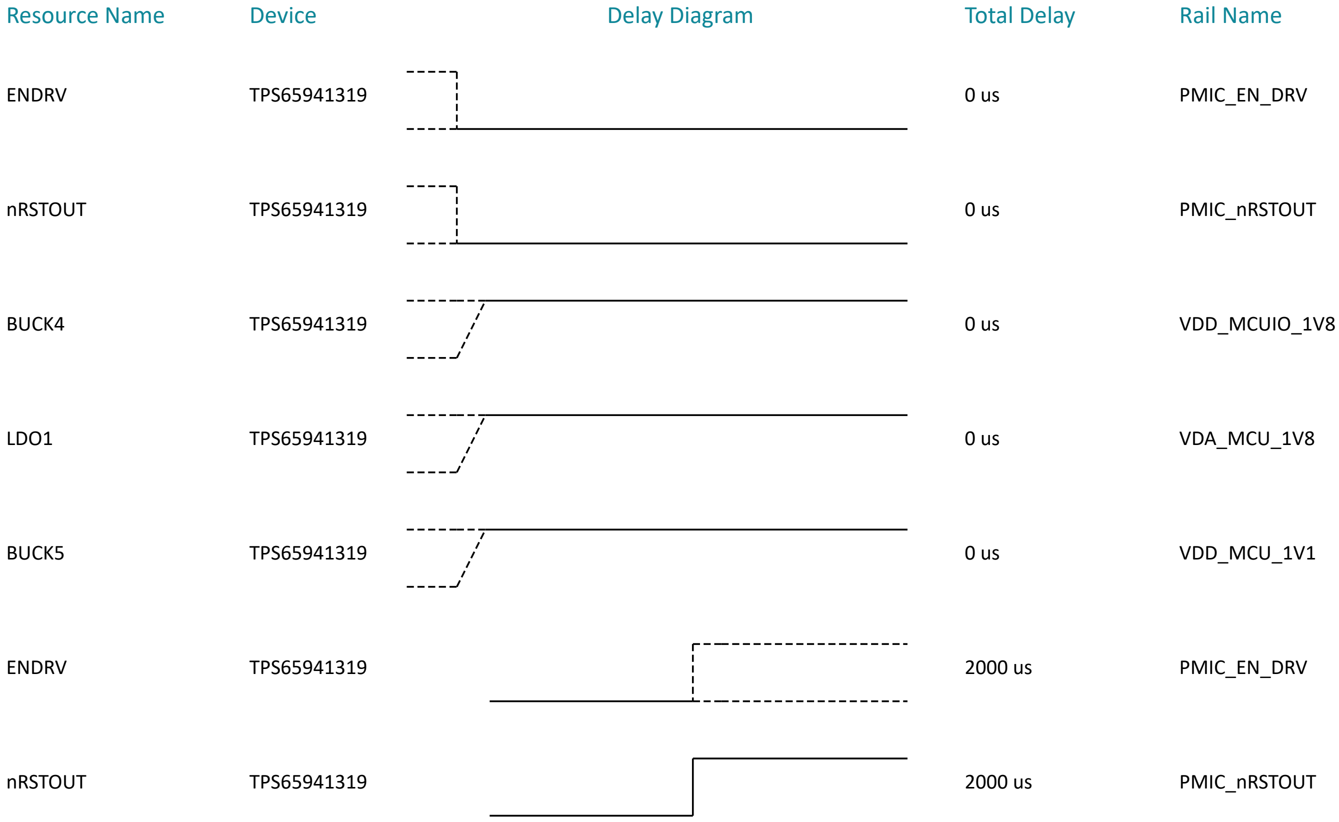
```
// Set FORCE_EN_DRV_LOW
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x08 MASK=0xF7
// Clear nRSTOUT
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x00 MASK=0xFE
// Increment Recovery Counter
REG_WRITE_MASK_IMM ADDR=0xa5 DATA=0x01 MASK=0xFE
```

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.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
ENDRV	TPS65941319		0 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		0 us	PMIC_nRSTOUT
BUCK4	TPS65941319		0 us	VDD_MCUIO_1V8
LDO1	TPS65941319		0 us	VDA_MCU_1V8
BUCK5	TPS65941319		0 us	VDD_MCU_1V1
ENDRV	TPS65941319		2000 us	PMIC_EN_DRV
nRSTOUT	TPS65941319		2000 us	PMIC_nRSTOUT

Figure 6-8. MCU_TO_WARM Sequence

Note

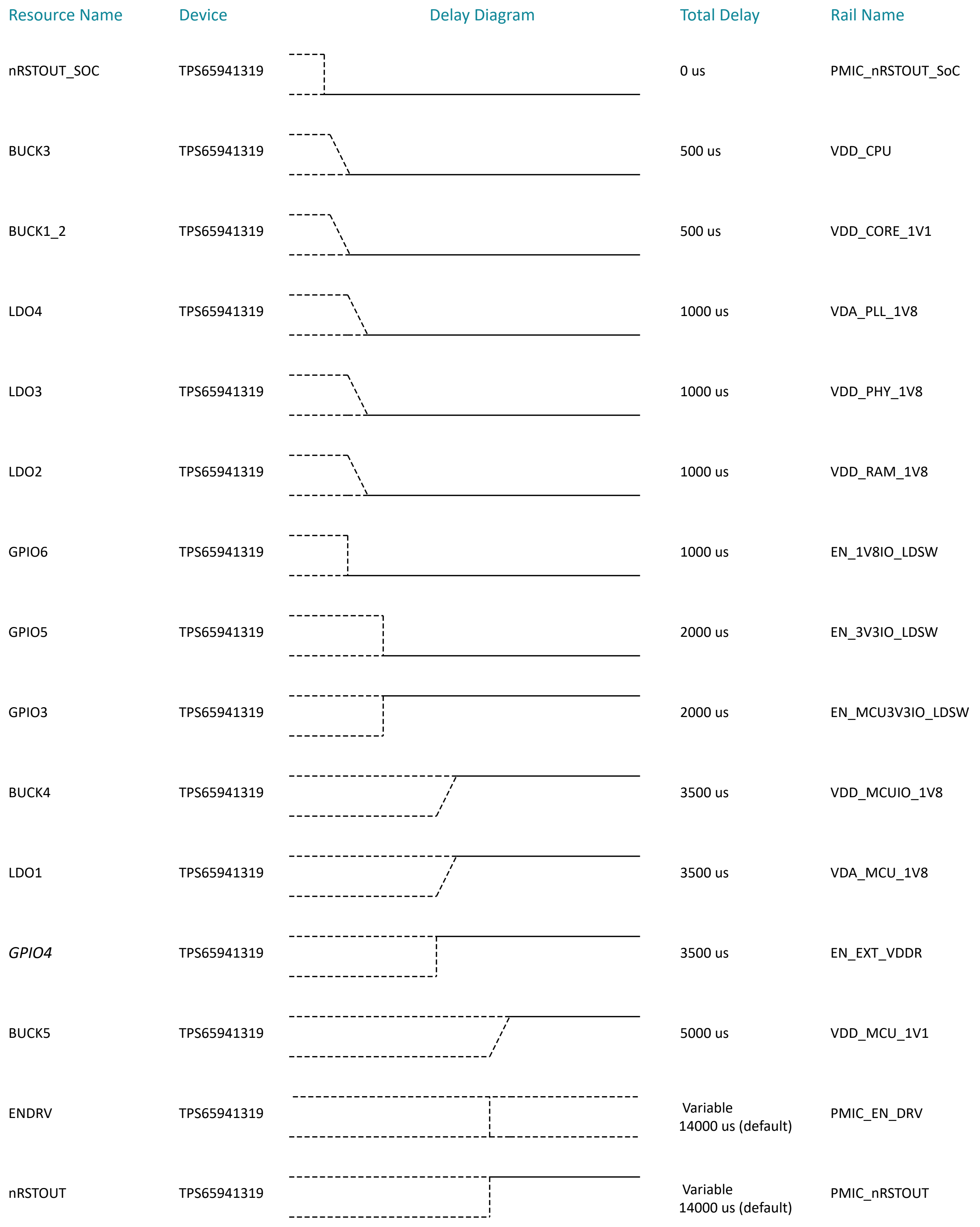
The regulator transitions do not represent enabling of the regulators but the time at which the voltages are restored to their default values. Since this sequence originates from the MCU_ONLY state these regulators are on.

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. The I2C_7 setting must be the same in each PMIC before triggering the sequence. If the bit is low, then EN_EXT_VDDR is disabled; Figure 6-10. If the I2C_7 bit is high, then EN_EXT_VDDR is enabled; Figure 6-9.

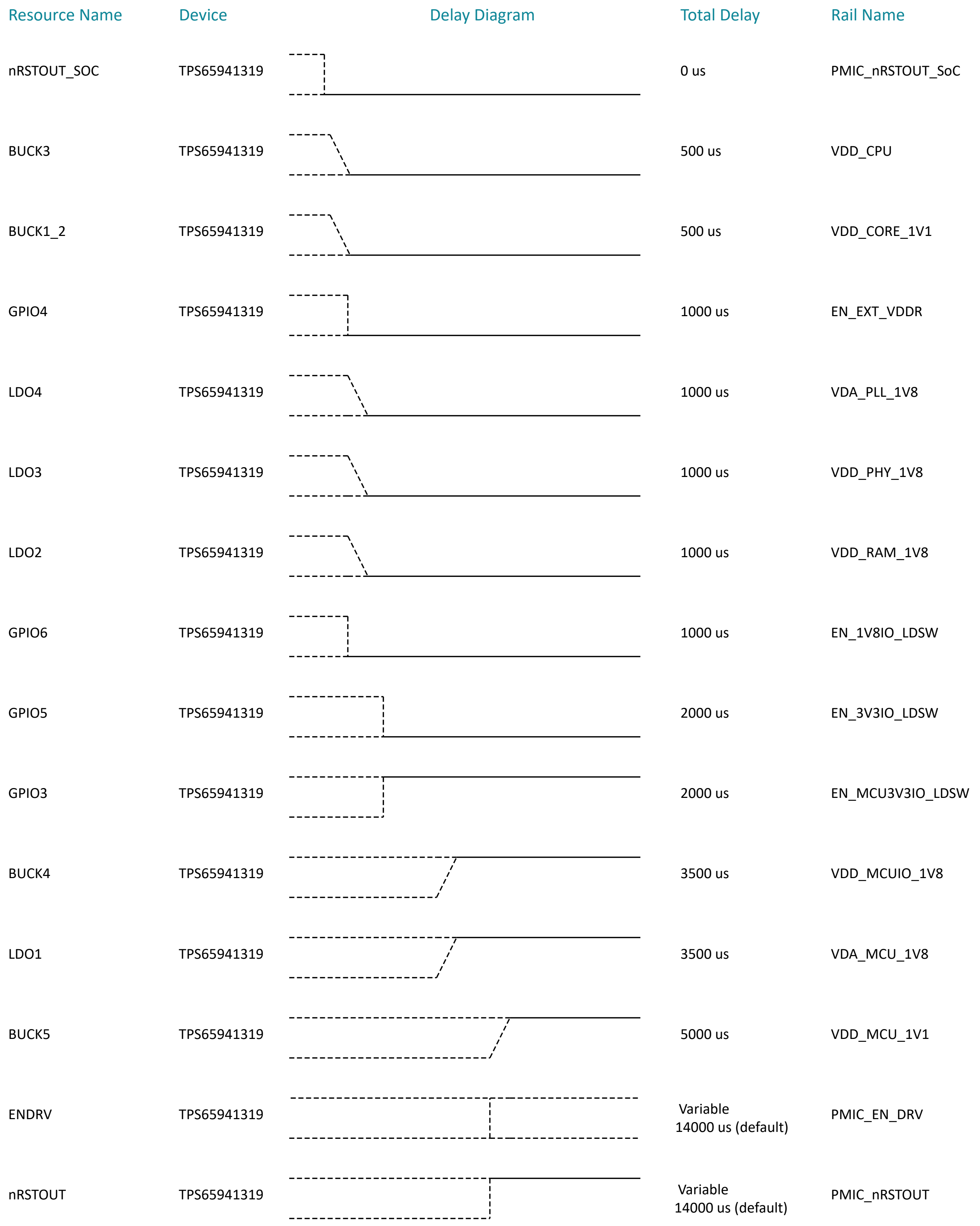
The first instructions of the TO_MCU sequence perform writes to the MISC_CTRL and ENABLE_DRV_STAT registers.

```
// Set AMUXOUT_EN, CLKMON_EN
// Clear LPM_EN, NRSTOUT_SOC
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x18 MASK=0xE1
// Clear SPMI_LP_EN
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xEF
```



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
GPIO5	TPS65941319		2000 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		2000 us	EN_MCU3V3IO_LDSW
BUCK4	TPS65941319		3500 us	VDD_MCUIO_1V8
LDO1	TPS65941319		3500 us	VDA_MCU_1V8
GPIO4	TPS65941319		3500 us	EN_EXT_VDDR
BUCK5	TPS65941319		5000 us	VDD_MCU_1V1
ENDRV	TPS65941319		Variable 14000 us (default)	PMIC_EN_DRV
nRSTOUT	TPS65941319		Variable 14000 us (default)	PMIC_nRSTOUT

Figure 6-9. TO_MCU with I2C_7 high in both PMICs



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
GPIO4	TPS65941319		1000 us	EN_EXT_VDDR
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
GPIO5	TPS65941319		2000 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		2000 us	EN_MCU3V3IO_LDSW
BUCK4	TPS65941319		3500 us	VDD_MCUIO_1V8
LDO1	TPS65941319		3500 us	VDA_MCU_1V8
BUCK5	TPS65941319		5000 us	VDD_MCU_1V1
ENDRV	TPS65941319		Variable 14000 us (default)	PMIC_EN_DRV
nRSTOUT	TPS65941319		Variable 14000 us (default)	PMIC_nRSTOUT

Figure 6-10. TO_MCU Sequence with I2C_7 low in both PMICs

The last instructions of the TO_MCU sequence also perform writes to the MISC_CTRL and ENABLE_DRV_STAT registers after the delay defined in the PFSM_DELAY_REG_1.

```
SREG_READ_REG ADDR=0xCD REG=R1
DELAY_SREG R1
// Clear SPMI_LPM_EN and FORCE_EN_DRV_LOW
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x00 MASK=0xE7
// Set NRSTOUT (MCU_PORZ)
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x01 MASK=0xFE
```

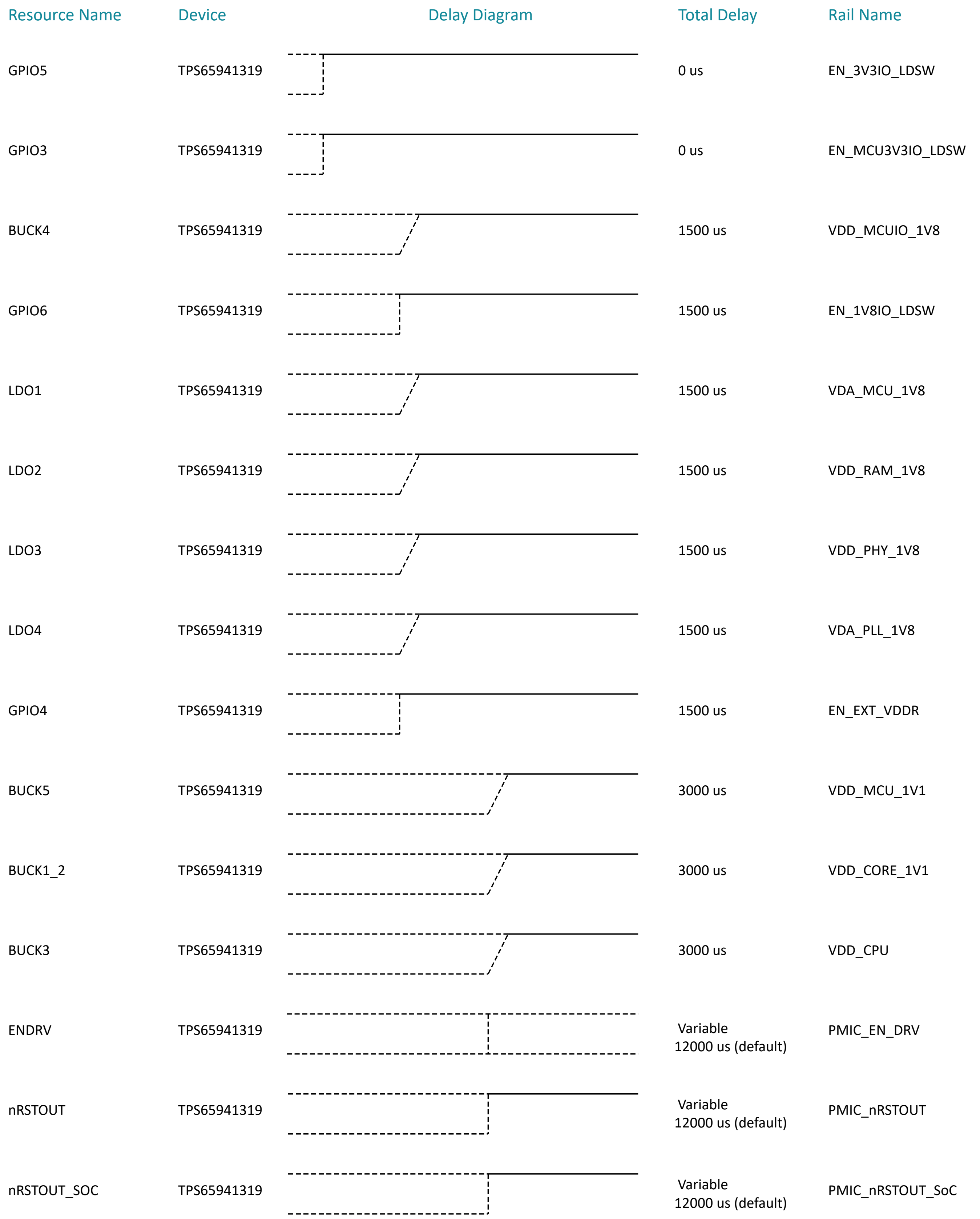
Note

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 as shown in Figure 6-11.

At the beginning of the TO_ACTIVE sequence the PMIC clears SPMI_LP_EN and LPM_EN and set AMUXOUT_EN and CLKMON_EN.



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
GPIO5	TPS65941319		0 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		0 us	EN_MCU3V3IO_LDSW
BUCK4	TPS65941319		1500 us	VDD_MCUIO_1V8
GPIO6	TPS65941319		1500 us	EN_1V8IO_LDSW
LDO1	TPS65941319		1500 us	VDA_MCU_1V8
LDO2	TPS65941319		1500 us	VDD_RAM_1V8
LDO3	TPS65941319		1500 us	VDD_PHY_1V8
LDO4	TPS65941319		1500 us	VDA_PLL_1V8
GPIO4	TPS65941319		1500 us	EN_EXT_VDDR
BUCK5	TPS65941319		3000 us	VDD_MCU_1V1
BUCK1_2	TPS65941319		3000 us	VDD_CORE_1V1
BUCK3	TPS65941319		3000 us	VDD_CPU
ENDRV	TPS65941319		Variable 12000 us (default)	PMIC_EN_DRV
nRSTOUT	TPS65941319		Variable 12000 us (default)	PMIC_nRSTOUT
nRSTOUT_SOC	TPS65941319		Variable 12000 us (default)	PMIC_nRSTOUT_SoC

Figure 6-11. TO_ACTIVE Sequence

At the end of the TO_ACTIVE sequence the 'FORCE_EN_DRV_LOW' bit is cleared. Additionally, the nRSTOUT and nRSTOUT_SOC signals are delayed by the value found in PFSM_DELAY_REG_1.

Note

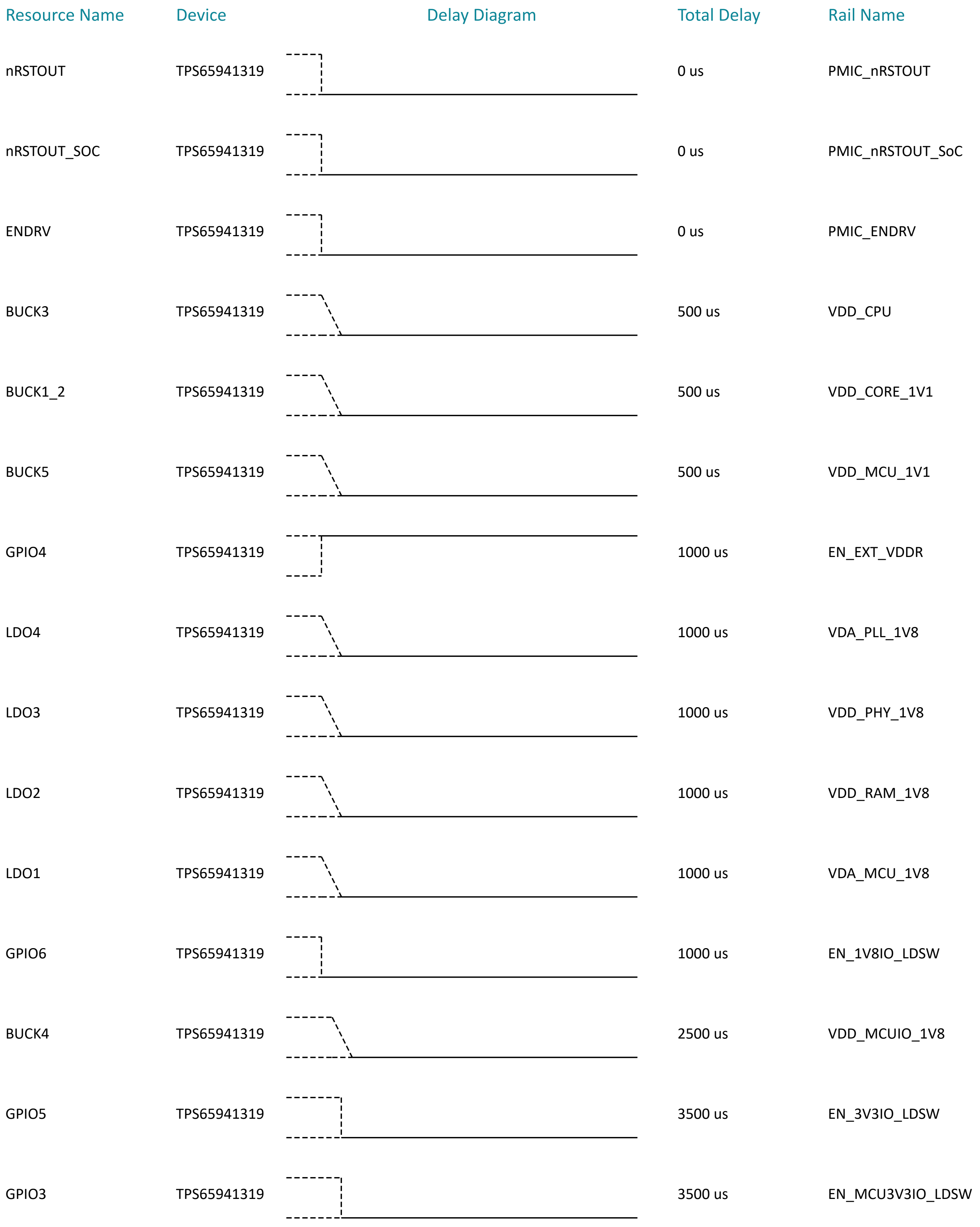
After the TO_ACTIVE sequence the MCU is responsible for managing the EN_DRV.

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 following PMIC PFSM instructions are executed automatically in the beginning of the power sequence to configure the PMIC:

```
// Set LPM_EN, Clear NRSTOUT_SOC and NRSTOUT
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xF8
// Set SPMI_LP_EN and FORCE_EN_DRV_LOW
REG_WRITE_MASK_IMM ADDR=0x82 DATA=0x18 MASK=0xE7
```



Resource Name	Device	Delay Diagram	Total Delay	Rail Name
nRSTOUT	TPS65941319		0 us	PMIC_nRSTOUT
nRSTOUT_SOC	TPS65941319		0 us	PMIC_nRSTOUT_SoC
ENDRV	TPS65941319		0 us	PMIC_ENDRV
BUCK3	TPS65941319		500 us	VDD_CPU
BUCK1_2	TPS65941319		500 us	VDD_CORE_1V1
BUCK5	TPS65941319		500 us	VDD_MCU_1V1
GPIO4	TPS65941319		1000 us	EN_EXT_VDDR
LDO4	TPS65941319		1000 us	VDA_PLL_1V8
LDO3	TPS65941319		1000 us	VDD_PHY_1V8
LDO2	TPS65941319		1000 us	VDD_RAM_1V8
LDO1	TPS65941319		1000 us	VDA_MCU_1V8
GPIO6	TPS65941319		1000 us	EN_1V8IO_LDSW
BUCK4	TPS65941319		2500 us	VDD_MCUIO_1V8
GPIO5	TPS65941319		3500 us	EN_3V3IO_LDSW
GPIO3	TPS65941319		3500 us	EN_MCU3V3IO_LDSW

Figure 6-12. TO_RETENTION

16ms after the GPIO3 is deasserted, Figure 6-12,the PMIC executes the following instructions:

```
// Set LPM_EN, Clear CLKMON_EN and AMUXOUT_EN
REG_WRITE_MASK_IMM ADDR=0x81 DATA=0x04 MASK=0xE3
// Make GPIO9 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x39 DATA=0x18 MASK=0x00
// Make GPIO10 an input with pulldown enabled
REG_WRITE_MASK_IMM ADDR=0x3A DATA=0x08 MASK=0x00
```

An additional delay is applied at the end of the sequence based upon the contents of the register (PFSM_DELAY_REG_2).

7 Application Examples

This section provides examples of how to interact with the PMICs from the perspective of the MCU and over I2C. Table 7-1 shows how the I2C commands are presented in the following sections. These examples, when used in conjunction with the data sheet, can be generalized and applied to other use cases.

Table 7-1. I2C Instruction Format

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

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

The default configuration of the NVM transitions the PMIC to the ACTIVE state when the ENABLE pin goes high (rising edge triggered). The nINT pin goes high to indicate to the MCU that interrupts have occurred in the PMIC. 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,meaning 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 I2C 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	NA	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
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
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, the WKUP1 function on GPIO8 is provided as an alternative 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. In this example GPIO8 on the PMIC is used to wake the device from RETENTION to ACTIVE.

```
Write 0x48:0x34:0xC0;0x3F  // Set GPIO8 to WKUP1 (goes to ACTIVE state)
Write 0x48:0x64:0x80:0x7F  // clear interrupt of gpio8, write to clear
Write 0x48:0x4F:0x00:0x7F  // unmask interrupt for GPIO8 falling edge
Write 0x48:0x86:0x00:0xFC  // trigger the TO_RETENTION power sequence
After the GPIO8 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 gpio8
```

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

```
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 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
```

7.2 Entering and Exiting Standby

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

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 GPIO8 or and RTC timer or alarm. In this example, I2C_0 trigger is used to enter the STANDBY state and the GPIO8 is used to enter the ACTIVE state.

Note

From the LP_STANDBY state GPIO8 is not available as a wakeup source.

```
Write 0x48:0xC3:0x00:0xF7  // LP_STANDBY_SEL=0
Write 0x48:0x7D:0xC0:0x3F  // Mask NSLEEP bits
Write 0x48:0x38:0xC0;0x3F  // Set GPIO4 to WKUP1 (goes to ACTIVE state)
Write 0x48:0x64:0x80:0x7F  // clear interrupt of GPIO8
Write 0x48:0x4F:0x00:0x7F  // unmask interrupt for GPIO8 falling edge
Write 0x48:0x85:0x01:0xFE  // set I2C_0 trigger, trigger TO_STANDBY sequence
After the GPIO8 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 0x48:0x64:0x80:0x7F  // clear interrupt of GPIO8
```

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

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 the expiration of the long window; 772 seconds, Table 5-12.

```
Write 0x12:0x09:0x00:0xBF  // Disable Watchdog
Write 0x48:0x38:0xC0:0x1F  // configure GPIO8 as WAKEUP1
```

When enabling the watchdog the WD_PWR_HOLD must also be cleared.

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

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. The 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, *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*

9 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (April 2023) to Revision A (June 2025) **Page**

- Updated abstract.. 1
- Updated entries in Figure 3-1 and Table 3-1.. 3

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