Application Report

Safety Manual for BQ79600 -UART/SPI to Daisy Chain Bridge IC

Jyothsna Gandham

ABSTRACT

This document is a safety manual for the Texas Instruments BQ79600 UART/SPI to Daisy Chain Bridge IC. This manual provides information to help developers integrate the BQ79600 device into safety related systems.

Note

Note that before you begin a safety related project that includes the BQ79600, you should contact your local TI sales person about safety documentation from TI in addition to this safety manual.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The system and equipment manufacturer or designer (as user of this document) is responsible to ensure that their systems (and any TI hardware or software devices incorporated in the systems) meet all applicable safety, regulatory and system-level performance requirements. All application and safety-related information in this document (including application descriptions, suggested safety measures, suggested TI products, and other materials) is provided for reference only. Users understand and agree that their use of TI devices in safety-critical applications is entirely at their risk, and that user (as buyer) agrees to defend, indemnify, and hold harmless TI from any and all damages, claims, suits, or expense resulting from such use.

This document is a safety manual for the Texas Instruments BQ79600. It provides information to help system developers create safety-related systems using the BQ79600. This document contains:

- An overview of the product architecture
- An overview of the development process used to reduce systematic failures
- An overview of the safety architecture for management of random failures and Assumptions of Use (AoU) that the system integrator may consider to use this device in an ISO26262 compliant system
- The details of architecture partitions and implemented safety mechanisms

The Safety Analysis Report documents the following information, not covered in this document:

- Failure rates estimation
- Qualitative failure analysis (design FMEA, pin-FMEA, DFA, FTA)
- Quantitative failure analysis (quantitative FMEDA)
- Safety metrics calculated per targeted standards per system example implementation

The safety case documents the following information, which is not covered in this document:

- Evidence of compliance to targeted standards
- Results of assessments of compliance to targeted standards

TI expects that the user of this document has a general familiarity with the BQ79600. This document is intended to be used in conjunction with the pertinent datasheet and other documentation for the products under development. This partition of technical content is intended to simplify development, reduce duplication of content, and avoid confusion as compared to the definition of safety manual as seen in IEC 61508:2010.

2 Product Overview

The BQ79600 is a bridge IC designed to interface between microcontroller (MCU) and TI battery monitoring ICs, e.g. bq7961X and bq79606. The BQ79600 connects directly to the MCU and is isolated from the battery monitoring IC's by either a transformer or a capacitor.

During Sleep/Shutdown modes, the BQ79600 can support reverse wake up feature. The BQ79600 can wake up the MCU or PMIC, if any unmasked fault is detected in a ring architecture.

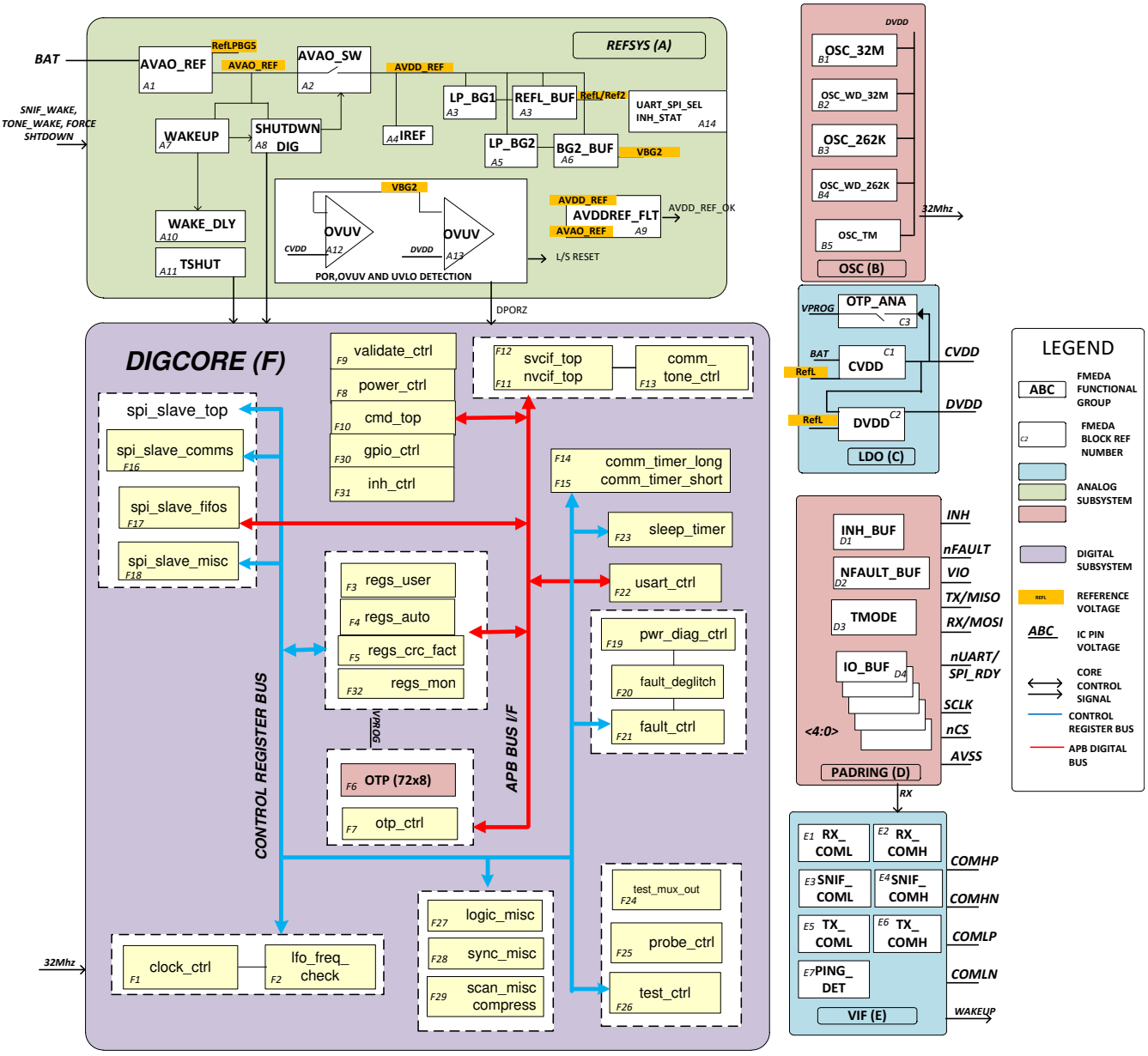

Figure 2-1. BQ79600-Q1 Architecture Overview

2.1 Target Applications

The BQ79600 is designed for use as the a bridge IC designed to interface between microcontroller (MCU) and TI battery monitoring ICs in the following applications:

- Full electric vehicle (EV), Hybrid electric vehicle (HEV) or Plug In Hybrid (PHEV) power train
- 48-V automotive battery systems
- Industrial safety applications, particularly Energy Storage Systems (ESS)

Analysis of multiple safety applications during concept phase enabled support of Safety Element out of Context (SEooC) development according to ISO 26262–10. In designing this device, TI made various assumptions about how it could be used so as to address expected industry requirements for Battery Monitoring Systems because these safety-critical systems are especially demanding.

Although TI has considered certain applications while developing these devices, this should not restrict a customer who wishes to implement other systems. With all safety-critical devices, the system integrator must rationalize the device safety concept to confirm that it meets the system safety needs.

In the case of overlapping requirements between target systems, TI has attempted to design the device respecting the most stringent requirement. For example, the fault-tolerant response-time intervals in an automotive battery application are typically on the order of 1 second. In such case, TI has performed timer subsystem analysis respecting a fault detection time interval of 100 ms for an assumed 96 battery cell application.

2.2 Product Safety Constraints

The BQ79600 safety analysis was performed under the following assumptions of system constrains:

- All inputs to the BQ79600 are within the recommended operating conditions defined in the device datasheet and do not exceed absolute operating conditions defined therein.
- The operating temperature of the BQ79600 is within the ambient and the maximum junction temperature limits defined in the device datasheet.
- All external devices to the BQ79600 meet the electrical characteristics defined in the device datasheet for the devices in question.
- The layout of the system board follows the layout guideline as defined in the BQ79600 datasheet.
- A micro-controller, FPGA, or other component capable of being a communication master, hereafter the host, is communicating directly with the ASIC through the UART interface.
- The host shall monitor the cell voltage and temperatures measured by the ASIC and shall be responsible for acting upon cell voltage and temperature information and put the system in a safe mode if appropriate.
- The host shall monitor for faults detected by the ASIC and shall be responsible for acting on faults detected by the ASIC and put the system in a safe mode if appropriate.
- The host shall monitor for loss of communication with the ASIC and shall be responsible to put the system in a safe mode if appropriate.
- Connection circuits for UART pin connections to the host and vertical interface communication pins between ASIC shall follow data sheet guidelines.
- Connection circuits for SPI pin connections to the host and vertical interface communication pins between ASIC shall follow data sheet guidelines.

3 BQ79600 Development Process for Management of Systematic Faults

For safety-critical development, it is necessary to manage both systematic and random faults. Texas Instruments has created a development process for safety-critical semiconductors, which greatly reduces the probability of systematic failures. This process builds on a standard quality-managed development process as the foundation for safety-critical development. A second layer of development activities, which are specific to safety-critical applications developments targeting IEC 61508 and ISO 26262, then augments this process. The development activity to manage systematic faults during development for the BQ79600 was done to comply with ASIL-D.

3.1 TI New-Product Development Process

Texas Instruments has been developing mixed-signal automotive ICs for safety-critical and non-safety critical automotive applications for over fifteen years. Automotive markets have strong requirements regarding quality management and product reliability. Though not explicitly developed for compliance to a functional safety standard, the TI new-product development process already featured many elements necessary to manage systematic faults.

The BQ79600 was developed using TI's new product development process which has been certified as compliant to ISO TS 16949 as assessed by Det Norske Veritas Certification, Inc.

The standard development process breaks development into phases:

- Business Planning
- Validate
- Create
- Evaluate
- Process to Production

Figure 3-1 shows the standard process.

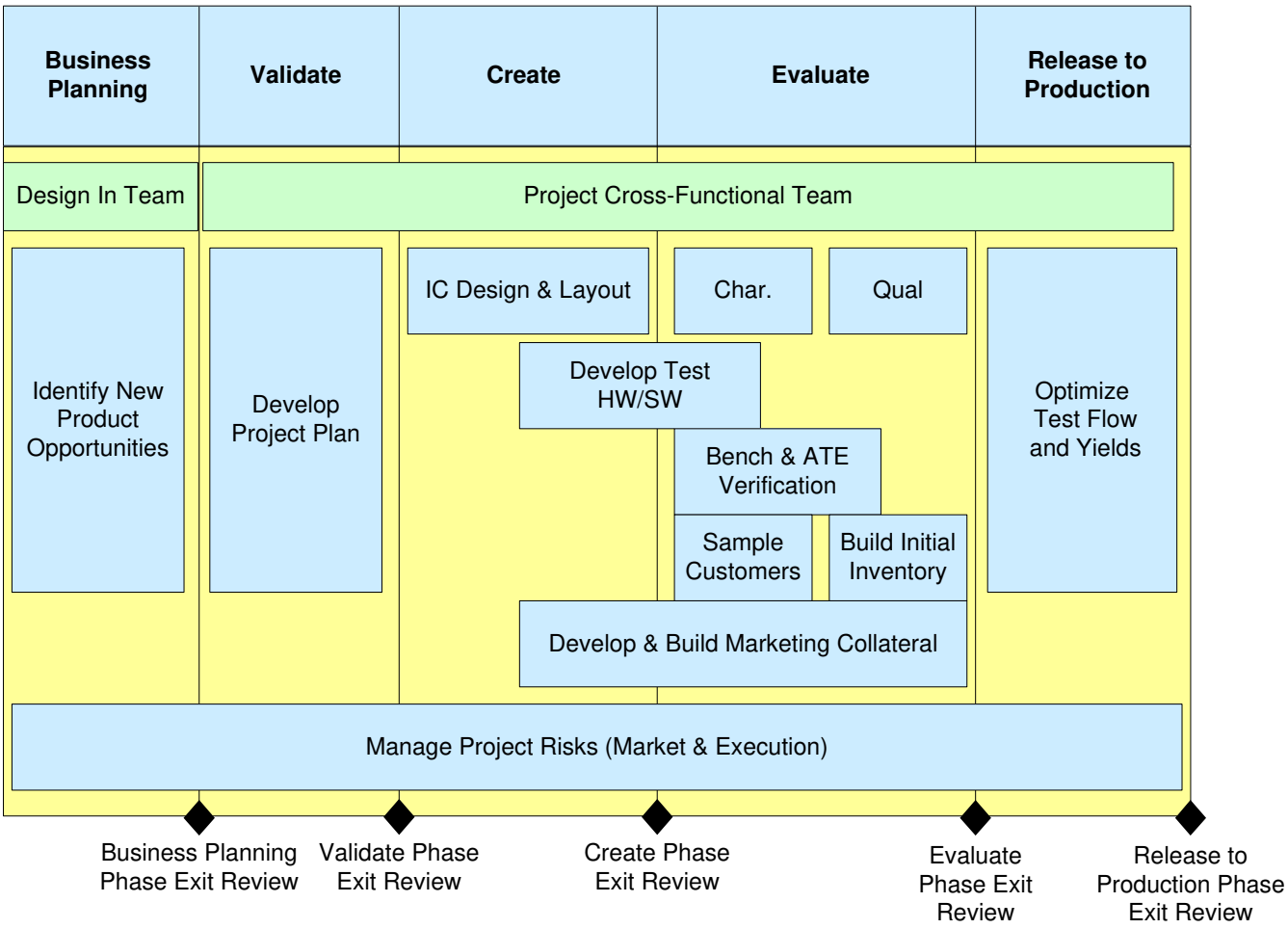

Figure 3-1. TI New-Product Development Process

3.2 TI Safety Development Flow

The TI safety-development flow derives from ISO 26262 as a set of requirements and methodologies to be applied to mixed-signal circuit safety-development flow. This flow is an integrated part of the TI new product development process. The goal of the safety-development flow is to reduce systematic faults.

The safety-development flow targets compliance to IEC 61508 second edition and ISO 26262 second edition, and is under a process of continuous improvement to incorporate state of the art best practices. While the safety-development flow is not directly targeted at other functional safety standards, TI expects that many customers will determine that other functional safety systems can readily use products developed to industry state-of-the-art.

Key elements of the TI safety-development flow are:

- Assumptions on system-level design, safety concept, and requirements based on TI's expertise in safety-critical systems development.
- Combined qualitative and quantitative or similar safety analysis techniques comprehending the sum of silicon failure modes and diagnostic techniques.
- Fault estimation based on multiple industry standards, as well as TI manufacturing data.
- Integration of lessons learned through multiple safety-critical developments to ISO 26262, IEC 61508, and participation in the functional safety international working group.

Table 3-1 lists these activities overlaid atop the standard QM development flow.

Table 3-1. TI New-Product Development Process

Business Opportunity Prescreen	Program Planning	Create	Validate, Sample, and Characterize	Quality	Ramp/Sustain
Determine if safety process execution is necessary	Define SIL/ASIL capability	Execute safety design	Validate safety design in silicon	Qualification of safety design	Implement plans to support operation and production
Execute development interface agreement (DIA) with lead customers and suppliers	Generate safety plan	Qualitative analysis of design (FMEA and FTA)	Release safety manual	Release safety case report	Update safety case report (if needed)
	Initiate safety case	Incorporate findings into safety design	Release safety analysis report	Update safety manual (if needed)	Periodic confirmation measure reviews
	Analyze assumed system to generate system level safety assumptions and requirements	Develop safety product preview	Characterization of safety design	Update safety analysis report (if needed)	
	Develop component level safety requirements	Validation of mixed-signal safety design at transistor, gate and RTL level	Confirmation measure review	Confirmation measure review	
	Validate component safety requirements meet system safety requirements	Quantitative analysis of design (FMEDA)			
	Implement safety requirements in design specification	Incorporate findings into safety design			
	Validate design specification meets component safety requirements	Validation of mixed-signal safety design at transistor/ gate/physical layout level			
	Confirmation measure review	Confirmation measure review			

3.3 Development Interface Agreement

The intent of a development interface agreement (DIA) is to define the responsibilities of the customer and supplier in facilitating the development of a functional safety system.

In custom developments, the DIA is a key document executed between customer and supplier early in the process of developing both the system and the custom TI device. As the BQ79600 device is a commercial, off-the-shelf (COTS) product developed as a safety element out of context (SEooC) a DIA between the customer and supplier is not a requirement of ISO 26262-8:2018. Refer requests for custom DIAs to your local TI sales office for disposition.

3.4 Requirements Development

The BQ79600 product is developed as a safety element out of context (SEooC) with a target safety goal of ASIL-D for communication during active mode and ASIL-B for communication during Sleep/Shutdown mode. The

safety requirement assumptions used were based on TI analysis of target safety applications. TI is willing to discuss acceptance of new customer safety requirements for future designs; please contact your local TI sales office for further information.

3.5 Availability of Safety Documentation

Table 3-2 lists the safety documentation for the BQ79600 device, which are made available either publicly or under a non-disclosure agreement (NDA):

Table 3-2. Safety Documentation

Deliverable Name	Contents	Confidentiality
Safety Manual for BQ79600 Communication Bridge Interface	User guide for the safety features of the product, including system level assumptions of use	
Safety Analysis Report Summary for BQ79600Communication Bridge Interface	Summary of FIT rates and device safety metrics according to ISO 26262 and/or IEC 61508 at device level.	
Detailed Safety Analysis Report for BQ79600 Communication Bridge Interface	Full results of all available safety analysis documented in a format that allows computation of custom metrics	NDA required

4 BQ79600 Product Architecture for Management of Random Faults

For safety-critical development, both systematic and random faults must be managed. The BQ79600 product architecture integrates several modules that can detect and report random faults, allowing a host microcontroller or other processing engine return the device to a safe state.

The device has a core set of modules allocated for continuously operating hardware safety mechanisms. It also provides programmable mechanisms to transition the device to the default(safe or shutdown state) operating mode in the event of systematic or random faults.

This section introduces the operation states and safe state of BQ79600.

4.1 Device Operating States

The BQ79600 has multiple operating states. These operating states should be monitored by the system developer in their software and system level design concepts. Refer to the product datasheet for theBQ79600 for details on the operation of the operating-states state machine. Figure 4-1 provides an overview of the operating-states state machine.

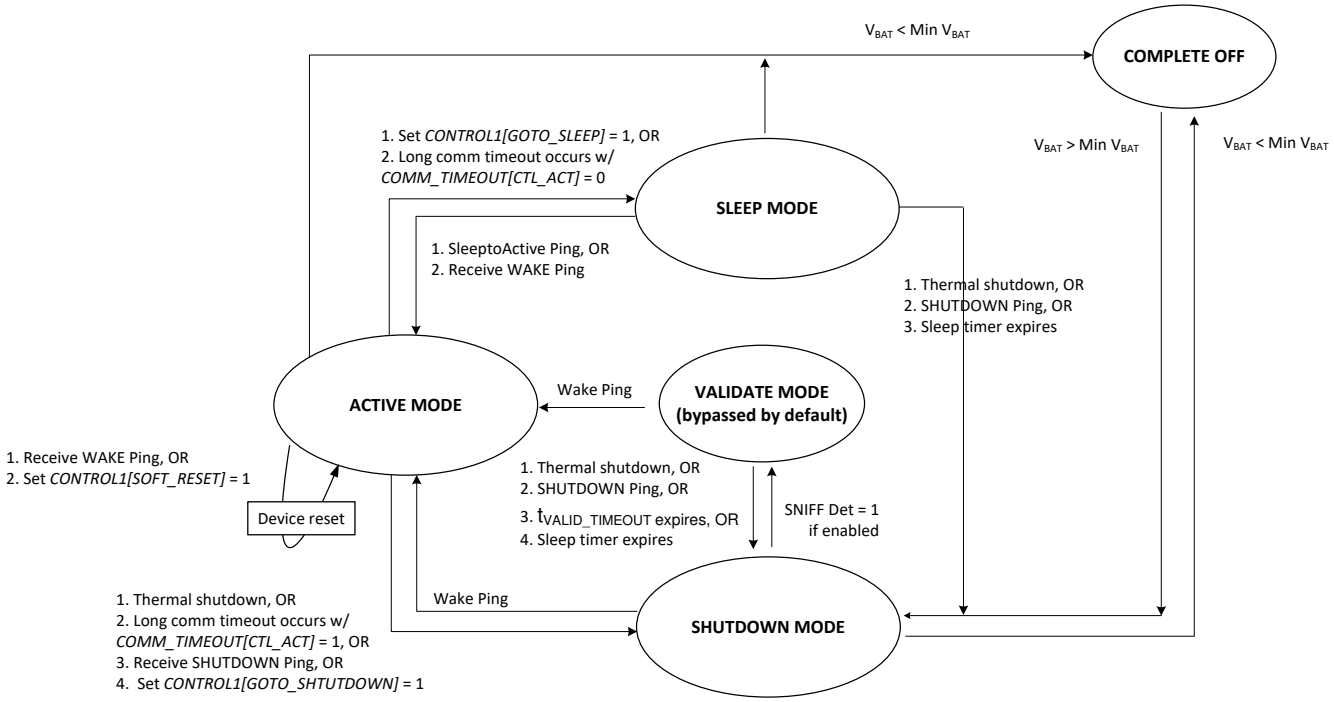

Figure 4-1. BQ79600 Operating State Machine

The BQ79600 operates in one of five modes. The mode depends on the VBAT voltage and the operational commands from the host microcontroller. A high level description of the modes is as follows:

- OFF – The VBAT voltage from the battery pack is less than Min VBAT threshold and the device is off. When the voltage on the VBAT pin is greater than Min VBAT threshold the device transition to the SHUTDOWN mode and begin to monitor for wake up signal.
- SHUTDOWN – The lowest power state available. To wake up to the ACTIVE mode the device monitors the RX pin for a WAKE ping input. The Snif Detector feature can be configured to be ON in this mode.
- ACTIVE – In the ACTIVE mode, the device is actively communicating with the host microcontroller and with the battery monitoring ASIC's in the stack. From the ACTIVE mode the device can enter the SLEEP mode or the SHUTDOWN mode upon Thermal shutdown or command or a communication timeout.
- SLEEP – In the SLEEP mode, the device has limited functionality. A SLEEPtoACTVIE or WAKE signal detection transitions the device into the ACTIVE mode. A SLEEP timeout signal or SHUTDOWN signal or thermal shutdown will transition the device into the SHUTDOWN mode.

- FAULT VALIDATE – The Fault validate mode has limited functionality. It exists only when the Snif Detector feature is enabled. When the device detects fault tones in Shutdown mode, it transitions to Fault Validate mode, to validate the fault tones. A status bit [VALIDATE_DET] is set on entering this mode. This mode is bypassed when Snif Detector feature is disabled.

4.2 Safe State

The device shall be considered in the safe state when no power is applied, or when device is in SHUTDOWN mode or when device is operating in a fully functional and fault-free integrated system. The device shall be considered in a safe state after a communication fault or supporting hardware fault is detected and signaled to the host. The host is responsible to monitor for fault signal communications from the device. The host is responsible for monitoring for communication loss or communication faults with the host. The external host element of the system/item is responsible for fault reaction and transition of the system to a system safe state.

5 BQ79600 Architecture Safety Mechanisms and Assumptions of Use

This section summarizes the safety mechanisms for each major functional block of the BQ79600 architecture and provides their assumptions of use. Each assumption of use is indicated by [AoUx] with *x* being the identification number. The safety analysis report notes the effectiveness of these safety mechanisms.

Naturally, the system integrator must comprehensively assess effectiveness in the context of the specific end use.

The safety measures described in this document may relate to one or more of the safety goals listed in Table 5-1.

Table 5-1. Assumed Safety Goal Number

Goal Number	Description
1	Communication in active mode
2	Communication in Sleep/Shutdown mode

The number of each safety measure is not strictly sequential. Table 5-2 describes the number range and the related functionality of the device covered.

Table 5-2. Safety Measure Numbering Scheme Description

Range	Coverage Description
0-99	Substantially related to supply rail and reference diagnostics
100-199	Substantially related to communication diagnostics
200-299	Safety measures covering device functions not primarily in other categories

Table 5-3. Safety Mechanism Categories

Diagnostic Interval	Description
FDTI	Mechanisms or diagnostic functions designed to be handled with external microcontroller assistance within each Fault Tolerant Detection Interval
MPFDI	Mechanisms or diagnostic functions designed to be executed with external microcontroller assistance at least once within Multi Point Fault Detection Interval.
AUTO	Mechanisms that are passive elements or automatically executed by the ASIC
PGM	Mechanisms or diagnostic functions for use during device programming and not used during normal operation

Note

Detection - a test which is run frequently or continuously for the purpose of preventing a single point safety goal violation (that is output driver over-voltage reporting).

Diagnostic - a test which is performed periodically (i.e. once per ignition cycle) for the purpose of preventing a latent safety goal violation, such as a failed detection (for example inject over-voltage to verify over-voltage detection works).

5.1 Safety Mechanisms per Design Block

Table 5-4. Safety Mechanisms

Safety Mechanisms by Design Block that are used for multiple blocks listed once and are not repeated in this table				
Design Block	**SM #**	**Safety Mechanism Name**	**Diagnostic Interval**	**Diagnostic/Detection**
DVDD_LDO	SM010	DVDD OV Detection	FDTI	Detection
DVDD_LDO	SM011	DVDD OC Protection	Auto	Detection
CVDD_LDO	SM012	CVDD OV Detection	FDTI	Detection
CVDD_LDO	SM013	CVDD UV Detection	FDTI	Detection
CVDD_LDO	SM014	CVDD OC Protection	Auto	Detection
REFSYS	SM015	AVDDREF OV Detection	FDTI	Detection
REFSYS	SM016	AVDDREF SW Fail Detection	FDTI	Detection
REFSYS	SM017	Power Supply Diagnostic Test Mode	MPFDI	Diagnostic
COMM	SM100	MCU loss of signal detection	FDTI	Detection
COMM	SM101	MCU unexpected data error detection	FDTI	Detection
UART	SM102	UART CRC Error detection	FDTI	Detection
VIF	SM103	Daisy Chain CRC Error detection	FDTI	Detection
COMM	SM104	Short Comm Timeout detection	FDTI	Detection
COMM	SM105	Long Comm Timeout detection	FDTI	Detection
UART	SM106	UART Comm Clear detection	FDTI	Detection
UART	SM107	UART STOP Bit Error detection	FDTI	Detection
COMM	SM108	Start of Frame error detection	FDTI	Detection
COMM	SM109	Byte error detection	FDTI	Detection
COMM	SM110	UNEXP communication detection	FDTI	Detection
COMM	SM112	IERR detection	FDTI	Detection
COMM	SM113	WAIT error detection	FDTI	Detection
VIF	SM114	Daisy Chain SYNC1 Error detection	FDTI	Detection
VIF	SM115	Daisy Chain SYNC2 Error detection	FDTI	Detection
VIF	SM116	Daisy Chain BIT Error detection	FDTI	Detection
VIF	SM117	Daisy Chain BYTE Error detection	FDTI	Detection
VIF	SM118	Daisy Chain Fault Signal diagnostic	MPFDI	Diagnostic
COMM	SM119	NFAULT pin diagnostic	FDTI	Diagnostic
VIF	SM120	Sleep Mode Fault Tone	FDTI	Detection
VIF	SM121	Sleep Mode Heartbeat	FDTI	Detection
VIF	SM122	Fast heartbeat detection	FDTI	Detection
COMM	SM123	Daisy Chain CRC diagnostic	MPFDI	Diagnostic
COMM	SM124	MCU Comm and Fault Mask diagnostic	MPFDI	Diagnostic
COMM	SM125	MCU Device Address diagnostic	MPFDI	Diagnostic
COMM	SM126	MCU communication fault diagnostics	MPFDI	Diagnostic
COMM	SM127	FMT Error Detection	FDTI	Detection
COMM	SM128	SPI Comm Clear Detection	FDTI	Detection
COMM	SM129	TX Data UNEXP Error Detection	FDTI	Detection
COMM	SM130	RX Data UNEXP Error Detection	FDTI	Detection
COMM	SM131	Correct Comm Interface Detection	FDTI	Detection
COMM	SM132	FIFO Register Diagnostic	MPFDI	Diagnostic
COMM	SM133	TXFIFO Underflow Detection	FDTI	Detection
COMM	SM134	TXFIFO Overflow Detection	FDTI	Detection
COMM	SM135	RX FIFO Overflow Detection	FDTI	Detection
COMM	SM136	MCU SPI Fault Diagnostics	MPFDI	Diagnostic

Table 5-4. Safety Mechanisms (continued)

Safety Mechanisms by Design Block that are used for multiple blocks listed once and are not repeated in this table				
Design Block	**SM #**	**Safety Mechanism Name**	**Diagnostic Interval**	**Diagnostic/Detection**
COMM	SM137	SPI Conflict Detection	FDTI	Detection
COMM	SM200	Snif Detector Diagnostic	MPFDI	Diagnostic
COMM	SM201	INH Pin Status Detection	FDTI	Detection
COMM	SM202	INH Driver Diagnostic	MPFDI	Diagnostic
LFOSC	SM203	LFOSC missing clock detection	Auto	Detection
HFOSC	SM204	HFOSC missing clock detection	Auto	Detection
HFOSC	SM205	LFOSC frequency mismatch detection	FDTI	Detection
NVM	SM206	Factory Register CRC detection	FDTI	Detection
NVM	SM207	FACT CRC diagnostic	MPFDI	Diagnostic
NVM	SM208	Customer registers integrity detection	FDTI	Detection
NVM	SM209	Customer registers integrity diagnostic	MPFDI	Diagnostic
OTP	SM210	OTP Factory Load Error	FDTI	Detection
TSHUT	SM211	Thermal Shutdown detection	Auto	Detection
PING	SM212	SHUTDOWN Status	Auto	Detection
test_ctrl	SM213	Fact Testmode Detection	FDTI	Detection

5.2 Architecture Safety Mechanisms Related to Supply Rail and Reference Voltages

The BQ79600 architecture safety mechanisms for the supply rail and reference voltages are described in the next sections.

5.2.1 SM010: DVDD OV Detection

The BQ79600 automatically compares the 1.8-V DVDD LDO output voltage against an over-voltage threshold. If a failure condition is valid, the DVDD_OV bit in register FAULT_PWR will be set.

[AoU1] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_PWR bit is 0.

5.2.2 SM011: DVDD Current Limit

The BQ79600 monitors the DVDD LDO output current and limits it according to the datasheet specifications. This protects circuits in the case of a short circuit or severe transient load.

Note

The current limit mechanism works continuously and has no status indication that can be monitored.

5.2.3 SM012: CVDD OV Detection

The BQ79600 compares the 5-V CVDD LDO output voltage against an over-voltage threshold and sets bit CVDD_OV in register FAULT_PWR.

[AoU1] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_PWR bit is 0.

5.2.4 SM013: CVDD UV DRST Detection

The BQ79600 compares the 5-V CVDD LDO output voltage against an under-voltage threshold and sets bit CVDD_UV_DRST in register FAULT_PWR. This condition will trigger a digital reset of the device.

[AoU1] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_PWR bit is 0.

5.2.5 SM014: CVDD Current Limit

The BQ79600 measures the CVDD LDO output current and limits it according to the datasheet specifications. This protects circuits in the case of a short circuit or severe transient load.

Note

The current limit mechanism works continuously and has no status indication that can be monitored.

5.2.6 SM015: AVDD_REF OV Detection

The BQ79600 compares the 2.4-V always-on internal AVDDREF voltage against an over-voltage threshold and sets bit AVDDREF_OV in register FAULT_PWR.

[AoU1] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_PWR bit is 0.

5.2.7 SM016: AVDDREF SW FAIL Detection

The BQ79600 compares the 2.4-V AVAO_REF output voltage against the AVDDREF voltage. The two rails are connected by a switch that should have a very small voltage drop across it. If the voltage drop exceeds the datasheet limit, the AVAO_SW_FAIL bit in register FAULT_PWR is set.

[AoU1] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_PWR bit is 0.

5.2.8 SM017: Power Supply Diagnostic Test Mode

The BQ79600 includes a Power Mode test mode to help detect latent faults of LDO's over-voltage detection circuit function.

This test mode covers the faults bits CVDD_OV, DVDD_OV, AVDDREF_OV in the FAULT_PWR register. The result of the diagnostic test are indicated in the CVDD_OV, DVDD_OV, AVDDREF_OV bits in register FAULT_PWR.

[AoU2] — The host MCU conducts the diagnostics every MPDTI.

[AoU3] — The host writes PWR_DIAG_GO = 1 (a self clear bit), to enable the diagnostic test mode.

[AoU4] — The host waits for PWR_DIAG_RDY= 1 before reading CVDD_OV, DVDD_OV, AVDDREF_OV bits in register FAULT_PWR to verify that the faults are set.

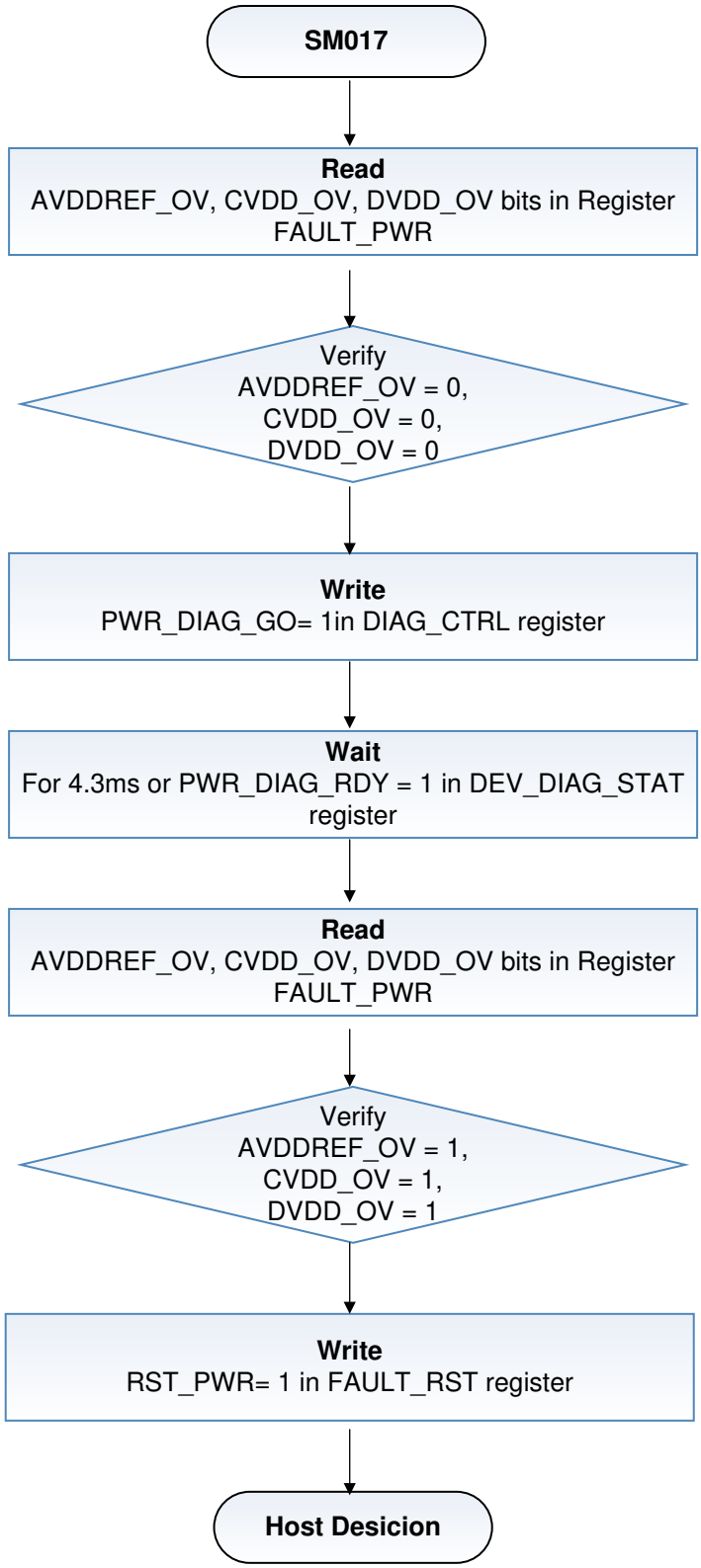

Figure 5-1. SM017: Power Supply Test Mode

5.3 Architecture Safety Mechanisms Related to Communication Diagnostics

The BQ79600 communication path has multiple diagnostics to help achieve the safety goals of the device. The host MCU should monitor for communication for expected results at all times and conduct communication diagnostic and monitor for any communication faults detected.

5.3.1 SM100: MCU loss of Signal Detection

The host MCU calculates a maximum response time for each communication based on the type of communication and the expected responses. If all the expected replies are not received within the allotted time the controller should recognize the loss of communication fault.

[AoU5] — The host MCU to verify the expected communication responses are received within the allotted time.

5.3.2 SM101: MCU Unexpected Data Error Detection

The host MCU calculates for each expected UART frame to be received from the ASIC the expected frame data length, the expected device address, the expected register address, the expected number of data bytes plus CRC. If the UART reply does not match the calculated expectation the controller should recognize its as an unexpected data received fault. The non-conforming data transmission should be discarded.

[AoU6] — The host MCU to verify the data received is as expected.

5.3.3 SM102: UART /SPI CRC Error Detection

The BQ79600 calculates a16-bit CRC of the UART/SPI data received from the host MCU and compares it to the CRC data sent in the UART/SPI frame. If the CRC calculated and received CRC do not match the BQ79600 sets the RC_CRC bit. RC_CRC bit is set if the CRC error occurs with a command frame in registers FAULT_COMM1 or FAULT_COMM2. There is a pair of CRC fault bits for the UART_Frame and SPI_Frame. The non-conforming data transmission is discarded.

The host MCU calculates a16-bit CRC from the UART/SPI data received and compares it to the CRC data sent in the UART/SPI frame. If the CRC calculated and sent CRC do not match the host MCU should recognize its as a CRC fault. The non-conforming data transmission should be discarded.

[AoU7] — The host MCU calculates a 16-bit CRC to be included as part of each UART/SPI communication frame sent.

[AoU8] — The host MCU to verify the CRC data received from is correct.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.4 SM103: Daisy Chain CRC Error Detection

TheBQ79600 automatically calculates a16-bit CRC for each VIF daisy chain communication frame received and compares it to the CRC data sent in the VIF communication frame. If the CRC calculated and sent CRC do not match the BQ79600 sets the RR_CRC bit in FAULT_COMM2 register. The RR_CRC bit is set for read frame. There is a pair of CRC fault bits for the COML_FRAME and COMH_FRAME. The non-conforming data transmission are discarded.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.5 SM104: Short Comm Timeout Detection

In the ACTIVE mode to help detect unexpected communication delay a short communication timeout checks for absence of a valid frame received from either UART/SPI or daisy chain. The timer is enabled by setting CTS_TIME[2:0] bits in the COMM_TIMEOUT_CONF register. The timer is reset when a valid frame received from either UART/SPI or daisy chain. If the short communication timeout expires the CTS bit in FAULT_SYS register is set. The device remains in the ACTIVE mode. This bit can be monitored to help detect unexpected delay in communication with the host MCU or from the stack of battery monitoring devices.

[AoU10] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_SYS bit is 0.

5.3.6 SM105: Long Comm Timeout Detection

In the event of an unexpected communication delay in the ACTIVE mode the BQ79600 long communication timeout can automatically put the device into lower power SLEEP mode or optionally into the SHUTDOWN mode. The timer is enabled by setting CTL_TIME[2:0] bits in the COMM_TIMEOUT_CONF register. The timer is reset when a valid frame received from either UART/SPI or daisy chain. If the long communication timeout expires the CTL bit in FAULT_SYS register is set and the device is put into the SLEEP mode. The CTL bit in FAULT_SYS register can be read after SLEEPtoACVITE transition to ACTIVE mode. The CTL bit in FAULT_SYS register will be reset by the register reset that occurs as part of the device wake up.

Optionally the CTL_ACT bit in the COMM_TIMEOUT_CONF register can be set so the device is put into the SHUTDOWN mode when the long communication timeout expires. The CTL bit in FAULT_SYS register will be reset by the register reset that occurs as part of the device wake up.

Note

In the event BQ79600 and the host MCU are unable to communicate the long communication timeout setting may be configured for the BQ79600 to remain in the ACTIVE mode, or to transition to the SLEEP mode, or transition to the SHUTDOWN mode.

[AoU10] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_SYS bit is 0.

[AoU11] — Device is configured to SHUTDOWN when the Long Communication timer expires.

5.3.7 SM106: UART Comm Clear Detection

The receiver continuously monitors the RX line for UART communication break condition indicating the next byte received will be a new start of frame. When a COMM CLEAR is detected the BQ79600immediately terminates the current communication and it sets the COMMCLR_DET bit in FAULT_COMM1 register. The host must wait at least tUART(RXMIN) after the COMM CLEAR to start sending a new communication frame.

Note

The STOP_DET bit FAULT_COMM1 register will also be set because the COMM CLEAR timing violates the typical byte timing.

The SLEEPtoACTIVE ping on the RX pin will also clear the UART receiver. The COMMCLR_DET bit is set when transiting from SLEEP mode to ACTIVE mode. If device is ACTIVE, the SLEEPtoACTIVE ping on the RX pin will set both COMMCLR_DET and STOP_DET bit.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

[AoU12] — After a communication break and normal communication is established the host MCU reads and resets the COMMCLR_DET bit.

5.3.8 SM107: UART STOP Bit Error Detection

The STOP bit indicates the end of the UART byte transmission. If a UART data byte is received that does not have the STOP bit, the STOP_DET bit FAULT_COMM1 register is set.

Note

A UART communication break on RX pin will set the STOP_DET bit FAULT_COMM1. The SLEEPtoACTIVE ping on the RX pin will set both COMMCLR_DET and STOP_DET bit.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

[AoU13] — After a communication break and normal communication is established the host MCU reads and resets the STOP_DET bit.

5.3.9 SM108: Start of Frame Error Detection

If a break or new start of frame is received before the current frame is finished, on either the UART/SPI or VIF stack communications, then the SOF bit will be set. One of six SOF bits may be set depending on the interface; UART, SPI, COML or COMH and the type of communication; receiving command(RC), receiving response(RR) or transmitting data(TR).

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.10 SM109: Byte Error Detection

The BYTE_ERR bit is set when a invalid bit count occurs on any byte in a frame received on the COMH/COML or when a STOP error occurs on any byte received on the UART/SPI when not followed by a communication clear. One of four BYTE_ERR bits may be set depending on the interface; UART, SPI, COML and COMH and the type of communication; receiving command(RC) or receiving response(RR). When the byte error occurs, all further bytes received on that interface are ignored. Any other frame errors that occur are ignored. Bytes received on COMH/COML are propagated up the stack, while bytes received on UART/SPI are not propagated.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.11 SM110: UNEXP Communication Detection

The UNEXP communication bit is set if communication is detected from an unexpected interface. For example, the received response direction is wrong. UNEXP bits may be set on COML and COMH on received response(RR). When communication from unexpected interface occurs, all further bytes received on that interface are ignored.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.12 SM112: IERR Detection

The IERR error bit is set when receiving an invalid frame initialization byte. One of four IERR bits may be set depending on the interface; UART, SPI, COML and COMH, that receives the invalid frame initialization byte. When an initialization byte error occurs, the UART/SPI disregards communication and does not forward.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.13 SM113: WAIT Error Detection

The WAIT error bit is set when the start of a new command is received prior to the completion of all the response from prior command. The two WAIT bits may be set depending on the interface; UART or SPI. Start of new communication must always wait for completion of communication reply from prior command.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.14 SM114: Daisy Chain SYNC1 Error Detection

The SYNC1 error bit is set when the synchronization data on the VIF communication bus have errors and the timing is likely not correct. This error indicates noise has corrupted the timing information in the first bits of the communicated data. The SYNC1 bit set is depending on the interface COML and COMH.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.15 SM115: Daisy Chain SYNC2 Error Detection

The SYNC2 error bit is set when the timing data extract from the first bits of the communicated data is outside of the expected window. It is likely that the data is not sampled correctly or noise has corrupted the timing

information in the first bits of the communicated data. The SYNC2 bit set is depending on the interface COML and COMH.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0..

5.3.16 SM116: Daisy Chain BIT Error Detection

The BIT error bit is set when the voltage level extract from the VIF communicated data is not enough samples to detect a reliable logic level, or if a bit is corrupted due to noise. The BIT error bit set is depending on the interface COML and COMH.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.17 SM117: Daisy Chain BYTE Error Detection

The PERR daisy chain BYTE error bit is set when the VIF communication data has a bit missing or incorrect complementing daisy chain signals and is unable to detect a valid data byte communication frame. The PERR error bit set is depending on the interface COML and COMH.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.18 SM118: Daisy Chain Fault Signal Diagnostic

The host MCU transmits a UART/SPI command to the BQ79600 to send a corrupt CRC value setting the FLIP_TR_CRC bit in register DIAG_CTRL. The nFAULT pin signal should be enabled and monitored by the host before and after sending an incorrect CRC. The host should then reset the CRC fault.

[AoU2] — The host MCU conducts the diagnostic every MPDTI

5.3.19 SM119: NFAULT Pin Diagnostic

The host MCU the host transmits a UART/SPI command to the base device with an incorrect initial byte or other command error. The nFAULT pin signal should be enabled and monitored by the host . The host should then reset the fault.

[AoU14] — The host MCU conducts the diagnostic every FDTI

5.3.20 SM120: Sleep Mode Fault Tone

In SLEEP mode, if an unmasked fault is detected on the stack, the stack device sends out a fault tone. The BQ79600 shall set the FTONE_DET in register FAULT_COM1 bit and nFAULT pin is asserted.

[AoU15] — When an application includes ring communication option for safety-relevant functions while in the SLEEP mode the host MCU enables sleep mode fault tone prior to SLEEP mode and then monitors the nFAULT pin while the devices in the stack are sleeping.

5.3.21 SM121: Sleep Mode Heartbeat

In SLEEP mode, with ring communication established if a heartbeat tone is not detected periodically the HB_FAIL bit in register FAULT_COMM1 is set and nFAULT pin is asserted.

[AoU16] — When an application includes ring communication option for safety-relevant functions while in the SLEEP mode the host MCU enables heartbeat tone prior to SLEEP mode and then monitors the nFAULT pin while the devices in the stack are sleeping.

5.3.22 SM122 Fast Heartbeat Detection

In SLEEP mode, with ring communication established, if a heartbeat tone is detected more often than expected the HB_FAST bit in register FAULT_COMM1 is set and nFAULT pin is asserted. This error indicates a problem with the configuration of heartbeat ring communication or noise interfering with the heartbeat tone.

[AoU17] — When an application includes ring communication option for safety-relevant functions while in the SLEEP mode the host MCU enables heartbeat tone prior to SLEEP mode and then monitors the nFAULT pin while the devices in the stack are sleeping.

5.3.23 SM123: Daisy Chain CRC Diagnostic

The BQ79600 has a diagnostic feature to intentionally create an incorrect CRC value in the VIF daisy chain communication transmissions response. When the FLIP_TR_CRC bit in register DIAG_CTRL is set an incorrect CRC value is created the by inverting all of the calculated CRC bits.

[AoU2] — The host MCU conducts the diagnostic every MPDTI

5.3.24 SM124: MCU Comm and Fault Mask Diagnostic

When customer control register (addr 0x306 - 0x2030) is written then the host MCU shall send a command sequence to theBQ79600 to read back and verify the register bit setting are correct. Then periodically, within the multipoint fault response time, the host MCU shall send a command sequence to read back and verify the register bit setting are correct.

[AoU18] — When the host MCU updates the register contents for customer control features the host MCU should read back the value written to verify the register value. Periodically the host MCU should read back the register value settings.

[AoU19] — When an application includes cell balance or protection during SLEEP mode the host MCU should read back the control registers values before entering SLEEP mode to verify the register value are correct.

5.3.25 SM125: MCU Device Address Diagnostic

Periodically, and within the multipoint fault response time the host MCU shall send a stack read command sequence while the ASIC are in the active mode to verify the response are correct for the communication configuration with correct number of ASIC, correct device addresses in the correct order.

[AoU2] — The host MCU conducts the diagnostic every MPDTI

5.3.26 SM126: MCU UART Communication Fault Diagnostics

Periodically, and within the multipoint fault response time, if the host communicates to the ASIC via UART, the MCU shall send separate UART frames with:

(1) an incorrect CRC

(2) an invalid initial byte

(3) an invalid wait time between commands

The MCU shall then verify the matching error flag register results and nFAULT pin status, clear the faults and proceed to the next diagnostic.

[AoU2] — The host MCU conducts the diagnostic every MPDTI.

5.3.27 SM127: FMT Error Detection

In a SPI communication mode, the ASIC monitors the receive commands for during non read mode by checking 1st byte of data after nCS falling edge and sets the bit SPI_PHY in register FAULT_COMM2, when it receives malformed commands.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.28 SM128: SPI Comm Clear Detection

In a SPI communication mode, the ASIC monitors the number of SCLK pulses it receives during comm clear and sets the SPI_PHY bit in register FAULT_COMM2 if it receives more than 8 SCLK pulses.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.29 SM129: TX Data Unexp Detection

In a SPI communication mode, the ASIC sets the SPI_PHY bit in register FAULT_COMM2 if it receives unexpected data from itself or from the stack devices after a COMMCLR or it receives unexpected data from daisy chain after a daisy chain timeout.

[AoU9] The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.30 SM130: RX Data Unexp Detection

In a SPI communication mode, the ASIC sets the SPI_PHY bit in register FAULT_COMM2 if the MCU sends data other than 0xFF during Device Read mode OR initiates SPI communication when SPI_RDY = 0 (e.g. While FIFO2 is being filled up, host continues reading FIFO2 right after FIFO1 is read out. SPI_RDY is low at this point).

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.31 SM131: Correct Communication Interface Diagnostic

Periodically, and within the multipoint fault response time, host should monitor the communication to verify the correct communication protocol is selected.

5.3.32 SM132: FIFO Register Diagnostic

The BQ79600 includes a test mode to help detect latent faults of FIFO Register. Periodically, and within the multipoint fault response time, the MCU shall enter the FIFO diagnostic test mode and write 32 bytes into the RX buffer. The ASIC will copy the RXFIFO as is for the first copy, and then rotate the data left by 1 bit for each subsequent copy.

The MCU shall then wait for SPI_RDY = 1 for the event to be complete and then read the TX FIFO to verify the data is as expected. The MCU should then send COMMCLR to exit the FIFO Diagnostic test mode.

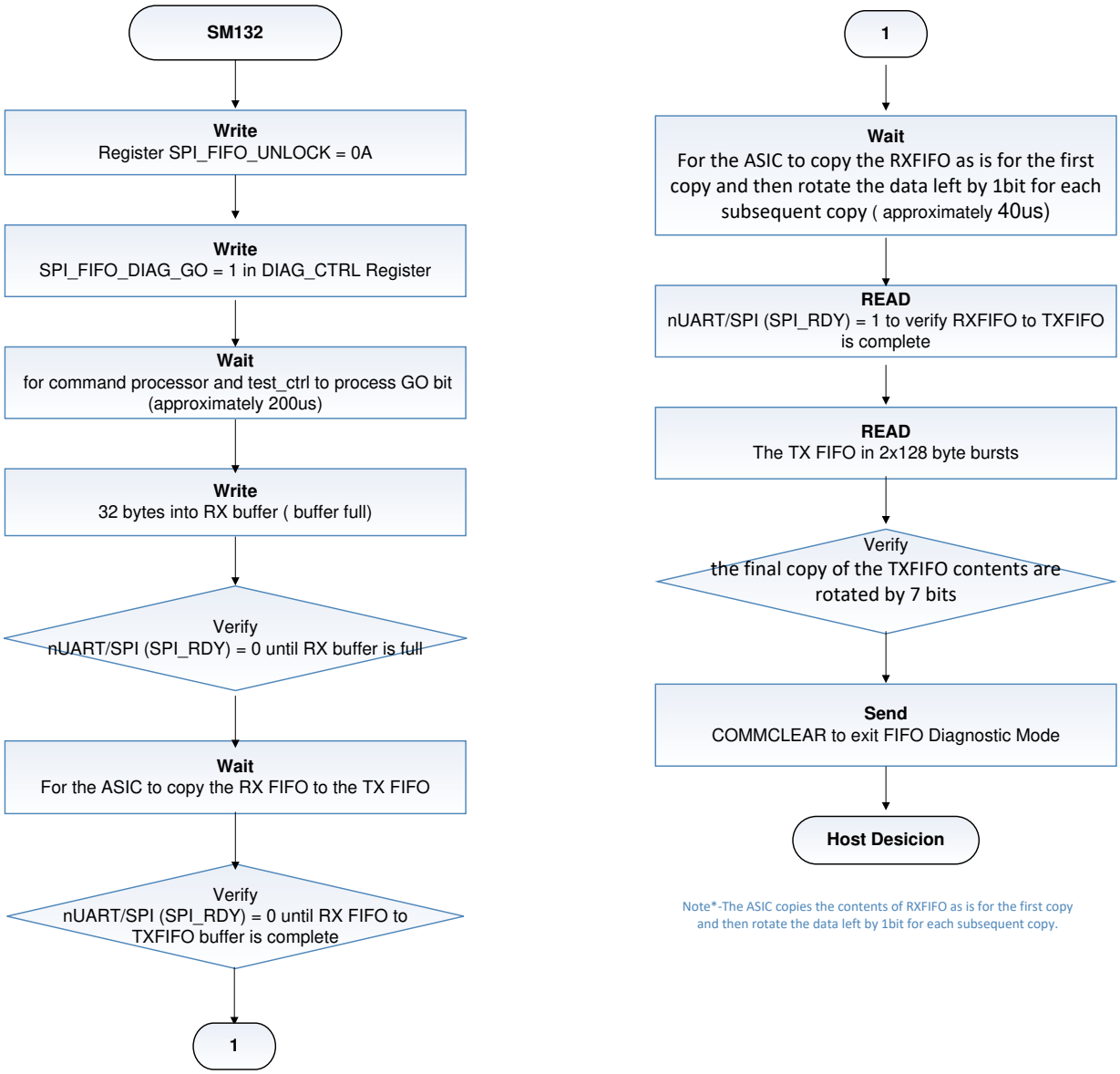

Figure 5-2. SM132: FIFO Register Diagnostic Flow Chart

Note

RX FIFO is 32 bytes, TX FIFO is 256 bytes. The ASIC copies the contents of RX FIFO as is for the 1st copy and then rotates the data left by 1bit for each subsequent copy, so the final copy of the RXFIFO contents will be rotated by 7 bits. The host can fill the pattern with any values other than 0x00(COMM_CLEAR), and can run the test 2 consecutive times if needed.

[AOU2] — The host MCU conducts the diagnostic every MPFDI.

[AoU20] — MCU shall write the unlock code (0x0A) to SPI_FIFO_UNLOCK followed by SPI_FIFO_DIAG_GO = 1 to start the FIFO Diagnostic.

[AoU21] — MCU shall read the TX FIFO in 2 bursts of 128 bytes each (256 bytes).

[AoU22] — Host shall not fill the RX FIFO with 0x00 to avoid ASIC considering it as COMM_CLEAR.

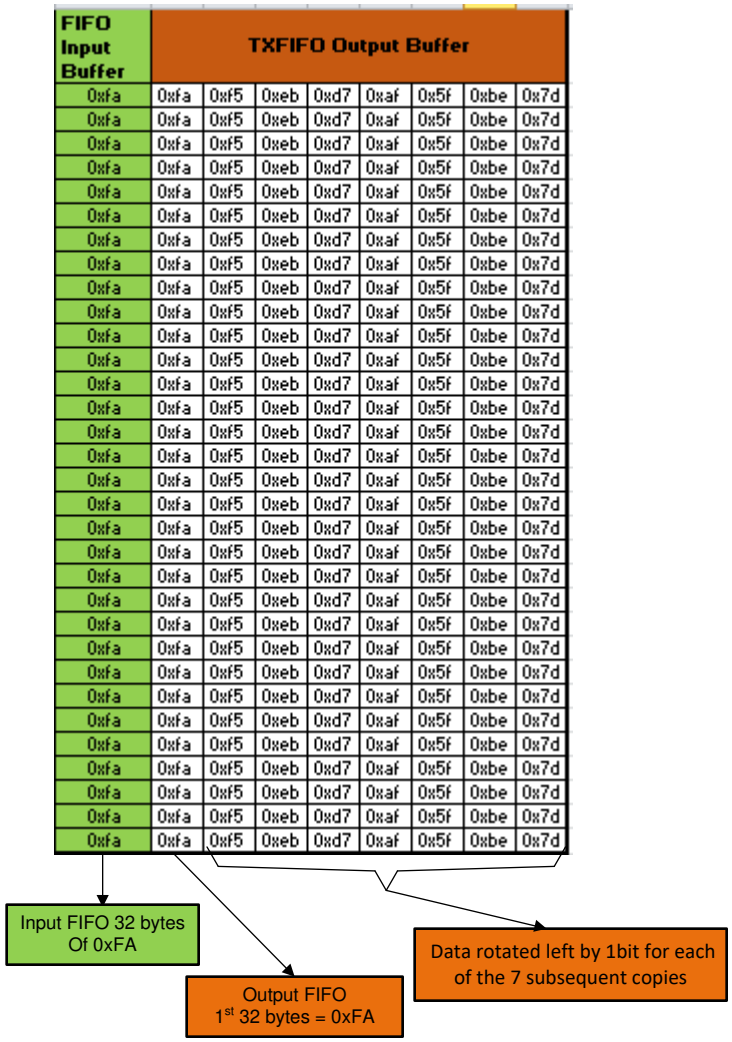

Figure 5-3. SM132: Example Pattern for FIFO Diag Test Mode

5.3.33 SM133: TX FIFO Underflow Detection

The ASIC detects underflow of the TX FIFO during SPI communication.

During Read mode when both the TX FIFO's are empty, if the MCU continues to send clocks for more data, the ASIC sets the SPI_PHY bit in register FAULT_COMM2.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.34 SM134: TX FIFO Overflow Detection

The ASIC detects underflow of the TX FIFO during SPI communication.

If the stack devices are sending data when the current TX FIFO is full and the other TX FIFO is not empty (MCU didn't complete reading the 2nd TX FIFO), the ASIC sets the SPI_PHY bit in register FAULT_COMM2.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.35 SM135: RX FIFO Overflow Detection

The ASIC detects overflow of the RX FIFO during SPI communication.

If the ASIC receives more data than it can accommodate in RX FIFO, the ASIC sets the SPI_PHY bit in register FAULT_COMM2.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.3.36 SM136: MCU SPI Fault Diagnostics

Periodically, and within the multipoint fault response time, if the host communicates to the ASIC via SPI, the MCU shall send separate SPI frames to test the following:

(1) FMT

(2) TX FIFO underflow

(3) TX FIFO overflow

(4) RX FIFO overflow

The MCU shall then verify the matching error flag register results and nFAULT pin status, clear the faults and proceed to the next diagnostic.

[AoU2] — The host MCU conducts the diagnostic every MPFDI.

5.3.37 SM137: SPI Conflict Detection

The ASIC detects SPI conflict during SPI communication.

During Read mode, if the MCU is sending command frames, the ASIC sets the SPI_PHY bit in register FAULT_COMM2.

[AoU9] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_COMM bit is 0.

5.4 Architecture Safety Mechanisms Related to Device Functions not in Other Categories

5.4.1 SM200: Snif Detector Diagnostic

To verify the functionality of the Snif Detector, the MCU shall periodically run the following diagnostic test and monitor the VALIDATE_DET bit in register FAULT_SYS.

The host sends a command to set CRC fault on the top of the stack device. The top of the stack device then sends fault tones out of COMH. The MCU then puts the bq79600 into SHUTDOWN mode and after 100ms OR after detecting NFAULT pin toggle low(once enter VALIDATE mode and fault tone is validated, it sets a fault), the MCU sends a wake ping to bq79600.

Host then verifies the bit VALIDATE_DET= ‘1’ to confirm SNIF DET is working.

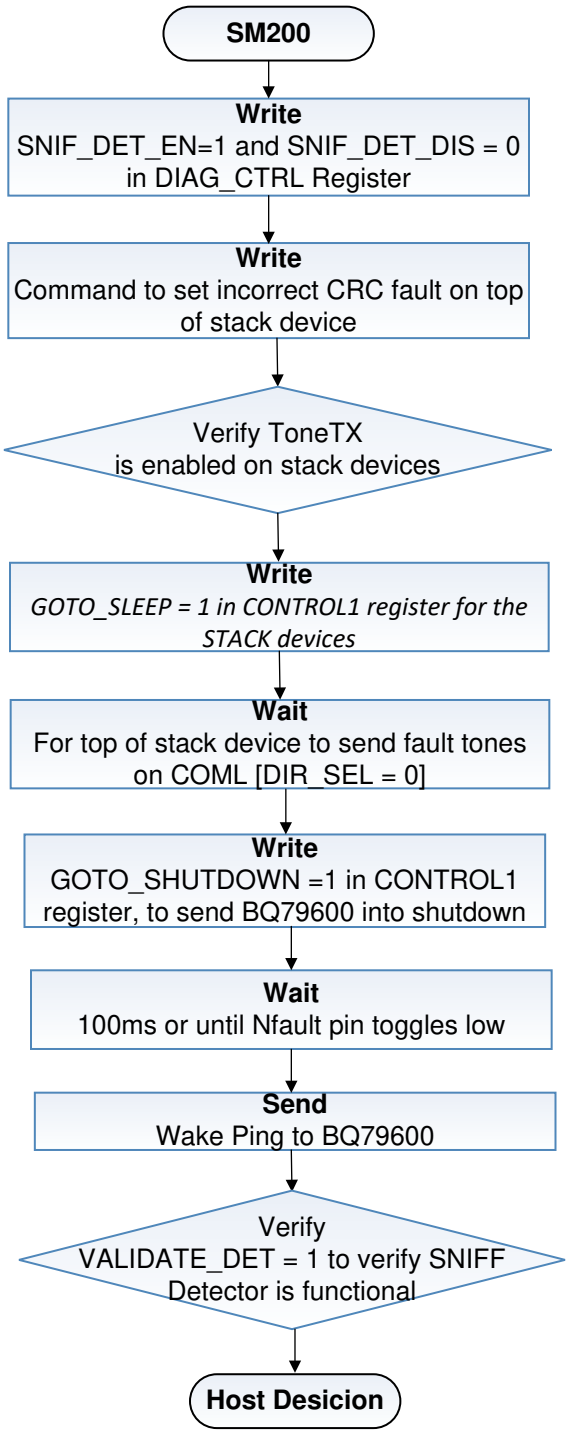

Figure 5-4. SM200: Snif Detector Diagnostic Flow Chart

[AoU22] — The host needs to run this diagnostic only if the Snif detector feature is used in the system.

[AoU23] — Sniff detector is only effective in SHUTDOWN mode. To enable the feature, the host MCU needs to set bit SNIFDET_EN = 1 & bit SNIFDET_DIS = ‘0 before transitioning to SHUTDOWN.

5.4.2 SM201: INH Pin Status Detection

The ASIC monitors the INH Pin status and sets the INH bit in register FAULT_SYS, when INH PMOS is activated.

5.4.3 SM202: INH Driver Diagnostic

To detect latent faults with the INH Driver, the MCU shall run the INH driver diagnostic test.

The host shall write the bit INH_SET_GO = 1 in register DIAG_CTRL and then monitor the status of bit INH_STAT = 1 to verify the INH driver is working.

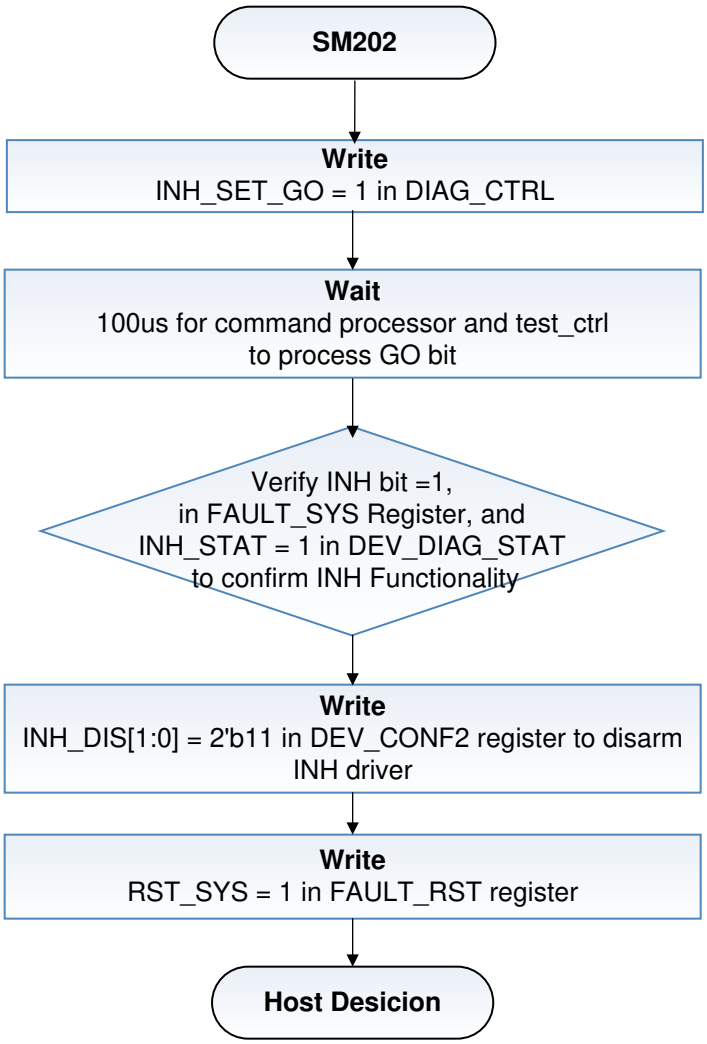

Figure 5-5. SM202: INH Driver Diagnostic Flow Chart

[AoU24] — The host MCU can can enable INH feature by setting INH_DIS[1:0] = 2'b00 and disable INH function by configuring bit INH_DIS[1:0] = 2'b11.

[AoU25] — To clear the fault, the host MCU needs to set bit INH_DIS[1:0] = 2'b11 (disarm INH driver), then write bit RST_SYS = 1. After this, to use INH feature, set INH_DIS[1:0] = 2'b00.

[AoU26] — If using INH driver -COMM fault(hb), PWR fault (CVDD OVUV), Register fault (bit flip) shall not be masked in Sleep Mode.

5.4.4 SM203 LFOSC Missing Clock Detection

The Low Frequency Oscillator is monitored by an independent LFO watchdog for clocking activity while the LFO is enabled. If the LFO does not transition high to low or low to high within allotted time the watch dog will reset the digital core and hold the digital core in reset state until the LFO watchdog signal is reset. The watch dog timer will reset and start a new timer period anytime the HFO clock transitions high to low or low to high.

Note

The LFO watchdog mechanism works continuously. Communications and voltage monitoring will stop while the digital core is in reset.

[AoU27] — The host MCU to verify the expected communication responses are received within the allotted time.

5.4.5 SM204: HFOSC Missing Clock Detection

The High Frequency Oscillator is monitored by an independent HFO watchdog for clocking activity while the HFO is enabled. If the HFO does not transition high to low or low to high within allotted time the watch dog will reset the digital core and hold the digital core in reset state until the HFO watchdog signal is reset. The watch dog timer will reset and start a new timer period anytime the HFO clock transitions high to low or low to high.

Note

The HFO watchdog mechanism works continuously. Communications and voltage monitoring will stop while the digital core is in reset.

[AoU27] — The host MCU to verify the expected communication responses are received within the allotted time.

5.4.6 SM205: LFOSC Frequency Mismatch Detection

The BQ79600 compares the LF oscillator frequency to the HF oscillator frequency using a counter. If the difference in the frequency is outside of the specified range, the LFO bit is set in register FAULT_SYS.

Note

While this detection indicates that the frequency difference between the two oscillator is out specification with respect to each other, it cannot identify which oscillator has drifted. HFO frequency drift will result in loss of UART communications.

[AoU10] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_SYS bit is 0.

5.4.7 SM206 Factory Register CRC Detection

The values in the factory nvm registers are checked cyclically in background by a CRC. The factory nvm CRC logic computes a checksum value from the factory register contents and compares it to the CRC checksum value stored in registers. If the stored CRC value and the calculated value do not match the FACT_CRC bit in the FAULT_OTP register is set.

Note

In the event the FACT_CRC bit is set the factory nvm registers values may be reloaded from the stored NVM memory by a device reset. After the reset the FACT_CRC bit shall be reset indicating the transient fault was reset.

[AoU28] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_REG bit is 0.

5.4.8 SM207: FACT CRC Diagnostic

An intentional fault may be injected into the factory nvm registers CRC check to diagnose operation of the CRC result comparison and the FACT_CRC fault bit. The FLIP_FACT_CRC bit in register DIAG_CTRL controls the diagnostic function. After the FLIP_FACT_CRC is set the FACT CRC status bit diagnostic shall fail setting the FACT_CRC bit in the FAULT_REG register.

[AoU2] — The host MCU conducts the diagnostic every MPFDI.

5.4.9 SM208: Customer Register Integrity Detection

To detect bit flips in customer registers, the ASIC monitors registers DEV_CONF1, DEV_CONF2 and FAULT_MSK.

The host MCU shall write the bit CONF_MON_GO = 1 on start up. The BQ79600 then takes a snap shot of the registers DEV_CONF1, DEV_CONF2 and FAULT_MSK and continously compares the register values to the snapshot values to detect bit flips. If the ASIC detects a bit flip it sets the CONF_MON_ERR bit in register FAULT_REG.

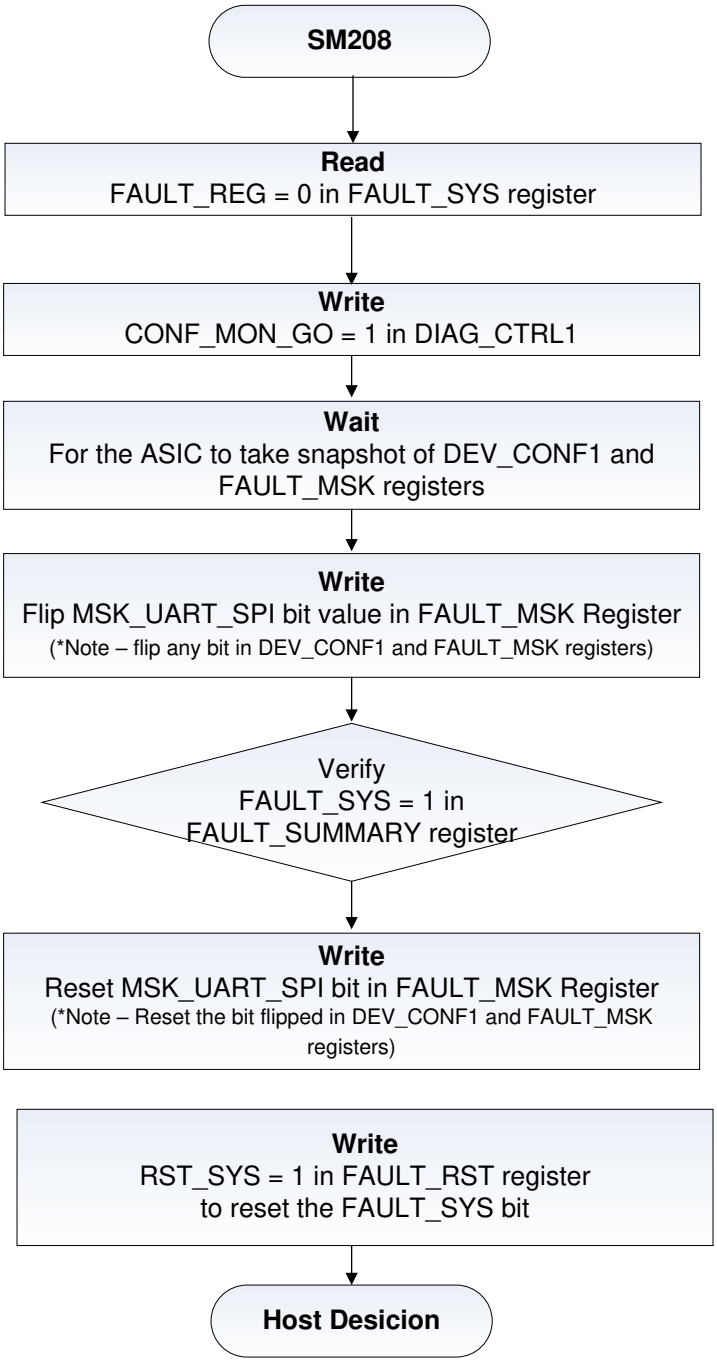

Figure 5-6. SM208: Customer Register Integrity Detection Flow Chart

[AoU28] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_REG bit is 0.

[AoU29] — When host MCU changes DEV_CONF1, DEV_CONF2 and FAULT_MSK register settings or any of the register bit flips, fault bit [CONF_MON_ERR] is set. Once host MCU changes the setting, it needs to write [CONF_MON_GO]=1 (resample 3 register values), write [RST_REG] =1 to clear the [CONF_MON_ERR] fault.

[AoU30] — After device reset (receive WAKE ping or SOFT_RESET = 1), the bit CONF_MON_ERR] = 0.

5.4.10 SM209: Customer Register Integrity Diagnostic

The host MCU shall periodically compare the customer register values to its expected customer register values to verify the data integrity of the customer registers.

[AoU2] — The host MCU conducts the diagnostic every MPFDI.

5.4.11 SM210: OTP Factory Load Error

OTP Factory Load Error indicates errors during the process of copying factory OTP to registers. This error detection happens automatically when the factory OTP data is transferred to registers. If an error is detected the FACTLDERR bit in register FAULT_REG is set.

[AoU28] — The host MCU reads the FAULT_SUMMARY register every FDTI to verify the FAULT_REG bit is 0.

5.4.12 SM211: Thermal Shutdown Detection

When the Thermal Shutdown sensor value is greater than the thermal shutdown temperature threshold the device is put into the SHUTDOWN mode automatically. There is no fault signaling done when a thermal shutdown event occurs. After a thermal shutdown event upon waking up the TSHUT bit in register FAULT_SYS is set indicating a thermal shutdown caused the device to enter the SHUTDOWN mode.

[AoU31] — After device transits from SHUTDOWN to ACTIVE mode, the host MCU to verify the TSHUT bit setting is '0'.

[AoU27] — The host MCU to verify the expected communication responses are received within the allotted time.

5.4.13 SM212: SHUTDOWN Status

The SHUTDOWN_REC bit in register FAULT_SYS is set 1 indicating the previous SHUTDOWN was caused by SHUTDOWN ping, or TSHUT which are not a usual SHUTDOWN method.

[AoU31] — After device transits from SHUTDOWN to ACTIVE mode, the host MCU to verify the SHUTDOWN_REC bit setting is '0'.

5.4.14 SM213: Fact Testmode Detection

The factory test mode shall be disabled at all times during normal operation. The test mode status is indicated by a non-zero value in register address 0x2601. A value of 0x00 in this register indicates that the device is in normal operating mode.

[AoU32] — The host MCU conducts the diagnostic every FDTI.

6 BQ79600 as Safety Element Out of Context (SEooC)

This section contains a Safety Element out of Context (SEooC) schematic of the BQ79600. Texas Instruments has made assumptions on the typical safety system configurations using this device. System-level safety analysis is the responsibility of the developer of these systems and not Texas Instruments. As such, this section is intended to be informative only to help explain how to use the features of the BQ79600 to assist the system designer in achieving a given ASIL level. Customers are responsible for putting this device into the context of their system and analyzing the ASIL coverage achieved therein. The BQ79600 has been designed to perform/ function in the ways described in this safety manual presuming that it is incorporated into a system that uses and interconnects the BQ79600 with other devices and elements as described. Note that the system designer may choose to use this BQ79600 in other safety-relevant systems.

6.1 BQ79600 - Typical Application Circuit

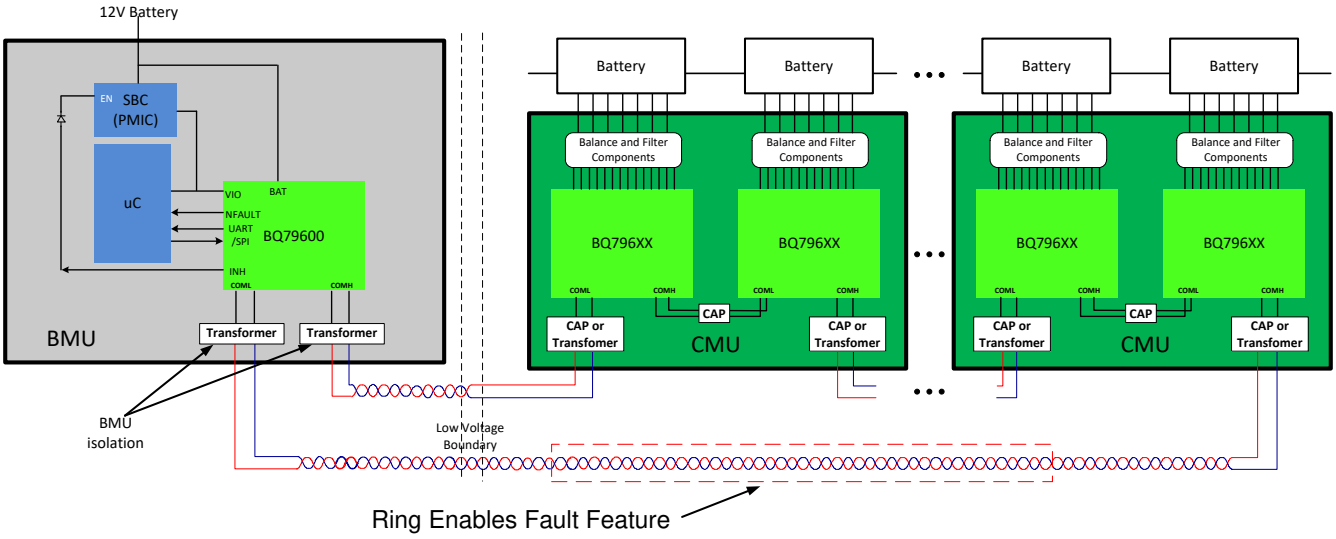

Figure 6-1. Typical Application Circuit

7 Revision History

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

Changes from Revision * (June 2020) to Revision A (January 2021)	Page
• Changed SM017 Diagnostic Interval from FDTI to MPFDI and Diagnostic/Detection from Detection to Diagnostic	11
• Deleted SM111	11
• Changed SM135 from MCU SPI Fault Detection to RX FIFO Overflow Detection	11
• Changed SM136 from SPI Conflict to MCU SPI Fault Diagnostics and Diagnostic Interval from FDTI to MPFDI and Diagnostic/Detection from Detection to Diagnostic	11
• Added SM137	11
• Deleted SM111 section	17
• Changed broadcast to stack in SM125 description	19
• Changed 0xE00 to 0x2601 in SM213 description	29

IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

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

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