Application Report
SPRACN0–August 2019

C2000™ Key Technology Guide

Matthew Pate

ABSTRACT

This application report is intended to provide a deeper look into the components that differentiate the C2000 Microcontroller Unit (MCU) as it pertains to Real-Time Control Systems. The focus is on features that Texas Instruments (TI) believes have measurable impact to overall system performance as well as increased bandwidth to the MCU. This document also expands on the specific aspects of the system that are improved with the covered topics. Finally, appropriate references and links to discrete Part Numbers (PNs), reference designs, and demonstration kits for each topic are provided.

Contents

List of Figures

List of Tables

Trademarks

Bosch is a trademark of Robert Bosch GmbH, Germany.
C2000 is a trademark of Texas Instruments.

1 C2000™ MCUs and Real-Time Control

A real-time control system is one that has time dependence to when it needs input to remain stable. This is counter to a batch processing type system in which data is manipulated by a processor and reported, but there is no critical system waiting for the data. Real-time control processors can perform batch type processing, but it is not typically their main use in a system. A simple visualization of a type of real-time control system is given in Figure 1. Thus far, only processing has been mentioned, but a fully realized controller in this case would be capable of measuring key parameters of the system (sensing), applying control algorithms to the incoming data (processing), and then affecting the system to achieve the desired change given by the control algorithm (actuation).

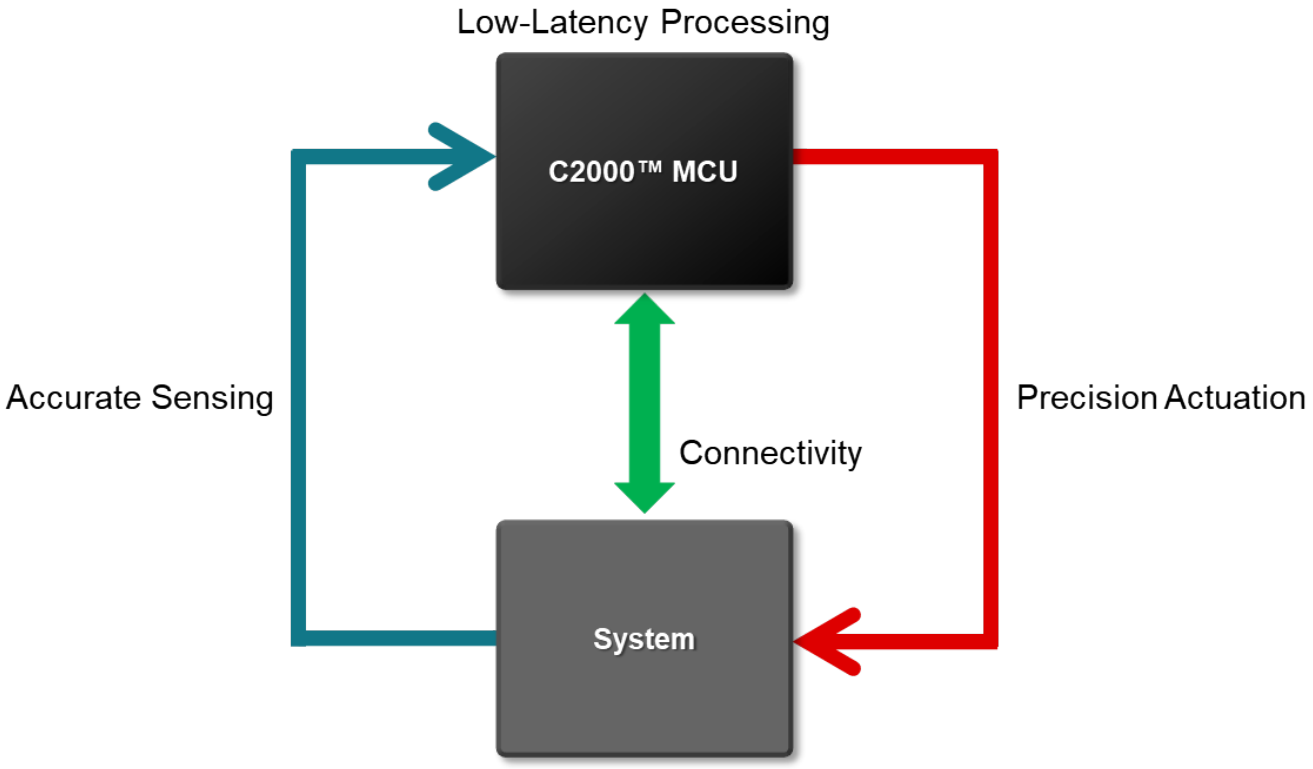

Figure 1. Simple Real-Time Control System

The amount of time that the system can be left alone (sometimes referred to as open loop) determines the speed at which the controller must perform all of the above functions. An important distinction here is that the raw speed of the processing block is not the only care about, but rather the MCU ecosystem as a whole contributes to the time it will take to react to different system events. Now that you have a general understanding of a real-time control system, take a closer look at how the C2000 MCU is constructed to service these needs.

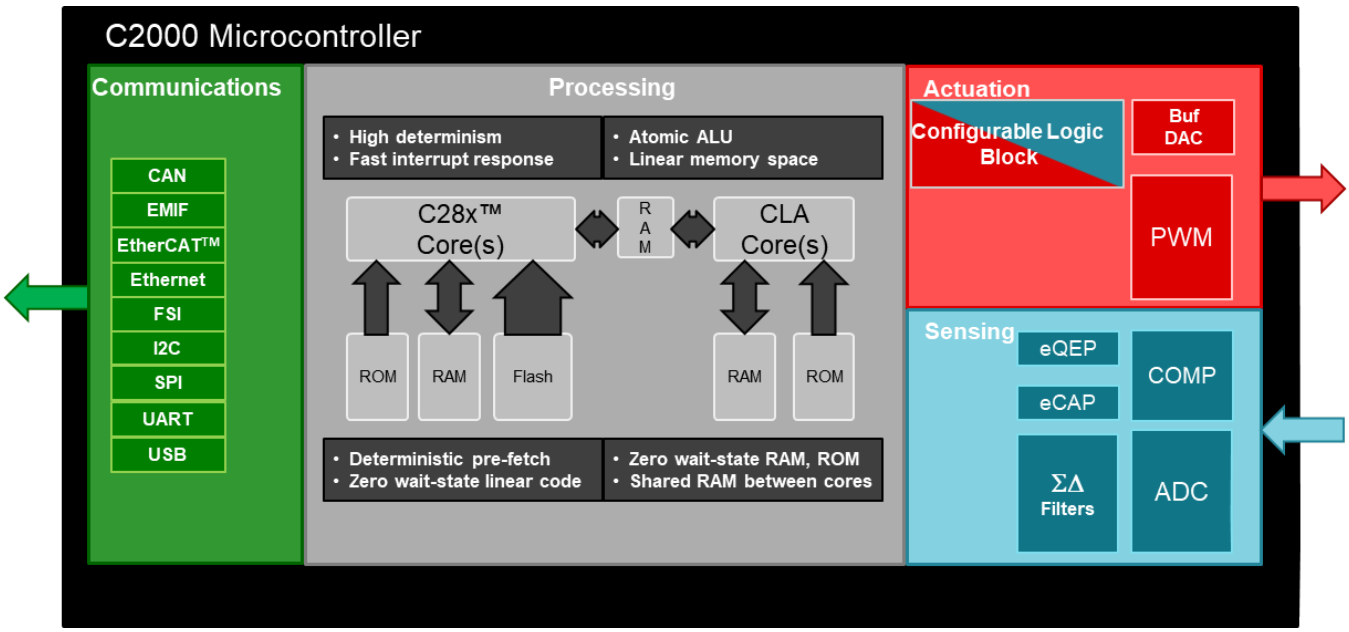

Figure 2. C2000 MCU Components

1.1 Processing

As seen in Figure 2, the C2000 MCU uses the C28x core as the main processing unit. This is a 32-bit floating point (single precision) core with dedicated instructions tailored to real-time control applications. Complementing the C28x core is a Control Law Accelerator (CLA) a 32-bit floating point co-processor capable of independent code execution increasing the system bandwidth versus a C28x core alone. There are both dual and single core implementations across the C2000 MCU family of devices.

1.2 Actuation

The actuation sub-system includes modules that will stimulate the system under control. Typically, this is done with a Pulse Width Modulation (PWM) output. This could also be the output of the on-chip Digital-to-Analog Converter (DAC) or just a General-Purpose Input/Output (GPIO) pin.

- PWM – Principle actuation module on the C2000 MCU. Responsible for driving the external Field Effect Transistors (FETs) that exist in most power electronics systems. Supports both standard and high resolution modes.
- Buffered DAC – 12-bit DAC capable of driving a defined external load. Typically used to create a bias voltage for the analog domain.
- Configurable Logic Block – Group of look up tables and state machine logic that operates on internal signal nodes in the hardware domain. Can be an endpoint or intermediary step to realize increased system performance.

1.3 Sensing

This sensing sub-system includes modules that translate the state of the external system under control (analog domain) into data usable by the MCU (digital domain). Often this is the work of the Analog-to-Digital Converters (ADCs) on the MCU, but could also be handled by comparators or demodulators for external ADCs. Other unit converters are included in this domain, such as quadrature encoders and time pulse measurement devices.

- ADC – Multiple 12 or 16-bit ADCs that are used primarily to convert the voltage or current (through a shunt) of the controlled system into the digital domain. Both an internal reference or external references are supported.
- Comparator (COMP) – Multiple on-chip comparators provide system protection as well as cycle by cycle PWM control by comparing a system voltage to an internal reference point(generated by the internal 12-bit DACs). Direct connection to the ePWM modules exist to change the output state as quickly as possible without need for CPU intervention.
- Sigma Delta Demodulator – On-chip logic used to decode the serial bit stream output from external sigma delta ADCs. Includes hardware threshold detection.
- Quadrature Encoder Pulse Measurement (eQEP) – Counts pulses from a variety of encoders to determine motor shaft position.
- Time Capture (eCAP) – Measures the time duration between external pulse events, useful for evaluating Hall Sensors. Both standard and high resolution modes are available.

1.4 Communications

An additional block that, while not essential in the real-time control of the system, is almost always needed from a system integration point of view. From serial data streams to multi-channel inputs, as well as an industry standard options as well as proprietary formats, the communications sub-system supports a wide array of communications options.

- Controller Area Network (CAN) – The CAN module supports the Bosch™ CAN protocol standard.
- External Memory InterFace (EMIF) – Parallel data bus typically used to support connections to SDRAM as well as wide bus peripherals.
- EtherCAT Slave Controller (EtherCAT) – This module allows for the C2000 MCU to act as a slave node in an EtherCAT network.
- Ethernet – 10/100Mbps Ethernet controller and physical interface for external communications across this bus.
- Fast Serial Interface (FSI) – 2 or 3 line simplex serial data transmit or receive. Designed to meet both the high speed (100Mbps) as well as the variable latency introduced when crossing an isolation boundary.
- Inter-Integrated Circuit (I2C) – Interface/controller for an I2C bus
- Serial Peripheral Interface (SPI) – Interface/controller for a standard Serial Peripheral Interface bus.
- Universal Asynchronous Receiver/Transmitter (UART) – Interface/controller for Universal Asynchronous Receiver Transmitter bus
- Universal Serial Bus (USB) – USB 2.0 MAC and PHY used to interface to standard USB network.

NOTE: Peripheral counts, as well as features, may vary from device to device. For a complete listing of the number of a peripherals on a device, see the data sheets referenced at the end of each section. For the feature sets supported on a given device, see the *C2000 Real-Time Control Peripherals Reference Guide*

2 Processing Key Technologies

2.1 Accelerated Trigonometric Math Functions

2.1.1 Value Proposition

Trigonometric functions are used heavily in real-time control systems, both in power applications as well as motor control. Park Transforms (Figure 3), Space Vector Generation, and resolver angle are a few of these examples that rely on trigonometric math. The Trigonometric Math Unit (TMU) on C2000 MCUs enables an extended instruction set targeted at 32-bit floating-point trigonometry based calculations.

$$\text{Park} \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} X \begin{bmatrix} i_\alpha \\ i_\beta \\ i_o \end{bmatrix}$$

Figure 3. Park Transform

2.1.2 In Depth

Many common mathematical techniques in real-time control rely on the use of trigonometric functions: sine, cosine, and arc tangent are all examples. The TMU adds dedicated instructions to the C28x core for these functions as well as their inverse, that supersede the standard C library calls. As shown in Figure 4, an 85% cycle count improvement for the Park Transform is seen when using the TMU instructions vs the native C28x floating point compiled instruction set.

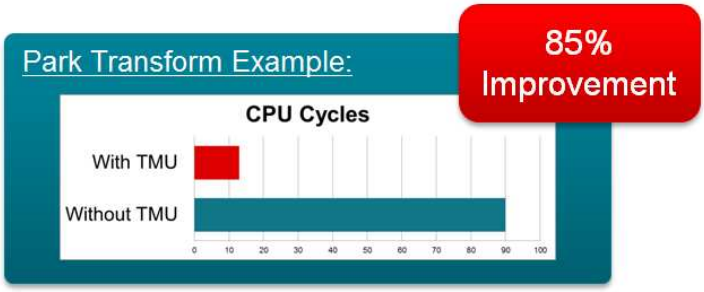

Figure 4. TMU improvement for Park Transform

There is also single instruction support for both square root as well as floating point division. These are often used in conjunction with the trigonometric functions previously listed. A full list of supported instructions and their cycle counts can be seen in Table 1. These instructions are inserted automatically by the C compiler on devices that have a TMU.

Table 1. TMU Supported Instructions Summary

Operation	C Equivalent Operation	C28x Pipeline Cycles
Multiply by 2*pi	a = b * 2pi	2 cycles + Sine/Cosine function
Divide by 2*pi	a = b / 2pi	2 cycles + Sine/Cosine function
Divide	a = b / c	5 cycles
Square Root	a = sqrt(b)	5 cycles
Sin Per Unit	a = sin(b*2pi)	4 cycles
Cos Per Unit	a = cos(b*2pi)	4 cycles
Arc Tangent Per Unit	a = atan(b)/2pi	4 cycles
Arc Tangent 2 and Quadrant Operation	Operation to assist in calculating ATANPU2	5 cycles

NOTE: While the C2000 MCUs listed below all have a TMU module, the C compiler used to generate the target code must have the correct options selected to utilize this HW. This is controlled in the Processor Options of the C2000 Compiler via the TMU support drop down, as well as using the "relaxed" setting for the floating point mode under C2000 Compiler → Optimizations. Specific TMU based functions can be called explicitly as inline functions in the C source if that is preferable to a global setting.

2.1.3 Device List

- TMS320F28378xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

2.1.4 Hardware/Software Resources

- TIDM-1007 Interleaved CCM Totem Pole Bridgeless Power Factor Correction (PFC) Reference Design
- TIDM-HV-1PH-DCAC Single-Phase Inverter Reference Design With Voltage Source and Grid Connected Modes
- TMDXIDDK379D C2000 DesignDRIVE Development Kit for Industrial Motor Control
- TMDSHVMTRINSPIN High Voltage Motor Control Kit with InstaSPIN-FOC and InstaSPIN-MOTION enabled Piccolo MCU for F280049C device lab7 and lab8

2.1.5 Documentation

- *Accelerators: Enhancing the Capabilities of the C2000™ MCU Family*
- *TMS320C28x Extended Instruction Sets Technical Reference Manual*

2.2 Flexible System Interconnect

2.2.1 Value Proposition

Three on-chip signal crossbars (X-bars): Input, Output, and ePWM provide the necessary mechanism in hardware to efficiently connect multiple subsystems across different control system implementations. Lower system latency, simpler PCB routing, and consistent timing are all key benefits.

2.2.2 In Depth

The inter-dependence of the Sensing and Actuation subsystems in a real-time control MCU is an obvious one. Whether it is incoming signals into the device, signals generated by on-chip logic (comparator, SDFM, ADC, and so forth), or outgoing signals; routing these signals in the system can be challenging at best. The on-chip X-bars provide a flexible mechanism to do that in hardware inside the MCU. There are three key benefits that these modules provide to the system:

- Simple routing of external signals into the chip: Any GPIO can go to multiple modules on-chip (Figure 5). For example, this allows the eCAP module to choose from any input pin as its source, or any input pin to go to the CPU/CLA as an external interrupt. This also gives flexibility in signal routing and layout of the Printed Circuit Board (PCB) since pins are not hard configured for a set group of functions.

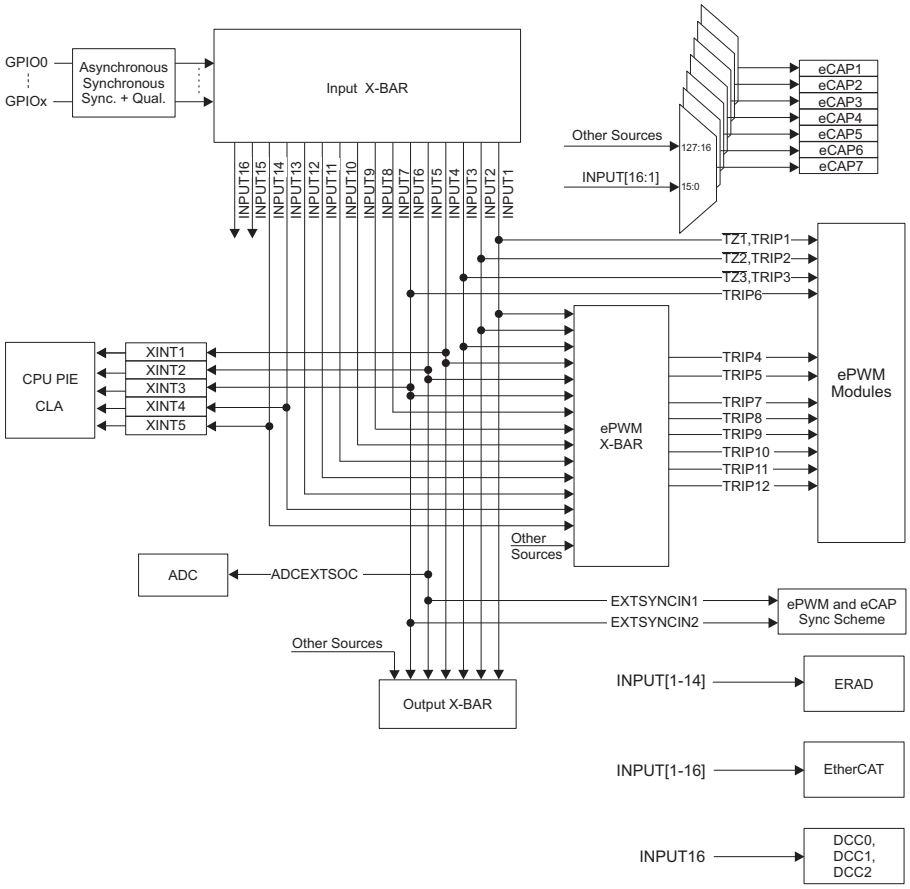

Figure 5. Input X-bar on the TMS320F2837xD MCU

- Lower system latency: For simple point-to-point transactions, there is no need for the main C28x CPU or the CLA to spend cycles routing signals from one domain to another when using the X-bars. Inside of each X-bar, there is also a simple logical OR (Figure 6) that allows combination of any of the inputs to the X-bar in hardware. This not only saves cycles for the processors, but allows flexibility of the system since the mux selections are all controlled in software and can be changed dynamically as needed.

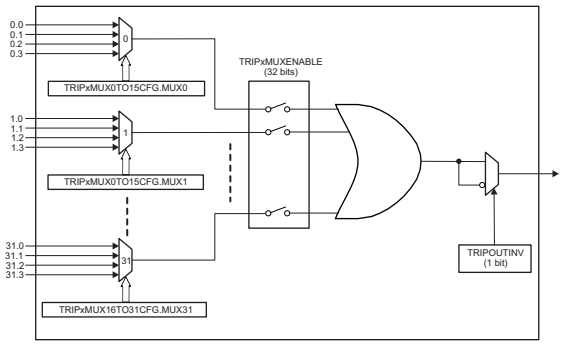

Figure 6. Local Mux and Logical OR on the TMS320F2837xD MCU

- Consistent timing: Similar to the above, since there is no CPU involvement, the signal propagates at the system clock in real time as it happens and is not contingent on another block to allow the signal to pass (Figure 7). This results in more predictable and repeatable system behavior. Given that the nature of many of the signals is to pass into the actuation sub-system timing is especially critical.

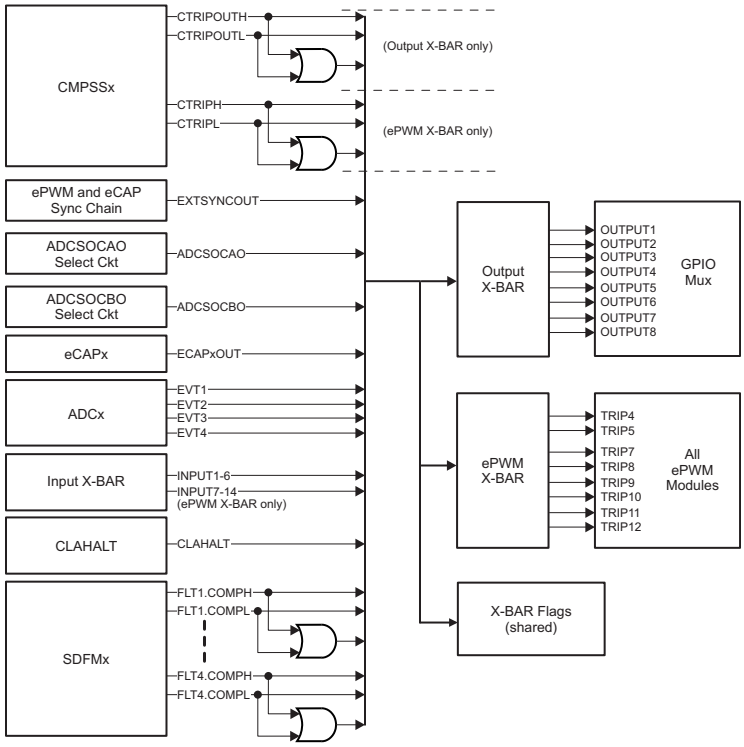

Figure 7. X-bar Sources and Destinations on the TMS320F2837xD MCU

2.2.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

2.2.4 Hardware/Software Resources

- C2000 DesignDRIVE Development Kit for Industrial Motor Control

2.2.5 Documentation

- *TMS320F2838x Microcontrollers Technical Reference Manual*
- *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual*
- *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*

2.3 Increasing Control Loop Bandwidth With An Independent Accelerator

2.3.1 Value Proposition

A primary concern in any control application is the time that elapses between sampling the system (sensing), applying the control function(processing), and applying the stimulus to the external system (actuation). The CLA was created specifically to address the need to minimize this time while increasing overall system throughput.

2.3.2 In Depth

The CLA found on the devices listed below is a fully parallel processor to the main C28x core. While the C28x core is a more traditional processor, executing instructions and servicing interrupts, the CLA is a task driven state machine. The CLA is a 32-bit floating point architecture.

Due to the nature of control systems, there are specific times when the sensing subsystem has new data to be processed. Advanced planning for these events with the other functions of an MCU can be difficult to time slice without introducing delay into both the system under control, but also any other functions the CPU has to perform.

As a task driven state machine, the CLA is constantly waiting in an idle state for an event, such as an ADC conversion, to process that data and actuate the system. Additionally, the CLA has full access to key control peripherals so it can fully realize the control system independent of the C28x CPU (Figure 8).

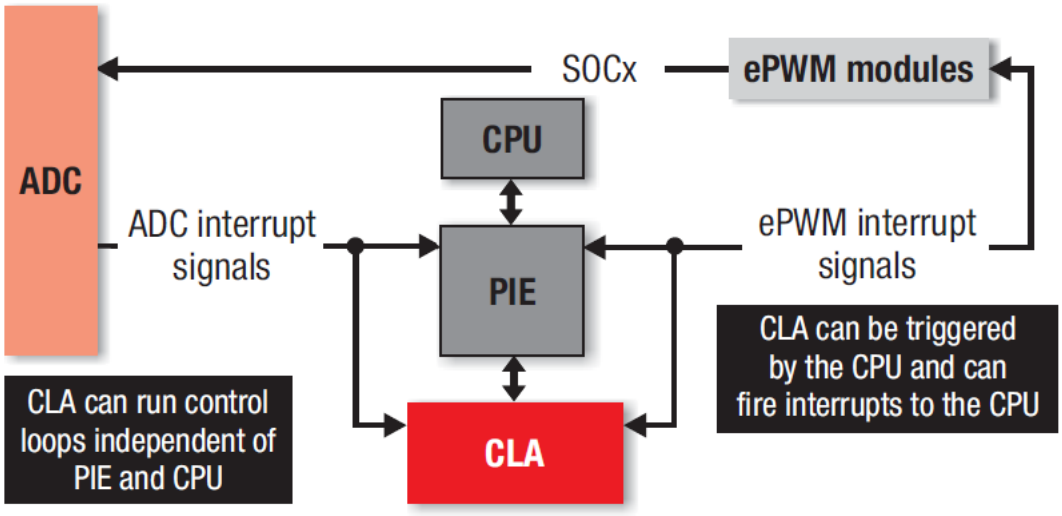

Figure 8. C28x and CLA interfacing with the ADC and ePWM modules

The above is beneficial for many reasons:

- There is little to no delay in processing the data, typically caused by the context switching of the main C28x core, and applying the new external stimulus to the system.
- There is no interruption or impact to the current C28x program execution.
- Potential to have parallel control systems running independently on the same MCU device.

The CLA has its own dedicated memory region for its code and shared memory for passing information between it and a C28x CPU in the system. The CLA is supported in Code Composer Studio by its own C compiler.

2.3.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

2.3.4 Hardware/Software Resources

- Valley switching boost power factor correction (PFC) reference design
- C2000 DesignDRIVE Development Kit for Industrial Motor Control

2.3.5 Documentation

- CLA Hands On Workshop
- CLA Usage in Valley Switching Boost Power Factor Correction (PFC) Reference Design
- CLA FAQ on E2E
- *Accelerators: Enhancing the Capabilities of the C2000™ MCU Family Technical Brief*

2.4 Deterministic Program Execution

2.4.1 Value Proposition

One important aspect of any real-time control system is the consistency of program execution over time. Whether it is performance across the system update period or from multiple power ups over time, optimization of real-time systems relies on critical system events taking place at known points in time. The well defined 8 stage CPU pipeline of the C28x MCU, as well as the complementary behavior of its interrupt handling logic, provides this level of determinism.

2.4.2 In Depth

A processor that is not consistent in its behavior over time can cause perturbations to a real-time control system, either by not actuating the system in a timely manner or by sampling the state of the system at the incorrect point in time. Furthermore, real-time control systems are largely interrupt driven in their program flow. For this reason, a cache memory is undesirable since it would need to be discarded often whenever a program discontinuity occurs (in this case quite often).

Instead, large amounts of fast memory are desirable for program execution as well as a processor with a instruction pipeline that is deep enough to parallelize instructions, but also shallow enough to not incur large time penalties when discontinuities occur. The C28x CPU employees an 8 stage pipeline as shown in Figure 9. Once an instruction has entered the D2 phase of the pipeline, it cannot be stopped from full execution by an interrupt. Conversely, any instruction that is in a pre-D2 phase of the pipeline will be flushed when the incoming interrupt is received by the C28x core. Upon returning from the interrupt program, execution begins again with the F1 fetch stage. It is beyond the scope of this article to go deeper into the nuances of the pipeline, but those details are covered in the *TMS320C28x DSP CPU and Instruction Set Reference Guide*.

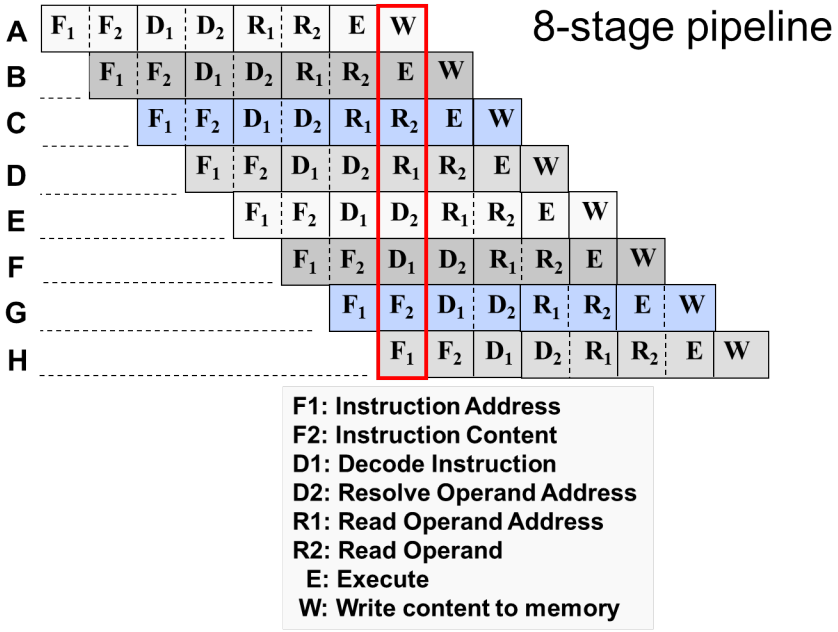

Figure 9. C28x Pipeline Visualization

While you can see that the behavior of the program execution is repeatable over time, what about the behavior of the incoming interrupt? Figure 10 shows that this, too, is deterministic. Note that unless manually altered all other interrupt requests will pend until the current interrupt has been fully serviced. This is important in order to keep servicing of any interrupt consistent in the time domain once it begins. Both of these components of the C28x core help maintain deterministic code execution for the system.

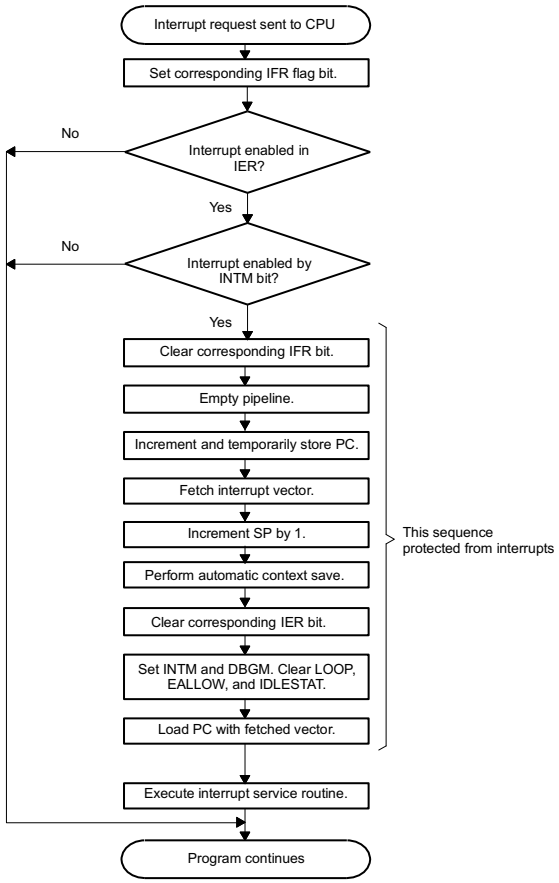

Figure 10. Standard Operation for a C28x CPU Maskable Interrupt

2.4.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

2.4.4 Hardware/Software Resources

- F28388D controlCARD evaluation module
- C2000 MCU F28379D LaunchPad™ development kit
- C2000 MCU F280049C LaunchPad™ development kit

2.4.5 Documentation

- *TMS320C28x CPU and Instruction Set Reference Guide*
- C2000™ F2837xD Microcontroller Workshop

3 Actuation Key Technologies

3.1 Reducing Limit Cycling in Control Systems

3.1.1 Value Proposition

Limit cycling in a PWM controlled system refers to the in-ability for the PWM output to physically converge on the mathematical solution to the control law. This causes the PWM output to cycle about the true solution, resulting in instability in the control system. The High Resolution PWM (HRPWM) module on the C2000 MCU has the ability to modulate the PWM edge in 150 ps increments. This represents a 60 fold improvement over traditional PWM creation techniques based off the system clock rate (Figure 11) and can be used to realize a higher order of accuracy in PWM edge placement. A waveform's period phase relationship to its complement, as well as deadband insertion time can all realize this high resolution benefit.

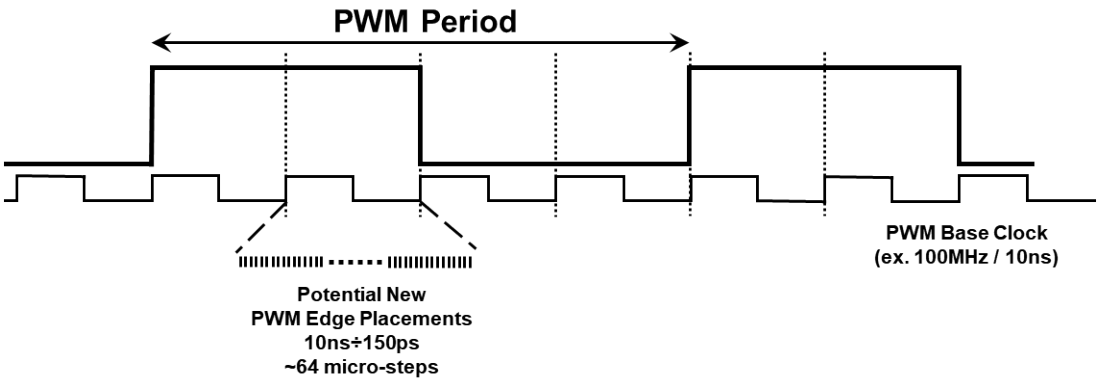

Figure 11. HRPWM Capability vs Traditional PWM Generation Methods

3.1.2 In Depth

All PWM-controlled power topologies are inherently bandwidth limited by the ability of the controller to place the PWM edge as close as possible to the mathematical solution of the control law. Whatever error exists from "rounding" the solution created in the form of an output PWM signal dictates the maximum efficiency that can be realized in the system.

In this sense, it may be helpful to think of the PWM as a type of DAC with a fixed resolution. Any error that results in the selection of the next available PWM edge placement would then be equivalent to quantization error term that is inherent to any DAC. Therefore, the minimum time step that can be achieved by a PWM module can be translated into "bits" of resolution of this equivalent DAC. As shown in Table 2, the increase of resolution of the C2000 MCU HRPWM vs a traditional PWM is very apparent, increasing the effective resolution by approximately 6 bits.

Table 2. Resolution for PWM vs HRPWM

PWM Freq (kHz)	Regular Resolution (PWM) 100 MHz EPWMCLK		High Resolution (HRPWM)	
	Bits	% Error	Bits	% Error
20	12.3	0.02	18.1	0.000
50	11	0.05	16.8	0.001
100	10	0.1	15.8	0.002
150	9.4	0.15	15.2	0.003
200	9	0.2	14.8	0.004
250	8.6	0.25	14.4	0.005
500	7.6	0.5	13.4	0.009
1000	6.6	1	12.4	0.018

Table 2. Resolution for PWM vs HRPWM (continued)

PWM Freq (kHz)	Regular Resolution (PWM) 100 MHz EPWMCLK		High Resolution (HRPWM)	
	Bits	% Error	Bits	% Error
1500	6.1	1.5	11.9	0.027
2000	5.6	2	11.4	0.036

3.1.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

3.1.4 Hardware/Software Resources

- TIDM-02002 Bidirectional CLLLC resonant dual active bridge for HEV/EV onboard charger
- TIDA-00961 Highly Efficient 1.6kW High Density GaN Based 1MHz CrM Totem-pole PFC Converter

3.1.5 Documentation

- *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual*
- C2000 F2837xD Microcontroller 1-Day Workshop Section 1.6: Control Peripherals skip to 14:38 for HRPWM

3.2 Shoot Through Prevention for Current Control Topologies

3.2.1 Value Proposition

C2000 MCUs have the ability to implement current control techniques, such as Peak Current Mode Control (PCMC), in hardware using the on-chip comparators to control the PWM duty cycle. Variable deadband insertion has been added to the Type 4 PWM, enabling the ability to tune out potential shoot through, without the need for any CPU overhead.

3.2.2 In Depth

For better power efficiency, many DC-DC systems implement a synchronous boost controller, where the secondary switch replaces the feed forward diode that exists in a regular boost controller (Figure 12). Peak Current Mode Control is one of the more common methods used to control this topology and the C2000 MCU has some unique features that allow it to implement this type of control very efficiently.

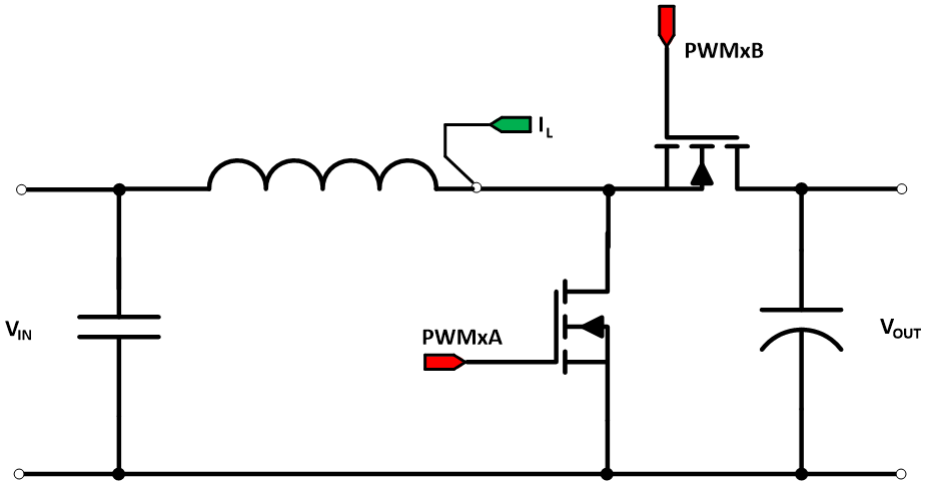

Figure 12. Synchronous Boost Controller

The addition of the second FET demands precise control of the ON/OFF time of the primary and secondary switch relative to one another. If the switches are in the “ON” state at the same time, there is a direct path to ground for the active current to flow, which is not only in-efficient but also potentially harmful to the lifetime of the FET switches.

An accurate way to implement this type of system is to have the comparator monitor the inductor current and actuate the FETs when the current exceeds a predefined threshold. Ideally, when one FET is switched ON, the other FET can be switched OFF at the same time. However, due to switch mis-match and board propagation delays, simultaneous switching from the controller likely will not result in simultaneous switching at the FETs, which creates the shoot through mentioned earlier. While techniques in software can help hold off the switching of the secondary switch to avoid this condition, these can be challenging to implement with the various hardware interdependencies coupled with the time constraints of the control loop.

The C2000 MCU has implemented programmable deadband control, derived from the comparator output itself, to prevent this condition while keeping the C28x CPU unloaded (Figure 13. This allows a complete PCMC solution to be realized outside of the CPU domain once initialized. This logic exists on all PWM modules on a given device, allowing multiple stages to have different deadbands, such as a Phase Shifted Full Bridge, where there are multiple switching pairs.

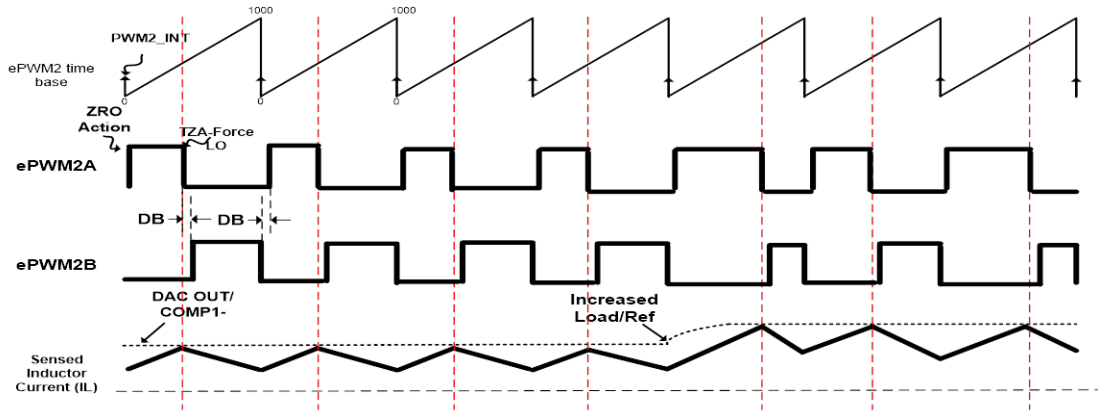

Figure 13. Cycle by Cycle Trip Action of the COMP Module With Configurable Deadband

3.2.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

3.2.4 Documentation

- *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual* - see the *Action-Qualifier (AQ) Submodule* and *Dead-Band Generator (DB) Submodule* sections of the *Enhanced Pulse Width Modulator (ePWM)* chapter
- *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual* - see the *Action-Qualifier (AQ) Submodule* and *Dead-Band Generator (DB) Submodule* sections of the *Enhanced Pulse Width Modulator (ePWM)* chapter
- F2837xD MCU 1 day Workshop Control Peripherals

4 Sensing Key Technologies

4.1 Accurate Digital Domain Representation of Analog Signals

4.1.1 Value Proposition

Many MCUs have integrated ADCs as part of their sensing subsystem. The ability of the ADC to accurately convert the analog domain to the digital space is one of the most crucial aspects of the MCU in order to realize a proper control system. The data sheet specifications for a C2000 MCU ADC are such that performance in the system can be properly evaluated prior to system implementation.

4.1.2 In Depth

The first step when selecting an MCU for a real-time control system is relatively straightforward process; comparing the components of the MCU to the system needs. There are questions of memory size, CPU speed, communications standards used, analog content, number of I/Os, and so forth. When looking at the fit for an analog module like the ADC, it can appear straightforward to base the decision on sampling rate, number of inputs, and bit level. In practice, however, there is much more to this decision.

Too often ADC selection is based solely on the top level specifications, only to realize during development there are limitations to the system performance due to the ADC itself:

- Will the system be using the analog inputs for frequency analysis? Then, AC specifications like SNR and THD become important to consider when picking an MCU with an on-chip ADC.
- Is overall accuracy a key care about? Looking at the DC specifications like INL, Gain, and Offset are key parameters to consider.
  A quick summary of ADC specifications and their relevance to the system:
  - AC Specifications: Parameters related to how accurately the converter can resolve the fundamental frequency tone of a signal from other noise sources. Includes SNR, SINAD, THD, and SFDR all expressed in dB. Also includes ENOB, which is the SINAD translated into number of bits. Typically SINAD and ENOB based on SINAD are considered when choosing an ADC, the importance will vary depending on the end application.
  - DC Specifications: Parameters related to the accuracy of the converter as it applies to representing an analog input in the digital domain. Includes Gain, Offset, DNL, and INL. The weighted summation of the Gain, Offset, and INL, $\sum \sqrt{(Err_{gain})^2 + (Err_{offset})^2 + (Err_{INL})^2}$ are often referred to as "Total Unadjusted Error". This equation is typically used to determine the real-world impact of these parameters on the accuracy of a conversion.

An example of how the C2000 ADC is specified and the parameters can be seen in Table 3, a dynamic link to this same table in the data sheet is located here.

One final aspect of all the parameters that C2000 devices list in the data sheet is what is implied by the inclusion of the parameter itself. For parameters that have a MIN/MAX, these are assured specs over the full operational range and lifetime of the device. The typical (TYP) column is also significant for all parameters, as it represents the mean performance of a parameter across its operational range.

Table 3. TMS320F28379D 16-bit ADC Specifications

Parameter	Test Conditions	Min	Typ	Max	Unit
ADC conversion cycles		29.6		31	ADCCLKs
Power-up time (after setting ADCPWDNZ to first conversion)				500	µs
Gain error		–64	±9	64	LSBs
Offset error		–16	±9	16	LSBs
Channel-to-channel gain error			±6		LSBs
Channel-to-channel offset error			±3		LSBs
ADC-to-ADC gain error	Identical V_{REFHI} and V_{REFLO} for all ADCs		±6		LSBs
ADC-to-ADC offset error	Identical V_{REFHI} and V_{REFLO} for all ADCs		±3		LSBs
DNL		> –1	±0.5	1	LSBs
INL		–3	±1.5	3	LSBs
SNR	V_{REFHI} = 2.5 V, f_{in} = 10 kHz		87.6		dB
THD	V_{REFHI} = 2.5 V, f_{in} = 10 kHz		–93.5		dB
SFDR	V_{REFHI} = 2.5 V, f_{in} = 10 kHz		95.4		dB
SINAD	V_{REFHI} = 2.5 V, f_{in} = 10 kHz		86.6		dB
ENOB	V_{REFHI} = 2.5 V, f_{in} = 10 kHz, single ADC		14.1		bits
	V_{REFHI} = 2.5 V, f_{in} = 10 kHz, synchronous ADCs		14.1		
	V_{REFHI} = 2.5 V, f_{in} = 10 kHz, asynchronous ADCs		Not supported		
PSRR	V_{DDA} = 3.3-V DC + 200 mV DC up to Sine at 1 kHz		77		dB
PSRR	V_{DDA} = 3.3-V DC + 200 mV Sine at 800 kHz		74		dB
CMRR	DC to 1 MHz		60		dB
V_{REFHI} input current			190		µA
ADC-to-ADC isolation	V_{REFHI} = 2.5 V, synchronous ADCs	–2		2	LSBs
	V_{REFHI} = 2.5 V, asynchronous ADCs		Not supported		

4.1.3 Device List

- TMS320F2838xD/S
- TMS320F28378xD/S
- TMS320F2807x
- TMS320F28004x

4.1.4 Hardware/Software Resources

All controlCARDs for their specific C2000 MCU have been verified to reproduce the DS specifications for the on-chip ADC

- TMDSCNCD28379D
- TMDSCNCD280049C

4.1.5 Documentation

- ADC Specifications for TMS320F2838xD/S
- ADC Specifications for TMS320F2837xD/S
- ADC Specifcations for TMS320F2807x
- ADC Specifications for TMS320F28004x

4.2 Resolving Tolerance and Aging Effects During ADC Sampling

4.2.1 Value Proposition

The ADC result is often manipulated mathematically before its used in the control law of a given system. This is typically done with some additional operations by the CPU, adding increased latency to the system as well as loading the CPU for such operations. The C2000 MCU as the ability to correct for this in hardware with no CPU overhead and no impact to ADC sample rate.

4.2.2 In Depth

Prior to using the ADC result in control calculations, it is often necessary to remove any known offset introduced by external factors such as component tolerances or layout differences (Table 4). While aspects of the above issues can be partially addressed through PCB layout or choosing higher tolerance/stable resistors there are always deviations from the ideal.

Table 4. Typical Resistor Tolerance Over Time and System Impact

Life Cycle Stage	Total Tolerance	Associated 12-Bit Error
Purchase	±0.05%	±2 LSBs
Post Assembly	±0.5%	±20 LSBs
Post Storage/Moisture	±0.75%	±30 LSBs
Temp Coeff and EOL	±1.00%	± 40 LSBs

C2000 MCUs implement an integrated hardware block to correct up to a 10-bit signed value co-incident to the ADC conversion process, saving valuable cycles in the system. The cycle value to the system is effectively doubled, as the ADC sample rate is maintained and no CPU cycles are used to perform the correction. Saturation is built in as well. For the implementation of the offset correction in addition to the other modules included in the ADC Post Processing Block, see Figure 14.

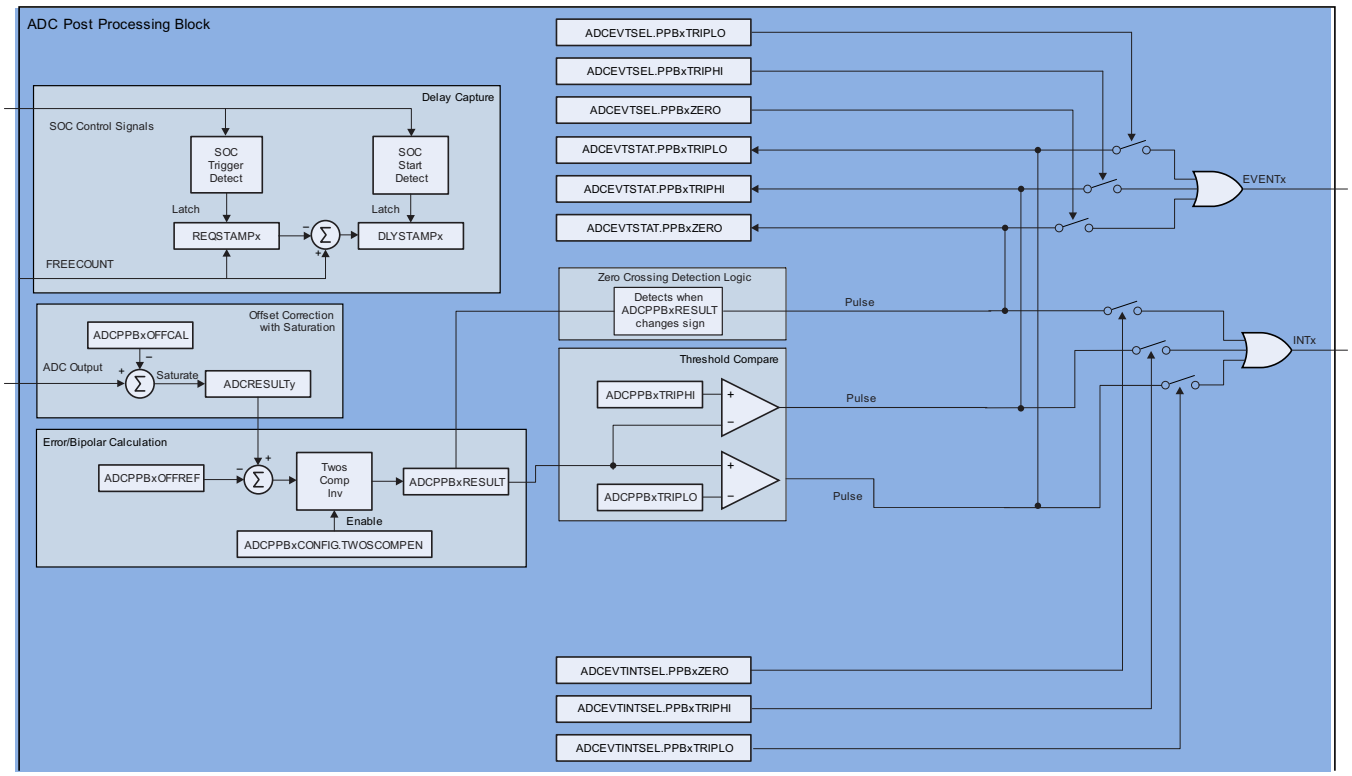

Figure 14. ADC Post Processing Block on TMS320F2837xD

4.2.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

4.2.4 Hardware/Software Resources

- TMDSIDDK379D

4.2.5 Documentation

- *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual* - see the *Post-Processing Blocks* section in the *Analog-to-Digital Converter (ADC)* chapter
- *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual* - see the *Post-Processing Blocks* section in the *Analog-to-Digital Converter (ADC)* chapter

4.3 Fast Detection of Over and Under Currents and Voltages

4.3.1 Value Proposition

Every control system can experience random events that can cause damage to the system. Fast detection and reaction to these events is critical to keeping the system safe and in good working condition. The on-chip comparators can detect and react to these events in a fraction of the time that it would take for an ADC and processor.

4.3.2 In Depth

Fault detection and reaction is important in most systems, not only for avoiding an undefined output, but for preventing damage to components both on and off the main Printed Circuit Board (PCB). The speed at which the fault detection takes place, as well as the final FET output state change, is critical to the system. A dedicated subsystem that ties together the analog and digital domains has been implemented on the C2000 MCU for handling this requirement: the Comparator Subsystem or CMPSS (Figure 15).

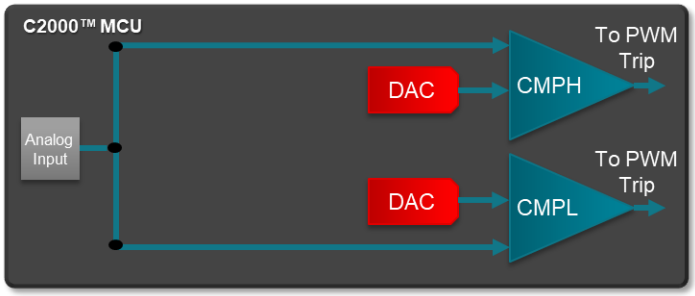

Figure 15. CMPSS Visualization

There are up to eight CMPSS modules on each C2000 MCU, with internal DACs that give the inverting/comparison detection level for the line that is being monitored. As shown in Figure 15, each CMPSS module has two comparators for simultaneous high and low detection. Using the CMPSS has several advantages over using the ADC for fault detection:

- System Overhead: Using the CMPSS to monitor a pin is an essentially zero overhead operation after the initial setup. The pin is always monitored against the comparison value until disabled. Other techniques would require periodic ADC conversions and threshold checking.
- Latency: While the ADC sampling rate can be simply factored into the period of the control loop, there is not a deterministic constant to a fault condition. As such there will be an inherent delay to detect the fail, both from a point of sample to the conversion time of the ADC itself. The comparator has no such trigger requirement or sample time, it is continuously monitoring the analog signal.
- Dedicated PWM Trip Zone input: The output of every CMPSS module can be tied directly into the Trip Zone of any PWM, and the action when the signal is received is configurable in software. This means there is no software overhead as there would be in processing an ADC ISR to then create the action to the PWM in software.
- No clock dependence: Since by definition a comparator is a purely analog domain circuit there are not clock dependencies to the changing state of its output based on the input. The C2000 MCU has carried this forward to give a asynchronous path from the comparator to the PWM. This allows for the fastest possible time from fault detection to pin state change, in addition to removing any clock dependence (Table 5).

Table 5. Comparison of Fault Detection and Trip Methods

Sampling Method	Sample Time (min)	Result Ready (min)	Latch and Change PWM Pin (@200 MHz SysClk)	Total Time From Fault To Trip
12-bit ADC	75 ns	260 ns	approximately 100 ns(inc ISR)	435 ns
12-bit ADC w/PPB	75 ns	260 ns	10 ns	355 ns
CMPSS	NA	NA	NA	60 ns

- Simultaneous high and low detection: Each input to a CMPSS module routes the signal to two physically independent comparators that give the ability to detect both overshoot and undershoot at the same time.

4.3.3 Device List

- TMS320F2838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

4.3.4 Hardware/Software Resources

The following kits implement the CMPSS to detect out of range current/voltage events

- TIDM-02002 Bi-Directional CLLLC Resonant Dual Active Bridge Reference Design for HEV/EV Onboard Charger
- TIDM-1022 Valley Switching Boost Power Factor Corrector
- TMDXIDDK379D C2000 Design DRIVE Development Kit for Industrial Motor Control

4.3.5 Documentation

- Comparator Specifications in the TMS320F28379D Data Sheet
- *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual* - see the *Comparator Subsystem (CMPSS)* chapter

4.4 Smart Sensing Across An Isolation Boundary

4.4.1 Value Proposition

High voltage control systems often make use of a sigma-delta ADC in order to easily pass the analog information of the high voltage domain to the lower voltage domain where the MCU exists. Before the data stream can be used, it must be processed by a filter and demodulator. This logic exists on the C2000 MCU and is called the Sigma Delta Filter Module (SDFM) module. The high and low comparators inside the SDFM on a C2000 MCU can actuate the PWMs without CPU intervention saving valuable time to better control the system.

4.4.2 In Depth

A sigma-delta type ADC is by design an over-sampling architecture. The ADC itself is a single bit design that over-samples the signal of interest to produce a higher order output. The modulation of this type of converter results in a binary output and, hence, a serial data stream. This serial data stream is then sent to the filter/demodulator on the C2000 MCU for re-construction into a higher bit order digital representation of the sampled signal (Figure 16) .

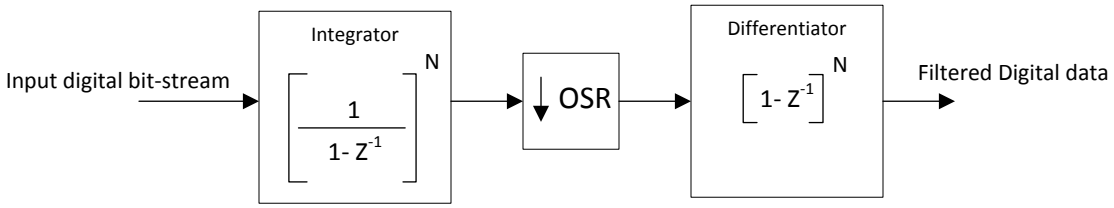

Figure 16. Filter and Demodulator Inside the C2000 SDFM

At this point an interrupt is generated to the MCU, informing the other domains there is new data to process and act upon.

The SDFM module on the C2000 MCU is unique in that not only can the converted data be read post filtering, but the PWMs can be switched based on this data automatically. Each SDFM module contains four channels. Inside each channel exists two filters: a primary filter that produces the SDFM data and a secondary filter containing both high and low limit comparators (Figure 17). This allows the system to control the PWM signals without waiting for CPU intervention, resulting in both lower latency as well as lower overall CPU utilization.

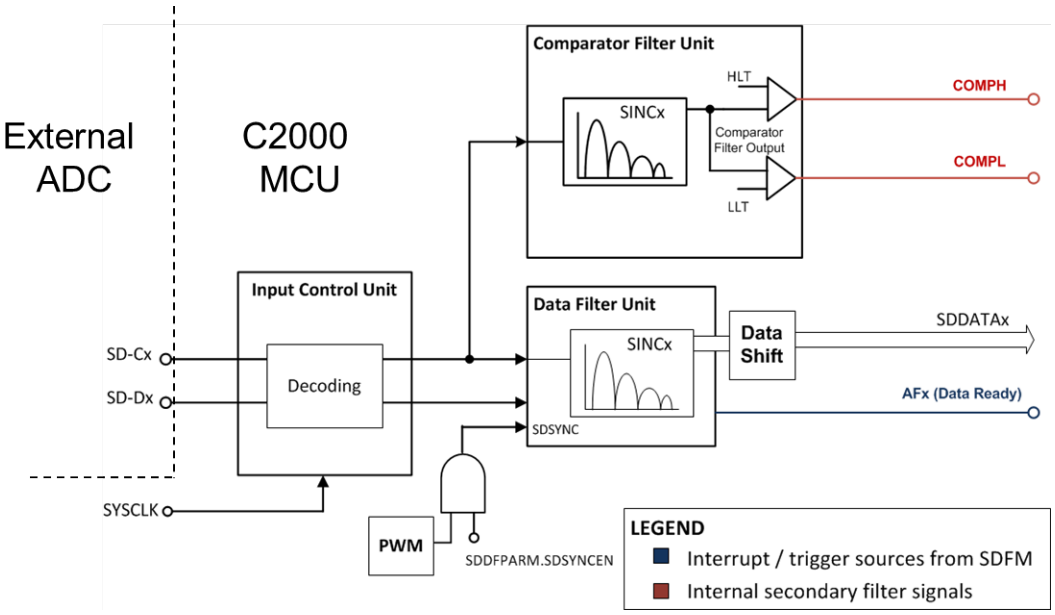

Figure 17. SDFM Module with both Primary and Secondary Filter Blocks on theTMS320F2837xD MCU

5 References

5.1 Device List

- TMS320F3838xD/S
- TMS320F2837xD/S
- TMS320F2807x
- TMS320F28004x

5.2 Hardware/Software Resources

- TIDM-1007 Interleaved CCM Totem Pole Bridgeless Power Factor Correction (PFC) Reference Design
- TIDM-HV-1PH-DCAC Single-Phase Inverter Reference Design With Voltage Source and Grid Connected Modes
- TMDXIDDK379D C2000 DesignDRIVE Development Kit for Industrial Motor Control
- TMDSHVMTRINSPIN High Voltage Motor Control Kit with InstaSPIN-FOC and InstaSPIN-MOTION enabled Piccolo MCU for F280049C device lab7 and lab8
- Vienna Rectifier-Based Three Phase Power Factor Correction Reference Design Using C2000 MCU
- TIDM-02002 Bidirectional CLLLC resonant dual active bridge for HEV/EV onboard charger
- TIDA-00961 Highly Efficient 1.6kW High Density GaN Based 1MHz CrM Totem-pole PFC Converter
- F28388D controlCARD evaluation module
- C2000 MCU F28379D LaunchPad™ development kit
- C2000 MCU F280049C LaunchPad™ development kit
- C2000 DesignDRIVE Development Kit for Industrial Motor Control
- Valley switching boost power factor correction (PFC) reference design
- TMDSIDDK379D
- TMDSCNCD28379D
- TMDSCNCD280049C
- TIDM-1022 Valley Switching Boost Power Factor Corrector

5.3 Documentation

- Texas Instruments: *C2000 Real-Time Control Peripherals Reference Guide*
- Texas Instruments: *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual*
- Texas Instruments: *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*
- Texas Instruments: *Accelerators: Enhancing the Capabilities of the C2000 MCU Family Technical Brief*
- Texas Instruments: *TMS320C28x Extended Instruction Sets Technical Reference Manual*
- Texas Instruments: *TMS320C28x CPU and Instruction Set Reference Guide*
- ADC Specifications for TMS320F2838xD/S
- ADC Specifications for TMS320F2837xD/S
- ADC Specifcations for TMS320F2807x
- ADC Specifications for TMS320F28004x
- C2000™ F2837xD Microcontroller Workshop
- CLA Hands On Workshop
- CLA Usage in Valley Switching Boost Power Factor Correction (PFC) Reference Design
- CLA FAQ on E2E
- C2000 F2837xD Microcontroller 1-Day Workshop Section 1.6: Control Peripherals

IMPORTANT NOTICE AND DISCLAIMER

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

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

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

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