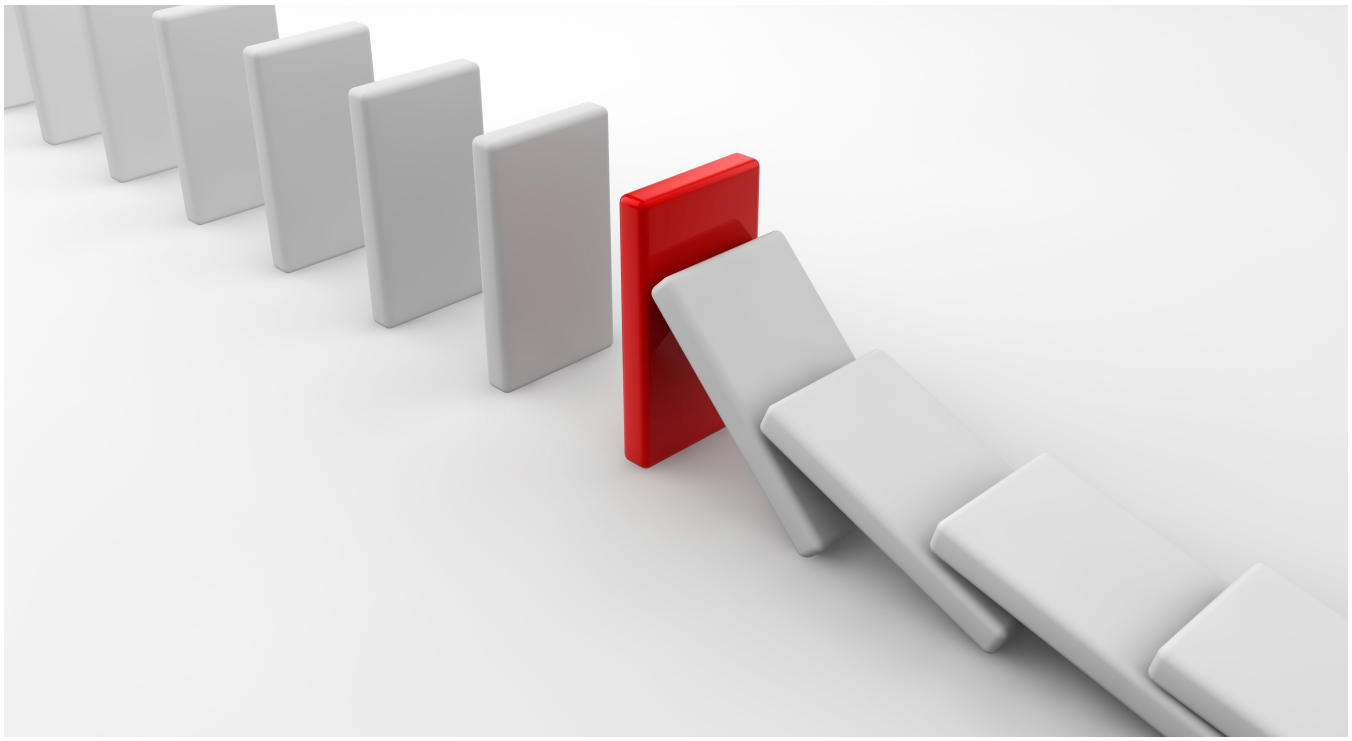
White Paper
SWRY027–June 2018

Achieving Coexistence of Safety Functions for EV/HEV Using C2000™ MCUs

Ashish Vanjari



As the market for electric vehicles and hybrid electric vehicles (EV/HEV) increases, there is rapid increase in the need for electrification of vehicle powertrains. The C2000™ MCU enables precise and accurate motor control for the electric powertrain, with high performance and integrated functional safety.

Contents

List of Figures

List of Tables

Trademarks

1 Introduction

As the market for electric vehicles and hybrid electric vehicles (EV/HEV) increases, there is rapid increase in the need for electrification of vehicle powertrains. Enabling precise and accurate motor control for the electric powertrain, with high performance and integrated functional safety, is of paramount importance.

Systems in EVs include a traction system, battery management system, and a DC/DC converter. There can be from one to four traction motors per each vehicle, assuming that every wheel has a motor of its own. Each motor requires the appropriate system to drive and control these motors safely and efficiently.

The torque on the wheel is controlled by regulating the current applied to the traction motor based on the torque command. Accurate and safe torque control for the traction motor in electric vehicles is critical to both consumers and auto manufacturers.

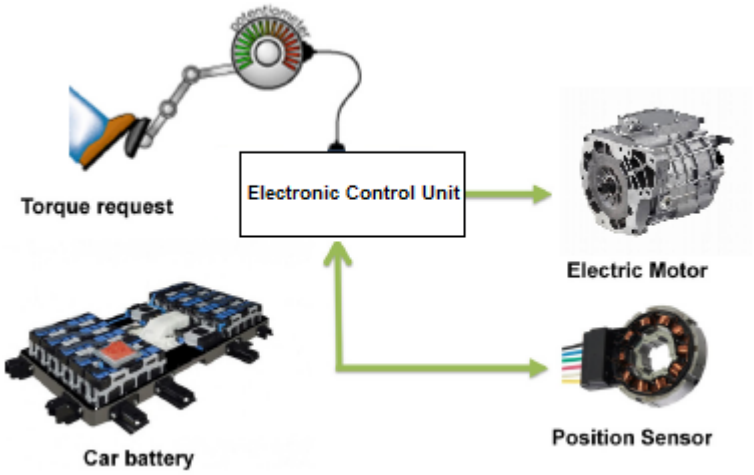


Figure 1. Electric Vehicle Traction Control System

Sophisticated digital control of the motors ensures high efficiency, stability, and comfort for the passengers. At the same time, there is a risk due to the possibility of an electronic malfunction. Safety is a key issue for electric automobile development, and must be handled as a part of overall system design in electric vehicles. The ISO 26262 functional safety standard provides guidelines for requirements and processes to achieve overall system safety within road vehicles. It uses an automotive-specific, risk-based approach to determine integrity levels, such as Automotive Safety Integrity Levels (ASIL), and specifies requirements to avoid unreasonable residual risk.

As an initial step, the system integrator performs a Hazard Analysis and Risk Assessment (HARA) according to ISO 26262-3 while considering various driving conditions. The goal is to identify hazardous events that must be addressed. A safety goal to prevent, mitigate, or control each hazardous event is defined. Each safety goal is characterized by a definition of a safety function (what it should do?), safe state (how should it react?), FTTI (within fault tolerant time interval), and at the ASIL level, which defines the rigor required in risk reduction while implementing the safety function. The same ECU can be assigned to perform many safety functions, each one with different ASIL targets. Then, the MCU responsible for implementing the safety function is allocated the safety requirements.

For example, HARA performed for an electric motor drive in a traction inverter responsible for providing torque to the wheels can result into identification of multiple safety goals, as shown in Table 1.

Table 1. Example Safety Goals

Sl No:	Hazard	Safety Goal	MCU Safe State	ASIL	FTTI
EVTR_SG1	Too low torque – vehicle passing	Avoid unintended low torque	(i) ERROR reported	C	10 ms
EVTR_SG2	Too high torque – city driving, approaching a STOP sign, or pedestrians	Avoid unintended high torque	(i) PWM Tristate (ii) ERROR reported	D	10 ms
EVTR_SG3	Torque in wrong direction – reversing car at a parking lot	Avoid torque in wrong direction	(i) PWM Tristate (ii) ERROR reported	B	10 ms
EVTR_SG4	Unintended torque generation – vehicle stopped at a traffic light	Avoid unintended driving of E-motor by MCU	(i) PWM Tristate (ii) ERROR reported	D	10 ms
EVTR_SG5	Operating beyond permissible temperature or in presence of an insulation fault	Avoid operation beyond permissible temperature range	(i) PWM Tristate (ii) ERROR reported	A	10 ms

In addition to safety functions, the MCU may be required to implement additional functions which do not have any functional safety requirements, such as data logging.

As shown in Table 1, the system integrator can allocate more than one safety function to the MCU, and those safety functions could be with different safety integrity levels (ASILs). Because multiple safety functions are implemented on same MCU, they may share resources such as memory, peripherals, on-chip interconnect, compute logic, and critical registers for device settings. Risk exists so that a fault in a safety function with a lower-assigned ASIL cascades into another safety function with a higher ASIL. This risk must be mitigated. ISO 26262 provides guidelines for implementing safety functions with different integrity levels.

2 ISO 26262 and the Coexistence of Safety Functions

ISO 26262-9:Clause 6 has guidelines on the criteria for coexistence of elements. It applies to sub-elements in safety-related functions, with either no ASIL or a lower ASIL.

- Interference is the presence of cascading failures from a sub-element with no ASIL assigned, or a lower ASIL, to a sub-element with a higher ASIL, leading to the violation of a safety requirement of the element.
- The possibility of interference requires the system integrator to identify potential threats to the safety concept, due to cascading faults, by either preventing the occurrence of the threats or revising safety requirements assigned to the sub-elements with the lower ASIL.
- The targeted elements of interference on a typical MCU are on-chip memories, peripherals, critical device configuration registers, and computing logic.

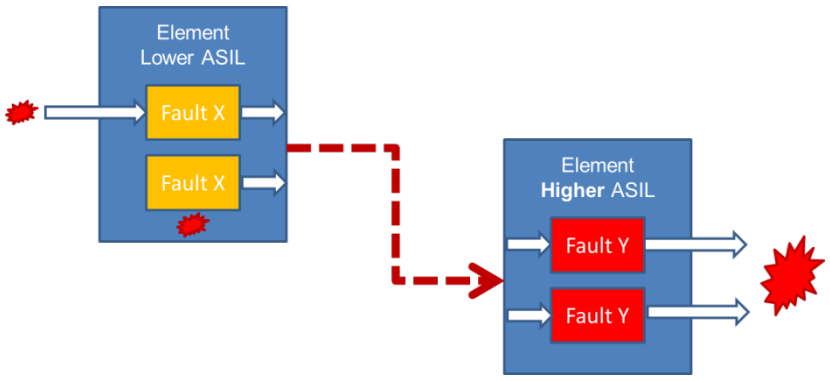


Figure 2. Example of Cascading Failure Causing Interference in Higher ASIL Safety Function

To ensure the coexistence of safety functions on an element, the system integrator is required to prove absence of interference: Coexistence = ! (Interference).

Prevention of interference is achieved either at the source – by preventing the generation of interference at an originating sub-element – or at the destination, by protecting the target sub-element vulnerable to the generated interference. The C2000 device has the necessary mix of central and distributed device features to help avoid interference.

3 C2000 Device Features

This section describes the features supported on C2000 devices that either detect or prevent the interferences from safety functions implemented with lower ASIL, thus achieving freedom from interference in the safety functions implemented on the device.

C2000 devices are a mix of dedicated and shared SRAMs. Shared SRAMs are configurable to achieve control for write, read, and fetch access from different masters, such as the CPU, CLA, and DMA. This gives the system integrator the flexibility to resize the allocation of memory to each master, based on the use case. However, this also introduces the possible risk of interference.

- Embedded Real Time Analysis and Diagnostic (ERAD): The ERAD module provides system analysis capabilities that can be used to detect faults in the CPU and other logic on the MCU by configuring the bus comparator units that monitor CPU bus accesses, and counter units that count events. This module, accessible by the application software, consists of the enhanced bus comparator units and benchmark system event counter units. The enhanced bus comparator units are used to monitor various CPU buses and generate events. The activity monitored and detected by these units can be used to generate breakpoints, watch-points or an interrupt (RTOSINT). After the application code sets up the ERAD module, the module can work independently to generate a RTOSINT interrupt when an event match occurs. This module can be used to detect the presence of interference originating from the lower ASIL software implemented on the CPU by continuously monitoring its buses.
- Dual Zone Code Security (DCSM): The DCSM module is a security feature incorporated on C2000 devices. It blocks access and visibility of on-chip secure memories from unauthorized users, to combat duplication and reverse engineering of proprietary code. Each CPU subsystem has its own dual zone CSM for code protection. DCSM can be used for functional safety where functions with different safety integrity levels can be executed from different security zones (zone1, zone2, and unsecured zone), acting as firewalls and thus mitigating the risk due to interference from one secure zone to another.

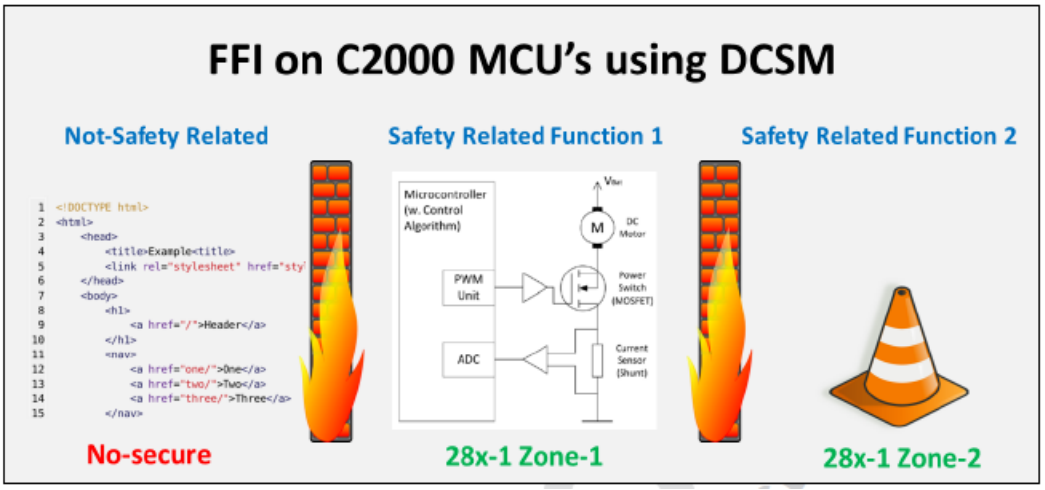


Figure 3. Using DCSM to Prevent Interference

- Memory access protection schemes: One of the unauthorized masters (CPU,CLA, DMA) can accidentally access or corrupt the memory locations containing critical variables used by a safety function with a higher integrity level. Memory access protection logic is implemented per instance of local and global shared SRAMs to detect access violations of WRITE, READ, or FETCH. The respective flag bit is set and the interrupt is generated to the CPU, thus alerting of possible interference and helping to achieve a safe state for the device. The access violation information is captured in the register to allow the software to take further corrective action.

- Flash memory: Flash memory is a secured resource, and each sector can be allocated to a particular secure zone. Each zone has its own CSM passwords: read and write accesses are not allowed to resources assigned to Zone 1 by code running from memory allocated to Zone 2, and vice versa. Before programming or erasing any secure flash sector from unsecured memory or another zone's memory, the user must unlock the flash sector's zone. One flash pump is shared for erase/program operations on flash memory. A semaphore mechanism is provided to avoid the conflict between Zone1 and Zone2. A zone must grab this semaphore to successfully complete the erase/program operation on the flash sectors allocated to that zone. Any accidental attempt, by erroneous or faulty software code implemented with a lower integrity level, to program or erase the flash sector is blocked, thus preventing the interference.
- Critical configuration registers protection: Integrity of the safety functions depends on the MCU's critical configuration registers that manage Power, Clock, Reset, and so forth. If the critical registers are corrupted by faulty software that coexists on the same hardware and is implemented with a lower ASIL, the safety functions can be compromised. This interference must be either prevented or detected. C2000 devices have critical registers designed with the EALLOW protection mechanism. This uses special CPU/CLA instructions EALLOW/MEALLOW and EDIS to enable and disable access to protected registers. This register protection is enabled by default at startup. While protected, all writes to protected registers by the CPU are ignored. Thus, software with lower integrity cannot corrupt the critical configuration of the device. The integrity of critical control registers, such as clock source selection, PLL multiplication, pre-scalar and post-divider, are essential to operate the device at the correct speed. Corruption of these registers due to faulty software can drastically affect the performance and safety of the overall system. On the C2000 device, many of these registers can be programmed and locked by software by configuring LOCK field, to prevent any further programming of the configuration until system reset. Also, writes to some critical control registers are protected by a specific KEY field embedded in the register definition, which enables or disables writes to it. Thus, software interference from a lower integrity level affecting the safety function can be mitigated.
- Peripherals: Peripherals on a device are also shared resources. These are responsible for critical functions such as control and communication peripherals for a safety function. C2000 devices provide access to these peripherals by multiple masters, orthogonally between CPUs, CLAs, and DMAs. There is a risk of interference from a master implementing a lower ASIL function by corrupt accesses to the peripherals. Control peripherals, ADC, EPWM, SDFM, Comparator subsystem, DAC, PGA, and Communication peripherals CAN, SPI, LIN, PMBUS, and FSI are protected by master access control logic in each instance. When programmed, this feature completely blocks accesses from certain masters. This reduces or eliminates the possibility of interference from a lower ASIL safety function.

4 Summary

Driven by increasing demand for electrification in the automotive market, EV and HEV applications require multiple safety functions to be implemented on the same hardware element. Each of the implemented safety functions may have different allocated safety integrity levels (ASIL). All these must coexist on the same hardware element or MCU. To ensure the integrity of the safety functions with higher ASIL requirements, the system integrator must prevent or detect interference from lower ASIL functions. ISO 26262 provides guidelines and requirements on this topic. Texas Instruments C2000 MCUs implement the right mix of centralized and distributed hardware safety mechanisms that can help either detect or prevent interference.

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