Safety Manual for C2000™ MCUs in IEC60730 Safety Applications

User's Guide

Literature Number: SPRUHI3A
April 2013–Revised August 2013

Contents

List of Figures

List of Tables

User's Guide
SPRUHI3A–April 2013–Revised August 2013

Safety Manual for C2000™ MCUs in IEC60730 Safety Applications

1 Introduction

You, as a system and equipment manufacturer or designer, are responsible to ensure that your systems (and any Texas Instruments hardware or software components incorporated in your 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. You understand and agree that your use of TI components in safety critical applications is entirely at your risk, and that you (as buyer) agree 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 C2000 32-bit real time C2000™ microcontrollers product family. This document introduces TI's C2000 family of controllers and its subsystem-based architecture that matches most real-time control systems. Its inherent safety features could enable existing and emerging systems that require IEC60730 and IEC60335 or UL1998 safety compliance.

1.1 About This Document

This safety manual provides information needed by system developers to assist in the creation of a IEC60730 class of safety system using a C2000 microcontroller (MCU). This document contains:

- An overview of C2000 MCU - Piccolo™ and Delfino™ MCU product **architecture**
- An overview of the development process utilized to reduce **systematic** failures
- An overview of the safety architecture for management of **random** failures
- The details of architecture partitions, implemented **safety mechanisms**, and recommended usage
- Software design and development process

It is expected that the user of this document should have a general familiarity with the C2000 product family. More information can be found at http://www.ti.com/C2000. This document is intended to be used in conjunction with the device-specific data sheets, technical reference manuals, 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 referenced in any of the IEC safety standards.

IEC60730, IEC60335 and UL1998 specification addresses safety in appliances and equipment's that uses microelectronic hardware and its related software. All these standards reference similar safety functions addressing appliances or industrial equipment. Throughout this document IEC60730 will be the common name used for this safety standard, unless called separately. This document will cover Class B and Class1 level of safety mentioned in these specifications. For this release of this document, all references are applicable IEC60730-1: 2010 Class B and UL1998:2008 Class 1 standards.

IEC60730 Safety manual is part of the C2000 MCU device-specific collateral material and help implement functional safety features enabling the hardware and software libraries. The IEC60730 Library release includes the following collaterals:

- IEC60730 software user's manual
- Software libraries
- Hardware kit for software evaluation, as applicable

Each of the C2000 MCU supporting safety applications may support limited FMEA analysis summary. These are available as NDA documents only. Please contact your TI field and support for product-specific FMEA.

1.2 C2000 and its Application Sectors

C2000 MCUs brings together highly efficient control CPU, control peripherals and rich analog peripherals to build a deterministic control applications. All C2000 MCUs use real-time 32-bit C28x CPU with or without floating-point unit (FPU). Higher end C2000 MCUs support highly differentiated math capabilities using Viterbi Complex Unit (VCU) modules. Piccolo (TMSS320F2806x/5x/3x/2x) and Delfino (TMS320F28023x/33x) MCU feature sets have been proven to deliver high performance in many of the following applications spaces and still grow with their built-in safety features to meet several safety compliance standards:

- Low and high appliances, white goods, and industrial drives
- Power conversion (invertors)
- Energy conversion (solar) controllers
- Power line communication (PLC) controllers

Piccolo and Delfino families of real-time microcontrollers are part of the C2000 MCUs. These devices are available in different flavors of peripheral selection. Product-specific nomenclature and its derivatives are explained in the Piccolo and Delfino data sheets. Special derivatives, if any, are referenced explicitly with its part numbers.

2 C2000 Architecture and Product Overview

C2000 MCUs are built around real-time 32-bit C28x CPUs with system-on-chip (SoC) microcontroller (MCU) features with independent communication and real-time control and analog subsystems. Piccolo (TMSS320F2806x/5x/3x/2x) and Delfino (TMS320F28023x/33x) are part of this family that can enable IEC60730 Class B safety in industrial control, appliances, and other applications. The C2000 MCUs architecture provides users many control topologies utilizing its diverse peripherals within one physical IC package.

Figure 1 provides a very high-level overview of the C2000 MCUs featured in Piccolo and Delfino family of MCU devices.

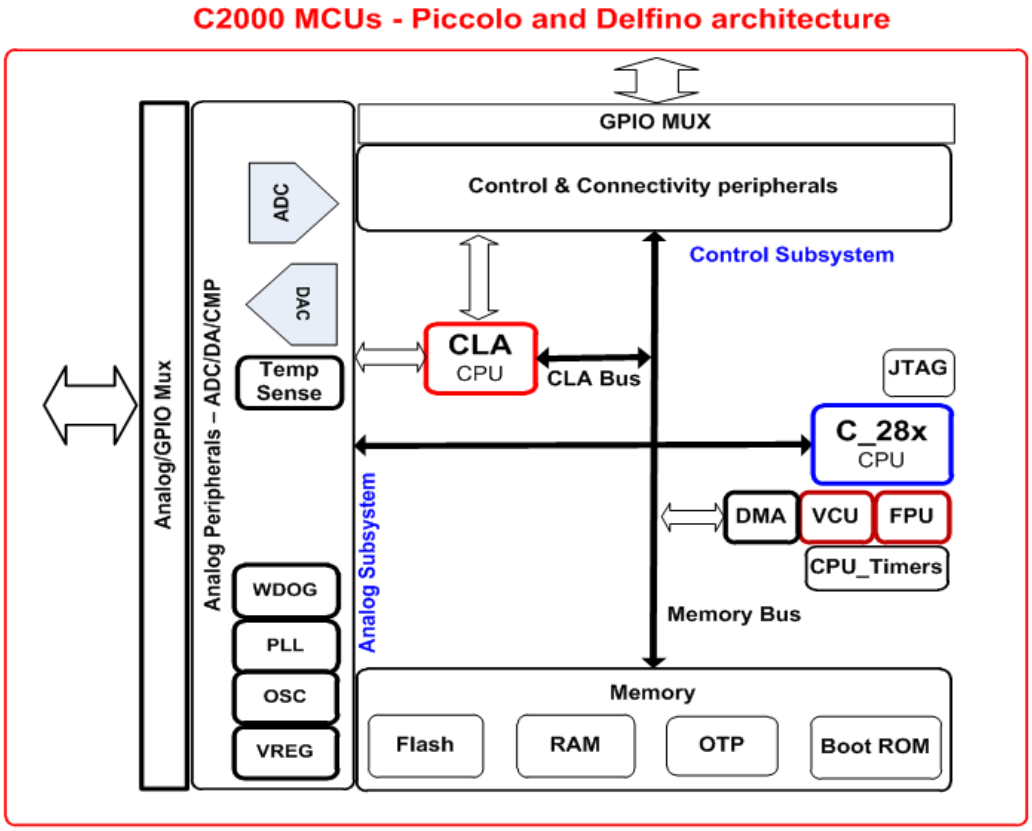

Figure 1. C2000 MCU for IEC60730 Applications - An Overview of System Architecture

2.1 C2000 MCU Architecture Classifications

The following configurations of CPU and subsystems are available across many of Piccolo and Delfino MCU devices. In this document subsystem refers to a collection of Control or Analog peripherals, critical for real time control applications. The above architecture breaks down to four types based on CPU processing power and subsystems.

- **Type A: C28x CPU + FPU + VCU + CLA** with Control and Analog Subsystem – Piccolo6x family
- **Type B: C28x CPU + CLA** with Control and Analog Subsystem – Piccolo3x/5x family
- **Type C: C28x CPU + FPU** with Control and Analog Subsystem – Delfino F2833x family
- **Type D: C28x CPU** with Control and Analog Subsystem – Piccolo2x and Delfino F2823x families

- **Type A: C28x CPU with FPU, VCU, CLA and CONTROL Subsystem:**

  32-bit microcontroller CPU with floating-point capabilities and control law accelerator (CLA) and dedicated I/O optimized for execution of real-time control algorithms, available on the Piccolo**6x** (F2806x) family of devices. This system contains multiple control peripherals (pulse width modulator (PWM), capture, encoders, serial interfaces). The CLA offers floating-point CPU with event optimized peripheral and I/O optimized for Analog subsystem. The CLA can also include safety functions in its capacity as a supervisory microcontroller. The Piccolo6x (2806x) family has an additional math unit VCU (Viterbi and Complex math unit) specifically to do advanced signal processing capabilities for control and power line communication applications

- **Type B: C28x CPU with FPU, CLA and CONTROL Subsystem:**

  32-bit microcontroller with fixed point and IQ math CPU capabilities, with CLA and I/O optimized for execution of real-time control algorithms, available on the Piccolo**3x** and Piccolo**5x** (F2803x/F2805x) family of devices. This system contains multiple control peripherals (PWM, capture, encoders, serial interfaces) for advanced control and sophisticated signal processing applications. The CLA offers a floating-point CPU with event optimized peripheral and I/O optimized for Analog subsystem. The CLA can also include safety functions in its capacity as a supervisory microcontroller.

- **Type C: C28x CPU with FPU and CONTROL Subsystem:**

  32-bit microcontroller CPU with floating-point capabilities and dedicated I/O optimized for execution of real-time control algorithms, available on the Delfino (F2833x) family of devices. This system contains multiple control peripherals (PWM, capture, encoders, serial interfaces) for advanced control and sophisticated signal processing applications.

- **Type D: C28x CPU and CONTROL Subsystem:**

  32-bit microcontroller with fixed point and IQ math CPU capabilities, with dedicated I/O optimized for execution of real-time control algorithms, available on the Piccolo**2x** (F2802x) and Delfino (F2823x) family of devices. This system contains multiple control peripherals (PWM, capture, encoders, serial interfaces) for advanced control and sophisticated signal processing applications.

- **ANALOG Subsystem:**

  The Piccolo MCUs contain a rich set of integrated analog components, such as high performance 12-bit analog-to-digital converter (ADC), analog comparators, zero-pin oscillators and voltage regulators.

  The Delfino MCUs have reduced analog integration, but with high performance 12-bit 12.5 MSPS ADCs and clocking peripherals. Some of its hardware limitation has to be complemented with on-board feature depending on the end application.

The Delfino and Piccolo devices are built in the 180 nm process technology. This is a highly mature silicon technology process with high volume production across TI MCUs for more than a decade. This offers high level of integration, performance and reliability.

The C2000 MCU IEC60730 Safety Manual is to be complemented with its device level documentation, as listed in Table 1, to provide the necessary details on functional implementation and application use:

Table 1. C2000 MCU Documentation

C2000 MCU Product Family Technical Documentation	Abstract
TMS320x28xx, 28xxx DSP Peripheral Reference Guide (SPRU566)	Summary and cross reference guide that describes the peripheral reference guides of the 28x digital signal processors (DSPs and MCUs).
TMS320F2806x Piccolo Microcontrollers Data Manual (SPRS698)	Technical data manual that describes the package, pin outs, and brief functional overview of all modules and electrical timings.
TMS320F2805x Piccolo Microcontrollers Data Manual (SPRS797)	
TMS320F2803x Piccolo Microcontrollers Data Manual (SPRS584)	
TMS320F2802x, TMS320F2802xx Piccolo Microcontrollers Data Manual (SPRS523)	
TMS320F2802x0 Piccolo Microcontrollers Data Manual (SPRS810)	
TMS320F2802x Piccolo MCU Silicon Errata (SPRZ292)	Describes silicon exceptions for system workaround and production readiness of silicon.
TMS320F2803x Piccolo MCU Silicon Errata (SPRZ295)	
TMS320F2806x Piccolo MCU Silicon Errata (SPRZ342)	

Table 1. C2000 MCU Documentation (continued)

C2000 MCU Product Family Technical Documentation	Abstract
These technical reference manuals (TRM) detail the integration, environment, functional description, and the programming models for each peripheral and subsystem. These are single TRMs or multiple peripheral guides as listed below:	
Piccolo Technical Reference Manual (TRM)	TMS320F2802x- See the list of documents under the Device Support section in *TMS320F28027/28026/28023/28022/28021/28020/280200 Piccolo Microcontrollers Data Manual* (SPRS523) or *TMS320x28xx, 28xxx DSP Peripherals Reference Guide* (SPRU566) cross reference guide
	TMS320F2803x - See the list of documents under the Device Support section in *TMS320F28030/28031/28032/28033/28034/28035 Piccolo Microcontrollers Data Manual* (SPRS584) or *TMS320x28xx, 28xxx DSP Peripherals Reference Guide* (SPRU566) cross reference guide
	TMS320x2806x Piccolo Technical Reference Manual (SPRUH18) - single TRM
TMS320F2833x Digital Signal Controllers (DSCs) Microcontrollers Data Manual (SPRS439) - Delfino	Technical data manual that describes the package, pin outs, and brief functional overview of all modules and electrical timings.
TMS320F2823x DSC Silicon Errata (SPRZ272) - Delfino	Describes silicon exceptions for system workaround and production readiness of the silicon.
Delfino Technical Reference Manual (TRM)	This technical reference manual (TRM) details the integration, the environment, the functional description and the programming models for each peripheral and subsystem. See the list of documents under the Device Support section in the *PMSM3-1: Sensored Field Oriented Control of 3-Phase PM Synchronous Motor User's Guide* (SPRU439) or *TMS320x28xx, 28xxx DSP Peripherals Reference Guide* (SPRU566) cross reference guide.
TMS320C28x CPU and Instruction Set Reference Guide (SPRU430)	This document describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x 32-bit fixed-point CPU. It also describes emulation features available on these devices.
TMS320C28x Floating Point Unit and Instruction Set Reference Guide (SPRUEO2)	This document describes the CPU architecture, pipeline, instruction set, and interrupts of the C28x floating-point DSP.

2.2 Targeted Applications and Product Safety Constraints

The C2000 MCU family is targeted primarily for appliance and industrial application spaces and can address general-purpose safety applications.

As these devices are mass market product rather than custom product, it cannot be said that a specific implementation configuration can be assumed. It is expected that the same modules used for the general function in the appliances space will be used for safety critical functions in industrial, power conversion and energy conversion markets. Energy conversion, like the solar inverter, is an emerging example that is increasingly complex and looking for compliance to functional safety standards within the product and equipment level.

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 time intervals for typical motor drives can be within few 10s of milliseconds. The C2000-based system can address fault tolerant times similar to these intervals or better. C2000 MCU's overall response can reach < 10 ms fault tolerant time interval.

The C2000 MCU’s device is designed to meet most of the safety features mentioned in the IEC60730 standards or equivalent. While these devices have several hardware safety features, the application level software is the critical part that adds value to the hardware features and enforces safety discipline and to meet functional safety compliance. A properly designed product using the C2000 MCUs should be capable of meeting safety requirements, if the designers use the device within its stated specifications and implement the required diagnostic hardware and software as stated in this document. System integrator should ensure component and system safety concepts are assessed and adhered during its development and product life cycle.

3 C2000 MCU Development Process for Management of Systematic Faults

For a safety critical development, it is necessary to manage both systematic and random faults. Texas Instruments has created a unique development process for safety critical semiconductors that greatly reduce probability of systematic failure. This process builds on a standard quality managed development as the foundation for safety critical development. The TI MCU teams use this process across all its family of MCUs that cater to industrial and automotive applications.

3.1 TI Standard Automotive MCU and C2000 MCU Development Process

Texas Instruments has been developing microcontrollers for industrial and automotive markets that require safety and non-safety applications for over twenty years. Industrial and automotive markets have strong requirements on quality management and high product reliability. Though not explicitly developed for compliance to a functional safety standard, the TI standard MCU development process already featured many elements necessary to manage systematic faults. This development process can be considered to be Quality Managed (QM), but does not achieve an IEC 61058 Safety Integrity Level (SIL) or ISO 26262 Automotive Safety Integrity Level (ASIL). The TI standard MCU automotive development process is certified compliant to ISO TS 16949 as assessed by Det Norske Veritas Certification, Inc. (Katy, Texas) under certificate CERT-07319CC10-2004-AQ-HOU-IATF (IATF certificate No 0113679). The development is also certified compliant to ISO 9001:2008 as assessed by DNV Certification B.V. (Netherlands) under Certificate CERT-06185-2003-AQ-HOU-RvA Rev. 2. These standards explicitly apply to TI MCUs (C2000 MCUs, Hercules, TMS570S and RM48x MCUs) that are developed in the same silicon technology nodes.

The standard development process breaks development into phases:

- Business opportunity pre-screen
- Program planning
- Create
- Validate, sample and characterize
- Qualify
- Ramp to production and sustaining production

Texas Instruments Incorporated (TI) standard silicon development process is illustrated in Figure 2.

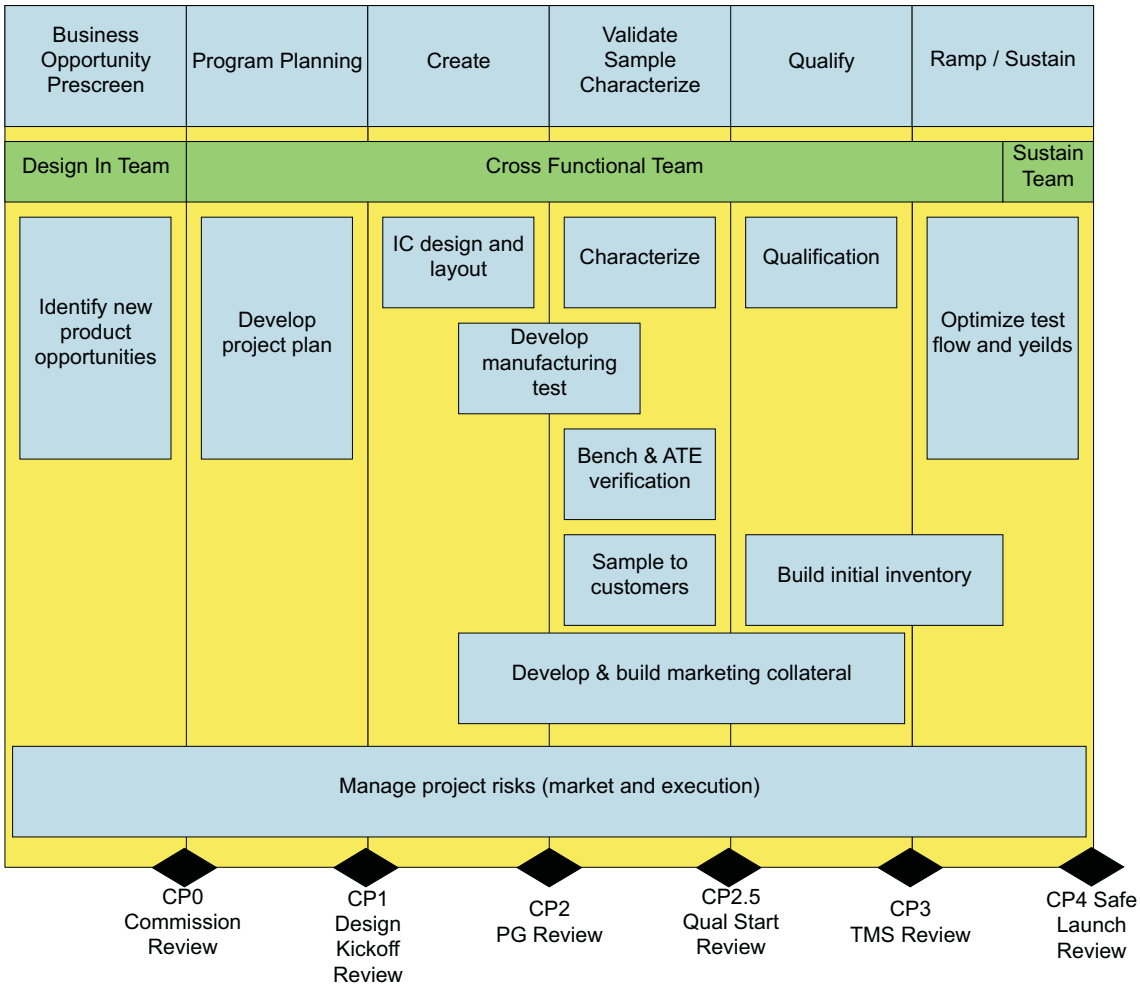

Figure 2. TI Standard MCU Automotive QM Development Process

3.2 C2000 MCU Development Process

The Piccolo and Delfino MCU silicon development process is as per TI MCU development methods and conforms to ISOTS16949 flow. The goal of the process development is to take the best aspects of each flow and collaborate to result in best in class capabilities to reduce systematic faults. The process flow is also targeted for compliance to IEC 61508 as well and is under a process of continuous improvement to incorporate new features. Piccolo and Delfino series of C2000 MCUs are developed to offer devices with several temperature grades to address consumer, industrial and automotive applications.

Key elements of the TI MCU process 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 known to TI
- Fault estimation based on multiple industry standards as well as TI manufacturing data
- C2000 MCU development process adopts QRASAP00160 new product development processes.

4 Product Architecture for Management of Random Faults

For a safety critical development, it is necessary to manage both systematic and random faults. The C2000 MCU product architecture includes safety mechanisms that can help detect and respond to random faults when used correctly. This section describes the architectural safety concept and software layers that need to be adopted in C2000 MCU devices.

4.1 Philosophy and Architecture Partition for Safety Analysis

Piccolo and Delfino MCUs address functional safety with limited hardware assist and layers of supervisory software libraries to detect system and device failure and prompt for corrective actions. This enables simplified functional system partitioning during run time and distribute system level supervisors across the memory, peripherals and analog modules.

Piccolo MCU family offers two distinct variant of safety implementation based on its hardware capabilities. Most of the low end Piccolo (Piccolo2x) derivatives use single C28x CPU configurations, hence, the safety features are primarily software functions along with main application firmware. Piccolo3x/6x and devices has two options of enhancing the safety supervision using both C28x CPU and CLA. This configuration can offer CPU level redundancies on critical functions to implement single and dual channel systems. This is strong differentiator as it offers benefits of dissimilar instruction set and diversity during firmware execution.

4.2 Management of Family Variants

The Piccolo and Delfino MCUs architecture supports multiple product variants. Table 2 shows an overview of CPUs memory and peripherals that each of the Piccolo and Delfino MCU architectures support. Among these families there are a number of devices each with different flavors of peripherals. To ascertain the available CPU, memory or peripheral mix for each of these families, see the device-specific data sheet.

These products could be implemented as unique silicon designs or they may be shared silicon designs that have elements disabled or not guaranteed by specification, even if present in silicon. Only the elements that are enabled in the device as specified in the device-specific data sheet and technical reference manual are to be used for safety feature enhancements or safety software implementation. Elements that are not part of the device, even though it is supported in the superset of the device family, are not guaranteed to be present and operate.

Table 2. C2000 MCU Features List

	C2000 32-Bit Microcontrollers - MCU					
	Picollo Family [1]				Delfino Family	
	TMS320F2806x	TMS320F2805x	TMS320F2803x	TMS320F2802x	TMS320F2833x	TMS320F2823x
32-Bit CPU	**90 MHz**	**80 MHz**	**60 MHz**	**60 MHz**	**150 MHz**	**150 MHz**
C28x - CPU	x	x	x	x	x	x
Floating-Point Unit (FPU)	x				x	
IQMath Capabilities	x	x	x	x	x	x
Viterbi Complex Math Unit (VCU)	x					
Control Law Accelerator (CLA)	x	x	x			
Direct Memory Access (DMA)	x				x	x
Memory						
Boot ROM	x	x	x	x	x	x
Random-Access Memory (RAM)	x	x	x	x	x	x
FLASH	x	x	x	x	x	x
Read-Only Memory (ROM)	x	x	x	x		
One-Time Programmable (OTP)	x	x	x	x	x	x
Peripheral Interrupts (PIE)	x	x	x	x	x	x
Control Peripherals						
32-Bit CPU Timers	x	x	x	x	x	x
PWM	x	x	x	x	x	x

(1) Piccolo2x also include Piccolo2x0 devices.

Table 2. C2000 MCU Features List (continued)

	C2000 32-Bit Microcontrollers - MCU					
	Picollo Family [1]				Delfino Family	
	TMS320F2806x	TMS320F2805x	TMS320F2803x	TMS320F2802x	TMS320F2833x	TMS320F2823x
32-Bit CPU	**90 MHz**	**80 MHz**	**60 MHz**	**60 MHz**	**150 MHz**	**150 MHz**
High-Resolution Pulse Width Modulator (HRPWM)	x		x	x	x	x
High-Resolution Capture (HRCAP)	x		x			
Quadrature Encoder Pulse (QEP)	x	x	x	x	x	x
Communications Peripherals						
Univeral Serial Bus (USB)	x					
Serial Peripheral Interface (SPI)	x	x	x		x	x
Serial Communications Interface (SCI)	x	x	x	x	x	x
Local Interconnect Network (LIN)	x		x			
Controller Area Network (CAN)	x	x	x		x	x
Inter-Integrated Circuit (I2C)	x	x	x	x	x	x
Multichannel Buffered Serial Port (McBSP)	x				x	x
External Memory Bus	x				x	x
Analog Peripherals						
2-Pin Oscillators	x		x	x	x	x
0-Pin Oscillators	x	x	x	x		
POR and BOR	x	x	x	x		
On-Chip Voltage Regulator (VREG)	x	x	x	x		
Temp Sensor	x	x	x	x		
Analog-to-Digital Converter (ADC)	x	x	x	x	x	x
OPAMP		x				
Digital-to-Analog Converter (DAC)	x	x	x	x		
Comparator	x	x	x	x		
Packaging	80 pin 100 pin	64 pin 80 pin	56 pin 64 pin 80 pin	38 pin 48 pin	176 pin 179 pin	176 pin 179 pin
Applications Space	Motor Control	Motor Control	Motor Control	Motor Control	Motor Control	Motor Control
	Energy Conversion		Energy Conversion	Energy Conversion	Energy Conversion	Energy Conversion
	Power Conversion	Power Conversion	Power Conversion	Power Conversion	Power Conversion	Power Conversion
	Power Line Comms				Automotive Radar	Automotive Radar

4.3 Piccolo MCU Family

Piccolo MCUs address a wide range of performance levels in real time control applications. The makeup of the family is differentiated at the 32-bit CPU performance, control and analog subsystem peripherals. Table 2 provides Piccolo MCU feature set break up. For detailed resource availability, see the device-specific data sheet on that section and on the TI web sites.

Piccolo MCUs have proven many control topologies in motor and other emerging applications. Tight integration of the 32-bit CPU performance and peripherals is mandatory to make these topologies a deterministic control system. Having proven in many of these applications, at the equipment level there is a demand to reach functional safety levels. Functional safety levels are governed by many of IEC and UL standards to enforce safety discipline in hardware and software development. IEC60730 and UL1998 have published a series of safety test across any microcontroller device.

The IEC60730 software library on Piccolo MCU architecture brings out the built-in hardware features and system level diagnostics to detect failures during power up and run time of the application. Since the Piccolo family is architecturally scalable to different features and cost segments, the IEC60730 library is mostly compatible to all its devices, except that some of the software functions need some porting to match the peripheral mix used in the end application system.

4.3.1 Type A: C28x CPU + FPU + VCU + CLA With Control and Analog Subsystem – Piccolo6x Family

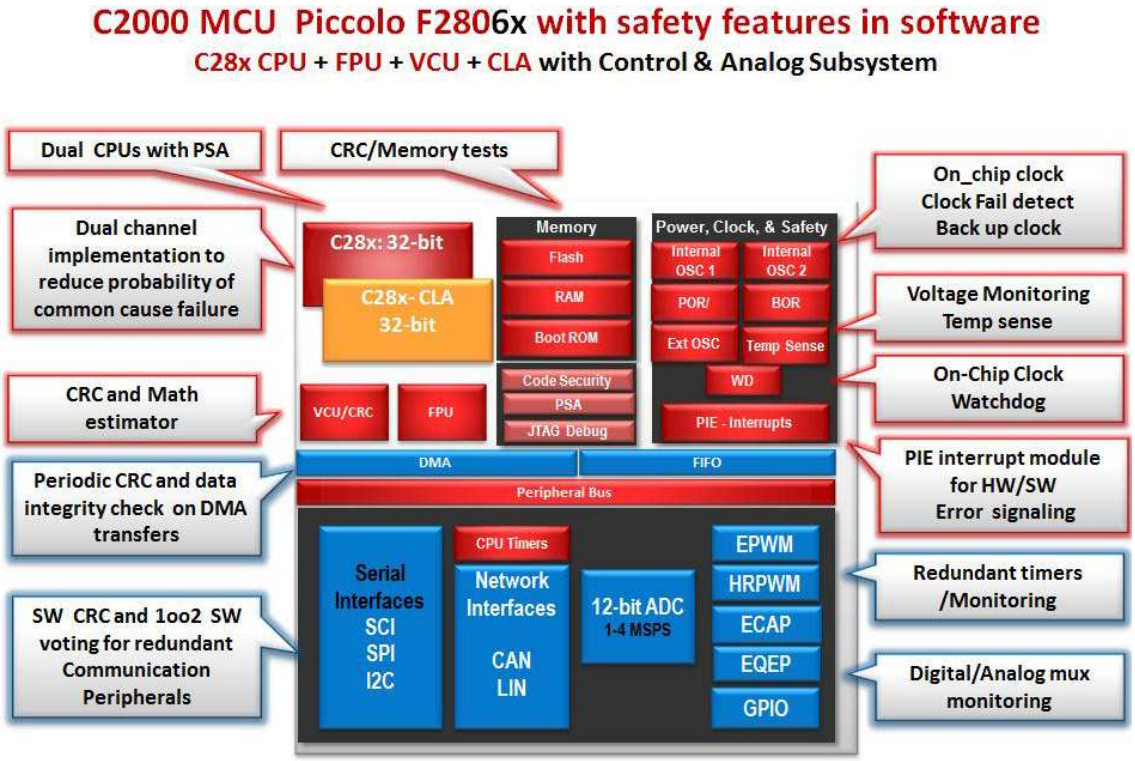

Figure 3. C2000 MCU Piccolo F2806x With Safety Features in Software

Piccolo6x: TMS320F2806x MCU is the full featured C2000 MCU with a dedicated 32-bit C28x main CPU, floating-point unit and Viterbi and math accelerators (VCU). CPU performance is further supported by an event optimized, CLA that is capable of 32-bit C28x CPU functions with floating-point precision. Having both a main C28x CPU and CLA accelerator on-chip enables many time critical functions, algorithms, and timing checks in parallel. This enables a virtual dual channel implementation by creatively partitioning CPUs and peripherals. Having two independent processors provides inherent safety for the system and protects against common cause failures.

The main CPU has a built-in 40-bit CRC checker, the parallel signature analysis (PSA) that can be used to do on-the-fly CRC calculations for any of the program code or memory blocks. This is a differentiating hardware working in parallel to the main CPU to estimate the CRC on the memory contents and CPU instructions. The IEC60730 software libraries have functions using the PSA hardware and details of its use conditions. While most MCUs can calculate CRC in software using CRC algorithms, the PSA hardware is a feature that prioritizes instruction and memory level integrity checks and enhances the safety coverage for the CPU and memory regions. In addition, the Piccolo6x architecture is capable of doing advanced CRC and math accelerations using its VCU unit. This offers a third method to do memory CRC calculations and algorithms. These features help to comply with the safety requirements easily.

Piccolo6x memory has non-volatile Flash, one-time programmable (OTP) Flash, read-only memory (BROM) and RAM blocks. PSA-based CRC or memory-based CRC and March 13 algorithms are made available to check memory integrity in periodic and power up software routines.

The system integrator is responsible for the functional partition of these processing engines, depending on the end application, as each application has different safety reactions and recovery times. The IEC60730 libraries enable easy implementation of safety software using the hardware and software functions.

Control Subsystem: Piccolo6x devices have proven control peripherals such as PWMs, enhanced capture (ECAP), and enhanced quadrature encoder pulse (EQEP) modules as part of the MCUs control subsystem. These hardware peripherals are instantiated in multiple instances on the same chip. This enables redundant channel implementation, thus enabling a higher level of safety in the end equipment. Having redundant channels enables 1oo2 (1 out of 2 architecture) level voting scheme on critical control channels.

The high-resolution pulse width modulator (HRPWM) module enhances the resolution of the PWM control and provides unique calibration logic to track the PWM modulation accuracy and input clock oscillator accuracy. The oscillator accuracy library functions use the SFO libraries, which use the inherent calibration logic, to measure the accuracy of the oscillator clock and PLL.

General-purpose input/output (GPIO) configuration registers are write-protected with special EALLOW instructions. Most of the GPIO inputs have a programmable de-glitching filter to screen spurious noise or debouncing. This is especially important for PWM output trip control signals, to avoid false tripping of PWM outputs.

The Control peripherals are complemented with three 32-bit CPU timers that can support time base checks, and as redundant channels to implement timing checks.

Piccolo MCUs use industry standard communication ports such as: controller area network (CAN), serial communications interface (SCI), serial peripheral interface (SPI), multichannel buffered serial port (McBSP) and inter-integrated circuit (I2C). These peripherals are often available in multiple instantiations that allow 1oo2 voting schemes and internal loop back paths that allow periodic serial data integrity checks.

Section 5 presents many of the functional safety diagnostics schemes that can be implemented to add safety in each of these MCU functions during run time and power up state.

Analog Subsystem: Piccolo6x analog subsystem supports all the necessary analog sensing and feedback channels. It has its own independent regulated power, oscillators, temperature sensor and PLL clock. At runtime, it is synchronized and tightly coupled to the main CPU and control subsystems. Piccolo MCU addresses the safety requirements of the operating clock redundancy with a minimum of two clocks sources and an additional external clock source. There are two zero-pin oscillators operating in parallel, OSC1 and OSC2, an additional on-chip crystal oscillator, and an external clock pin.

The oscillator clocks feed the on-chip watchdog module to provide time-out monitoring of time critical loops. Device reset is offered through an external pin. This is a bidirectional pin allowing on-chip watchdog reset to propagate to external devices. The reset input signal is always qualified with a deglitching filter (about 200 ns) that helps to block reset line noise due to any electrical disturbance.

The dependencies of these clock domains and recovery mechanisms are presented in the Piccolo technical reference manual (see Table 1). The on-chip regulator supports POR and BOR logic to generate under voltage detection.

4.3.2 Type B: C28x CPU + CLA With Control and Analog Subsystem – Piccolo3x Family

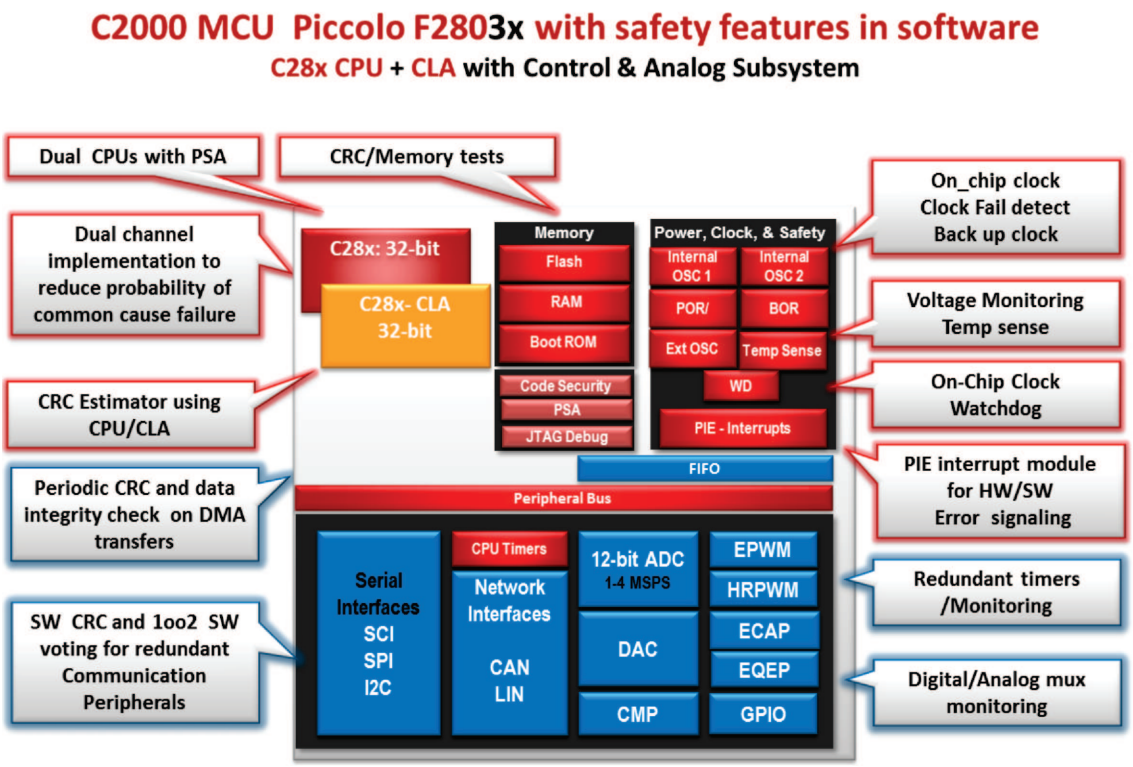

Figure 4. C2000 MCU Piccolo F2803x With Safety Features in Software

Piccolo3x: TMS320F2803x MCU is functionally identical to TMS320F2806x except for the FPU, VCU and USB modules. It has a dedicated 32-bit C28x main CPU. CPU performance is further supported by an event optimized, CLA that is capable of 32-bit C28x CPU functions with floating point precision. Having both a main C28x CPU and CLA on-chip enables running many time critical functions, algorithms, and timing checks in parallel. This enables a virtual dual channel implementation by creatively partitioning CPUs and peripherals. Having two independent processors provides inherent safety for the system and protects against common cause failures.

The main CPU has a built-in 40-bit CRC checker, the PSA that can be used to do on-the-fly CRC calculations for any of the program code or memory blocks. This is a differentiating hardware working in parallel to the main CPU to estimate the CRC on the memory contents and CPU instructions. The IEC60730 software libraries have functions using the PSA hardware and details of its use conditions. While most MCUs can calculate CRC in software using CRC algorithms, the PSA hardware is a feature that prioritizes instruction and memory level integrity checks and enhances the safety coverage for the CPU and memory regions. In addition, the Piccolo6x architecture is capable of doing advanced CRC and math accelerations using its VCU unit. This offers a third method to do memory CRC calculations and algorithms. These features help to comply with the safety requirements easily.

Piccolo3x memory has non-volatile Flash, OTP Flash, read-only memory (BROM) and RAM blocks. PSA-based CRC or memory-based CRC and March 13 algorithms are made available to check memory integrity in periodic and power up software routines.

The system integrator is responsible for the functional partition of these processing engines, depending on the end application, as each application has different safety reactions and recovery times. The IEC60730 libraries enable easy implementation of safety software using the hardware and software functions.

Control Subsystem: Piccolo3x devices have proven control peripherals such as PWMs, ECAP, and EQEP modules as part of the MCUs control subsystem. These hardware peripherals are instantiated in multiple instances on the same chip. This enables redundant channel implementation, thus enabling a higher level of safety in the end equipment. Having redundant channels enables 1oo2 (1 out of 2 architecture) level voting scheme on critical control channels.

The HRPWM module enhances the resolution of the PWM control and provides unique calibration logic to track the PWM modulation accuracy and input clock oscillator accuracy. The oscillator accuracy library functions use the SFO libraries, which use the inherent calibration logic, to measure the accuracy of the oscillator clock and PLL.

GPIO configuration registers are write-protected with special EALLOW instructions. Most of the GPIO inputs have a programmable de-glitching filter to screen spurious noise or debouncing. This is especially important for PWM output trip control signals, to avoid false tripping of PWM outputs.

The Control peripherals are complemented with three 32-bit CPU timers that can support time base checks, and as redundant channels to implement timing checks.

Piccolo MCUs use industry standard communication ports such as: CAN, SCI, local interconnect network (LIN), SPI, and I2C. These peripherals are often available in multiple instantiations that allow 1oo2 voting schemes and internal loop back paths that allow periodic serial data integrity checks.

Section 5 presents many of the functional safety diagnostics schemes that can be implemented to add safety in each of these MCU functions during run time and power up state.

Analog Subsystem: Piccolo3x analog subsystem supports all the necessary analog sensing and feedback channels. It has its own independent regulated power, oscillators, temperature sensor and PLL clock. At runtime, it is synchronized and tightly coupled to the main CPU and control subsystems. Piccolo MCU addresses the safety requirements of the operating clock redundancy with a minimum of two clocks sources and an additional external clock source. There are two zero-pin oscillators operating in parallel, OSC1 and OSC2, an additional on-chip crystal oscillator, and an external clock pin.

The oscillator clocks feed the on-chip watchdog module to provide time-out monitoring of time critical loops. Device reset is offered through an external pin. This pin is a bidirectional pin allowing on-chip watchdog reset to propagate to external devices. The reset input signal is always qualified with a deglitching filter (about 200 ns) that helps to block reset line noise due to any electrical disturbance.

The dependencies of these clock domains and recovery mechanisms are presented in the Piccolo technical reference manual (see Table 1). The on-chip regulator supports POR and BOR logic to generate under voltage detection.

4.3.3 Type B: C28x CPU + CLA With Control and Analog Subsystem – Piccolo5x Family

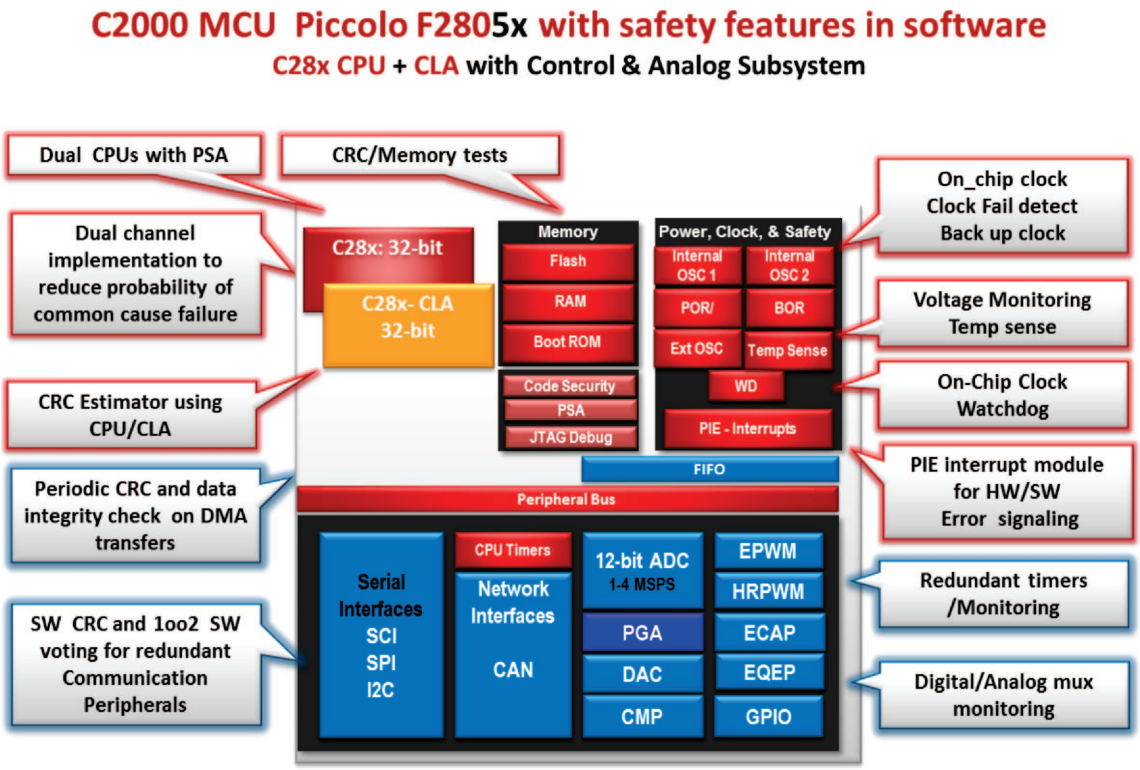

Figure 5. C2000 MCU Piccolo F2805x With Safety Features in Software

Piccolo5x: TMS320F2805x MCU is functionally identical to TMS320F2803x except for the PGA, CAN and LIN modules. It has a dedicated 32-bit C28x main CPU supports fixed point and IQmath capabilities to accelerate time critical functions, and control algorithms.

The main CPU has a built-in 40-bit CRC checker, the PSA that can be used to do on-the-fly CRC calculations for any of the program code or memory blocks. This is a differentiating hardware working in parallel to the main CPU to estimate the CRC on the memory contents and CPU instructions. The IEC60730 software libraries have functions using the PSA hardware and details of its use conditions. While most MCUs can calculate CRC in software using CRC algorithms, the PSA hardware is a feature that prioritizes instruction and memory level integrity checks and enhances the safety coverage for the CPU and memory regions. In addition, the Piccolo6x architecture is capable of doing advanced CRC and math accelerations using its VCU unit. This offers a third method to do memory CRC calculations and algorithms. These features help to comply with the safety requirements easily.

Piccolo5x memory has non-volatile Flash, OTP Flash, read-only memory (BROM) and RAM blocks. PSA-based CRC or memory-based CRC and March 13 algorithms are made available to check memory integrity in periodic and power up software routines.

The system integrator is responsible for the functional partition of these processing engines, depending on the end application, as each application has different safety reactions and recovery times. The IEC60730 libraries enable easy implementation of safety software using the hardware and software functions.

Control Subsystem: Piccolo5x devices have proven control peripherals such as PWMs, ECAP, and EQEP modules as part of the MCUs control subsystem. These hardware peripherals are instantiated in multiple instances on the same chip. This enables redundant channel implementation, thus enabling a higher level of safety in the end equipment. Having redundant channels enables 1oo2 (1 out of 2 architecture) level voting scheme on critical control channels.

The HRPWM module enhances the resolution of the PWM control and provides unique calibration logic to track the PWM modulation accuracy and input clock oscillator accuracy. The oscillator accuracy library functions use the SFO libraries, which use the inherent calibration logic, to measure the accuracy of the oscillator clock and PLL.

GPIO configuration registers are write-protected with special EALLOW instructions. Most of the GPIO inputs have a programmable de-glitching filter to screen spurious noise or debouncing. This is especially important for PWM output trip control signals, to avoid false tripping of PWM outputs.

The Control peripherals are complemented with three 32-bit CPU timers that can support time base checks, and as redundant channels to implement timing checks.

Piccolo MCUs use industry standard communication ports such as: CAN, SCI, SPI, and I2C. These peripherals are often available in multiple instantiations that allow 1oo2 voting schemes and internal loop back paths that allow periodic serial data integrity checks.

Section 5 presents many of the functional safety diagnostics schemes that can be implemented to add safety in each of these MCU functions during run time and power up state.

Analog Subsystem: Piccolo5x analog subsystem supports all the necessary analog sensing and feedback channels. It has its own independent regulated power, oscillators, temperature sensor and PLL clock. At runtime, it is synchronized and tightly coupled to the main CPU and control subsystems. Piccolo MCU addresses the safety requirements of the operating clock redundancy with a minimum of two clocks sources and an additional external clock source. There are two zero-pin oscillators operating in parallel, OSC1 and OSC2, an additional on-chip crystal oscillator, and an external clock pin.

The oscillator clocks feed the on-chip watchdog module to provide time-out monitoring of time critical loops. Device reset is offered through an external pin. This pin is a bidirectional pin allowing on-chip watchdog reset to propagate to external devices. The reset input signal is always qualified with a deglitching filter (about 200 ns) that helps to block reset line noise due to any electrical disturbance.

The dependencies of these clock domains and recovery mechanisms are presented in the Piccolo technical reference manual (see Table 1). The on-chip regulator supports POR and BOR logic to generate under voltage detection.

4.3.4 Type D: C28x CPU With Control and Analog Subsystem – Piccolo2x Family

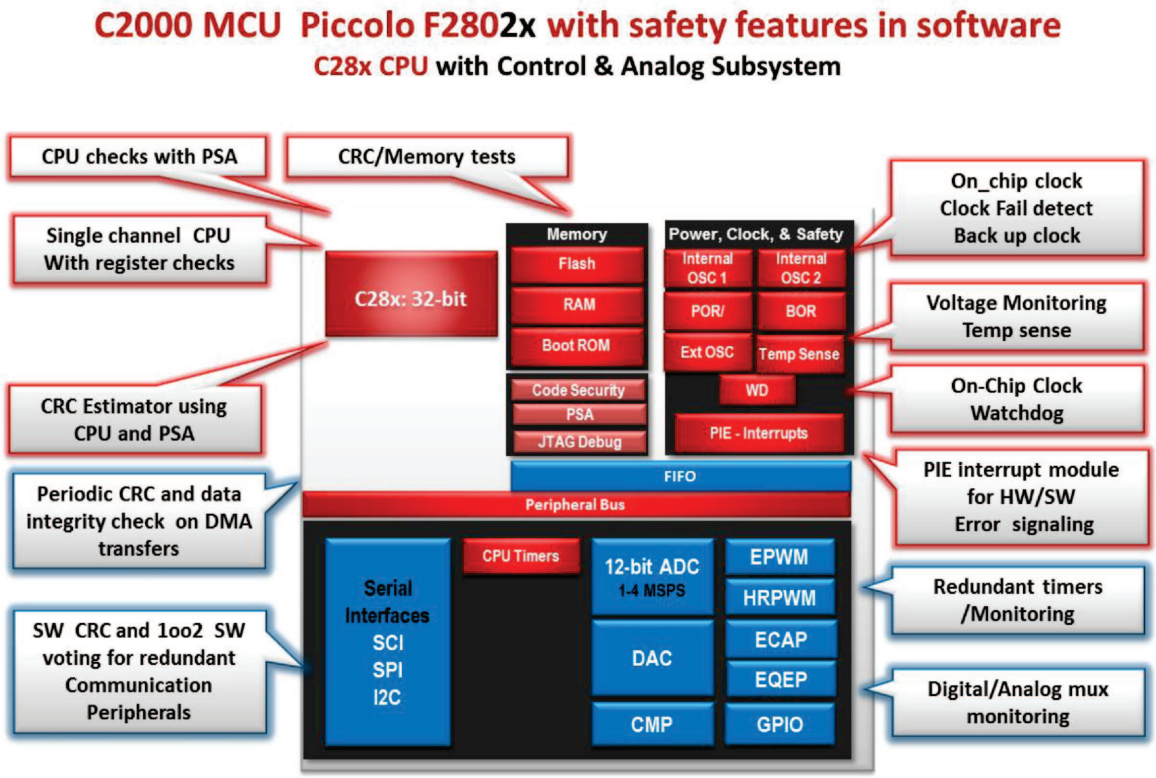

Figure 6. C2000 MCU Piccolo F2802x With Safety Features in Software

Piccolo2x: TMS320F2802x MCU is functionally identical to TMS320F2803x except for the CLA, CAN and LIN modules. It has a dedicated 32-bit C28x main CPU supports fixed point and IQmath capabilities to accelerate time critical functions, and control algorithms.

The main CPU has a built-in 40-bit CRC checker, the Parallel Signature Analysis (PSA) that can be used to do on-the-fly CRC calculations for any of the program code or memory blocks. This is a differentiating hardware working in parallel to the main CPU to estimate the CRC on the memory contents and CPU instructions. The IEC60730 software libraries have functions using the PSA hardware and details of its use conditions. While most MCUs can calculate CRC in software using CRC algorithms, the PSA hardware is a feature that prioritizes instruction and memory level integrity checks and enhances the safety coverage for the CPU and memory regions. In addition, the Piccolo6x architecture is capable of doing advanced CRC and math accelerations using its VCU unit. This offers a third method to do memory CRC calculations and algorithms. These features help to comply with the safety requirements easily.

Piccolo2x memory has non-volatile Flash, OTP Flash, read-only memory (BROM) and RAM blocks. PSA-based CRC or memory-based CRC and March 13 algorithms are made available to check memory integrity in periodic and power up software routines.

The system integrator is responsible for the functional partition of these processing engines, depending on the end application, as each application has different safety reactions and recovery times. The IEC60730 libraries enable easy implementation of safety software using the hardware and software functions.

Control Subsystem: Piccolo2x devices have proven control peripherals such as PWMs, ECAP, and EQEP modules as part of the MCUs control subsystem. These hardware peripherals are instantiated in multiple instances on the same chip. This enables redundant channel implementation, thus enabling a higher level of safety in the end equipment. Having redundant channels enables 1oo2 (1 out of 2 architecture) level voting scheme on critical control channels.

The HRPWM module enhances the resolution of the PWM control and provides unique calibration logic to track the PWM modulation accuracy and input clock oscillator accuracy. The oscillator accuracy library functions use the SFO libraries, which use the inherent calibration logic, to measure the accuracy of the oscillator clock and PLL.

GPIO configuration registers are write-protected with special EALLOW instructions. Most of the GPIO inputs have a programmable de-glitching filter to screen spurious noise or debouncing. This is especially important for PWM output trip control signals, to avoid false tripping of PWM outputs.

The Control peripherals are complemented with three 32-bit CPU timers that can support time base checks, and as redundant channels to implement timing checks.

Piccolo MCUs use industry standard communication ports such as; SCI, SPI, and I2C. These peripherals are often available in multiple instantiations that allow 1oo2 voting schemes and internal loop back paths that allow periodic serial data integrity checks.

Section 5 presents many of the functional safety diagnostics schemes that can be implemented to add safety in each of these MCU functions during run time and power up state.

Analog Subsystem: Piccolo2x analog subsystem supports all the necessary analog sensing and feedback channels. It has its own independent regulated power, oscillators, temperature sensor and PLL clock. At runtime, it is synchronized and tightly coupled to the main CPU and control subsystems. Piccolo MCU addresses the safety requirements of the operating clock redundancy with a minimum of two clocks sources and an additional external clock source. There are two zero-pin oscillators operating in parallel, OSC1 and OSC2, an additional on-chip crystal oscillator, and an external clock pin.

The oscillator clocks feed the on-chip watchdog module to provide time-out monitoring of time critical loops. Device reset is offered through an external pin. This pin is a bidirectional pin allowing on-chip watchdog reset to propagate to external devices. The reset input signal is always qualified with a deglitching filter (about 200 ns) that helps to block reset line noise due to any electrical disturbance.

The dependencies of these clock domains and recovery mechanisms are presented in the Piccolo technical reference manual (see Table 1). For more information, see Table 1. The on-chip regulator supports POR and BOR logic to generate under voltage detection.

4.4 Delfino MCU Family

Delfino is high performance member of the C2000 MCU addressing a wide of industrial applications. The makeup of the family is differentiated at the 32-bit C28x CPU performance, control and analog subsystem peripherals. These offer devices offer 100 to 150MHz CPU Performance. The Table 2 above provides Delfino MCU feature set break up between F2833x and F2823x devices. For detailed resource availability, see the device-specific data sheet for that section and on the TI web site.

Integration of the 32-bit CPU performance and peripherals is mandatory to make these topologies a deterministic control system. Having proven in many of these applications, at the equipment level there is a demand to reach the emerging functional safety levels. Functional safety levels are governed by many of IEC and UL standards to enforce safety discipline in hardware and software development. IEC60730 and UL1998 have published a series of safety test across any microcontroller device. Some of the Delfino applications target IEC61508 level safety as well. These configurations are primarily in industrial drive application co-designed with companion chips to reach required level system level safety.

Since Delfino MCU's architecture is identical to the Piccolo MCU with less analog integration. The following sections will explain the device configurations. Delfino devices have to be used with companion chips to implement some of the functional safety requirements. Most of the Piccolo6x, IEC60730 software library functions are applicable to Delfino devices. Delfino version of IEC60730 will leverage the Piccolo6x library to build its system level diagnostics. Delfino family is architecturally scalable to different feature and cost segments, the IEC60730 library is mostly compatible to all of its devices, except that some of the software functions need some porting to match the peripheral mix used in the end application system.

4.4.1 Type C: C28x CPU + FPU With Control and Analog Subsystem – Delfino F2833x Family

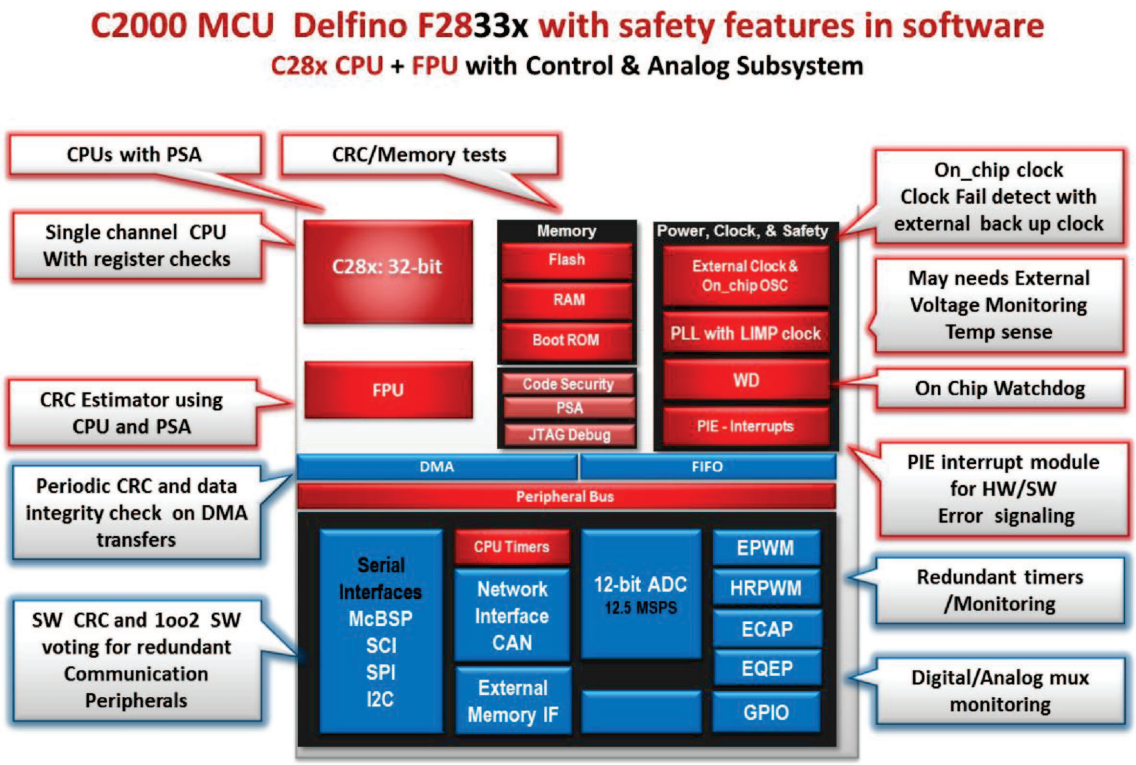

Figure 7. C2000 MCU Delfino F2833x With Safety Features in Software

Delfino: TMS320F2833x MCU is the full featured C2000 MCU with a dedicated 32-bit C28x main CPU, floating point unit, designed for high performance (150MHz)). CPU performance is capable of fixed point math, IQmath and floating point precision. The main CPU has a built-in 40-bit CRC checker, the PSA that can be used to do on-the-fly CRC calculations for any of the program code or memory blocks. This is a differentiating hardware working in parallel to the main CPU to estimate the CRC on the memory contents and CPU instructions. The IEC60730 software libraries have functions using the PSA hardware and details of its use conditions. While most MCUs can calculate CRC in software using CRC algorithms, the PSA hardware is a feature that prioritizes instruction and memory level integrity checks and enhances the safety coverage for the CPU and memory regions.

Delfino memory has non-volatile Flash, OTP Flash, read-only memory (BROM) and RAM blocks. PSA-based CRC or memory-based CRC and March 13 algorithms are made available to check memory integrity in periodic and power up software routines.

The system integrator is responsible for the functional partition of these processing engines, depending on the end application, as each application has different safety reactions and recovery times. The IEC60730 libraries enable easy implementation of safety software using the hardware and software functions.

Control Subsystem: Delfino devices have proven control peripherals such as PWMs, ECAP, and EQEP modules as part of the MCUs control subsystem. These hardware peripherals are instantiated in multiple instances on the same chip. This enables redundant channel implementation, thus enabling a higher level of safety in the end equipment. Having redundant channels enables 1oo2 (1 out of 2 architecture) level voting scheme on critical control channels.

The HRPWM module enhances the resolution of the PWM control and provides unique calibration logic to track the PWM modulation accuracy and input clock oscillator accuracy. The oscillator accuracy library functions use the SFO libraries, which use the inherent calibration logic, to measure the accuracy of the oscillator clock and PLL.

GPIO configuration registers are write-protected with special EALLOW instructions. Most of the GPIO inputs have a programmable de-glitching filter to screen spurious noise or debouncing. This is especially important for PWM output trip control signals, to avoid false tripping of PWM outputs.

The Control peripherals are complemented with three 32-bit CPU timers that can support time base checks and as redundant channels.

Delfino MCUs use industry standard communication ports such as; CAN, SCI, SPI, McBSP and I2C. These peripherals are often available in multiple instantiations that allow 1oo2 voting schemes and internal loop back paths that allow periodic serial data integrity checks.

Section 5 presents many of the functional safety diagnostics schemes that can be implemented to add safety in each of these MCU functions during run time and power up state.

Analog Subsystem: Delfino analog subsystem supports all the necessary analog sensing and feedback channels. It has limited analog resources compared to the Piccolo family. Analog enhancement is primarily on the ADC convertor module. On-chip ADC is pipeline architecture capable of 12.5 MSPS to integrate sophisticated DSP algorithms in automotive radar and solar invertor applications. The device needs dual rail external 3.3 V and 1.8 V power supplies. It supports a crystal oscillator, external clock in and a PLL clock. PLL also supports a limp mode clock for clock recovery for safe shut down during input clock failure. External clock voltage monitors and check-micro helps to meet necessary functional safety compliance.

The oscillator clocks feed the on-chip watchdog module to provide time-out monitoring of time critical loops. Device reset is offered through an external pin. This pin is a bidirectional pin allowing on-chip watchdog reset to propagate to external devices. The reset input signal is always qualified with a deglitching filter (about 200 ns) that helps to block reset line noise due to any electrical disturbance.

4.4.2 Type D: C28x CPU With Control and Analog Subsystem – Delfino F2823x Family

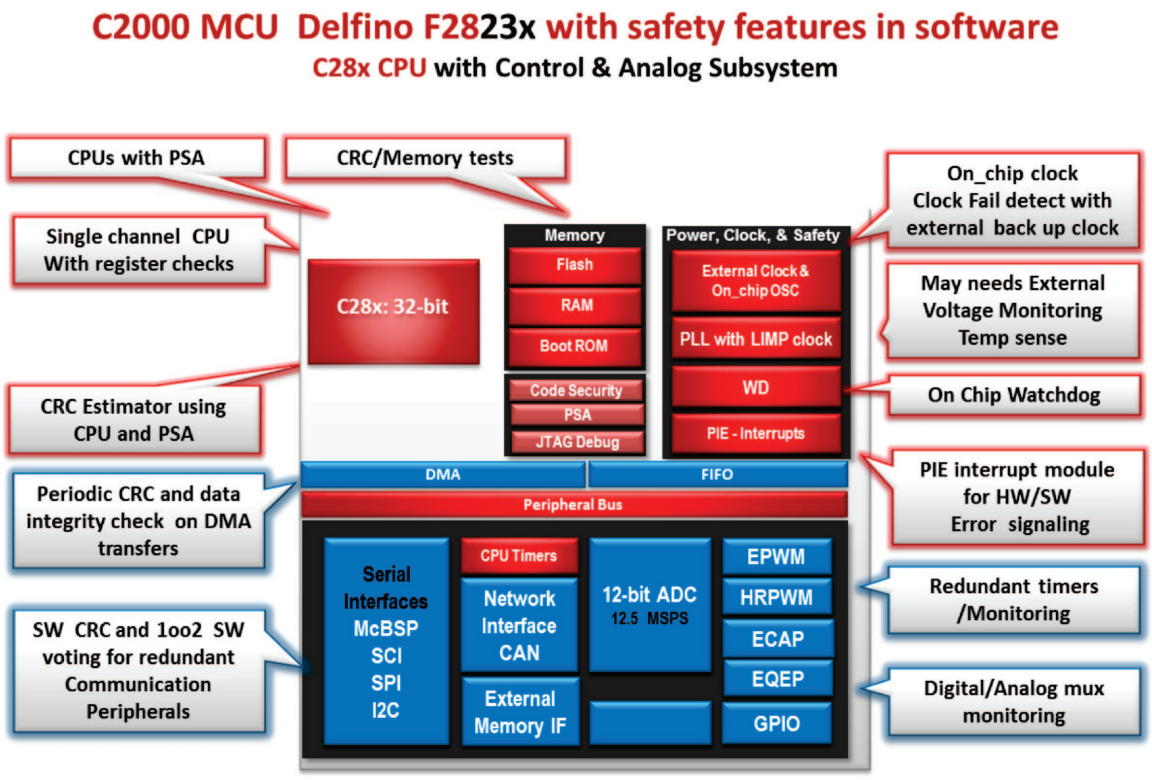

Figure 8. C2000 MCU Delfino F2823x With Safety Features in Software

Delfino: TMS320F2823x MCU is functionally identical to TMS320F2833x family except for the floating-point unit. For more details on this family, see Table 2.

4.5 C2000 MCUs Operating States

The C2000 MCU products have a common architectural definition of operating states. These operating states should be observed by the system developer in their software and system level design concepts. The operating states state machine is shown in Figure 9 and described below.

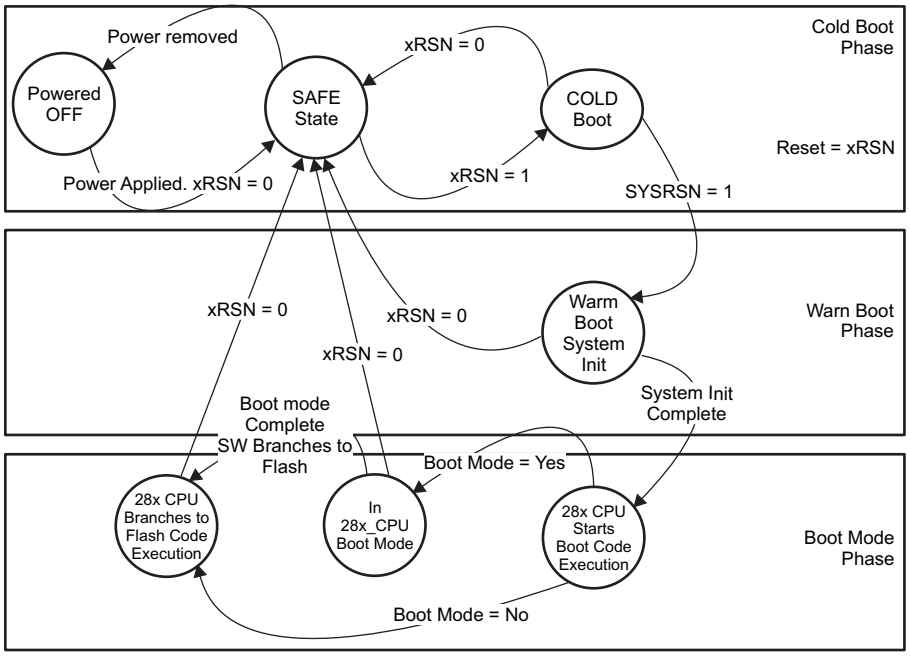

Figure 9. Piccolo and Delfino MCUs Operating States

- Powered Off - This is the initial operating state of the C2000 MCUs. No power is applied to either core (V_{DD} –1.8 V/1.2 V) or I/O power supply (V_{DDIO} –3.3 V) and the device are non-functional. This state can only transition to the safe state, and can only be reached from the safe state.
- Safe - In the safe state, the 28x CPU and its subsystems are powered but non-operational. The XRSN (power-on reset, also known as cold reset) is asserted by the system but is not released until power supplies have ramped to a stable state. The internal voltage monitor (POR and BOR) safety mechanism also continues to assert the SYSRSN internal to the device if power supplies are not within a minimum operational range. When the product is in the safe state, the CPUs and peripherals are non-functional. Output drivers are tri-stated and input/output pins are kept in an input only state.
- Cold Boot - In the cold boot state, key analog elements, digital control logic, and debug logic are initialize for future use. The CPUs and the peripherals remain powered but non-operational. When the cold boot process is completed, the SYSRSN signal is internally released, leading to the warm boot stage.
- Warm Boot - The warm boot mode resets digital logic and enables the C28x CPU. The C28x_CPU begins executing software from Boot ROM, initializes the system and awaits boot mode. There is no hardware interlock to say that warm boot is completed; this is a software decision in C28x Boot code.
- Operational and Boot Phase - In this state, only the C28x CPU enters Boot phase, checks for boot modes and enters boot function if valid else C28x CPU branches to flash memory to do software initialization and application code execution defined for the end application.

4.6 Management of Exception and Errors

The Piccolo and Delfino MCU product architectures leverage CPU interrupt and peripheral interrupt expansion (PIE) blocks for event triggers and preemption of software functions. These interrupts can be prioritized using the PIE module and respond to respective interrupt flags from various peripherals. These interrupts can also be activated using software interrupts instruction such as TRAP. All the hardware safety mechanism use these interrupt flags to generate error conditions to the CPU. The IEC60730 safety library routines can be tied to either hardware interrupt flags or software interrupt initiated functions to preempt CPU to do necessary safety recovery or reach safe state. The IEC60730 libraries support an error and exception reporting function using the on-chip UART and SCI port, providing an external communication port-based error function allows system error logging during validation and runtime use of the application. This function can be easily customized in the end application based on the error reporting needs. For details of PIE and CPU interrupt mapping, see the Piccolo or Delfino MCU technical reference manuals (see Table 1).

5 C2000 MCU Architecture Safety Mechanisms and Assumptions of Use

You, as a system and equipment manufacturer or designer, are responsible to ensure that your systems (any TI hardware or software components incorporated in your 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. You understand and agree that your use of TI components in safety critical applications is entirely at your risk, and that you (as buyer) agree to defend, indemnify, and hold harmless TI from any and all damages, claims, suits, or expense resulting from such use. In this section, the safety mechanisms for each major functional block of the C2000 MCU architecture are summarized and general assumptions of use are provided. This information should be used to determine the strategy for utilizing functional safety mechanisms. The details of each safety mechanism can be found in the device-specific technical reference manual for the MCU used.

TI classifies technical recommendations for the use of safety mechanisms into a number of categories. The TI recommendations should not be considered infallible. There are many diverse ways to implement safe systems and alternate safety mechanisms may be possible that can provide support to achieve desired safety metrics. The categories of recommendation are as follows:

- **Mandatory** - A mandatory notation indicates a safety mechanism that is always operable during normal functional operation and cannot be disabled by user action.
- **High Recommended** - A highly recommended notation indicates a safety mechanism that TI believes to provide a high value of diagnostics, which are difficult to implement by other means. The user retains the choice of whether or not to utilize the safety mechanism in their design, as user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.
- **Recommended** - A recommended notation indicates a safety mechanism that TI believes to provide a valuable diagnostic, which may also be implemented by other means. The user retains the choice of whether or not to utilize the safety mechanism in their design, as user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.
- **Optional** - An optional notation indicates a safety mechanism that TI believes to provide a lower value diagnostic, which may also be implemented by other means. The user retains the choice of whether or not to utilize the safety mechanism in their design, as user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.

5.1 Standard Safety Diagnostic Functions With C2000 MCUs and Subsystems

The C2000 MCU system has several functional safety features that can initiate fault or error interrupts when the fault occurs. However, except for a few special hardware modules, not all of these safety modules are able to generate alarms in their dormant state. The IEC60730 class of safety software libraries help to monitor all the MCU regions, CPU, memory and peripherals and generate error conditions, if they show any abnormalities. For example, memory locations may show failure due to device marginality or external influence during application runtime. These error conditions can be detected using periodic execution of IEC60730 memory check algorithms.

Essentially, CPUs, registers or subsystem failures usually remain dormant or inactive unless a software or hardware mechanism monitors them periodically. Table 3 lists the commonly used safety diagnostic functions that enable safety features and attributes across subsystems. These functions are referenced in Table 6.

These functions are the recommended common methods that can be enhanced as needed to suite each application with its internal or external hardware assist mechanisms.

Table 3. Functional Descriptions of Safety Modules in C2000 MCUs

Safety Diagnostics Functions to Enable Safety Attributes and Features	Description
Self Test , Autocoverage	Periodic execution enables inherent self-checking features. This helps to detect latent faults.
Periodic Read Back	Periodic read back of configuration registers can provide a diagnostic for inadvertent writes or disturb of these registers. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. This is a recommended operation.
Read Back of Written Configuration	In order to ensure proper configuration of memory-mapped control registers, it is highly recommended that software implement a test to confirm proper operation of all control register writes.
Interrupt Sweep	C28x CPUs support interrupt controllers, each have expansion modules to address wide range of interrupts. These interrupts sources are managed by enable and mask registers for prioritization. Interrupt sweep tests are internal self-tests that trigger interrupts when each CPU sets the interrupt flag registers. This function enables all the interrupt flags, sequentially, blocking the peripheral sources to check periodicity and correct occurrence of interrupts. This is a recommended function.
Run Time Code Trace	This function is a practical and easy method to trace the program execution to detect orderly execution of the critical interrupts among the configured peripherals. In most systems, program code jumps and branches in a predetermined code execution path, particularly in critical control loops. Tracing the hardware or software semaphores as a signature of orderly entry and exit checks proper timing of interrupts and peripheral functionality. This is a highly recommended function.
CRC and Read and Write	Piccolo and Delfino MCUs do not support ECC and parity logic for memory blocks to provide error correction. To enable the memory safety and autocoverage, a CRC calculation and periodic memory read and write operation has to be implemented. This function is a highly recommended function.
Loopback Tests	Not all communication ports have data integrity check during transmit sessions. However, most of these peripherals do have an internal loopback. This function is intended to periodically enable loopback mechanism and check for data integrity. It is recommended these tests are done with IO mux in input mode to avoid external signal visibility at the pin level, during test. This function is a recommended function.
SFO Clock Accuracy Check	Clock timing and frequency check is necessary to guarantee proper operations of the clocking mechanisms. This ensures the overall timing dependencies are with operating conditions. Scale factor Optimizer (SFO) functions are software libraries that are built with internal hardware assist logic in C2000 MCU devices that can help to measure and clock accuracies. Details of SFO functions are in explained in the *TMS320x2806x Piccolo Technical Reference Manual* (SPRUH18). This is a highly recommended function.
PSA Function	Parallel signature analysis is a hardware assist safety unit, part of C28x architecture that can calculate 40-bit CRC across specified memory buses. This function is a software function with specific instructions to enable the PSA feature and calculate the CRC of memory reads and writes and program fetches. This function is available as part of the IEC60730 library function to the C28x CPU. This is a highly recommended function.
Handshake Function	All of the dual CPU (CPU +CLA) systems require periodic interprocessor communication exchange for safe operation and to implement virtual dual channel topologies. Implementing software and hardware linked handshake function, during runtime help to maintain timing and system integrity. These are user application specific functions and can be easily adapted to build the system level handshakes. This ensures reliable data exchange and status monitoring. This is a highly recommended function.
Custom Application Module	Most MCU firmware implements custom hardware or software functions (IP intellectual property) to differentiate the application. These are either ROM/Flash or RAM-based functions invoked during runtime of the applications. These functions should have explicit data integrity checks (program code CRC) and functional checks using test vectors ensuring the custom software block will function correctly for the intended application.
1oo2 Software Voting	One of two channels scheme is a method to implement software voting of correct functioning using similar peripherals, with one of the spare peripherals acting as a reference to compare. For example, use spare and redundant Timer resources to compare the correct functioning of the primary Timer resources. This is a recommended function.

Table 3. Functional Descriptions of Safety Modules in C2000 MCUs (continued)

Safety Diagnostics Functions to Enable Safety Attributes and Features	Description
Information Redundancy Techniques	Information redundancy techniques can be applied via software as an additional runtime diagnostic on any memory block, serial communication peripherals and control peripherals. There are many techniques that can be applied, such as read back of written values and multiple reads of the same target data with comparison of results. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. The use of information redundancy techniques as a means to do functional safety diagnostic is a highly recommended or recommended function depending the criticality of the peripheral block in the end system.

5.2 Functional Descriptions of Safety Modules in C2000 MCUs

The following section briefly describes the functional use of some of the Piccolo and Delfino MCU modules to enhance safety modes. For more information, see the block diagrams for the Piccolo and Delfino MCU families mentioned in Section 4.

5.2.1 Power Supply

The Piccolo device family requires an external regulator device to supply the necessary voltages and currents for proper operation. Piccolo devices are capable of taking independent power rails one for the analog and digital logic, respectively. The device operates with external 3.3 V/1.8 V (V_{DDIO}/V_{DD}) supplies when their internal regulators are not enabled. Most systems use internal regulators to provide the logic power (V_{DD} 1.8 V) from 3.3 V input supplies. The device also supports internal POR and BOR to monitor proper regulation of the internal regulator. This helps to monitor under voltage conditions and initiate interrupts or reset. For the recommended tolerances of these supplies, see the device-specific data sheet and the *Concerto F28M35x Technical Reference Manual* (SPRUH22).

External voltage supervisors for under voltage and overvoltage conditions are necessary if the on-chip detection thresholds are not meeting the performance goals. It is the responsibility of the system integrator to make the safety tradeoffs.

The Delfino family requires external power supply and voltage supervisors as they do not have on-chip regulators.

5.2.2 External Voltage Supervisor

The Piccolo platform highly recommends the use of an external voltage supervisor to monitor overvoltage protection for input power rails. Overvoltage threats are more common to industrial products and, therefore, an external supervisor complements the system topologies. Most overvoltage detection circuits can warn Piccolo subsystems quickly to avoid catastrophic shutdown of control systems. This is applicable to Delfino systems as well.

5.2.3 Reset

The Piccolo and Delfino device family products require an external reset at cold and warm power-up cycles to place all asynchronous and synchronous logic into a known state. The power-on reset generates an internal warm reset signal to reset the majority of the subsystem blocks, before it initiates the boot process. The XRSN reset signal is a bidirectional pin that toggles the reset pin when asserted internally through watchdog. In some instances these reset can be directed as an interrupt to the CPU to monitor functionality of the watchdog modules. For more information on the reset functionality, see the device-specific data sheet.

the XRSN pin has an internal glitch filter active all the time and effective for any reset input. These structures filter out noise and transient signal spikes on the input reset pins in order to reduce unintended activation of the reset circuitry. The glitch filters are continuously operating and their behavior cannot be changed by the software. Use of the glitch filters is mandatory.

5.2.4 Clocks

The Piccolo devices are primarily synchronous logic devices and as such require clock signal for proper operation. The clock management logic is elaborate to provide synchronous clock across subsystems and provide failsafe clock when there is main clock failure. Piccolo devices have two internal oscillators, OSC1 and OSC2, and in some derivative an external oscillator and clock input to provide standalone clock for the device. For the clock selection and failsafe switching, see the Piccolo MCU data sheet and user guide mentioned in the Table 1.

Delfino devices do not support on-chip oscillators; they support the on-chip crystal oscillator and external clock only. For details, see device-specific data sheets mentioned in Table 1.

5.2.5 CPU Interrupts for Illegal Instructions

28x CPU has an inherent feature to detect illegal instruction fetch and cause interrupt for safe recovery. For specific usage of this instruction, see the *TMS320C28x CPU and Instruction Set Reference Guide* (SPRU430). The user applications should include error recovery interrupt service routine to manage this error in the end application.

5.2.6 Division by Zero Conditions

28x CPU instructions do not have a hardware feature to detect division by zero operation. User algorithm, if it builds a division function using the 28x instructions, then the user code should add an explicit check for division by zero conditions. The FPU unit in the subsystems with 28x CPU, has hardware flags to detect division by zero error conditions when FPU instructions are used. User algorithms should include error check using this flag after division operations. For more details, see the *TMS320C28x Floating Point Unit and Instruction Set Reference Guide* (SPRUEO2).

5.2.7 Low-Power Mode Wake Up Checks

Piccolo and Delfino MCU devices support several modes of low power mode to save power. System designer should consider appropriate method to wake up with input triggers signals and also add redundant recovery mode, in case the device wake inputs fail to work.

6 Next Steps in Your Safety Development

TI's support for your safety development does not stop with the delivery of the safety manual. Customers have a wide range of support options, such as:

- SafeTI solutions and product support on the web: http://www.ti.com/ww/en/functional_safety/safeti/SafeTI-60730.html
- Access C2000 MCU page s any time on the web: http://www.ti.com/lsds/ti/microcontroller/32-bit_c2000/overview.page
- The Piccolo Wiki page provides answers to many commonly asked questions: http://www.ti.com/lsds/ti/microcontroller/32-bit_c2000/training.page

Table 4. C2000 MCU IEC60730/UL1998/IEC60335 product functional safety deliverables

IEC60730 Safety Documentation	Contents	Status	Release
C2000 MCU Safety Manual – Preliminary	Overview of safety considerations in product development and product architecture details for Piccolo and Delfino MCUs	Preliminary	Nov 2012
C2000 MCU Safety Manual - Final	Safety module documentation and system level description for customer implementation	Public	May 2013
V300Beta library and Source	Software libraries for IEC60730 Class B	Beta4 For Piccolo6x/3x/2x	Dec 2012
V300 Software lib User Manual	User Manual for Software libraries IEC60730 Class B	Beta4 For Piccolo6x/3x/2x	Dec 2012

Table 4. C2000 MCU IEC60730/UL1998/IEC60335 product functional safety deliverables (continued)

IEC60730 Safety Documentation	Contents	Status	Release
Control Card Software source for IEC60730 lib evaluation	Software libraries for IEC60730 Class B	Beta4 For Piccolo6x/3x/2x	Dec 2012
V300 library and Source	User Manual and Software libraries for IEC60730 Class B	Beta5 For Piccolo6x/3x/2x/5x/ Delino33x	Jan 2013
V4_IEC60730 STL library and Source	User Manual and Software libraries for IEC60730 Class B	V4_00_00_00 For Piccolo/ Delfino MCUs	May 2013

Appendix A Summary of Recommended Safety Feature Usage

Table 6 provides a summary of the safety concept recommendations for C2000 MCU. Each recommendation is given a unique identifier to aid in requirements management. This is a generic table applicable per MCU module or peripheral. These modules may not be available on all the MCUs. For its availability in the selected MCU, see the device-specific data sheet. For each safety feature or diagnostic, the recommendation is noted in simplified form as follows:

Table 5. Legend

Notation	Category	TI Recommendation	User Choice On Safety Features
M	Mandatory	Safety feature always operational	Cannot be disabled
++	Highly Recommended	Provides high value diagnostics that are difficult to implement by other means	Can be enabled or disabled
+	Recommended	Provides high value diagnostics that may be implemented by other means	Can be enabled or disabled
o	Optional	Provides low value diagnostics that may be implemented by other means	Can be enabled or disabled

Table 6. Summary of Safety Features and Recommendations

Device Partition	Device Partition Module Level	Safety Features or Diagnostic	Feature Recommendation		Possible Diagnostic Measures
			Boot Time	Periodic	
C2000 MCU On-Chip Feature					
Processor					
28x CPU	C28x_CPU	Register check for CPU registers	++	++	Self test using PSA hardware on critical code segments, autocoverage
FPU	C28x_FPU	Register check for CPU registers		++	Self test, autocoverage
VCU	C28x_VCU	Register check for CPU registers		++	Self test, autocoverage
CLA	C28x_CLA	Register check for CPU registers		++	Self test, autocoverage
Memory					
RAM	C28x_RAM	Periodic CRC, read and write operation, March13	++	++	Self test with PSA, CRC and March13 Autocoverage/
Flash	C28x_NVMEM	Periodic CRC, read operation	++	++	Self test with PSA,CRC autocoverage
OTP	C28x_OTP	Periodic CRC, read operation	++	++	Self test with PSA,CRC autocoverage
ROM	C28x_BROM	Periodic CRC, read operation	++		Self test with PSA,CRC autocoverage
Interrupts and DMA					
PIE	C28x_PIE	Check PIE registers and RAM Interrupt Sweep test		++	RAM tests, autocoverage
DMA	C28x_DMA	DMA Registers and CRC check on data transfers		++	Software CRC on data transferred
System Control					
Clock registers, protected ranges	C28x_CLK	Check for register and memory range that are protected		+	Software response on register configuration and error check
Watch DOG	C28x_WD	Check registers and timing check		++	Internal watchdog interrupt and Software response
32-Bit Timer	C28x_TIMx	Timer Timing check		++	Timing check with SFO or other clock references

Table 6. Summary of Safety Features and Recommendations (continued)

Device Partition	Device Partition Module Level	Safety Features or Diagnostic	Feature Recommendation		Possible Diagnostic Measures
			Boot Time	Periodic	
Control and Communication Peripherals					
Communication peripherals: SCI,SPI,I2C,McBSP,USB	C28x_SCIx	Enable Internal Loop back to check data integrity and transfer			Software error response on Internal loop back
EQEP	C28x_EQEPx	Register, QEP timer check			Periodic QEP Timer check
ECAP	C28x_EQEPx	Check for APWM event triggers, and interrupt order, with GPIOs as inputs		++	Internal loop back and timing check
EPWM	C28x_EPWMx	Check for PWM event triggers, and interrupt order with GPIOs as inputs	++	++	Software response to periodic PWM event trace
HRPWM	C28x_HPWM	Check SFO functions to estimate and presence of MEP steps on HRPWM	++	++	Software response to Internal SFO calibration
TRIP Zone	C28x_TRIP	Internal Trip trigger checks		+	Software response to internal TRIP requests
Analog Subsystem					
ADC	C28x_ADC	Check for PWM event triggers to ADC and interrupt order	++	+	Software response to periodic ADC results signature
DAC and CMP	C28x_DAC	Check for PWM event triggers to DAC compare	++		Periodic software response to runtime event trace with PWM module
Comparator	C28x_CMP	Register check, internal trigger limit check		+	Software response to internal TRIP requests
POR and BOR					POR, BOR detect
Oscillator_1	A_OSC	OSC clock accuracy check using HRPWM	++	++	OSC clock fail detect
Oscillator_2		OSC clock accuracy check using Timer2	++	++	Software response to Timer 2
External CLKIN	A_CLKIN	OSC clock accuracy check using HRPWM	++	++	Clockin fail detect
PLL	A_PLL	OSC and PLL clock accuracy check using HRPWM	++		SFO library check
Temp Sensor	A_TEMP	Check for Temp sensor select and conversion		++	Temp sensor compare with in system response
GPIO	A_GPIO	Register check and mux select			Software response on registers and inputs - Internal interrupt on Pins
Custom Software					
Custom IP software function	IP_Custom	Check Code integrity - CRC and code functionality check with test vectors	++	++	Periodic Software response on runtime use of Code integrity and test vector validation

Appendix B IEC60730-Class B/UL1998 Class 1 MCU Safety Compliance Features

Table 7 lists the MCU safety features that are typical to appliance and industrial equipment, and maps the safety features as available for C2000 MCU and Piccolo family.

Table 7. Coverage for MCU Hardware Failure Modes and Software Safety Function

IEC60730/UL 998/IEC60335	MCU Component	Fault and Error	Acceptable Measure
1	**CPU**		
1.1	Registers	Stuck at	Functional and periodic
1.3	Program counter	Stuck at	Functional and periodic
2	Interrupt Handling and execution	No Interrupt	Functional test and time slot monitoring
3	Clock	Wrong Frequency	Frequency monitoring, time slot monitoring
4	**Memory**		
4.1	Non volatile	Single bit fault	Periodic and multiple check sum
4.2	Variable memory	Dynamic cross links	Periodic static memory test
4.3	Addressing	Stuck at	Word protection
5	**Internal data path**		
5.1	Data	Stuck at	Protocol test
5.2	Addressing		
6	**External Communication**		
6.1	Data	Hamming distance	Word protection, CRC or protocol test
6.2	Addressing	Wrong address and multiple addressing	Word protection, CRC or protocol test
6.3	Timing	Wrong point in time and sequence	Time slot monitoring
7	**Input/Output**		
7.1	Digital IO	Incorrect value	Plausibility check
7.2.1	Analog IO	Incorrect value	Plausibility check
7.2.2	Analog mux	Wrong addressing	Plausibility check
9	**Custom Logic**		
9.1 (1)	Custom Hardware	Static and dynamic limits	POST and periodic tests
9.2 (1)	Custom Software	SStatic and dynamic limits	POST and periodic tests

(1) Item 9.1 and 9.2 are C2000 MCU-specific implementation and recommendations added to this table.

Appendix C C2000 MCU IEC60730 Software Safety Development Process

The C2000 MCU IEC60730 self-test library functions are written in C28x CPU assembly and C. Assembly functions are intended to do self test more efficiently and leverage architecture features. All these functions are user friendly calling functions that can be included in the end application. The following sections provide the highlights of coding rules conventions used in C2000 MCU self-test software libraries.

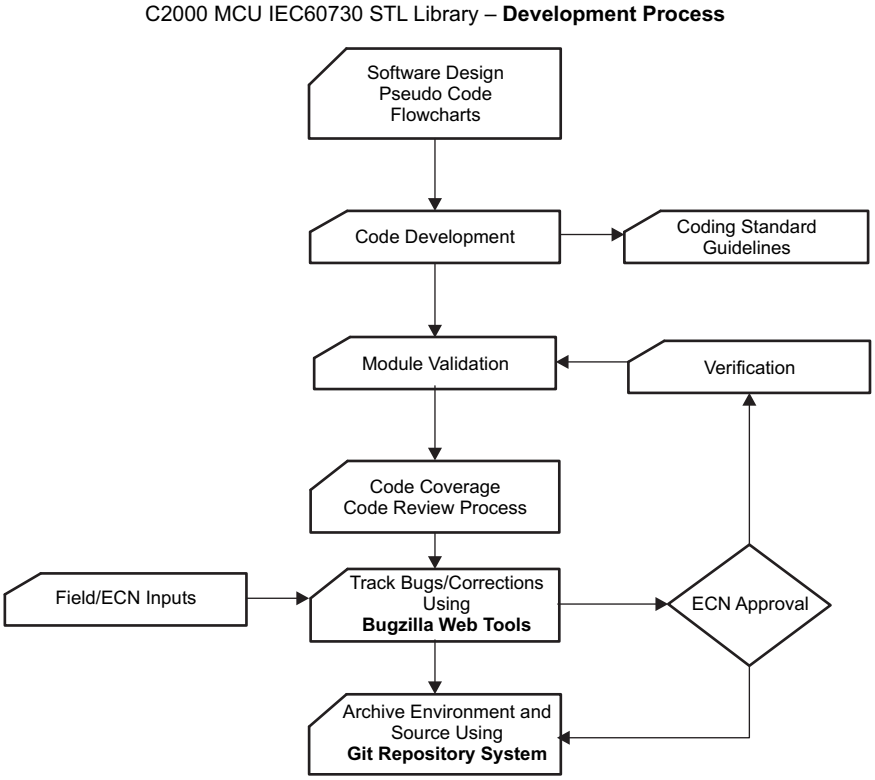

Figure 10. C2000 STL Library Development Process

C.1 Software Design

Design of each module is documented through the use of a flowchart or pseudo code. The choice depends on the module and which best represents the functionality.

C.2 Coding Standard and Naming Conventions

The coding standards and naming conventions are outlined below. The standard is enforced by code review. LDRA programming standard checking will also be applied.

C.2.1 Variable Names

- All variables are camel case starting with a lowercase letter. Global variables must begin with the letter ‘g’.
  - uint16_t loopCounter; uint16_t gDataBuffer
- Type definitions must begin with their module name in all capital letters, followed by an underscore (‘_’).
  - MODULE_FruitTypes
- Defined constants must be in all caps and start with the module name.
  - #define MODULE_REGISTER_BIT (1 << 4)

C.2.2 Function Names

- All functions must begin with their module name in all capital letters, followed by an underscore ('_').
- Functions names must be camel case starting with a lowercase letter and should start with a verb describing the action of the function
  - UART_putChar(); PWM_setPeriod(); TIMER_getValue();

C.2.3 Comments

- All comments must use the single line comment delimiter "//" and must be indented to the same level of the code they document.

C.2.4 Function Prototypes

- All functions must be prototyped in a module's corresponding header file.
- Prototypes must include Doxygen comments. For more details, see the header files in the software source in the IEC60730 STL library.

C.2.5 Indentation

- Each indent level is four spaces and all white space only contains spaces (no tabs)
- Continued lines must be indented eight spaces

C.2.6 Bracketed Statements

Bracketed statements must be in one of the following forms:

```
if (expr)
{
   stmt;
}
else if (expr) {
   stmt;
}
else {
   stmt;
}
```

```
do
{
   stmt;
}
```

```
for(expr; expr; expr)
{
   stmt;
}
```

```
switch (expr)
{
   case CONST:
      break;
   case CONST:
      break;
   default:
      break:
}
```

C.2.7 File Structure

- All software in a file must reside in one of the following sections. The sections must appear in the following order within a file:
  - - header, includes, defines, typedefs, globals, function prototypes
  - For more details, see the header files in the software source in the IEC60730 STL library.

C.2.8 Data Type

Data Types: The following C99 data type must be used:

- _Bool for Boolean types
- (u)int_leastX_t (portable)
- (u)intX_t (architecture specific), where X is 8, 16, 32 and 64

C.2.9 Macro Definitions

- Only single line macros are allowed. Inline functions must be used to optimize functions.

C.2.10 Static Code Coverage

- MISRA and LDRA rule checker is being used to static analysis tools

C.2.11 Code Review Process

- Face-to-face code review. Most issues are resolved in real time with lingering issues logged in the Bugzilla system for later resolution.
- LDRA toolset will be used to track peer review comments.

C.2.12 Archive and ECN Process

- Archive is through a centralized .git repository system. Each release is done on its own branch and an appropriate label applied upon release.
- Bugs and issues are reported through the Bugzilla web-based tracking system.

C.3 MISRA C Exceptions for C2000 MCU C28x Architecture

1	Mandatory rules that are violated			
		a.	Use of single line comment(s)//	MISRA-C:2004 2.2
		b.	Pointer arithmetic on non-array entities	MISRA-C:2004 17.1,17.4
		c.	#pragma used,	MISRA-C:2004 3.4
2	Suggested and advisory rules that are violated			
		a.	Basic type declaration used	MISRA-C:2004 6.3 (use of the term ‘interrupt’)
		b.	Casting operation on a pointer	MISRA-C:11.1,11.4
		c.	Expression is not Boolean	MISRA-C:2004 12.6,13.2
		d.	Casting from integer to pointer	MISRA-C:2004 11.3

Appendix D C2000 Compiler and Tools Development Process and Tracking

Software code development and firmware debug is made easy using the Code Composer Studio™ tool set from Texas Instruments. Details of the tools and the environment are listed in Table 8. These tools are developed with TS16949 product development flow. These are under evaluation and IEC61508 compliance tests, primarily driven by the Texas Instruments Software Development Systems group (SDS). Details of the compiler, IDE and emulation tool releases are controlled and archived per TI quality standards. The following external wiki link is a useful reference for customer to generate quality related document for C2000 products.

Table 8. TI Wiki Links for Software Tools

Wiki Links for on Software Development and Status	Description
http://processors.wiki.ti.com/index.php/Category:CCS	Code Composer IDE updates
http://processors.wiki.ti.com/index.php/Category:Compiler	Code Generation tools update
http://processors.wiki.ti.com/index.php/XDS100	Emulation tools update

Appendix E STL Test Suite Release Process

Design and test release of IEC60730 - STL test suite was developed with the elements of IEC60730 V-model in mind. This method allows design, early validation, regression tests and customization. They are living working examples to explain code functionality and enables easy in-system validation.

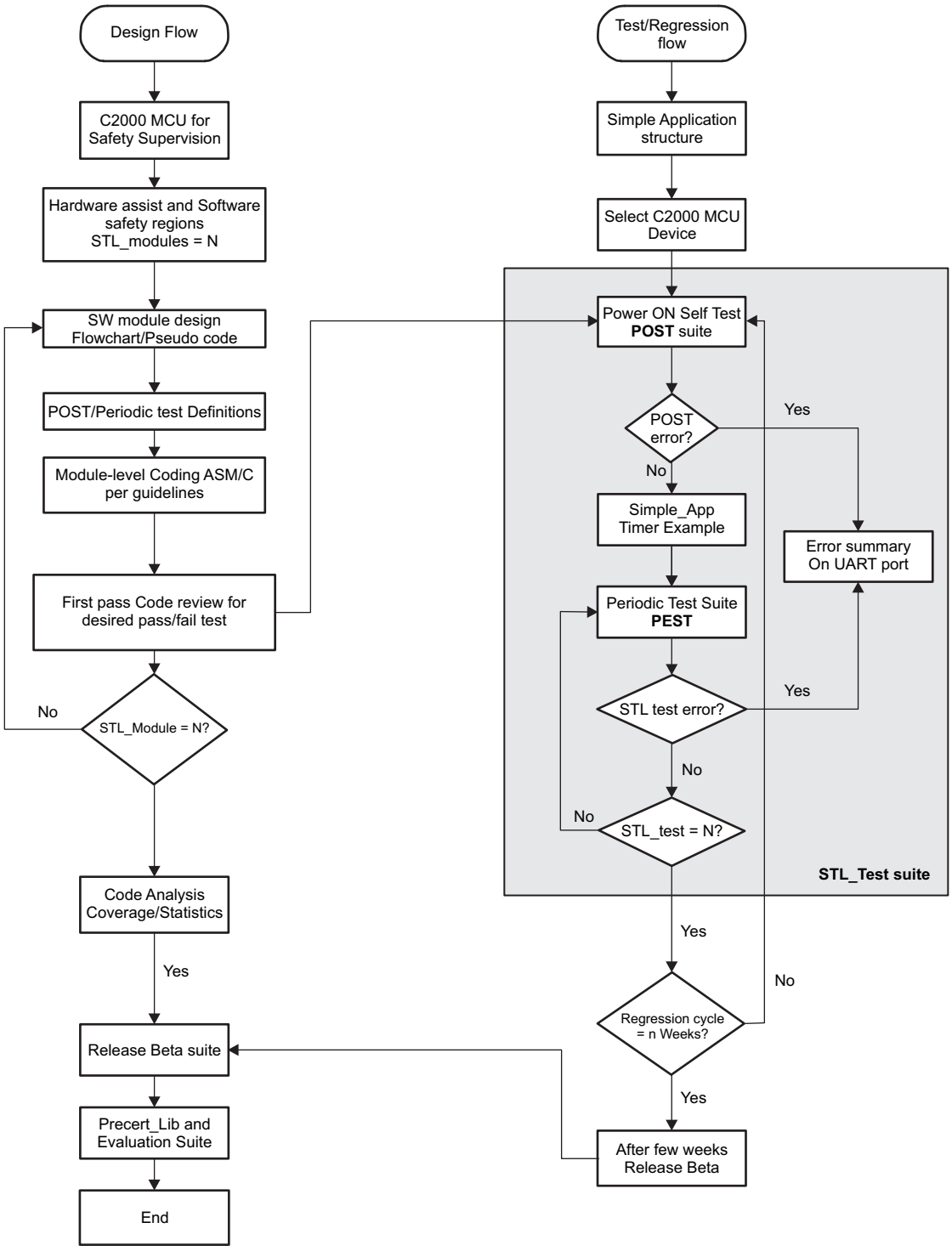

Figure 11. C2000 IEC60730 STL Library – Design, Test and Regression Flow - DTR

NOTE: POST and PEST are acronyms for a set of software functions that will be executed at power on of the MCU and periodically during the application.

Appendix F Typical Application Firmware With IEC60730 Safety Supervisory Functions

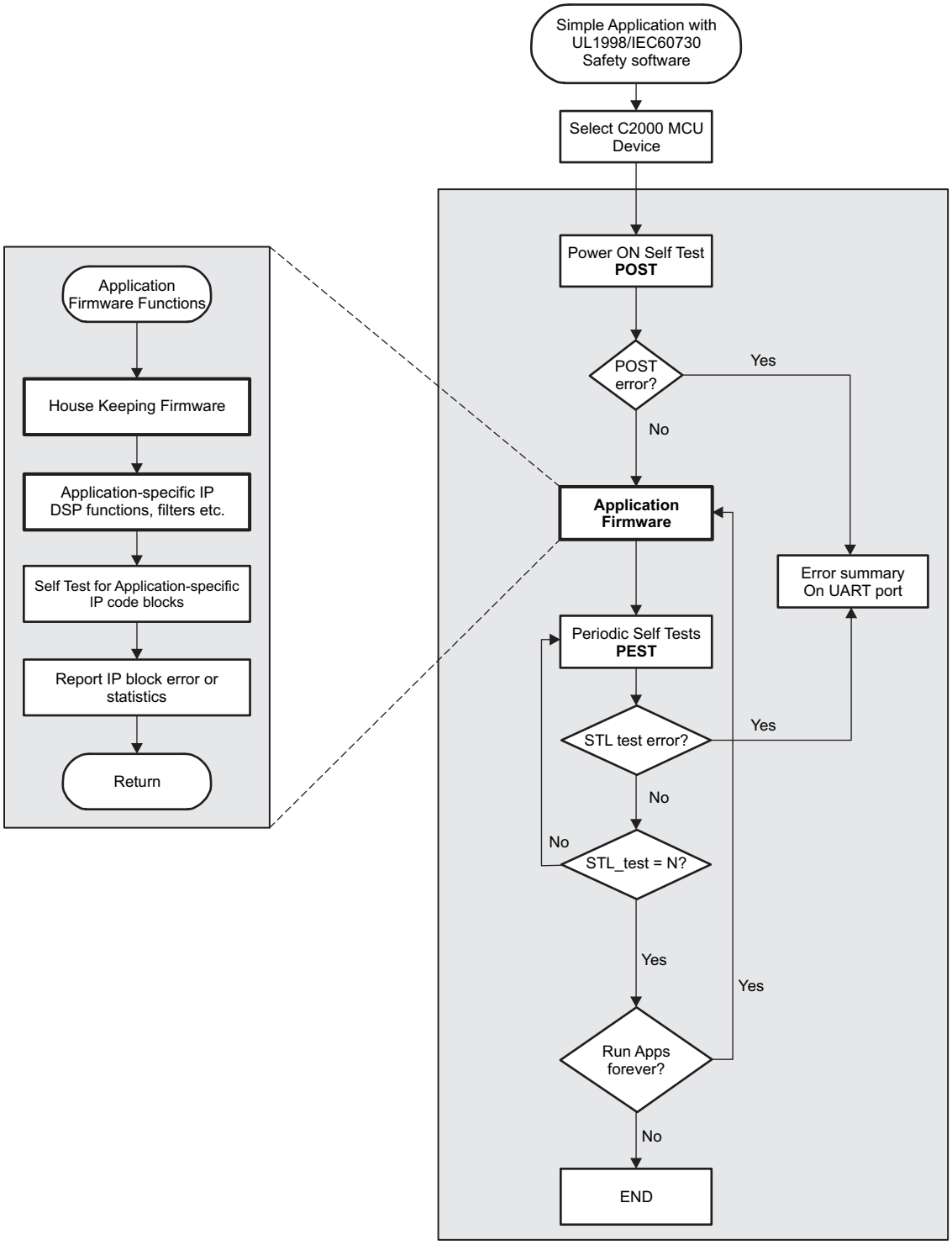

Figure 12. Typical Application Firmware Components With C2000 IEC60730 STL (Self Test Libraries) – POST and PEST Functions

Appendix G Glossary

Table terms and definition for ready reference are listed in Table 9.

Table 9. Table Terms and Definitions

Terminology and Abbreviations	Definition and Full Expression
CCF	Common Cause Failure
Class B and Class 1	Class B and Class1 are IEC60730 and UL1998 classification of safety standards for appliance space using MCU (microelectronics controller)
Class C and Class 2	Class C and Class2 are IEC60730 and UL1998 classification of safety standards for Industrial equipment that prevent hazards, such as fire and explosion using MCU (microelectronics controller)
E/E/PE	Electrical/Electronic/Programmable Electronic, example: E/E/PE
EEPROM	Electrically Erasable Programmable Read-Only Memory
EUC	Equipment Under Control
FIT	Failure In Time (1 FIT = 1 failure 109 h)
FMEA	Failure Mode, Effects and Analysis
FMEDA	Failure Mode, Effects and Diagnostics Analysis
FPGA	Field Programmable Gate Array
Functional Safety	Part of the overall safety relating to the EUC and the EUC control system that depends on the correct functioning of the E/E/PE safety-related systems and other risk reduction measures
HFT	Hardware Fault Tolerance
IEC	International Electro technical Commission – Source for IEC Specifications
ISO	International Organization for Standardization – Source for ISO specifications
MooN	M out of N channel architecture with Diagnostics
MooND	M out of N Channel Architecture With Diagnostics
MTBF	Mean Time Between Failure
Safe State	State of the EUC When Safety is Achieved
Safety	Freedom From Unacceptable Risk
SFF	Safe Failure Fraction
SIF	Safety Instrumented Function
SIL	Safety Integrity Level
TI	Texas Instruments
C2000 MCU Specific	
C28x CPU	Texas Instruments 32-Bit Central processing Unit
CLA	Texas Instruments 32-Bit processor to Function as Control Law Accelerator
FPU	Floating-Point Unit designed to work with C28x CPUs
IQ Math	Floating-Point Unit designed to work with C28x CPUs
PSA	Parallel Signature Analyzer
VCU	Viterbi and Complex math unit designed to work with C28x CPUs
POST	Power On Self Test
PEST	Periodic Self Test

Appendix H Revision History

This document has been revised from SPRUHI3 to SPRUHI3A because of the following technical change(s).

Table 10. SPRUHI3A Revisions

Location	Additions, Deletes, and Edits
Section 5.2.5	Added new section.
Section 5.2.6	Added new section.
Section 5.2.7	Added new section.

IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

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

TI objects to and rejects any additional or different terms you may have proposed.

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