User's Guide

Migration Between TMS320F2802x, TMS320F2803x and F28E12x Guide

ABSTRACT

This application report describes the differences between the Texas Instruments TMS320F2802x/ TMS320F2803x and the F28E12x microcontrollers for the purpose of assisting with application migration. Functions that are identical in both devices are not necessarily included. All efforts have been made to provide a comprehensive list of the differences between the two device generations within the C2000™ product family; however, the descriptions are explained only to the extent of highlighting areas that require attention when migrating from one device to another. For a detailed description of features specific to each device, see the most recent device-specific data sheet, technical reference manual, errata, user guides, and software packages.

Table of Contents

List of Tables

Trademarks

C2000™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

The TMS320F2802x/TMS320F2803x and F28E12x are device members of the C2000™ MCU product family. These devices are most commonly used within embedded control applications. The F28E12x devices feature an updated version of the enhanced control peripherals found on the TMS320F2802x/TMS320F2803x, which allows for greater flexibility and improved application performance. In addition, the F28E12x devices feature a boot mode flow that enables expanded booting options that provide the ability to use alternate, reduce, or completely eliminate boot mode selection pins, and an analog subsystem interconnect that enables a very flexible pin usage, allowing for smaller device packages. Central to the analog subsystem interconnect are multiple analog comparator subsystems (CMPSS). Enhancements include the addition of a floating-point unit (FPU) to the CPU, a direct memory access (DMA) controller and three cross-bars (XBARs) for providing a flexible means for interconnecting multiple inputs, outputs, and internal resources.

For the purposes of migration, TMS320F2802x and TMS320F2803x – these devices are referenced as the F2802x and F2803x, respectively. When referenced as both the F2802x and F2803x, this combined group of devices can be referenced as F2802x/03x. If a feature is unique to a specific device type, it is referenced as F2802x or F2803x.

For a full list of devices currently available within the F2802x, F2803x, and F28E12x families, see the TI website at http://www.ti.com/c2000.

As the focus of this document is to describe the differences between the two device groups, the descriptions are explained only to the extent of highlighting areas that require attention when moving an application from one device to the other. For a detailed description of features specific to each device, see the device-specific technical reference manuals and user's guides available on the TI website at http://www.ti.com/c2000. This application report does not cover the silicon exceptions or advisories that can be present on each device.

Consult the following silicon errata for specific advisories and workarounds:

- *TMS320F28E12x MCUs Silicon Errata*
- *TMS320F2803x MCUs Silicon Silicon Errata*
- *TMS320F2802x, TMS320F2802xx MCUs Silicon Errata*

Note

For information regarding any electrical specifications, always refer to the TMS data sheet.

1.1 Abbreviations

The following abbreviations are used in this document:

- F2802x: Refers to the TMS320F2802x devices. For example, TMS320F28027, TMS320F28026, TMS320F28023, TMS320F28022, TMS320F28021, TMS320F28020, and TMS320F280200. The individual devices in this group are abbreviated as F28027, F28026, F28023, F28022, F28021, F28020, and F280200.
- F2803x: Refers to the TMS320F2803x devices. For example, TMS320F28035, TMS320F28034, TMS320F28033, TMS320F28032, TMS320F28031, and TMS320F28030. The individual device in this group is abbreviated as F28035, F28034, F28033, F28032, F28031, and F28030.
- F28E12x: Refers to the F28E12x devices. For example, F28E120SC and F28E120SB. The individual devices in this group are abbreviated as F28E12x.

For a full list of devices currently available within the F2802x, F2803x, and F28E12x families, see the TI website at http://www.ti.com/c2000.

2 Central Processing Unit (CPU)

The F28E12x devices extend the capabilities of the existing TI C28x 32-bit fixed-point CPU architecture by adding a floating-point unit (FPU). No changes have been made to existing instructions, pipeline, or memory bus architecture; and programs written for the C28x CPU are completely compatible with these architectural enhancements.

The addition of the Floating-Point Unit (FPU) to the C28x fixed-point CPU core enables support for hardware IEEE-754 single-precision floating-point format operations. The FPU adds a set of floating-point registers (R0H to R7H, STF, RB) and instructions as an extension to the standard C28x architecture, providing seamless integration of floating-point hardware into the CPU. In the pipeline decode stage, the instruction is decoded to determine if it is a standard C28x instruction or a FPU instruction, and is routed accordingly. Since the FPU instructions are extensions of the standard C28x instruction set, most instructions operate in one or two pipeline cycles and some can be done in parallel. Also, the FPU latched overflow and underflow flags are connected to the peripheral interrupt expansion (ePIE) block that assists in debugging overflow and underflow issues.

The C28x CPU and FPU architecture and instruction set are documented in the following reference guides:

- *TMS320C28x CPU and Instruction Set Reference Guide*
- *TMS320C28x Extended Instruction Sets Technical Reference Manual*
- *C2000 Real-Time Control Peripherals Reference Guide*
  - This document describes the differences for the above ‘unit’ types; where a type change represents a major functional feature difference.
- *F28E12x Microcontrollers Technical Reference Manual*

3 Development Tools

The F28E12x devices have a new set of bit field header files, a new driver library, and new code examples, which are available in C2000Ware. C2000Ware is the successor to controlSUITE as the centralized repository for software and documentation. It has a new structure and all new content updates are available through C2000Ware and Software Development Kits (SDK) only. Note that C2000Ware, unlike controlSUITE, is versioned at the package level and results in a separate directory installation for each revision. C2000Ware can be downloaded from:

- http://www.ti.com/tool/C2000WARE

The SDKs are not included in the base C2000Ware download and needs to be downloaded and installed separately. Each SDK contains the development kit files and collateral related to the specific application solution. The SDKs also include a full version of the C2000Ware package. For additional information, see the *controlSUITE™ to C2000Ware Transition Guide*.

3.1 Driver Library (Driverlib)

The Driver Library (Driverlib) is a set of drivers for accessing the peripherals and device configuration registers. While Driverlib is not drivers in the pure operating system sense (does not have a common interface and does not connect into a global device driver), they do provide a software layer to facilitate a slightly higher level of programming. Driverlib provides a more readable and portable approach to peripheral register programming. This portability allows for an easier migration to future device families since the function calls can remain the same even though the control bits may change within and between registers.

3.2 Migrating Between IQ_Math and Native Floating-Point

The following steps must be taken to convert a project written in IQmath format to native floating point:

1. Select FLOAT_MATH in the IQmath header file. The header file converts all IQmath function calls to their floating-point equivalent.
2. Convert the floating-point number to an integer when writing a floating-point number into a device register. Likewise, when reading a value from a register, it needs to be converted to float. In both cases, this is done by multiplying the number by a conversion factor. For example, to convert a floating-point number to IQ15, multiply by 32768.0 as shown below:

```
#if MATH_TYPE == IQ_MATH
   PwmReg = (int16)_IQtoIQ15(Var1);
#else // MATH_TYPE is FLOAT_MATH
   PwmReg = (int16)(32768.0L*Var1);
#endif
```

   To convert from an IQ15 value to a floating-point value, multiply by 1/32768.0 or 0.000030518.
3. Do the following to take advantage of the on-chip floating-point unit:
   a. Use Code Composer Studio with the C28x codegen tools version 19.6.0 or later.
   b. Set the compiler so that it can generate native C28x floating-point code. To do this, use the –v28 --float_support=fpu32 compiler switches. In Code Composer Studio, the float_support switch is under Compiler Options → Processor Options.
   c. Use the correct run-time support library for native 32-bit floating-point. It is recommended to include libc.a rather than directly including the RTS library. Note that libc.a is an index that automatically selects the correct .lib based on the project properties.
   d. Consider using the C28x FPU Fast RTS Library (C2000Ware → libraries → math → FPUfastRTS) to get a performance boost from math functions such as sin, cos, div, sqrt, and atan. The Fast RTS Library can be linked in before the normal run-time support library.

3.3 Embedded Application Binary Interface (EABI) Support

The F28E12x is one of the C2000 device families to migrate from Common Object File Format (COFF) to Embedded Application Binary Interface (EABI). EABI overcomes several limitations of COFF, which includes the symbolic debugging information not being capable of supporting C/C++, and the limit on the maximum number of sections and length of section names and source files. Note that EABI and COFF are not compatible and conversion between the two formats is not possible. The following is a brief summary of EABI differences compared to COFF.

- Direct initialization
  - Uninitialized data is zero by default in EABI.
  - Initialization of RW data is accomplished via linker-generated compressed copy tables in EABI.
- C++ language support
  - C++ inline function semantics: In COFF, inline functions are treated as static inline and this causes issues for functions that cannot be inlined or have static data. In EABI, inline functions without the ‘static’ qualifier have external linkage.
  - Better template instantiation: COFF uses a method called late template instantiation and EABI uses early template instantiation. Late template instantiation can run into issues with library code and can result in long link times. Early instantiation uses ELF COMDAT to guarantee templates are always instantiated properly and at most one version of each instantiation is present in the final executable.
  - Table-Driven Exception Handling (TDEH): Almost zero impact on code performance as opposed to COFF that uses setjmp/longjmp to implement C++ exceptions Features enabled by EABI.
- Features enabled by EABI
  - Location attribute: Specify the run-time address of a symbol in C-source code.
  - Noinit/persistent attribute: Specify if a symbol cannot be initialized during C auto initialization.
  - Weak attribute: Weak symbol definitions are pre-empted by strong definitions. Weak symbol references are not required to be resolved at link time. Unresolved weak symbols resolve to 0.

  - External aliases: In COFF, the compiler makes A an alias to B if all calls to A can be replaced with B. A and B must be defined in the same file. In EABI, the compiler makes A an alias to B even if B is external.
- Calling convention
  - Scalar calling convention is identical between COFF and EABI.
  - Struct calling convention (EABI):
    - Single field structs are passed/returned by value corresponding to the underlying scalar types.
    - For FPU32, homogenous float structs with size less than 128 bits are passed by value.
    - Passed in R0H-R3H, then by value on the stack.
    - Structs that are passed by value are also candidates for register allocation.
    - For FPU64, the same applies for 64-bit doubles (R0-R3).
- Double memory size
  - In EABI, double is 64-bit size while in COFF, double is still represented as 32-bit size.
  - C/C++ requires that double be able to represent integer types with at least 10 decimal digits, which effectively requires 64-bit double precision.

Table 3-1 summarizes the compiler-generated section names used by COFF and EABI.

Table 3-1. Section Names

Description	COFF	EABI
Read-Only Sections		
Const data	.econst	.const
Const data above 22-bits	.farconst	.farconst
Code	.text	.text
Pre-main constructors	.pinit	.init_array
Exception handling	N/A	.c28xabi.exidx/.c28xabi.extab
Read-Write Sections		
Uninitialized data	.ebss	.bss
Initialized data	N/A	.data
Uninitialized data above 22-bits	.farbss	.farbss
Initialized data above 22-bits	N/A	.fardata
Heap	.esysmem	.sysmem
Stack	.stack	.stack
CIO Buffer	.cio	.bss:cio

For more information about EABI and the migration process, see the following reference guides:

- *TMS320C28x Assembly Language Tools User's Guide*
- *TMS320C28x Optimizing C/C++ Compiler User's Guide*
- *C28x Embedded Application Binary Interface*
- C2000 Migration from COFF to EABI https://software-dl.ti.com/ccs/esd/documents/C2000_c28x_migration_from_coff_to_eabi.html

4 Package and Pinout

Both the F2802x and the F28E12x devices are available in an 48-pin low-profile quad flatpack (LQFP) package; however they are not pin-compatible. All other package options for the F2802x/03x and the F28E12x devices are neither package nor pin-compatible. Any application being migrated from the F2802x/03x to the F28E12x requires a new board layout to accommodate the changes in the pinout and/or the package. For more information, see the *F28E12x Microcontrollers Data Sheet*.

5 Operating Frequency and Power Management

The F28E12x devices have a maximum operating frequency of 160MHz. By comparison, the F2802x devices have a maximum operating frequency of 60MHz, 50MHz, or 40MHz depending upon the specific device family member, and the F2803x devices have a maximum operating frequency of 60MHz.

The F28E12x devices are powered by a 3.3V supply and use an internal 1.2V LDO Voltage Regulator (VREG) to supply the required 1.2V to the core (VDD). The F28E12x devices do not have a VREGENZ pin, and therefore the internal VREG is always enabled. The F2802x and F2803x devices are powered by a 3.3V supply and have the option to power the 1.8V core (VDD) externally or through the internal LDO Voltage Regulator (VREG). Unlike the F28E12x devices, the F2802x and F2803x devices have a VREGENZ pin. The POR and BOR circuits on the F28E12x and F2802x/03x devices are functionally the same. For details related to power management, see the *F28E12x Microcontrollers Data Sheet*.

6 Power Sequencing

Before powering the F28E12x devices, no voltage larger than 0.3V above VDDIO can be applied to any digital pin, and no voltage larger than 0.3V above VDDA can be applied to any analog pin (including VREFHI). The 3.3V supplies VDDIO and VDDA should be powered up together and kept within 0.3V of each other during functional operation. The VDD sequencing requirements are handled by the device. The F2802x and F2803x devices do not have specific power sequencing requirements. For information related to any electrical specifications, see the *F28E12x Microcontrollers Data Sheet*.

7 Memory Map

The memory map between the F28E12x and F2802x/03x are different and code must be rebuilt accordingly. This section covers the key differences. For specific details about the memory map, see the *F28E12x Microcontrollers Data Sheet*.

7.1 Random Access Memory (RAM)

This section highlights the major differences in the RAM memory. The various memory block addresses and sizes differ between the F28E12x devices and F2802x/03x devices, except for memory blocks M0 and M1. Additionally, memory blocks M0 and M1 on the F28E12x are ECC protected.

- The F28E12x has 8K x 16 words of RAM (parity-protected). This memory is divided into 2K x 16 words of dedicated (M0 – M1) and 6K x 16 words of global shared (GS0) RAM. By comparison, the F2802x has up to 6K x 16 words of RAM available and the F2803x has up to 10K x 16 words of RAM available.
- The F28E12x GS0 RAM block can be used as a source and/or destination for each of the two DMA channels. DMA is not available on the F2802x/03x devices.

Table 7-1 summarizes the F28E12x RAM memory blocks by size, address range, access, and protection/ security.

Table 7-1. F28E12x RAM Addresses and Memory Block Sizes

Memory	Size	Address Range	DMA Access	ECC/ Parity	Access Protection	Security
M0 RAM	1K x 16	0x0000 0000 - 0x0000 03FF	-	Parity	Yes	-
M1 RAM	1K x 16	0x0000 0400 - 0x0000 07FF	-	Parity	Yes	-
GS0 RAM	6K x 16	0x0000 C000 - 0x0000 D7FF	Yes	Parity	Yes	-

7.2 Flash and OTP

This section highlights the major differences in the flash and OTP memory.

7.2.1 Size and Number of Sectors

The size and number of sectors has changed and code must be rebuilt accordingly. The exact flash size as well as sector configuration varies from device to device. For details, see Table 7-2. Note that code to program the flash can be executed out of RAM and there cannot be any kind of access to the flash bank when an erase or program operation is in progress.

Table 7-2. Flash Sector Configuration by Device

	F280200	F28026 F28022 F28020	F28027 F28023 F28021	F28030	F28033 F28032 F28031	F28035 F28034	F28E120SC	F28E120SB
Sectors	2	4	4	4	8	8	64	32
Size	4K x 16	4K x 16	8K x 16	4K x 16	4K x 16	8K x 16	64K x 16	32K x 16
Total	8K x 16	16K x 16	32K x 16	16K x 16	32K x 16	64K x 16	64K x 16	32K x 16

7.2.2 Flash Parameters

The flash parameters differ between the F28E12x and F2802x/03x. For details, see the *F28E12x Microcontrollers Data Sheet*.

7.2.3 Entry Point into Flash

The flash entry point can be set to one of four pre-defined address options. Assigning the specific entry address to use is determined by the user-configurable boot definition table BOOTDEF registers Z1-OTP-BOOTDEF-HIGH/Z1-OTP-BOOTDEF-LOW (note that BOOTDEF registers Z2-OTP-BOOTDEF-HIGH/Z2-OTP-BOOTDEF-LOW is used when Z2-OTP-BOOTPIN-CONFIG is configured). These registers are located in the DCSM OTP. During development and debug, the emulation equivalent BOOTDEF registers EMU-BOOTDEF-HIGH/ EMU-BOOTDEF-LOW allow experimenting with different boot mode options without programming the OTP.

Table 7-3 summarizes the flash entry points.

Table 7-3. Flash Entry Points

Option	BOOTDEF Value	Flash Entry Point	Flash Sector
0 (default)	0x03	0x0008 0000	Bank 0 Sector 0
1	0x23	0x0008 8000	Bank 0 Sector 32

7.2.4 Dual Code Security Module (DCSM) and Password Locations

The DCSM offers protection for two zones (zone-1 and zone-2), and is intended to block access and visibility to the various on-chip memory resources with the purpose of preventing duplication and reverse engineering of proprietary code. The options for both zones are identical, and each memory resource can be assigned to either zone. Either zone can protect each sector of flash individually, each LSx memory block individually, User OTP, and secure ROM.

Each zone is secured by its own 128-bit (four 32-bit words) user defined CSM password, which is stored in its dedicated OTP location based on a zone-specific link pointer. The user accessible CSMKEY registers are used to secure and unsecure the device, and a new or un-programmed device is unlocked by default. Since the OTP cannot be erased, flexibility is provided by using a link pointer to select the location of the active zone region within the OTP block, allowing the user to make multiple modifications to the configuration up to thirty times. This is accomplished by exploiting the fact that each bit in the OTP can be programmed one bit at a time, and a “1” can be programmed to a “0”, but not erased back to a “1”. The most significant bit position in the link pointer that is programmed to a “0” defines the valid offset base address for the active zone region within the OTP block. This differs from the F2802x/03x devices where the 128-bit (eight 16-bit words) password is stored in the last eight locations in flash.

7.2.5 OTP

The entire OTP is reserved and unlike the F2802x/F2803x it is not available for user code/data (that is, no boot ROM entry point into the OTP). It consists of two 1K x 16 bit error correction code (ECC) protected sectors. The TI OTP sector is reserved for TI internal use only and it contains device calibration/trim data and settings used by the flash state machine for erase and program operations. The DCSM OTP (also known as User OTP) sector contains the user-configurable locations for security and boot process.

7.2.6 Flash Programming

The flash technology used with the F28E12x devices is different than that of F2802x/03x devices. The F28E12x flash offers faster erase and program operations. It also supports ECC for safety reasons. Since ECC is supported, F28E12x flash API allows users to program in four modes – Fapi_DataOnly, Fapi_AutoEccGeneration, Fapi_DataAndEcc, and Fapi_EccOnly. Also, F28E12x flash allows programming 128-bits at-a-time, whereas F2802x/03x allows programming only 16-bits at-a-time. F28E12x flash API supports all these enhanced features and hence API prototypes are not compatible with that of F2802x/3x. For more details, see the *F28E12x Flash API Reference Guide*.

7.3 Boot ROM

This section highlights the boot ROM enhancements of the F28E12x device. The boot ROM has increased in size to 64K x 16 words on F28E12x from 8K x 16 words on F2802x/03x. The F28E12x boot ROM origin address starts at 0x003F 0000 and contains the contents of Table 7-4.

Table 7-4. Boot ROM Contents

Boot ROM Contents
Zone 1 Secure Flash Boot
Zone 1 CRC Code
ROM Signature
Version
FPU32 Fast Tables
FPU32 Twiddle Tables
Boot Code
Flash API Tables
AES Tables
RTS Lib
IQmath Table
Interrupt Handlers
Vector Table

In order to take advantage of the F28E12x FPU, the boot ROM contains floating-point math tables which are used by the C28x FPU Fast RTS Library, located in C2000Ware at (C2000Ware → libraries → math → FPUfastRTS). For additional information about the boot ROM, see the *F28E12x Microcontrollers Technical Reference Manual*.

7.3.1 Boot ROM Reserved RAM

On the F28E12x devices the boot ROM reserved memory is the first 0x01BE words starting at 0x0002. This section contains the boot status, boot mode, and boot stack. Do not to allocate code or data to these memory locations until bootloading is complete.

7.3.2 Boot Mode Selection

The F28E12x device is extremely flexible in its ability to use alternate, reduce, or completely eliminate boot mode selection pins by programming a BOOTPIN_CONFIG register, whereas the F2802x/03x boot mode pins are hard-coded and provides limited boot options and flexibility. Table 7-5 compares the two boot mode options which are available for the respective device families.

Table 7-5. Comparison of Boot Mode Options

	F2802x / F2803x		F28E12x	
Boot mode pins	GPIO37 and GPIO34 are boot pins for this device. Boot mode pins cannot be modified.		GPIO24 and GPIO32 are default boot mode pins for this device. Other GPIOs can be configured to use as boot mode pin by configuring Z1/Z2-OTP-BOOTPIN-CONFIG in stand-alone mode and EMU-BOOTPIN-CONFIG in emulation mode.	
Bootloader options	**Stand-alone mode:** OTP_KEY and OTP_BMODE can be configured to select boot modes available below. **Emulation mode:** EMU_KEY and EMU_BMODE can be configured to select boot modes available below.		**Stand-alone mode:** Z1/Z2-OTP-BOOTDEF-LOW and Z1/Z2-OTP-BOOTDEF-HIGH can be configured to select boot modes available below. **Emulation mode:** EMU-BOOTDEF-LOW and EMU-BOOTDEF-HIGH can be configured to select boot modes available below.	
	Boot mode	No. of options available	Boot mode	No. of options available
	Parallel Boot	1	Parallel Boot	2
	SCI Boot	1	SCI Boot	4
	SPI Boot	1	SPI Boot	3
	I2C Boot	1	I2C Boot	3
	RAM Boot	1	RAM Boot	1
	Flash Boot	1	Flash Boot	2
	Secure Flash Boot	Not Available	Secure Flash Boot	2
	OTP Boot	1	OTP Boot	Not Available
	Wait Boot	1	Wait Boot	2

For additional information about the boot mode selection, see the *F28E12x Microcontrollers Technical Reference Manual*.

7.3.3 Bootloaders

The F28E12x allows for flexible GPIO assignment selection when using a peripheral bootloader, such as SCI, SPI, or I2C. Unlike F2802x/03x, the F28E12x does not support the OTP boot mode, but does support secure flash boot.

8 Architectural Enhancements

The F28E12x devices include many new architectural enhancements. This section briefly describes the architectural changes from F2802x/F2803x to F28E12x devices. For more information, see the *F28E12x Microcontrollers Technical Reference Manual*.

8.1 Clock Sources and Domains

There are numerous enhancement and changes to the clock sources and additional clock domains on the F28E12x device. These major enhancements and changes include:

- Increase in the number of Peripheral Clock Gating Register to handle the additional and new peripherals
- SYSOSC is the primary internal clock source and is the default system clock at reset
- WROSC is a backup clock source which normally only clocks the watchdog timers and missing clock detection circuit (MCD)
- External Clock Source (XTAL) can be used as the main system; frequency limits and timing requirements can be found in the *TMS320F28E12x Microcontrollers Data Sheet*

- External Clock Output (XCLKOUT) can be connected to either GPIO16 or GPIO18, and the available clock sources are PLLSYSCLK, PLLRAWCLK, SYSCLK, SYSOSC, WROSC, and XTAL
- The System PLL features a feedback loop including:
  - SYSPLL Reference Divider (PDIV) can be /1, /2, /4, or /8
  - SYSPLL Integer Multiplier (QDIV) can be from /2 to /128
  - SYSPLL Output Divider (RDIVCLK0) can be even values from /2 to /32
- PLLSYSCLK Divide Select (PLLSYSCLKDIV) can be from /1 to /64
- XCLKOUT Divide Select (XCLKOUTDIVSEL) has /8 (default on reset) in addition to /1, /2, and /4

8.2 Dual-Clock Comparator (DCC) Module

The F28E12x has a dual-clock comparator (DCC) module. The DCC is used for evaluating and monitoring the clock input based on a second clock, which can be a more accurate and reliable version. This instrumentation is used to detect faults in clock source or clock structures, thereby enhancing the system's safety metrics.

8.3 Watchdog Timer

The F28E12x watchdog timer includes a WDCLK divider in addition to the watchdog prescaler and an optional "windowing" feature that is used to set a minimum delay between counter resets. Utilizing the WDCLK divider, the WDCLK derived from INTOSC1 can be divided by 2 to 4096 in powers of 2. This, along with the watchdog prescaler, provides a very wide range of timeout values for safety-critical applications. The WDCLK divider defaults to divide by 512 for backwards compatibility. On the F2802x/03x devices the WDCLK divider was fixed at divide by 512. The minimum window check feature complement the timeout mechanism in helping protect against error conditions that bypass large parts of the normal program flow but still include watchdog handling. A WDWCR register contains the desired minimum watchdog count. Any attempt to service the watchdog when WDCNTR is less than WDWCR will trigger a watchdog interrupt or reset. When WDCNTR is greater than or equal to WDWCR, the watchdog can be serviced normally. At reset, the window minimum is zero, which disables the windowing feature.

8.4 Peripheral Interrupt Expansion (PIE)

The F28E12x PIE module multiplexes up to four peripheral interrupts into each of the twelve CPU interrupt group lines, further expanding support for up to 48 peripheral interrupt signals. The interrupt vector table addressing is effectively split into two tables, where peripheral group interrupts 1 to 4ranges from 0x0D40 to 0x0D7F. This provides backwards compatibility for the lower range peripheral interrupt vector addresses. The PIE vector table has been updated to accommodate the interrupts issued by the additional peripherals. By comparison, the F2802x/03x multiplexes up to eight peripheral interrupts into each of the twelve groups for up to 96 peripheral interrupt signals.

8.5 Lock Protection Registers

The F28E12x devices utilize “LOCK” protection with several configuration registers to protect from spurious CPU writes. Once these associated LOCK register bits are set the respective locked registers can no longer be modified by software. Table 8-1 is a summary of the available LOCK registers. For more information, see the *F28E12x Microcontrollers Technical Reference Manual*.

Table 8-1. LOCK Registers

CLKCFGLOCK1	CPUSYSLOCK2	SYNCSOCLOCK	RX_LOCK_CTRL
CPUSYSLOCK1	DMACHSRCSELLOCK	INPUTSELECTLOCK	TX_LOCK_CTRL

8.6 General-Purpose Input/Output (GPIO)

The F28E12x allows up to twelve independent peripheral signals to be multiplexed on a single GPIO-enabled pin in addition to the CPU-controlled I/O capability. Each pin can be controlled by either a peripheral or the CPU. The analog signals on this device are multiplexed with digital inputs. These analog I/O (AIO) pins do not have digital output capability and they are assigned to a single port: Port H consists of GPIO224-GPIO247. For more information, see the *TMS320F28E12x Microcontrollers Technical Reference Manual*.

Note

GPIO18/X2 and GPIO19/X1 have different timings due to the load placed on them by the oscillator circuit. For information on using GPIO18/X2 and GPIO19/X1 as GPIOs, see the TMS320F28E12x Microcontrollers Data Sheet.

8.7 External Interrupts

The F28E12x has five external interrupts (two more than the F2802x/03x). Each external interrupt (XINT1-5) can be mapped to any GPIO pin via the Input X-BAR. Like the F2802x/03x, XINT1-3 has a free-running 16-bit counter which can measure the elapsed time between interrupts.

8.8 Crossbar (X-BAR)

The X-BARs provide a flexible means for interconnecting multiple inputs, outputs, and internal resources in various configurations. The F28E12x device contains three X-BARs: the Input X-BAR, the Output X-BAR, and the PWM X-BAR.

- Input X-BAR – is used to route external GPIO signals into the device. It has access to every GPIO pin where each signal can be routed to any or multiple destinations which include the ADCs, eCAPs, PWMs, Output X-BAR, and external interrupts. The F28E12x Input X-BAR has sixteen inputs (INPUT1 through INPUT16) and any of the sixteen inputs can be selected as an external input to each of the eCAP modules.

  Note

  This differs from the F2802x/03x devices which uses the GPIO multiplexer to select a specific dedicated input pin to access the eCAP module.

- Output X-BAR – is used to route various internal signals out of the device. It contains eight outputs that are routed to the GPIO structure, where each output has one or multiple assigned pin positions, which are labeled as OUTPUTXBARx. Additionally, the Output X-BAR can select a single signal or logically OR up to 32 signals.
- PWM X-BAR – is used to route signals to the MCPWM Digital Compare submodules of each MCPWM module for actions such as trip zones and synchronizing. It contains eight outputs that are routed as TZx signals to each MCPWM module. Likewise, the MCPWM X-Bar can select a single signal or logically OR up to 32 signals.

9 Peripherals

New peripherals have been added and most of the existing peripherals have been updated. The Control Law Accelerator is not available on the F28E12x devices. This section briefly describes the additions and changes from F2802x/F2803x to F28E12x devices. For an overview of the available peripherals, see the *C2000 Real-Time Control Peripherals Reference Guide*.

9.1 New Peripherals

The F28E12x devices include new peripherals that are not available on the F2802x/F2803x devices. For more information, see the *F28E12x Microcontrollers Technical Reference Manual*.

9.1.1 Direct Memory Access (DMA)

The direct memory access (DMA) module is an event-based machine that provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, effectively freeing up the CPU for other functions. Using the DMA is ideal when an application requires a significant amount of time spent moving large amounts of data. Additionally, the DMA is capable of orthogonally rearranging the data as it is transferred into blocks, “ping-pong” buffering, and binning for optimal CPU processing.

A DMA transfer is started by a peripheral or software trigger. There are six independent DMA channels, where each channel can be configured individually and each DMA channel has its own unique PIE interrupt for CPU servicing. All six DMA channels operate the same way, except channel 1 can be configured at a higher priority over the other five channels. At its most basic level the DMA is a state machine consisting of two nested loops and tightly coupled address control logic which gives the DMA the capability to rearrange the blocks of data during the transfer for post processing. Major features of the DMA are:

- Six channels with independent PIE interrupts
- Each DMA channel can be triggered from multiple peripheral trigger sources independently
- Word Size: 16-bit or 32-bit (SPI limited to 16-bit)
- Throughput: 3 cycles/word without arbitration

9.1.2 Analog Subsystem Interconnect

The F28E12x utilizes an analog subsystem interconnect which enables a very flexible pin usage, allowing for smaller device packages. The CMPSS inputs and digital inputs are multiplexed with the ADC inputs. This type of interconnect permits a single pin to route a signal to multiple analog modules. The analog pins are organized into analog groups around a CMPSS module, and the routing is defined in an analog pin and internal connections table.

9.1.3 Comparator Subsystem (CMPSS)

The F28E12x CMPSS provides substantial enhancements over the simple analog comparator with the integrated 10-bit reference digital-to-analog (DAC) circuits found on the F2802x/03x devices. There are four CMPSS modules available on the F28E12x devices. Each module contains two comparators, two reference 12-bit DACs, and two digital filters. The CMPSS modules are useful for voltage trip monitoring, which are used in applications such as switched-mode power control and power factor correction. The module is designed around a pair of analog comparators which generates a digital output indicating if the voltage on the positive input is greater than the voltage on the negative input. The positive input to the comparator is always driven from an external pin. The negative input can be driven by either an external pin or an internal programmable 12-bit DAC as a reference voltage. Values written to the DAC can take effect immediately or be synchronized with ePWM events. Each comparator output is fed through a programmable digital filter to prevent electrical switching noise from causing spurious trip signals. The output of the CMPSS generates trip signals to the MCPWM event trigger submodule and GPIO structure. Additionally, the CMPSS features PWM blanking capability to clear-and-reset existing or imminent trip conditions near the MCPWM cycle boundaries. Also, by using the analog subsystem interconnect scheme the CMPSS comparator positive and negative input signals are independently selectable. It is noteworthy to mention that the output of DACL in CMPSS1 module can be brought out on the pin.

9.1.4 Programmable Gain Amplifier (PGA)

The F28E12x adds one Programmable Gain Amplifier (PGA) inline with the ADC. Supporting unity gain and gains by a factor of 2 from 2 to 64, the PGA can be used to amplify small signal sources to take advantage of the full dynamic range of the on-chip ADC. Post gain filtering is also supported. Please consult the F28E12x documentation for the full feature set supported.

9.2 Control Peripherals

This section describes the changes and additions to the F28E12x device control peripherals. For more information, see the following reference guides:

- *F28E12x Microcontrollers Technical Reference Manual*
- *C2000 Real-Time Control Peripherals Reference Guide*

9.2.1 Enhanced Pulse Width Modulator (MCPWM)

The MCPWM module is a reduced version of the type-4 EPWM module featured on other C2000 devices.

MCPWM has the following enhancements/removals compared to type 4 EPWM:

- **Memory Mapped Active/Shadow Registers**: Additional registers have been added to view the contents of the active register and the shadow register separately for CMPx, TBPRD, DBRED, DBFED, AQCTLA, and AQCTLB.

- **6 Channels per MCPWM module**: In contrast to EPWM, the MCPWM module can feature up to 6 channels on a single module. The 6 channels are treated as 3 pairs of signals, similar to 3 EPWM modules; however, some settings are shared across all 3 channel pairs such as TBPRD, DBRED, DBFED, and TBPHS. This can reduce design flexibility compared to 3 separate EPWM modules.
- **Removed Submodules/feature**: When compared to type-4 EPWM, the following features/submodules have been removed on MCPWM:
  - HRPWM
  - Separate interrupts for TZ events
  - Down-Count mode
  - Digital Compare submodule
  - Chopper module
  - EPWMXLINK
  - T1/T2 action qualifier events
  - Dead-band half-cycle clocking mode Refer to the EPWM to MCPWM Migration Guide for more detail on feature changes and subsequent workarounds when migrating from EPWM to MCPWM.

The MCPWM peripheral is capable of generating complex pulse width waveforms with minimal CPU overhead or intervention. Similar to type-4 EPWM, the MCPWM is split into separate modules each with an independent function. This chapter is divided into individual sections by each MCPWM submodule. For most PWM applications, all modules in MCPWM must be understood and utilized to generate the desired PWM output. In this document, the letters x and y within a signal or submodule name is used to indicate a generic MCPWM instance and channel pair on a device. For example, output signals MCPWMx_yA and MCPWMx_yB refer to the output signals from the MCPWMx instance and y channel pair (note that there are up to 3 channel pairs per MCPWM instance). Thus, MCPWM1_1A and MCPWM1_1B belong to MCPWM1 channel pair 1 and likewise MCPWM2_3A and MCPWM2_3B belong to MCPWM2 channel pair 3.

9.2.2 Enhanced Capture Module (eCAP)

The F28E12x devices have one eCAP module. Enhancements to the eCAP module include:

- Input Multiplexer – selects one of 128 signals, which can be from internal or external sources. Any GPIO pin can be used as an input source via the Input X-BAR. Input source is selected by INPUTSEL bit fields in the ECCTL0 register.

Note

The F2802x/03x devices use the GPIO multiplexer to select a specific dedicated input pin to access the eCAP module.

- Event Filter Reset Bit – CTRFILTRESET bit field in the ECCTL2 register clears the event filter, modulo counter, and any pending interrupts flags. Resetting the bit is useful for initialization and debug.
- Modulo Counter Status Bits – MODCNTRSTS bit fields in the ECCTL2 register indicate which capture register will be loaded next. It was not possible to know current state of modulo counter with the F2802x/03x eCAP modules.
- DMA Trigger Source – CEVT[1-4] can be configured as the source for eCAPxDMA.
- EALLOW Protection – added to critical registers. To maintain software compatibility with F2802x/03x devices, configure DEV_CFG_REGS.ECAPTYPE to make these registers unprotected.
- ECAPxSYNCINSEL Register – added for each eCAP to select an external SYNCIN. Each eCAP module can have a separate SYNCIN signal.

9.2.3 Enhanced Quadrature Encode Pulse Module (eQEP)

The F28E12x devices have one eQEP module. The F2803x devices have one eQEP module and the eQEP module is not available on the F2802x devices. Enhancements to the F28E12x eQEP module include:

- QEP Mode Adapter (QMA) – the QMA evaluates transitions on the external EQEPA and EQEPB signal lines and generates direction and clock signals for supporting industrial drive applications. To use the QMA the eQEP module needs to be configured in the Direction-Count mode.
- Latching Position Counter – on ADCSOCA/ADCSOCB event from ePWM module.
- SinCos – support for decoding signals from SinCos transducers.

9.3 Analog Peripherals

This section describes the changes and additions to the F28E12x device analog peripherals. For more information, see the following reference guides:

- *F28E12x Microcontrollers Technical Reference Manual*
- *C2000 Real-Time Control Peripherals Reference Guide*

9.3.1 Analog-to-Digital Converter (ADC)

Unlike the ADC found on the F2802x/03x where a single ADC has two sample-and-hold (S/H) circuits, the F28E12x utilizes one ADC with a single S/H circuit. This allows the F28E12x to efficiently manage multiple analog signals for enhanced overall system throughput. The ADC module is implemented using a successive approximation type ADC with a resolution of 12 bits and it provides a throughput of 8MSPS.

Like the F2802x/03x, ADC triggering and conversion sequencing is managed by a series of start-of-conversion (SOCx) configuration registers. However, the round robin and high priority modes are implemented on the F28E12x devices. Also, the F28E12x has four flexible PIE interrupts rather than nine found on the F2802x/03x.

To further enhance the capabilities of the F28E12x ADC, the ADC module incorporates three post-processing blocks (PPB), and each PPB can be linked to any of the ADC result registers. The PPBs can be used for hardware oversampling, offset correction, calculating an error from a set-point, detecting a limit and zero-crossing, and capturing a trigger-to-sample delay:

- The oversampling mode allows the application to easily perform multiple back-to-back samples from a single trigger pulse.

  When used in conjunction with the aggregation options in the post-processing block, this mode enables oversampling and averaging.
- Offset correction can simultaneously remove an offset associated with an ADCIN channel that was possibly caused by external sensors or signal sources with zero-overhead, thereby saving processor cycles.
- Error calculation can automatically subtract out a computed error from a set-point or expected result register value, reducing the sample to output latency and software overhead.
- Limit and zero-crossing detection automatically performs a check against a high/low limit or zero-crossing and can generate a trip to the MCPWM and/or generate an interrupt. This lowers the sample to MCPWM latency and reduces software overhead. Also, it can trip the MCPWM based on an out-of-range ADC conversion without any CPU intervention, which is useful for safety conscious applications.
- Sample delay capture records the delay between when the SOCx is triggered and when it begins to be sampled. This can enable software techniques to be used for reducing the delay error.
- The 12-bit ADC modules in this device include a sample capacitor reset feature to help mitigate memory crosstalk.

9.4 Communication Peripherals

This section describes the changes and additions to the F28E12x device communication peripherals. For more information, see the following reference guides:

- *F28E12x Microcontrollers Technical Reference Manual*
- *C2000 Real-Time Control Peripherals Reference Guide*

9.4.1 SPI

The F28E12x SPI includes the following enhancements:

- Transmit and receive FIFOs have increased from 4-levels to 16-levels
- High-speed mode
- Delayed transmit control
- DMA support

Also included is the STEINV inversion bit for digital audio interface receive mode on devices with two SPI modules (which is available on F2803x, but not F2802x). The F28E12x has one SPI module, whereas the F2803x has up to two SPI modules depending on the device package and the F2802x has only one SPI module.

9.4.2 SCI

The F28E12x SCI module remains functionally the same as the F2802x/03x except transmit and receive FIFOs have increased from 4-levels to 16-levels.

9.4.3 UART

The F28E12x has one UART module that performs the functions of parallel-to-serial and serial-to-parallel conversions. The UART module is similar in functionality to SCI, but is not register-compatible. The UART module is not available on the F2802x/F2803x.

9.4.4 I2C

Similar to F2802x/03x devices, the F28E12x has one I2C module. Transmit and receive FIFOs have increased from 4-levels to 16-levels, and the bug related to the timing of the XRDY transmit interrupt has been fixed.

10 Emulation – JTAG Port

The F28E12x devices support two JTAG modes:

- IEEE 1149.1 Standard JTAG mode using TCK, TMS, TDI and TDO pins; note that there is no TRSTn pin on F28E12x devices
- IEEE 1149.7 Compact JTAG (cJTAG) using TCK and TMS pins; in this mode the TMS pin is bi-directional and carries TDI, TMS, and TDO information

For more information, see the *F28E12x Microcontrollers Data Sheet*.

11 Silicon Errata

For the most current known exceptions to the functional specifications (advisories) and usage notes (where the device's behavior may not match presumed or documented behavior), see the following document:

- *F28E12x MCUs Silicon Errata*

12 Device Comparison Summary

In this section, Table 12-1 provides a general feature comparison between the F28E12x, F2803x, and the F2802x device families. For specific details about device features, maximum clock frequency, memory sizes, and peripheral availability and quantity, see the device-specific data sheet.

Table 12-1. Device Comparison Matrix

	F28E12x	F2803x	F2802x
Clock	160MHz	60MHz	60MHz
FPU	Yes	-	-
CLA	-	Yes (Type 0) (1)	-
2-Channel DMA	Yes (Type 0)	-	-
Flash	64K x 16	64K x 16	32K x 16
RAM	8K x 16	10K x 16	6K x 16
32-bit CPU Timers	Yes	Yes	Yes
Watchdog Timer	Yes	Yes	Yes
On-Chip Oscillators	Yes	Yes	Yes
DCC	Yes (Type 1)	-	-
ADC	Yes (Type 7)	Yes (Type 3)	Yes (Type 3)
CMPSS_LITE w/ 2 DACs	Yes (Type 0)	-	-
CMP w/ DAC	-	Yes (Type 0)	Yes (Type 0)
MCPWM	Yes (Type 0)	-	-
ePWM	-	Yes (Type 1)	Yes (Type 1)

Table 12-1. Device Comparison Matrix (continued)

	F28E12x	F2803x	F2802x
eCAP	Yes (Type 2)	Yes (Type 0)	Yes (Type 0)
HRCAP	-	Yes (Type 0)	-
eQEP	Yes (Type 2)	Yes (Type 0)	-
PGA	Yes (Type 3)	-	-
CAN (2)	-	Yes (eCAN)	Yes (eCAN)
UART	Yes (Type 0)	-	-
I2C	Yes (Type 2)	Yes (Type 0)	Yes (Type 0)
SCI	Yes (Type 0)	Yes (Type 0)	Yes (Type 0)
SPI	Yes (Type 2)	Yes (Type 1)	Yes (Type 1)
LIN	-	Yes (Type 0)	-
X-BARs	Yes (Type 0)	-	-

(1) A type change represents a major functional feature difference in a peripheral module. Within a peripheral type, there can be minor differences between devices that do not affect the basic functionality of the module. These device-specific differences are listed in the TRM.
(2) DCAN and eCAN are not software compatible.

For more detailed device information, see the following data sheets:

- *F28E12x Microcontrollers Data Sheet*
- *TMS320F2803x Microcontrollers Data Sheet*
- *TMS320F2802x Microcontrollers Data Sheet*

13 References

- Texas Instruments: *F28E12x Microcontrollers Data Sheet*
- Texas Instruments: *F28E12x Microcontrollers Technical Reference Manual*
- Texas Instruments: *F28E12x MCUs Silicon Errata*
- Texas Instruments: *TMS320F2803x Microcontrollers Data Sheet*
- Texas Instruments: *TMS320F2803x Microcontrollers Technical Reference Manual*
- Texas Instruments: *TMS320F2803x MCUs Silicon Errata*
- Texas Instruments: *TMS320F2802x Microcontrollers Data Sheet*
- Texas Instruments: *TMS320F2802x,TMS320F2802xx Technical Reference Manual*
- Texas Instruments: *TMS320F2802x, TMS320F2802xx MCUs Silicon Errata*
- Texas Instruments: *TMS320C28x CPU and Instruction Set Reference Guide*
- Texas Instruments: *TMS320C28x Extended Instruction Sets Technical Reference Manual*
- Texas Instruments: *C2000 Real-Time Control Peripherals Reference Guide*
- Texas Instruments: *TMS320C28x Assembly Language Tools User's Guide*
- Texas Instruments: *TMS320C28x Optimizing C/C++ Compiler User's Guide*
- Texas Instruments: *controlSUITE™ to C2000Ware Transition Guide*
- http://www.ti.com/tool/C2000WARE
- Texas Instruments: *C28x Embedded Application Binary Interface*
- C2000 Migration from COFF to EABI https://software-dl.ti.com/ccs/esd/documents/C2000_c28x_migration_from_coff_to_eabi.html
- Texas Instruments: *F28E12x Flash API Reference Guide*
- Texas Instruments: *Programming Examples and Debug Strategies for the DCAN Module*

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