RAA271005

Safety Application Note

Document ID: LLWEB-20111609

Ver 4.0

Date: Oct.15, 2024			
Product	Approved by	Checked by	Prepared by
RAA271005	Arun V.	Preeti Metha Ingoo Jung Seiichi Kuosokabe Kenichi S. Atsushi W.	Estella S.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This application note explains the recommended usage of Safety Mechanisms (SM) for RAA271005.

This application note is made based on the use cases described in SRS Section 1.9 General Assumed Use Cases. Please refer to the latest version of the RAA271005 datasheet when using this application note.

This application note provides support for fulfilling the SEooC assumptions in the preparation of the required work products at the system level. There is no guarantee given that the required ASIL level can be reached at system level. This is strictly dependent on the system configuration used. The aim of this document is to provide supplemental information for the assumed HW and system solutions to assist the compliance of ISO26262 in system integration.

Each section describes a safety mechanism and contains the following sub-sections:

- Overview

  Description of each safety mechanism
- Hardware Description

  Description of how the safety mechanism is implemented in hardware
- Recommended usage

  Any possible recommendations to bear in mind during the usage of the RAA271005 to prevent possible misbehaviors.
- Fault Control

  Definition of the assumed fault reaction and control
- Operation

  Description of the software test description including references to register
- Fault Detection and Safety Mechanism

  Description of the fault detection within the safety mechanism itself

Caution

In case that read access to memory/register is executed as part of a fault detection procedure, user shall ensure that the read operation is performed on the intended target resource. Data is available for READ in target register 62.5ns after the IRQ pin is asserted.

2 Abbreviations

Abbreviation	Meaning
ADC	Analog to Digital Converter
ASIL	Automotive Safety Integrity Level
CRC	Cyclic Redundancy Check
CVM	(SoC) Core Voltage Monitor
IIR	Infinite Impulse Response filter
MUX	Multiplexer
OVP	Over Voltage Protection
OT	Over temperature (protection)
OTP	One-Time Programmable memory
OV	Over Voltage
PGA	Programmable Gain Amplifier
PMIC	Power Management Integrated Circuit.
SAN	Safety Application Note
SEooC	Safety Element out of Context
SM	Safety Mechanism
UVP	Under Voltage Protection
UV	Under Voltage

3 Assumptions of Use

Refer to LLWEB-20150186 RAA271005 SRS Section 1.9 General Assumed Use Cases & RAA271005 Datasheet Figure 2. Typical Application Diagram with 5 Bucks.

4 General Notation and Terminology

This chapter describes the general notation and terminology used in this document

(A) Requirement Levels

These following bolded words appearing in this document are always interpreted to indicate requirement levels as follows:

Shall Indicates that the method is highly recommended to fulfill the SEooC assumptions. ("++" in this document indicates same level.)

Should indicates that the method is highly recommended to fulfill the SEooC assumptions but strongly depends on user application. ("+" in this document indicates same level.)

Can indicates that multiple measures (including HW measures) are available to be used to fulfill the SEooC assumptions and the proposed measure is one of them. ("o" in this document indicates same level.)

(B) Product Dependency

Target Device
RAA271005

(C) SAN ID, SM ID and Module

At the top of each section, the SAN ID, relevant SM IDs and module names are listed.

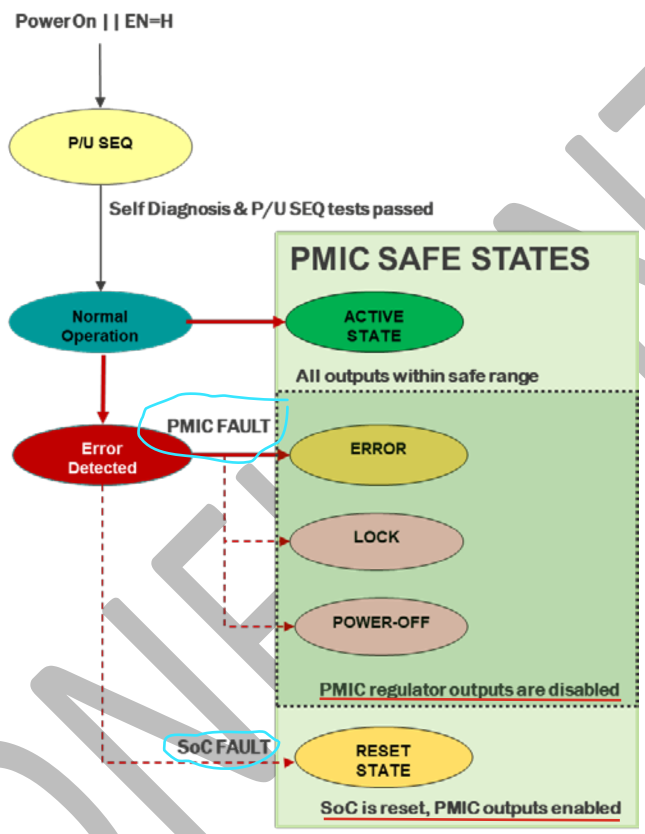
(D) Glossary

AWO – Always On
Cold Start – First power up of device in manufacturing / Car key on
Full Run – Normal system operation
Deep Stop – System low current state, where only the AWO supplies are active. Protection block is disabled.
Cyclic Run – Periodic state where System is woken up to run field BIST
Backup or Suspend to RAM – State where memory rails are active along with AWO rails
EXPOC – Signal from PMIC to SoC to indicate AWO supplies are good
PWRCTL_SOCISO – Signal from SoC to PMIC to indicate entry to & exit from Full Run mode, if applicable
PMIC POST – PMIC self diagnostics including all BIST
For more details, refer also to SoC User Manual.

5 RAA271005 Safe States

The RAA271005 operations shown below: ACTIVE STATE (with all outputs within safe range), ERROR, LOCK, and POWER OFF are considered safe states. In the RAA271005 safe states, all output powers are within safe range or all of the RAA271005 output rails are disabled. PRESET# de-asserted is also an RAA271005 Safe State.

Figure 1 PMIC Safe States



Should multiple faults (with differing reactions) occur at the same time, priority is:

(1) Error state

(2) Reset state

5.1 Safety State Machine Operation

The State Machine Operation is described in Figure 41.

5.2 Safe State conditions

1. ACTIVE state
    - With all outputs within safe range.

2. ERROR state:
    - Entered upon various faults as shown in Table 36
    - ERROR state counter gets incremented. If maximum allowed fault is reached, system moves to LOCK state. **0x014 - FUSA_STATUS_4[3:0]:The SafetyCtrl_ErrCnt** register is updated.
    - Faults which are an entry condition to ERROR state are masked to allow system to try restarting.
    - System starts checking if Temp2 < OT_WARN is satisfied once TIMEOUT_MIN_ERROR_ST expires in order to move to P/U SEQUENCE state.
    - TIMEOUT_MIN_ERROR_ST shall be greater than maximum shutdown delays of all regulators including discharge detect time out if applicable
      The ERROR state timer can be configured to any of the following:
      10ms, 20ms, 40ms, 60ms, 80ms, 130ms, 260ms, 1.04s, 2.0 s, & 4.0 s.
    - System stays in ERROR state until Temp2 < OT_WARN is satisfied.
    - FORCE_ERROR_ST register can be used to force the system to go and stay in ERROR state. **0x014 - FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** will not be incremented if the system is forced to ERROR state through register write.
    - PRESET# is de-asserted
    - Outputs are stopped according to set delay.
    - SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

3. RESET state:
    - Entered upon various faults as shown in Table 36
    - RESET state counter is incremented. The **0x014 - FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt** register is updated.
    - PRESET#=L is applied and PRESETOUT0#=L loop back is checked.
    - If PRESETOUT0#=L received within TIMEOUT_PRESETOUT, system goes to SoC ACTIVATION state after the timer controlled by **0x118 - FUSA_TIMER_3[2:0]: TIMEOUT_PRESETOUT_DLY_TIME** expires.
    - If RESET loopback fails, RAA271005 moves to ERROR state.
    - During this state, all monitoring are still applied per Table 36

- **0x00C - FUSA_CTRL_5[1:0] FORCE_ST_ERROR_RESET** register can be used to force the system to go and stay in RESET state. Once **0x00C - FUSA_CTRL_5[1:0] FORCE_ST_ERROR_RESET** register is cleared, system will resume normal operation. SOC Fault counter will not be incremented if the system is forced to RESET state through register write.

4. LOCK state:

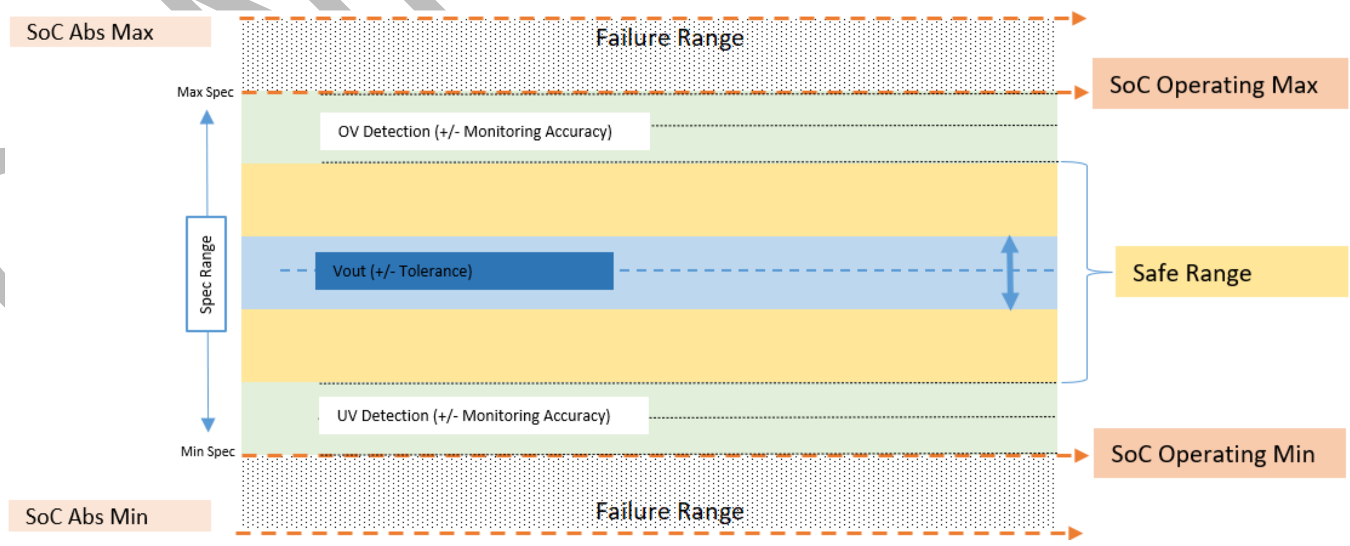
   - All power rails are discharged immediately.
   - SELF DIAGNOSIS test status registers are cleared and PMIC powers down
   - System can move from this state through power cycling (toggling EN)
   - At recovery from LOCK state, the PMIC is forced to go through SELF DIAGNOSIS state again.

5. POWER OFF state:

   - Will be entered from any state when EN=L or Avin<V_UVLO
   - Power rails are discharged immediately.
   - All registers reset to initial states.
   - Part completely powered off.

6. Output Safe Range

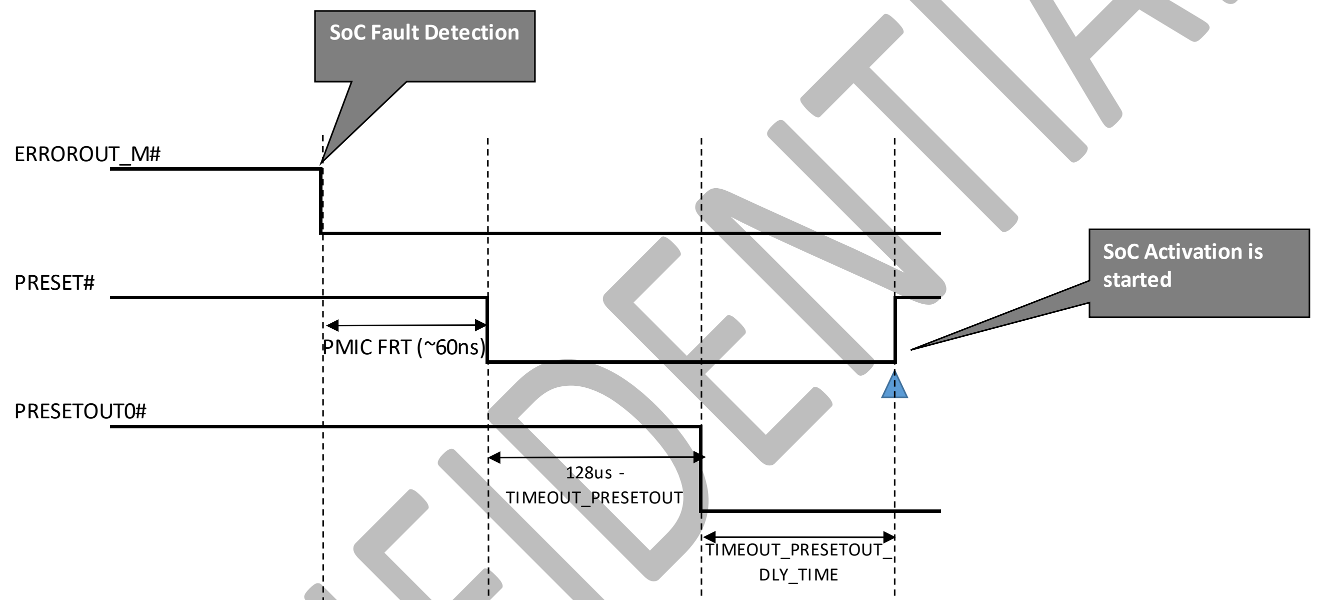
Figure 2 PMIC Output Safe Range



5.3 PMIC Recovery from Safe states

The PMIC transitions to RESET state after an error signal is received from the SoC.
While the SoC Error count has not reached the maximum threshold, the PMIC will re-try and recover from RESET state, as described in the timing diagram below.

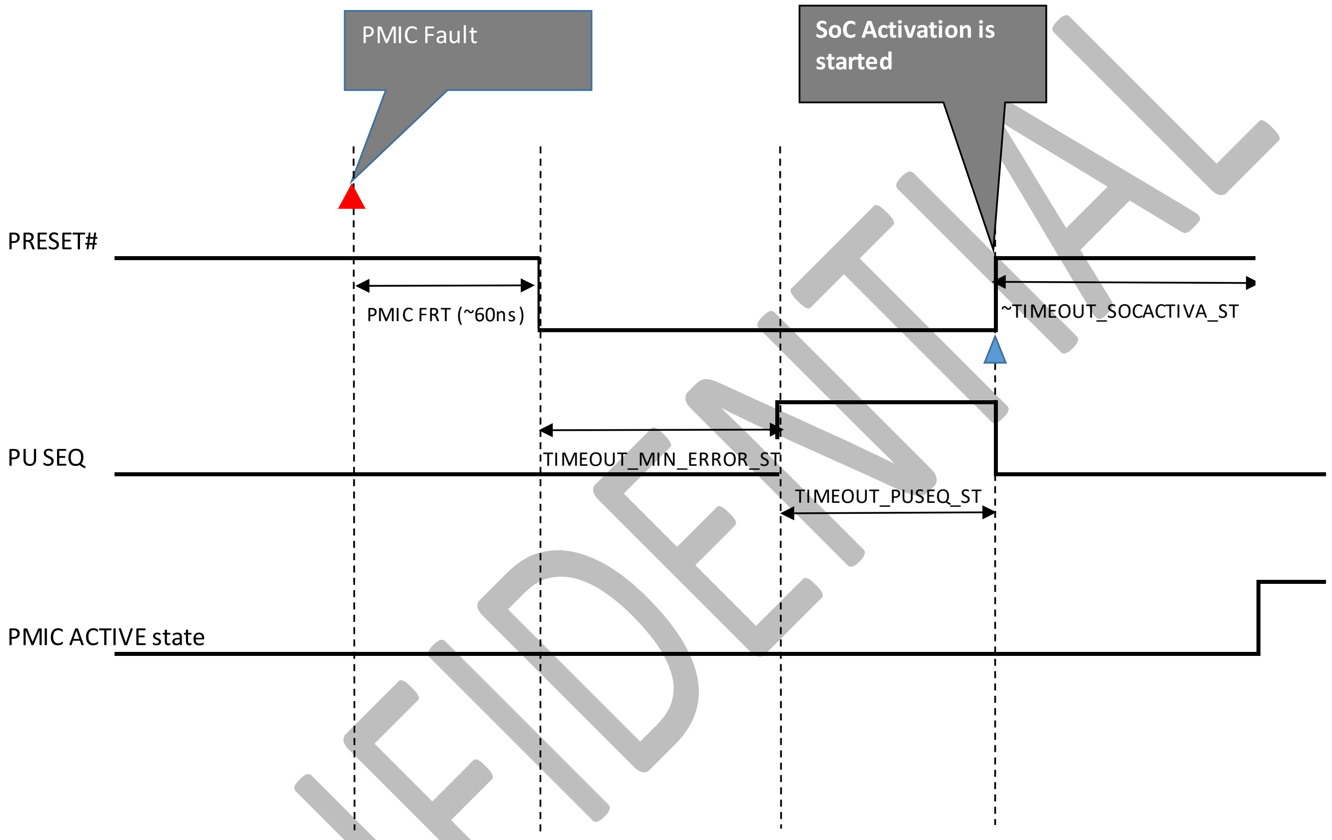
Figure 3 Recovery from RESET state



The PMIC transitions to ERROR state when a PMIC fault is detected by its on-chip safety mechanisms or when an SDI is configured to transition the PMIC to ERROR state.
- While the PMIC Error count has not reached the maximum threshold, the PMIC will re-try and recover from ERROR state, as described in Figure 41 labels TR_6_2 & TR_2_6. During recovery, the PMIC ensures that no PMIC faults are present at the entry to SoC Activation Sequence.
- The SoC & the rest of the system are in safe state while PMIC retries to recover. While in ERROR state, the PMIC outputs are disabled, PRESET# and the SDO's are engaged to ensure SoC is in safe state & system communication is disengaged.
- If the PMIC's recovery attempts are unsuccessful, the PMIC will transition to LOCK state. See **Appendix G. PMIC Recovery from LOCK State** for LOCK state recommendations.

The PMIC transitions to ERROR state after a fault is detected by PMIC safety mechanisms.
While the PMIC Error count has not reached the maximum threshold, the PMIC will re-try and recover from ERROR state, as described in the timing diagram below.

Figure 4 Recovery from ERROR state



6 Summary Table of RAA271005 Safety Mechanisms

The following table lists the Safety Mechanisms (SM) in the RAA271005 PMIC.
It includes the:

- SM ID number as referred to in the safety documents
- The SM name
- A short description of the SM
- The main element protected by the SM
- The SM classification
- Safe State Transition Mechanisms. These are on-chip or system level mechanisms that transitions the device to safe state when a fault is detected by on-chip SMs or system level SMs.
  - (1) Safe State transition mechanisms controlled by the PMIC are described in Section 8.These include: PMIC Error Pin: PRESET# and Safety Control.
  - (2) Safe State transition mechanisms which the System will trigger to bring the device to safe state are described in Section 9. These include: Regulator Disable Control, and Forcing Device to RESET or ERROR States.
- The state the PMIC transitions to when the SM detects a fault
- How latent faults of the SM is detected via its test concept
- The SM timing parameters – DTI & FRT.

SM Classification:

(1) The SM classified as Type "D": Flag available for Detection but action required by the application.

This category is used in cases like CRC over I2C. If a fault occurs this safety mechanism will only detect it. It will not be able to correct, or mask, the fault. If the transition to Safe state is not controlled by another on-chip element(s) an action from the application is needed. These kinds of SMs have to be handled with care because if they are used with regards to residual faults and the safety concept from Renesas does not consider an action following the detection, no coverage can be claimed (no protection against the faulty behavior). Such safety mechanisms will provide a positive contribution to both SPFM and LFM if the safety concept is robust enough.
Note: to understand if the transition to safe state is controlled by the PMIC directly please check column "Safe State Transition Mechanism" in the FMEDA.

(2) The SM classified as Type "C": Automatic Control with no flagging

This category is used in cases like redundancy with a voter and no flag. If a fault occurs, the voter will only control the data, but it will not flag its occurrence out; in this way the application layer will not be aware that a fault has occurred. Such safety mechanisms will provide a positive contribution to SPFM but a negative to LFM.

(3) The SM classified as Type "D & C": Automatic Control with flagging for Detection

This category is to be used in cases like redundancy with a voter and flag. If a fault occurs, this SM will not only control the data, but it will also flag its occurrence out; in this way the application is made aware that the voter has identified a discrepancy. Such safety mechanisms will provide a positive contribution to both SPFM and LFM.

Table 1 Summary Table of RAA271005 Safety Mechanisms

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 1	UV/OV Monitor	Detects over-voltage & under-voltage on Bucks, LDOs, Internal regulators, Bandgap references, and input supplies, when voltage at the sense pins rises above, or below defined thresholds.	Bucks & LDOs	D	Safety Control, PRESET#	ERROR	ADC BIST	10ms
SM 4	TempSensor2	Detects junction temperature of the protection block exceeding over-temperature thresholds.	Bucks & LDOs	D	Safety Control, PRESET#	ERROR	OT BIST	10ms

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 5	TempSensor3	A separate Temp sensor to serve as reference when self-testing Temp Sensor2.	Temp Sensor2	D	Safety Control, PRESET#	ERROR	OT BIST	10ms
SM 6	Clock Monitor	Monitors internal High Frequency & Low Frequency clocks. If the Clock Monitor detects a clock error, it ensures that digital operation needed to transition to safe state is not compromised.	HS CLK, LF CLK	D	Safety Control, PRESET#	ERROR	Continuous cross-check between HF & LF clocks	10ms
SM 12	Q&A Watchdog	Detects errant software or system hangs in the SoC. The Q&A Watchdog has 16 questions, each associated with a 32-bit expected answer.	SoC	D	Safety Control, PRESET#	RESET	LBIST	10ms
SM 13	OTP CRC & OTP Redundancy	Checks OTP error during self-diagnosis through an OTP CRC and a CRC DONE flag is set after each check.	OTP	D & C	-	ERROR	LBIST	10ms

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 16	SPI CRC	Detects invalid SPI transactions. The SoC adds protection information (CRC) to the data sent to RAA271005. The RAA271005 evaluates the received data and reports the result.	SPI	D	System will trigger the device to safe state	RESET	LBIST	1 SPI CLK Cycle + 62.5ns
SM 17	SPI message counter	A SPI message counter implemented for each write operation, for the SoC to verify.	SPI	D	System will trigger the device to safe state	-	LBIST	125ns
SM 18	I2C CRC	Detects invalid I2C transactions. The SoC adds protection information (CRC) to the data sent to RAA271005. The RAA271005 evaluates the received data and reports the result.	I2C	D	System will trigger the device to safe state	RESET	LBIST	5us+ 62.5ns
SM 19	I2C message counter	An I2C message counter implemented for each write operation for the SoC to verify.	I2C	D	System will trigger the device to safe state	-	LBIST	125ns

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 22	ADC BIST	Tests the 12bit SAR ADC, the test is done at startup. ADC conversion, the multiplexer that selects the signals to be monitored, and Fault Detection Logic are tested.	ADC	D	Safety Control, PRESET#	ERROR	-	10ms
SM 23	LBIST	Tests the Digital Protection circuitry to ensure viability through LBIST at startup.	Digital Core	D	Safety Control, PRESET#	ERROR	-	1/ Drive cycle
SM 24	Protection OT BIST	Tests TempSensor2 & TempSensor3 at startup to detect errors & faults.	Temp Sensor 2,3	D	Safety Control, PRESET#	ERROR	-	10ms
SM 26	SDO Test	RAA271005 and SoC test for the SDO pins.	SDO	D	System will trigger the device to safe state	ERROR	Startup Test	1/ Drive cycle
SM 27a	PRESET Check	Tests PRESET – PRESETOUT0 loop to ensure proper operation of SoC reset loop.	PRESET#	D	Safety Control, PRESET#	ERROR	-	1/ Drive cycle
SM 27b	Serial Interface Check	Checks communication interface for WRITE and READ operation.	Serial Interface	D	Safety Control, PRESET#	RESET	-	1/ Drive cycle

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 27c	External Pin Check2	Checks SoC error Signals for stuck at condition & verify connection.	SoC Error Signals	D	Safety Control, PRESET#	RESET	-	1/ Drive cycle
SM 27d	CVM Test	Checks the SoC CVM for correct operation, and connectivity to PMIC.	SoC Core Voltage Monitor (CVM)	D	Safety Control, PRESET#	RESET	-	1/ Drive cycle
SM 31	SoC CVM Monitor	Monitors SoC CVM Error detection status via CVM Error output pins.	SoC Core Voltage Monitor	D	Safety Control, PRESET#	RESET	ADC BIST, LBIST, FAULT Detect BIST	10ms
SM 32	OC Monitor	Detects OC condition & tristates the Buck output						
SM 33	Current Limit	Prevent current at the LDO output when an over current condition is present.						
SM 41	IRQ Test	Tests IRQ pin for correct operation.	IRQ Pin	D	System will trigger the device to safe state	RESET	-	1/ Drive cycle
SM 44	Gate Control (GC) Self Test	Tests the GC pin for stuck condition	GC Pin	D	Safety Control, PRESET#	ERROR	-	1/ Drive cycle

SM No.	SM Name	SM Description	Main Element Protected	SM Type	Safe State Transition Mechanism	PMIC State*	Test Concept	DTI + FRT**
SM 45	Gate Control (GC)	Signal to control an input supply cut-off switch	PMIC	D & C	Signal output, Safety Control, PRESET#	ERROR	GC Self Test	10ms

* See description of the different PMIC states in Section RAA271005 Safe States.

** ADC-related DTI is affected by the IIR filter configuration, which is a system integrator consideration. In the PMIC safety concept, the DTI + FRT of ADC-related SMs are =/<10ms.

7 Details of RAA271005 On-chip Safety Mechanisms

7.1 SM1: UV/OV Monitor

a) Overview

The RAA271005 makes use of an on-board successive approximation (SAR) 12-bit analog to digital converter (ADC) to continuously monitor Over voltage (OV) & Under voltage (UV) condition on the RAA271005 output rails, internal regulators (LDOs), references (Bandgaps), GND, input supplies: AVIN, PVIN, and Temperature Sensor outputs. The DC Monitoring Accuracy via ADC is +/-1.5%.

Table 2 ADC Channel Allocation

Channel #	Channel name	Alias used in Register Map
0	Internal ADC Offset Measurement (For Renesas Internal use)	Offset
1	Differential measurement of TEMP SENSOR2	Temp2_SENSOR, Temp2
2	Differential measurement of TEMP SENSOR3	Temp3
3	VBG Regu	BG_REGU
4	PGND3	PGND_Regu
5	External LDO1	ExtLDORegu_0
6	External LDO2	ExtLDORegu_1
7	External LDO3	ExtLDORegu_2
8	External LDO4	ExtLDORegu_3
9	External LDO5	ExtLDORegu_4
10	External LDO6	ExtLDORegu_5
11	Internal LDO Prot1	IntLDOProt_0
12	Internal LDO Prot2	IntLDOProt_1
13	ADC5-ADC4 Differential measurement	-
14	AVIN2	AVIN2_Prot
15	AUX1 (Used for measuring external ADC channels 1-5 or 1-16)	ExtINPs
16	AVIN1	AVIN1_Regu
17	Internal LDO Regu1	IntLDORegu_0
18	Internal LDO Regu2	IntLDORegu_1
19	Internal LDO Regu3	IntLDORegu_2
20	Internal LDO Regu4	IntLDORegu_3
21	PVIN1	Buck_1_PVIN

22	VOUT1	Buck_1_VOUT
23	PVIN2	Buck_2_PVIN
24	VOUT2	Buck_2_VOUT
25	PVIN3	Buck_3_PVIN
26	VOUT3	Buck_3_VOUT
27	PVIN4	Buck_4_PVIN
28	VOUT4	Buck_4_VOUT
29	PVIN5	Buck_5_PVIN
30	VOUT5	Buck_5_VOUT
31	AUX2 (Used for measuring external ADC channels 1-5 or 1-16)	ExtINPs
32	Internal LDO Regu5	IntLDORegu_4
33	Internal LDO Regu6	IntLDORegu_5
34	Temp Sensor 4	Temp4
35	Spare1	SPARE_1

Additional 5 ADC channels are available to monitor external signals, including the error signals from the SoC Core Voltage Monitor (CVM): VTHREF0 and VTHSENSE0. Using an external multiplexer, up to 16 external channels that can be used for additional monitoring. See **RAA271005 Datasheet Section 5 Monitoring ADC.**

In order to address Common Cause Failures, the ADC resides inside the Protection Block, an independent unit in the RAA271005, with its own separate Bandgap reference, internal regulators, clocks, and digital core.

A Fault Detection Logic monitors the output voltage, taken at the associated sense pins, and reports a fault if the reading rises above the defined over-voltage threshold, or drops below the defined under-voltage threshold. The RAA271005 output OV and UV thresholds are stored in the **ADC Monitor Limit Registers (*ADCMON*)**. These are OTP programmed at startup and cannot be modified during operation.

The SoC is recommended to read ADC measurement results using the **ADC Monitor Copy Registers** (See Appendix D).

ADC is operational in Full Run mode only.

The Register details are found in **Register Map.**

Detection Time Interval (DTI):

Minimum DTI is equivalent to the ADC sampling rate in Table 4. This time is affected by the Digital Filter (IIR Coefficient) setting, at the discretion of the system integrator.

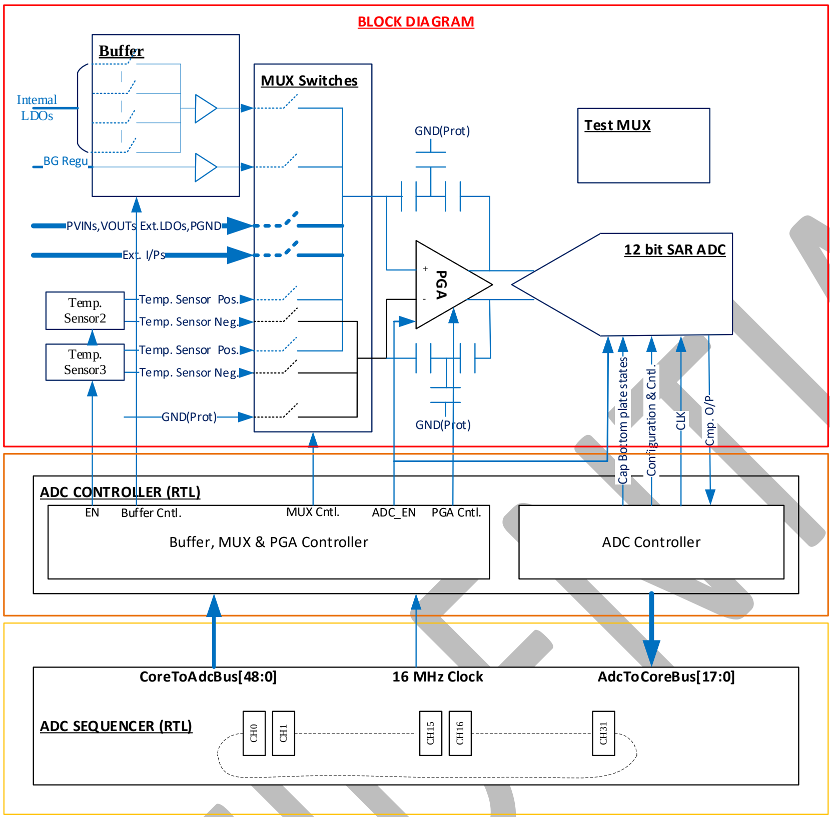
ADC AUXMODE

In the ADC Auxmode, external ADC channels 1 to 8 are maskable so that a detected fault would not lead to PMIC state transition directly.

External ADC channels 9 to16 do not have dedicated registers, and can only be controlled by SoC by software.

b) Hardware Description

Figure 5 RAA271005 Monitoring System



This 12-bit SAR ADC and associated mux, Programmable Gain Amplifier (PGA), Controller, and Sequencer

Information about using the 12-bit ADC can be found in the **RAA271005 Datasheet Section 5 Monitoring ADC**

There are 8 PGA settings available to maximize the ADC full scale range for each channel.
Each ADC channel has a default PGA gain setting. These gain settings can be programmed individually via **ADCMON_Gain_IIRCoeff_<*MonitoredOutput*>** registers. Where: ***MonitoredOutput*** are the different ADC channels described in Table 2. The recommended PGA gain settings for different types of measurements are described in **Table 23 of RAA271005 Datasheet.**

ADC Digital filter

An IIR filter is applied to smooth glitches and reduce noise, and as the number of averaging (IIR coefficient value) is increased, accuracy is improved. **However, increasing the IIR coefficient value, increases detection time, and increases possibility of faults being missed. This shall be a careful system consideration.**

The IIR Filter Coefficient setting is set in the **ADCMON_Gain_IIRCoef_< *MonitoredOutput* >[2:0] ADC_IIR_Coeff_< *MonitoredOutput* >** registers, where **< *MonitoredOutput* >** are the different ADC channels described in Table 2.

The PMIC Fault Detection Logic evaluates the ADC Average output value to determine PASS/FAIL status, as shown in the equation below:

$$Average[n] = \frac{(Sample[n] - Average[n-1])}{2^m} + Average[n-1]$$

Where:

m = IIR filter coefficient programmable from 0 to 7

n = n^{th} *sample time*

n-1 = $(n\text{-}1)^{th}$ *sample time*

Table 3 IIR Coefficient Settings

ADC_IIR_Coeff_<*MonitoredOutput*>:ADCMON_Gain_IIRCoef_<*MonitoredOutput* >[2:0] register value	**m**	**Number of Samples Averaged**
0x0	0	No averaging
0x1	1	2 samples averaged
0x2	2	4 samples averaged
0x3	3	8 samples averaged
0x4	4	16 samples averaged
0x5	5	32 samples averaged
0x6	6	64 samples averaged
0x7	7	128 samples averaged

The recommended optimum IIR Filter Coefficient (m) setting is 4.

ADC Sampling Interval and Sampling Rate

The ADC conversion time is 1.25usec per conversion. The ADC-based monitoring system measures 36 signals in sequence per monitoring iteration. Of the 36 ADC measurements per monitoring iteration, 33 are dedicated for the internal signals and 2 for external signals. One signal is left as a spare. Each internal signal that is monitored is sampled once every monitoring iteration (45us). The slot for monitoring external channels come every 22us. If there are 5 external channels, each signal is sampled once per 110us, or once per 360usec for 16 external channels. **This sequence is not configurable.**

The table below summarizes the ADC sampling rate. More information can be found in the **RAA271005 Datasheet Section 6.1.5 Sampling Rate**

Table 4 ADC Sampling Rate for IIR Coefficient = 0

PMIC OUTPUTS	**PMIC INTERNAL**	**ADC 1-5 Pins Without MUX**	**ADC 1-2 Pins With MUX**
45us	45us	110us	360us

Because of the cyclic nature of the ADC sampling, the different signals are sampled at various points in time, eliminating any frequency dependency between the measured signal and the sampling.

Debouncing

There is no additional delay applied to the monitored signals. The user has to tune the IIR filter coefficients to prevent voltage spikes, generating unwanted OV faults.

Fault Detection logic

This is responsible for comparing the ADC measured value against OV & UV thresholds to detect out of spec conditions, updates fault registers, and report the status to the Safety Control Logic.
The RAA271005 output OV and UV thresholds can be changed by adjusting the 16-bit values in the **ADC Monitor Limit Registers**. See details in Table 5 Output OV/UV Threshold Registers.

Safety Control logic

This controls fault reaction based on the error condition detected.
The RAA271005 fault response to specific fault conditions shall be configured by the system integrator via OTP. These registers are not accessible during normal operation and are protected by password & lockout.

c) Recommended Usage

The safety mechanism shall be active before the device performs a safety – related operation and continue to remain active.

The system integrator shall determine the optimum ADC Digital Filter setting depending on the system requirements. The system integrator shall take the sampling rate in Table 4 **and multiply this with the corresponding number of samples in** Table 3 **to determine the total detection time.**

For accuracy improvement, the PGA gain shall be chosen such that the input to the ADC is closer to 0.8V.

The PMIC does not proceed to ACTIVE state if any of the Built in Self Tests (BIST) fails at startup. This can be confirmed by checking that LBIST, ADC BIST and Fault Detect BIST passed at startup. See Section 7.11 and Section 7.12 for the registers that hold the status of BIST.

d) Fault Control and Operation

See Table 36, for details on how the PMIC outputs behave according to faults detected, and the Safe State transitioned to.

The RAA271005 response to an over-voltage or under-voltage condition is controlled though thresholds, stored in registers, that are used to compare the corresponding voltage reading against.
For example, for the Buck outputs, the Fault Detection Logic will compare the measured voltage against the values stored in 4 registers: 2 registers hold the high threshold, and 2 hold the low threshold.

Table 5 Output OV/UV Threshold Registers

REGISTER	DESCRIPTION
VOUT_<*Number*>_ADCMON_LimHighMSB_VOUT[7:0]: ADCMON_LimHighMSB_VOUT	holds the upper byte of the over voltage threshold
VOUT_<*Number*>_ADCMON_LimHighLSB_VOUT[7:0]: ADCMON_LimHighLSB_VOUT	holds the lower byte of the over voltage threshold.
VOUT_<*Number*>_ADCMON_LimLowMSB_VOUT[7:0]: ADCMON_LimLowMSB_VOUT	holds the upper byte of the under voltage threshold
VOUT_<*Number*>_ADCMON_LimLowLSB_VOUT[7:0]: ADCMON_LimLowLSB_VOUT	holds the lower byte of the under voltage threshold
extLDO_<*LDO#*>_ADCMON_LimHighMSB_ExtLDO[7:0]: ADCMON_LimHighMSB_ExtLDO	holds the upper byte of the over voltage threshold
extLDO_<*LDO#*>_ADCMON_LimHighLSB_ExtLDO[7:0]: ADCMON_LimHighMSB_ExtLDO	holds the lower byte of the over voltage threshold.
extLDO_<*LDO#*>_ADCMON_LimLowMSB_ExtLDO[7:0]: ADCMON_LimHighMSB_ExtLDO	holds the upper byte of the under voltage threshold
extLDO_<*LDO#*>_ADCMON_LimLowLSB_ExtLDO[7:0]: ADCMON_LimHighMSB_ExtLDO	holds the lower byte of the under voltage threshold

Where: *Number* corresponds to the following Buck outputs:

Number	Outputs
0	Buck1
1	Buck2
2	Buck3
3	Buck4
4	Buck5

LDO# corresponds to the following LDO outputs:

Number	Outputs
0	LDO1
1	LDO2
2	LDO3
3	LDO4
4	LDO5
5	LDO6

When the ADC reading exceeds the high threshold, or falls below the low threshold, the Fault Detection Logic will record the undervoltage (UV) or overvoltage (OV) condition by setting the corresponding **FLT_RECORD_<*MonitoredOutput*>[n:m]: FaultStatus_<*MonitoredOutput*>** bit in the **Fault Record Registers: 0x019 to 0x021**.

Where: ***MonitoredOutput*** are the different ADC channels described in Table 2, and ***n***, ***m*** are the assigned field bits.

Specifically, for Bucks1-5 and LDO1-6 output voltages, the following status bits are set:

Table 6 Output OV/UV Fault Registers

BUCK	STATUS BIT
Buck1	**FLT_RECORD_BUCKS_B[1]: FaultStatus_Buck_1_VOUT**
Buck2	**FLT_RECORD_BUCKS_A[1]: FaultStatus_ Buck_2_VOUT**
Buck3	**FLT_RECORD_BUCKS_A[3]: FaultStatus_ Buck_3_VOUT**
Buck4	**FLT_RECORD_BUCKS_A[5]: FaultStatus_ Buck_4_VOUT**
Buck5	**FLT_RECORD_BUCKS_A[7]: FaultStatus_ Buck_5_VOUT**
LDO1	**FLT_RECORD_ExtLDOs[0]: FaultStatus_ExtLDORegu_0**
LDO2	**FLT_RECORD_ExtLDOs[1]: FaultStatus_ExtLDORegu_1**

LDO3	**FLT_RECORD_ExtLDOs[2]: FaultStatus_ExtLDORegu_2**
LDO4	**FLT_RECORD_ExtLDOs[3]: FaultStatus_ExtLDORegu_3**
LDO5	**FLT_RECORD_ExtLDOs[4]: FaultStatus_ExtLDORegu_4**
LDO6	**FLT_RECORD_ExtLDOs[5]: FaultStatus_ExtLDORegu_5**

The Regulation block is informed to disable the outputs. The fault status is also indicated to the Safety Control Logic which controls the following UV or OV condition response within FTTI:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

e) Test Concept

See Section 7.11 for details on ADC BIST

7.2 SM4: TempSensor2 & TempSensor4

a) Overview

RAA271005 detects the junction temperature of the device with an accuracy of +/-4deg, through two temperature sensors located at two PMIC hot spots. TempSensor2 is in the Protection unit & TempSensor4 is in the Regulation unit. The outputs of both temperature sensors are continuously monitored by the ADC and compared against SHUTDOWN Thresholds, both set by default at 150deg Celsius.

A WARNING Threshold can be set for TempSensor2 only. The WARNING threshold shall be set by the system integrator according to the system application temperature profile.

The RAA271005 temperature thresholds are stored in the **ADC Monitor Limit Registers**.
These are OTP programmed at startup and cannot be modified during operation.

Detection Time Interval (DTI):
Minimum DTI is equivalent to the ADC sampling rate in Table 4. This time is affected by the Digital Filter (IIR Coefficient) setting, at the discretion of the system integrator.

b) Hardware Description

TempSensor2 and TempSensor4 are part of the Monitoring System in the Protection block (Figure 5). All the user configurable options discussed in Section 7.1 applies here as well.

TempSensor2 and TempSensor4 temperature shutdown & warning thresholds are programmable from 0 to 255degC. 1LSB = 0.25degC.

TempSensor2 SHUTDOWN threshold range, is bounded by:

Table 7 TempSensor2 Temperature Shutdown Threshold Registers

REGISTER ADDR	REGISTER	DESCRIPTION
0x13C	**ADCMON_ShutDNLimitMSB_Temp[7:0]: ADCMON_ShutDNLimitMSB_Temp**	upper byte of the SHUTDOWN threshold
0x13D	**ADCMON_ShutDNLimitLSB_Temp[7:0]: ADCMON_ShutDNLimitLSB_Temp**	lower byte of the SHUTDOWN threshold

TempSensor4 SHUTDOWN threshold range, is bounded by:

Table 8 TempSensor4 Temperature Shutdown Threshold Registers

REGISTER ADDR	REGISTER	DESCRIPTION
0x140	**ADCMON_ShutDNLimitMSB_Temp4[7:0]: ADCMON_ShutDNLimitMSB_Temp 4**	upper byte of the SHUTDOWN threshold
0x141	**ADCMON_ShutDNLimitLSB_Temp4[7:0]: ADCMON_ShutDNLimitLSB_Temp4**	lower byte of the SHUTDOWN threshold

Tempsensor2 WARNING threshold range, is bounded by:

Table 9 TempSensor2 Temperature Warning Threshold Registers

REGISTER ADDR	REGISTER	DESCRIPTION
0x13E	**ADCMON_WarnLimitMSB_Temp[7:0]: ADCMON_WarnLimitMSB_Temp**	upper byte of the WARNING threshold
0x13F	**ADCMON_WarnLimitLSB_Temp[7:0]: ADCMON_WarnLimitLSB_Temp**	lower byte of the WARNING threshold

c) Recommended Usage

[1] Both TempSensor2 and TempSensor4 are recommended to be enabled (not masked) for more accurate detection of an over temperature condition.
The safety mechanism shall be active at startup and continue to remain active.
If the system integrator has activated OV/UV monitoring via ADC, the Temperature monitoring is activated as well.

[2] If the system integrator wishes to change the default TempSensor2 & TempSensor4 Over Temperature SHUTDOWN (OT_SHUTDOWN) or TempSensor2 WARNING (OT_WARN) thresholds, the system integrator shall be responsible for making sure that the thresholds selected follow the allowed temperature range of the PMIC, and that LOCKOUT is implemented so that the settings do not change during normal operation.

[3] A status of “0x01” on **FLT_RECORD_BG_Temp[2]: FaultStatus_TempWarn,** shall trigger system-level measures to help bring temperatures down and avoid an OT shutdown event.

[4] Thresholds shall be set such that:

OT_WARN < OT_SHUTDOWN (SOC ACTIVATION) < OT_SHUTDOWN

OT_WARN < Regulation Over-temp Falling threshold

If the system integrator does not wish to follow the recommended threshold settings, the system integrator shall be responsible for making sure that the thresholds selected follow the allowed temperature range of the PMIC for smooth transition in and out of an OT condition.
ie. If OT_WARN > Regulation Over-temp Falling threshold, The PMIC will not restart until temperature is below Regulation Over-temp Falling threshold.

[5] Temperature Monitoring during SoC Activation Sequence:
During SoC Activation Sequence a separate set of temperature shutdown threshold can be used by the system, which is configured via the registers:

Table 10 SoC Activation Temperature Shutdown Threshold Registers

REGISTER	DESCRIPTION
ADCMON_ShutDNLimitMSB_Temp_2B[7:0] **ADCMON_ShutDNLimitMSB_Temp_2B**	Upper byte of the SHUTDOWN threshold applied during SoC Activation state
ADCMON_ShutDNLimitLSB_Temp_2B[7:0] **ADCMON_ShutDNLimitLSB_Temp_2B**	Lower byte of the SHUTDOWN threshold applied during SoC Activation state

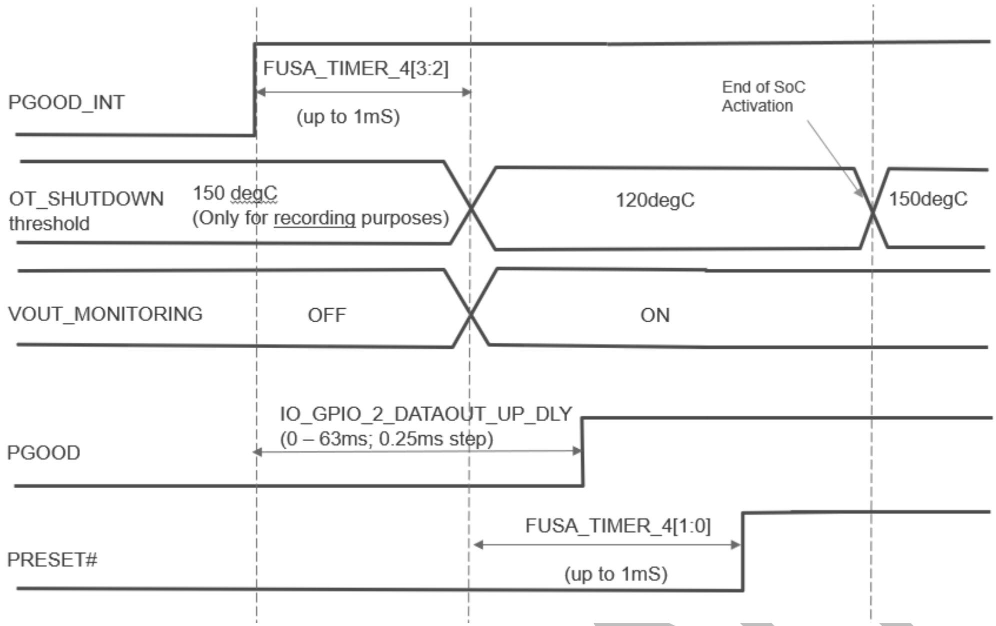
[6] To prevent SoC startup when junction temperature > OT Shutdown SoC Activation Level, during Cold Start & Warm Start:

a) FUSA_TIMER_4[3:2] ≥ 750us to guarantee sufficient samples for ADC monitoring

b) IO_GPIO_2_DATAOUT_UP_DLY - FUSA_TIMER_4[3:2] ≥ 100us to guarantee PGOOD stay low @ faults & monitoring spec at PGOOD high transition

c) {FUSA_TIMER_4[3:2] + FUSA_TIMER_4[1:0]} > IO_GPIO_2_DATAOUT_UP_DLY to guarantee the PGOOD-PRESET Timing

Figure 6 Timing recommendations



d) Fault Control and Operation

The Fault Detection Logic compares the temperature sensors output against the values stored in the SHUTDOWN threshold and WARNING threshold registers.

If the OT WARNING threshold is exceeded during ACTIVE state, the **FLT_RECORD_BG_Temp[2]: FaultStatus_TempWarn** bit in the status register is set, the device continues operation & monitors temperature. If the junction temperature continues to rise and exceeds the set OT SHUTDOWN threshold set for TempSensor2, the **FLT_RECORD_BG_Temp[3]: FaultStatus_TempShdn** status bit is set, and the device goes to ERROR state.

If the die temperature exceeding the set OT SHUTDOWN threshold for TempSensor4 is detected by TempSensor4, the **FLT_RECORD_BG_Temp[1]: FaultStatus_Temp4_Shdn_Regu** status bit is set, and the device goes to ERROR state.

When an OT SHUTDOWN condition is detected by either TempSensor2 or TempSensor4 during ACTIVE state, the Safety Control Logic controls the following within FTTI:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate)

When an OT WARNING condition is detected during ACTIVE state, IRQ is asserted.

When PMIC recovers from ERROR state, PMIC junction temperature below OT WARNING threshold is one of the transition conditions to transition from ERROR State to PowerUp sequence. If TempSensor2 WARNING is masked via register **ADCMON_MASK_BG_TEMP[2]: FaultMask_Temp2_Warn_Prot** = 1, this transition condition is ignored.

e) Test Concept

The test concept of TempSensor2 involves comparing its output against the result of the redundant TemperatureSensor3.

The TempSensor2 & TempSensor3 test fails when the discrepancy between the two temperature sensors is greater than the set allowed difference, or when the TempSensor reading is 0V. A 0V reading of any or both of the sensors can indicate a fault in the temperature sensors, a failure in the sensor Enable signals, or an error in the temp sensors' supply. See Section 7.13.

There is no test concept for TempSensor4. Latent faults of TempSensor4 are covered by the redundancy offered by TempSensor1, TempSensor2 and TempSensor3.

7.3 SM5: TempSensor3

a) Overview

Temperature Sensor 3 is a redundant implementation of TempSensor2. Its hardware description, and operation are similar to Temperature Sensor2. RAA271005 uses Temperature Sensor3 as the reference sensor to check TempSensor2 output at startup and continuously during normal operation.

b) Hardware Description

Same as TempSensor2.

c) Recommended Usage

The safety mechanism shall be active at startup and continue to remain active.
The safety Mechanism is used in conjunction with SM24: Protection OT BIST.

d) Fault Control and Operation

SM24 Fault Control & Operation.

e) Test Concept

Protection OT BIST.

7.4 SM6: Clock Monitor

a) Overview

The Clock monitor is made up of 4 sub-monitors to detect internal clock frequency errors. Both the regulation and protection blocks of the RAA271005 have one HF (32MHz) and one LF (32KHz) independent oscillator. The reference clock is the 32MHz clock in each block, the PROT_32M in the Protection block or REGU_32M in the Regulation block.

The 2 HF clocks (Protection 32M and Regulation 32M) continuously checks the 2 LF clocks (Protection 32K and Regulation 32K) frequency, across each clock, by applying the frequency counter concept to check the LF Clock frequency relative to the HF clock frequency. A clock fault is reported if the HF clock / LF clock ratio is > (+/-) 50% of the ideal ratio (which is 1000).

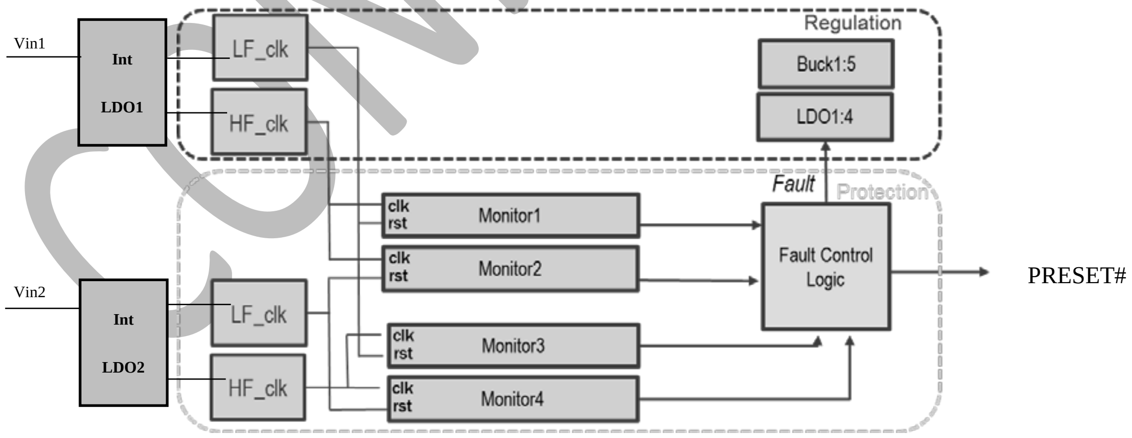
When a clock fault is detected on the Protection block's 32MHz clock, that clock is switched with the Regulation block's 32MHz clock to ensure that the transition to safe state is completed correctly.

Clock Stuck detection is not enabled in this chip revision.

Detection Time Interval (DTI) of the Clock Monitor is 62.5ns and is fixed.

b) Hardware Description

Figure 7 Clock Monitor Architecture



Monitor1: Checks REGU_32M & REGU_32K
Monitor2: Checks REGU_32M & PROT_32K

Monitor3: Checks PROT_32M & REGU_32K
Monitor4: Checks PROT_32M & PROT_32K
The LF_clk & HF_clk in Regulation and Protection units have different architectures to mitigate any common cause of failure between them. In addition, the cross-monitoring is done between clocks in the Regulation unit and the independent Protection unit. The LF_clk & HF_clk in the protection unit have a separate internal voltage domain, and independent reset sources.

The monitoring thresholds can be configured to 2 possible options via Bit 7 in the **FUSA_CTRL_A[7:5]: CLK_MON_CTRL** register:

Table 11 Clock Monitor Configuration

BIT	NAME	R/W	DESCRIPTION
7:5	CLK_MON_CTRL	RW	[Bit 7] : clk mon threshold 0x0 : clock monitoring threshold narrow: freq diff : 12.5% < (warning region) < 25%, beyond 25% is fail. 0x1 : clock monitoring threshold wide: freq diff : 25.0% < (warning region) < 50%, beyond 50% is fail. [Bit 6] : enable for clock monitoring based on Prot 32M clock 0x0 : clk mon disable 0x1 : clk mon enable [Bit 5] : enable for clock monitoring based on Regu 32M clock 0x0 : clk mon disable 0x1 : clk mon enable

The thresholds are set by OTP and cannot be modified during normal operation.

c) Recommended Usage

The safety mechanism shall be active before performing a safety operation and continue to remain active.
The system integrator ensures this by setting the following registers in OTP.
FUSA_CTRL_A[5]:CLK_MON_CTRL= 0x1
FUSA_CTRL_A[6]:CLK_MON_CTRL= 0x1

The PMIC does not proceed to ACTIVE state if any of the Built in Self Tests fail at startup.

The system integrator shall poll **FLT_RECORD_B[3]: FLT_ClkMon** to detect if the PMIC was not able to successfully transition to a working Protection 32MHz clock if this clock failed.

d) Fault Control and Operation

Monitor1: REGU_32M checks REGU_32K.A frequency error can indicate an error in either REGU_32M or REGU_32K.
Monitor2: REGU_32M checks PROT_32K. A frequency error can indicate an error in either REGU_32M or PROT_32K.
Monitor3: PROT_32M checks REGU_32K. A frequency error can indicate an error in either PROT_32M or REGU_32K.
Monitor4: PROT_32M checks PROT_32K. A frequency error can indicate an error in either PROT_32M or PROT_32K.

The Safety Control Logic (Fault Control Logic in Figure 7) performs additional analysis on the results of Monitor1 – 4 to identify the failing clock:

- PROT_32K clock is wrong, if an error is detected by Monitor 2 and Monitor 4
- REGU_32K clock is wrong, if an error is detected by Monitor 1 and Monitor 3
- PROT_32M clock is wrong, if an error is detected by Monitor 3 and Monitor 4
- REGU_32M clock is wrong, if an error is detected by Monitor 1 and Monitor 2

If PROT_32M clock is wrong, the Safety Control Logic switches the HF clock in the Protection block to use the HF clock in the Regulation block.

The response to a clock fault condition is controlled though the selected monitoring threshold stored in the **FUSA_CTRL_A[7]: CLK_MON_CTRL** register. Monitor1-4 compare the actual measured frequency against the selected thresholds and indicate an error when measured frequency falls outside the thresholds.

If any of the Monitor1-4 indicates an error, the Fault Detection Logic sets the **FLT_RECORD_B[3:3]: FLT_ClkMon** register, informs the Regulation block to disable the outputs, and informs the fault condition to the Safety Control Logic to control the following within FTTI:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay

- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

e) Test Concept

The Clock monitor is tested at startup via LBIST. See more details on LBIST in Section 7.12

7.5 SM12 Question-Answer Watchdog (Q&A WDT) for SoC Monitoring

a) Overview

RAA271005 monitors the SoC's proper operation through the Q&A Watchdog. It can be configured into 3 modes via the WDT_CFG0[7:0] register

Table 12 WDT Configuration

BIT	NAME	R/W	DESCRIPTION
7	WDT_DIS_LFSR	RW	0: Question value is generated by Pseudo random 1 :Question value generated by Sequential order
4:3	WDT_KICK_SEL	RW	0 = I2C/SPI only
2	WDT_WWDT_ADV_MODE	RW	0: Basic Windowed kick mode 1: Q&A mode
1	WDT_WWDT_ADV_16Q	RW	0: 4 Q&A mode 1: 16 Q&A mode
0	WDT_WWDT_EN	RW	enable bit for watchdog timer 0: Disable WDT 1: Enable WDT

The Watchdog operation in 16Q&A and 4Q&A modes is described in Figure 12.

Detection Time Interval (DTI) of the Q&A WDT is affected by the Configurable watchdog window + 100ns.

b) Hardware Description

Figure 8 Watchdog Hardware Architecture

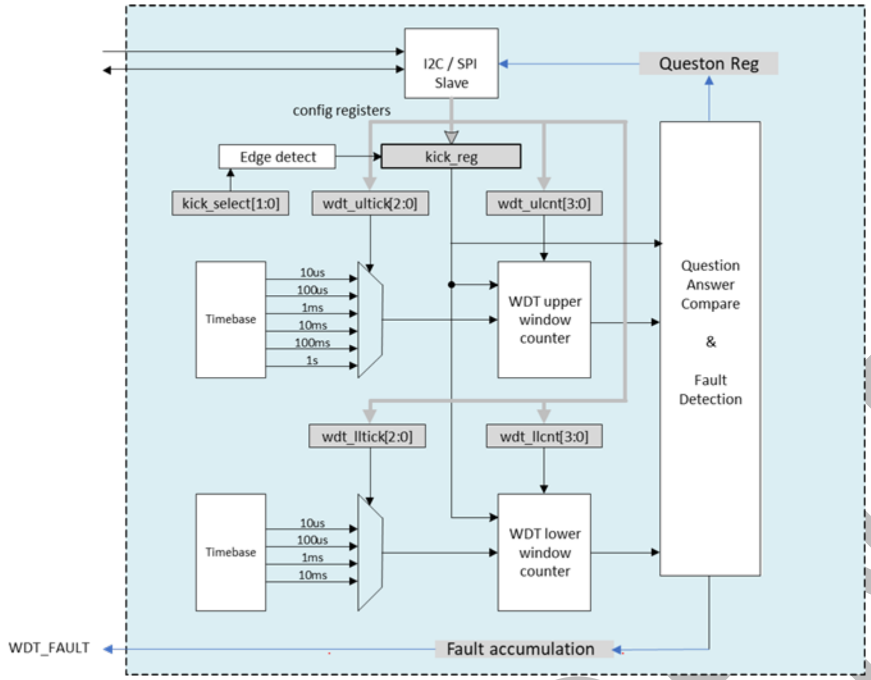
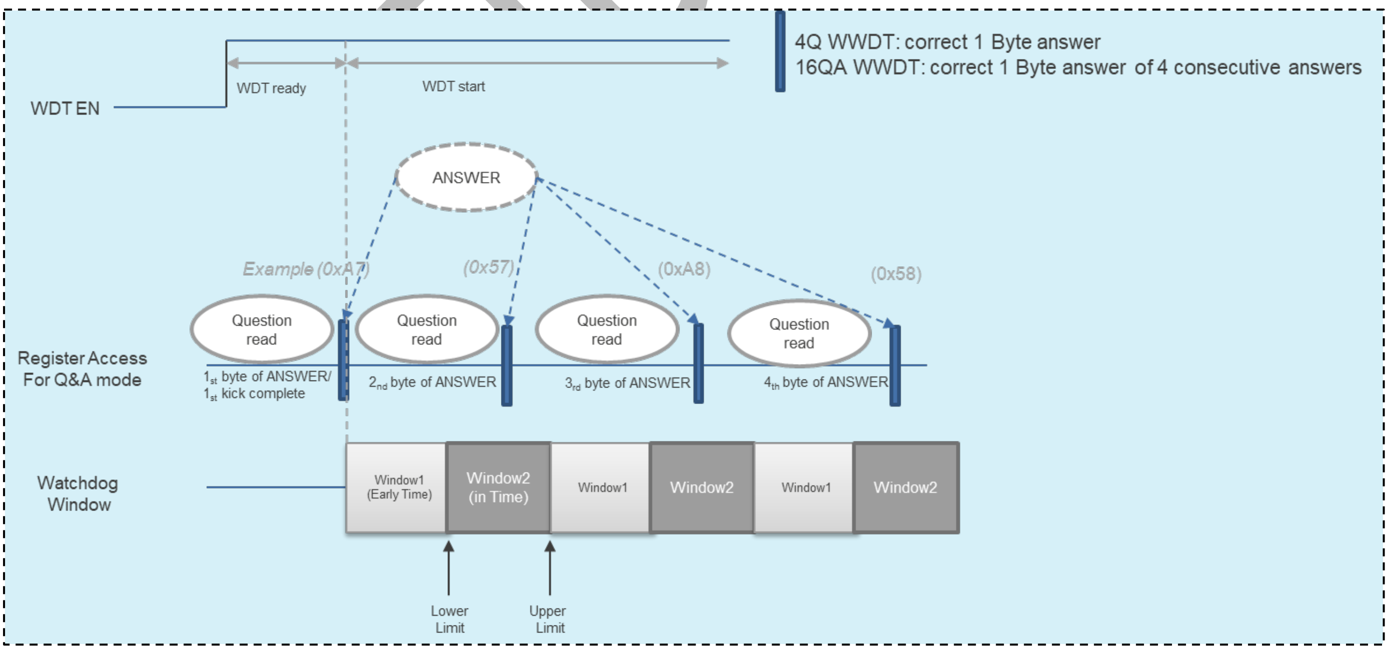


Figure 9 Watchdog Timing Diagram

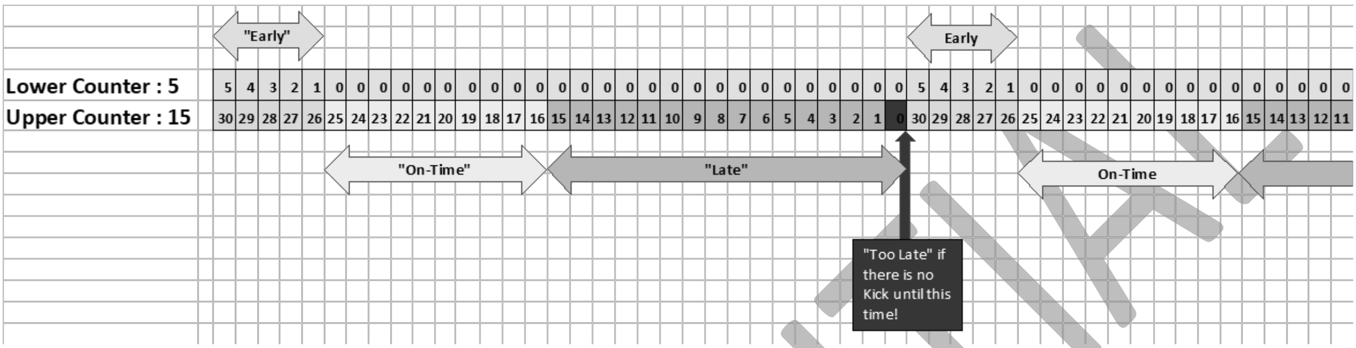


Question Read is optional while answering the remaining 4 bytes of answer.

Each byte of the ANSWER shall be written one byte at a time within its allocated Window2

Figure 10 shows an example of an “Early”, “On-Time”, “Late”, and “Too late” response with respect to the configured WDT window.

Figure 10 WDT Window Configuration Example



The WDT timer is reset after PMIC receives each byte of answer.

c) Recommended Usage

Recommended Usage for the 3 Watchdog modes are in the following sections.

16 Question Q&A Mode Configuration

[1] The Q&A WDT is active at startup waiting to receive the first answer from the SoC.
This is configured in OTP via register **WDT_CFG0[0:0]: WDT_WWDT_EN = 0x1**.

[2] Since the Q&A process is done over the communication interface (SPI/I2C), SPI/I2C CRC shall be included in each transaction.
CRC is configured in OTP via register **IO_FUNC_CFG [0:0]: REGCRC_EN = 0x1**.

Details on Communication Interface CRC is described in the following sections: Section 7.7, for SPI and Section 7.9, for I2C.

[3] Setting the Watchdog Window:

The WDT upper window and lower window is configured by setting the following registers:

WDT_CFG1[7:4]: WDT_ULCNT	- WDT Upper Time Limit (unitless) can be set from 1 to 15
WDT_CFG1[3:0]: WDT_LLCNT	- WDT Lower Time Limit (unitless) can be set from 1 to 15
WDT_CFG2[5:3]: WDT_ULTICK	- WDT Window Upper Limit Timing Unit
WDT_CFG2[2:0]: WDT_LLTICK	- WDT Window Lower Limit Timing Unit

The WDT upper window = WDT Upper Time Limit * WDT Window Upper Limit Timing Unit
= **WDT_ULCNT * WDT_ULTICK**
The WDT lower window = WDT Lower Time Limit * WDT Window Lower Limit Timing Unit
= **WDT_LLCNT * WDT_LLTICK**

The WDT Window Upper and Lower Timing Units have to be configured from the default value of 0x00.

The WDT Window Upper Timing Unit can be configured to any of the following, by setting the **WDT_CFG2[5:3]: WDT_ULTICK** register as:

Table 13 WDT Window Upper Timing Unit Configuration

REGISTER SETTING	WDT UPPER TIMING UNIT SETTING
0x01	10u
0x02	100u
0x03	1m
0x04	10m
0x05	100m
0x06	1s

The WDT Lower Timing Unit can be configured by setting **WDT_CFG2[2:0]: WDT_LLTICK** register as:

Table 14 WDT Window Lower Timing Unit Configuration

REGISTER SETTING	WDT LOWER TIMING UNIT SETTING
0x01	10u
0x02	100u
0x03	1m
0x04	10m
0x05	100m

[4] To mitigate the clock discrepancy between the SoC clock & the RAA271005 clock, and in order to set optimum window limits, it is recommended to do a characterization of the SPI /communication latency in the SoC, preferably with no other SoC operation running. This information can be compared to the RAA271005 clock data which can be derived from the RAA271005 Free- Running Counters:

Table 15 RAA271005 Clock Data

REGISTER ADDR	REGISTER	DESCRIPTION
0x017	**CLK_CNT_1[7:0]: FREERUN_CNT_UPPER**	upper byte of free running counter (32MHz)
0x018	**CLK_CNT_2[7:0]: FREERUN_CNT_LOWER**	lower byte of free running counter (32MHz)

[5] The maximum WDT error threshold is stored in the **WDT_CFG2[7:6]: WDT_WWDT_ACC_TH** register. The error threshold can be set to the following values: 1, 15, 31 and 63.

16 Question Q&A Mode Operation

At the end of a successful SoC Activation Sequence, the SoC shall write **WDT_KICK_REG[7:0] = 0xA7, 0x57, 0xA8, 0x58** to officially start the Question & Answer process between the PMIC & the SoC.

Successful WDT ANSWER received is informed by PMIC via register **FUSA_STATUS_1[3:3]: WDT_FirstMSG_done = 0x1.**

The Q&A WDT operation shall proceed as described in Figure 12.

[1] The RAA271005 updates the **Question/ TOKEN** via the **WDT_LFSR[7:0]** register. Consider only the upper nibble.
[2] The SoC writes the expected 32 bit **ANSWER** to the **WDT_KICK_REG[7:0]** register in 8-bit increments, and in the following order: **ANSWER-0, ANSWER-1, ANSWER-2, ANSWER-3 of Table 16.**
[3] The RAA271005 evaluates each **ANSWER** for correctness and timeliness.
The ANSWER shall be received by the PMIC within the set upper and lower WDT timing limits in order to be evaluated “in-time”.

[4] The WDT error count is stored in the **WDT_CFG3[7:0]** register. The counter is incremented as:

- An incorrect but In-time answer increments the error counter +1
- A correct but Not-in-time answer increments the error counter +1
- A correct & In-time answer increments the counter -1
- An Early & Incorrect answer increments the counter +2
- A Late & Incorrect answer increments the counter +1
- A Too Late answer increments the error counter +2, and reset the WDT

Definitions:

LLTIME (Lower Limit Time) = LLCNT * LLTICK
ULTIME (Upper Limit Time) = ULCNT * ULTICK
TooLateTime = (2*ULCNT + 1) * ULTICK
In-time = LLTIME ~ ULTIME
Not-in-time (Early) = 0 ~ LLTIME

Not-in-time (Late) = ULTIME ~ TooLateTime

Too Late = TooLateTime

5] If all 4 bytes of the **ANSWER** are correct, the PMIC updates the **Question/ TOKEN**

Table 16 16 Question Q&A WDT Operands & Expected Answers

Question (TOKEN) in WDT_LFSR Register	**WD ANSWER (To be written into WDT_KICK_REG Register)**			
TOKEN	**Answer-3**	**Answer-2**	**Answer-1**	**Answer-0**
0h	FFh	0Fh	F0h	00h
1h	B0h	40h	BFh	4Fh
2h	E9h	19h	E6h	16h
3h	A6h	56h	A9h	59h
4h	75h	85h	7Ah	8Ah
5h	3Ah	CAh	35h	C5h
6h	63h	93h	6Ch	9Ch
7h	2Ch	DCh	23h	D3h
8h	D2h	22h	DDh	2Dh
9h	9Dh	6Dh	92h	62h
Ah	C4h	34h	CBh	3Bh
Bh	8Bh	7Bh	84h	74h
Ch	58h	A8h	57h	A7h
Dh	17h	E7h	18h	E8h
Eh	4Eh	BEh	41h	B1h
Fh	01h	F1h	0Eh	FEh

[6] The Fault Detection Logic in the PMIC will compare the accumulated WDT error count (**WDT_CFG3[7:0]: WDT_WWDT_ACC**) against the maximum WDT error threshold (**WDT_CFG2[7:6]: WDT_WWDT_ACC_TH)**

[7] If the maximum WDT error threshold is exceeded, the **FLT_RECORD_B[5:5]: FLT_WDT** register in the PMIC will be set and the Safety Control Logic will control the PMIC to RESET state.

Table 17 16 Question Q&A WDT Recommended Usage Summary

Register Address	Register	Register field/bit	Recommended Usage
0x107	WDT_CFG0[0]	WDT_WWDT_EN = 1	Confirm WDT is enabled
0x108	WDT_CFG0[1]	WDT_WWDT_ADV_16Q = 1	Confirm 16 Q&A Mode selected
0X010	FUSA_STATUS_1[3]	FUSA_FirstMSG_done = 1	Confirm First WDT message is received For the 16 QA mode, this is *"0xA7"*.
0x12C	FLT_MASK_B[4]	FLT_MaskWDT = 0	Confirm WDT errors are not masked
0x109	WDT_CFG2[5:0]	WDT_ULTICK, WDT_LLTICK	Confirm WDT window timing unit setting
0x108	WDT_CFG1[7:0]	WDT_ULCNT WDT_LLCNT	Confirm WDT window time limit setting
0x01A	FLT_RECORD_B[5]	FLT_WDT = 0	Confirm WDT is fault-free at each FTTI
0x10A	WDT_CFG3[7:0]	WDT_WWDT_ACC	Check WDT error counter status
0x096	WDT_LFSR[7:4]		Read WDT questions
0x095	WDT_KICK_REG[7:0]		Write answers to WDT questions

4 Question Q&A Mode Configuration & Operation

The same procedures apply as in 16 Q&A Mode.
The only difference is, the number of questions is reduced to 4.

Table 18 4 Question Q&A WDT operands & expected answers

Question (TOKEN) in WDT_LFSR Register		WD ANSWER (To be written into WDT_KICK_REG Register: D5 – D0 bits)
Do7	Do6	
0	0	Copy Do5-Do0 bits for D5-D0
0	1	Shift Do5-Do0 bits 1bit to the left for D5-D0
1	0	Shift Do5-Do0 bits 1bit to the right for D5-D0
1	1	Invert Do5-Do0 bits for D5-D0

Table 18a 4 QA WDT example

Question in WDT_LFSR Register	ANSWER
01111111	01 | 111110
10111111	10 | 011111
11000000	11 | 111111
00000000	00 | 000000

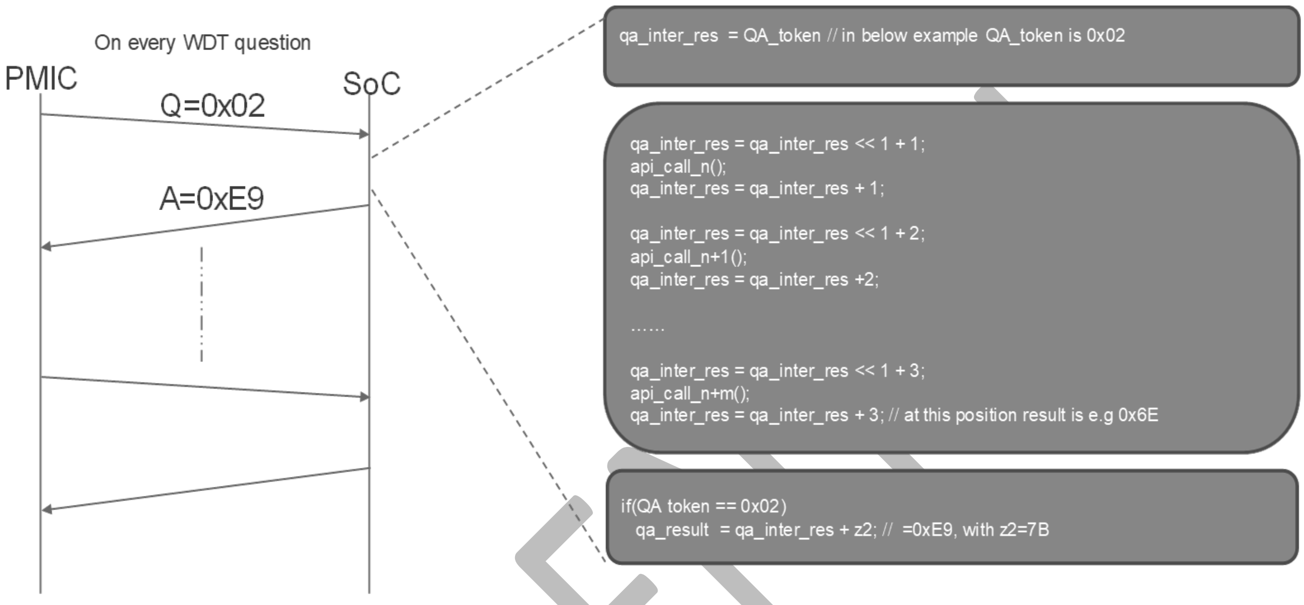
Basic Windowed Kick Mode

In this mode, the SoC continuously performs single byte write of value 0x2A to the WDT_KICK_REG register within the configured time window.

A wrong data in the WDT_KICK_REG register, or an I2C/SPI transaction command not received on time, is immediately considered a fault, and the PMIC transitions to RESET state.

Using the Q&A WDT for Program Flow Monitoring

Figure 11 Example Q&A WDT Flow for Program Flow Monitoring



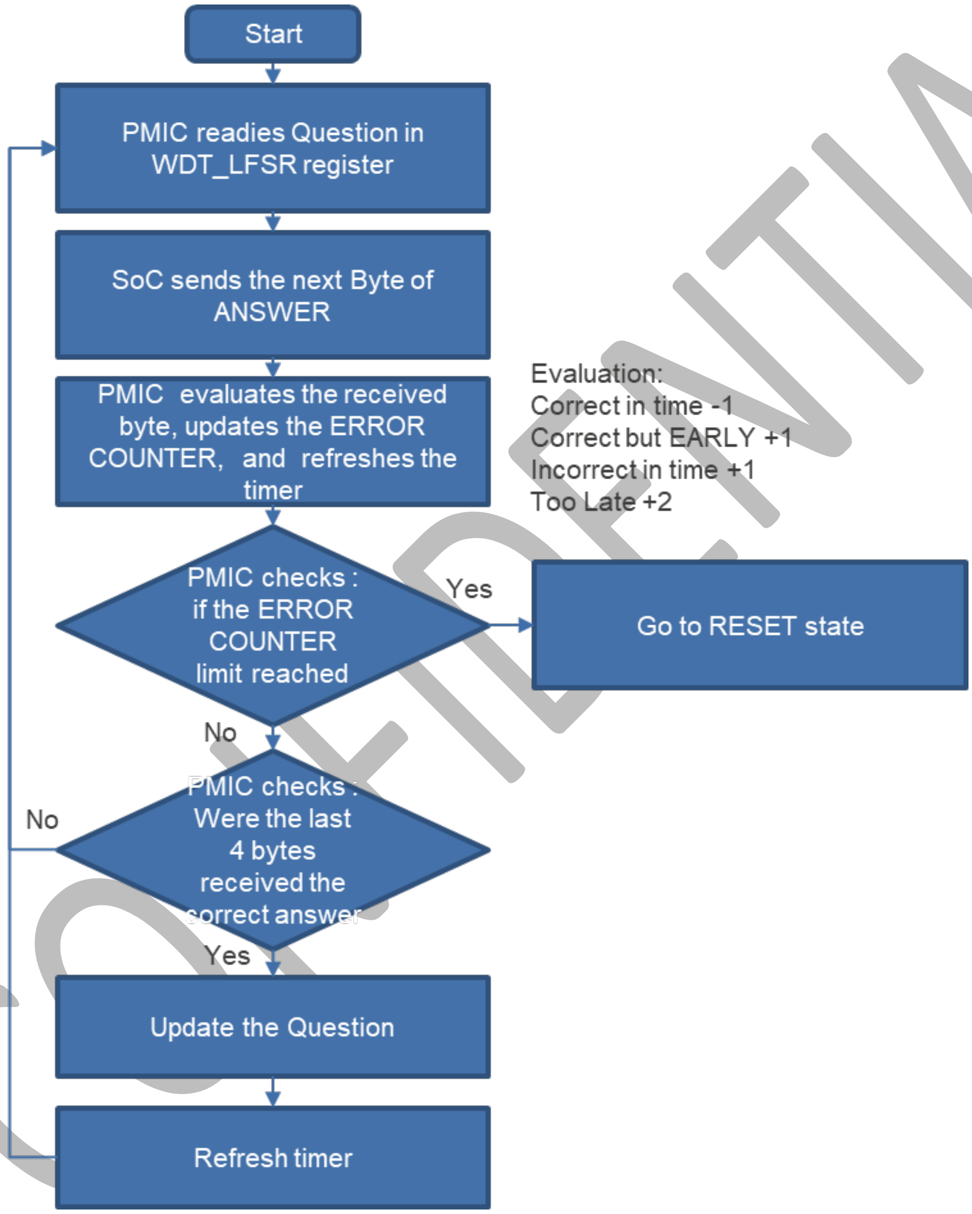
Note: The constant z2 has to be computed at compile time for each of 16 token. Above procedures is simplified and should just show the principle. Wrong order and omission of API call can be detected by the procedure. Its user's responsibility to select the appropriate formula and enough monitoring points for his application.

When used as a Program Flow Monitor (PFM), the maximum WDT error threshold shall be **WDT_CFG2[7:6]: WDT_WWDT_ACC_TH = 1.**

d) Fault Control and Operation

Figure 12 illustrates the Q&A WDT operation followed by both 16 Q&A and 4Q&A WDT modes.

Figure 12 Q&A WDT Operation



When WDT error condition is detected, the Safety Control Logic controls the following within FTTI:

- The device transitions to ‘RESET’ state.
- The **FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt** register is updated with the incremented SoC Error Count.
- PRESET#=L is applied and PRESETOUT0#=L loop back is checked
- If RESET loopback completes and passes, RAA271005 moves to SoC Activation State
- If RESET loopback fails, RAA271005 moves to ERROR state
- During this state, all monitoring are still applied per Table 36.

e) Test Concept

The WDT is tested at startup via LBIST. See more details on LBIST in Section 7.12

A system-level test can be added during SoC Activation Sequence to check that the WDT is active and able to detect an error, before normal operation, assuming there is sufficient time to do so:

1. During SoC Activation state, WDT errors detected are masked/disabled
2. The system integrator should ensure that WDT Check is allocated within the set SoC Activation state duration (**FUSA_TIMER_2[7:5]: TIMEOUT_SOCACTIVA_ST**).
3. To check WDT, SoC should write the correct ANSWER-0, incorrect ANSWER-1, incorrect ANSWER-2, and correct ANSWER-3 to the **WDT_KICK_REG[7:0]**
4. SoC should verify that the **WDT_CFG3[7:0]: WDT_WWDT_ACC** register is incremented
5. SoC should verify that the same question will remain in the **WDT_LFSR[7:0]** register
6. SoC should send the correct ANSWER-0, ANSWER-1, ANSWER-2, ANSWER-3 to clear the **WDT_CFG3[7:0]: WDT_WWDT_ACC** register, exit SoC Activation State & proceed to Active state
7. All WDT error reaction are un-masked in Active state.

7.6 SM13: OTP CRC & OTP Redundancy

a) Overview

The RAA271005 configuration is stored in One Time Programmable (OTP) memory.
Each configuration data and its inverted version are stored in separate memory cells and used for data confirmation.
A second copy of the original & inverted configuration data is stored in redundant memory.
The memory macro handles the differential redundancy checks internally and forms the final data. A CRC-16 polynomial is used to do a CRC on the formed OTP data, and the incoming data byte is compared with the written data byte during OTP download.
A fault is reported if a CRC error or a register map double check error is detected.

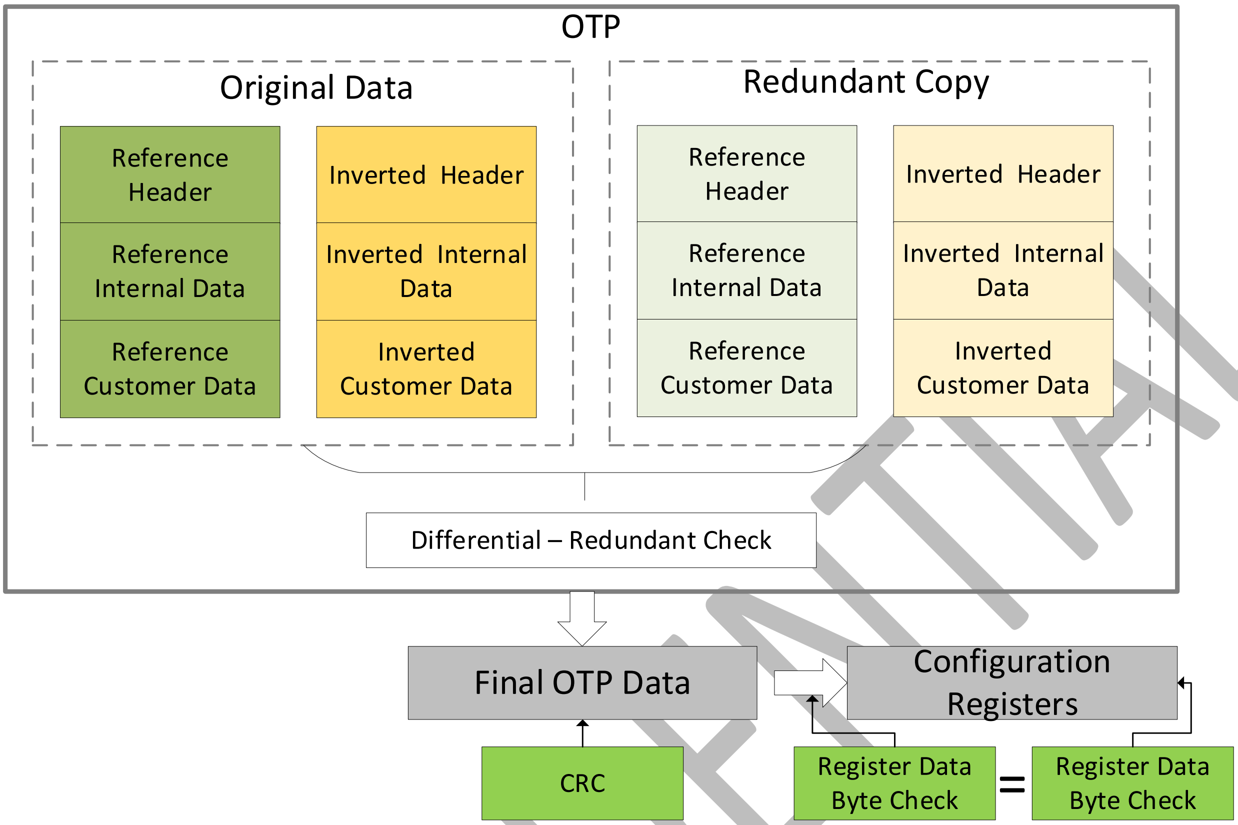
Detection Time Interval (DTI) of the OTP CRC is 62.5ns.

b) Hardware Description

The RAA271005 configuration data is differential redundant. The information is stored in 4 memory locations as shown in Figure 13. The same architecture is used in both Regulation & Protection units.

The OTP CRC 16 Polynomial used is: $x^{16} + x^{15} + x^2 + 1$
This polynomial provides a Hamming Distance of 4 for data lengths of up to 32K bits. The PMIC OTP data length is 2048 bits.

Figure 13 OTP architecture



c) Recommended Usage

The safety mechanism is active at startup.
The PMIC does not proceed to Power Up Sequence if an OTP CRC error is detected during Self Diagnosis state.

Important Configuration Registers shall be protected by LOCKOUT bits.
The System Integrator shall activate the LOCKOUT protection of these registers by using two registers; **I_LOCK_OUT_CFG[7:0]** on Protection unit and **LOCK_OUT_CFG[7:0]** on the Regulation unit. Each bit of these registers are used as write block for different sections of each register map. If the associated bit is written “1”, WRITE to those specific groups will be blocked. Moreover, these two registers are a special type, such that once their bits are written “1”, they cannot be changed as long as the device has power. Both registers have OTP fields.

Recommended register settings:
I_LOCK_OUT_CFG = 0xFF
LOCK_OUT_CFG = 0xFF

There are two options to set these registers:

1. Set via OTP.
   The system integrator identifies group of registers that have to be LOCKED and sets the corresponding bits in the **I_LOCK_OUT_CFG** and **LOCK_OUT_CFG** registers in OTP. The PMIC will automatically disable the write to those registers as OTP content gets downloaded at every power up.
2. Program LOCKOUT before the end of SoC Activation Sequence.
   Before the end of SoC Activation Sequence, the SOC writes to the **I_LOCK_OUT_CFG** and **LOCK_OUT_CFG** registers as desired and during that run cycle, write to the associated registers will be blocked.

Table 19 Protection Block I_LOCK_OUT_CFG Register

BIT	NAME	R/W	DESCRIPTION
7:0	I_LOCK_OUT	RW	Bit - lock 0- Password access: IO_KEY* registers are locked for WRITE 1- FUSA general setup: Registers IO_FUNC_CFG, FLT_MASK_B, FUSA_CTRL_<1,A,B,C,MTE>, FUSA_TIMER<1,4>, IO_MODECTRL, SOC_PIN_DATA<1-6> are locked for WRITE 2- Soft reset control: IO_SOFTRESET register is locked for WRITE 3- Vmon thresholds, threshold delays, ADC setup: Registers with the ff. addresses are blocked for WRITE: 0x136 – 0x1DA, ADCMON_CFG, ADCMON_EXT_CFG 4- Watchdog Settings: Registers WDT_CFG<0,1,2,3,4>, WDT_LFSR are locked for WRITE 5- I/O setup: Registers IO_PINMODE, IO_I2C_SPICFG, IO_I2CADDR, addresses 0x1DB-0x1EA, 0x256- 0x25F are blocked for WRITE 6- FLT masking for Vmon: Register addresses: ADCMON_MASK* (addresses 0x12D-0x133) are blocked for WRITE 7- SoC Activation: Registers: FUSA_CHK_CVM*, FUSA_CTRL_CVM*, FUSA_CTRL_<D,E>, FUSA_TIMER_2, FUSA_TIMER_3 are blocked for WRITE

d) Fault Control and Operation

When an OTP CRC error is detected, the Fault Control Logic updates the corresponding fault status registers:

Table 20 OTP CRC Fault Table

REGISTER	DESCRIPTION
OTP_RWADDR[7:7]: OTP_CRC_FAULT	Indicates an error in the OTP CRC in Protection unit
OTP_RWADDR_REGU[7:7]: OTP_CRC_FAULT*	Indicates an error in the OTP CRC in Regulation unit
OTP_FLT_RECORD[5:5]: OTP_FLT_CRC_FailPage0	Indicates CRC error on Page0/1 registers in Protection unit
FLT_RECORD_OTP[2:2]: FLT_OTPCRCPAGE01*	Indicates CRC error on Page0/1 registers in Regulation unit
OTP_FLT_RECORD[4:4]: OTP_FLT_CRC_FailPage2	Indicates CRC error on Page2 registers in Protection unit
FLT_RECORD_OTP[3:3]: FLT_OTPCRCPAGE2*	Indicates CRC error on Page2 registers in Regulation unit

**Regulation block register*

An OTP fault results in a SelfDiagnosis fail.

When SelfDiagnosis fails due to an OTP fault, the Safety Control Logic controls the following:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

The Regulation register map will be reset when an OTP CRC error is detected.
The Protection register map will be reset when the following error conditions are detected: OTP CRC error, Reg map double check error, OTP download timeout, and OTP initialization failure.

If a Regulation OTP failure occurs, the system can force an OTP download re-try by creating a falling edge on the ADC3 pin. This pin is otherwise ignored for this retry feature when the OTP download is error-free.

e) Test Concept

OTP CRC & OTP Redundant memory are tested at startup via LBIST. See more details on LBIST in Section 7.12

7.7 SM16: SPI CRC

a) Overview

Each WRITE and READ transaction between the SoC & the PMIC via SPI is checked with a CRC-8 polynomial.
The $x^8 + x^2 + x^1 + 1$ polynomial provides a hamming distance (HD) of 3 for up to 247 bits. The PMIC data length is 32 bits (conservative case). So, HD=3 means that the PMIC cannot detect 3 or more bit errors but can actually detect any 2 incorrect bits within the serial transaction itself.
CRC based errors disable all writes of the packet.
Packets must follow the SPI protocol or the hardware behavior is un-defined and must be reset.

Detection Time Interval (DTI) of SPI CRC is 1 SPI Clock Cycle.

b) Hardware Description

Not relevant.

c) Recommended Usage

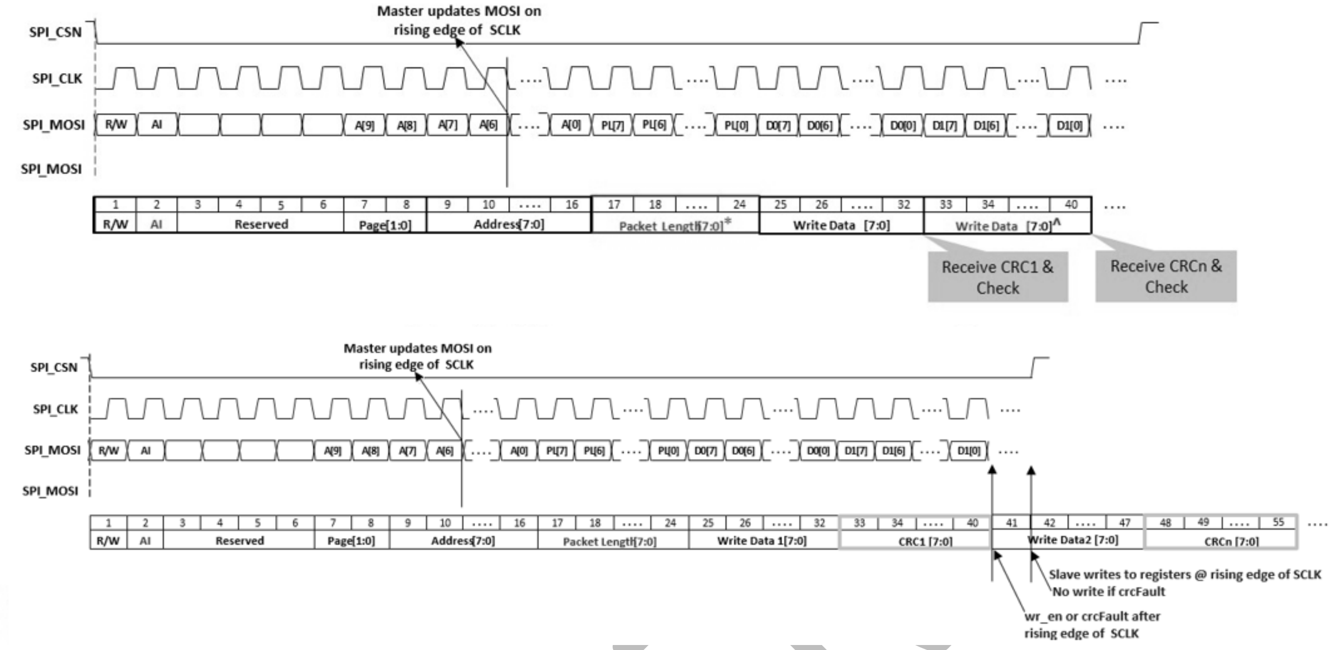
The safety mechanism shall be active at startup, before performing the SoC Activation Sequence, and continue to remain active. The system integrator shall ensure this by setting Protection unit register **IO_FUNC_CFG [0:0]: REGCRC_EN = 0x1**, and Regulation unit register **IO_I2CCFG[7]: IO_USE_CRC = 0x1** in OTP.

The system integrator shall ensure that CRC fault masking is set via **FLT_MASK_B[3:3]: FLT_MaskRegCRC** register = 0x1 in OTP if going to RESET state at each CRC fault is undesirable for the system.

For WRITE transactions, the SoC shall send a CRC byte after each data write. The PMIC checks the CRC at the end of the data write transaction and writes the data to the register if no CRC error is detected. See Figure 14.

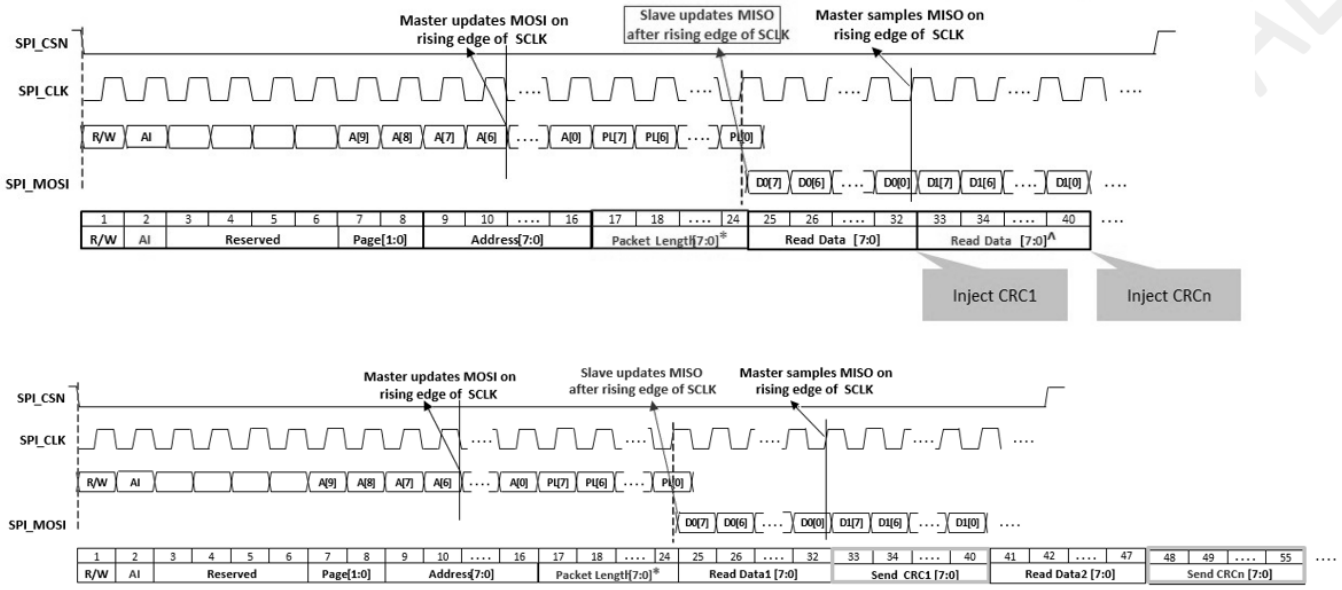
Note: Only 1-Byte transactions are supported by RAA271005.

Figure 14 SPI CRC Insertions For WRITE Transactions



For READ transactions, the PMIC calculates and injects a CRC byte after the data to be read, for SoC to check. See Figure 15.

Figure 15 SPI CRC Insertions For READ Transactions



d) Fault Control and Operation

When a SPI CRC fault is detected from a communication error with the Protection block, all writes to the packet is disabled, the SPI CRC fault status register **FLT_RECORD_B[4:4]: FLT_RegCRC** is updated, and the Interrupt pin (IRQ) is asserted.
The SoC shall read **FLT_RECORD_B[4:4]: FLT_RegCRC** register to clear the fault and try the transaction again.

If the SPI CRC fault is from a communication error with the Regulation block, the **FLT_RECORD_IF[1]: FLT_SPI** is updated, and the Interrupt pin (IRQ) is asserted. This fault will also set **FLT_RECORD_B[2:2]: FLT_ReguOT** register in the Protection block**.**
The SoC shall write 0x0 to **FLT_RECORD_IF[1]: FLT_SPI** register in Regulation to clear the fault and also read Protection **FLT_RECORD_B[2:2]: FLT_ReguOT** register to clear the fault and try the transaction again.

The SoC shall monitor the frequency of the occurrence of the CRC fault, or use the CRC fault status information together with the status of the SPI Message Counter (SM17) to determine the severity of the SPI transaction errors, and issue an error signal to the RAA271005, to trigger a RESET

When the RAA271005 receives an SoC Error signal, the Safety Control Logic controls the following within FTTI:

- The device transitions to 'RESET' state.
- The **FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt** register is updated with the incremented SoC Error Count.
- PRESET#=L is applied and PRESETOUT0#=L loop back is checked
- If RESET loopback completes and passes, RAA271005 moves to SoC Activation State
- If RESET loopback fails, RAA271005 moves to ERROR state
- During this state, all monitoring are still applied per Table 36.

e) Test Concept

SPI CRC is tested at startup via LBIST. See more details on LBIST in Section 7.12

7.8 SM17: SPI Message Counter

a) Overview

The SPI Message Counter increments whenever a valid WRITE transaction to Protection block takes place via the SPI communication interface. This safety mechanism is for detecting the following communication failure modes: repetition, loss, and insertion of data.
The register is reset once a data overflow event occurs.
The register resets during SelfDiagnosis.

Detection Time Interval (DTI) of SPI Message Counter is 62.5ns.

b) Hardware Description

Not relevant.

c) Recommended Usage

The safety mechanism is active at startup and continue to remain active.

1] The SoC shall monitor the **IO_HOST_MSGCNT[7:0]: HOST_MSGCNT** register, to compare the number of valid WRITE transactions received by the RAA271005 with the number of WRITE transactions attempted by the SoC. This is only applicable to WRITEs to the Protection Block.

2] The SoC shall determine the severity of the discrepancy and issue an error signal to the PMIC.

3] The SoC shall use the Message Counter status, together with the SPI CRC fault status information on **FLT_RECORD_B[4:4]: FLT_RegCRC** Protection block register and the IRQ signal status, to make a more informed judgement.

4] At every recovery from RESET state, the SoC shall read the PMIC message counter value at the end of SoC Activation and apply it as an SoC message count offset value entering into ACTIVE state.

5] The SoC shall factor in the offset in [4] with the maximum message counter value of 255, to anticipate overflow and reset the SoC counter.

d) Fault Control and Operation

The safety mechanism is an additional means for the system to determine an error in the communication between the SoC & the RAA271005 and perform system level measures to control it.

e) Test Concept

SPI Message Counter is tested at startup via LBIST. See more details on LBIST in Section 7.12

7.9 SM18: I2C CRC

a) Overview

Each WRITE and READ transaction between the SoC & the PMIC via I2C is checked with a CRC-8 polynomial.
The $x^8 + x^2 + x^1 + 1$ polynomial provides a hamming distance (HD) of 3 for up to 247 bits. The PMIC data length is 32 bits (conservative case). So, HD=3 means that the PMIC cannot detect 3 or more bit errors but can actually detect any 2 incorrect bits within the serial transaction itself.
CRC based errors disable all writes of the packet.
Packets must follow the I2C protocol or the hardware behavior is un-defined and must be reset.

Detection Time Interval (DTI) of I2C CRC is 5us.

b) Hardware Description

Not relevant.

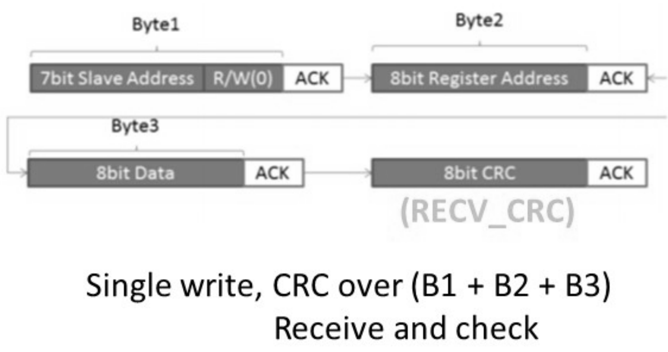
c) Recommended Usage

The safety mechanism shall be active at startup, before performing the SoC Activation Sequence, and continue to remain active. The system integrator shall ensure this by setting Protection unit register **IO_FUNC_CFG [0:0]: REGCRC_EN = 0x1**, and Regulation unit register **IO_I2CCFG[7]: IO_USE_CRC = 0x1** in OTP.

The system integrator shall ensure that CRC fault masking is set via **FLT_MASK_B[3:3]: FLT_MaskRegCRC** register = 0x1 in OTP if going to RESET state at each CRC fault is undesirable for the system.

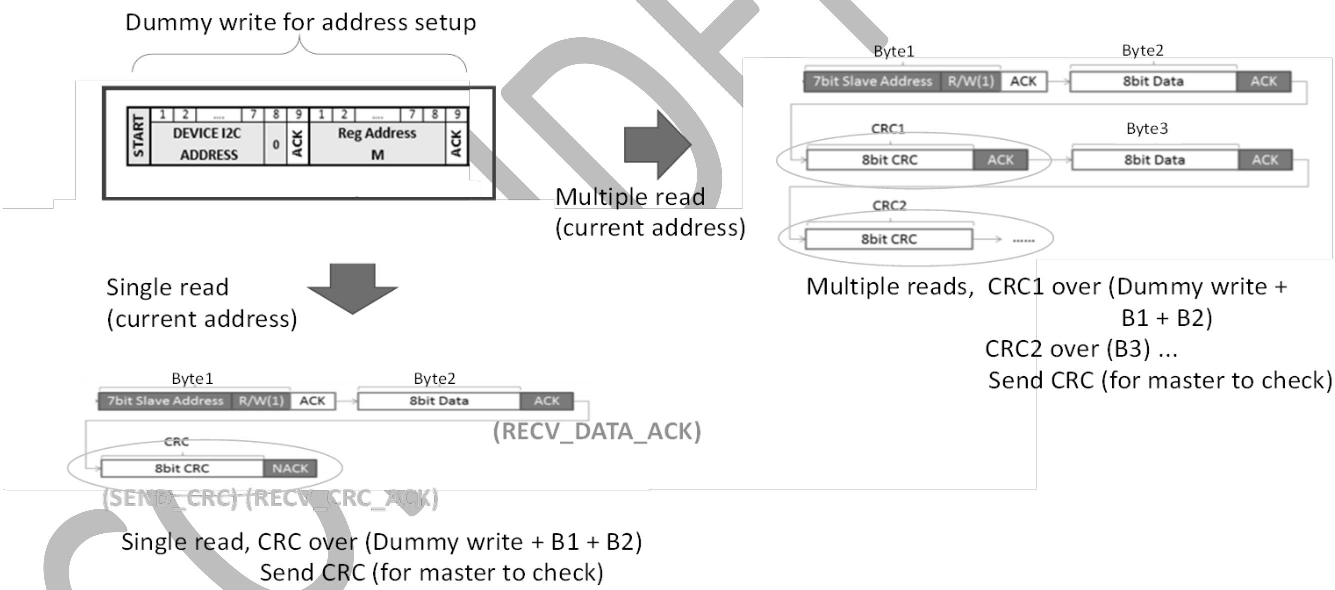
For WRITE transactions, the SoC shall send a CRC byte after each data write. The PMIC checks the CRC at the end of the data write transaction and writes the data to the register if no CRC error is detected. See Figure 17.

Figure 16 I2C CRC Insertions For WRITE Transactions



For READ transactions, the PMIC calculates and injects a CRC byte after the data to be read, for the SoC to check. See Figure 17.

Figure 17 I2C CRC Insertions For READ Transactions



Note: Only 1-Byte transactions are supported by RAA271005.

d) Fault Control and Operation

When an I2C CRC fault is detected, all writes to the packet is disabled, the I2C CRC fault status register **FLT_RECORD_B[4:4]: FLT_RegCRC** is updated, and the Interrupt pin (IRQ) is asserted.
The SoC shall read **FLT_RECORD_B[4:4]: FLT_RegCRC** register to clear the fault and try the transaction again.

If the I2C CRC fault is from a communication error with the Regulation block, the **FLT_RECORD_IF[0:0]:FLT_I2C** is set, and the Interrupt pin (IRQ) is asserted.This fault will also set **FLT_RECORD_B[2:2]: FLT_ReguOT** register in the Protection block**.**
The SoC shall write 0x0 to **FLT_RECORD_IF[0:0]:FLT_I2C** register to clear the fault in Regulation. The SoC shall read **FLT_RECORD_B[2:2]: FLT_ReguOT** register in Protection to clear the fault and try the transaction again.

The SoC shall monitor the frequency of the occurrence of the CRC fault, or use the CRC fault status information together with the status of the I2C Message Counter (SM17) to determine the severity of the SPI transaction errors, and issue an error signal to the RAA271005, to trigger a RESET.

When the RAA271005 receives an SoC Error signal, the Safety Control Logic controls the following within FTTI:

- The device transitions to 'RESET' state.
- The **FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt** register is updated with the incremented SoC Error Count.
- PRESET#=L is applied and PRESETOUT0#=L loop back is checked
- If RESET loopback completes and passes, RAA271005 moves to SoC Activation State
- If RESET loopback fails, RAA271005 moves to ERROR state
- During this state, all monitoring are still applied per Table 36.

e) Test Concept

I2C CRC is tested at startup via LBIST. See more details on LBIST in Section 7.12

The following sections illustrate how to check for the I2C CRC functionality:

Figure 18 Example of Regulation Unit CRC Test

SM18: Normal test

1. Regulation Unit access check (CRC valid)

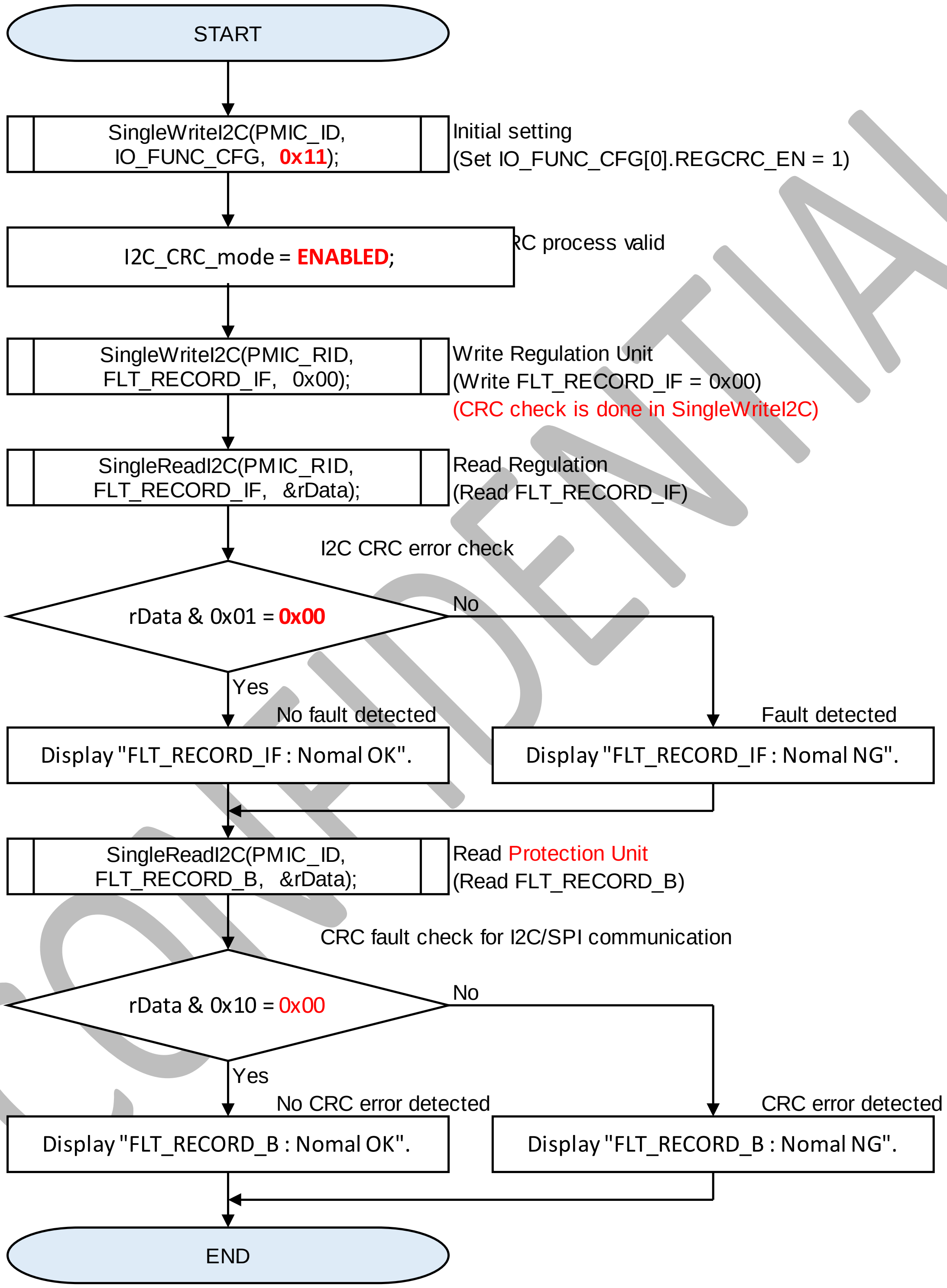


Figure 19 Example of Regulation Unit Invalid CRC Test

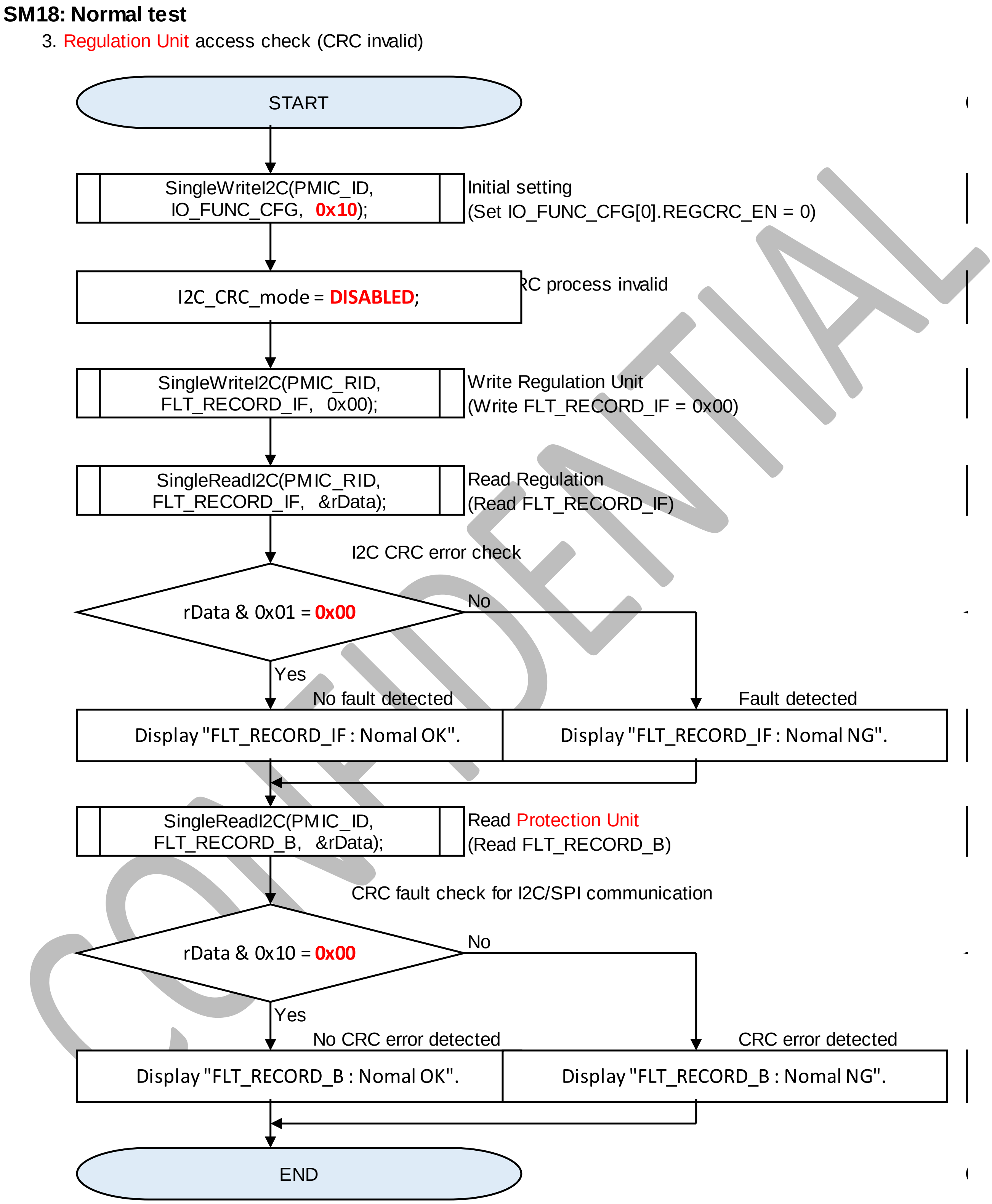


Figure 20 Example of Regulation Unit CRC Fault Test

SM18: Abnormal Test

1. Regulation Unit access check (CRC valid)

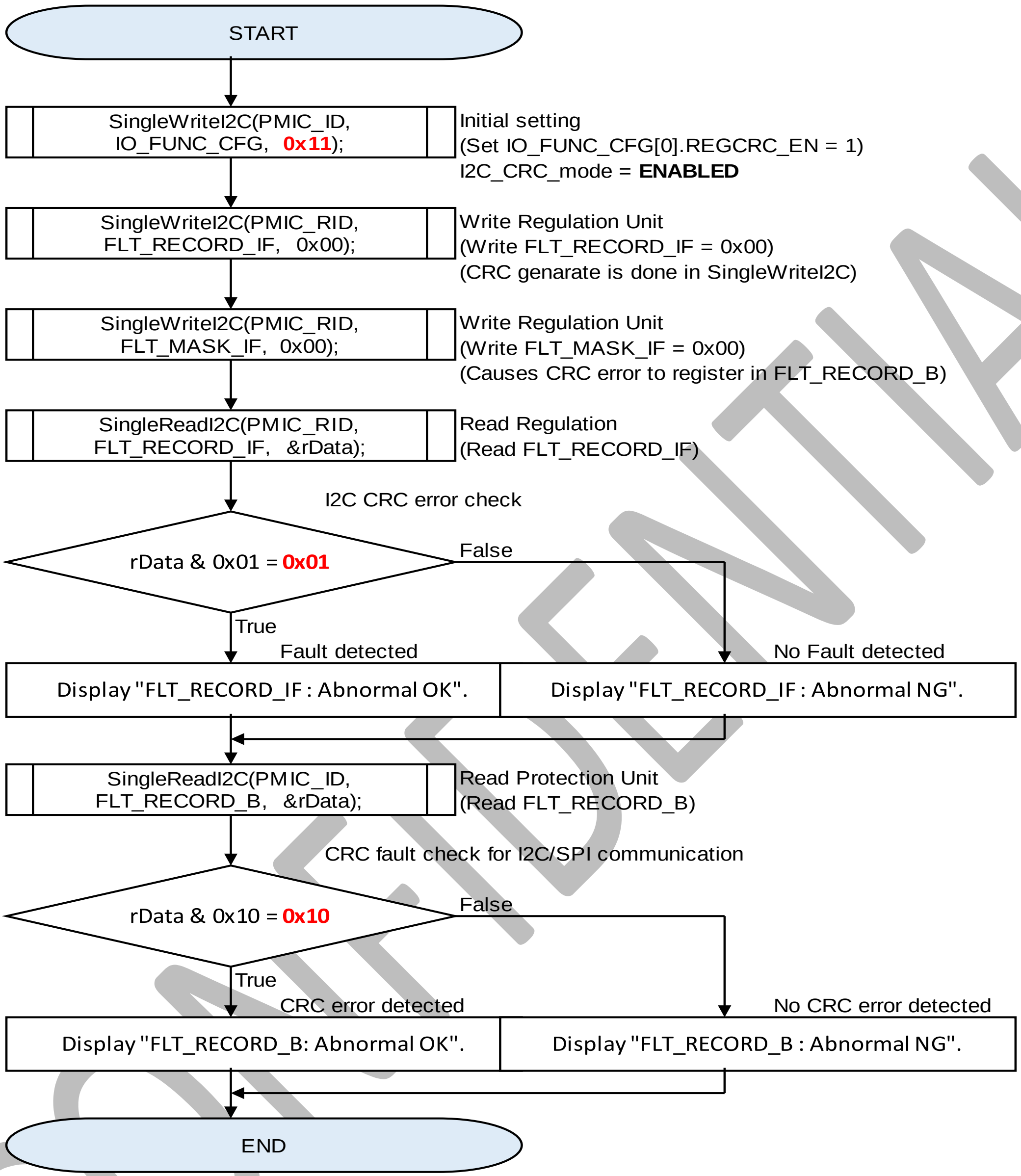


Figure 21 Example of Protection Unit CRC Test

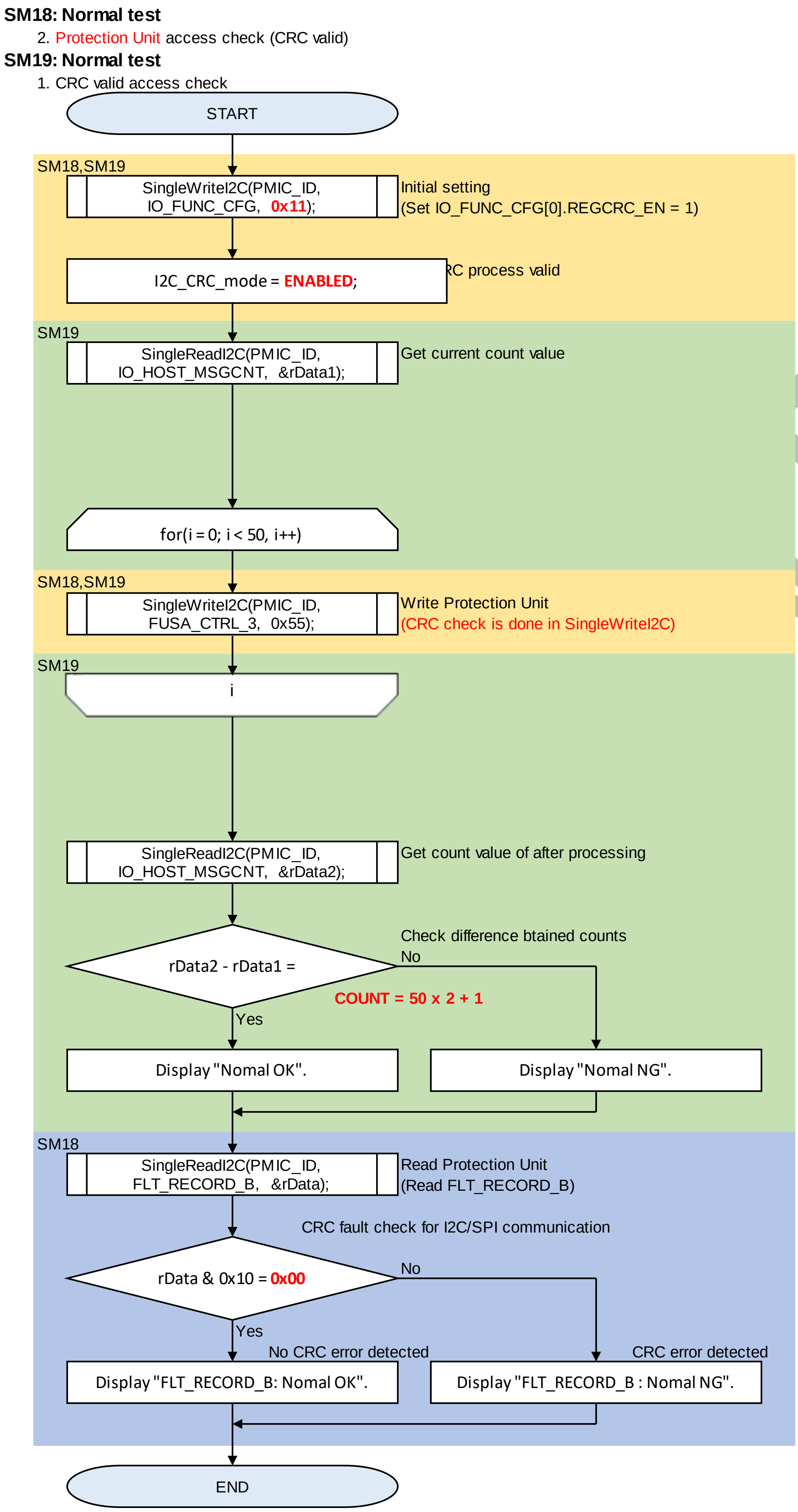


Figure 22 Example of Protection Unit Invalid CRC Test

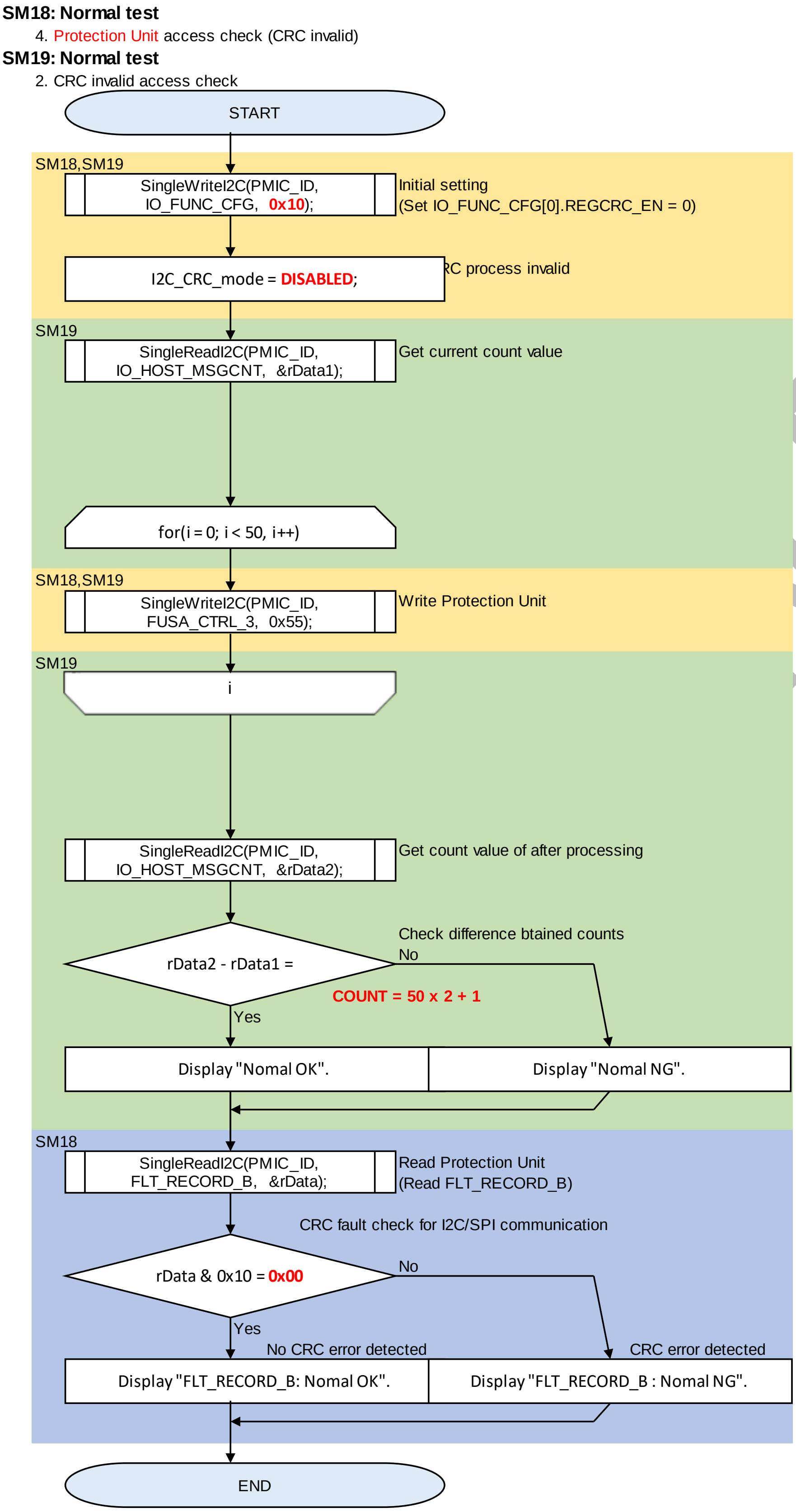


Figure 23 Example of Protection Unit CRC Fault Test

SM18: Abnormal Test

2. Protection Unit access check (CRC valid)

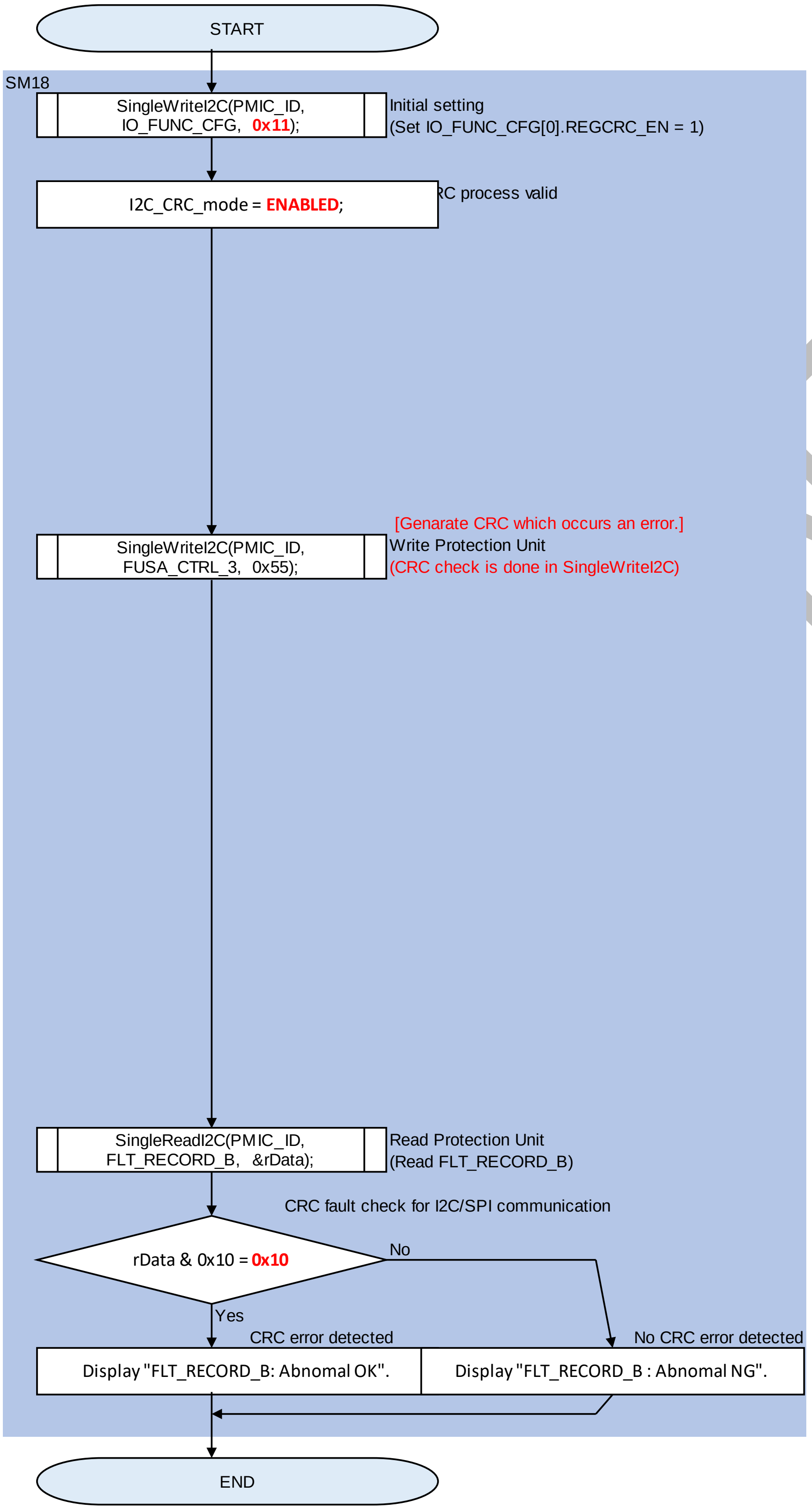
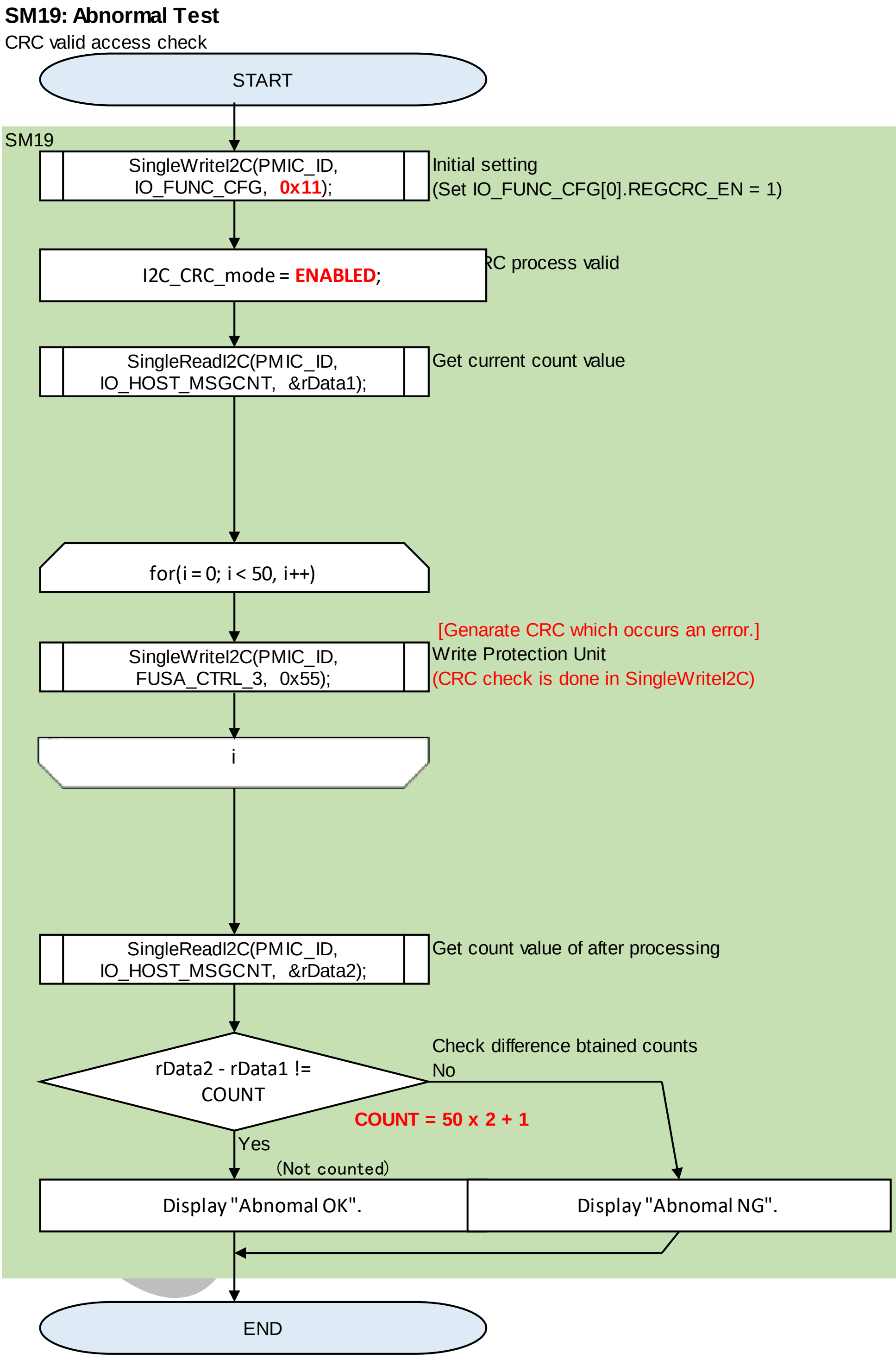


Figure 24 Example of Protection Unit Message Counter Test



7.10 SM19: I2C Message Counter

a) Overview

The I2C Message Counter increments whenever a valid WRITE transaction to the Protection Block takes place via the I2C communication interface. This safety mechanism is for detecting the following communication failure modes: repetition, loss, insertion and wrong sequence of data.
In case of READ transactions, since a READ involves 1 or more WRITE transactions (page and address writes), the message counter will increment accordingly.
The register is reset once a data overflow event occurs.
The register resets during SelfDiagnosis.

Detection Time Interval (DTI) of I2C Message Counter is 62.5ns.

b) Hardware Description

Not relevant.

c) Recommended Usage

The safety mechanism is active at startup and continue to remain active.

1] The SoC shall monitor the **IO_HOST_MSGCNT[7:0]: HOST_MSGCNT** register, to compare the number of valid WRITE transactions received by the RAA271005 with the number of WRITE transactions attempted by the SoC. This is only applicable to WRITEs to the Protection Block.

2] The SoC shall determine the severity of the discrepancy and issue an ERROR signal to the PMIC.

3] The SoC shall use the Message Counter status, together with the I2C CRC fault status information on **FLT_RECORD_B[4:4]: FLT_RegCRC** Protection block register, and the IRQ signal status, to make a more informed judgement.

4] At every recovery from RESET state, the SoC shall read the PMIC message counter value at the end of SoC Activation and apply it as an SoC message count offset value entering into ACTIVE state.

5] The SoC shall factor in the offset in [4] with the maximum message counter value of 255, to anticipate overflow and reset the SoC counter.

d) Fault Control and Operation

The safety mechanism is an additional means for the system to determine an error in the communication between the SoC & the RAA271005 and perform system level measures to control it.

e) Test Concept

I2C Message Counter is tested at startup via LBIST. See more details on LBIST in Section 7.12

7.11 SM22: ADC BIST

a) Overview

ADC BIST is done at startup automatically by the RAA271005, as part of the self diagnosis state.
It consists of testing the ADC conversion, ADC Mux functionality, and Fault Control Logic operation when a fault is detected.

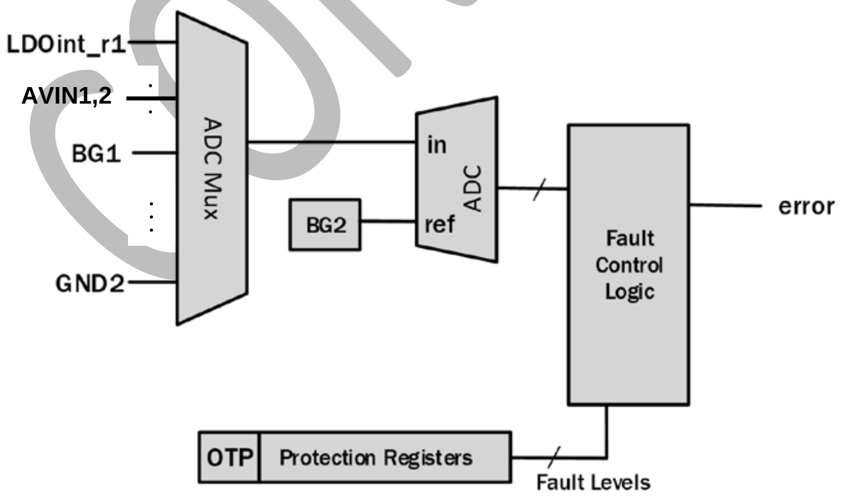
ADC conversion is tested by comparing the measured values of eleven on-chip voltages (Internal LDO voltage, BG1 and GND) and 2 external voltages (AVIN1 and AVIN2) against thresholds. The full range of the ADC multiplexer input signals are tested by selecting these eleven on-chip voltages, which cover all of the Mux selection bits' conditions. The Fault Control Logic's operation is tested by injecting an OV condition on all ADC channels and confirming an OV fault is detected on all of them. **FLT_RECORD_B[1:1]: FLT_FaultDetectBist** register is set if an OV event is not detected on any one channel. The actual ADC offset value is used while performing Fault Detect BIST to accurately confirm that the ADC's configuration will be able to detect a fault condition. See Table 21 for recommendations.

ADC BIST completes within 3ms at startup.
The monitoring of these signals continue throughout normal operation.

Detection Time Interval (DTI) of ADC BIST is once per driving cycle.

b) Hardware Description

Figure 25 ADC BIST Architecture



c) Recommended Usage

The safety mechanism is active at startup, and throughout normal operation.
The system integrator shall ensure that ADC BIST is not bypassed during PMIC Self-Diagnosis state by setting **FUSA_CTRL_A[1:1]: ADC_BIST_BYPASS=0x0** in OTP.
The PMIC does not proceed to Power Up state if ADC BIST fails.

OV limits shall not be set above the maximum value indicated in Table 21 to ensure that Fault Detect BIST do not generate a false FAIL during Self Diagnosis.
The recommended maximum OV limit is derived from VSCALE -1 codes, where VSCALE is the maximum code in decimal. These are programmed in **I_ADC_VSCALEPGA*MSB/LSB** registers for each PGA gain.

Table 21 Recommended Max OV limit setting depending on PGA Gain Setting

PGA Code	PGA Gain	Register	Max OV Limit (V)
0	0.125	**I_ADC_VSCALEPGA0MSB/LSB**	7.478
1	0.5	**I_ADC_VSCALEPGA1MSB/LSB**	1.868
2	2	**I_ADC_VSCALEPGA2MSB/LSB**	0.4667
3	8	**I_ADC_VSCALEPGA3MSB/LSB**	0.1162
4	0.333	**I_ADC_VSCALEPGA4MSB/LSB**	2.806
5	0.8	**I_ADC_VSCALEPGA5MSB/LSB**	1.1682
6	1.25	**I_ADC_VSCALEPGA6MSB/LSB**	0.7167
7	1.4	**I_ADC_VSCALEPGA7MSB/LSB**	0.6652

For PVIN, the recommendations are:

PGA Code	PGA Gain	Register	Max OV Limit (V)
6	1.25	**I_ADC_VSCALEPGA6MSB/LSB**	5.5*

** Internally calibrated for PVIN channels.*

d) Fault Control and Operation

ADC BIST reports the fault status of each of the voltages measured during the test, as well as the Fault Detection Logic test in the following registers:

Table 22 ADC BIST Fault Registers

REGISTER	DESCRIPTION
FLT_RECORD_IntLDOs[6:6]: FaultStatus_IntLDOProt_0	Protection Int LDO(1) fault status
FLT_RECORD_IntLDOs[7:7]: FaultStatus_IntLDOProt_1	Protection Int LDO(2) fault status
FLT_RECORD_IntLDOs[5:0]: FaultStatus_IntLDO<Regu_0..5>	Regulators' Int LDO fault status
FLT_RECORD_GND_AVIN[1:1]: FaultStatus_PGND_Regu	Regulation GND fault status
FLT_RECORD_GND_AVIN[0:0]: FaultStatus_Offset	Protection GND fault status
FLT_RECORD_BG_TEMP[0:0]: FaultStatus_BG_REGU	Regulation Bandgap status
FLT_RECORD_GND_AVIN[3:3]: FaultStatus_AVIN2_Prot	Protection AVIN2 supply fault status
FLT_RECORD_GND_AVIN[2:2]: FaultStatus_AVIN1_Regu	Regulation AVIN1 supply fault status
FLT_RECORD_A[3:3]: FLT_FaultDetectBist	Fault Detection Logic BIST status

When an ADC BIST fail is detected, the the Safety Control Logic controls the following:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

e) Test Concept

ADC BIST fulfills the test concept of the ADC and its supporting circuits.
Testing the ADC BIST is beyond the multiple-points-of-failure scope of the RAA271005.

7.12 SM23: LBIST

a) Overview

LBIST is done at startup automatically by the RAA271005, as part of the Self Diagnosis state. It is executed after a successful OTP download. It tests the digital logic of the Digital Core of the Protection block. When LBIST fails, the PMIC re-runs LBIST every time Self Diagnosis State is visited upon recovery from ERROR state or LOCK state. The maximum LBIST re-tries is the same as the maximum ERROR counter threshold.
LBIST coverage is 92.84%.
LBIST takes ~2ms to complete at startup.

Detection Time Interval (DTI) of LBIST is once per driving cycle.

b) Hardware Description

Not relevant.

c) Recommended Usage

The safety mechanism shall be active at startup.
The system integrator shall ensure that LBIST execution is not disabled during PMIC Self-Diagnosis state by setting **FLT_MASK_B[0:0]: FLT_MaskLBIST =0x0** in OTP.
The PMIC does not proceed to Power Up state if LBIST fails.

d) Fault Control and Operation

During Cold Start & Return from Deep Stop:

When an LBIST fail is detected:

- The device transitions to ERROR state.
- The status **FLT_RECORD_B[0:0]: FLT_LBIST** register is updated.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count.
- Outputs are stopped according to set delay.
- PRESET# is de-asserted.
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

At recovery from ERROR state & LOCK state:

When LBIST fails, or when LBIST failed in a previous run:

- LBIST retry is done and **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** is incremented.
- The maximum retry is the same as the value set in the **PMIC_ERR_CNT_MAX** register.
- When the maximum retries allowed is reached, the PMIC goes to LOCK state.
- If LBIST passes during the retry, the PMIC proceeds to Power Up state.

e) Test Concept

LBIST fulfills the test concept of the Protection unit digital logic.

Testing LBIST is beyond the multiple-points-of-failure scope of the RAA271005.

7.13 SM24: Protection OT BIST

a) Overview

Protection OT BIST is done automatically by the RAA271005 at startup as part of ADC BIST during Self-Diagnosis State, and continuously throughout normal operation

OT BIST involves comparing TempSensor2 and TempSensor3 outputs and calculating the difference. The difference is compared against a difference threshold. The allowed temperature difference between the two temp sensors are set by the system integrator and programmed into OTP.

The TempSensor2 & TempSensor3 test fails when the difference between the two temperature sensors are greater than the set allowed difference, or when the TempSensor reading is 0V.

Protection OT BIST is included in the ADC BIST execution time at startup.

Detection Time Interval (DTI):
Minimum DTI is equivalent to the ADC sampling rate in Table 1. This time is affected by the Digital Filter (IIR Coefficient) setting, at the discretion of the system integrator.

b) Hardware Description

See Figure 5.

c) Recommended Usage

The safety mechanism shall be active at startup and continue to remain active.
If the system integrator has activated OV/UV monitoring via ADC, OT BIST is activated as well.

The PMIC does not proceed to ACTIVE state if any of the Built in Self Tests (BIST) fails at startup. This can be confirmed by checking that LBIST, ADC BIST and Fault Detect BIST passed at startup See Section 7.11 and Section 7.12 for the registers that hold the status of BIST.

d) Fault Control and Operation

During the ADC self-test, and throughout operation, TempSensor3 ADC reading is compared against TempSensor2 ADC reading.

The registers involved are:

Table 23 OT BIST Registers

REGISTER	DESCRIPTION
ADCMON_DATAMSB_Temp2[7:0]	Upper byte for ADC result for temperature sensor TEMP2 (protection) measurement
ADCMON_DATALSB_Temp2[7:0]	Lower byte for ADC result for temperature sensor TEMP2 (protection) measurement
ADCMON_DATAMSB_Temp3[7:0]	Upper byte for ADC result for temperature sensor TEMP3 (protection) measurement
ADCMON_DATALSB_Temp3[7:0]	Lower byte for ADC result for temperature sensor TEMP3 (protection) measurement

A 0V reading of any or both of the sensors can indicate a fault in the temperature sensors, a failure in the sensor Enable signals, or an error in the temp sensors' supply.

An ADC BIST fail status is reported to the Safety Control Logic, and the **FLT_RECORD_BG_Temp[4]: FaultStatus_TEMP2_SENSOR** bits are set.

A TempSensor test fail results to an ADC BIST fail. When this condition is detected, the Safety Control Logic controls the following within FTTI:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- Outputs are stopped according to set delay
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate).

e) Test Concept

Not Applicable.

7.14 SM26: SDOs TEST

a) Overview

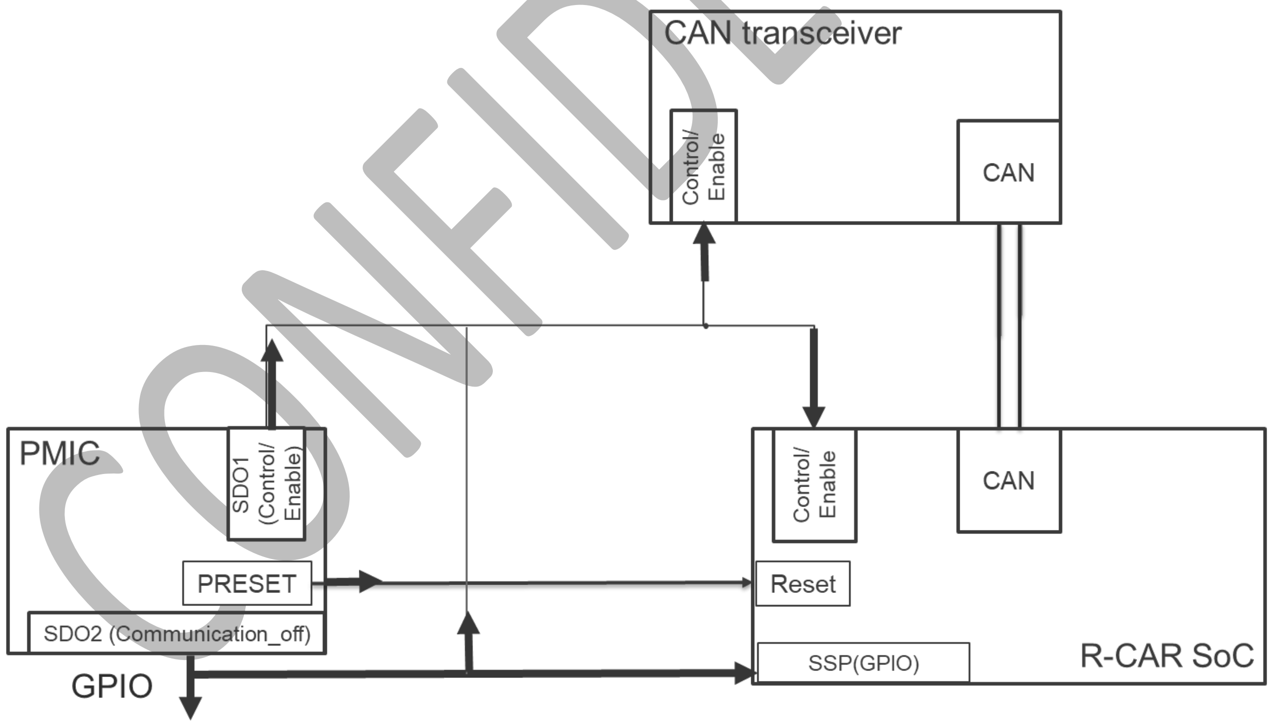
These are safety state dependent output signals whose logic levels are configurable per safety state according to the recommendation from the System Integrator.

Detection Time Interval (DTI):
Once per driving cycle, included in the Total SoC Activation Sequence time which can be a maximum of 250ms.

b) Hardware Description

The figure below is an example of the SDO connectivity in a system. In the assumed connection, the SDO signals are LOW during PMIC safe states. Testing the SDOs is accomplished via SPI/I2C.

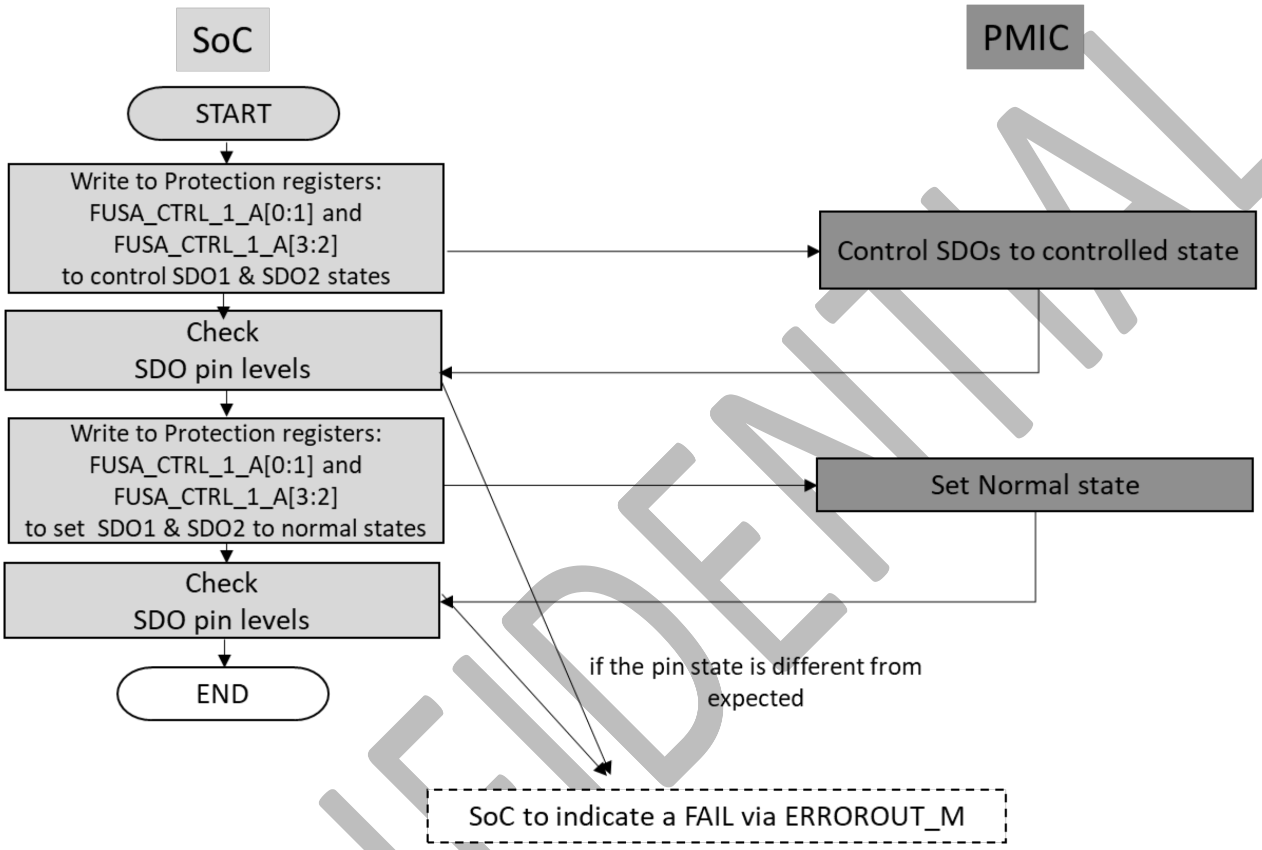
Figure 26 Example connection of SDOs in a system using a CAN transceiver



c) Recommended Usage

The test shall be performed during SoC Activation Sequence, in parallel with CVM test. See Figure 39.

Figure 27 SDOs Test Concept



Details of Protection Block Assert Register for testing:

0x008 - FUSA_CTRL_1_A

BIT	NAME	R/W	DESCRIPTION
7:4	Reserved	RO	
3:2	SDO_1_ASSERT	RW	0x0 : No assert 0x1 : Assert 1 0x2 : Assert 0 0x3 : Tri-stated

BIT	NAME	R/W	DESCRIPTION
1:0	SDO_0_ASSERT	RW	0x0 : No assert 0x1 : Assert 1 0x2 : Assert 0 0x3 : Tri-stated

d) Fault Control and Operation

When PMIC receives ERROROUT_M, the Safety Control Logic controls the following within FTTI:

- The device transitions to 'RESET' state.
- The **FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt** register is updated with the incremented SoC Error Count.
- PRESET#=L is applied and PRESETOUT0#=L loop back is checked
- If RESET loopback completes and passes, RAA271005 moves to SoC Activation State
- If RESET loopback fails, RAA271005 moves to ERROR state
- During this state, all monitoring are still applied per Table 36.

e) Test Concept

Not applicable.

7.15 SM27a, SM27b, SM27c, SM27d: SOC Activation Sequence Tests

a) Overview

RAA271005 shall assist in the SoC Activation Test process[1].
The SoC Activation Sequence is a series of tests including the SoC POST and boot, that takes place before the PMIC goes to ACTIVE state.

The Duration of SoC Activation state can be programmed through register **FUSA_TIMER_2[7:5]: TIMEOUT_SOCACTIVA_ST.**

FUSA_STATUS_1[4:4]: SOC_ACTIVATED register status (0x1) indicates that SoC activation executed successfully.

The order of the tests:
- SM27a: PRESET# Check
- SM27b: Serial Interface Check
- SM27c: External Pin Check2
- SM27d: CVM Test

PRESET# Check is executed by the PMIC automatically.

Serial Interface Check is initiated by the SoC through writing to a specific register. Failure to do this will generate a fault.

External Pin Check2 (ERROROUT_M#, PRESETOUT0#, VMONOUT#(0,1), SDOs) and the CVM tests are initiated by the SoC through writing to a specific register.
If the registers are not written to during the SoC Activation State, RAA271005 will not generate a fault reaction.

The system integrator shall ensure that the External Pin Check2 and CVM Tests are executed during the SoC Activation Sequence.

Detection Time Interval (DTI) of Device for SoC Activation Tests is once per driving cycle, included in the Total SoC Activation Sequence time which can be a maximum of 250ms.

---

[1] The RAA271005 can support the SoC Activation Test according to “R-CarGen4_SAN”.

b) Hardware Description

Figure 28 PMIC + S4 SoC Activation Test at Cold Start

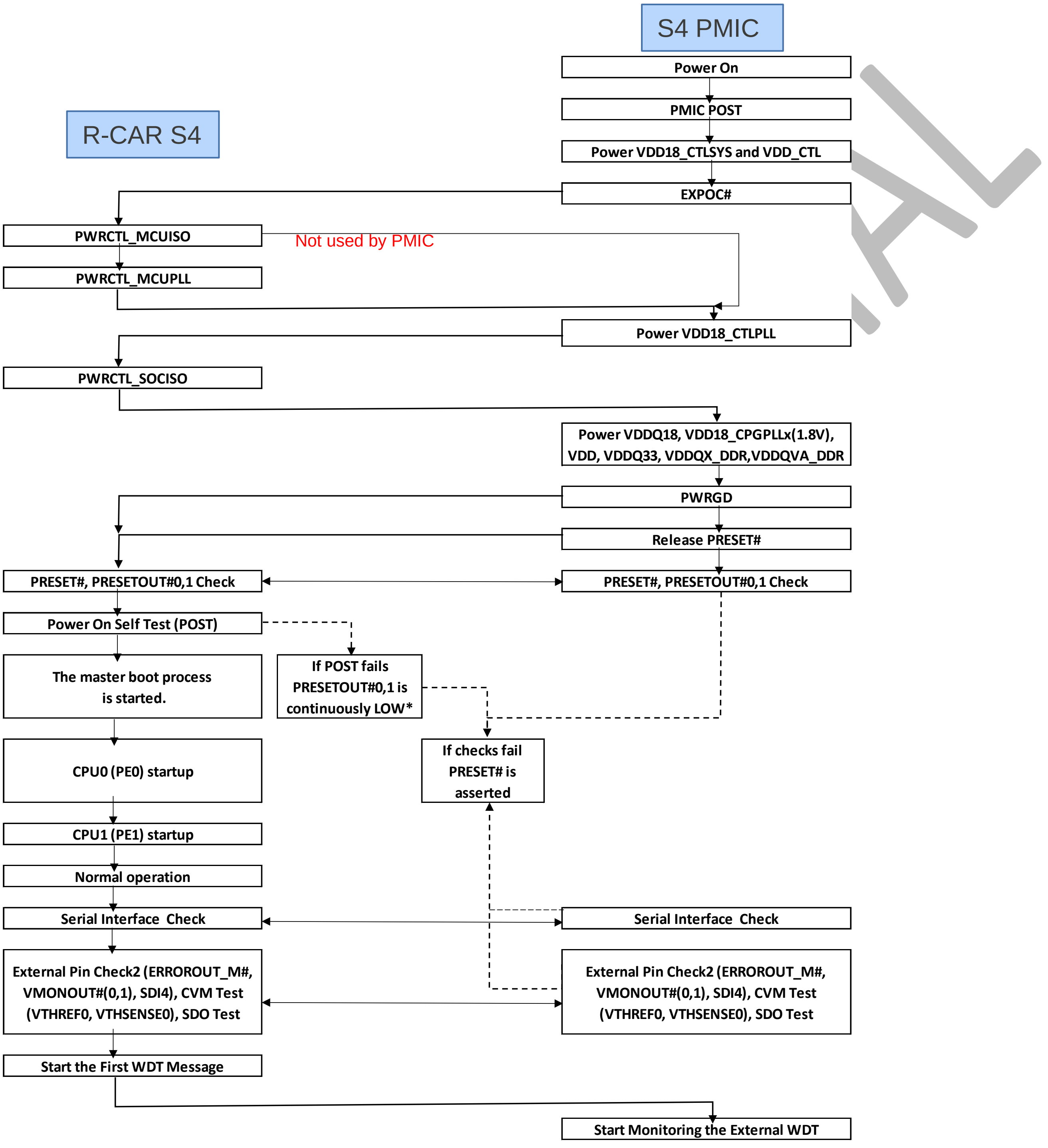


Figure 29 PMIC + S4 SoC Activation Test at transition from Deep Stop or DDR Backup to Run Mode

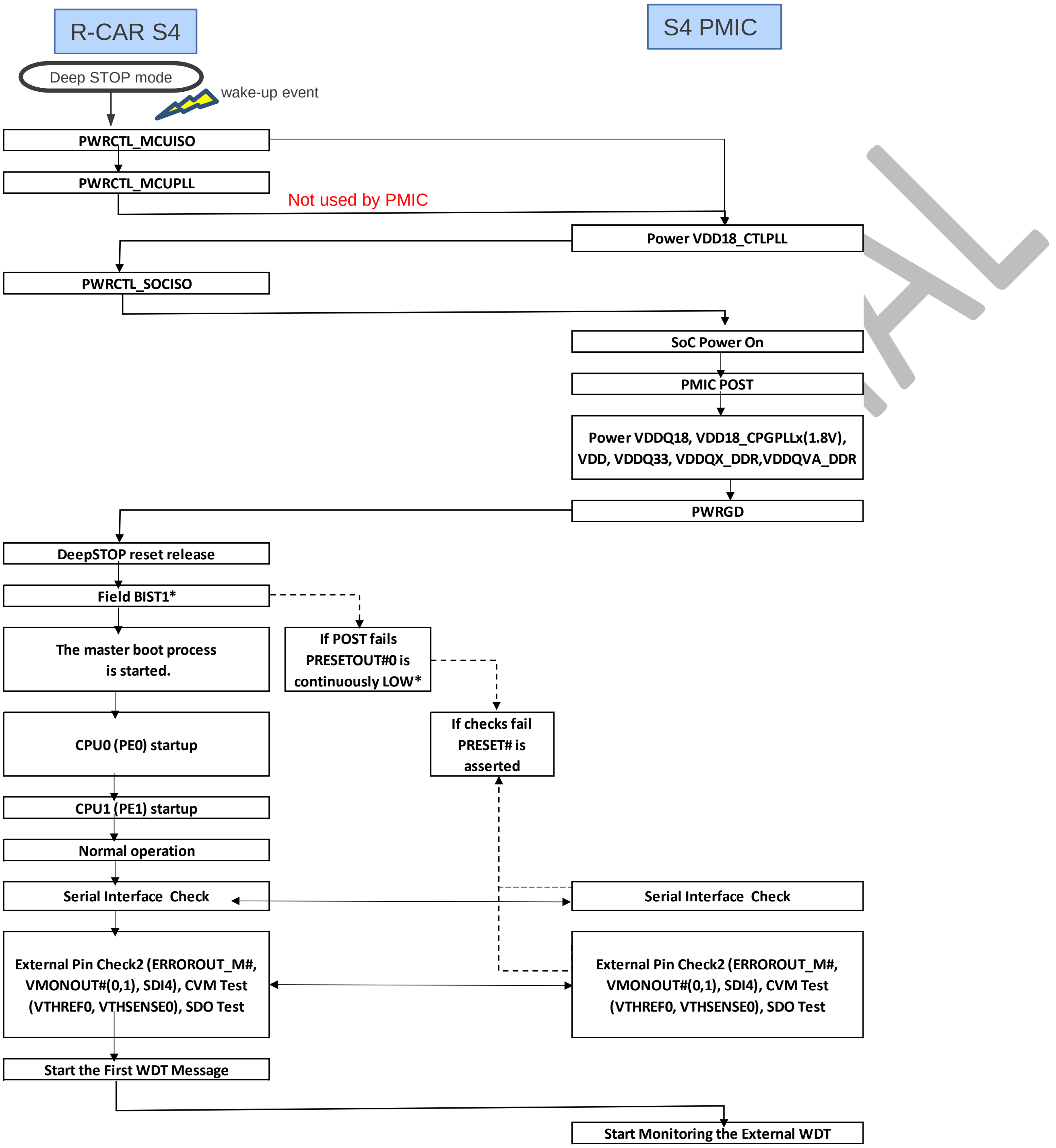
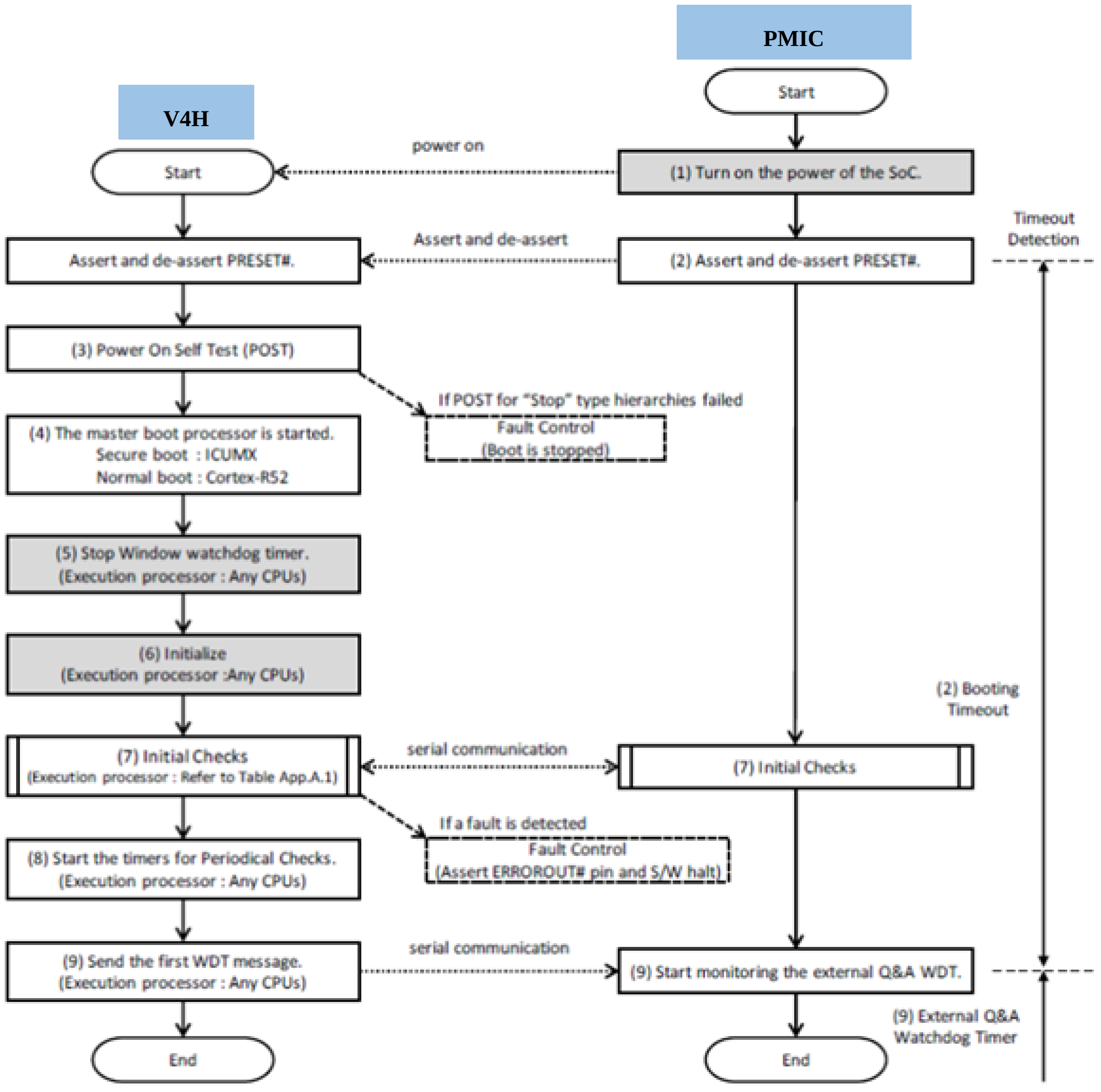


Figure 30 PMIC + V4H SoC Activation Test at Cold Start & RAM Refresh Mode.



c) Recommended Usage, Fault Control & Operation

SoC Activation Sequence is enabled by default in the PMIC.
See following sections for more details.

SM27a: PRESET#, PRESETOUT0# Check

Cold start to Full Run:

This is executed when PRESET# is released by RAA271005.

The purpose of this test is to check any stuckat condition in the PRESET# and PRESETOUT0# IO's, check proper connection between PMIC & SoC with regards to PRESET# & PRESETOUT0# and confirm that SoC POST PASSED.

Deep Stop or DDR Backup mode to Full Run:

This is executed when PWRGD goes HIGH.

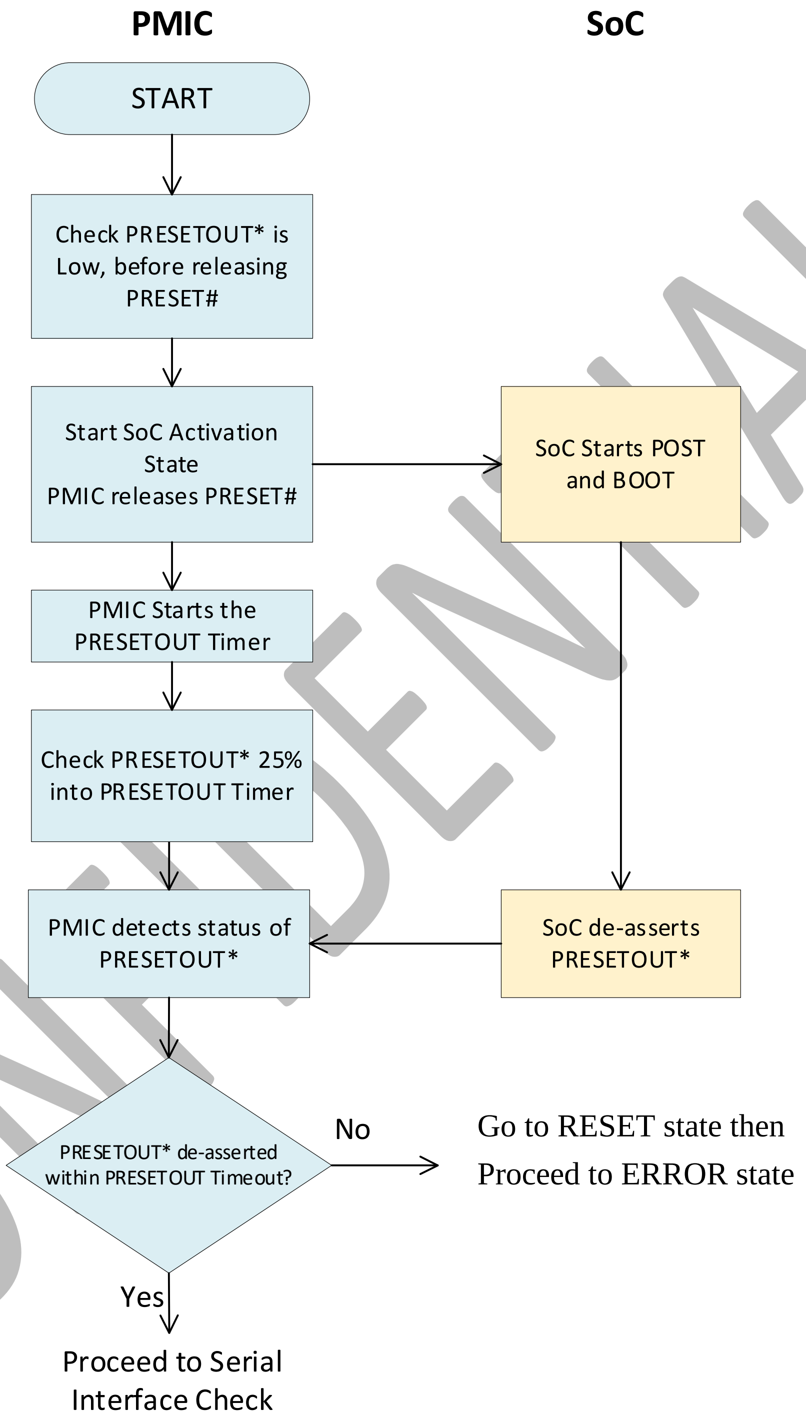
If applicable, the purpose of this test is to confirm that SoC Field BIST PASSED.

The system integrator shall determine the optimum PRESETOUT0# timeout setting that needs to be programmed into OTP register: **FUSA_TIMER_3[7:6]: TIMEOUT_PRESET_CHK_TOUT**. This register sets the maximum time the PMIC waits for PRESETOUT0# to be de-asserted after PMIC de-asserts PRESET#. This value is between 8ms to 32ms.

PRESET# Check Test Procedure in Figure 31:

1. PMIC releases PWRGD after all supplies are within spec
2. PMIC performs Test1: Check PRESETOUT0# = Low, before releasing PRESET#
3. PMIC releases PRESET# 100us after PWRGD signal is high
4. PMIC Performs Test1 again, 25% into TIMEOUT_PRESET_CHK_TOUT duration.
5. PMIC starts a timer –different setting for this timer are provided through OTP register **FUSA_TIMER_3[7:6]:TIMEOUT_PRESET_CHK_TOUT** and expects PRESETOUT0# to be released before the timer runs out.

Figure 31 PRESET# Check Test Procedure



Pass Criteria:

(1) Detecting PRESETOUT* "Low" when PRESET# is de-asserted, and 25% into TIMEOUT_PRESET_CHK_TOUT

(2) Detecting PRESETOUT* de-asserting within FUSA_TIMER_3[7:6]:TIMEOUT_PRESET_CHK_TOUT timer.

Fault Reaction:

RAA271005 goes to RESET state after TIMEOUT_PRESET_CHK_TOUT timeout and then goes to ERROR state. Both SoC RESET Counter and PMIC Error Counter will increment. PMIC will write 0x1 to **FLT_RECORD_A[7:7]** if PRESET check fails.

Figure 32 PRESET#, PRESETOUT* Check at Cold Start to Full Run Timing Diagram:

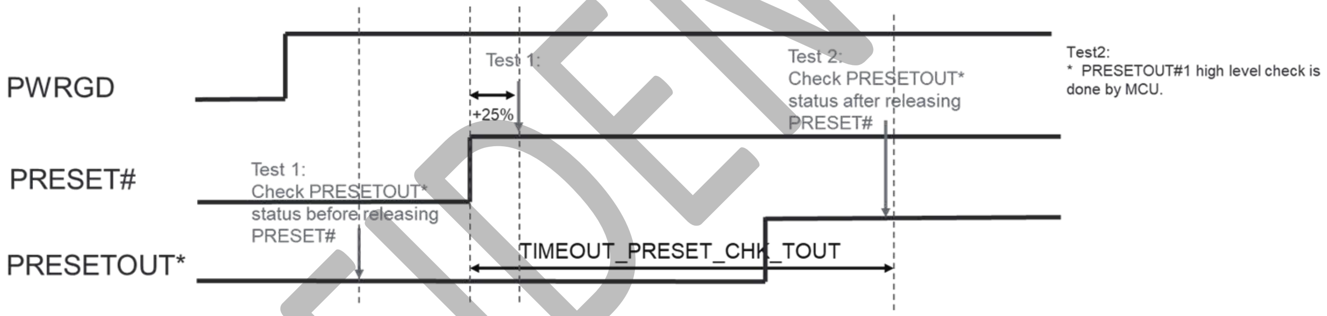


Figure 33 PRESETOUT* Check at Deep Stop or DDR Backup mode to Full Run Timing Diagram: Applicable to RCAR- S4 only.

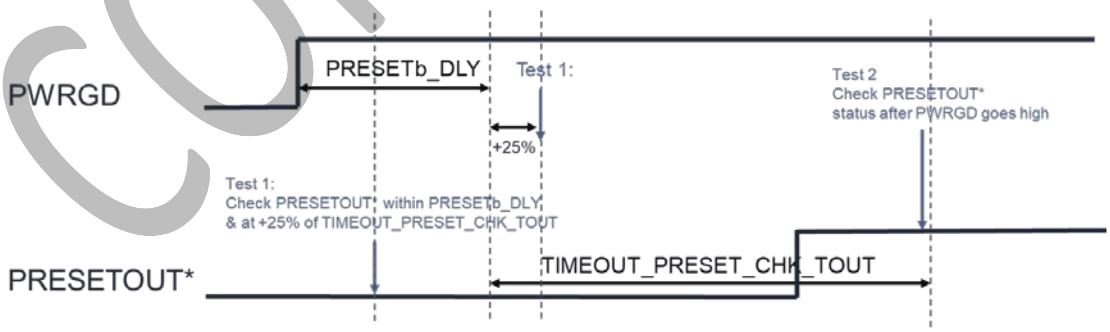


Table 24 PRESET# Check Related Registers

REGISTER ADDRESS	NAME	DESCRIPTION	
0x118	**FUSA_TIMER_3[7:6]:** *TIMEOUT_PRESET_CHK_TOUT*	Maximum time which PMIC waits before issuing a fault if PRESETOUT is not de-asserted after PRESET release.	
		2'b00	8mS
		2'b01	16mS
		2'b10	24mS
		2'b11	32mS

SM27b: Serial Interface (SINT) Check

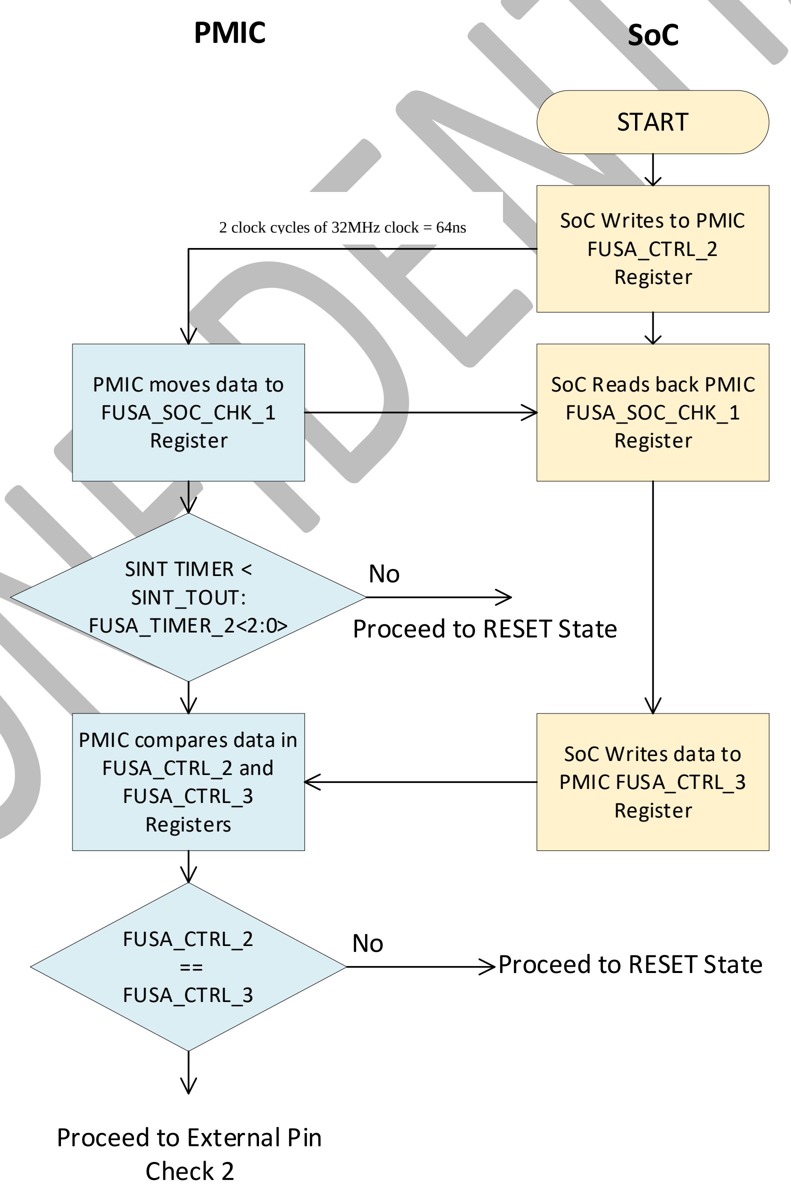
This is executed after the PRESET# Check.

This checks the serial communication interface between the SoC & PMIC.

This also checks the correct functioning of the I2C/SPI Interface Selection Multiplexer.

The system integrator shall determine the optimum Serial Interface Test timeout setting that needs to be programmed into OTP register **FUSA_TIMER_2[2:0]: SINT_TOUT.** This register sets the maximum allowed time for the Serial Interface test to finish after the SoC initiates the test by writing to the **FUSA_CTRL_2[7:0]** register.

Figure 34 Serial Interface Check Test Procedure



Pass Criteria:

FUSA_CTRL_2[7:0] == FUSA_CTRL_3[7:0] within FUSA_TIMER_2[2:0]:SINT_TOUT timer.

No write to FUSA_CTRL_2 during the SoC Activation state will be considered as fault and the fault reaction will happen at the end of SoC Activation state.

Fault Reaction:

If SINT fails due to wrong register value WRITE, RAA271005 goes to RESET state immediately. If SINT fails due to no WRITE, RAA271005 goes to RESET state after **FUSA_TIMER_2[2:0]: SINT_TOUT** timer expires. PMIC will write 0x1 to **FLT_RECORD_A[6:6]** if SINT fails.

Table 25 Serial Interface Check Related Registers

REGISTER ADDRESS	NAME	DESCRIPTION	
0x009	**FUSA_CTRL_2[7:0]**	Writing to this register starts SINT check.	
0x015	**FUSA_SOC_CHK_1[7:0]**	Data copy register	
0x00A	**FUSA_CTRL_3[7:0]**	Data rewrite register	
0x117	**FUSA_TIMER_2[2:0]:***SINT_TOUT*	Maximum allowed time for SINT check to finish after FUSA_CTRL_2[7:0] register write.	
		2'b011	1056uS
		2'b100	2mS
		2'b101	4mS
		2'b110	8mS
		2'b111	16ms

SM27c: External Pin Check2

This checks for the SDI1(ERROROUT*), SDI2 (VMONOUT0#/CVM_OUT), and SDI3 (VMONOUT1#) signals' stuckat condition, and also checks for connectivity of the signals between SoC & PMIC.

The system integrator shall ensure that the External Pin Check2 is executed during the SoC Activation Sequence. This is done by properly configuring the OTP register **FUSA_CTRL_D[4:0]: EXTPINCHK2_EN** to select or ignore the SoC – PMIC interface pins to check. See Table 27.

Figure 35 External Pin Check2 Test Procedure

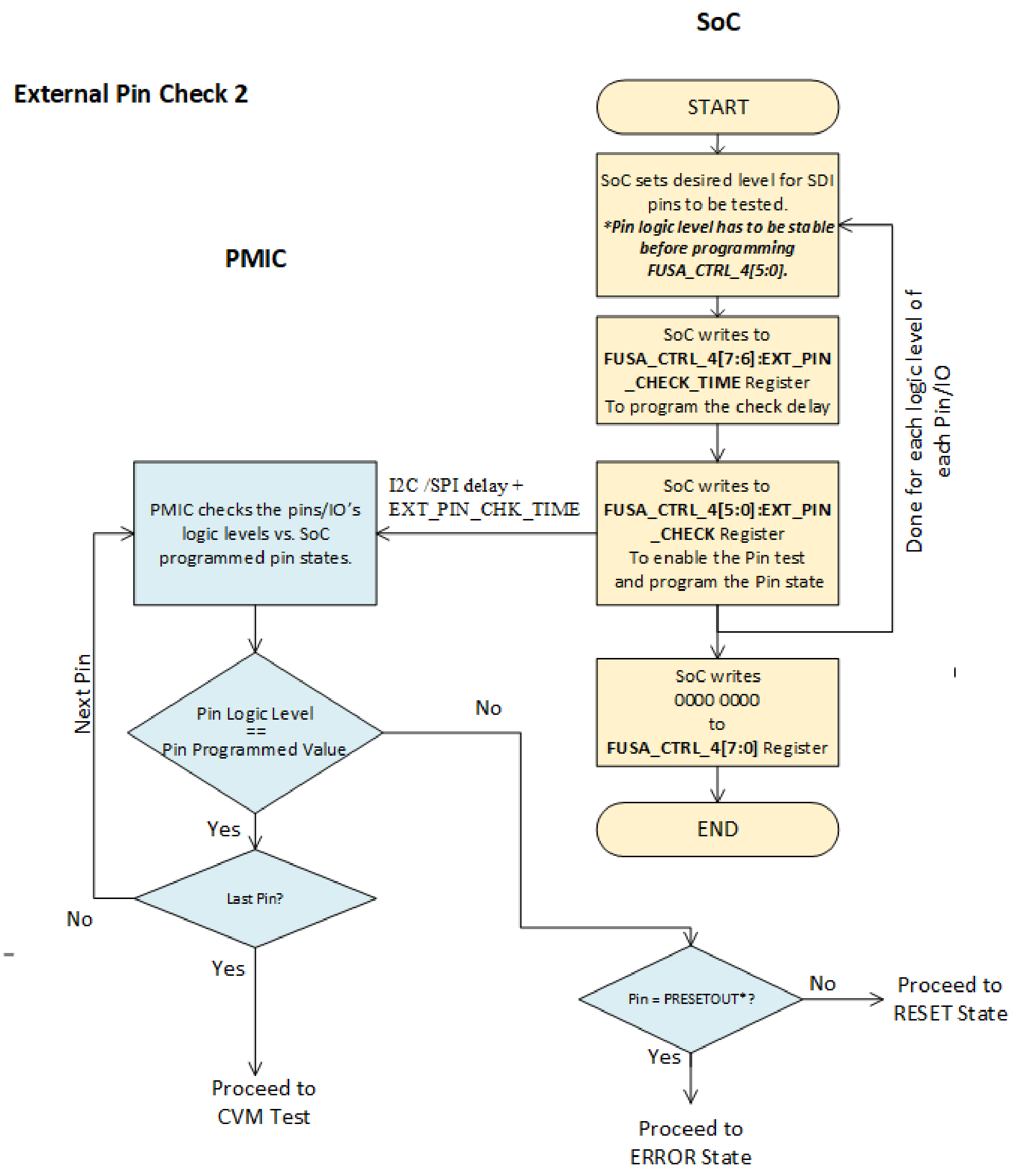
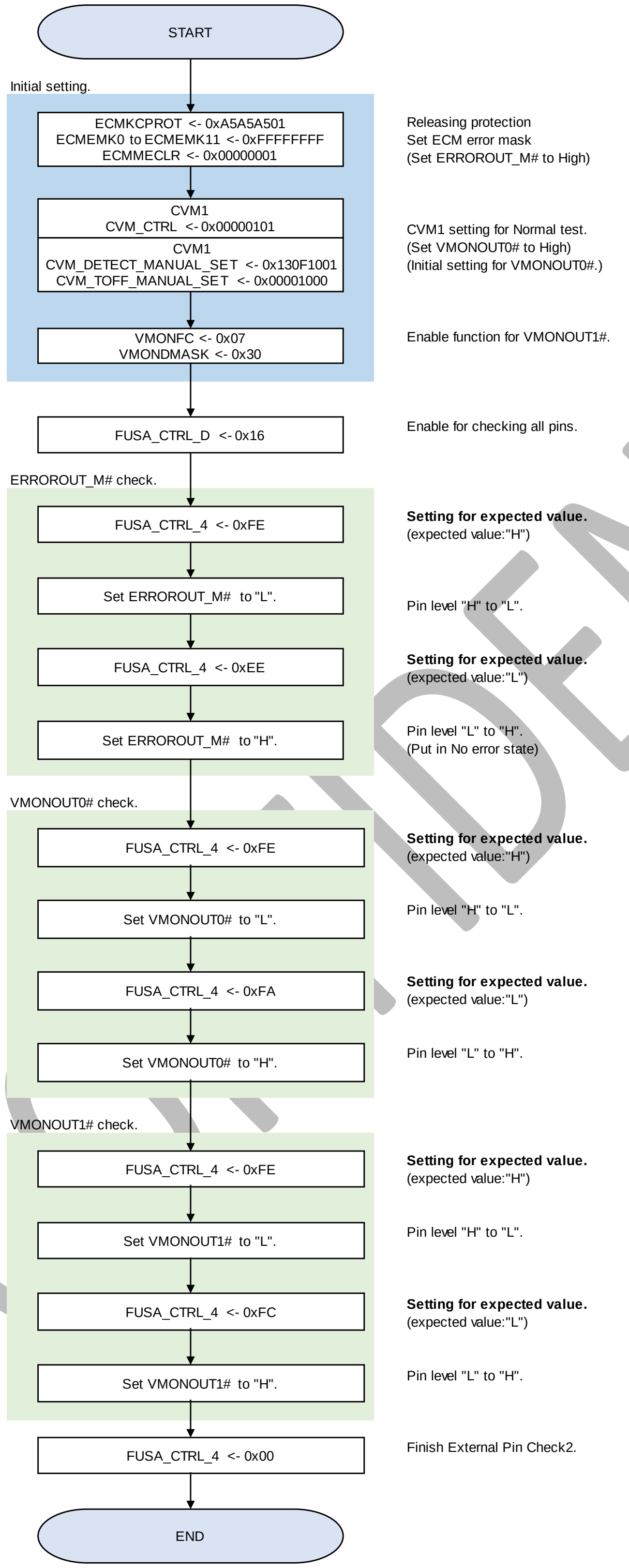


Figure 36 (a) & (b) Examples of External Pin Check2 Test Procedures for S4 SoC

(a)Normal Test



(b)Abnormal Test

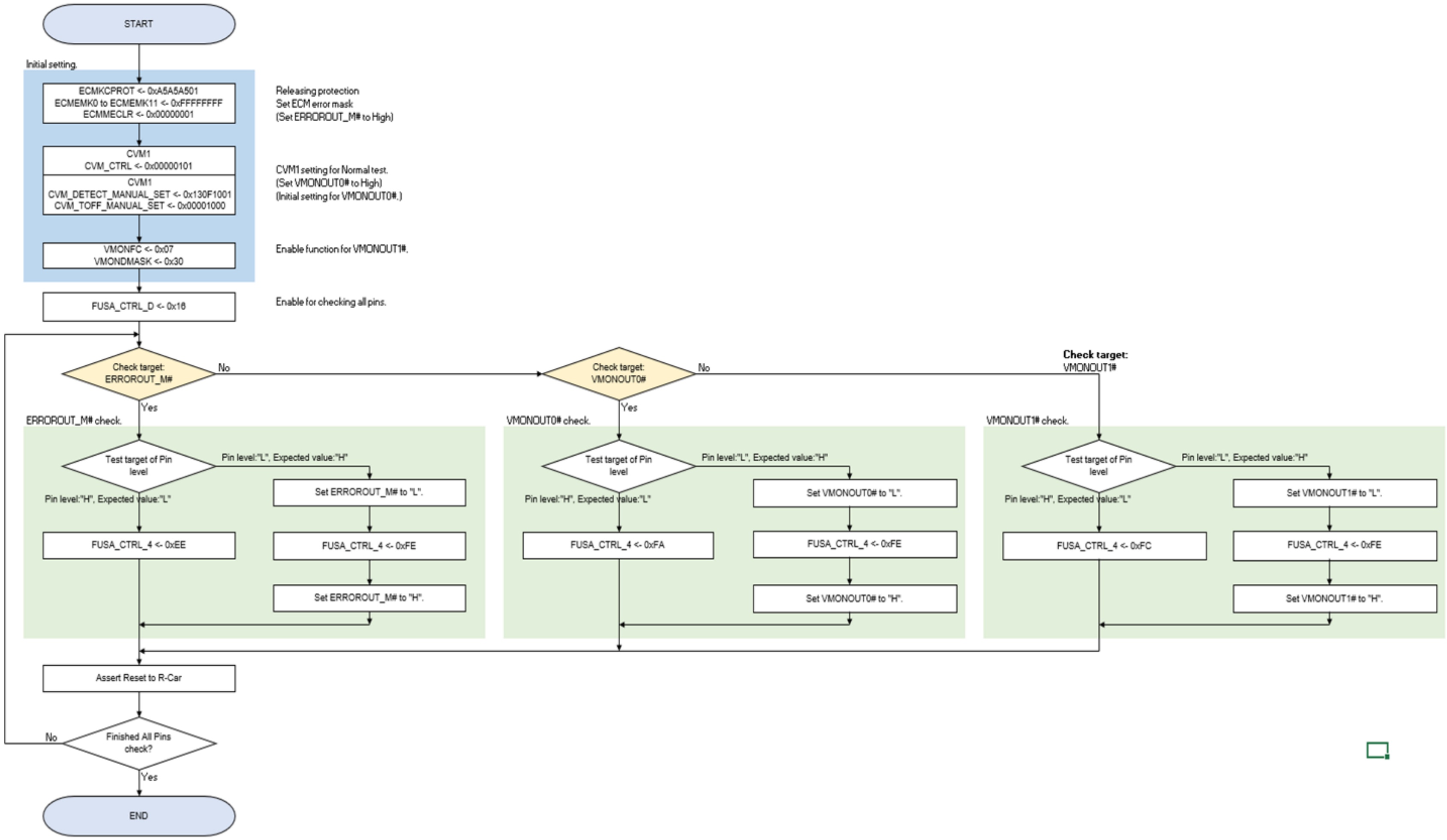


Table 26 Example Programming Sequence of FUSA_CTRL_4

FUSA_CTRL_4 [7:0]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
	Delay		EN	IO7 SDI1	IO11 PRESETOUT (not required)	IO12 SDI2	IO15 SDI3	SDI4
xx0x_xxxx			No Check					
0011_1111	4uS		✓	H	H	H	H	H
0010_1111	4uS		✓	L	H	H	H	H
0011_1111	4uS		✓	H	H	H	H	H
0111_0111	8uS		✓	H	L	H	H	H
0111_1111	8uS		✓	H	H	H	H	H
1011_1011	16uS		✓	H	H	L	H	H
1011_1001	16uS		✓	H	H	L	L	H
1011_1011	16uS		✓	H	H	L	H	H
1011_1111	16uS		✓	H	H	H	H	H
1111_1110	32uS		✓	H	H	H	H	L
1111_1111	32uS		✓	H	H	H	H	H

Pass Criteria:

Logic levels of all enabled IOs are the same as programmed values.

Fault Reaction:

RAA271005 goes to ERROR state when PRESETOUT pin fails External Pin Check test. RAA271005 goes to RESET state after SoC Activation Timeout expires, when other SDI pins fail. PMIC will write 0x1 to **FLT_RECORD_A[5:5]** in the case of an External Pin Check test failure.

No write to FUSA_CTRL_4 register during the SoC Activation state will **NOT** be considered as fault.

Table 27 External Pin Check2 Related Registers:

REGISTER ADDRESS	NAME	DESCRIPTION	
0x00B	**FUSA_CTRL_4[5:0]:***EXT_PIN_CHK*	[5]	1'b0=Check disabled 1'b1=Check enabled
		[4]	Expected state for SDI1 1'b0=Logic Low 1'b1=Logic High
		[3]	Expected PRESETOUT pin state (*not required) 1'b0=Logic Low 1'b1=Logic High
		[2]	Expected SDI2 pin state 1'b0=Logic Low 1'b1=Logic High
		[1]	Expected SDI3 pin state 1'b0=Logic Low 1'b1=Logic High
		[0]	Expected SDI4 pin state 1'b0=Logic Low 1'b1=Logic High
0x00B	**FUSA_CTRL_4[7:6]:***EXT_PIN_CHK_TIME*	Delay time for PMIC to verify expected pin states after every **FUSA_CTRL_4** register write [4u,8u,16u,32u]	
0x11D	**FUSA_CTRL_D[4:0]:***EXTPINCHK2_EN*	[4]	1'b0=ERROROUT* (SDI1) is NOT included to this check. 1'b1=ERROROUT* (SDI1) is included to this check.
		[3]	1'b0=PRESETOUT* is NOT included to this check. 1'b1=PRESETOUT* is included to this check.
		[2]	1'b0=VMONOUT0#/CVM_OUT is NOT included to this check. 1'b1=VMONOUT0#/CVM_OUT is included to this check.
		[1]	1'b0=VMONOUT#1 is NOT included to this check. 1'b1=VMONOUT#1 is included to this check.
		[0]	1'b0=SDI4 is NOT included to this check. 1'b1=SDI4 is included to this check.

SM27d: CVM TEST (VTHREF0, VTHSENSE0)

The CVM test checks the SoC Core Voltage Monitor (CVM) functionality. It also checks the CVM error signal pins VTHREF0 and VTHSENSE0, for stuckat condition and correct interfacing between SoC & PMIC.

RAA271005 can support 4 “target” voltages for CVM testing (ex. R-CAR Gen4 predefined voltages are: 0V, 0.8V, 1.24V, 2.03V). Eight 8bit registers are provided for the system to program “high” and “low” thresholds for each target voltage.

RAA271005 supports CVM testing on 6 slots. Each slot has an 8 bit control register to enable/disable the “slot”, program the “slot” delay time, and the target voltage of the “slot”.

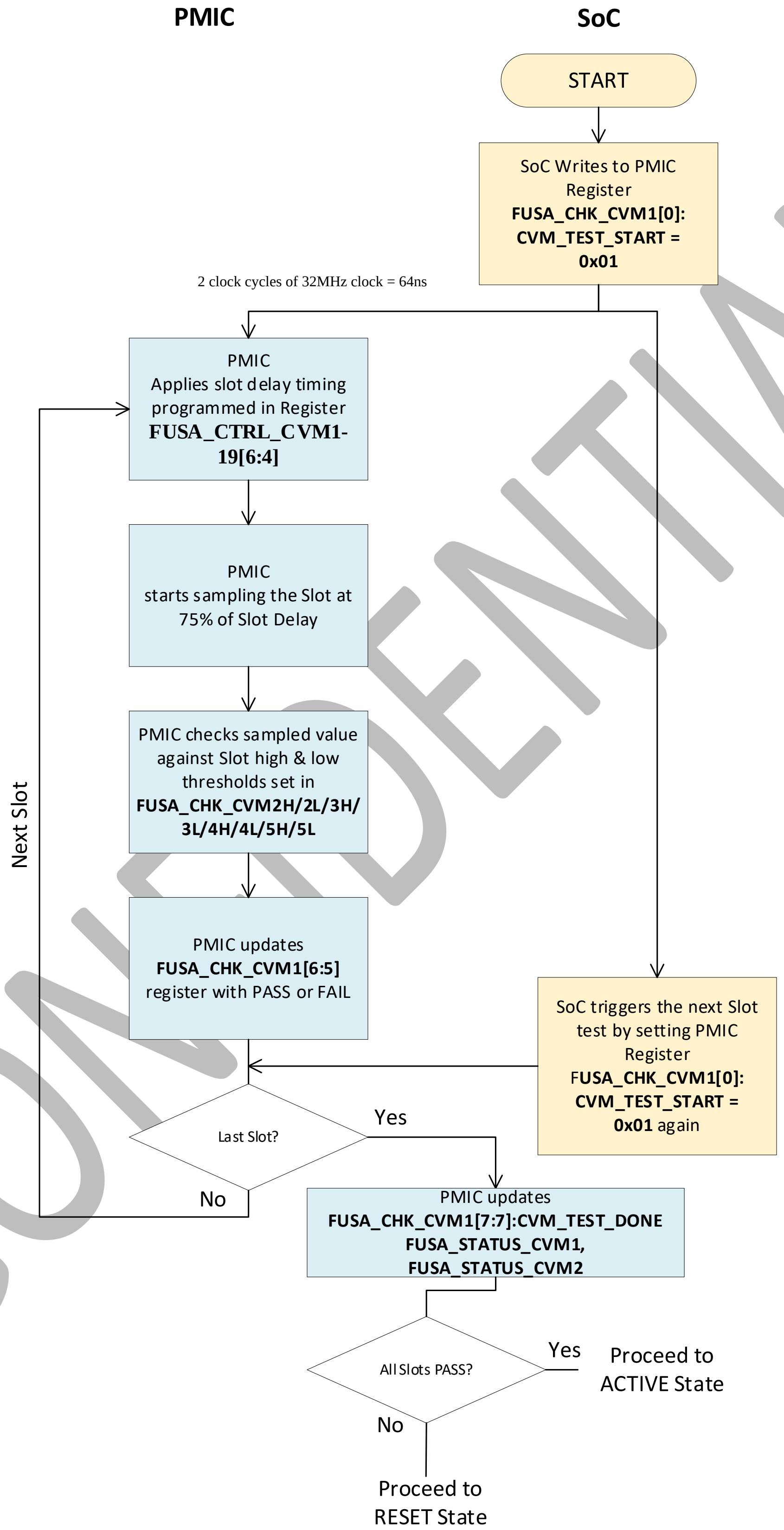
For R-CAR Gen4, the “slot” is identified by the associated CVM Mode described in R-CAR Gen4 SAN, the expected voltage, and the applied High & Low Threshold, for both VTHREF0 and VTHSENSE0 pins.

<u>**Considerations for CVM Test:**</u>

1. VTHREF0 and VTHSENSE0 signals shall be routed to RAA271005 External channels ADC1 and ADC2 respectively.
2. Configuration of CVM slots via **FUSA_CTRL_CVM<*1-6*>[7:0]** registers should be done by OTP to ensure that configuration is set before start of SoC Activation, otherwise CVM Test results will not be recorded.
3. SoC program the sampling interval (also called “slot delay”) and the target voltages for ADC1 & ADC2 pins at every measurement “slot”. This shall be done via OTP. See Table 30.
4. Signals shall be settled 40uS before the programmed measurement times.
5. SoC to configure **FUSA_CHK_CVM<*2H,2L/3H,3L/5H,5L*>** registers to program high and low thresholds of the target voltages which RAA271005 will apply to the measured values. This shall be done via OTP.
6. To disable a slot measurement, the SoC sets the applicable **FUSA_CTRL_CVM<*1-6*>[7]** register bit to zero.
7. The PMIC & SoC waits out the configured slot timing of each CVM slot before the next CVM slot can begin testing.

8. The CVM slot test status is updated in registers **FUSA_STATUS_CVM<1,2>**, once PMIC completes SoC activation sequence & transitions to ACTIVE state when no faults are detected, or RESET state when faults are detected.
9. The CVM test PASS/FAIL status per slot is updated dynamically in register **FUSA_CHK_CVM1[6:5]**. SoC should wait until after the slot time has expired or before each new CVM test pulse is sent to read the register.
10. Write “1” to CVM_TEST_START at every enabled CVM slot, to test all enabled slots, otherwise PMIC will wait until the end of SoC Activation timeout timer before going to RESET state.
11. ADC1/2 monitoring should not be masked (FaultMask_EXT_0_Prot=0 & FaultMask_EXT_1_Prot =0).

Figure 37 CVM Test Procedure



Pass Criteria:

Measured values are always within the allowed limits of target voltages.

Fault Reaction:

RAA271005 goes to RESET state as soon as PMIC completes SoC activation sequence.

Figure 38 R-CAR Gen4 SoC CVM Test Sequence & SoC Communication events during CVM Test

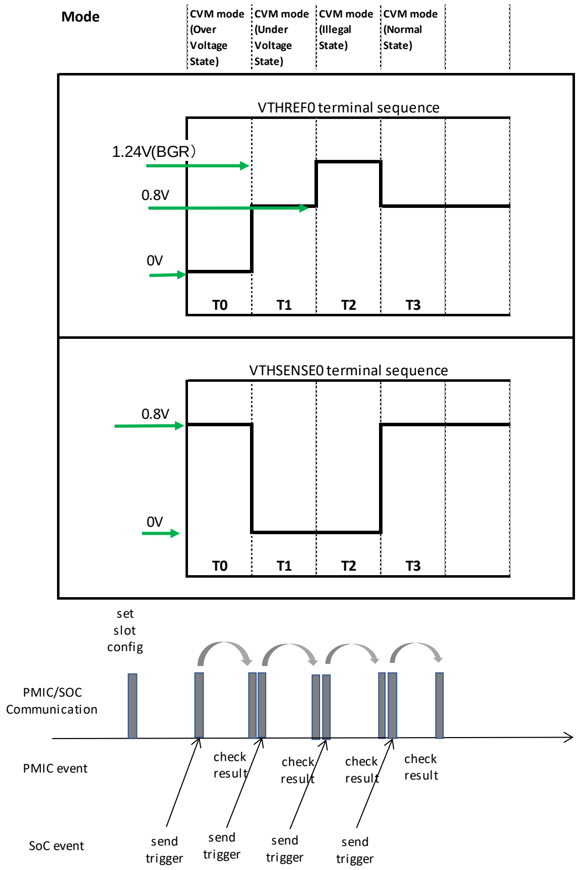


Table 27 shows an example of the measurement times made at each sampling interval. The system integrator shall ensure that slot delay time > Wait Time.

Table 28 R-CAR Gen4 CVM SLOT Timings

Number	Status	CVM SLOT Number	VTHREF0		VTHSENSE0		Wait time[uS]
			Voltage[V]	Analog buffer time[uS]	Voltage[V]	Analog buffer time[uS]	
T0	CVM mode(Over Voltage State)	1	0	260	0.8	0	260
T1	CVM mode(Under Voltage State)	2	0.8	140	0	260	260
T2	CVM mode(Illegal State)	3	1.24	220	0	0	220
T3	CVM mode(Normal State)	4	0.8	1270	0.8	140	1270

Table 29 shows what values to set CVM threshold registers based on the expected slot voltage. Register details are provided in Table 30.

Table 29 R-CAR Gen4 CVM Threshold Registers Settings

Number	CVM SLOT Number	Status	VTHREF0					VTHSENSE0					CVM Slot Register
			Expected VTHREF0 Voltage[V]	Register to set High Threshold	High Threshold	Register to set Low Threshold	Low Threshold	Expected VTHSENSE0 Voltage[V]	Register to set High Threshold	High Threshold	Register to set Low Threshold	Low Threshold	CVM Slot Register
T0	1	CVM mode(Over Voltage State)	0	FUSA_CHK_CVM2H	0x14	FUSA_CHK_CVM2L	0xEE	0.8	FUSA_CHK_CVM3H	0x3A	FUSA_CHK_CVM3L	0x2B	FUSA_CTRL_CVM1
T1	2	CVM mode(Under Voltage State)	0.8	FUSA_CHK_CVM3H	0x3A	FUSA_CHK_CVM3L	0x2B	0	FUSA_CHK_CVM2H	0x14	FUSA_CHK_CVM2L	0xEE	FUSA_CTRL_CVM2
T2	3	CVM mode(Illegal State)	1.24	FUSA_CHK_CVM4H	0x52	FUSA_CHK_CVM4L	0x49	0	FUSA_CHK_CVM2H	0x14	FUSA_CHK_CVM2L	0xEE	FUSA_CTRL_CVM3
T3	4	CVM mode(Normal State)	0.8	FUSA_CHK_CVM3H	0x3A	FUSA_CHK_CVM3L	0x2B	0.8	FUSA_CHK_CVM3H	0x3A	FUSA_CHK_CVM3L	0x2B	FUSA_CTRL_CVM4

Table 30 CVM Test Related Registers

REGISTER ADDRESS	NAME	DESCRIPTION	
0x125 – 0x128	FUSA_CTRL_CVM1-4[7:0]	[7]	1'b0=Slot check disabled 1'b1=Slot check enabled
		[6:4]	Slot delay for ADC measurement 3'b000: 256u 3'b001: 384u 3'b010: 512u 3'b011: 640u 3'b100: 1024u 3'b101: 1280u 3'b110: 1920u 3'b111: 2816u
		[3:2]	Expected VTHREF0 voltage (from ADC1 pin) 2'b00: 0V 2'b01: 0.8V 2'b10: 1.24V
		[1:0]	Expected VTHSENSE0 voltage (from ADC2 pin) 2'b00: 0V 2'b01: 0.8V

NAME	DESCRIPTION
FUSA_CHK_CVM1[0]:*CVM_TEST_START*	Writing 1'b1 to this register address starts the CVM test.
FUSA_CHK_CVM1[6:5]:*CVM_TEST_FAIL*	[6] 1'b1=Fault in ADC1 measurements [5] 1'b1=Fault in ADC2 measurements
FUSA_CHK_CVM1[7]:*CVM_TEST_DONE*	1'b0=Test is not executed 1'b1=Test has passed successfully
FUSA_CHK_CVM2H[7:0]	High threshold value to be applied to 0V target for VTHREF & VTHSENSE. LSB is +16mV
FUSA_CHK_CVM2L[7:0]	Low threshold value to be applied to 0V target for VTHREF & VTHSENSE. LSB is -16mV
FUSA_CHK_CVM3H[7:0]	High threshold value to be applied to 0.8V target for VTHREF & VTHSENSE. LSB is +16mV
FUSA_CHK_CVM3L[7:0]	Low threshold value to be applied to 0.8V target for VTHREF & VTHSENSE. LSB is -16mV
FUSA_CHK_CVM4H[7:0]	High threshold value to be applied to 1.24V target for VTHREF. LSB is +16mV
FUSA_CHK_CVM4L[7:0]	Low threshold value to be applied to 1.24V target for VTHREF. LSB is -16mV
FUSA_CHK_CVM5H[7:0]	High threshold value to be applied to any other target voltage (</= 2.032V) for VTHSENSE. LSB is +16mV
FUSA_CHK_CVM5L[7:0]	Low threshold value to be applied to any other target voltage (</= 2.032V) for VTHSENSE. LSB is -16mV
FUSA_CHK_CVM6H[7:0]	High threshold value to be applied to any other target voltage (</= 2.032V) for VTHSENSE. LSB is +16mV
FUSA_CHK_CVM6L[7:0]	Low threshold value to be applied to any other target voltage (</= 2.032V) for VTHSENSE. LSB is -16mV

Table 31 Other SoC Activation Checks Registers

REGISTER ADDRESS	NAME	DESCRIPTION
0x117	**FUSA_TIMER_2[7:5]:***TIMEOUT_SOCACTIVA_ST*	Maximum time for all SoC Activation checks to finish (from PRESET release to the first WDT kick). 16mS-32mS-48mS-64mS-80mS-96mS-128m-256mS
0x10	**FUSA_STATUS_1[4]**:*SOC_ACTIVATED*	1'b1: SoC Activation State passed successfully
0x10	**FUSA_STATUS_1[3]**:*WDT_FirstMSG_done*	1'b1: First WDT message received successfully

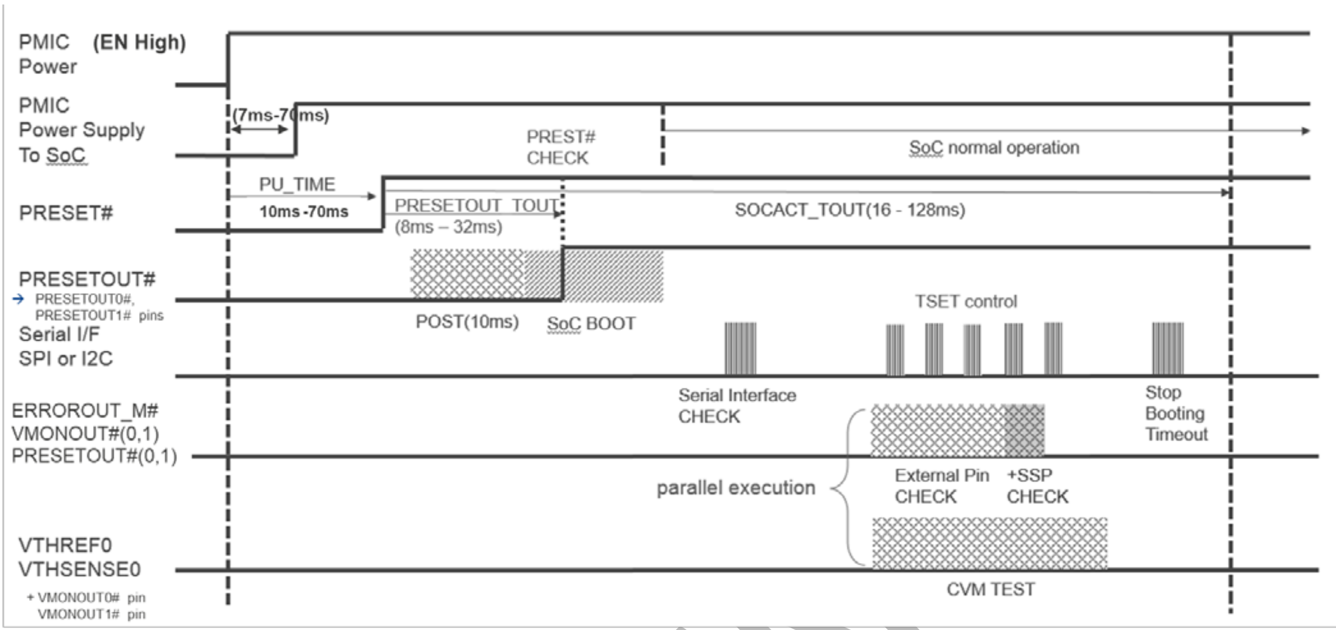
Table 32 SoC Activation Checks Debug Registers

REGISTER ADDRESS	NAME	DESCRIPTION
0x11D	**FUSA_CTRL_D[6]:***PRESET_CHECK_DIS*	1'b0=Check enabled 1'b1=Check disabled
0x10	**FUSA_STATUS_1[2:0]**:*SoCActiva_STATE*	SoC Activation State Bits

d) Test Concept

SoC Activation Sequence fulfills the test concept of the Safety Signaling between the SoC & the RAA271005.
Testing the SoC Activation Sequence is beyond the multiple-points-of-failure scope of the RAA271005.

Figure 39 RAA271005 – Example of SoC Activation Sequence Timing Supported by RAA271005



7.16 SM31: SoC CVM Monitor

a) Overview

The PMIC monitors the status of the SoC Core Voltage Monitor (CVM) through the connection of SoC CVM error outputs VMONOUT* (S4) or CVM_OUT# (V4H) to the PMIC via SDI pins.
See more details about SDIs in Section 12.1 Safety Dependent Inputs (SDI).

b) Hardware Description

Not relevant.

c) Recommended Usage

PMIC IO12 (SDI2) shall be connected to the Application Domian CVM Error signal **VMONOUT0#**(S4) or **CVM_OUT**(V4H).

PMIC IO15 (SDI3) shall be connected to the Control Domain CVM Error signal **VMONOUT1#**(S4).

Fault reaction of CVM error detected shall be configured to transition the PMIC to ERROR state.
This shall be configured by setting:
FUSA_CTRL_E[5] = 0x1 & **FUSA_CTRL_E[6] = 0x1** (S4)**,**
FUSA_CTRL_E[5] = 0x1(V4H)

d) Fault Control and Operation

When **FUSA_CTRL_E** is configured to ERROR state when PMIC receives SDI2 or SDI3, the Safety Control Logic controls the following within FTTI:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate)
- If any failure is detected at startup, while PMIC Error Count is < Maximum allowed threshold, SelfDiagnosis is retried.
- If any failure is detected during normal operation, outputs are stopped according to set delay

e) Test Concept

SDIs are tested at startup during External Pin Check2 in SoC Activation Sequence. See Section 7.15 SM27c and Figure 35 External Pin Check2 Test Procedure.

7.17 SM41: IRQ Test

a) Overview

IRQ can be used by the system to indicate a fault event detected by PMIC safety mechanisms that have been blocked from generating a fault reaction (PRESET# de-asserted). See details on IRQ in Section 11.1.
The IRQ test is accomplished by the SoC & the PMIC to determine a fault in the IRQ IO, and the PMIC digital control of the IO.

Detection Time Interval (DTI) is once per drive cycle, included in the Total SoC Activation Sequence time which can be a maximum of 250ms.

b) Hardware Description

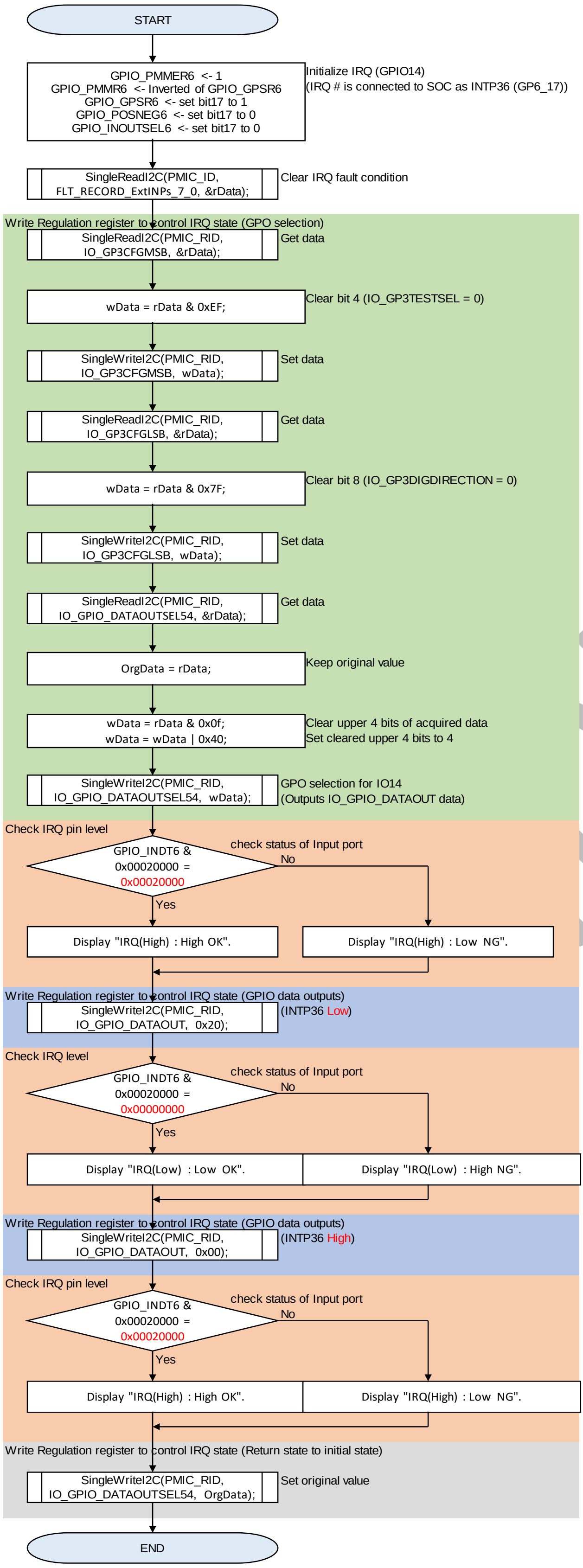
Not relevant.

c) Recommended Usage

The test shall be performed during SoC Activation Sequence, in parallel with CVM test. See Figure 40.

There are 2 options to test IRQ.
Option1: The SoC performs the recommended steps in Figure 41.

Figure 40 IRQ Test Concept (Normal Test)



Details of Regulation block registers:

IO_GPIO_DATAOUTSEL5:IO_GPIO_DATAOUTSEL54[7:4]

0x059 - IO_GPIO_DATAOUTSEL54

BIT	**NAME**	**R/W**	**DESCRIPTION**
7:4	IO_GPIO_DATA OUTSEL5	RW	GPO selection for IO14 0x0 Pin fallows EN_VPP pin state 0x1 PGOOD_GLB (see PGOOD_GLB_CFG_BUCK/LDO) 0x2 PGOOD_CTL1 0x3 PGOOD_CTL2 **0x4 Outputs IO_GPIO_DATAOUT data** 0x5 PGOOD_S2R 0x6 PGOOD_ALWON (EXTPOC) 0x7 BKUP 0x8 IRQ_Regu 0x9 PGOOD_SW (PGOOD for SW SHDN regulators)

0x057 - IO_GPIO_DATAOUT

BIT	NAME	R/W	DESCRIPTION
7:0	IO_GPIO_DATA OUT	RW	8-bit GPIO data outputs for software to control devices in the system. The IO_GPxCFGxSB need to be configured accordingly. IO_GPxDIGDIRECTION should be set to Ouput (0x0); IO_GPxTESTSEL should be set to 0x0;

Option2:

Built-in IRQ Test :

During SoC Activation Sequence, after CVM test, the PMIC drives IRQ low, triggered by false ADC external channels EXT0 and EXT1 faults.

The system has to clear the IRQ fault condition by reading ADC EXT0 & EXT1 Fault Registers:

FLT_RECORD_ExtINPs_7_0[0:0]:FaultStatus_EXT_0_Prot

FLT_RECORD_ExtINPs_7_0[1:1]: FaultStatus_EXT_1_Prot

Fault bits are Clear-on-Read. When read, IRQ is restored to its fault-free state, ready for normal operation.
The system confirms the status of IRQ = High
This tests IRQ stuck condition, as well as the correct IRQ response to fault bit being cleared.

d) Fault Control and Operation

When PMIC receives ERROROUT_M, the Safety Control Logic controls the following within FTTI:

- The device transitions to ‘RESET’ state.
- The FUSA_STATUS_4[7:4]: SafetyCtrl_ErrCnt register is updated with the incremented SoC Error Count.
- PRESET#=L is applied, and PRESETOUT0#=L loop back is checked
- If RESET loopback completes and passes, RAA271005 moves to SoC Activation State
- If RESET loopback fails, RAA271005 moves to ERROR state
- During this state, all monitoring is still applied per Table 36.

e) Test Concept

Not Applicable.

7.18 SM44: GC Self Test

a) Overview

GC Self-test covers the following failure modes of the Over Voltage Protection Gate Control (Datasheet Section 5.7)

Table 33 Over Voltage Protection Gate Control Faults Covered by GC Self Test

	Failure Mode
1	External **PMOS Gate shorted to GND**
2	**GC pin - PGND pin shorted**
3	**GC pin - AVIN1 shorted**
4	**GC pin open**
5	**GC pin - LDO5,6 pin shorted**
6	**GC pin - PH4 pin shorted**
7	External **PMOS Gate - VIN_PRE shorted**

Detection Time Interval (DTI) is once per drive cycle, included in the Self Diagnosis time.

b) Hardware Description

Not relevant.

c) Recommended Usage

For assumed use case Design 2:
FUSA_CTRL_1[0]: Mask_GC_SelfT_fault = 0x0 shall be set to ensure PMIC fault reaction occurs when any of the failure modes in Table 33 is detected during SelfDiagnosis.

If Gate Control (GC) is not used by the system, the GC self test shall be disabled to avoid generating a fault at startup via register: **FUSA_CTRL_C[6]:GC_SelfT_FltRecord_DIS.**

d) Fault Control and Operation

If a GC Self Test fault is detected, the status F**LT_RECORD_A[4]: FLT_GC_SelfT_fault** register is updated.

When an GC Self Test fault is detected, the Safety Control Logic controls the following:

- The device transitions to ERROR state.
- The **FUSA_STATUS_4[3:0]: SafetyCtrl_ErrCnt** register is updated with the incremented PMIC Error Count .
- PRESET# is de-asserted
- SSP is asserted. The asserted state is configurable as (Hi, Lo, or Tristate)
- If any failure mode is detected at startup, while PMIC Error Count is < Maximum allowed threshold, SelfDiagnosis is retried.
- If any failure mode is detected during normal operation, outputs are stopped according to set delay

e) Test Concept

Not Applicable.

7.19 SM45: Gate Control (GC)

a) Overview

See RAA271005 Datasheet Section 5.7

b) Hardware Description

See RAA271005 Datasheet Section 5.7

c) Recommended Usage

See RAA271005 Datasheet Section 5.7

d) Fault Control and Operation

GC continuously monitors VIN_SENSE voltage and assert the GC pin High when VIN_SENSE exceeds a pre-programmed VIN OV threshold to cut off the input voltage to the PMIC.

e) Test Concept

See Section 7.18 SM44: GC Self Test

8 On Chip Device Safe State Transition Mechanisms

8.1 PMIC Error Pin: PRESET#

a) Operation

PRESET# is de-asserted by RAA271005 when a fault is detected by its safety mechanisms to transition the PMIC to its safe state.
PRESET# is also de-asserted by RAA271005 when an SoC error is detected or received by the PMIC through the SoC Error pin(s), to transition the SoC to its safe state.

See Table 36 for fault conditions that can be detected at each state, its corresponding safe state transition, and the PRESET# condition.

Table 34 summarizes PRESET# condition in all PMIC states.

Table 34 PRESET# Condition at each RAA271005 State

PMIC STATE	PRESET# State
Self Diagnosis	Lo
PU Sequence	Lo
SoC Activation	Hi
STIL (System Test)	Hi
ACTIVE (Full Run)	Hi
RESET	Lo
ERROR	Lo
LOCK	Lo
Debug Mode	Hi
Deep Stop	Hi
Suspend to RAM	Hi

When PRESET# is de-asserted, PRESETOUT0# asserted is expected by the RAA271005 within ~100usec to signal successful receipt of the reset signal by the SoC Reset (RST) Module. The RAA271005 then proceeds to SoC Activation state. When PRESETOUT0# is not received during the expected time, the RAA271005 asserts the secondary safety path (SSP) to disable all communication in the system because the SoC operation is unreliable. See Figure 3.

b) System Considerations

Pin Connections:

A pull down resistor at the PRESET# output shall be used to avoid floating conditions due to pin failure and to ensure that the SoC safe state is maintained.

The RAA271005 PRESET# output shall be connected to the input of the Reset (RST) Module of the SoC.

The output of the SoC Reset Module shall be connected to RAA271005 PRESETOUT# pin.

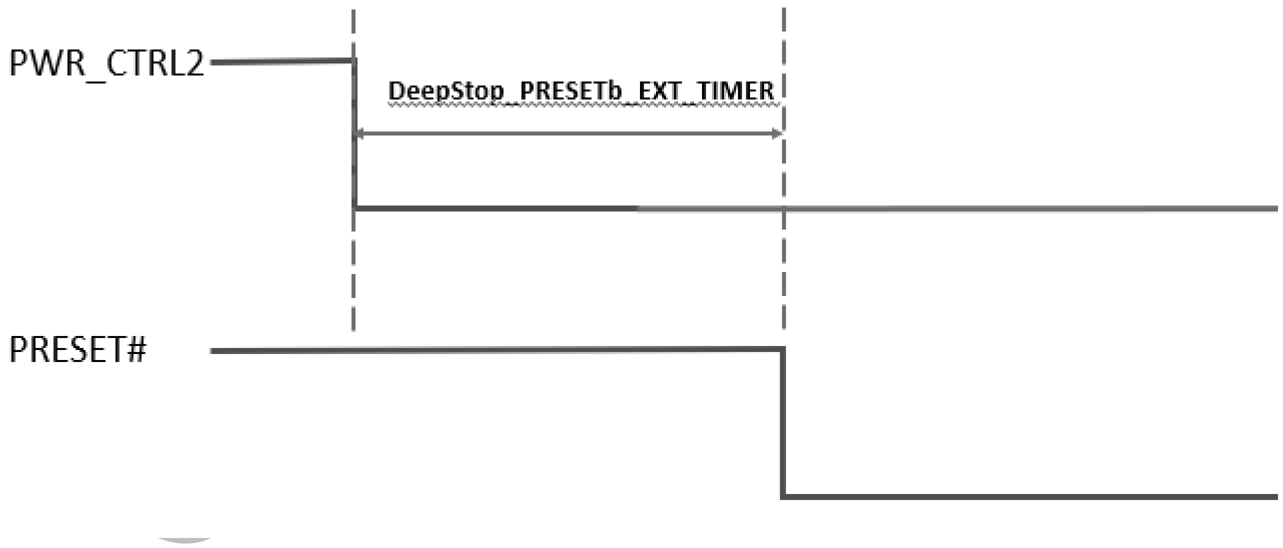
Deep Stop/Low Power Mode:

PRESET# can be configured by the system to be HIGH or LOW during Low Power mode or Suspend-to-RAM modes via **FUSA_CTRL_C[5:5]: PRESETb_DEEPSTOP** register.

Register bit 0x1: PRESET# is HIGH

Register bit 0x0: PRESET# is LOW

FUSA_TIMER_5[1:0]: DeepStop_PRESETb_EXT_TIMER can be set to extend PRESET# output High before shifting to Low at the entrance to Deep Stop/Low Power Mode, if **PRESETb_DEEPSTOP** is set to 0x0.



8.2 Safety Control

a) Operation

This block is the main state machine that performs error response control based on fault detected from PMIC and SoC and brings the PMIC and/or SoC to appropriate safe states.

The state machine transitions the RAA271005 to safe state when a fault is detected by the Fault Detection Logic during startup and while in normal operation.
Table 36 provides the details of the PMIC safe state, the type of faults monitored at each state, and the output conditions of the PMIC at the different safe states.

The following diagram shows the state machine's operation.

Figure 41 State Machine operation

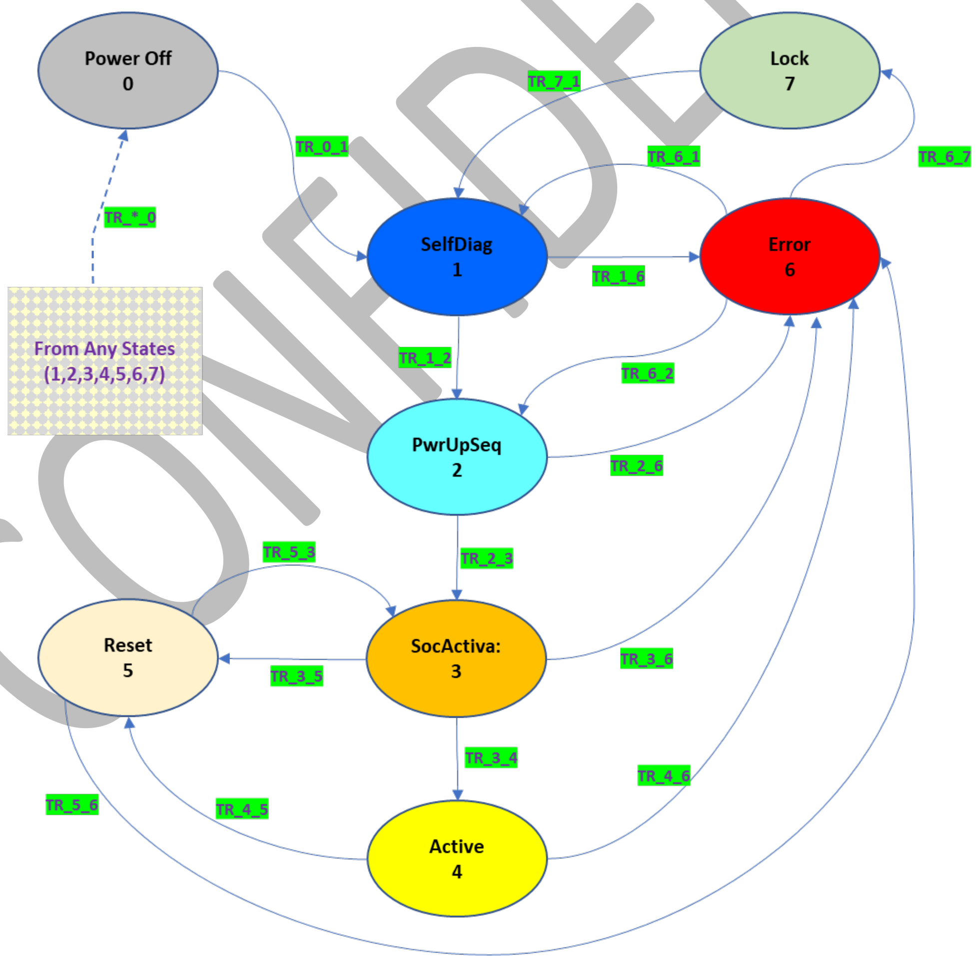


Table 35 State Transition Conditions

TRANSITION	CONDITIONS
TR_*_0	EN = LOW || VDDD = POR_N
TR_0_1	EN = HIGH
TR_1_6	**TIMEOUT_SELFD_ST** TIMEOUT && SELF DIAGNOSIS FAULT DETECTED
TR_1_2	(**TIMEOUT_SELFD_ST** TIMEOUT && NO SELF DIAGNOSIS FAULT DETECTED) || SELF DIAGNOSIS BYPASS[##]
TR_2_6	FORCE_ERROR == 1 || ADC BIST FAULT|| CLOCK FAULT || GC SELF TEST FAULT || REGU FAULT* || PMIC FAULTS DETECTED AT THE END OF PWRUPSEQ
TR_2_3	**Rev C01: TIMEOUT_PUSEQ_ST** TIMEOUT || **PGOOD_DLY** DONE && (NO OT WARN || NO OT SHUTDOWN) **Rev C02: (TIMEOUT_PUSEQ_ST** TIMEOUT || **PGOOD_DLY** DONE)
TR_3_6	FORCE_ERROR == 1 || BUCKS AND LDOS OV/UV || AVIN AND PVIN OV/UV || SPARE SLOTS OV/UV || ADC BIST FAULT|| CLOCK FAULT || GC SELF TEST FAULT || REGU FAULT* || OT FAULT
TR_3_5	FORCE_RESET == 1 || PRESET CHECK FAIL || SINT CHECK FAIL|| EXTERNAL PIN CHK2 FAIL || CVM TEST FAIL || WDT FAIL
TR_3_4	FORCE_ACTIVE == 1 || **SOC_ACTIVATED**
TR_4_6	FORCE_ERROR == 1 || BUCKS AND LDOS OV/UV || AVIN AND PVIN OV/UV || SPARE SLOTS OV/UV || EXTERNAL ADC OV/UV || SDI FAULTS** ADC BIST FAULT|| CLOCK FAULT || GC SELF TEST FAULT || REGU FAULT* || OT FAULT PMIC ERROR COUNT > MAX*** || SOC ERROR COUNT > MAX***
TR_4_5	FORCE_RESET == 1 || WDT FAULT || I2C/SPI CRC FAULT || SDI FAULT[#] || ERROROUT_M# = LOW
TR_5_6	FORCE_ERROR == 1 || BUCKS AND LDOS OV/UV || AVIN AND PVIN OV/UV || SPARE SLOTS OV/UV || EXTERNAL ADC OV/UV || ADC BIST FAULT|| CLOCK FAULT || GC SELF TEST FAULT || REGU FAULT* || OT FAULT **TIMEOUT_PRESETOUT** TIMEOUT ||PMIC ERROR COUNT > MAX*** || SOC ERROR COUNT > MAX*** || PRESET CHECK FAIL || EXT PIN CHK 2(PRESETOUT CHECK) FAULT
TR_5_3	NO PRESETOUT FAULT && **TIMEOUT_PRESETOUT_DLY_TIME** TIMEOUT

TR_6_7	AVIN OV && **ErrorToLock_TR_OPT_A** == 1 && **TIMEOUT_MIN_ERROR_ST** TIMEOUT PMIC ERROR COUNT > max || SoC ERROR COUNT > max
TR_6_1	SELF DIAGNOSIS FAULT DETECTED && **TIMEOUT_MIN_ERROR_ST** TIMEOUT
TR_6_2	FORCE_RESET || FORCE_ACTIVE NO OT WARN && NO OT SHUTDOWN && NO AVIN OV/UV && NO PVIN OV/UV && NO FAULT_REGU* && **TIMEOUT_MIN_ERROR_ST** TIMEOUT
TR_7_1	LOCK release via SW or LOCK release pin

Legend:

* Option provided
** SDI Faults configured to ERROR state transition
*** When masked
# SDI Faults configured to RESET state transition
Not customer configurable

The transitions shown in Table 36 are according to order of priority.

Table 36 RAA271005 Fault Handling

C01:

FAULT SOURCE	Detection State	SAFE STATE	PMIC OUTPUTS	PROTECTION BLOCK	PMIC ERR COUNT	SoC ERR COUNT	IRQ	Notes
ADC MONITORING CHANNELS								
TEMP2 WARNING	P/U SEQ*, SoC ACT, ACTIVE, RESET, ERROR*	N/A	ON	ON			+	*Condition checked at the end of the State. If it persists, PMIC stays in the State.
TEMP2/TEMP4 SHUTDOWN	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
TEMP3 vs TEMP2 DELTA (OT BIST)	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
VBG_REGU OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
PGND OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
Int. LDO_<0-7> OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
ADC4-ADC5 OV/UV	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AVIN2 OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET, ERROR	ERROR*	OFF	ON	+		+	Condition checked at the end of ERROR state
AUX1 OV/UV (Used for EXT ADC CH)	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AVIN1 OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET, ERROR	ERROR*	OFF	ON	+		+	Condition checked at the end of ERROR state
LDOO1 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
LDOO2 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
LDOO3-6 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
PVIN<1-5> OV/UV	SoC ACT, ACTIVE, RESET, ERROR	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
VOUT<1-5> OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AUX2 OV/UV (Used for EXT ADC CH)	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
OTP CRC FAULT	Self_D	ERROR	OFF	ON	+		+	
LBIST FAULT	Self_D	ERROR	OFF	ON	+		+	
CLK MON FAULT	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
REGULATION FAULT (Including OT)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
REGULATION I2C/SPI FAULT	ACTIVE	RESET	ON	ON		+	+	INTb can set during SOC_ACT
PROTECTION I2C/SPI FAULT	ACTIVE	RESET	ON	ON		+	+	INTb can set during SOC_ACT
PMIC_ERR_CNT > MAX	ERROR	LOCK	OFF	OFF				
SoC_ERR_CNT > MAX	RESET	LOCK	OFF	OFF				
WDT FAULT	SoC ACT, ACTIVE	RESET	ON	ON		+	+	
ERROROUT = LOW	ACTIVE	RESET	ON	ON		+	+	
PRESETOUT ACK FAULT	RESET	ERROR	OFF	ON	+		+	
GC_SELFT_FAULT	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
SOC ACTIVATION TESTS								
Serial Interface Check Error	SoC ACT	RESET	ON	ON		+	+	
External Pin (PRESET) Check Error	SoC ACT	ERROR	OFF	ON	+		+	Checked twice (EXTPINCK1, EXTPINCK2)
External Pin (others) Check Error	SoC ACT	RESET	ON	ON		+	+	
CVM Test Error	SoC ACT	RESET	ON	ON		+	+	
WDT Initilization Error	SoC ACT	RESET	ON	ON		+	+	
TEMP2 SHUTDOWN	SoC ACT	ERROR	OFF	ON	+		+	Using SoC Activation-specific threshold

C02:

FAULT SOURCE	Detection State	SAFE STATE	PMIC OUTPUTS	PROTECTION BLOCK	PMIC ERR COUNT	SoC ERR COUNT	IRQ	Notes
ADC MONITORING CHANNELS								
TEMP2 WARNING	P/U SEQ, SoC ACT, ACTIVE, RESET, ERROR*	N/A	N/A	ON			+	*Condition checked at the end of ERROR state. If it persists, PMIC stays in ERROR
TEMP2/TEMP4 SHUTDOWN	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
TEMP3 vs TEMP2 DELTA (OT BIST)	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
VBG_REGU OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
PGND OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
Int. LDO_<0-7> OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
ADC4-ADC5 OV/UV	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AVIN2 OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET, ERROR	ERROR*	OFF	ON	+		+	Condition checked at the end of ERROR state
AUX1 OV/UV (Used for EXT ADC CH)	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AVIN1 OV/UV (ADC BIST)	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET, ERROR	ERROR*	OFF	ON	+		+	Condition checked at the end of ERROR state
LDOO1 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
LDOO2 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
LDOO3-6 OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
PVIN<1-5> OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
VOUT<1-5> OV/UV	SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
AUX2 OV/UV (Used for EXT ADC CH)	ACTIVE, RESET	ERROR	OFF	ON	+		+	INTb can set during SOC_ACT
OTP CRC FAULT	Self_D	ERROR	OFF	ON	+		+	
LBIST FAULT	Self_D	ERROR	OFF	ON	+		+	
CLK MON FAULT	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
REGULATION FAULT (Including OT)	P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
REGULATION I2C/SPI FAULT	ACTIVE	RESET	ON	ON		+	+	INTb can set during SOC_ACT
PROTECTION I2C/SPI FAULT	ACTIVE	RESET	ON	ON		+	+	INTb can set during SOC_ACT
PMIC_ERR_CNT > MAX	ERROR	LOCK	OFF	OFF				
SoC_ERR_CNT > MAX	RESET	LOCK	OFF	OFF				
WDT FAULT	SoC ACT, ACTIVE	RESET	ON	ON		+	+	
ERROROUT = LOW	ACTIVE	RESET	ON	ON		+	+	
PRESETOUT ACK FAULT	RESET	ERROR	OFF	ON	+		+	
GC_SELFT_FAULT	Self_D, P/U SEQ, SoC ACT, ACTIVE, RESET	ERROR	OFF	ON	+		+	
SOC ACTIVATION TESTS								
Serial Interface Check Error	SoC ACT	RESET	ON	ON		+	+	
External Pin (PRESET) Check Error	SoC ACT	ERROR	OFF	ON	+		+	Checked twice (EXTPINCK1, EXTPINCK2)
External Pin (others) Check Error	SoC ACT	RESET	ON	ON		+	+	
CVM Test Error	SoC ACT	RESET	ON	ON		+	+	
WDT Initilization Error	SoC ACT	RESET	ON	ON		+	+	
TEMP2 SHUTDOWN	SoC ACT	ERROR	OFF	ON	+		+	Using SoC Activation-specific threshold

Note:

A “+” in IRQ means the fault reaction can be configured to assert IRQ instead of PRESET#

***** Can be configured to transition to LOCK. If configured to ERROR state, the PMIC ERROR count is incremented at first visit to ERROR state, but does not get incremented if condition still exists at the end of ERROR state timer.

See section 5 for details on PMIC Safe State conditions.

8.3 PMIC ERROR COUNTER

a) Overview

An ERROR state counter gets incremented whenever faults are detected and the RAA271005 is transitioned to ERROR state: See Table 36 for faults that brings the PMIC to ERROR state.

If the maximum allowed Error count is reached, the RAA271005 updates the **FLT_RECORD_B[7:7]:FLT_PmicErrExceed** register and moves to LOCK state. In LOCK state:

- All power rails are discharged immediately.
- SELF DIAGNOSIS test status registers are cleared and system is forced to go through SELF DIAGNOSIS state again once it recovers from LOCK state. See also Appendix E. Self Diagnosis State
- System shall recover from LOCK state through power cycling. See also Appendix G. PMIC Recovery from LOCK State

b) System Considerations

The system integrator shall define the maximum PMIC error count and set the **FUSA_CTRL_B[4:3]: PMIC_ERR_CNT_MAX** register. This shall be set in OTP and cannot be changed during normal operation.

<u>PMIC Reaction to Maximum Error Count</u>

If the maximum allowed Error count is reached, the PMIC reaction is configurable in 2 ways:

(1) PMIC goes to LOCK state

(2) Mask PMIC Error Counter fault reaction

This is done by configuring PMIC IRQ to toggle when the fault condition is detected, and let SoC decide the system reaction:

- Set Protection Register INTERRUPT_MASKING_OPT:IO_MODECTRL[3:3] = 0x0 to configure IRQ when a masked fault is detected.
- FLT_MaskPmicErrExceed :FLT_MASK_B[7:7] = 0x1 to disable reaction for the max error counter fault.
- SoC monitors IRQ, and reads **FLT_RECORD_B[7:7]:FLT_PmicErrExceed** register to identify the fault condition & decides the next course of action.

The effect of disabling the PMIC error counter shall be the responsibility of the system integrator. The effect to the PMIC safety analysis has to be considered.

8.4 SoC RESET Counter

a) Overview

A RESET state counter gets incremented whenever SoC errors are detected and the RAA271005 is transitioned to RESET state: See Tale 36 for faults that brings the PMIC to RESET state.

If the maximum allowed RESET count is reached, the RAA271005 updates the **FLT_RECORD_B[6:6]:FLT_SocErrExceed** moves to LOCK state. In LOCK state:

- All power rails are discharged immediately.
- SELF DIAGNOSIS test status registers are cleared and system is forced to go through SELF DIAGNOSIS state again once it recovers from LOCK state. See also Appendix E. Self Diagnosis State
- System shall recover from LOCK state through power cycling. See also Appendix G. PMIC Recovery from LOCK State

b) System Considerations

The system integrator shall define the maximum SoC ERROR count and set the **FUSA_CTRL_B[6:5]: SOC_ERR_CNT_MAX** register. This shall be set in OTP and cannot be changed during normal operation.

<u>PMIC Reaction to Maximum SoC Error Count</u>

If the maximum allowed SoC Error count is reached, the PMIC reaction is configurable in 2 ways:

(3) PMIC goes to LOCK state

(4) Mask PMIC SoC Error Counter fault reaction

This is done by configuring PMIC IRQ to toggle when the fault condition is detected, and let SoC decide the system reaction:

- Set Protection Register INTERRUPT_MASKING_OPT:IO_MODECTRL[3:3] = 0x0 to configure IRQ when a masked fault is detected.
- **FLT_MaskPmicErrExceed:FLT_MASK_B[6:6**] = 0x1 to disable reaction for the max SoC error counter fault.
- SoC monitors IRQ, and reads **FLT_RECORD_B[6:6]: FLT_SocErrExceed** register to identify the fault condition & decides the next course of action.

The effect of disabling the SoC error counter shall be the responsibility of the system integrator. The effect to the PMIC safety analysis has to be considered.

8.5 Regulation Unit Protection Mechanisms

a) Operation

The Regulation unit self-protection mechanisms, in addition to the safety mechanisms in the Protection unit, are enabled by default.
These mechanisms detect conditions that will cause damage to the regulators, as well as to the supplied circuits. These protection mechanisms tristate or shut-off the regulators in these conditions.

The table below lists these mechanisms.

Table 37 Regulation Protection Mechanisms

PROTECTION MECHANISM	DESCRIPTION
BUCKs OV & UV Comparators	Detects Buck output under voltage & over voltage. The thresholds are set such that the comparators do not trip within the spec range, but effective for detecting hard faults (shorts) & fast transients. OV/UV Programmability: +/-60mV, +/-100mV, +/-150mV, +/-200mV, +/-250mV Sampling time: Vout <1.2V = <2us Vout > 1.2V = 4.5us - Sampling is done continuously as long as Buck is enabled.
BUCKs Over Current & Under Current Comparators (HS & LS)	Detects over current & under current at the inductor due shorts or HS & LS switch failures.
LDOs UV Comparator	Detects LDO output under voltage due to a short to GND.
LDOs Current Limiter	Limits LDO output current.
Temp Sensor 1**, Temp Sensor 4*	Detects junction temperature & shuts down the regulator when shut down limit is reached.

BUCK4 PVIN UV Monitor	Detects PVIN under voltage condition. Buck4 PVIN is monitored to ensure that Buck4 input is at the correct level needed to generate the specified Buck4 output.
AVIN OV & UVLO	Detects AVIN1 over voltage. An internal UVLO detects under voltage condition.

b) System Considerations

All Regulation unit protection mechanisms shall be enabled. This shall be set in OTP.
These provide redundant monitoring to Protection unit monitoring via ADC. PMIC reaction to faults detected by the Regulation unit protection mechanisms are independently configured, and not affected by Protection unit fault masking configuration.

To improve system availability, the following mechanisms can be optional, and the redundant mechanism in the Protection unit takes over the monitoring function.

REGULATION UNIT PROTECTION MECHANISM	PROTECTION UNIT SAFETY MECHANISM
Temp Sensor 4*	TempSensor2
BUCK4 PVIN UV Monitor	PVIN OV/UV Monitoring via ADC

** Temp Sensor 1 has an accuracy of +/- 9 degC and is located in a regulation unit hot spot. It's shutdown limits are set through the following registers, with a default set at 150degC.
*Temp Sensor 4 has an accuracy of +/- 4 degC and is located in a regulation unit hot spot. It's shutdown limits are set through the following registers, with a default set at 150degC.

Table 38 Temp Sensor 4 Temperature Shutdown Threshold Registers

REGISTER	DESCRIPTION
ADCMON_ShutDNLimitMSB_Temp4 [7:0] **ADCMON_ShutDNLimitMSB_Temp4**	Upper byte of the SHUTDOWN threshold applied to Temp Sensor 4
ADCMON_ShutDNLimitLSB_Temp_4[7:0] **ADCMON_ShutDNLimitLSB_Temp_4**	Lower byte of the SHUTDOWN threshold applied Temp Sensor 4

c) Fault Detection and Fault Response During Deep Stop & Suspend-to-RAM (S2R) Modes

During S2R and Deep Stop modes, fault reaction will be handled by regulation block, as Protection block is disabled.

Each Buck / LDO’s response to a regulation fault will be programmed to “hiccup”.
The faulting regulator will attempt to restart automatically for the allowed number of times configured by the system integrator via register **HICCUP_CNT_LIM**.

9 Device Safe State Transition Controlled by the System

9.1 Regulator Disable Control

a) Overview

This is a hardware-supporting-software measure that can used by the system to disable individual, or all regulators, if disabling the PMIC via “ENABLE” pin fails.

The following sequence shall be followed when using this safety mechanism to disable the PMIC.

1. The SoC shall write to the Regulation Unit **IO_MODECTRL_REGU** Register to disable individual buck regulators, and **IO_MODECTRL2_REGU** for individual LDOs.
(When regulators are disabled or enabled via writing to the **IO_MODECTRL_REGU or IO_MODECTRL2_REGU** registers, the correct shutdown, or startup behavior is followed.)

2. The SOC shall confirm the status of the regulator(s) by reading the Regulation Unit **CHIPTSTATE_BUCK_PGOOD** & **CHIPTSTATE_LDO_PGOOD** Register.
Each bit in this register represents the Power state of the regulator.
 0x1 = indicates regulator power is good (soft start is complete, no OV or UV is detected, and regulation OV & UV is not masked)
 0x0 = indicates regulator power is not good or is shutdown.

3. If **CHIPTSTATE_<BUCK/LDO>_PGOOD** status does not conform with **IO_MODECTRL_REGU or IO_MODECTRL2_REGU** command, the system shall write to the Protection Unit **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET** register to force the PMIC to ERROR state. Set **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x2.**

9.2 Forcing Device to RESET or ERROR States

a) Overview

The PMIC can be forced to ERROR or RESET states from ACTIVE state through the following procedures:

9.2.1 Force the whole device to ERROR state in Debug mode:

a) Set PMIC to MTE Mode: **IO_MODECTRL[6:6]:MTE_DIS =0x0**

b) PMIC controls exit from forced state: **FUSA_CTRL_5[2:2]: FORCE_ST_OPT = 0x1**

c) Force PMIC to ERROR state: **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x2.**

The following events take place:

- State machine moves from ACTIVE to ERROR state
- PRESET# is LOW
- SSP follows SSP PIN Value defined in **SDO_0_PIN_VAL_ERROR[5:4]:SOC_PIN_DATA_2** register
- PMIC outputs follow shut down sequence
- PMIC waits for **FUSA_TIMER_1[5:2]**: **TIMEOUT_MIN_ERROR_ST** to expire and then moves to P/U SEQUENCE state if Temp2 < OT_WARN
- If Temp2 < OT_WARN is not achieved, the PMIC stays in ERROR state
- PMIC goes to ACTIVE state regardless of fault situation

9.2.2 Force the whole device to ERROR state in Normal mode:

a) Set PMIC to Normal Mode: **IO_MODECTRL[6:6]:MTE_DIS =0x1**

b) PMIC controls exit from forced state: **FUSA_CTRL_5[2:2]: FORCE_ST_OPT = 0x1**

c) Force PMIC to ERROR state: **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x2.**

The following events take place:

- State machine moves from ACTIVE to ERROR state
- PRESET# is LOW
- SSP follows SSP PIN Value defined in SOC_PIN_DATA registers
- PMIC outputs follow shut down sequence
- PMIC waits for FUSA_TIMER_1[3:2]: TIMEOUT_MIN_ERROR_ST to expire and then moves to P/U SEQUENCE state if Temp2 < OT_WARN

- If Temp2 < OT_WARN is not achieved, PMIC stays in ERROR state
- PMIC goes to ACTIVE state if no faults are detected, otherwise PMIC goes back to ERROR state.

9.2.3 Force the whole device to RESET state in Debug mode:

a) Set PMIC to MTE Mode: **IO_MODECTRL[6:6]:MTE_DIS =0x0**

b) PMIC controls exit from forced state: **FUSA_CTRL_5[2:2]: FORCE_ST_OPT = 0x1**

c) Force PMIC to RESET state: **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x1.**

The following events take place:

- State machine moves from ACTIVE to RESET state
- PRESET#=LOW is applied and PRESETOUT0#=L loop back is checked.
- If PRESETOUT0#=LOW is received within **FUSA_TIMER_1[1:0]:TIMEOUT_PRESETOUT**, system goes to SoC ACTIVATION state after the timer controlled by **FUSA_TIMER_3[2:0]: TIMEOUT_PRESETOUT_DLY_TIME** expires.
- If RESET loopback fails, RAA271005 moves to ERROR state.
- During this state, all monitoring are still applied

9.2.4 Force the whole device to RESET state in Normal mode:

a) Set PMIC to Normal Mode: **IO_MODECTRL[6:6]:MTE_DIS =0x1**

b) PMIC controls exit from forced state: **FUSA_CTRL_5[2:2]: FORCE_ST_OPT = 0x1**

c) Force PMIC to RESET state: **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x1.**

The following events take place:

- State machine moves from ACTIVE to RESET state
- PRESET#=LOW is applied and PRESETOUT0#=L loop back is checked.
- If PRESETOUT0#=LOW is received within **FUSA_TIMER_1[1:0]:TIMEOUT_PRESETOUT**, system goes to SoC ACTIVATION state after the timer controlled by **FUSA_TIMER_3[2:0]: TIMEOUT_PRESETOUT_DLY_TIME** expires.
- If RESET loopback fails, RAA271005 moves to ERROR state.
- During this state, all monitoring are still applied

When **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET** register is used to force the PMIC to either ERROR or RESET states, the error counters do not get incremented.

b) System Considerations

When the FORCE RESET or FORCE ERROR operation is finished, the PMIC will be in ACTIVE state. Before PMIC goes to ACTIVE state, the PMIC proceeds to SoC Activation Sequence, in both Debug mode & Normal Mode.

However, in MTE Mode, the PMIC goes immediately to ACTIVE state.
In Normal Mode, the PMIC will perform SoC Activation and then goes to ACTIVE state if no faults are detected during SoC Activation Sequence.

If the system needs to force the device to ERROR or RESET state multiple times, **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET = 0x0** shall be set in ACTIVE state, to reset the forced state condition.

10 Safety Dependent Outputs (SDOs)

10.1 Used as Secondary Disable Signals:

a) Overview

In the PMIC safety concept, these pins disable system communication if safety mechanisms determine that the SoC's or PMIC's condition is unsafe, or they cannot be brought to safe state. These are state dependent outputs of the RAA271005. The state of the pins is configurable as High, Low, or Hi-Z for most of the RAA271005 states.

Table 39 Secondary Safety Path Configurability

IO STATE	SDO
SELF-DIAGNOSIS	Hi/Lo/Tri
PU SEQ	Hi/Lo/Tri
SoC ACTIVATION	Hi/Lo/Tri
System Test	Hi/Lo/Tri
ACTIVE	Hi/Lo/Tri
RESET	Hi/Lo/Tri
ERROR	Hi/Lo/Tri
LOCK	Hi/Lo/Tri
Debug Mode	Hi/Lo/Tri
Deep Stop*	Tri (not configurable)
Suspend To Ram*	Tri (not configurable)

***** *See System Considerations*

Table 40 The registers that configure the SDOs

Registers	Description
SOC_PIN_DATA_1, SOC_PIN_DATA_2	Configures SDO1 pin value in different PMIC states
SOC_PIN_DATA_3, SOC_PIN_DATA_4	Configures SDO2 pin value in different PMIC states

b) System Considerations

The RAA271005 SDO configuration and safe state is configurable by the system integrator.

The system integrator shall implement system measures to avoid floating conditions either due to pin failure, to ensure that the System safe state is maintained; or during Deep Stop and suspend-to-RAM modes (if applicable), to ensure that current consumption is within targets.

11 Device Fault Information for System Level Use

11.1 PMIC Interrupt Pin: IRQ

a) Overview

IRQ is an active low, push-pull output signal that indicates fault events that have been blocked from generating a fault reaction by the PMIC.

Fault information can be read from **FLT_RECORD__<*id*>** ; where *id* is a fault type identifier (ex. LDO, BUCKx, TEMP, etc), registers (See Section 11.2), to identify the fault condition that triggered IRQ.

IRQ will be released only if all fault report registers are cleared by the user.
Regulation block **FLT_RECORD_<*id*>** registers will be cleared by writing 0 to the registers. Protection block **FLT_RECORD_<*id*>** registers will be cleared on read.

NOTE: If a fault occurred on a non-masked monitored channel, IRQ is always generated.

b) System Considerations

IRQ is a PMIC feature that can be used by the system to address availability requirements.
Unless configured, all detected faults by the PMIC safety mechanisms trigger PRESET#.

To configure the IRQ pin to react to specific detected fault by a PMIC safety mechanism, the following settings have to be done:

(1) Set Protection Register **IO_MODECTRL[3:3]: INTERRUPT_MASKING_OPT** to configure IRQ when a masked fault is detected.

3	INTERRUPT_MASKING_OPT	RW	Interrupt Pin Masking Option 0x1 : fault record of only unmasked faults will propagate to INTb pin. (fault reaction not masked) 0x0 : fault record of all faults (both masked and unmasked) will propagate to INTb pin

(2) Set the desired field(s) in the any following Protection registers to **0x1** to block the applicable fault event from generating a fault reaction:
FLT_MASK_B[7:0] *(See protection register address 0x12C in Register Map)*

ADCMON_MASK_<*meas*> ; where *meas* stands for any measured ADC channel
(See protection register addresses 0x12D to 0x133 in Register Map)

(3) To configure IRQ to react to faults detected by individual comparators in the regulators
Set the desired field(s) in any of the following Protection registers to 0x1, to block the applicable fault event from generating a fault reaction:
ADCMON_MASK_BUCKs_A[7:0] (See register address 0x132 in Register Map)
ADCMON_MASK_BUCKs_B[1:0] (See register address 0x131 in Register Map)
ADCMON_MASK_ExtLDOs[7:0] (See register address 0x130 in Register Map)

Set the following Regulation registers so that the applicable faults are not masked, not ignored, and not "do nothing". *(See regulation register addresses 0x129 to 0x13E in Datasheet)*
FLT_CTRL_<id>
FLT_CTRL1_<id>
FLT_CTRL2_<id>
FLT_CTRL1
FLT_CTRL1A
FLT_CTRL2
FLT_CTRL2A

However, this setup has to be done with careful consideration since individual comparators in the regulators are implemented to protect the regulators from damage.
The system integrator shall be responsible for the classification of non-fatal PMIC faults that do not control a fault reaction, according to their system requirements. If fault reaction via PRESET# is blocked, the SoC shall judge the impact of the fault and decide when to transition the system to safe state.

The SoC can force the PMIC to ERROR or RESET state via the **FUSA_CTRL_5[1:0]: FORCE_ST_ERROR_RESET** register. (See section 9.2).
Setting this register to 0x1, forces the PMIC to RESET state.
Setting this register to 0x2, forces the PMIC to ERROR state
The impact of blocking a fault reaction, to the FMEDA, shall also be considered.
A pull down resistor at the IRQ output shall be used to avoid floating conditions due to pin failure.
To clear the fault that triggered IRQ, and restore IRQ to fault-free condition, the SoC shall write 0x0 to the **FLT_RECORD*** register to clear the fault in Regulation.
The SoC shall read **FLT_RECORD*** register in Protection to clear the fault in Protection.

11.2 Fault Status Registers for Each Fault Condition

a) Overview

The following tables lists the RAA271005 Fault Registers accessible to the SoC for reading.

Table 41 Regulation unit Fault Registers:

Fields / bits	**Register**	**Description**
FLT_INVALID_DEEPSTOP	FLT_RECORD_TEMP	Invalid Deep Stop detected. An invalid deep stop is a condition when a deep stop entry command fails because PGOOD signal is low.
FLT_TEMPSDR	FLT_RECORD_TEMP	Over Temperature (OT) Shutdown (rising threshold)
FLT_TEMPSDF	FLT_RECORD_TEMP	Over Temperature (OT) Shutdown (falling edge) (Shutdown – Hysteresis)
FLT_BUCK1_DISC_DET	FLT_RECORD_BUCK1	Discharge Detect error for BUCK1
FLT_BUCK1_LSWUC	FLT_RECORD_BUCK1	LS Way Under Current (WOC) for BUCK1
FLT_BUCK1_LSWOC	FLT_RECORD_BUCK1	LS Way Over Current (WOC) for BUCK1
FLT_BUCK1_HSWOC	FLT_RECORD_BUCK1	HS Way Over Current (WOC) for BUCK1
FLT_BUCK1_OV	FLT_RECORD_BUCK1	Over Voltage (OV) for BUCK1
FLT_BUCK1_UV	FLT_RECORD_BUCK1	Under Voltage (UV) for BUCK1
FLT_BUCK2_DISC_DET	FLT_RECORD_BUCK2	Discharge Detect error for BUCK2

Fields / bits	**Register**	**Description**
FLT_BUCK2_LSWUC	FLT_RECORD_BUCK2	LS Way Under Current (WOC) for BUCK2
FLT_BUCK2_LSWOC	FLT_RECORD_BUCK2	LS Way Over Current (WOC) for BUCK2
FLT_BUCK2_HSWOC	FLT_RECORD_BUCK2	HS Way Over Current (WOC) for BUCK2
FLT_BUCK2_OV	FLT_RECORD_BUCK2	Over Voltage (OV) for BUCK2
FLT_BUCK2_UV	FLT_RECORD_BUCK2	Under Voltage (UV) for BUCK2
FLT_BUCK3_DISC_DET	FLT_RECORD_BUCK3	Discharge Detect error for BUCK3
FLT_BUCK3_LSWUC	FLT_RECORD_BUCK3	LS Way Under Current (WOC) for BUCK3
FLT_BUCK3_LSWOC	FLT_RECORD_BUCK3	LS Way Over Current (WOC) for BUCK3
FLT_BUCK3_HSWOC	FLT_RECORD_BUCK3	HS Way Over Current (WOC) for BUCK3
FLT_BUCK3_OV	FLT_RECORD_BUCK3	Over Voltage (OV) for BUCK3
FLT_BUCK3_UV	FLT_RECORD_BUCK3	Under Voltage (UV) for BUCK3
FLT_BUCK4_DISC_DET	FLT_RECORD_BUCK4	Discharge Detect error for BUCK4
FLT_BUCK4_PVIN4_OK	FLT_RECORD_BUCK4	PVIN4_OK for BUCK4
FLT_BUCK4_LSWUC	FLT_RECORD_BUCK4	LS Way Under Current (WOC) for BUCK4
FLT_BUCK4_LSWOC	FLT_RECORD_BUCK4	LS Way Over Current (WOC) for BUCK4
FLT_BUCK4_HSWOC	FLT_RECORD_BUCK4	Way Over Current (WOC) for BUCK4

Fields / bits	Register	Description
FLT_BUCK4_OV	FLT_RECORD_BUCK4	Over Voltage (OV) for BUCK4
FLT_BUCK4_UV	FLT_RECORD_BUCK4	Under Voltage (UV) for BUCK4
FLT_BUCK5_DISC_DET	FLT_RECORD_BUCK5	Discharge Detect error for BUCK5
FLT_BUCK5_LSWUC	FLT_RECORD_BUCK5	LS Way Under Current (WOC) for BUCK5
FLT_BUCK5_LSWOC	FLT_RECORD_BUCK5	LS Way Over Current (WOC) for BUCK5
FLT_BUCK5_HSWOC	FLT_RECORD_BUCK5	Way Over Current (WOC) for BUCK5
FLT_BUCK5_OV	FLT_RECORD_BUCK5	Over Voltage (OV) for BUCK5
FLT_BUCK5_UV	FLT_RECORD_BUCK5	Under Voltage (UV) for BUCK5
FLT_LDO6_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO6
FLT_LDO5_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO5
FLT_LDO4_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO4
FLT_LDO3_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO3
FLT_LDO2_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO2
FLT_LDO1_UV	FLT_RECORD_LDO	Under Voltage (UV) for LDO1
FLT_VBAT_OV	FLT_RECORD_IF	VBAT Over voltage
FLT_SPI	FLT_RECORD_IF	SPI CRC error
FLT_I2C	FLT_RECORD_IF	I2C CRC error
FLT_LDO6_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error
FLT_LDO5_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error
FLT_LDO4_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error

Fields / bits	Register	Description
FLT_LDO3_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error
FLT_LDO2_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error
FLT_LDO1_DISC_DET	FLT_RECORD_LDO_DISC_DET	Discharge Detect error

Regulation faults that trigger Protection FLT_ReguOT

1. FLT_INVALID_DEEPSTOP
2. FLT_TEMPSDR
3. FLT_TEMPSDF
4. FLT_BUCK<1to5>_DISC_DET
5. FLT_BUCK<1to5>_LSWUC
6. FLT_BUCK<1to5>_LSWOC
7. FLT_BUCK<1to5>_HSWOC
8. FLT_BUCK<1to5>_OV
9. FLT_BUCK<1to5>_UV
10. FLT_BUCK4_PVIN4_OK
11. FLT_LDO<1to6>_UV
12. FLT_VBAT_OV
13. FLT_SPI
14. FLT_I2C
15. FLT_LDO<1to6>_DISC_DET
16. OTP faults

Table 42 Protection Unit Fault Registers

Fields / bits	Register	Description
FLT_ExtPinCheck2	FLT_RECORD_A	External Pin Check2 test fail
FLT_SintCheck	FLT_RECORD_A	Serial Interface Check fail
FLT_Presetout	FLT_RECORD_A	PRESET/PRESETOUT0 test fail
FLT_GC_SelfT_fault	FLT_RECORD_A	GC Self Test Fail
FLT_SDI_3 (SDI4)	FLT_RECORD_A	Fault detected on SDI4
FLT_SDI_2 (VMONOUT1)	FLT_RECORD_A	Fault detected on SDI3
FLT_SDI_1 (VMONOUT0/CVM_OUT)	FLT_RECORD_A	Fault detected on SDI2
FLT_SDI_0 (ERROROUT)	FLT_RECORD_A	Fault detected on SDI1
FLT_PMICErrExceed	FLT_RECORD_B	Maximum PMIC Error Counter fault
FLT_SocErrExceed	FLT_RECORD_B	Maximum SOC error counter fault
FLT_WDT	FLT_RECORD_B	WDT operation fail
FLT_RegCRC	FLT_RECORD_B	CRC fault for I2C/SPI communication
FLT_ClkMon	FLT_RECORD_B	Clock monitoring fault
FLT_ReguOT	FLT_RECORD_B	Regulation unit fault
FLT_FaultDetectBist	FLT_RECORD_B	Fault Detect BIST fail
FLT_LBIST	FLT_RECORD_B	LBIST result fail
FaultStatus_AVIN2_Prot	FLT_RECORD_GND_AVIN	AVIN2 power supply (Protection) out of range
FaultStatus_AVIN1_Regu	FLT_RECORD_GND_AVIN	AVIN1 power supply (Regulation) out of range

Fields / bits	Register	Description
FaultStatus_PGND_Regu	FLT_RECORD_GND_AVIN	Monitoring result for PGND (Regulation) out of range
FaultStatus_Temp_ExtreemLow	FLT_RECORD_BG_Temp	Temperature sensor extreme low status
FaultStatus_TEMP2_SENSOR	FLT_RECORD_BG_Temp	Temperature sensor health status
FaultStatus_TempShdn	FLT_RECORD_BG_Temp	Temp Shutdown status
FaultStatus_TempWarn	FLT_RECORD_BG_Temp	Temp warning status
FaultStatus_Temp4_Shdn_Regu	FLT_RECORD_BG_Temp	Temp sensor 4 shutdown status
FaultStatus_BG_REGU	FLT_RECORD_BG_Temp	Regulation BG status
FaultStatus_ExtLDORegu_5	FLT_RECORD_ExtLDOs	LDO 6 Status
FaultStatus_ExtLDORegu_4	FLT_RECORD_ExtLDOs	LDO 5 Status
FaultStatus_ExtLDORegu_3	FLT_RECORD_ExtLDOs	LDO 4 Status
FaultStatus_ExtLDORegu_2	FLT_RECORD_ExtLDOs	LDO 3 Status
FaultStatus_ExtLDORegu_1	FLT_RECORD_ExtLDOs	LDO 2 Status
FaultStatus_ExtLDORegu_0	FLT_RECORD_ExtLDOs	LDO 1 Status
FaultStatus_Buck_1_VOUT	FLT_RECORD_BUCKS_B	Buck1 Status
FaultStatus_Buck_1_PVIN	FLT_RECORD_BUCKS_B	PVIN 1 Status
FaultStatus_Buck_5_VOUT	FLT_RECORD_BUCKS_A	Buck5 Status
FaultStatus_Buck_5_PVIN	FLT_RECORD_BUCKS_A	PVIN 5 Status
FaultStatus_Buck_4_VOUT	FLT_RECORD_BUCKS_A	Buck4 Status
FaultStatus_Buck_4_PVIN	FLT_RECORD_BUCKS_A	PVIN 4 Status
FaultStatus_Buck_3_VOUT	FLT_RECORD_BUCKS_A	Buck3 Status
FaultStatus_Buck_3_PVIN	FLT_RECORD_BUCKS_A	PVIN 3 Status

Fields / bits	Register	Description
FaultStatus_Buck_2_VOUT	FLT_RECORD_BUCKS_A	Buck2 Status
FaultStatus_Buck_2_PVIN	FLT_RECORD_BUCKS_A	PVIN 2 Status

If details on other internal & external monitored signals (32 internal & 8 external) are desired, Fault Records for each monitored signal is available. See details in the FAULT RECORD registers (Register addresses 0x019 to 0x021 in* ***Register Map**.)*

b) System Considerations

Information from fault registers is used for debug purposes, in the PMIC safety concept. (See also Section 8.3 b).
However, the system can use this information, together with configuring IRQ (See section 11.1), to improve system availability.

The system integrator shall read the status of the fault registers, at the end of the SoC Activation Sequence. This shall be done at cold start and after each recovery from RESET or ERROR states, in order to confirm fault free status going into ACTIVE state, and clear faults that generated the RESET or ERROR condition.

All fault bits in the Protection block are cleared upon READ and remains cleared, until a fault is detected. All fault bits in the Regulation block are cleared upon WRITE of “0” and remains cleared, until a fault is detected.

Diagnosis:
For debug purposes, it is not necessary to mask or deactivate the fault-reaction (e.g. shutdown of rails, assert PRESET) to maintain diagnosability. Fault information is still available in the fault registers (FLT_RECORD_*) after recovery from ERROR & RESET states.

As long as PMIC Enable is not pulled low, AVIN > UVLO, VIO is available, & I2C/SPI is available, PMIC fault registers can be read after recovery from ERROR & RESET states.

12 System Fault Information for Device Level Use

12.1 Safety Dependent Inputs (SDI)

a) Overview

The RAA271005 has 4 Safety Dependent Inputs. These pins are used to monitor error notification signals from the system. The PMIC reaction to the individual error signal can be configured through **FUSA_CTRL_E[7:4]** register. The reaction can be configured to either transition the PMIC to RESET state or ERROR state.

0x11E - FUSA_CTRL_E (OTP)

	NAME	R/W	DESCRIPTION
7:4	SDI_FltReAct	RW	Determine state where to go when SDI fault in ACTIVE state SDI_FltReAct[3] : If '1', State machine moves to Error state when SDI[3] fault. If '0', State machine moves to Reset state when SDI[3] fault. SDI_FltReAct[2] : If '1', State machine moves to Error state when SDI[2] fault. If '0', State machine moves to Reset state when SDI[2] fault. SDI_FltReAct[1] : If '1', State machine moves to Error state when SDI[1] fault. If '0', State machine moves to Reset state when SDI[1] fault. SDI_FltReAct[0] : If '1', State machine moves to Error state when SDI[0] fault. If '0', State machine moves to Reset state when SDI[0] fault.

Referring to the RAA271005 use cases, IO7 (SDI1) is allocated to the SoC Error signal **ERROROUT***. When the **ERROROUT*** is asserted by the SoC, the RAA271005 brings the SoC to safe state by de-asserting **PRESET#**. The PMIC waits for the confirmation from the SoC that the reset has been received, via the **PRESETOUT*** pin. When **PRESETOUT*** is received, the PMIC starts the SoC Activation sequence.

IO12 (SDI2) is allocated to the Application Domian CVM Error signal **VMONOUT0#**(S4) or **CVM_OUT**(V4H).
IO15 (SDI3) is allocated to the Control Domain CVM Error signal **VMONOUT1#**(S4).
IO16 (SDI4) is an additional safety dependent input which is allocated to the fault output signal of the pre-regulator in the RAA271005 use cases.

b) System Considerations

Important Assumption: All SoC Error Output signals are gathered to **ERROROUT***.

13 Other Recommended System Measures

13.1 Additional Tests Recommended at System - Level

1. Tests for shorts of all pins to PVIN, AVIN & GND
2. Testing of External ADC Channels ADC3, ADC4, ADC5, and any other ADC channels used for safety-related functions.
3. If ExtAdcIO_AuxMODE is set to use an external mux for supporting additional external ADC channels, testing of the external mux shall be done at startup. See also Appendix H.
4. Watchdog Test

13.2 SPI/I2C End To End Protection

To achieve high diagnostic coverage for the SPI/I2C End to End Protection, the RAA271005 performs CRC on all SPI/I2C transactions (See sections: 7.7 and 7.9).

Additional coverage is achieved by system-level measures including:

1. The SoC shall monitor the SPI/I2C Message Counters (See sections: 7.8 and 7.10)
2. The SoC shall perform a **Write – Read – Verify** on every WRITE transaction
3. The SoC shall apply a **Timeout Monitor** for each transaction

13.3 System Measures for Transient Faults due to Small Possibility of Oscillation

To detect potential for transient or repetitive transients generated by operation of the loading device (SoC), Power-On Self Test including a load transient stress is recommended. Successful completion of the POST detects systematic and latent faults that could lead to transient voltages outside of the monitoring threshold.

Other safety mechanisms in the SoC, such as CVM, Clock Monitor, ECC, etc., provide detection coverage depending on use case. ASIL D systems are recommended to include a temporally offset lock-step operation to detect transient failures.

14 Safety Considerations for Specific RAA271005 Modes

14.1 Debug (MTE) Mode

Debug Mode is a special mode for flashing in the plant. During flashing, communication in the system is possible and the SoC is not in RESET state. Connection to the ECU is possible only at the vehicle connector.
RAA271005 is in Debug mode by default.
In Debug mode, the RAA271005 is forced to stay in ACTIVE state (PRESET# is de-asserted) and any fault detected by the Protection unit safety mechanisms during this state will not create any fault reaction.

Table 43 shows the register settings of the MTE register while Debug mode is enabled. The **MTE_DIS** bit is used to exit & re-enter Debug mode, as needed, during system development.

Table 43 MTE Register Settings in MTE/Debug Mode

Address	Register Name	Bit Field	Debug Mode Setting
0x102	IO_MODECTRL	MTE_TM	0x0: MTE Test mode disable*
	IO_MODECTRL	MTE_DIS	0x0: MTE mode is available
	IO_MODECTRL	MTE_PGM_DUR	0x1: MTE program duration is 16 cycle with 32K clock
	IO_MODECTRL	MTE_POST_BYP	0x1: Self-Diagnosis tests are bypassed in the MTE mode.
0x094	MTE_CFG_CTRL_A	MTE_BURN	0x1: Burn MTE Fuses after Flashing

** Used for RAA271005 Debug*

IO_MODECTRL Register Details

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Field Name	MTE_TM	MTE_DIS	MTE_PGM_DUR	MTE_POST_BYP	INTERRUPT_MASKING_OPT	IO_ENVPPPULLDOWN	-	IO_REGVALID
Bit Field value	0	0	1	1	x	x	x	X

Once the system has established that there is no need to visit Debug mode again, Debug mode can be disabled permanently.

IMPORTANT: The SoC shall send a command to the RAA271005 to leave Debug Mode.
This is done by the SoC writing “1” to register MTE_CFG_CTRL_A[0]: MTE_BURN, which will blow the 2 poly fuses in the RAA271005 to take it out of Debug mode. Successful blowing of either one of the fuses is enough to disable Debug mode. Once RAA271005 is out of Debug mode, all masks will revert to OTP defaults and RAA271005 will function as in Normal mode.

NOTE: Regulation faults reaction are not included and should be configured separately.

After this is done, the RAA271005 is not able to enter the Debug Mode again in its complete lifecycle.

MTE_CFG_CTRL_A* Register Details

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Field Name	-	-	-	-	-	-	-	MTE_BURN
Bit Field value	x	x	x	x	x	x	x	1

*To burn poly fuses after Flashing

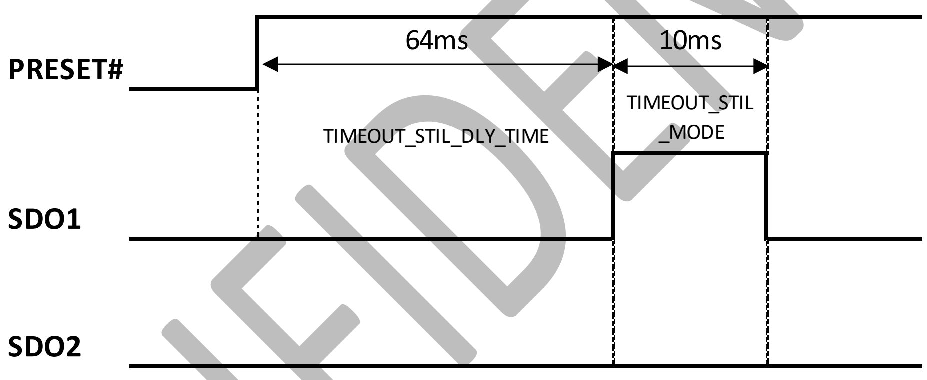
The SoC shall confirm the PMIC is out of Debug Mode by reading register:
FUSA_STATUS_3[0]:MTE_MODE_STATUS = 0x0 after 1ms.

14.2 System Test (STIL) Mode

It is defined as part of the SoC Activation state and aimed to be used by the system integrator for system level tests of CAN communication. During the STIL mode, logic levels of SDO signals will switch to STIL mode values. These values should be selected such that it enables the CAN communication. Once STIL mode ends, these signals will revert back to SoC Activation logic levels.

Entry delay to STIL mode within the SoC Activation state (from the release of PRESET#) is controlled by **TIMEOUT_STIL_DLY_TIME[2:0]** register. This delay has to be chosen long enough to allow all RAA271005 SoC Activation tests to finish (see Section 7.15). The duration of STIL mode is controlled by **TIMEOUT_STIL_MODE[1:0]** register.

Figure 42 Example STIL Mode Setup



If **FUSA_CTRL_1[2:2]:STIL_MODE_EN** is = 0x01, SDO pins will have the STIL values defined in **"SDO_0_PIN_VAL_SACTIVASTIL**", and **"SDO_1_PIN_VAL_SACTIVASTIL"** registers.

All FuSa features are active during this mode.

Table 44 Summary of STIL Mode Registers (as shown in example in Figure 42)

Register Address	Register Name	Field Name	Setting
0x10C	FUSA_CTRL_1	STIL_MODE_EN	0x1: SDO1, SDO2 will have STIL values
0x11F	SOC_PIN_DATA_1	SDO_0_PIN_VAL_SACTIVASTIL	0x1: SDO1 is “High” in STIL Mode
0x121	SOC_PIN_DATA_3	SDO_1_PIN_VAL_SACTIVASTIL	0x0: SDO2 is “Low” in STIL Mode
0x118	FUSA_TIMER_3	TIMEOUT_STIL_DLY_TIME	0x7: 64ms delay before SDO1, SDO2 STIL Mode values apply
0x117	FUSA_TIMER_2	TIMEOUT_STIL_MODE	0x1: RAA271005 stays in STIL mode for 10ms

15 Miscellaneous

15.1 Important Regulation Register settings for Safety considerations:

1. FLT_CTRL2 = 0x7C, and FLT_CTRL2A = 0x3F which ensures that the regulators are disabled when a fault condition is detected.

Appendix A

Has been moved to Datasheet

Appendix B

Has been moved to Datasheet

Appendix C. Masking Monitoring Faults

The PMIC has registers that allow the system integrator to mask the reaction of faults detected by Monitoring Safety Mechanism, these include the over voltage and under voltage faults. Masking can only be programmed in OTP. Once programmed, it cannot be changed in normal operation.

If any regulation faults are masked from generating a fault reaction (PRESET = Low), and IRQ is used to inform the system instead, **the system shall read regulation Fault registers at least once per FTTI.**

IMPORTANT: All monitoring in the Regulation and Protection units, and the fault reactions, are enabled in the PMIC safety concept. The effect of masking any PMIC fault reaction is the responsibility of the system integrator. The effect to the PMIC safety analysis has to be considered.

Appendix D. Reading ADC Sampled Value

When reading ADC values directly from the channels register, it is not guaranteed that the data content of the register is valid data at the time of the READ. Reading the ADC values using ADCMON_COPY_SAMPLE ensures that the valid ADC data is ready for reading. This is indicated by the ADCMON_COPY_SAMPLE_RDY bit.

When ADC conversion is run, data is updated into 2 8-bit registers: an MSB & an LSB register. The LSB register is updated first, then the MSB register. The user can end up reading at the time when the MSB register has not been updated, but the LSB register has been updated. If IIR = 4 or higher, it is unlikely that the MSB register will change much, so this has minimal impact.

The scenario where this will create an error is:
If MSB =0F and LSB =FF, & LSB has to transition to 00 & MSB has to be incremented by 1.

If using the copy-sample approach, there is no need to poll the copy sample register. The ADCMON_COPY_SAMPLE_RDY bit sets when data has been updated to both MSB & LSB registers. The user can use a fixed delay to check that the ADCMON_COPY_SAMPLE_RDY bit is set & read the COPY SAMPLE register.

Example: An I2C communication (1MHz) would mean a clock period of 1uS. Writing to COPY SAMPLE register would take ~32uS. The next read cycle to read COPY_SAMPLE_RDY would take another ~40uS. If the two commands are sent back to back then you would see COPY_SAMPLE_RDY bit set. No fixed delay is needed for I2C.

For SPI communication (24MHz), the clock period is 42nS. Writing to COPY SAMPLE register would take 1.34uS. With SPI communication, a delay between writing to COPY SAMPLE register and reading back COPY_SAMPLE_RDY bit should be at least 2uS apart. This would ensure that COPY_SAMPLE_RDY bit is set in the first read itself.

Below are the steps to read the filtered ADC sampled value. The sampled value is the final data with gain & filtering applied:

1. Send a request to read the ADC sample by setting

   ADCMON_COPY_SAMPLE[5:0]:ADCMON_COPY_SAMPLE to the channel of interest:

5:0	ADCMON_COPY_SAMPLE	RW	ADC_COPY_SAMPLE[5:0] Index 0x00: CH_INT_GND_PROT 0x01: CH_INT_TEMP_2 0x02: CH_INT_TEMP_3 0x03: CH_INT_BG_REGU 0x04: CH_INT_GND_REGU 0x05: CH_INT_LDO_REGU_1 0x06: CH_INT_LDO_REGU_2 0x07: CH_INT_LDO_REGU_3 0x08: CH_INT_LDO_REGU_4 0x09: CH_INT_LDO_REGU_5 0x0a: CH_INT_LDO_REGU_6 0x0b: CH_INT_LDO_PROT_1 0x0c: CH_INT_LDO_PROT_2 0x0d: CH_RSV_SPARE 0x0e: CH_PIN_AVIN_2 0x0f: CH_PIN_AUX_1 0x10: CH_PIN_AVIN_1 0x11: CH_PIN_LDO_OUT_1 0x12: CH_PIN_LDO_OUT_2 0x13: CH_PIN_LDO_OUT_3 0x14: CH_PIN_LDO_OUT_4 0x15: CH_PIN_PVIN_1 0x16: CH_PIN_VOUT_1 0x17: CH_PIN_PVIN_2 0x18: CH_PIN_VOUT_2 0x19: CH_PIN_PVIN_3 0x1a: CH_PIN_VOUT_3 0x1b: CH_PIN_PVIN_4 0x1c: CH_PIN_VOUT_4 0x1d: CH_PIN_PVIN_5

			0x1e: CH_PIN_VOUT_5 0x1f: CH_PIN_AUX_2 Note: When ADC_DATACOPY is written, the sample data from the channel that selected by ADC_COPY_SAMPLE will be copied to ADC_DATA

2. Read **ADCMON_COPY_SAMPLE[7:7]: ADCMON_COPY_SAMPLE_RDY** register

   0x1: means the copied sample is Ready

3. Read **ADCMON_COPY_SAMPLE[7:0]: ADCMON_COPY_DATA_MSB**

   & **ADCMON_COPY_SAMPLE[7:0]: ADCMON_COPY_DATA_LSB** for the upper byte & the lower byte of the requested ADC channel data respectively.

When data from an External ADC channel needs to be sampled:

1. Send a request to read the ADC sample by setting

   ADCMON_COPY_SAMPLE[6]:ADCMON_COPY_EXTADC = 0x1

   ADCMON_COPY_SAMPLE[5:0]:ADCMON_COPY_SAMPLE to the channel of interest:

   0x0 = External ADC Channel 1
   0x1 = External ADC Channel 2
   0x2 = External ADC Channel 3
   0x3 = External ADC Channel 4
   0x4 = External ADC Channel 5
   0x5 = External ADC Channel 6
   0x6 = External ADC Channel 7
   0x7 = External ADC Channel 8

COPY_SAMPLE_RDY bit sets after ~2uS, if request is made during conversion. This bit is set within 100nS if request comes during conversion of some other channel.
The COPY_SAMPLE_RDY bit stays high until another COPY_SAMPLE request is made.
COPY_SAMPLE data registers will not update until a new request is made.

An I2C communication (1MHz) would mean a clock period of 1uS. Writing to COPY SAMPLE register would take ~32uS. The next read cycle to read COPY_SAMPLE_RDY would take another ~40uS. If the two commands are sent back to back then you would see COPY_SAMPLE_RDY bit set. No fixed delay is needed for I2C.

For SPI communication (24MHz), the clock period is 42nS. Writing to COPY SAMPLE register would take 1.34uS. With SPI communication, a delay between writing to COPY SAMPLE register and reading back COPY_SAMPLE_RDY bit should be at least 2uS apart. This would ensure that COPY_SAMPLE_RDY bit is set in the first read itself.

Sample Code:

```
//Declare adc return type
typedef enum adc_data_flags{
    D_DATA_NOT_READY,
    D_CHANNEL_OUT_OF_BOUNDS,
    D_DATA_FRESH
}pmic_adc_return_t;

pmic_adc_return_t pmic_read_adc(uint8_t adc_ch_select, uint16_t * adc_data){

    //Declare Local Variables
    uint8_t comms_data_tmp = 0x00;
    uint8_t comms_return_tmp = 0x00;
    uint16_t data = -1;

    //Set the base status for the return type
    pmic_adc_return_t adc_data_status = D_CHANNEL_OUT_OF_BOUNDS;


    // Check to see if the channel is in bounds

    if (adc_ch_select <= MAX_ADC_COPY_SAMPLE_REGISTER){

        //Select The Channel because we know it's in bounds
        comms_return_tmp = r_pmic_i2c_write(D_PMIC_I2C_PROT_ADDRESS, ADCMON_COPY_SAMPLE,adc_ch_select);


        //Look for bit 7 by reading the copy sample register
        comms_data_tmp = r_pmic_i2c_read(D_PMIC_I2C_PROT_ADDRESS, ADCMON_COPY_SAMPLE);

        //Check to see if the high bit is set to indicate that the data is ready,
        //this happens within 2usec of the strobe being done
        if ((comms_data_tmp & BIT_7) == BIT_7){

            //Set the data status to fresh
            adc_data_status = D_DATA_FRESH;
            //Read the MSB and LSB
            comms_data_tmp = r_pmic_i2c_read(D_PMIC_I2C_PROT_ADDRESS, ADCMON_COPY_DATA_MSB);
            data = (comms_data_tmp <<8);
            comms_data_tmp = r_pmic_i2c_read(D_PMIC_I2C_PROT_ADDRESS, ADCMON_COPY_DATA_LSB);
            data |=  (comms_data_tmp);

        }

        else{
            /*Data is not ready to be read, try again*/
            adc_data_status = D_DATA_NOT_READY;
        }
    }
    else{
        /*Channel is out of bounds */
    }

    *adc_data = data;


    //Return the Status
    return adc_data_status;
}


// This is an example of what it looks like in a function call

adc_rtn_code = pmic_read_adc(0x0e,&adc_data);
```

Appendix E. Self Diagnosis State

The following are checked during Self Diagnosis state (not in the order of execution):

1. OTP is downloaded into registers, OTP CRC passed and register content after download is checked to be correct
2. LBIST is run and Passed
3. ADC BIST is run and Passed.
4. Fault Detect BIST is run and Passed.
5. GC BIST is run & Passed.

The fault evaluation is done at the end of Self Diagnosis timer.

PMIC reports the status of the Self Diagnosis test via the Protection **FLT_RECORD*, FUSA_STATUS_1, and FUSA_STATUS_3** registers.

PMIC also exits Self Diagnosis state when the **FUSA_TIMER_1[7:6]: TIMEOUT_SELFD_ST** timer expires.

Note: The Clock Monitor is disabled during self diagnosis.

Appendix F. Configuring GPIOs for System Testing

If the system requires to control a GPIO and check its voltage level, the following steps are followed:

The following example is for IO4.

If IO4 is needed to be set as an output and controlled HIGH & LOW, the registers to set are:

(1) Set **IO_GPIO_DATAOUTSEL10[7:4]: IO_GPIO_DATAOUTSEL1** = 0x4

(2) Set **IO_GPIO_DATAOUTSEL10[3:0]: IO_GPIO_DATAOUTSEL0** = 0x4

0x05B - IO_GPIO_DATAOUTSEL10 (OTP)

BIT	NAME	R/W	DESCRIPTION	OTP	DEFAULT
7:4	IO_GPIO_DATAOUTSEL1	RW	GPO selection for IO4	0x4	0x4
3:0	IO_GPIO_DATAOUTSEL0	RW	GPO selection for IO4 0x0 Pin fallows EN_VPP pin state 0x1 PGOOD_GLB (see PGOOD_GLB_CFG_BUCK/LDO) 0x2 PGOOD_CTL1 0x3 PGOOD_CTL2 **0x4 Outputs IO_GPIO_DATAOUT data** 0x5 PGOOD_S2R 0x6 PGOOD_ALWON (EXTPOC) 0x7 BKUP 0x8 IRQ_Regu 0x9 PGOOD_SW (PGOOD for SW SHDN regulators)	0x0	0x0

(2) Set the desired data on register IO_GPIO_DATAOUT

0x057 - IO_GPIO_DATAOUT (OTP)

BIT	NAME	R/W	DESCRIPTION	OTP	DEFAULT
7:0	IO_GPIO_DATAOUT	RW	6-bit GPIO data outputs for software to control devices in the system. The IO_GPxCFGxSB need to be configured accordingly. IO_GPxDIGDIRECTION should be set to Ouput (0x0); IO_GPxTESTSEL should be set to 0x0;	0x0	0x00

Appendix G. PMIC Recovery from LOCK State

The PMIC can recover from LOCK state by toggling ENABLE, or via ADC external channel 3 (ADC3).

ADC3 can be configured as a LOCK-release pin in OTP via register:

0x264 - I_FUSA_CTRL_DBG (OTP)

BIT	NAME	R/W	DESCRIPTION	OTP	DEFAULT
1:0	I_LOCK_RELEASE_PinSel	RW	Setting this register to accept the LOCK release function by the PIN(ADC3). : 2'b00 : Use LockReleasePin Rising Edge : 2'b01 : Use LockReleasePin Falling Edge : 2'b10 : Use LockReleasePin level high(active high) : 2'b11 : LOCK release function will not be performed by the PIN	0x1	0x1

When the PMIC is released from LOCK state, a state machine transition happens from LOCK state to SelfDiag state. In SelfDiag state, OTP downloading will be performed again. So all OTP registers are refreshed with OTP programmed values.

LOCK recovery via toggling ENABLE is the recommended option. In this option, LBIST is always re-run during Self Diagnosis.
If ADC3 is used as a LOCK-release option instead, LBIST is performed during Self Diagnosis only if an LBIST failure was recorded prior to LOCK entry. The system integrator shall take this into careful consideration.

Appendix H. External Mux Test Concept

Solution 1:

Analog signals can be stimulated with different voltage levels inside the driving SoC (e.g., like in R-Car startup test for voltage monitor).

R-Car reads sampled ADC values from PMIC via I2C / SPI

R-Car compares the ADC values with the expected values.

Expected values are defined in the customer‘s SW

In case of detected error, the customer shall define appropriate reactions.

Solution 2:

In case each analog signal has different voltage level, stimulation for analog signals for startup test is not required.

R-Car reads sampled ADC values from PMIC via I2C / SPI

R-Car compares the ADC values with the expected values.

Expected values are defined in the customer‘s SW

In case of detected error, the customer shall define appropriate reactions.

Assumption: the analog signals have constant values (e.g., 0.8 V).

Appendix I. Deep Stop / Low Power Mode

Secure Deep Stop Entry

Since the Protection block & all Safety Mechanisms are disabled during Deep Stop mode, Deep Stop entry shall only be as intended by the system.

The following procedure shall be used, at system integrator’s discretion, to prevent the PMIC from transitioning to Deep stop due to external factors other than PWRCTL2 control from the SoC unless unintended Deep Stop transitions can be detected by end-to-end protection by the communicating ECU:

(1) Write 0x3 to Regulation register **IO_PD[7:6]: REQ_LOWIQB_CFG** and confirm that 0x3 is written on Read Back

0x125 - IO_PD (OTP)

BIT	NAME	R/W	DESCRIPTION	OTP	DEFAULT
7:6	REQ_LOWIQB_CFG	RW	<00>: Always tied high <01>: Following PWRCTL2 <10>: With BACKUP_CTRL[1] register write (When software-based entry to S2R is used) <11>: PWRCTL2 & register write (Secure option for pin-controlled protection reset)	0x0	0x0

(2) Write 0x1 to Regulation register **BACKUP_CFG[7]: REQ_LOWIQB_PIN_SEC** and confirm that 0x1 is written on Read Back

0x176 - BACKUP_CFG (OTP)

BIT	NAME	R/W	DESCRIPTION	OTP	DEFAULT
7	REQ_LOWIQB_PIN_SEC	RW	0x0: Even if PWRCTL2 falls to "Lo", Deep Stop transition is not allowed 0x1: Allow Deep Stop transition when PWRCTL2 is "Lo".	0x0	0x0

0x1 must be set before Deep Stop transition.
If PWRCTL2 is set to "L" before writing 0x1 to REQ_LOWIQB_PIN_SEC, the Protection block detects voltage monitoring fault and transitions to the Error state.
REQ_LOWIQB_PIN_SEC shall be returned to 0x0 after returning from Deep Stop.
Timing for writing 0x1: the control is executed immediately before the Deep Stop transition.
The recommended timing for writing 0x0 is after the Serial Interface Check is completed in SoC Activation.

(3) Lo control of PWRCTL2 from SoC

Handling SoC CVM output signals (VTHREF, VTHSENSE)

VTHREF/VTHSENSE is 0V 500us before PWRCTL2=L

To prevent PMIC from detecting a UV condition at ADC1 & ADC2 and transition to ERROR state before the Deep Stop control is received from SoC, ADC1 & ADC2 Hi/Lo limits shall be set as:

Low Threshold = -0.3V

High Threshold = 5V.

Handling R-CAR Gen4 Error signals to avoid detecting unintended error and issuing a RESET during Deep Stop transition:

Initial setting:

I_LOCK_OUT_CFG.bit7 = 0x0 (unlock SoC Activation registers including FUSA_CTRL_E register)

Deep stop transition procedure:

(1) Set **FUSA_CTRL_E [3:0]** = 0xF to mask SDI_4 – SDI_1 pins .

(2) Wait for the completion of (1) via SPI/I2C communication interface.

(3) Start the Deep stop transition of R-Car S4.

Document Change History

NEW TO ISSUE	SECTIONS	CHANGE	CHANGED BY	DATE
0.1	-	First version	E. Silva	2021/06/11
0.2	Multiple	Sect 6.3, Figure 18,19, Section 8.1, Table 23, Figure 26, Table25 & 26, Table 31	E. Silva	2021/07/30
0.3	Multiple	Updated register addresses in register tables Address RAA271005_SAN_Modification_list_03_update Address R-CarS4-PMIC_REA_QA_211012	E. Silva	2021/10/20
0.4	8.16, 8.14	Updated descriptions	E. Silva	2021/10/20
0.5	Multiple	Updates on flow diagrams on I2C Tests & SoC Activation tests Update Section 9.2 Table 31 & 32 State diagram, & Fault Control Removed Section on Recommended Register checks	E. Silva	2022/01/26
0.6	Multiple	Updated AppendixB, Section 8.9, Figure 18, 19, Table 27, Regulation I_LOCK_OUT register description, Section 14	E. Silva	2022/04/04
0.7	Multiple	Updated System Considerations for Sections 9.3 Updated Register Map for RevB silicon Table 2, Table 4, Table 6, Table 15, Table 17, New Table 19, Table 20, Table 25, Table 26, Table 27, Table 29, Table 30, Table 32, Table 33, Table 37, Table 40 Figure 1, Figure 8, Figure 24, Figure 25, Section 13 Updated ERROR state description Added Appendix G	E. Silva	2022/07/4
0.8	Multiple	Address RAA271005_SAN_modification_list_07	E. Silva	2022/07/05
0.9	Multiple	Figure 14.2, Table 43 Section 5 item(6), Added Missing captions to some figures Section 8.4 Bucks OV & UV Comparators, (b) system considerations Table28	E. Silva	2022/07/25
0.91		Section 9.2, Section 10, Section 12.1 Fixed cross-reference links Table 29	E. Silva	2022/08/10

0.92	Multiple	Moved Section 3 to SRS Corrected TIMEOUT_SOCACTIVA_ST options SM27d: Considerations for CVM Test FLT_RECORD_A FLT_SDI* fields description correction Table 34: TR_2_3, TR_5_6, TR_3_5, TR_3_6 Table 35, Table 27 Considerations for CVM Test Section 7.15 (make relevant for both S4 & V4H) Section 8.3, Section 8.4, Section 9.2, Section 11.1, Section 13.1 Appendix G, Appendix H Figure 37, WDT Basic Windowed Kick Mode operation	E. Silva	2022/10/21
0.93	Mulitiple	Table 41 Section 11.1, 11.2, 7.16 (added SM31), Section 9.2 Added SM45 Gate Control Figure 35 updated	E. Silva	2023/01/04
0.94	Multiple	AppendixD Removed Figure 1 RAA271005 Application Diagram. Reference pointed to Datasheet. Table 1. Corrected DTI+FRT of SM19 Updated Table 20	E. Silva	2023/02/11
0.95		Updated Appendix D Section 7.1 c), Section 11.1b) Removed Section 15.2 Updated Section 7.2 SM4 Updated CVM thresholds on Table 29	E. Silva	2023/04/27
0.96		Appendix G, Appendix E Figure 8, 34 Section 5.2 (Error State Timer), 7.15 (a-d) Fault Reaction, Section 7.6, 7.8, 7.10, 7.11, 7.12, 7.18, 7.20 (Discharge Detect), 14.1 Table 29, Table 35, Table 36 Section 8.5 (b) Regulation unit fault detection & reaction during Deep Stop mode	E. Silva	2023/06/13
0.97		Updated recommended usage in section 7.4 clock monitor. Updated Considerations for CVM Test:	E. Silva	2023/08/26

0.98		Added details in Section 5.3 PMIC Recovery from Safe state Updated Section 7.2 Table 36	E. Silva	2023/10/06
0.99		Table 18a, Table 36, Deleted section 7.20 Discharge Detect., Section 7.8 & 7.10 related to message counter recommendation, Section 7.2, Section 5.2 Error state timers, Section 7.12, Section 7.4 disclose clock stuck monitoring is disabled, Appendix E, Updated RevC register changes, Section 8.1 PRESET configurability. Figure 33, Appendix I, Section 7.6 (d), Section 8.5 (c),	E. Silva	2024/1/08
0.991		Updated Table 41. Added List of Regulation faults that trigger FLT_ReguOT, Section 7.2 (c)	E. Silva	2024/02/29
0.992		Table 21 (PVIN), Identified C01 & C02 behavior in Table 35 & Table 36	E. Silva	2024/03/11
0.993		Corrected error in Table 35 for TR_2_3	E. Silva	2024/03/18
0.994		Updated Appendix I – Sections on: Secure Deep Stop Entry, Handling R-CAR Gen4 Error signals … during Deep Stop transition	E. Silva	2024/04/10
1.0		Updated CVM Fault reaction for clarification Updated Section 7.15 Overview for clarification on PRESET check & Serial Interface Check. Reviewed & Approved.		2024/05/20
2.0		Updated Section 7.15 SM27a Fault Reaction description for clarification.		
2.1		Updated Table 19 for typo Updated Appendix I with customer requested clauses Corrected Typos Checked register consistency with Datasheet & register map.	E. Silva	2024/07/11
3.0		Approved	Arun V.	2024/07/11
3.1		Fixed Table 36 C01 & C02 for Temp2/Temp4 Shutdown in & out of SoC Activation Fixed Figure 20 Example of Regulation Unit CRC Fault Test Fixed Register address for MTE_CFG_CTRL_A in Table 43 Fixed typo Appendix D ADC_COPY_SAMPLE bits from [4:0] to [5:0]		2024/10/11
4.0		Approved	Arun V.	2024/10/15

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment

for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document .
13. This document shall not be reproduced or duplicated in any form or disclosed to any third party, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

（Rev.5.0-2　October 2020）