Safety Class B with iMOTION™

Safety Class B with iMOTION™

About this document

Scope and purpose

The MCE software for the second generation of iMOTION™ products was designed in compliance with the safety standard IEC / UL60730-1 for automatic electronic controls. The software architecture includes the relevant components of Annex H, which describes the details of the tests and diagnostic methods to ensure safe operation of embedded control hardware and software for household appliances.

Intended audience

The audience of this application note are engineers of home appliance manufacturers who are integrating an iMOTION™ product into their system design in compliance with IEC / UL60730-1 Class B classification.

Table of contents

About this document **1**
Table of contents **1**
1 Nomenclatures **4**
2 Class B Coverage **5**
3 Safety Precautions **6**
4 Introduction **7**
4.1 Purpose of the Control 7
4.1.1 Motor Control 7
4.1.2 PFC 8
4.2 Fault Reaction Time 10
4.3 Traceability 12
5 Controller Safety **13**
5.1 Construction 13
5.1.1 Measures to Control Faults 13
5.1.2 Redundant Memory 13
5.1.3 Software Protection 13
5.1.4 Fault Response 13
5.1.5 Reset Actions 14
5.2 Power On Self Test 14
5.2.1 WDT Test 15
5.2.2 IRQ Test 15
5.2.3 CLK Test 15
5.2.4 CPU Test 16
5.2.5 PC Test 16
5.2.6 RAM Test Parity 16
5.2.7 AD Test 16
5.2.8 RAM Test 16

5.2.9 FLASH Test ... 16
5.3 Built In Self Tests ... 17
5.3.1 Safety Task Handler ... 17
5.3.2 Execution Monitoring & IRQ ... 18
5.3.3 BIST CLK ... 21
5.3.4 BIST FLASH ... 21
5.3.5 BIST RAM ... 21
5.3.6 BIST STACK ... 21
5.3.7 BIST IO ... 21
5.3.8 BIST WDT ... 21
5.3.9 BIST CPU Load ... 22
5.4 External Communication ... 22
5.4.1 CRC16 ... 22
5.4.2 Link Break Protection ... 22
6 Parameter Safety ... 23
6.1 Parameter Handler ... 23
6.1.1 Value Range Control Mechanism ... 23
6.1.2 Access Control Mechanism ... 23
6.2 Parameter Loader ... 23
7 Functional Safety ... 24
7.1 Introduction ... 24
7.1.1 Combined actions ... 24
7.2 PFC Protection ... 24
7.2.1 AC Input Frequency Protection ... 24
7.2.2 AC Input Over / Under-Voltage Protection ... 24
7.2.3 PFC Over-Current Protection ... 24
7.3 Motor Control Protection ... 25
7.3.1 Motor Phase Loss Protection ... 25
7.3.2 Motor Over-Current Protection ... 25
7.3.3 Rotor Lock Protection ... 27
7.3.4 Flux PLL Out-Of-Control Protection ... 27
7.3.5 Invalid Hall Protection ... 28
7.3.6 Hall Timeout Protection ... 28
8 Using Safety Class B Funcion ... 30
9 Failure Mode and Effects Analysis ... 32
9.1 Adjacent Pin Short / Open / Stuck Analysis ... 32
9.1.1 Motor and PFC PWM Output Pins ... 32
9.1.2 Pins Related to Protection Mechanisms ... 32
9.1.2.1 NTC Input Pin ... 32
9.1.2.2 VDC Input Pin ... 32
9.1.2.3 IPFCTRIP Input Pin ... 33
9.1.2.4 GK Input Pin ... 33
9.1.2.5 IU, IV, IW Input Pins ... 33
9.1.2.6 ISS input pin ... 33
9.1.3 Control Interface Pins ... 33
9.1.3.1 UART Interface ... 33
9.1.3.2 VSP ... 34
9.1.3.3 DUTYFREQ ... 34
9.1.3.4 DIR ... 34

9.1.3.5 LED ... 34
9.1.3.6 PAR0..3 / PARAM ... 35
9.1.4 GPIOs and AIN pins ... 35
9.2 Critical Parameter Wrong Value Setting Analysis ... 35
9.2.1 Motor Control Parameter ... 36
9.2.1.1 Class B Functions ... 36
9.2.1.2 Control Performance and Efficiency ... 36
9.2.1.3 Control Mode and Target Values ... 37
9.2.1.4 High Current ... 37
9.2.1.5 DC-Bus Voltage Monitoring ... 37
9.2.1.6 Over-Temperature Detection ... 37
9.2.1.7 Startup failure ... 38
9.2.1.8 Faulty Current Measurement ... 38
9.2.1.9 Control Interface ... 38
9.2.2 PFC Control Parameter ... 38
9.2.2.1 Class B ... 38
9.2.2.2 Control Performance ... 39
9.2.2.3 Functional Deterioration ... 39
9.2.2.4 DC-Bus Voltage Monitoring ... 39
9.2.2.5 Control Interface ... 39
9.3 System Performance Effect and Analysis ... 40
9.3.1 Class B POST ... 40
9.3.1.1 Power-up Diagnostics Error ... 40
9.3.2 Class B BIST ... 40
9.3.2.1 CPU Overload ... 40
9.3.2.2 Flash Memory Error ... 40
9.3.2.3 RAM Memory Error ... 40
9.3.2.4 Watchdog Error ... 40
9.3.2.5 Clock Error ... 40
9.3.2.6 ADC Error ... 41
9.3.2.7 DAC Error ... 41
9.3.2.8 Safety Handler ... 41
10 References ... 42
11 Appendix ... 43
Revision history ... 46

1 Nomenclatures

Fault Reaction Time: time between the occurrence of a fault and the point where the control has reached a defined state [1]

iMOTION™: digital motor control product trademark of Infineon Technologies AG.

PMSM: permanent magnet synchronous motor

FOC: field oriented control

PFC: power factor correction

PWM: pulse width modulation

MCE: motion control engine used in iMOTION™ products

March C-: a commonly used algorithm for memory test.

POST: Power On Self Test

BIST: Built In Self Test

IRQ: Interrupt Request

ISR: Interrupt Service Routine

WDT: Watchdog

PC: Program Counter

AD: Address Decoder

2 Class B Coverage

All iMOTION™ products featuring the MCE firmware versions listed below are certified by UL to be in compliance with UL / CSA 60730-1 under file number E498177. Users can find more details of certified iMOTION™ MCE software library from official UL website.

Table 1 Class B Coverage

MCE FW Version	iMOTION™ Products
V1.01.05	IMC101T-T038, IMC101T-Q048, IMC101T-F064
	IMC02T-F064
V1.03.03 / V1.03.07	IMC101T-T038, IMC101T-F048, IMC101T-Q048, IMC101T-F064
	IMC102T-F064
	IMC301A-F048, IMC301A-F064
	IMC302A-F048, IMC302A-F064
	IMD111T-6F040
	IMD112T-6F040
	IMM101T-015M, IMM101T-046M, IMM101T-056M
	IMM102T-015M, IMM102T-046M, IMM102T-056M
V5.03.00.6.XXXX	IMC101T-T038, IMC101T-F048, IMC101T-Q048, IMC101T-F064
	IMC102T-F048, IMC102T-F064
	IMC301A-F048, IMC301A-F064
	IMC302A-F048, IMC302A-F064
	IMD111T-6F040
	IMD112T-6F040
	IMM101T-015M, IMM101T-046M, IMM101T-056M
	IMM102T-015M, IMM102T-046M, IMM102T-056M
	IMI111T-026H, IMI111T-046H

3 Safety Precautions

Table 1 Precautions

 **Fault Reaction Time**	***Attention:*** ***If user attempts to touch the moving part of a mechanical system with a motor within the specified fault reaction time, it may involve unexpected movement due to pending energized system conditions. Failure to observe this may result in bodily injury.***
 **Electrical Shock**	***Attention:*** ***Any variable speed drive and / or inverter system based on iMOTION™ controllers may have energized DC bus capacitors which still remain charged after removing energy source. Verification of remaining energy is required prior to diagnose the system. Failure to do this may result in bodily injury.***
 **Enable Class B Option**	***Attention:*** ***Parameter 'SafetyEnable' must be set to `0xCB34` to enable Class B option prior to utilizing safety related features by the UL/CSA 60730-1 standard. Failure to do so may result in bodily injury.***
 **CPU Overload**	***Attention:*** ***CPU overload condition could occur by the configuration of Class B related parameters such as 'SafetyEnable' and 'SysTaskTime' as well as PWM carrier frequencies of motor and PFC. CPU loading should be monitored for motor and PFC control functions + Class B related safety tasks. Failure to do so may result in unexpected motor behavior and bodily injury.***

4 Introduction

Annex H of IEC / UL60730-1 standard defines 3 software classifications for automatic electronic controls [1]:

- Class A – Control functions which are not intended to be relied upon for the safety of the application;
- Class B – Control functions which are intended to prevent an unsafe state of the appliance;
- Class C – Control functions which are intended to prevent special hazards such as explosion or whose failure could directly cause a hazard in the appliance.

According to the standard [1], a manufacturer of automatic electronic controls shall design its Class B software using one of the following structures:

- Single channel with functional test;
- Single channel with periodic self-test;
- Dual channel without comparison.

The iMOTION™ software was designed in compliance with Class B classification using the single-channel structure with functional tests as well as periodic self-tests.

Table H.1 (Acceptable measures to address fault / errors) in IEC / UL60730-1 [1] specifies that for control with software Class B, the manufacturer shall provide within the control measures to address the fault / errors in safety-related segments and data. iMOTION™ MCE includes software modules to address those controller safety related requirements. Please refer to Table 5 in Appendix for details of acceptable measures to address fault / errors for Class B. Table 5 also includes a cross-reference column to indicate what specific modules provided by iMOTION™ MCE software are designed to address those fault / errors accordingly.

In addition to that, iMOTION™ MCE software also provides additional measures to address parameter safety as well as functional safety related faults regarding to the operational control of the motor and PFC. The complete list of Class B related software modules provided by iMOTION™ MCE software can be found in Table 6 in Appendix.

Conformity tests under supervision of UL have been performed. The certified products are listed on the web site. Using Class B compliant iMOTION™ software helps the manufacturers of automatic electronic controls simplify the process of qualifying their system design with IEC / UL60730-1 Class B standard.

4.1 Purpose of the Control

The iMOTION™ software contains two electronic control actions:

- The control of a PMSM
- The control of a PFC circuitry

4.1.1 Motor Control

The motor control action is configurable and drives a motor with one of the following control schemes:

- Open loop voltage control
- Closed loop sensorless FOC with cascaded speed control
- Closed loop Hall Effect sensor based FOC with cascaded speed control

The following Figure 1 shows a typical sensorless FOC block diagram, where flux estimator & PLL shall be replaced by an encoder interface for Hall effect sensor based FOC block diagram.

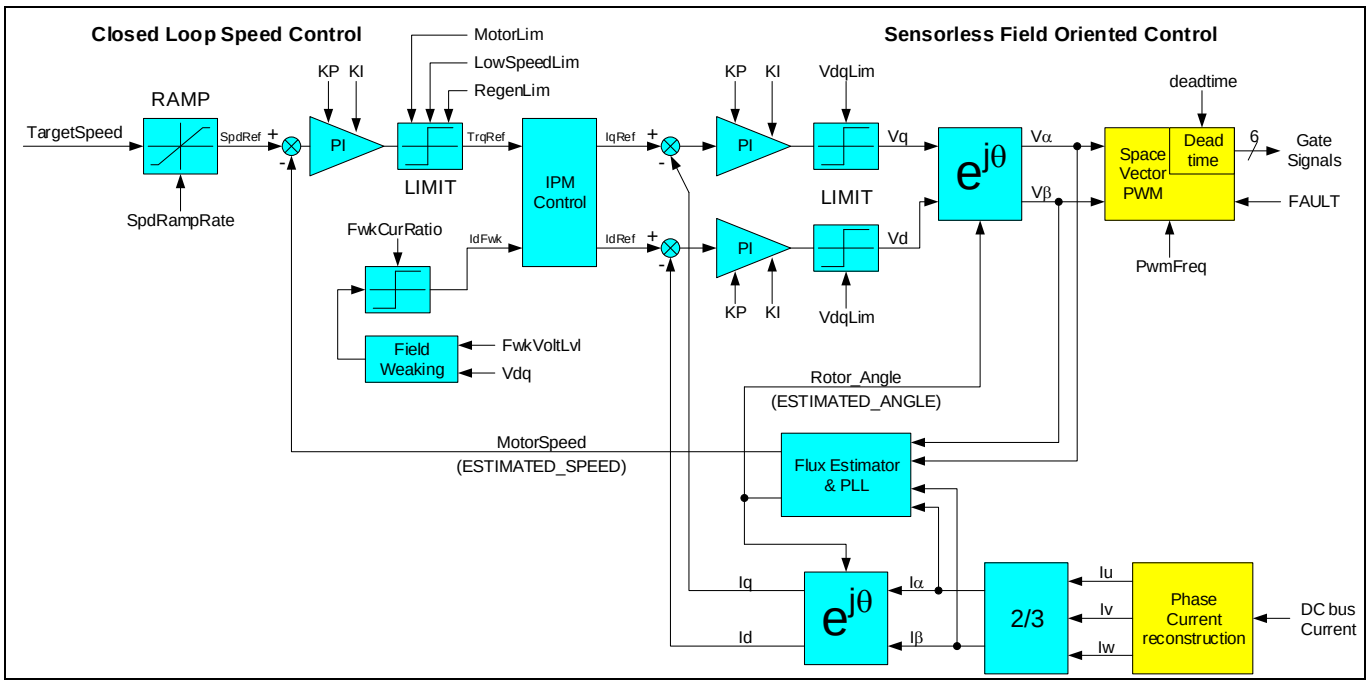

Figure 1 Sensorless FOC Block Diagram

4.1.2 PFC

The PFC control is configurable with one of the following topologies:

- Single switch boost PFC (Figure 2)
- Dual switch totem pole PFC (Figure 3)

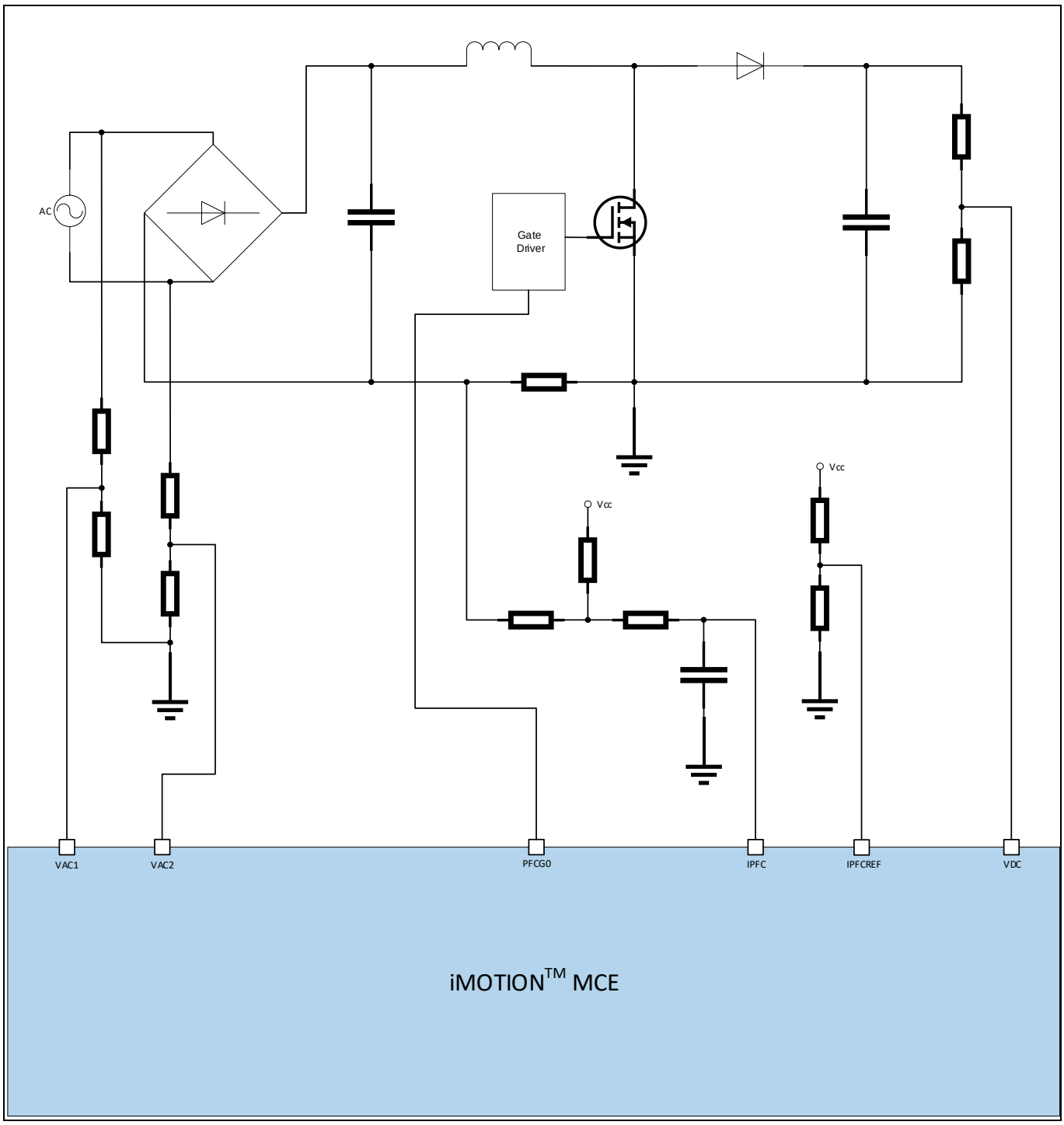

Figure 2 Single Switch Boost PFC Block Diagram

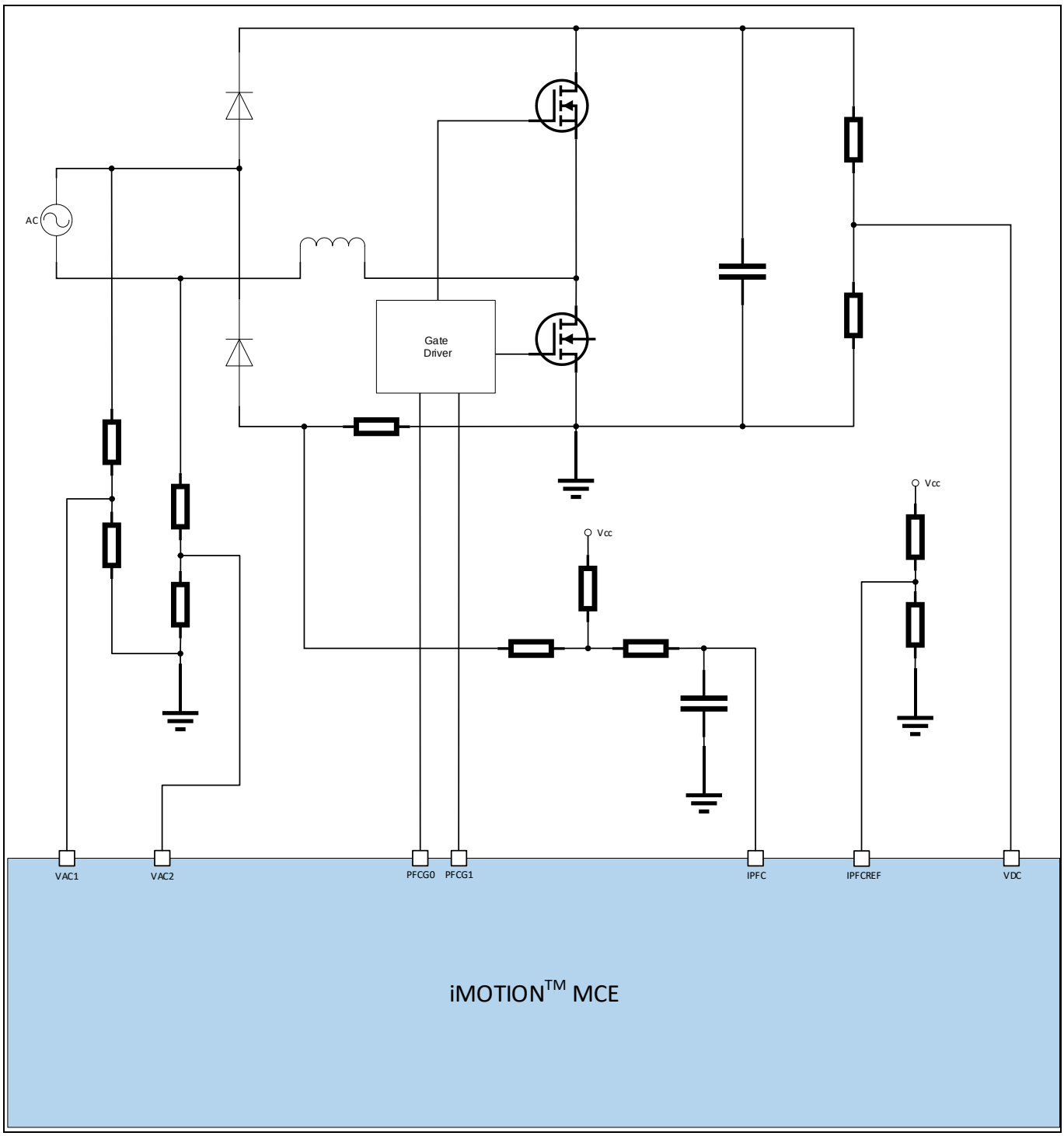

Figure 3 Dual Switch Totem Pole PFC Block Diagram

4.2 Fault Reaction Time

The fault reaction time depends on the timing configuration of several safety modules. The following Table 2 lists all relevant configuration parameters and their influences on the fault reaction time.

Table 2 Fault Reaction Time Dependencies

Safety Module	Parameter	Fault Reaction Time Dependencies
POST	SafetyEnable	
BIST	SafetyEnable SysTaskTime	$t_{Fault\ Reaction} = SysTaskTime \cdot 1000\ ms$ $SysTaskTime = 1\ (typical)$ $t_{Fault\ Reaction} = 1\ second\ (typical)$
UART Link Break Protection	FaultEnable[13] LinkBreakCount SysTaskTime	$t_{Fault\ Reaction} = LinkBreakCount \cdot SysTaskTime\ (ms)$ $LinkBreakCount = 10000\ (typical)$ $SysTaskTime = 1\ (typical)$ $t_{Fault\ Reaction} = 10\ seconds\ (typical)$
Motor Flux PLL Out-of-Control Protection	FaultEnable[4] PLL_OutSyncTime	$t_{Fault\ Reaction} = PLL_OutSyncTime \cdot 10\ ms$ $PLL_OutSyncTime = 800\ (typical)$ $t_{Fault\ Reaction} = 8\ seconds\ (typical)$
Motor Rotor Lock Protection	FaultEnable[7] RotorLockTime	$t_{Fault\ Reaction} = RotorLockTime \cdot 10\ ms$ $RotorLockTime = 1000\ (typical)$ $t_{Fault\ Reaction} = 10\ seconds\ (typical)$
Motor Over-Current Protection	GateKillFilterTime	$t_{Fault\ Reaction} = GateKillFilterTime \cdot 10.417\ ns + 231\ ns$ $GateKillFilterTime = 96\ (typical)$ $t_{Fault\ Reaction} < 2\ \mu s\ (typical)$
Motor Phase Loss Protection	FaultEnable[8] SysTaskTime	$t_{Fault\ Reaction} = 2 \cdot SysTaskTime\ (ms)$ $SysTaskTime = 1\ (typical)$ $t_{Fault\ Reaction} = 2\ ms\ (typical)$
Motor Invalid Hall Protection	FaultEnable[15] SafetyTaskTime	$Invalid\ pattern: t_{Fault\ Reaction} = T_{Hall_event} \cdot 3 + SafetyTaskTime(ms)$ $Wrong\ expected\ pattern: t_{Fault\ Reaction} = T_{Hall_event} \cdot 2 + SafetyTaskTime(ms)$ $SafetyTaskTime = 16\ ms\ (typical)$
Motor Hall Timeout Protection	FaultEnable[14] HallTimeoutPeriod SafetyTaskTime	$t_{Fault\ Reaction} = 4096 \cdot T_{Base\ Rate} + HallTimeoutPeriod \cdot SafetyTaskTime(ms)$ $SafetyTaskTime = 16\ ms\ (typical)$
Motor Current Offset Protection	FaultEnable[9] OffsetSample SysTaskTime	$t_{Fault\ Reaction} = T_{Base\ Rate} \cdot 2^{OffsetSample} + SysTaskTime\ (ms)$ $OffsetSample = 13\ (typical)$ $SysTaskTime = 1\ (typical)$
PFC Over-Current Protection	PFC_GateKillTime	$t_{Fault\ Reaction} = PFC_GateKillTime \cdot 10.417\ ns + 85\ ns$ $PFC_GateKillTime = 48\ (typical)$ $t_{Fault\ Reaction} < 1\ \mu s\ (typical)$
AC Input Over-Voltage Protection	FaultEnable[5] SysTaskTime	$t_{Fault\ Reaction} = One\ Line\ Cycle\ (ms) + SysTaskTime\ (ms)$ $t_{Fault\ Reaction} < 100\ ms\ (typical)$
AC Input Under-Voltage Protection	FaultEnable[4] SysTaskTime	$t_{Fault\ Reaction} = One\ Line\ Cycle\ (ms) + SysTaskTime\ (ms)$ $t_{Fault\ Reaction} < 100\ ms\ (typical)$
AC Input Frequency Protection	SysTaskTime	$t_{Fault\ Reaction} = One\ Line\ Cycle\ (ms) + SysTaskTime\ (ms)$ $t_{Fault\ Reaction} < 100\ ms\ (typical)$

4.3 Traceability

A unique hash code stamp was generated per internal software module version for traceability. In order to maintain the upward compatibility for any future release, key modules among listed modules are kept unchanged. All listed modules are included in the current MCE software version found on the Infineon Website: MCE_Software (V1.01.05, V1.03.03, V1.03.07, and V5.03.00.6.XXXX)

5 Controller Safety

5.1 Construction

5.1.1 Measures to Control Faults

- Single channel with functional test (H.2.16.5)
- Single channel with periodic self-test (H.2.16.6)
- Single channel with periodic self-test and monitoring (H.2.16.7)

5.1.2 Redundant Memory

When redundant memory with comparison is provided on two areas of the same component, the data in one area is stored in a different format from that in the other area.

5.1.3 Software Protection

The software is protected from user alteration of safety-related segments and data. Software updates provided by the manufacturer and transmitted to the control are checked prior to its use regarding to the following items:

- Data corruption through communication ensuring Hamming distance = 3 for software class B. (Refer to Table H.1 [1] for external communication.)
- Software compatibility with the hardware product.

Additionally, the software which performs the abovementioned checks contains measures to control the fault / error conditions specified in H.11.12.2.

5.1.4 Fault Response

As shown in the following Figure 4, iMOTION™ MCE is designed to operate in one of the following five operation modes: Secure Bootstrap Loader (SBSL) Mode, Config Mode (CD), Application Mode(AD), Standby Mode (A5), and Failsafe Mode (AF). By default, the MCE starts up in SBSL Mode, waiting for the MCE firmware to be programmed. Once the appropriate firmware is programmed and verified, the MCE would reboot. After the reboot, the MCE would go through POST sequence first to perform self tests if safety Class B function is enabled. If one of the POST tests fails, then the MCE would be trapped in an infinite loop until reset. If the POST test goes through successfully, then the MCE would go to Config Mode if the parameters are not programmed. Otherwise, the MCE would go to Application Mode where BIST tests are being performed periodically in the background. If there occurs any BIST fault, then the MCE would request entering Failsafe Mode and initiate a reboot. After the reboot, the MCE would go through POST sequence first. Should any POST test fail, the MCE would be trapped in an infinite loop until reset. If there appears no POST fault, then the MCE would go into Failsafe Mode in which peripherals are reset, outputs are set to predefined values. The MCE would stay in Failsafe Mode indefinitely if restart time is not specified. If restart time is specified, then the MCE would reboot after the specified amount of time.

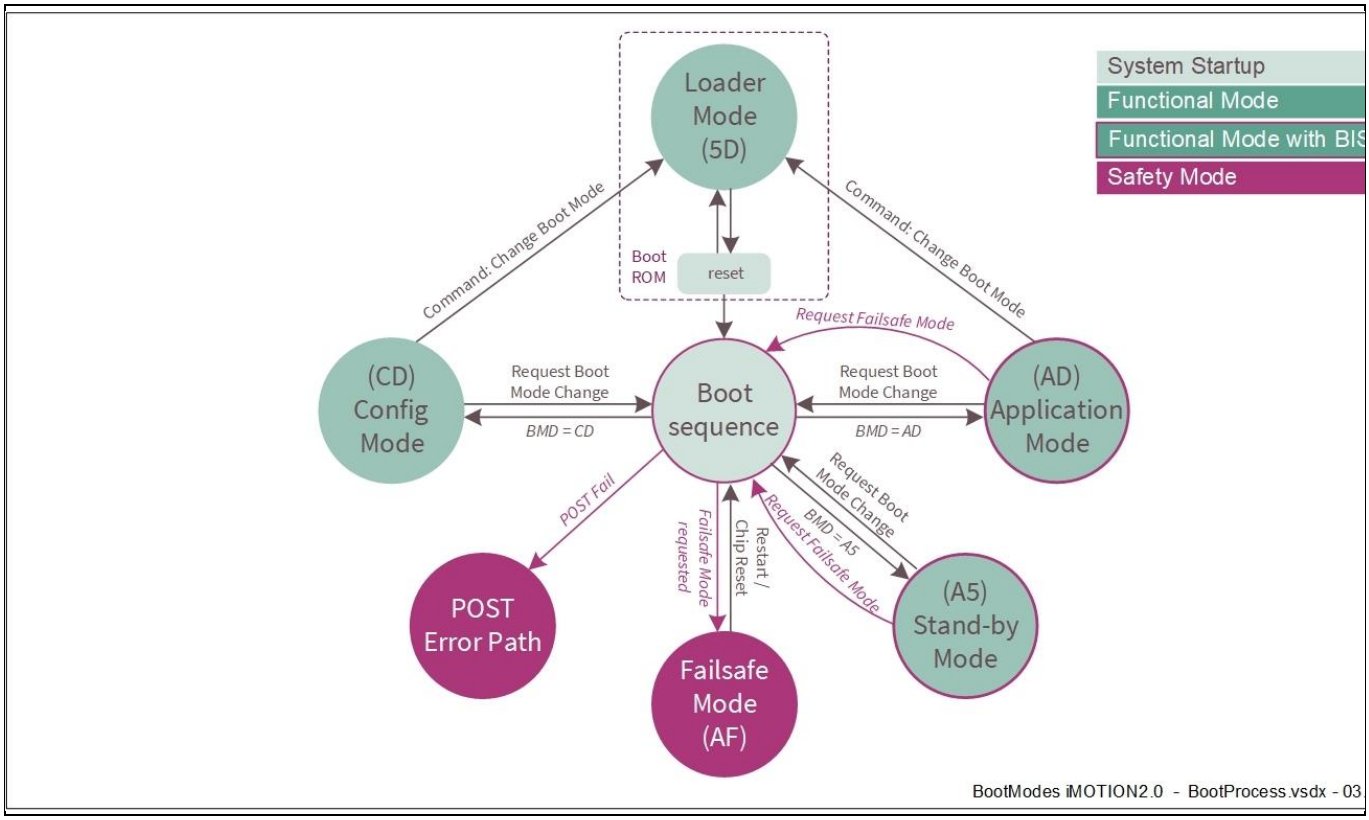

Figure 4 MCE Operation Modes

5.1.5 Reset Actions

Maximum number of reset actions within a time period (H.11.12.4.3.6, H.11.12.4.3.4)

5.2 Power On Self Test

Power-On Self Tests (POST) are implemented in order to ensure a safe startup of the system. This startup is implemented in two steps as shown in the following Figure 5.

First the system checks the integrity of the startup software initialization data which includes stack pointer address, reset handler address at which the reset vector points, and the clock initialization. The integtrety check compares the sum of this data with the checksum value in the FLASH memory.

Second, if safety Class B function has been enabled (`SafetyEnable = 0xCB34`), then all POST test routines are executed. If a POST test routine reports a fault, then the execution immediately enters the error path and waits forever (trapped in an infinite loop).

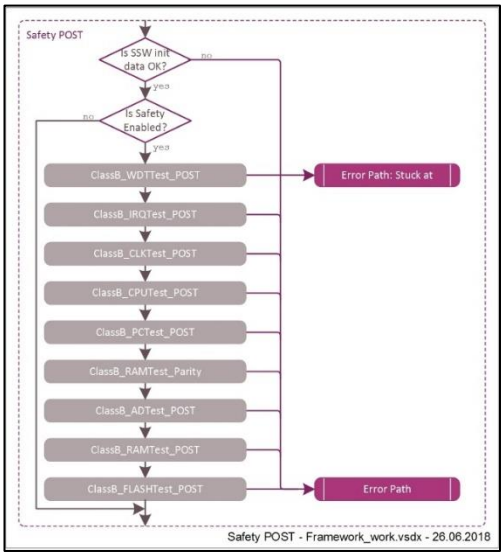

Figure 5 Safety POST Sequence

These POST test routines are described in the following sub-sections.

5.2.1 WDT Test

This is the first test performed during POST. The watchdog timer and the reset mechanism are tested by configuring the watchdog and then waiting for a reset. If a restart due to the watchdog test is detected, then the test is considered as passed.

The following mechanisms are implemented:

- IEC / UL 60730-1 Reference Hardware:
  - H.2.18.3 comparator
- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test
  - H.2.18.10.4 time-slot monitoring of the programme sequence

5.2.2 IRQ Test

The IRQ POST verifies the priority logic for a set of interrupts with different priorities.

- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test
  - H.2.18.10.4 time-slot monitoring of the programme sequence

5.2.3 CLK Test

The clock test ensures that the internal clock sources are functioning correctly by checking the internal clock sources as well as the oscillator watchdog state.

- IEC / UL 60730-1 Reference Hardware:
  - H.2.18.10.1 frequency monitoring
- IEC / UL 60730-1 Reference Software:
  - H.2.18.10.4 time-slot monitoring of the programme sequence

5.2.4 CPU Test

The CPU test ensures that there are no stuck or transition faults in any of the CPU registers.

- IEC / UL 60730-1 Reference Hardware:
  - H.2.18.9 internal error detection
- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test

5.2.5 PC Test

The Program Counter test verifies the addressing mechanism of the CPU core. For this test several different sub-routines are called. The correct execution of the sub-routines is verified by unique return codes and a global counter. The sub-functions are located in separate Flash sections, which are separate from the caller, to ensure that the compiler generates a branch instruction.

- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test

5.2.6 RAM Test Parity

The RAM (volatile memory) provides a safety mechanism in hardware (parity check) which will be tested and enabled by this POST routine.

- IEC / UL 60730-1 Reference Hardware:
  - H.2.18.9 internal error detection
- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test

5.2.7 AD Test

The address decoder is checked to ensure that it has full access to all memory locations and that there are no stuck address lines.

- IEC / UL 60730-1 Reference Hardware:
  - H.2.18.9 internal error detection
- IEC / UL 60730-1 Reference Software:
  - H.2.16.5 single channel with functional test

5.2.8 RAM Test

The RAM test calls an optimized March C- test algorithm for the entire RAM.

- IEC / UL 60730-1 Reference Software:
  - H.2.19.6.2 marching memory test

5.2.9 FLASH Test

The Flash test calculates the CRC32 of the code section in the flash and compares it with the actual value from the CRC table in the flash. The CRC table is located in a flash segment which is excluded from the CRC recalculation.

- IEC / UL 60730-1 Reference Software:
  - H.2.19.4.2 CRC – double word

5.3 Built In Self Tests

During the normal program execution, additional Built-In Self Tests (BIST) are executed. Only if the safety option has been enabled (`SafetyEnable = 0xCB34`), then each self test function is registered at startup as a separate safety task and executed by the safety task handler. Typical CPU usage is approximately 5% when the safety option is enabled based on default configuration.

The safety task handler is executed with the same frequency as the system tasks, which are defined by parameter `SysTaskTime` in millisecond. It executes one BIST task call per handler step. A complete BIST cycle is finished when all BIST tasks have been called with the predefined number of repetitions (weight number).

The safety task ID, number of calls per safety BIST cycle, as well as the worst case and best case number of calls for completion are listed in Table 3.

Table 3 BIST Safety Tasks

BIST Safety Task (ID)	Number of calls per BIST cycle (Weight number)	Worst case number of calls for completion	Best case number of calls for completion
BIST_CLK(1)	1	1	1
BIST_FLASH(2)	12	597	596
BIST_RAM(3)	4	197	196
BIST_STACK(4)	1	1	1
BIST_IO(5)	1	40	37
BIST_WDT(6)	1	50	50

5.3.1 Safety Task Handler

The safety tasks are executed in runtime in a dedicated safety task list which is separated from the functional task lists. The safety task list is handled in a round-robin manner. The following Figure 6 shows the safety task handler block diagram.

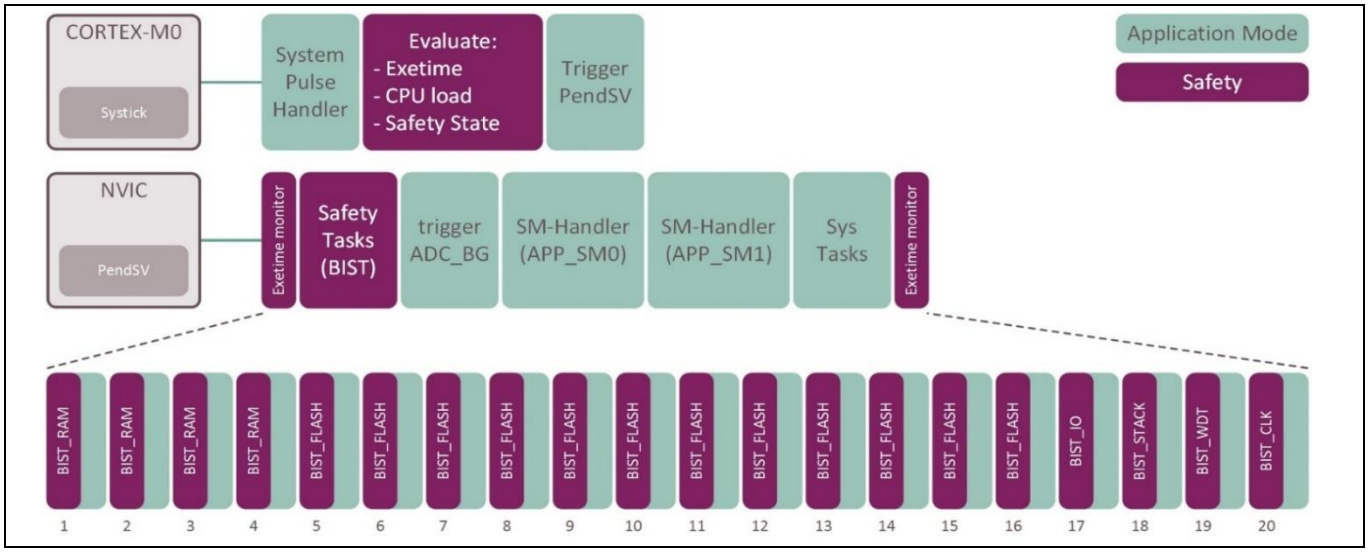

Figure 6 Safety Task Handler Block Diagram

The safety task handler executes one safety task in each system task cycle. It is the first function call in each cycle before the functional tasks are executed. After execution, the response as well as the number of calls for completion of the safety task is evaluated. A safety task responds as OK, COMPLETE or with individual fault response value.

- OK indicates a proper finalization of the task.
- COMPLETE indicates a proper completion of the task and triggers the worst case / best case execution time evaluation. The execution time evaluation is performed by comparing the actual execution time against the worst case and best case values of the finalized task. In case of a violation, the system enters failsafe mode.
- Any other response value causes the system to enter failsafe mode.

  The fault reaction time can be calculated by taking the worst case execution time of a safety task and dividing this number with the weight number of this task. The result is the number of BIST rounds. This is then multiplied with the number of safety task calls per BIST cycle, which is the sum of the weight numbers of all the registered safety tasks. The result is adjusted by parameter 'SysTaskTime' to get fault reaction time. For example, for BIST_FLASH safety task, its worst case execution time is 597, and its weight number is 12. Given the number of tasks per BIST cycle is 20 and SysTaskTime is 1, then the maximum fault reaction time is 597 / 12 * 20 * 1 * 1 ms = 997 ms. Using this method, the worst case fault reaction time of other BIST safety tasks can be estimated.

5.3.2 Execution Monitoring & IRQ

The real-time scheduler provides six parallel executed program sequences as shown in the following Figure 7. These are the two timer based state machine tasks (APP_SMx Task0 for motor control fast loop, and APP_SMy Task1 for PFC control fast loop), and their corresponding primary control tasks (APP_SMx Task1 for motor control primary loop, and APP_SMy Task1 that is not being used), the system and safety tasks, as well as the background tasks. They are monitored on maximum execution time and the peak values are stored in order to detect and control any fault in any of these program sequences. Servicing the WDT is one of the safety tasks.

Peak value evaluation:

- All peak values are compared to their maximum limit;
- The CPU load is compared to its maximum limit;
- In case of any violation, failsafe mode is entered.

The following Table 4 lists the maximum limits of execution for differenet tasks that are being monitored.

Table 4

Task	Maximum Limit
Motor control fast loop ISR task	90% of task update cycle
Motor control primary loop ISR task	75% of task update cycle
PFC control fast loop ISR task	90% of task update cycle
PendSV ISR task	95% of task update cycle
CPU load	96%

IEC / UL 60730-1 Reference Software:

- H.2.18.10.2 logical monitoring of the programme sequence
- H.2.18.10.4 time-slot monitoring of the programme sequence
- H.2.18.10.3 time-slot and logical monitoring (a combination of H.2.18.10.2 and H.2.18.10.4)

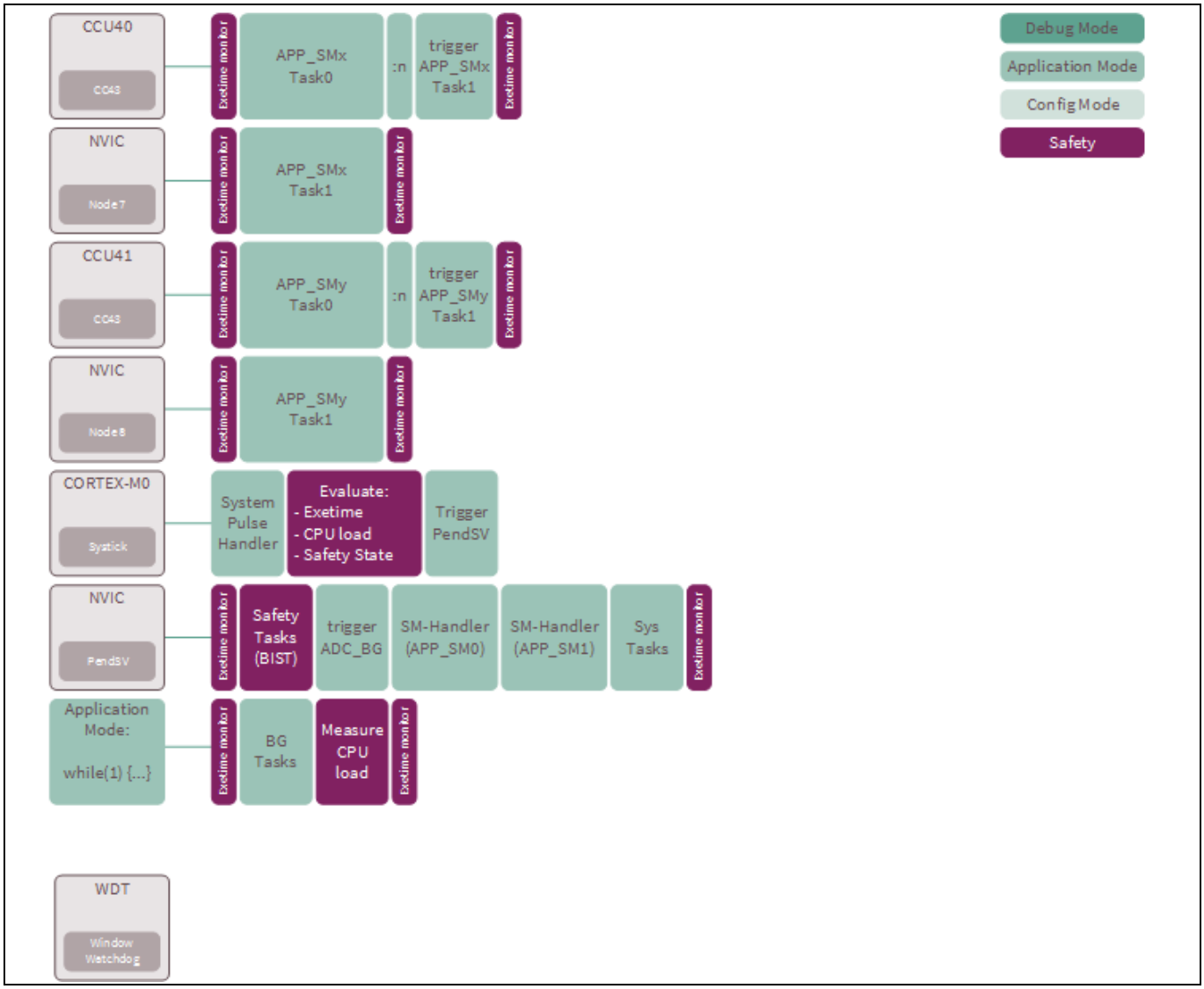

Figure 7 Real-Time Scheduler Structure

The motor control fast loop ISR task gets updated every motor PWM cycle. And the motor control primary loop ISR task's update cycle is an integral multiple of the motor PWM cycle. The PFC control fast loop ISR task is updated every PFC PWM cycle. Increasing the motor carrier frequency or PFC switching frequency might end up causing interrupts not being able to be serviced in time, thus resulting in violation of maximum limit of execution and triggering Failsafe Mode.

MCEWizard [4] can be used to estimate the CPU usage. If the CPU usage estimation using MCEWizard is higher than 90% as shown in the following Figure 8 with safety Class B function enabled, then system is likely to enter Failsafe Mode. It is highly recommended to keep the CPU usage estimation to no more tha 90% when the users intend to enable safety Class B function.

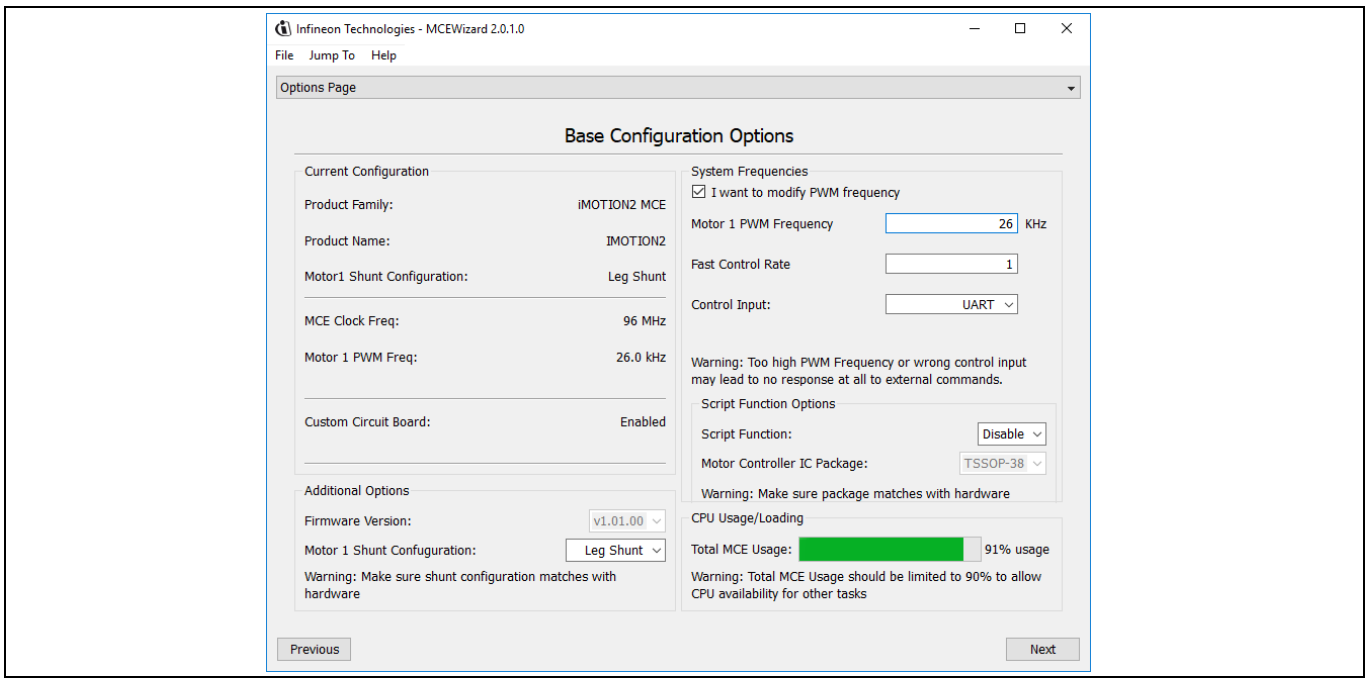

Figure 8 CPU Usage Estimation Using MCEWizard

The actual CPU load status can be obtained by reading the system parameter ‘CPU Load’ [2] using MCEDesigner [3]. The CPU load is represented in 0.1% [2]. The following Figure 9 shows an example of using MCEDesigner to read out ’CPU Load’ parameter when the system was running.

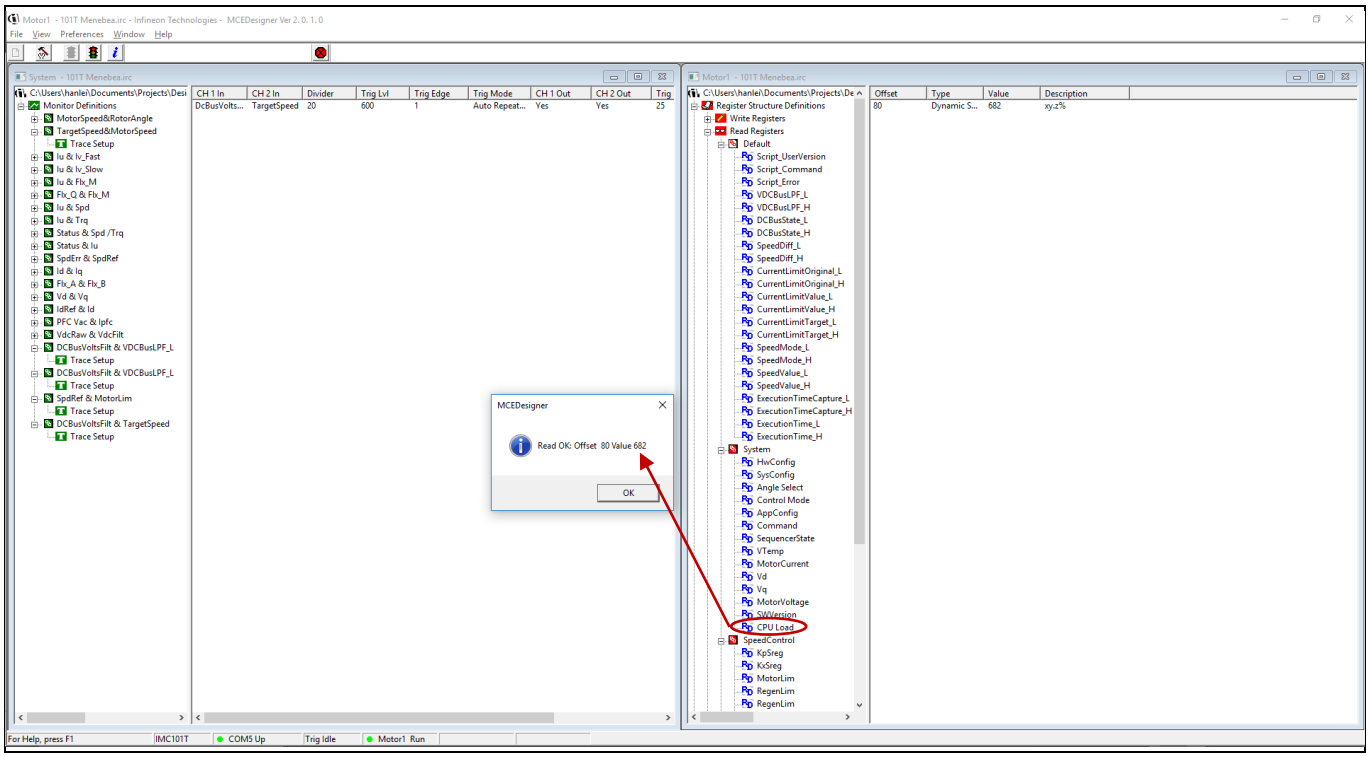

Figure 9 Reading ‘CPU Load’ Parameter Using MCEDesigner

5.3.3 BIST CLK

The clock test ensures that the internal clock sources and their monitoring features are functioning correctly.

5.3.4 BIST FLASH

The FLASH (non-volatile) memory of the iMOTION™ controller device is equipped with both, a parity check and an Error Correction Code (ECC) mechanism. The error correcting code (ECC) for the data blocks is a one-bit error correction and detection as well as partial double-bit error detection.

The BIST FLASH task scans the entire flash memory in small portions per call and tests the ECC hardware on proper functionality once per completion cycle.

5.3.5 BIST RAM

The RAM (volatile) memory of the iMOTION™ controller device is equipped with a parity check mechanism. It supports 8-bit, 16-bit and 32-bit writes, and generates one parity bit for each 8 bits of written data. A read operation will check for parity errors on the 32-bit read data.

The BIST RAM task scans the entire RAM memory in small portions per call and tests the parity hardware on proper functionality.

5.3.6 BIST STACK

The BIST STACK task monitors the stack pointer on overrun and underrun conditions.

5.3.7 BIST IO

The BIST IO task performs the following tests round robin during run-time:

- ADC measure and verify VGND
- ADC measure and verify VREF
- DAC plausibility check of all output values

This task identifies pin open, short to VDD, short to GND and short to adjacent pin.

5.3.8 BIST WDT

The on-chip window watchdog timer with its independent time base monitors the programme function. The watchdog first sets a pre-warning alarm at the first time overrun and triggers a master reset when it overruns the second time.

The BIST WDT supports two modes in order to service the watch dog timer:

- In normal mode, the watchdog is served in time and no overrun occurs.
- In test mode at a fixed test-rate the watchdog is not serviced. As a result it sets the pre-warning alarm at overrun. The occurrence of the pre-warning alarm is checked by the software.

In case of a pre-warning in normal mode, failsafe mode is entered. The master reset function of the watchdog timer acts as last line of defense in case the safety task handler does not execute the safety tasks in the desired sequence or timing.

5.3.9 BIST CPU Load

It was estimated that the execution of BIST tasks would take up to approximately 5% of CPU load during run-time. The following Figure 10 shows that the CPU load is increased as the motor carrier frequency increases in the case of a single motor per system. The average CPU load was measured around 84% when the motor carrier frequency was set to 26 kHz with safety Class B function disabled. When the safety Class B function was enabled, the average CPU load went up to 89% with the same 26 kHz motor carrier frequency.

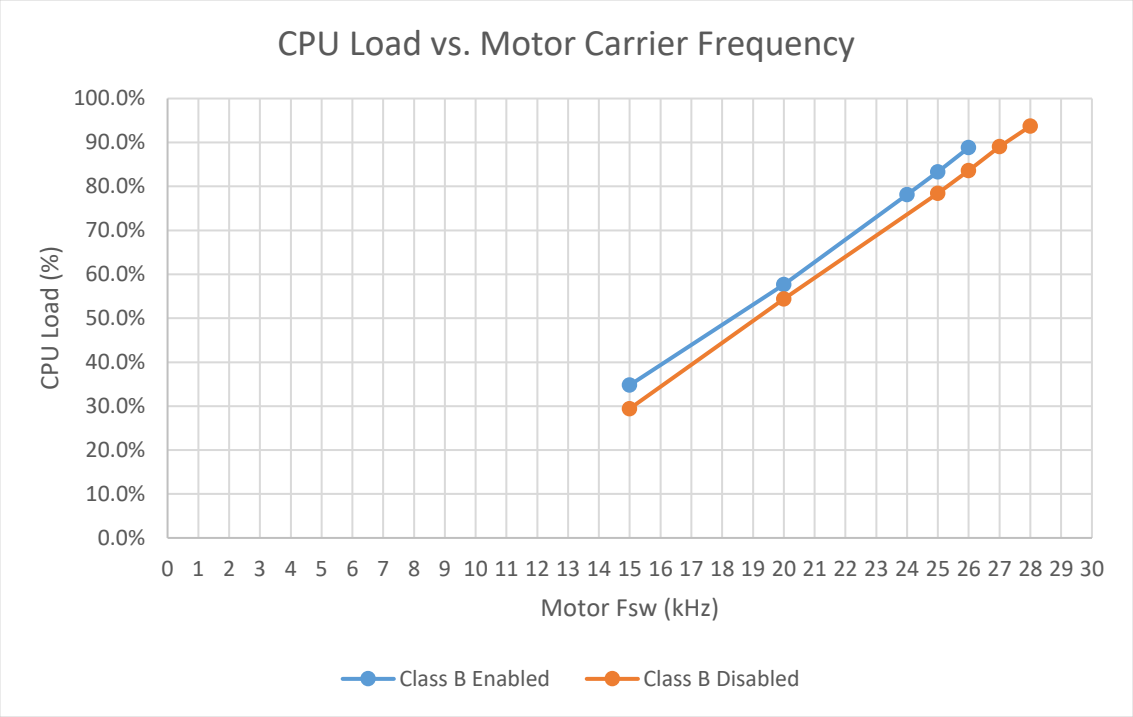

Figure 10 CPU Load vs. Motor Carrier Frequency with / without Safety Class B

5.4 External Communication

The iMOTION™ software provides an external communication via UART user protocol with CRC16-CCITT.

5.4.1 CRC16

The format of the data frame is an 8 byte frame with 16bit CRC16 checksum. The CRC16-CCITT polynomial 0x1021 (x16 + x12 + x5 + 1) is used to check the integrity of the message. This allows a hamming distance of 4 as the number of transmitted bits is below 32751.

In case of a faulty CRC, the controller does not execute the command and does not respond.

5.4.2 Link Break Protection

Link break protection is to stop the motor if there is no UART communication for certain period of time. The controller expects a valid command within a configured period of time. When the period is expired, the control enters fault state.

6 Parameter Safety

6.1 Parameter Handler

The iMOTION™ software provides a safe parameter handling mechanism which comprises the following components:

- Value range control mechanism
- Access control mechanism

6.1.1 Value Range Control Mechanism

All parameters are 16bit values. They are checked against their minimum and maximum limits prior to their handling within the iMOTION™ software. Only parameter values within the allowed range are used to adjust the parameter of the control.

6.1.2 Access Control Mechanism

The parameter handling provides two independent access control mechanism.

One mechanism controls the access with the following flags:

- Static: Initial configuration only, not adjustable during runtime; value is stored within the parameter set.
- Dynamic: Initial configuration, adjustable during runtime; value is stored within the parameter set.

The other mechanism locks safety relevant parameters and prohibits adjustments during runtime. As a result, the software always remains in a safe operation range. With the parameter lock enabled the host controller is not required to be included into the safety mechanism.

6.2 Parameter Loader

During system startup, the parameter loader reads the parameter set of each application. Each parameter set contains a checksum in order to identify single bit faults of the values. The parameter loader checks the stored checksum value against the recalculated value. Only valid parameter sets are allowed to be used for the application.

In case of one invalid parameter set, the startup procedure reports a fault to the applications and they remain in IDLE state until valid parameters are installed. Then the applications transition to STOP state.

7 Functional Safety

7.1 Introduction

7.1.1 Combined actions

The control can be configured to provide 2 actions (motor control and PFC). It was constructed that each of the controls remains operative after failure of any portion unique to the other action. For iMOTION™ MCE software, if it detects a motor related fault, then it will stop the PFC operation automatically. In contrast, if the PFC is detected to have a fault, then the motor will continue to run.

7.2 PFC Protection

7.2.1 AC Input Frequency Protection

The AC input frequency max and min limits are configured by MCEWizard [4] automatically based on the selected nominal AC input frequency. If the AC input frequency nominal value is selected as 50Hz, then the valid range of actual AC input frequency is from 45 to 55Hz. If the AC input frequency nominal value is selected as 60Hz, then the valid range of actual AC input frequency is from 55 to 65Hz.

AC input frequency min limit is checked at the beginning of a new line cycle. If the measured positive or negative half line cycle is lower than the min limit, then the 3rd bit in 'PFC_FaultFlags' PFC variable is set [2]. This fault cannot be masked. As a result, the PFC state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the PFC to stop running.

AC input frequency max limit is checked in the process of finding zero-crossing point every PFC PWM cycle. If a zero-crossing point is not found within the max valid half cycle time, then the 3rd bit in 'PFC_FaultFlags' PFC variable is set [2]. This fault cannot be masked. As a result, the PFC state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the PFC to stop running.

7.2.2 AC Input Over / Under-Voltage Protection

AC input voltage is being sampled every PFC PWM cycle, based on which the Root-Mean-Square (RMS) value of the AC input voltage is calculated. The AC over-voltage fault is checked by comparing the calculated AC input voltage RMS value against the value of 'PFC_VacOvLevel' PFC parameter. If the AC input voltage RMS value is higher than 'PFC_VacOvLevel', then the 5th bit in 'PFC_FaultFlags' PFC variable is set [2]. If the 5th bit in 'PFC_FaultEnable' PFC parameter is set, then this fault will be reflected in 'PFC_SwFaults' PFC variable [2], and the PFC state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the PFC to stop running. If this bit is not set, then the corresponding bit in 'PFC_SwFaults' PFC variable will be masked by 'PFC_FaultEnable' parameter, so that this fault will not be reflected in 'PFC_SwFaults' variable, and the PFC state machine will not shift to FAULT state and the PFC will keep running.

7.2.3 PFC Over-Current Protection

The MCE provides PFC over-current protection function by comparing the PFC inductor current against a pre-configured level and disables the PWM output when the inductor current exceeds the tripping level.

As shown in the following Figure 11, the over-current tripping mechanism makes use of an internal comparator of MCE. The tripping level can be programmed externally using a voltage divider driven by a reference voltage whose output is connected to PFCREF pin (non-inverting input of the comparator). The inductor current is sampled by Rs and going through offset adjustment and is connected to PFCITRIP pin (inverting input of the comparator). The actual iTripLevel can be calculated using the following equation.

iTripLevel (V) = iTripCurrentLevel(A) * CurrentInputScale (V/A) + AmplifierOffset (V), where CurretInputScale = -Rshunt * External Amplifier Gain;

An internal configurable digital filter is available to avoid any high frequency noise. The customer can tweak the over-current fault reaction time by tweaking the value of 'PFC_GateKillTime' PFC parameter. The input signal needs to remain stable for the specified digital filter period to trigger the over-current fault. This fault cannot be disabled.

Controls shall be capable of carrying the currents likely to flow in abnormal conditions within the specified fault reaction time.

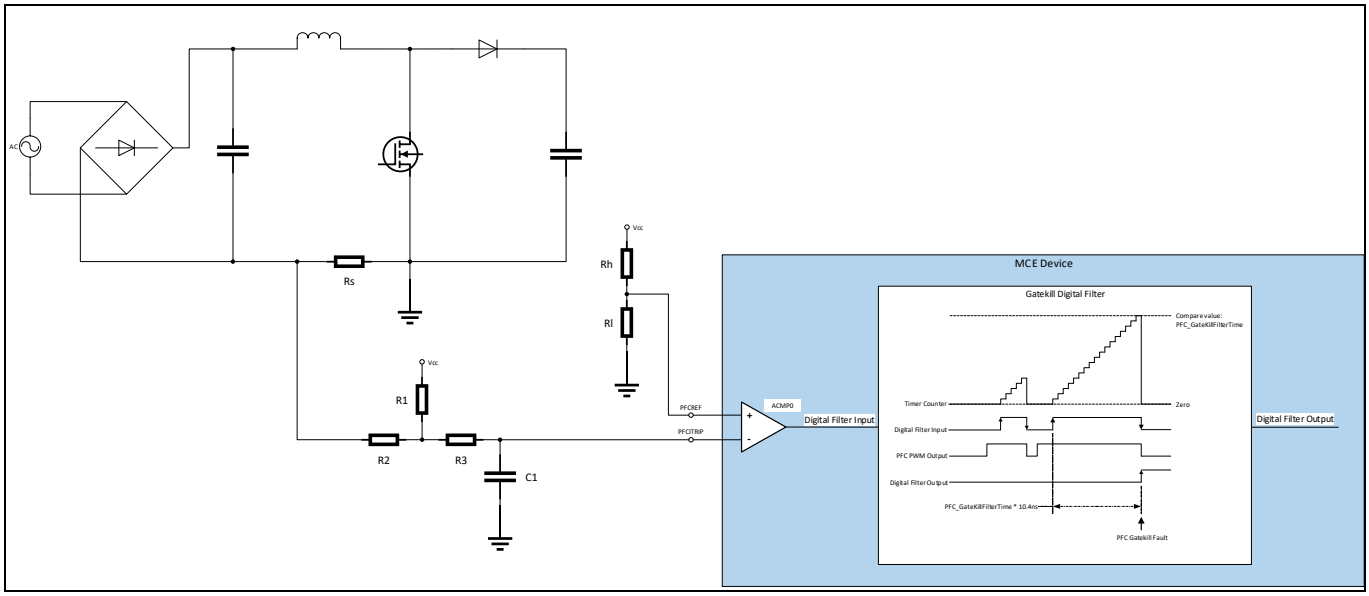

Figure 11 PFC Over-Current Protection Block Diagram

7.3 Motor Control Protection

7.3.1 Motor Phase Loss Protection

If one of the motor phases is disconnected, or the motor windings are shorted together, the parking currents will not have the correct value. During parking state of drive startup, 3 motor phase current values (IU, IV, and IW) are compared against the value of 'PhaseLossLevel' motor parameter every 'SysTaskTime' interval (1ms typically) to determine whether a phase loss (connection between inverter and motor) is presented. If one or more than one of the phase current values is lower than 'PhaseLossLevel', then the phase loss counter is incremented. Otherwise, the phase loss counter is reset. If the counter is greater or equal to 2, then a phase loss fault is confirmed, and the 8th bit of in 'FaultFlags' motor variable is set [2]. If the 8th bit in 'FaultEnable' motor parameter is set, then this fault will be reflected in 'SwFaults' motor variable [2], and the motor state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the motor to stop running. If this bit is not set, then the corresponding bit in 'SwFaults' variable will be masked by 'FaultEnable' parameter, so that this fault will not be reflected in 'SwFaults' motor variable, and the motor state machine will not shift to FAULT state and the motor will keep running.

7.3.2 Motor Over-Current Protection

Motor over-current condition is detected by 2 sources of inputs:

- Direct Gate Kill (GK) pin: Motor over-current fault is set if input is LOW
- Internal comparators

It is possible to select either both or any one of the 2 sources for motor over-current detection logic. Over-current detection source can be selected by configuring the system parameter ‘GKConf’.

If internal comparators are used, the current tripping level can be configured using MCEWizard [4] by setting the value of ‘CompRef’ motor parameter. In the case of leg shunt current measurement configuration as shown in the following Figure 12, three internal comparators (ACMP0, ACMP1, and ACMP2) are used to detect over-current condition.

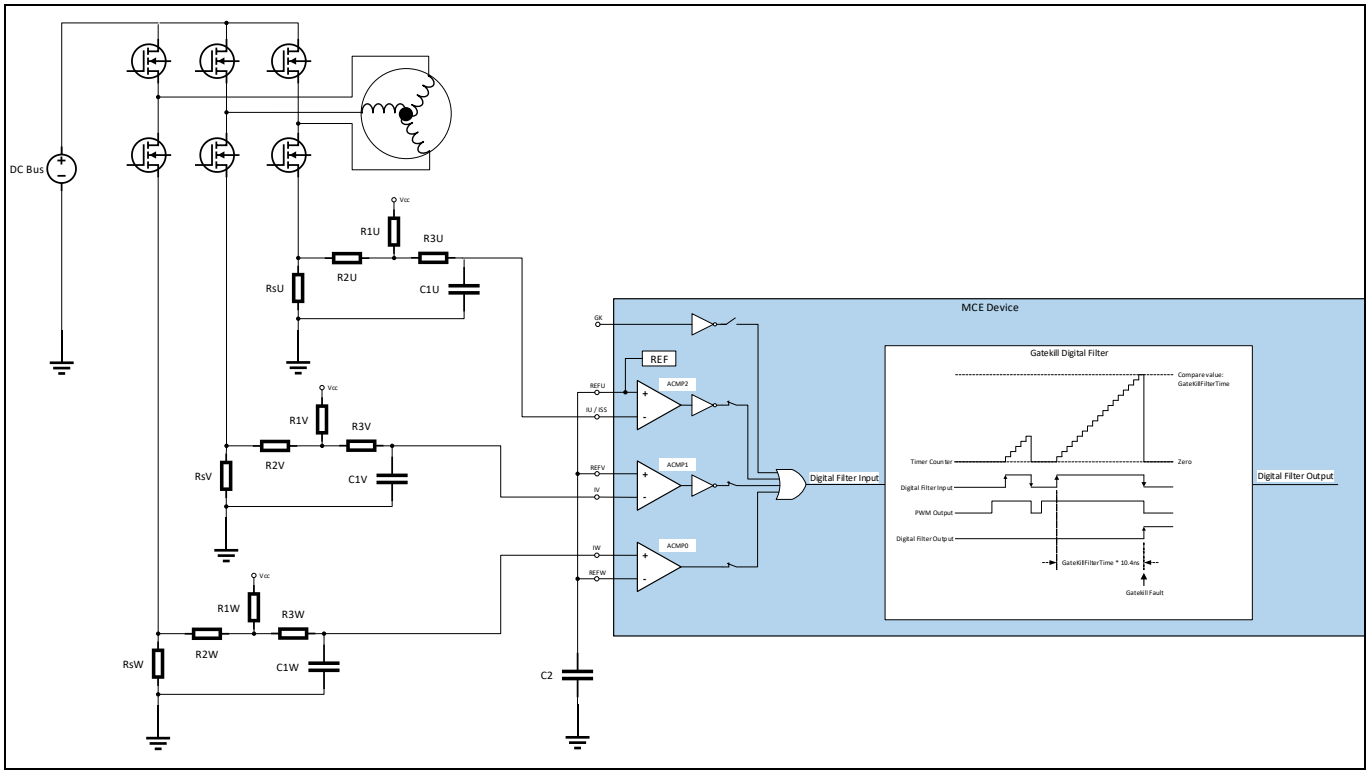

Figure 12 Motor Over-Current Protection Diagram with Leg Shunt Configuration Using Internal Comparators

In the case of single shunt configuration as shown in the following Figure 13, only one internal comparator (ACMP2) is used to detect over-current condition.
An internal configurable digital filter is available to avoid any high frequency noise. The customer can tweak the over-current fault reaction time by tweaking the value of ‘GateKillFilterTime’ motor parameter. The input signal needs to remain stable for the specified digital filter period to trigger the over-current fault. This fault cannot be disabled.
Controls shall be capable of carrying the currents likely to flow in abnormal conditions within the specified fault reaction time.

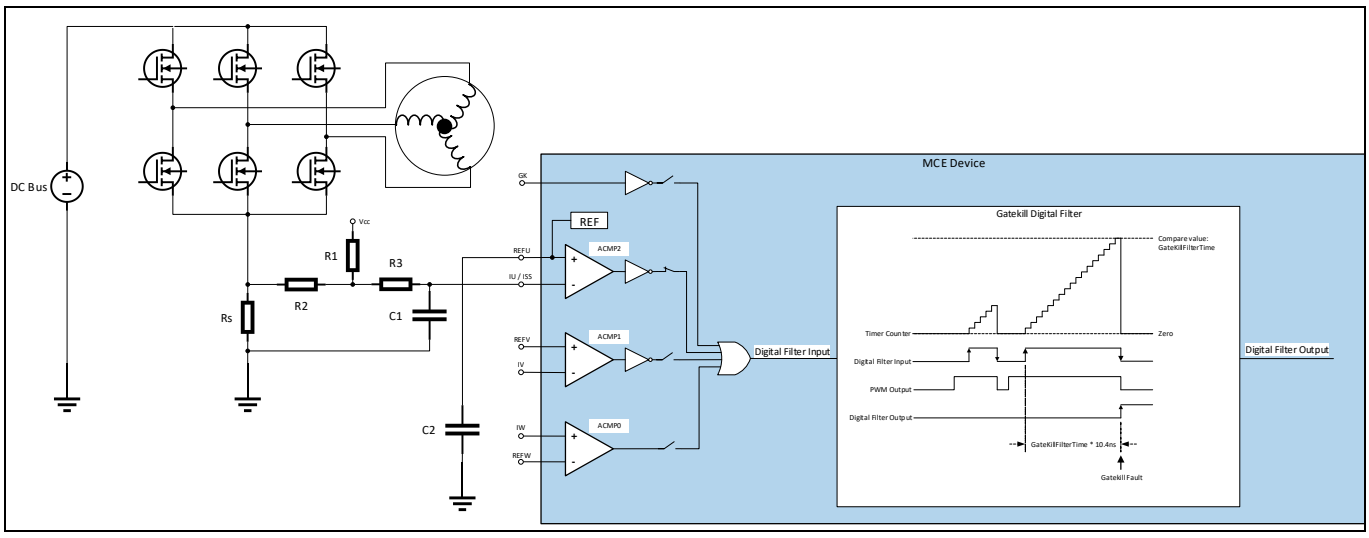

Figure 13 Motor Over-Current Protection Diagram with Single Shunt Configuration Using An Internal Comparator

7.3.3 Rotor Lock Protection

Rotor lock fault is detected if speed PI output (TrqRef) is being saturated for specified period of time (configured by 'RotorLockTime' motor parameter). When the motor speed is above 25% of maximum RPM, rotor lock check is disabled. This is to avoid erroneous fault report at higher speed. Rotor lock fault is checked every 10ms typically. If speed PI output remains saturated for the specified period of time, then the rotor lock fault is confirmed and the 7th bit in 'FaultFlags' motor variable is set [2]. If the 7th bit in 'FaultEnable' motor parameter is set, then this fault will be reflected in 'SwFaults' motor variable [2], and the motor state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the motor to stop running. If this bit is not set, then the corresponding bit in 'SwFaults' motor variable will be masked by 'FaultEnable' motor parameter, so that this fault will not be reflected in 'SwFaults' motor variable, and the motor state machine will not shift to FAULT state and the motor will keep running.

Please note if rotor lock detect time is configured too short, it may trigger the fault during acceleration or momentary high load condition.

7.3.4 Flux PLL Out-Of-Control Protection

When the Flux PLL is locked to correct rotor angle, 'Flx_M', which is a motor variable that represents the fundamental flux amplitude of the PMSM, should be a DC value normalized at 2048 counts. Instead, if the PLL is not locked to correct rotor angle, 'Flx_M' value becomes either unstable or far off from 2048 counts. Flux PLL Out-Of-Control protection is the mechanism designed to detect this fault condition.

The MCE keeps monitoring the value of 'Flx_M' motor variable. Within certain period of time (configured by 'PLL_OutSyncTime' parameter), if its value is below 512 or above 8192 in 8 continuous time slots (each time slot time equals to 'PLL_OutSyncTime' / 8), flux PLL is considered 'out-of-control'. The following Figure 14 shows the details of Flux PLL Out-Of-Control fault triggering conditions and timings.
If the Flux PLL Out-Of-Control fault is confirmed, then it will be reported by setting the 4th bit in 'FaultFlags' motor variable, and the motor speed loop gets reset. If the 4th bit in 'FaultEnable' motor parameter is set [2], then this fault will be reflected in 'SwFaults' motor variable [2], and the motor state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the motor to stop running. If this bit is not set, then the corresponding bit in 'SwFaults' motor variable will be masked by 'FaultEnable'

motor parameter, so that this fault will not be reflected in ‘SwFaults’ motor variable, and the motor state machine will not shift to FAULT state and the motor will keep restarting thanks to the speed loop reset.

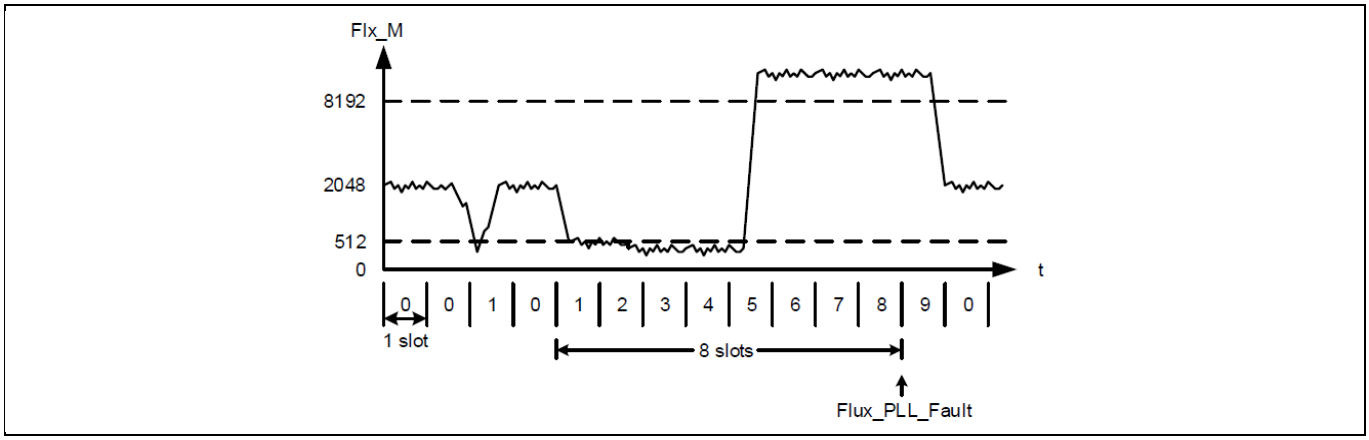

Figure 14 Flux PLL Out-Of-Control Protection Triggering Conditions & Timings

7.3.5 Invalid Hall Protection

‘Invalid Hall’ fault is detected when the Hall sensor signal input pattern is not a valid value (ex: [000] or [111]). The invalid pattern check is only applicable to 3 digital Hall configuration.
Hall pattern is formed as a binary number ([H3, H2, H1]b) by using the 3 digital inputs, H1, H2 and H3 of Hall Event Capture block, and assumes that H3 is bit 2, H2 is bit 1, and H1 is bit 0 . For example, if H3 is logic high, H2 is logic low, and H3 is logic high, then the Hall pattern is recognized as [101]b= 5.
Hall pattern validation starts by comparing the newly sampled Hall pattern with an expected Hall pattern from a pre-determined Hall pattern sequence based on motor rotating direction.
If the newly sampled Hall pattern is [111] or [000], then it is considered as an invalid pattern fault. If two consecutive occurrences of the invalid pattern fault are detected, then ‘Hall Invalid’ fault is confirmed and the 15th bit of variable ‘APP_MOTOR0.FaultFlags’ is set.
If the newly sampled Hall pattern is valid but doesn’t match either the expected Hall pattern from the CW rotating Hall pattern sequence or from the CCW rotating Hall pattern sequence, then it is considered as an unexpected pattern fault. If three consecutive occurrences of the unexpected pattern fault are detected, then ‘Hall Invalid’ fault is confirmed and the 15th bit of variable ‘APP_MOTOR0.FaultFlags’ is set.
If the bit 15 in ‘APP_MOTOR0.FaultEnable’ motor dynamic parameter is set, then this fault will be reflected in ‘APP_MOTOR0.SwFaults’ motor variable, and the motor state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the motor to stop running. If this bit is not set, then the corresponding bit in ‘APP_MOTOR0.SwFaults’ variable will be masked by "APP_MOTOR0.FaultEnable" parameter, so that this fault will not be reflected in ‘APP_MOTOR0.SwFaults’ variable, and the motor state machine will not shift to FAULT state and the motor will keep running.
This fault can be cleared by writing 1 to ‘APP_MOTOR0.FaultClear’ motor variable.

7.3.6 Hall Timeout Protection

‘Hall Timeout’ fault is detected when there is no change in Hall pattern for a configured period of time. The time interval between Hall pattern change is measured and compared against a threshold. This allows the detection of rotor lock condition when Hall sensors are being used.
When the motor control is using Hall sensor interface (‘APP_MOTOR0.AngleSelect’ = 1 or 3), if the time interval between the two sequential Hall transition events is longer than a threshold Tzf, then it is considered as a Hall zero frequency fault. The threshold Tzf is calculated following this equation Tzf = 4096 × TPWM. Once the time interval between the two sequential Hall transition events is shorter than the thresholdTzf, this fault is

automatically cleared.
The equivalent motor speed that would trigger Hall zero frequency fault consistently with 2 or 3 digital Hall sensor configurations can be calculated as follows:

$$\omega_{zf_3Hall}(rpm) = \frac{1}{4096 \times T_{PWM}} \times \frac{1}{6} \times \frac{60}{pole_pair}$$

If this Hall zero frequency fault lasts as long as ‘ThallTimeOut’ time, then a ‘Hall Timeout’ fault is confirmed by ‘Motor0_HallTimeoutProtection’ safety task. The value of ‘ThallTimeOut’ time can be configured using iSD which would automatically generate the value for the parameter ‘APP_MOTOR0.HallTimeoutPeriod" following this equation: APP_MOTOR0.HallTimeoutPeriod = THallTimeOut [s] / 16ms.
If a ‘Hall Timeout’ fault is confirmed, then it will be reported by setting the bit 14 in ‘APP_MOTOR0.FaultFlags’ motor variable. If the bit 14 in ‘APP_MOTOR0.FaultEnable’ motor dynamic parameter is set, then this fault will be reflected in ‘APP_MOTOR0.SwFaults’ motor variable, and the motor state machine will shift to FAULT state where the PWM outputs will be set to their passive levels respectively, causing the motor to stop running. If this bit is not set, then the corresponding bit in ‘APP_MOTOR0.SwFaults’ variable will be masked by ‘APP_MOTOR0.FaultEnable’ parameter, so that this fault will not be reflected in ‘APP_MOTOR0.SwFaults’ variable, and the motor state machine will not shift to FAULT state and the motor will keep running.
This fault can be cleared by writing 1 to ‘APP_MOTOR0.FaultClear’ motor variable.

8 Using Safety Class B Funcion

Using FW V1.03.xx, Class B safety function can be enabled or disabled using MCEWizard [4] on ‘Question 4’ page as shown in Figure 15, or on ‘Advanced Mode’ page / ‘System’ tab as shown in Figure 16.

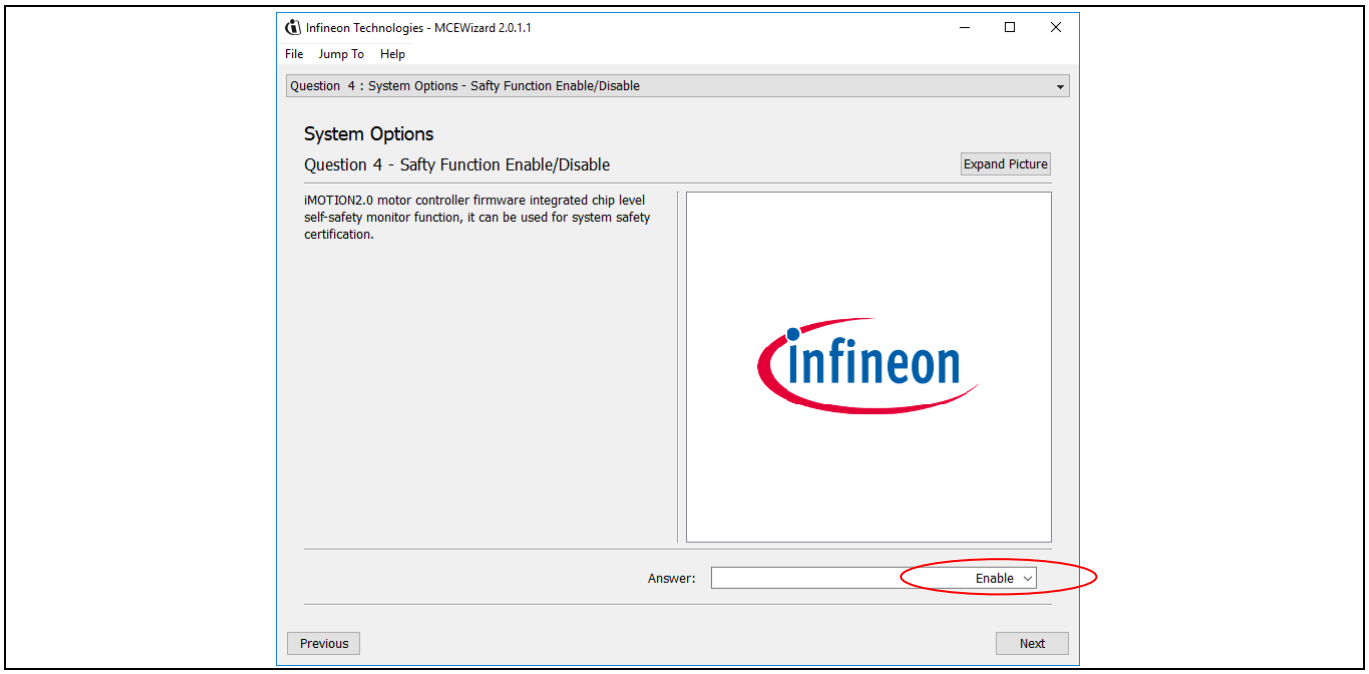

Figure 15 ‘Question 4’ Page of MCEWizard

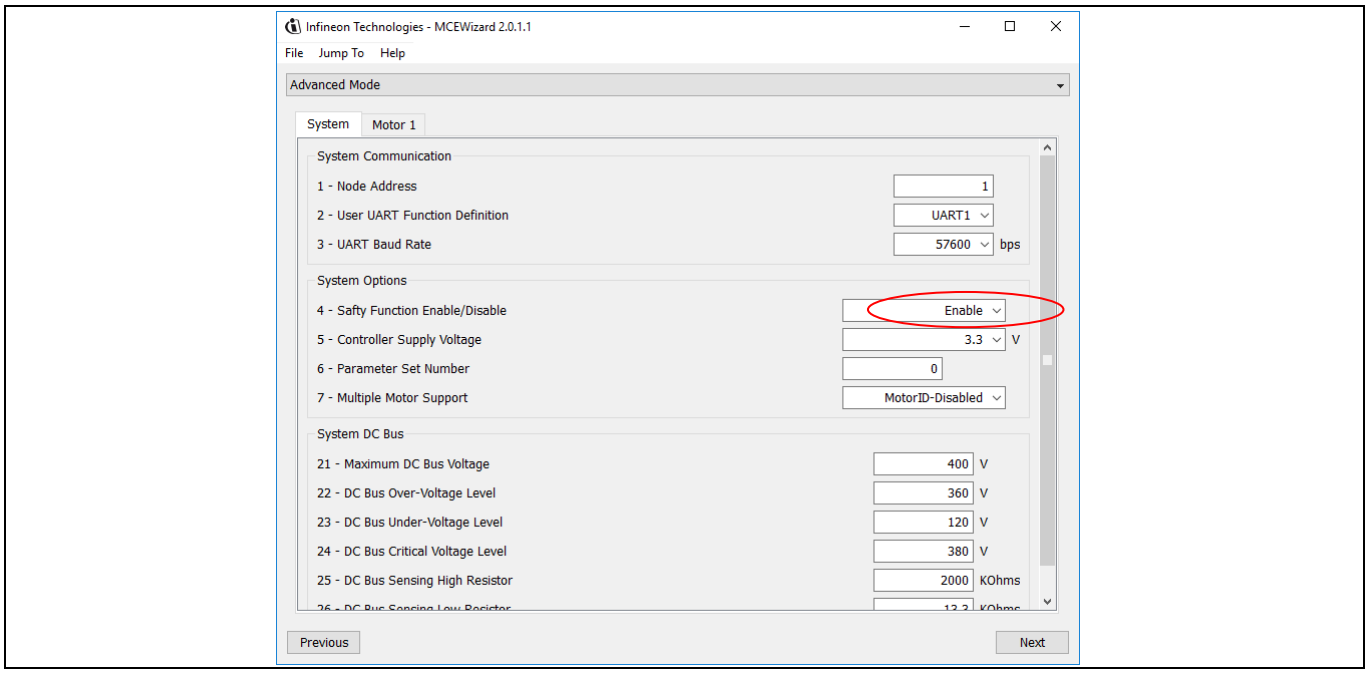

Figure 16 ‘Advanced Mode’ Page / ‘System’ Tab of MCEWizard

Using FW V5.03.xx, Class B safety function can be enabled or disabled using iSD in ‘IC Configuration’ group as shown in Figure 17.

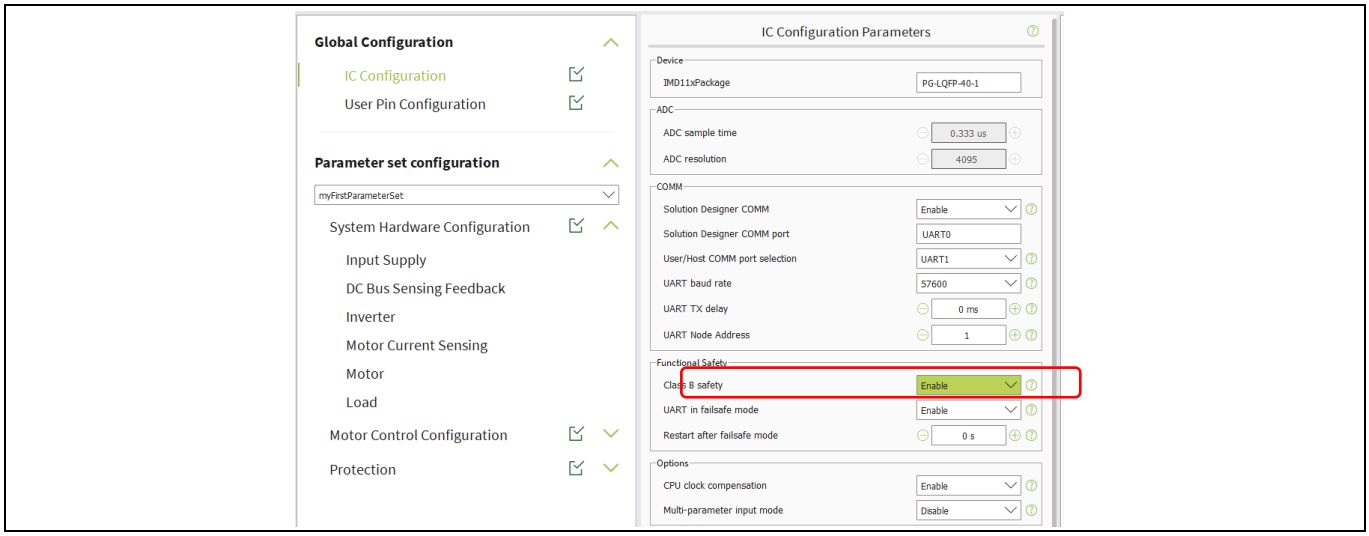

Figure 17 'IC Configuration' Group from Configuration Wizard of iSD

9 Failure Mode and Effects Analysis

During the design process, the following failure mode and effects analyses (FMEAs) have been performed. The result is summarized in this chapter.

9.1 Adjacent Pin Short / Open / Stuck Analysis

The following potential failure modes have been analyzed:

- Pin open
- Pin stuck at VDD
- Pin stuck at VSS
- Pin connected to adjacent pin

9.1.1 Motor and PFC PWM Output Pins

Potential effect of failure:	Malfunction of motor leads to higher DC bus current. Motor sees phase imbalance lacking one phase current which leads to overcurrent; possibly higher over temperature of motor windings due to current imbalance.
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Over current protection

Attention: ***The use of gate drivers with shoot through protection is recommended.***

9.1.2 Pins Related to Protection Mechanisms

9.1.2.1 NTC Input Pin

Potential effect of failure:	No over temperature protection
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity. Monitor and check NTC pin input reading value to see if it is within valid range

Attention: ***Poor soldering may lead to unexpected motor drive fault.***

9.1.2.2 VDC Input Pin

Potential effect of failure:	Malfunction, no over voltage protection
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity. Monitor and check VDC pin input reading value to see if it is within valid range

Attention: ***Poor soldering may lead to unexpected motor drive fault.***

9.1.2.3 IPFCTRIP Input Pin

Potential effect of failure:	No protection of PFC overcurrent, or detected at PFC starting instance at ACV zero crossing to see if any current on PFC
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity.

Attention: ***Poor soldering may lead to unexpected fault.***

9.1.2.4 GK Input Pin

Potential effect of failure:	No gate kill protection
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix/Check a cold solder and continuity. Use of internal comparator for Itrip

Attention: ***Poor soldering may result in a wrong configuration which may lead a motor fault. In order to reduce the risk, the use of internal comparator for overcurrent protection is recommended.***

9.1.2.5 IU, IV, IW Input Pins

Potential effect of failure:	Wrong current measurement of motor phase current resulting no overcurrent
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity. Could be detected by rotor lock protection or flux estimator fault

Attention: ***Poor soldering may lead to unexpected motor drive fault.***

9.1.2.6 ISS input pin

Potential effect of failure:	Wrong current measurement of all phases resulting no overcurrent
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity. Could be detected by rotor lock protection or flux estimator fault

Attention: ***Poor soldering may lead to unexpected motor drive fault.***

9.1.3 Control Interface Pins

9.1.3.1 UART Interface

Potential effect of failure:	If Link Break protection is disabled, AND the user program changes the motor speed, AND/OR fast deceleration AND / OR the overcurrent limit, then it may cause a motor fault

Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Checksum AND / OR CRC16 will prevent from transferring wrong data

Attention: ***Interface faults may lead to unexpected motor drive fault***

9.1.3.2 VSP

Potential effect of failure:	Possible motor at full speed right after power up
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Fix / Check a cold solder and continuity

Attention: ***Poor soldering may result in a wrong configuration which may lead to situations, where the motor is continuously running.***

9.1.3.3 DUTYFREQ

This fault may occur with pin connected to adjacent GPIO only.

Potential effect of failure:	Possible motor at full speed right after power up
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Use other pin instead of adjacent GPIO if configured PWM output, Fix/Check the solder and continuity

Attention: ***Poor soldering may result in a wrong configuration which may lead a motor fault.***

9.1.3.4 DIR

Potential effect of failure:	Motor may run in wrong direction, the system may enter fault condition
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Redundant use of other I/O pin for the same purpose, check continuity of user application hardware soldering

Attention: ***Wrong programming by Script Language may lead to unexpected motor drive fault***

9.1.3.5 LED

Potential effect of failure:	Misleading fault indication, not safety related
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Redundant use of other I/O pin for the same purpose, check continuity of user application hardware soldering

9.1.3.6 PAR0..3 / PARAM

Potential effect of failure:	Wrong parameter set, wrong configuration of protection functions
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Redundant use of other I/O pin for the same purpose, check continuity of user application hardware soldering

Attention: ***Poor soldering may result in a wrong configuration which may lead to a motor fault.***

9.1.4 GPIOs and AIN pins

Potential effect of failure:	Without use of Script language, no effect as unused pin. If defined by Script malfunction of system. If additionally programmed to change the motor speed, AND / OR current limit AND / OR fast deceleration, the system may enter fault condition.
Potential cause of failure:	Cold solder or poor solder or degradation of solder
Prevention method:	Redundant use of other I/O pin for the same purpose, check continuity of user application hardware soldering

Attention: ***Wrong programming by Script Language may lead to unexpected motor drive fault***

9.2 Critical Parameter Wrong Value Setting Analysis

The parameter analysis shows the severity of value altered situation which occurs unintended, but under proper transmission of parameters. It does not consider transmission faults, as they are detected by the CRC16 mechanism. The following severity classification has been used:

Severity	Effect Description	Application Effect
1-2: Very Minor	not safety relevant	1: no effect, safe operation 2: May cause functional fails, but safe operation
3-4: Minor	not safety relevant	3: Likely to cause functional fails, but safe operation 4: Will cause functional fails, but safe operation
5-6: Low	Malfunction may result in higher temperatures	5: Will cause functional fails and low temperature increase 6: Will cause functional fails and medium temperature increase
7: High	Malfunction resulting in overstress	7: Will cause functional fails and high temperature increase
8: Very High	Malfunction of directly related protection mechanism	8: May cause critical temperature increase
9-10: Unknown	Might be safety relevant on system level	9: Functionality is well defined 10: Functionality is undefined

Attention: ***Safety relevance on system level has to be assessed in detail.***

9.2.1 Motor Control Parameter

9.2.1.1 Class B Functions

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(12) FaultEnable	disable protection	not protected	8
(16) RotorLockTime	no protection in time	not protected	8
(18) PLL_OutSyncTime	no protection in time	not protected	8
(74) PhaseLossLevel	wrong phase loss threshold level	not protected	8
(81) FaultRetryPeriod	unwanted restart	unwanted restart	8
(71) AppConfig	wrong application control	unexpected movement	7

These parameters are locked for safe operation of the control, if Class B is enabled.

9.2.1.2 Control Performance and Efficiency

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(43) PllKp	control performance	flux fault protection or overcurrent trip	4
(44) PllKi	control performance	flux fault protection or overcurrent trip	4
(45) PllFreqLim	control performance	overcurrent trip	4
(46) AngMTPA	control performance	overcurrent trip	4
(48) AtanTau	control performance	overcurrent trip	4
(55) KpIreg	control performance	overcurrent trip	4
(56) KpIregD	control performance	overcurrent trip	4
(57) KxIreg	control performance	overcurrent trip	4
(62) AngDel	control performance	overcurrent trip	4
(63) AngLim	control performance	overcurrent trip	4
(73) PrimaryControlLoop	control performance	overcurrent trip	4
(75) TrqCompGain	control performance	overcurrent trip	4
(77) TrqCompLim	control performance	overcurrent trip	4
(30) KpSreg	speed control instability	malfunction of speed control	4
(31) KxSreg	speed control instability	malfunction of speed control	4
(59) FwkKx	speed control instability	malfunction of speed control	4
(160) ZeroVectorReq	unwanted zero vector	active motor break, force to stop	4
(3) AngleSelect	control behavior	control performance	3
(37) SpdRampRate	speed control overshoot	malfunction of speed control	3
(34) RegenSpdThr	low speed control behavior	reduced control performance	2
(38) MinSpd	low speed control behavior	reduced control performance	2
(52) SpdFiltBW	speed control performance	reduced control performance	2

9.2.1.3 Control Mode and Target Values

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(4) CtrlModeSelect	wrong control mode	overcurrent trip	3
(121) TargetSpeed	wrong target speed value	wrong target speed	4
(128) IdRef_Ext	high current	overcurrent trip	4
(129) IqRef_Ext	high current	overcurrent trip	4
(130) Vd_Ext	high current	overcurrent trip	4
(131) Vq_Ext	high current	overcurrent trip	4
(65) Pwm2PhThr	stay at three phase modulation	reduced efficiency	2

9.2.1.4 High Current

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(27) IS_Pulses	high current at startup	overcurrent trip	3
(28) IS_Duty	high current at startup	overcurrent trip	3
(29) IS_IqInit	high current at startup	overcurrent trip	3
(32) MotorLim	higher current limit	overcurrent trip	3
(33) RegenLim	higher current limit	overcurrent trip	3
(37) SpdRampRate	speed control overshoot	malfunction of speed control	3
(58) FwkLevel	higher current, reduced torque	reduced control performance	3
(60) FwkCurRatio	higher current, reduced torque	reduced control performance	3
(76) TrqCompAngOfst	control performance	reduced control performance	2
(78) TrqCompOnSpeed	control performance	reduced control performance	2
(79) TrqCompOffSpeed	control performance	reduced control performance	2
(80) PolePair	control performance	reduced control performance	2
(64) .IdqFiltBW	current control performance	reduced control performance	2

9.2.1.5 DC-Bus Voltage Monitoring

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(13) VdcOvLevel	wrong threshold of VDC	malfunction in over voltage detection	4
(14) VdcUvLevel	wrong threshold of VDC	malfunction in under voltage detection	4
(15) CriticalOvLevel	wrong threshold of VDC	malfunction in over voltage detection	4

9.2.1.6 Over-Temperature Detection

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(67) TShutdown	higher threshold of over-temperature detection	malfunction of over-temperature shut down	4

9.2.1.7 Startup failure

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(21) BtsChargeTime	longer BTS charge or less efficiency during startup	longer startup	2
(22) TCatchSpin	delayed startup	startup failure	2
(23) DirectStartThr	different startup behavior	startup failure	2
(24) ParkTime	delayed startup	startup failure	2
(25) ParkAngle	delayed startup	startup failure	2
(26) OpenloopRamp	different startup behavior	startup failure	2
(35) LowSpeedLim	higher current limit	reduced control performance	2

9.2.1.8 Faulty Current Measurement

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(8) SHDelay	current measurement not correct	overcurrent trip	3
(9) TMinPhaseShift	single shunt current measurement not correct	overcurrent trip	3
(10) TCntMin	single shunt current measurement not correct	overcurrent trip	3
(11) PwmGuardBand	leg shunt current measurement not correct	overcurrent trip	3

9.2.1.9 Control Interface

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(120) Command	unwanted start or stop	functional effects	2
(134) FaultClear	unwanted transition from fault state to stop state	no effect	1

9.2.2 PFC Control Parameter

9.2.2.1 Class B

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(20) VacOvLevel	wrong threshold of VAC protection	no protection	8
(21) VacUvLevel	wrong threshold of VAC protection	no protection	8
(26) FaultEnable	disable protection	not protected	8
(4) TMinOff	cannot sense current	destruction of power stage	7
(6) SHDelay	cannot sense current	destruction of power stage	7

These parameters are locked for safe operation of the control, if Class B is enabled.

9.2.2.2 Control Performance

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(19) VacZCThr	AC wrong voltage swelling	protected, stop PFC	4
(89) TargetVolt	wrong output voltage	protected by over voltage protection	4
(10) KpVreg	unstable current	over current trip	3
(11) KxVreg	unstable current	over current trip	3
(12) KpIreg	unstable current	over current trip	3
(13) KxIreg	unstable current	over current trip	3
(16) TrackingGain	unstable current	over current trip	3
(24) AcDcScale	overcurrent	over current trip	3

9.2.2.3 Functional Deterioration

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(7) IRectLim	clamping of current	functional deterioration	3
(8) IGenLim	current shape	minor functional deterioration	2
(14) TrackingDoff	current shape	minor functional deterioration	2
(15) TrackingCycle	current shape	minor functional deterioration	2
(25) LFactor	current shape	minor functional deterioration	2
(9) VdcRampRate	DC bus fluctuation	no effect	1

9.2.2.4 DC-Bus Voltage Monitoring

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(22) VdcOvLevel	wrong threshold of VDC	malfunction in over voltage detection	4
(23) VdcUvLevel	wrong threshold of VDC	malfunction in under voltage detection	4

9.2.2.5 Control Interface

(Index) Parameter	Potential Failure Mode	Potential Effect of Failure	Severity
(82) Command	unwanted start or stop	functional effects	2
(85) FaultClear	unwanted transition from fault state to stop state	no effect	1

9.3 System Performance Effect and Analysis

9.3.1 Class B POST

9.3.1.1 Power-up Diagnostics Error

Potential effect of failure:	WDT fault, memory fault, clock fault: loop forever in startup sequence without fault indication
Potential cause of failure:	end of life
Detection mechanism:	Class B POST routines called at startup

9.3.2 Class B BIST

9.3.2.1 CPU Overload

Potential effect of failure:	CPU overload due to wrong BIST scan time setting or high PWM carrier frequency update rate requirement resulting in CPU overload error or performance not meeting application Fault with stop or premature overcurrent trip at normal run state
Potential cause of failure:	Performance limit per Application
Prevention method:	Check CPU load during development

Attention: ***CPU overload will trigger failsafe mode.***

9.3.2.2 Flash Memory Error

Potential effect of failure:	ECC and/or parity error enter failsafe mode, motor stop
Potential cause of failure:	NVM failure mode: moving bit (0->1)
Detection mechanism:	BIST FLASH

9.3.2.3 RAM Memory Error

Potential effect of failure:	Parity error enter failsafe mode, motor stop
Potential cause of failure:	soft error due to alpha decay within the package
Detection mechanism:	BIST RAM

9.3.2.4 Watchdog Error

Potential effect of failure:	Watchdog timer overrun enter failsafe mode, motor stop
Potential cause of failure:	main clock fault, heavy CPU overload
Detection mechanism:	BIST WDT

9.3.2.5 Clock Error

Potential effect of failure:	clock too slow or too fast control loop instability, delayed control
Potential cause of failure:	Critical over temperature
Detection mechanism:	BIST CLK

9.3.2.6 ADC Error

Potential effect of failure:	ADC conversion with fault result control loop instability, increased current
Potential cause of failure:	Critical over temperature
Detection mechanism:	BIST IO

9.3.2.7 DAC Error

Potential effect of failure:	ADC loopback test fails ADC/DAC does not converge within specified limit
Potential cause of failure:	External bias condition, C_REF pin short/open
Detection mechanism:	BIST IO

9.3.2.8 Safety Handler

Potential effect of failure:	safety tasks are not properly executed "failsafe mode, motor stop
Potential cause of failure:	memory overlap with application functions
Prevention method:	no use of dynamic allocated memory
Detection mechanism:	duplicated values

10 References

[1] IEC 60730-1 (Edition 5.1 2015-12): Automatic electrical controls

[2] iMOTION™ Motion Control Engine Software Reference Manual (VER 1.35, 2023-08-14)

[3] iMOTION™ Motion Control Engine Functional Reference Manual (VER1.3, 2024-01-10)

[4] MCEDesigner User Guide (REV 2.0.1.1)

[5] MCEWizard 2.0 User Guide (REV 2.0.1.1)

[6] iMOTION™ Solution Designer User Guide (REV1.2, 2024-02-12)

11 Appendix

Table 5 Cross-Reference Between Acceptable Measures to Address Fault / Errors Required for Class B (Table H.1 of IEC / UL60730-1 [1]) and iMOTION™ Class B Modules and Functions

Component	Fault / error	Example of acceptable measures	Measures used by iMOTION™ Software	Relevant iMOTION™ Class B Modules & Functions
1. CPU 1.1 Registers	Stuck at	Functional test, or periodic self-test using either: static memory test or word protection with single bit redundancy	Functional test	POST CPU Test
			Periodic self-test using word protection with single bit redundancy	BIST RAM Test, BIST FLASH test
1.3 Programme counter	Stuck at	Functional test, or periodic self-test, or independent time-slot monitoring of the program sequence, or logical monitoring of the programme sequence	Functional test	POST PC Test, POST WDT Test, POST IRQ Test, POST CLK Test
			Independent time-slot monitoring of the program sequence	POST WDT Test, POST IRQ Test, POST CLK Test, Execution Monitoring & IRQ, BIST CLK Test, BIST WDT Test
2. Interrupt handling and execution	No interrupt or too frequent interrupt	Functional test; or time-slot monitoring	Functional test	POST WDT Test, POST IRQ Test, POST CLK Test,
			Time-slot monitoring	POST WDT Test, POST IRQ Test, POST CLK Test, Execution Monitoring & IRQ, BIST CLK Test, BIST WDT Test
3. Clock	Wrong frequency	Frequency monitoring, or time slot monitoring	Time-slot monitoring	POST WDT Test, POST IRQ Test, POST CLK Test, Execution Monitoring & IRQ, BIST CLK Test, BIST WDT Test
4.1 Invariable memory	All single bit faults	Periodic modified checksum; or multiple checksum, or word protection with single bit redundancy	Word protection with single bit redundancy	BIST FLASH Test

Component	Fault / error	Example of acceptable measures	Measures used by iMOTION™ Software	Relevant iMOTION™ Class B Modules & Functions
4.2 Variable memory	DC fault	Periodic static memory test, or word protection with single bit redundancy	Word protection with single bit redundancy	BIST RAM Test
4.3 Addressing (relevant to variable memory and invariable memory)	Stuck at	Word protection with single bit redundancy including the address	Word protection with single bit redundancy including the address	POST AD Test, POST RAM Test Parity, BIST RAM Test, BIST FLASH Test
5. Internal data path 5.1 Data	Stuck at	Word protection with single bit redundancy	Word protection with single bit redundancy	POST RAM Test Parity, BIST RAM Test, BIST FLASH Test, BIST STACK Test
5.2 Addressing	Wrong address	Word protection with single bit redundancy including address	Word protection with single bit redundancy including address	POST AD Test, POST RAM Test Parity, BIST RAM Test, BIST FLASH Test
6 External communication	Hamming distance 3	Word protection with multi-bit redundancy, or CRC –single word, or transfer redundancy, or protocol test	CRC –single word	CRC16
6.2 Addressing	Wrong address	Word protection with multi-bit redundancy, including the address, or CRC – single word including the address, or transfer frequency or protocol test	CRC – single word including the address	CRC16
6.3 Timing	Wrong point in time	Time-slot monitoring, or scheduled transmission	Time-slot monitoring	POST WDT Test, POST IRQ Test, POST CLK Test, Execution Monitoring & IRQ, BIST CLK Test, BIST WDT Test
			Scheduled transmission	Link Break Protection
	Wrong sequence	Logical monitoring, or time-slot monitoring,	Time-slot monitoring	POST WDT Test, POST IRQ Test, POST CLK Test, Execution

Component	Fault / error	Example of acceptable measures	Measures used by iMOTION™ Software	Relevant iMOTION™ Class B Modules & Functions
		or scheduled transmission		Monitoring & IRQ, BIST CLK Test, BIST WDT Test
			Scheduled transmission	Link Break Protection
7. Input / output periphery 7.1 Digital I/O	Fault conditions specified in Clause H.27	Plausibility check	Plausibility check	BIST IO Test
7.2 Analog I/O 7.2.1 A/D- and D/A-convertor	Fault conditions specified in Clause H.27	Plausibility check	Plausibility check	BIST IO ADC Test, BIST IO DAC Test
7.2.2 Analog multiplexer	Wrong addressing	Plausibility check	Plausibility check	BIST IO Test, BIST IO ADC Test, BIST IO DAC Test
9. Custom chips for example, ASIC, GAL, Gate array	Any output outside the static and dynamic functional specification	Periodic self-test	N / A	N / A

Table 6 iMOTION™ Class B Module List

1. POST – One module with the following sub-functions a. WDT Test b. IRQ Test c. CLK Test d. CPU Test e. PC Test f. RAM Test Parity g. AD Test h. RAM Test i. FLASH Test
2. BIST – Each following modules separately listed in certification a. Safety Task Handler b. Execution Monitoring & IRQ c. BIST CLK d. BIST FLASH e. BIST RAM f. BIST STACK g. BIST IO h. BIST IO ADC i. BIST IO DAC j. BIST WDT
3. Parameter Handler & Load Fault
4. UART with CRC16 and Link Break
5. Functional Safety – Each following modules separately listed in certification a. Flux PLL Out-of-Control Protection b. Rotor Lock Protection c. Over-Current Protection d. Phase Loss Protection e. Invalid Hall Protection f. Hall Timeout Protection g. PFC Over-Current Protection h. AC Input Over / Under-Voltage Protection i. AC Input Frequency Protection

Revision history

Document version	Date of release	Description of changes
V1.0	8/28/2018	Initial version
V1.1	12/18/2018	Added Chapter 7, modified 4.3.9, 4.3.1, 4.3.2
V1.2	8/25/2020	Added Section 2 ‘Class B coverage’
V1.3	4/5/2024	Section 2, Section 4.2, Section 4.3, Section 7.3, and Section 8 modified

Edition 4/5/2024

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN2018-22

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.