Application Report

SPRACQ1–May 2020

Migration Between TMS320F2837x and TMS320F2838x

Vivek Singh

ABSTRACT

This migration guide describes the hardware and software differences to be aware of when moving between the F2837x and F2838x C2000™ MCUs. This document highlights the features that are unique between the two devices for all available packages in a device comparison table. Section 2 discusses hardware considerations when migrating between the F2837x and F2838x devices with the 337-ZWT package. The digital general-purpose input/output (GPIO) comparison tables show pin functionality between the two MCUs. This is a good reference for hardware design and signal routing when considering a move between the two devices. Lastly, the F2838x software support is only in EABI format. The EABI migration is discussed in Section 4.

Contents

List of Figures

List of Tables

Trademarks

C2000, Code Composer Studio are trademarks of Texas Instruments.
Arm, Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
All other trademarks are the property of their respective owners.

1 Feature Differences Between F2837x and F2838x

F2838x is based on the F2837x MCU architecture with many new features including a connectivity manager (CM) subsystem that provides additional MIPS for communication stacks. It is possible to migrate between F2838x and F2837x with the caveats in this document taken into account.

NOTE: This comparison guide focuses on the super-set devices: F28388D and F28379D. Other part numbers in this product family have reduced feature support. For details specific to part numbers, see the device-specific data sheet.

1.1 F2837x and F2838x Feature Comparison

The functional block diagram of F2838x is shown in Figure 1, while the feature comparison of the superset part numbers for theF2837x and F2838x devices are shown in Table 1.

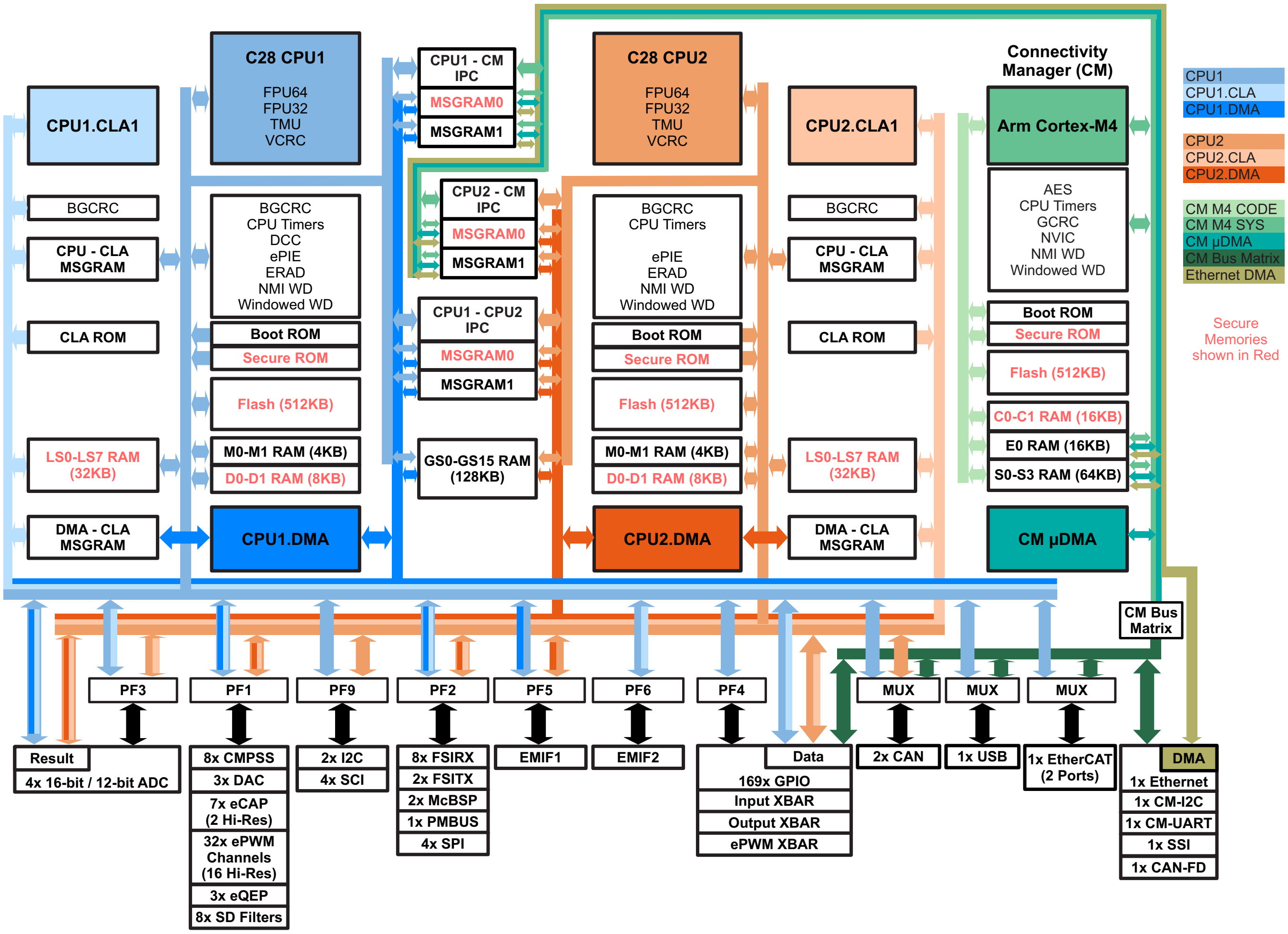

Figure 1. F2838x Functional Block Diagram

Table 1. F28379D and F28388D (Superset) Device Comparison

FEATURE (1)		F28388D	F28379D
C28x Subsystem			
C28x	Number	2	
	Frequency (MHz)	200	
	Floating-Point Unit (FPU)	32 bit and 64 bit	32 bit
	VCRC	Yes	-
	VCU-II	-	Yes
	TMU	Yes – Type 0	
	FINTDIV	Yes	-
CLA	Number	2 (1 per CPU) – Type 2	2 (1 per CPU) – Type 1
	Frequency (MHz)	200	200
C28x Flash		1MB (512KW) [512KB(256KW) per CPU]	1MB (512KW) [512KB (256KW) per CPU]
C28x RAM	Dedicated RAM	24KB (12KW) [12KB (6KW) per CPU]	
	Local Shared RAM	64KB (32KW) [32KB (16KW) per CPU]	48KB (24KW) [24KB (12KW) per CPU]
	Global Shared RAM	128KB (64KW) (Shared between CPUs)	
	Total RAM	**216KB (108KW)**	**200KB (100KW)**
Background Cyclic Redundancy Check (BGCRC) module		4 (2 per CPU/CLA)	-
Configurable Logic Block (CLB)		8 tiles	4 tiles
32-bit CPU timers		6 (3 per CPU)	
6-Channel DMA		2 (1 per CPU) – Type 0	
Dual-zone Code Security Module (DCSM) for on-chip flash and RAM		Yes	
Embedded Real-time Analysis and Diagnostic (ERAD)		Yes	-
Dual Clock Comparator (DCC)		Yes	-
EMIF	EMIF1 (16-bit or 32-bit)	1	
	EMIF2 (16-bit)	1	
External interrupts		5	
GPIO	I/O pins (shared among CPU1, CPU2, and CM)	169	
	Input XBAR	Yes	
	Output XBAR	Yes	
Message RAM	C28x CPU1, C28x CPU2, and Cortex-M4	24KB (4KB each direction between each of the three pairs)	4KB [2KB per CPU]
	C28x CPUs and CLAs	1KB (256 bytes each direction between each CPU and CLA pair)	
	DMAs and CLAs	1KB (256 bytes each direction between each DMA and CLA pair)	-
Nonmaskable Interrupt Watchdog (NMIWD) timers		2 (1 per CPU)	
Watchdog (WD) timers		2 (1 per CPU)	
Connectivity Manager (CM) Subsystem			
Arm® Cortex®-M4		125 MHz	-
M4 Flash		512KB	-
M4 RAM		96KB	-
Advanced Encryption Standard (AES) Accelerator		1	-
CPU timers		3	-
Generic Cyclic Redundancy Check (GCRC) module		1	-
Memory Protection Unit (MPU) for Cortex-M4, µDMA, and Ethernet DMA		3	-
CM Nonmaskable Interrupt (CMNMI) Module		1	-

Table 1. F28379D and F28388D (Superset) Device Comparison (continued)

<table>
<tr><th>FEATURE (1)</th><th></th><th>F28388D</th><th>F28379D</th></tr>
<tr><td colspan="2">Trace Port Interface Unit (TPIU)</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">µDMA</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">Watchdog (WD) timer</td><td>1</td><td>-</td></tr>
<tr><td colspan="4"><b>C28x Analog Peripherals</b></td></tr>
<tr><td colspan="2">Analog-to-Digital Converter (ADC) (configurable to 12-bit or 16-bit)</td><td colspan="2">4</td></tr>
<tr><td rowspan="4">ADC 16-bit mode</td><td>MSPS</td><td colspan="2">1.1</td></tr>
<tr><td>Conversion Time (ns)</td><td colspan="2">915</td></tr>
<tr><td>Input channels (single-ended mode)</td><td>24</td><td>-</td></tr>
<tr><td>Input channels (differential mode)</td><td colspan="2">12</td></tr>
<tr><td rowspan="3">ADC 12-bit mode</td><td>MSPS</td><td colspan="2">3.5</td></tr>
<tr><td>Conversion Time (ns)</td><td colspan="2">280</td></tr>
<tr><td>Input channels (single-ended)</td><td colspan="2">24</td></tr>
<tr><td>Temperature sensor</td><td></td><td colspan="2">1</td></tr>
<tr><td colspan="2">Comparator subsystem (CMPSS) (each CMPSS has two comparators and two internal DACs)</td><td colspan="2">8</td></tr>
<tr><td colspan="2">Buffered Digital-to-Analog Converter (DAC)</td><td colspan="2">3</td></tr>
<tr><td colspan="4"><b>C28x Control Peripherals</b></td></tr>
<tr><td rowspan="2">eCAP/HRCAP</td><td>Total inputs</td><td>7 – Type 1</td><td>6 – Type 0</td></tr>
<tr><td>Channels with high-resolution capability</td><td>2</td><td>-</td></tr>
<tr><td rowspan="2">ePWM/HRPWM</td><td>Total channels</td><td>32 – Type 4</td><td></td></tr>
<tr><td>Channels with high-resolution capability</td><td colspan="2">16</td></tr>
<tr><td>ePWM XBAR</td><td></td><td colspan="2">Yes</td></tr>
<tr><td>eQEP modules</td><td></td><td>3 – Type 2</td><td></td></tr>
<tr><td>SDFM channels</td><td></td><td>8 – Type 2</td><td></td></tr>
<tr><td colspan="4"><b>C28x Communications Peripherals</b></td></tr>
<tr><td colspan="2">Fast Serial Interface (FSI) RX</td><td>8 – Type 1</td><td></td></tr>
<tr><td colspan="2">Fast Serial Interface (FSI) TX</td><td>2 – Type 1</td><td>-</td></tr>
<tr><td colspan="2">Inter-Integrated Circuit (I2C)</td><td colspan="2">2 – Type 0</td></tr>
<tr><td colspan="2">Multichannel Buffered Serial Port (McBSP)</td><td colspan="2">2 – Type 1</td></tr>
<tr><td colspan="2">Power Management Bus (PMBus)</td><td>1 – Type 0</td><td>-</td></tr>
<tr><td colspan="2">Serial Communications Interface (SCI)</td><td colspan="2">4 – Type 0</td></tr>
<tr><td colspan="2">Serial Peripheral Interface (SPI)</td><td>4 – Type 2</td><td>3 – Type 2</td></tr>
<tr><td colspan="2">Controller Area Network (CAN) 2.0B</td><td>2 – Type 0 (can be assigned to CPU1, CPU2, or CM)</td><td>2 – Type 0 (can be assigned to CPU1 or CPU2)</td></tr>
<tr><td colspan="2">Universal Serial Bus (USB)</td><td>1 – Type 0 (can be assigned to CPU1 or CM)</td><td>1 – Type 0 (only on CPU1)</td></tr>
<tr><td colspan="2">uPP</td><td>-</td><td>1</td></tr>
<tr><td colspan="4"><b>Connectivity Manager (CM) Communications Peripherals</b></td></tr>
<tr><td colspan="2">CAN with Flexible Data-Rate (CAN-FD)</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">Ethernet for Control Automation Technology (EtherCAT)</td><td>1 (can be assigned to CPU1 or CM)</td><td>-</td></tr>
<tr><td colspan="2">Ethernet Media Access Controller (EMAC)</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">CM Inter-Integrated Circuit (CM-I2C)</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">Synchronous Serial Interface (SSI)</td><td>1</td><td>-</td></tr>
<tr><td colspan="2">CM Universal Asynchronous Receiver-Transmitter (CM-UART)</td><td>1</td><td>-</td></tr>
<tr><td colspan="4"><b>Package Options</b></td></tr>
<tr><td rowspan="2">Package Options</td><td>337-Ball ZWT</td><td colspan="2">Yes</td></tr>
<tr><td>176-Pin PTP</td><td>future</td><td>Yes</td></tr>
</table>

Table 1. F28379D and F28388D (Superset) Device Comparison (continued)

FEATURE [1]		F28388D	F28379D
	100-Pin PZP	No	Yes

[1] A type change represents a major functional feature difference in a peripheral module. Within a peripheral type, there may be minor differences between devices that do not affect the basic functionality of the module. For more information, see the *C2000 Real-Time Control Peripherals Reference Guide*.

2 PCB Hardware Changes

This section describes considerations to take when switching boards between the F2837x and F2838x devices.

2.1 VDD Pin

The F2838x requires a 56 Ω resistor between VDD and VSS. If a F2837x PCB design needs to be reused for F2838x without PCB modification, it is acceptable to replace a single VDD decoupling capacitor with a 56 Ω resistor.

2.2 VREGENZ Pin

Internal VREG is not supported on the F2838x device. Pin that has VREGENZ functions on F2837x is not internally connected on F2838x. It may be left open or connected to any voltage within the maximum operating conditions.

2.3 Analog Pin Assignment

Analog pin mapping on F2838x is the same as that on F2837x. For analog pin connections, see the guidelines provided in the *TMS320F2838x Microcontrollers With Connectivity Manager Data Manual*.

2.4 GPIO Pin Assignment

The number of GPIO pins are identical on F2837x and F2838x devices. All functions available on each pin on F2837x are also available on F2838x for the modules that are common between two devices. Additional mux options have been provided for new modules as well as for some common modules.

For more information, see Section 3.6.1.

2.5 controlCARD

The F2838x device controlCARD has some changes due to additional modules like EtherCAT and EtherNET.

Table 2 provides details on some key differences between F2837x and the F2838x controlCARD.

Table 2. controlCARD Comparison

Feature	Short Description	F2838x	F2837x
Input Clock	Input clock via XTAL	Rev-A - 20MHz default Rev-B - 25MHz default	20MHz
HD connector	60 pin high density connector support EMIF	No. Removed to enable ECAT and ENET support	Yes
EtherCAT	-	Yes	NA
Ethernet	-	Yes	NA
EMIF support through HSEC	Edge connector's EMIF support	Yes (only 8 bit Data and no WAIT)	Yes
USB	-	Yes	Yes, selection jumper required
VDD and VDDIO	Monitoring locations for VDDIO and VDD	Test point and brought to HSEC for external monitoring	Test Points
ERRORSTS Pin	Monitoring locations for ERRORSTS Pin	Test point and brought to HSEC for external monitoring	Test Point
FSI header		Yes Rev-A - 1 data line support Rev-B - 2 data line support	NA

3 Feature Differences for System Consideration

This section outlines the differences and similarities that exist when migrating between the F2838x and F2837x devices.

3.1 New Features in F2838x Device

This section outlines features that are new on the F2838x device.

3.1.1 Fast Integer Division (FINTDIV)

The C28x processor Fast Integer Division (FINTDIV) unit provides an open and scalable approach to facilitate different data type sizes (16/16, 32/16, 32/32, 64/32, 64/64), signed and unsigned or mixed data type versions (ui32/ui32, i32/ui32, i32/i32). For additional performance, the operations return both the integer and remainder portion of the calculation simultaneously. The division operations are interruptible so as to enable minimum latency for higher priority tasks, a critical requirement for high performance real-time control applications. Unique to this fast integer division unit is support for Truncated, Modulo and Euclidean division formats without any cycle penalty. Each of these formats represents the integer and remainder result in different forms. Below is a brief summary of the various division formats:

- Truncated format is the traditional division performed in C language (/ = integer, % = remainder), however, the integer value is non-linear around zero.
- Modulo division is commonly found when performing division on an Excel worksheet.
- Euclidean format is another format similar to Modulo, the difference is the sign on the remainder value.

Both the Euclidean and Modulo formats are more appropriate for precise control applications because the integer value is linear around the zero point and avoid potential calculation hysteresis. The C28x compiler supports all three division formats for all data types.

3.1.2 VCRC Unit

The VCRC unit extends the capabilities of the C28x CPU by adding additional instructions to support Cyclic Redundancy Check (CRC) or a polynomial code checksum. The VCRC unit on F2838x is a subset of the VCU-II unit on F2837x.

3.1.3 EtherCAT Slave Controller (ESC)

The EtherCAT Slave Controller (ESC) is a new module on the F2838x device. Ethernet for Control Automation Technology (EtherCAT) is an Ethernet-based field bus system that was invented by Beckhoff Automation. EtherCAT is standardized in IEC 61158. All of the slave nodes that are connected to the bus interpret, process, and modify data as data is addressed to them, without having to buffer the frame inside the node. It supports up to 2 MII ports to connect to external PHYs. For more information, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.4 Background CRC (BGCRC)

The Background CRC (BGCRC) is a new module for the F2838x device that can compute the CRC-32 value of a configurable block of memory. It accomplishes this by fetching the specified block of memory during idle cycles (when the CPU, CLA, or DMA is not accessing the memory block). It is an upgrade on the CLAPROMCRC found in the F28004x device to test more memories than just the CLA ROM. For more information on the BGCRC, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.5 Diagnostic Features (PBIST/HWBIST)

The F2837x documents refer to PBIST as the controller that executes configurable memory tests routines as part of the boot up sequence. In F2838x documents and in future C2000 device documents, this module is referred to as memory power on self-test (MPOST). MPOST is enabled as part of the boot up sequence in both the F2837x and F2838x devices. HWBIST is a self-test controller for the CPU for fault coverage in safety applications. HWBIST can be invoked from user application code on both F2838x and F2837x device.

3.1.6 Power Management Bus Module (PMBus)

The Power Management Bus Module (PMBus) is a new module for the F2838x devices, which provides an interface between the microcontroller and is device compliant with the SMI Forum PMBus Specification Part I version 1.0 and Part II version 1.1. PMBus is based on SMBus that uses a similar physical layer to I2C. It is assumed that you are familiar with the PMBus, SMBus, and I2C bus specifications. For more information on the PMBus, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.7 Fast Serial Interface (FSI)

Fast Serial Interface (FSI) is a new module for the F2838x devices that is a serial communication peripheral capable of reliable high-speed communication across isolation devices. Galvanic isolation devices are used in situations where two different electronic circuits, that do not have common power and ground connections, must exchange information. Though isolation devices facilitate these signal communications, they can also introduce a large delay on the signal lines and add skew between the signals. The FSI is designed specifically to ensure reliable high-speed communication for system scenarios that involve communication across isolation barriers without adding components. The FSI consists of independent transmitter (FSITX) and receiver (FSIRX) cores. The FSITX and FSIRX cores are configured and operated independently. For more information on the FSI, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.8 Embedded Real-time Analysis and Diagnostic (ERAD)

The Embedded Real-time Analysis and Diagnostic (ERAD) is a new module for F2838x devices that enhance the debug and system-analysis capabilities of the device. The ERAD module consists of the Enhanced Bus Comparator units and the System Event Counter units. The Enhanced Bus Comparator units are used to generate hardware breakpoints, hardware watch points, and other output events. The System Event Counter units are used to analyze and profile the system.

The key features provided by the ERAD module are listed below:

- Provides eight additional hardware break points, hence total 10 hardware breakpoints are available during debug.
- Monitor data read address buses, data write address buses, data write data bus, and generate RTOSINT
- Generate an event output which can be used by other modules. This is done through monitoring any of the program address buses, Virtual Program Counter (VPC), or the Program Counter of the CPU.
- System Event Counter (SEC) units can count duration between specified memory reads and writes.
- System Event Counter (SEC) units can count system events (such as interrupts) as well as duration between such events.
- System Event Counter (SEC) units can measure maximum amount of time spent in between a pair of events, measured over multiple iterations.
- It has cyclic redundancy check (CRC) units that monitor CPU buses and compute CRC when the self-test code is executed.
- The ERAD module is accessible by the debugger and by the application software. This significantly increases the debug capabilities of many real-time systems.

For more details on the ERAD, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.9 Dual-Clock Comparator (DCC)

The Dual-Clock Comparator (DCC) is a new module for the F2838x devices that 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.

The main features of each of the DCC modules are:

- Allows the application to ensure that a fixed ratio is maintained between frequencies of two clock signals
- Supports the definition of a programmable tolerance window in terms of the number of reference clock cycles
- Supports continuous monitoring without requiring application intervention
- Supports a single-sequence mode for spot measurements
- Allows the selection of a clock source for each of the counters, resulting in several specific use cases

For more details on the DCC, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.1.10 Connectivity Manager (CM)

Similar to F2837x, F2838x supports dual-core C28x architecture. However, it also has a new connectivity manager subsystem. The connectivity manager (CM) subsystem is based on the industry standard 32-bit Arm Cortex-M4 CPU and features a wide variety of communication peripherals as listed below:

- Shared with CPU1 - EtherCAT, USB, DCAN
- Accessible by CM only - EtherNET, MCAN (CAN-FD), UART, SSI, I2C

Targeting performance and flexibility, the connectivity manager is based on 125 MHz Cortex-M4 architecture and provides a variety of integrated memories including its own Flash Bank. The primary goals of the Connectivity Manager (CM) are to:

- Allow easy porting of standard communication software stacks from the Arm eco system
- Provide additional communication MIPS

For details on the Connectivity Manager (CM), see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2 Features Differences/Enhancements in F2838x

3.2.1 System

3.2.1.1 Reset

Reset topology on F2838x is the same as on F2837x. Just like F2837x, CPU1 controls the reset for CPU2 and CM as well. CPU1 system reset (CPU1.SYSRSn) resets CPU2 and CM subsystem as well. SOFTPRESx registers, that have reset control bits for all the peripherals accessible from CPU1 (and CPU2 if shared with CPU2), are accessible from CPU1 only. CM has CMSOFTPRESETx registers to control the reset for all the peripherals accessible from CM.

Below is the list of new enhancements on F2838x:

- A new software configuration register SIMRESET has been added in CPU_SYS_REGS. This register has two bit field to issue software reset:
  - XRSn - Writing to this bit will pull the XRSn pin low for 512 INTOSC1 clock cycles.
  - CPU1RSn - Writing a 1 to this field generates a reset to CPU1.
- A new Reset Cause Clear (RESCCLR) register has been added to clear the status in the Reset Cause (RESC) register.

For more details on Reset, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.1.2 Clocking

3.2.1.2.1 PLL

The PLL blocks of F2837x and F2838x devices are different. Table 3 lists the PLL features for both of these devices for comparison. For more information, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

Table 3. PLL Features

Feature	F2838x	F2837x
VCO Range	220 - 600 MHz	120 - 400 MHz
PLL Raw Clock Range	6 - 400 MHz	120 - 400 MHz
X1 Input Range (PLL enabled)	10 - 25 MHz	2 - 20 MHz
REFCLK Divider	Yes [1..32]	No
PLL Slip Detect	No (use DCC)	Yes
Fractional PLLMULT	No	Yes

Due to the differences in register names and bit fields between the two devices of the PLL module, TI recommends to use the PLL setup function, SysCtrl_setClock() in C2000Ware to ensure proper PLL setting.

3.2.1.2.2 X1CNT

Similar to F2837x, F2838x also has the X1CNT counter that counts the X1 clock pulse. This counter can be used to check that the X1 clock is running before switching the clock source to the X1 clock. But, on F2837x, this counter did not have a clear mechanism to clear the counter to 0 after it reaches to maximum value of 0x3FF. On F2838x, a configuration bit has been added to the X1CNT (X1CNT.CLR) register to clear the counter to 0.

3.2.1.2.3 XCLKOUT

Similar to F2837x, F2838x supports the XCLKOUT feature. On F2838x, the following additional source has been added to XCLKOUT. Because of this, the LKSRCCTL3.XCLKOUTSEL register bit field has been extended to 4 bit instead of 3.

Table 4. XCLKOUT Source Select

XCLKOUTSEL	F2838x	F2837x
000	PLLSYSCLK (default on reset)	PLLSYSCLK (default on reset)
001	SYSPLLCLK	PLLRAWCLK
010	CPU1.SYSCLK	CPU1.SYSCLK
011	CPU2.SYSCLK	CPU2.SYSCLK
100	AUXPLLCLK	AUXPLLRAWCLK
101	INTOSC1	INTOSC1
110	INTOSC2	INTOSC2
111	XTAL OSC (o/p clock)	Reserved
1000	CMCLK	NA
1100	PLLRAWCLK	NA
1101	AUXPLLRAWCLK	NA
others	Reserved	NA

Also on the F2838x devices, XCLKOUT has been connected to one of the inputs of the OUTPUTXBAR switch to provide more flexibility to user.

3.2.1.3 Pie Channel Mapping and Interrupt

Pie channel mapping between F2837x and F2838x is different due to the peripheral module changes between these devices.

Table 6 summarizes the common and unique pie channels on these two devices.

Table 5. Pie Channel Legend

Color	Description
⬜	Pie channel common for both devices
🟥	Pie channel applicable only for F2837x
🟧	F2837x INT has been replaced with new INT on F2838x
🟩	Pie channel applicable only for F2838x

Table 6. PIE Channel Mapping

	INTx.1	INTx.2	INTx.3	INTx.4	INTx.5	INTx.6	INTx.7	INTx.8	INTx.9	INTx.10	INTx.11	INTx.12	INTx.13	INTx.14	INTx.15	INTx.16
INT1.y	ADCA1	ADCB1	ADCC1	XINT1	XINT2	ADCD1	TIMER0	WAKE/ WDINT	I2CA	SYS_ ERR	ECAT SYNC0 (CPU1 only)	ECAT INTn (CPU1 only)	CIPC0	CIPC1	CIPC2	CIPC3
INT2.y	EPWM1_ TZ	EPWM2_ TZ	EPWM3_ TZ	EPWM4_ TZ	EPWM5_ TZ	EPWM6_ TZ	EPWM7_ TZ	EPWM8_ TZ	EPWM9_ TZ	EPWM10_ TZ	EPWM11_ TZ	EPWM12_ TZ	EPWM13_ TZ	EPWM14_ TZ	EPWM15_ TZ	EPWM16_ TZ
INT3.y	EPWM1	EPWM2	EPWM3	EPWM4	EPWM5	EPWM6	EPWM7	EPWM8	EPWM9	EPWM10	EPWM11	EPWM12	EPWM13	EPWM14	EPWM15	EPWM16
INT4.y	ECAP1	ECAP2	ECAP3	ECAP4	ECAP5	ECAP6	ECAP7	- (1)	FSITXA_ INT1	FSITXA_ INT2	FSITXB_ INT1	FSITXB_ INT2	FSIRXA_ INT1	FSIRXA_ INT2	FSIRXB_ INT1	FSIRXB_ INT2
INT5.y	EQEP1	EQEP2	EQEP3	-	CLB1	CLB2	CLB3	CLB4	SDFM1	SDFM2	ECAT RSTINTn (CPU1 only)	ECAT SYNC1 (CPU1 only)	SDFM1 DR1	SDFM1 DR2	SDFM1 DR3	SDFM1 DR4
INT6.y	SPIA_RX	SPIA_TX	SPIB_RX	SPIB_TX	MCBSPA_ RX	MCBSPA_ TX	MCBSPB_ RX	MCBSPB_ TX	SPIC_RX	SPIC_TX	SPID_RX	SPID_TX	SDFM2 DR1	SDFM2 DR2	SDFM2 DR3	SDFM2 DR4
INT7.y	DMA_CH1	DMA_CH2	DMA_CH3	DMA_CH4	DMA_CH5	DMA_CH6	-	-	FSIRXC_ INT1	FSIRXC_ INT2	FSIRXD_ INT1	FSIRXD_ INT2	FSIRXE_ INT1	FSIRXE_ INT2	FSIRXF_ INT1	FSIRXF_ INT2
INT8.y	I2CA	I2CA_ FIFO	I2CB	I2CB_ FIFO	SCIC_RX	SCIC_TX	SCID_RX	SCID_TX	FSIRXG_ INT1	FSIRXG_ INT2	FSIRXH_ INT1	FSIRXH_ INT2	CLB5	CLB6	CLB7	CLB8
INT9.y	SCIA_RX	SCIA_TX	SCIB_RX	SCIB_TX	CANA_0	CANA_1	CANB_0	CANB_1	MCANSS_ INT0 (CPU1 only)	MCANSS_ INT1 (CPU1 only)	MCANSS_ ECC_ CORR_ PUL_INT (CPU1 only)	MCANSS_ WAKE_ AND_TS_ PLS_INT (CPU1 only)	PMBUSA	CM_ STATUS (CPU1 only)	USBA (CPU1 only)	-
INT10.y	ADCA_ EVT	ADCA2	ADCA3	ADCA4	ADCB_ EVT	ADCB2	ADCB3	ADCB4	ADCC_EVT	ADCC2	ADCC3	ADCC4	ADCD_EVT	ADCD2	ADCD3	ADCD4
INT11.y	CLA1_1	CLA1_2	CLA1_3	CLA1_4	CLA1_5	CLA1_6	CLA1_7	CLA1_8	CMTOCPUx IPCINTR0	CMTOCPUx IPCINTR1	CMTOCPUx IPCINTR2	CMTOCPUx IPCINTR3	CMTOCPUx IPCINTR4	CMTOCPUx IPCINTR5	CMTOCPUx IPCINTR6	CMTOCPUx IPCINTR7
INT12.y	XINT3	XINT4	XINT5	MPOST	FMC. DONE	VCU	FPU OVER FLOW	FPU UNDER FLOW	EMIF_ ERROR	ECAP6 INT2	ECAP7 INT2	RAM_ACCE SS_VIOLAT ION	CPUxCRC_ INT	CLA1CRC_ INT	CLA OVER FLOW	CLA UNDER FLOW

(1) Cells marked "-" are Reserved. CPUx is CPU1 for CPU1 PIE and CPU2 for CPU2 PIE.

3.2.1.3.1 SYS_ERR Interrupt

On the F2838x devices, most of the system level error interrupts are combined into one interrupt called SYS_ERR interrupt.

Table 7 shows a list of all the interrupts that are combined into SYS_ERR interrupt.

Table 7. SYS_ERR Interrupt Input

Input	Description	F2838x	F2837x
DCC2	Interrupt from DCC2 module	SYS_ERR Interrupt	NA
DCC1	Interrupt from DCC1 module		NA
DCC0	Interrupt from DCC0 module		NA
RAM_ACC_VIOL	RAM memory access violation interrupt		Mapped on PIE Channel INT12.12
FLASH_CORRECTABLE_ERR	Flash Correctable Error interrupt		Mapped on PIE Channel INT12.11
RAM_CORRECTABLE_ERR	RAM Correctable Error interrupt		Mapped on PIE Channel INT12.10
EMIF_ERR	Error interrupt from EMIF1 or EMIF2 module		Mapped on PIE Channel INT12.9

All of the input to the SYS_ERR interrupt are latched in the SYS_ERR_INT_FLG register if the respective bit in the SYS_ERR_MASK register is cleared. You need to configure the SYS_ERR registers in addition to the configuration done on F2837x in order to trigger the SYS_ERR interrupt from any of these inputs.

3.2.1.4 ERRORSTS Pin

The Following enhancement have been made on the F2838x devices for the ERRORSTS pin logic:

- Polarity of the Error pin has been made configurable (configure ERRORCTL.ERRORPOLSEL). Default polarity is active low, which is opposite of F2837x (active high).
- To enable testing of the Error pin, capability to force and clear the Error pin from software has been provided.
- Additional sources of error have been added to ERRORSTS:
  - CPU1 Watchdog reset.
  - Error on a PIE vector fetch.
  - NMI on CM

3.2.2 Watchdog and NMI Watchdog

Following changes have been made to watchdog and NMI watchdog logic on F2838x device -

- Watchdog module on F2837x device has fixed divider of /512 where as F2838x device has programmable divider (WDCR.WDPRECLKDIV) to divide the input clock from /2 to /4096.
- On both device NMI watchdog get trigger by multiple NMI source and status of all the NMI sources get updated in NMIFLG register. On F2837x device all the status bits in NMIFLG registers get reset by XRSn whereas on F2838x these status bits get reset by SYSRSn.
- Below is the list of new NMI source on F2838x device:
  - CRC_FAIL - CRC fail status from BGCRC module
  - ECATNMIn - NMI from EtherCAT (ESC) module
  - CMNMIWDRSn - NMI when CM NMIWD trigger reset to CM subsystem
  - ERADNMI - NMI from ERAD module

3.2.3 Memory

3.2.3.1 *Internal SRAM/ROM*

Total amount of internal RAMs and ROM has been increased on F2838x compared to F2837x. Also some enhancements have been made in the RAM/ROM controller on F2838x.

Below are some of the enhancements on the F2838x devices:

- All ROM on the F2838x devices are Parity protected and also have test logic to test Parity hardware.
- All LSxRAM on the F2838x devices are ECC protected.
- On the F2837x CPU debugger, access was not allowed to the CLA program RAM. On the F2838x CPU debug, access is allowed to the CLA program RAM.
- On the F2838x CLA-DMA, MSG RAMs have been added to enable CLA to use DMA for data transfer.
- Size of IPC MSG RAMs between all the CPU have been increased (2 block of 2KB).

3.2.3.2 *Flash*

Flash memory logic on F2838x remains the same as on F2837x. Similar to F2837x, every CPU subsystem has 512KB of flash (total 1.5MB flash). The F2837x device family has parts with 1MB flash on CPU1. This option is not available on any parts in the F2838x device family. CPU1 has max 512KB of flash on all the parts.

Table 8 shows the different memory blocks available on the CPU1/CPU2 subsystem of the F2838x devices. The features highlighted in green are new or enhanced from F2837x.

Table 8. F2838x CPU1/CPU2 Subsystem Memory

Memory Block	Size (On each CPU)	CPUx Access	CPUx.DMA Access	CPUx.CLA Access	ECC/Parity	Secure
CPU BOOT ROM	96KB	Y	N	N	PARITY	NO
CPU SECURE ROM	64KB	Y	N	N	PARITY	YES
CLA DATA ROM	8KB	Y	N	Y	PARITY	NO
Flash	512KB	Y	N	N	ECC	YES
USER OTP	2KB	Y	N	N	ECC	YES (only CPU1)
MxRAM	2x2KB	Y	N	N	ECC	NO
DxRAM	2x4KB	Y	N	N	ECC	YES
LSxRAM	8x4KB	Y	N	Y	ECC	YES
GSxRAM (total)	16x8KB	Y	Y	N	PARITY	NO
CPU-TO-CPU MSGRAM0	2x2KB	Y	Y	N	PARITY	YES
CPU-TO-CPU MSGRAM1	2x2KB	Y	Y	N	PARITY	NO
CPU-CLA MSGRAM	2x256B	Y	N	Y	PARITY	NO
CLA-DMA MSGRAM	2x256B	Y	Y	Y	PARITY	NO

3.2.4 Dual Code Security Module (DCSM)

Dual Code Security Module (DCSM) has been enhanced on the F2838x devices to provide some additional feature. One major difference on the F2838x devices is that instead of each CPU subsystem having their own dual-zones (Zone1 and Zone2), there are only two zones and secure resources of all the CPU subsystems are allocated to these two zones. Only the security settings programmed in CPU1 USER OTP and CPU2 and CM USER OTP are available to program your application code or data.

Table 9 shows the new and enhanced security features on the F2838x devices.

Table 9. DCSM Feature comparison

Feature	Short Description	F2838x	F2837x
Dual Zone	Two independent security zone for 3rd party development	Only two zones on device and secure resources of all CPU subsystems are allocated to these two zones only.	CPU1 and CPU2 each have separate two zones
Security Settings	Security Settings are programmed in USER OTP	All security settings are programmed in CPU1 USER OTP only.	CPU1 and CPU2 have security settings in their own USER OTP
Default Password	All 4 x 32 bit password values are 0xFFFF_FFFF	ALL_1 password is invalid password and makes device unlock. TI programs few bits of CSMPSWD1 to 0.	ALL_1 password makes zone un-secure
Number of valid Zone_Select_Block	Zone_Select_Block address is based on link pointer	15	30
JTAGLOCK	Feature to disable the JTAG access on device	YES. Password based JTAGLOCK which user can unlock if needed.	NO
SECURE BOOT	This BOOTMODE authenticate the user code before executing it.	YES	NO
Secure MSG RAM	One MSG RAM block can be allocated to Zone1 or Zone2	YES	NO
CLA Security		CLA is like CPU and CLA registers are secure when executing secure code.	CLA can be made secure by allocating it to Zone1 or Zone2

In addition, the address mapping for different security configurations in CPU1 USER OTP on F2838x has changed. You have to make the appropriate changes to code while importing it from F2837x.

3.2.5 ROM Code and Peripheral Booting

Both F2837x and F2838x have a boot-ROM that initializes the device upon a reset and then boots to the application based on the boot-mode settings. Many enhancements and new features have been added to the F2838x boot-ROM. However, the default options have been retained between the two devices so that migration between the devices has minimal impact from a device-boot perspective.

Table 10 provides the list of items available in different section of ROM of the F2838x devices.

Table 10. F2838x ROM contents

ROM	CPU1	CPU2	CM
UNSECURE	• Bootloaders • IQmath • FPU32/FPU64 Math and FFT Tables • AES Tables	• IQmath • FPU32/FPU64 Math and FFT Tables • AES Tables	---
SECURE	• Secure Copy Code • Secure CRC • Secure boot to flash	• Secure Copy Code • Secure CRC • Secure boot to flash •	• Secure Copy Code • Secure CRC • Secure boot to flash •
CLA DATAROM	• Math and FFT Tables	• Math and FFT Tables	NA

Table 11 provides a quick comparison of the BOOTMODE option on F2837x vs F2838x.

Table 11. Boot-ROM Comparison

BOOT Feature	F2838x	F2837x
Default BOOT Pins	GPIO72 and GPIO84	GPIO72 and GPIO84
Number of BOOTMODE pins	Can be customized to use 0 to 3 boot mode select GPIOs (default 2 BOOTMODE pins)	Requires 2 boot mode select GPIOs
Zero BOOTMODE Pin option	YES	NO
Custom BOOTMODE	Up to 8 custom boot mode options can be set in OTP	1 custom boot mode option can be set in OTP
Flash entry point	4 flash entry addresses	1 flash entry address
Reset	CPU2 and CM reset are not released to boot until done so by CPU1 application via IPC	CPU2 is released out of reset to boot during CPU1 boot
BOOT Support on CPU2 (and CM)	Only CPU1 contains bootloaders in ROM	YES, CPU1 and CPU2 both contain bootloader in ROM
BOOTMODE setting via USER OTP	YES, Z1 and Z2 used for boot OTP config, where Z2 has priority (Z2 is checked before Z1)	YES, Z1 and Z2 used for boot OTP config, where Z1 has priority (Z1 checked before Z2)
Wait BOOT	• Wait BOOT is replaced with CAN BOOT • SCI BOOT can be used as Wait BOOT	YES, dedicated Wait BOOT mode option.
Full IPC library in boot-ROM	NO	YES
RAM Initialization (Clear all RAMs to 0x0)	RAM initialization occurs on POR	RAM initialization occurs on POR and XRS
MPOST Support	YES	NO
SECURE BOOT	YES, all three subsystem (CPU1/CPU2 and CM) supports SECURE BOOT	NO

For more details about ROM Code and Peripheral Booting, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.6 External Memory Interface (EMIF)

External Memory Interface (EMIF) module on F2838x and F2837x are identical. The only change on F2838x is the addition of optional re-mapping (dual mapping) of the SDRAM (CS0) space for EMIF1 in the lower 22 bits address range of memory. Default mapping of the SDRAM space of EMIF1 is the same as on F2837x, which is at address 0x8000 0000. This address is beyond the 22-bit address space. Therefore, execution as well as direct addressing is not allowed. You can re-map this to address 0x0020 0000 by writing TYPE bit-field in MEMTYPE register to "01". Also on F2838x, the EMIF_ERR (or EMIF_ERROR) interrupt is combined with other system interrupts on SYS_ERR.

3.2.7 Communication Modules

The communication module changes between F2837x and F2838x only affect the number of peripherals or accessability from the CM subsystem, which was specifically added on F2838x for communication. Module functionality is maintained for both devices. Table 1 shows the number of instance of each communication module on F2838x and F2837x that should be considered when migrating applications between F2837x and F2838x.

3.2.8 Control Modules

There are changes in the control modules between F2837x and F2838x. Most of the control modules on F2838x are of new "Type" compared to F2837x. Also, the number of instance (or channels) have been increased on the F2838x devices. Table 1 shows the module instance differences and module Type differences that should be considered when migrating applications between F2837x and F2838x.

3.2.8.1 Enhanced Pulse Width Modulator (ePWM) and ePWM Sync Scheme

The ePWM Module Type on the F2838x and F2837x devices is the same, which means that the feature set of the ePWM module is the same on both the devices, but the number ePWM channels has been increased and a new synch scheme has been implemented on the F2838x devices.

Table 12 provide the list of changes associated with the ePWM module between the F2837x and F2838x devices.

Table 12. ePWM Feature Comparison

Feature	F2838x	F2837x
EPWM Type	Type 4	Type 4
Number of EPWM Channels	32	24
Number of EPWM channel with High-resolution capability (HRPWM)	16	16
MAX PWM INPUT CLK Frq	200 MHz	100 MHz
HRPWM Clocking	HRPWM is clocked with their respective EPWM input clock	All the HRPWM is always clocked by EPWM1 input clock
SYNC Scheme	Generic Any-To-Any Scheme	Daisy Chain
Configuration register for SYNC Select	EPWMSYNCINSEL	SYNCSELECT

In the new SYNC scheme, any ePWM SYNCOUT or ECAP can be selected as SYNC input for any ePWM module. There are no restrictions like those on F2837x. The new SYNC scheme breaks the code compatibility with F2837x, therefore, you need to modify the respective code while migrating between the F2837x and F2838x devices.

For more details on the ePWM module and new SYNC scheme, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.8.2 Enhanced Capture (eCAP)

The eCAP module on F2838x is "Type 2", which has some additional features compared to those on the F2837x devices ("Type 0"). F2838x also have HRCAP features available on some of eCAP input. Sync scheme for eCAP also has been changed on F2838x (same as ePWM Sync)

Table 13 provide the list of changes associated with eCAP module between F2837x and F2838x.

Table 13. eCAP Feature Comparison

Feature	F2838x	F2837x
Type	Type 2	Type 0
Number of eCAP Inputs	7	6
Inputs channels with high-resolution capability (HRCAP)	Yes (2, eCAP6 and eCAP7)	No
Event filter reset bit	Yes (ECCTL2.CTRFILTRESET)	No
Modulo counter (4 bit sequencer) status bits	Yes (ECCTL2.MODCNTRSTS)	No
DMA Event generation	Yes (ECCTL2.DMAEVTSEL)	No
EALLOW protection for registers	Yes	No
Input select for eCAP	Input for each eCAP module can be selected out of 128 input source including 16 outputs of INPUT XBAR. Input selection is done by ECCCTLx.INPUTSEL	Fixed input from INPUT XBAR for each eCAP module
SYNC Scheme	Generic Any-To-Any Scheme	Daisy Chain
Configuration register for SYNC Select	ECAPSYNCINSELECT	SYNCSELECT

Even though there are some enhancements to the existing feature of the eCAP module, like different input selection logic and EALLOW protection for configuration register, care has to be taken to avoid breaking the compatibility with F2837x code.

The Following implementation has been done to avoid this:

- Default value of ECCTLx.INPUTSELECT is 0x7F (127) and that selects the same input source as on F2837x.
- ECAPTYPE.TYPE configuration bit is added to enable EALLOW protection for eCAP registers. By default, EALLOW protection is not enabled. You can set ECAPTYPE.TYPE = 1 to enable EALLOW protection.

The new SYNC scheme breaks the code compatibility with F2837x. Therefore, you need to modify the respective code while migrating between the F2837x and F2838x devices. For more details on the eCAP module and new SYNC scheme, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.8.3 Enhanced Quadrature Encoder Pulse (eQEP)

The eCAP module on F2838x is "Type 2. This has some additional features compared to F2837x, which is "Type 0".

Below is the list of new feature and enhancement in eQEP module F2838x:

- A new function block Quadrature Mode Adapter (QMA) has been added, which modifies the QEPA and QEPB signals (does not resemble the traditional quadrature phase shifted signals) such that the modified signals resemble “directional count mode” signals, hence, the directional count mode of eQEP can be used. By default, QMA logic is bypassed and the EQEPA and EQEPB inputs from the pins go directly into the eQEP module. You can enable the QMA feature by setting the QMACTRL[MODE] register bit to 1.
- Hook-up the enable latching position count on ADCSOCA and ADCSOCB signal.
- Addition of the QEPSRCSEL register to configure the source selection for QEPA/QEPB/QEPI and QEPS to support SinCos transducers.

By default on reset, all of the new features of the eQEP module on the F2838x devices are disabled to make F2837x code compatible. For more details on the eQEP module, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.8.4 Sigma Delta Filter Module (SDFM)

The SDFM module on F2838x has some enhancements, but Mode1/Mode2 and Mode3 are no longer supported.

Table 14 provides the list of changes associated with SDFM module between F2837x and F2838x.

Table 14. SDFM Feature Comparison

Feature	F2838x	F2837x
SDFM pin configuration	ASYNC Option Only	ASYNC Option QUAL Option (3-sample qual)
SDFM Input qualification option	Yes, SDCLK and SDDATA can be synchronized	No
Mode Supported	Mode 0	Mode 0 Mode1 Mode 2 (Not recommended for new design) Mode 3
Single clock source for all filters	SDCLK1 (SD-C1) can be used to clock all 4 filters in SDFM by configuring SDCTLPARMx.SDCLKSEL register bit.	Not supported
SDSYNC event source	Any ePWM can be selected to drive SDSYNC event by configuring SDSYNCx.SDSYNCSEL register bit.	Only ePWM11 and ePWM12 can drive SDSYNC event
Comparator Filter	Comparator Filter outout is memory mapped.	Comparator Filter output is not memory mapped
Data Filter	SDFM saturation issue is fixed	SDFM saturation issue (Errata item)
FIFO Support	Supports 16 x 32 bit FIFO	Not supported
Interrupt	Each SDFM has 5 interrupt lines - • Each SDFM DATA Ready event has it's own interrupt line • SDFM Error events have separate interrupt line	Each SDFM has 1 interrupt line Both Data Ready event and SDFM error events share single interrupt line

As listed in Table 14, there are many changes in the SDFM module on the F2838x devices. It is recommended to review the code from the F237x devices and make appropriate changes before using it on the F2838x devices. For more details on the SDFM module, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.2.9 Analog Modules

This section outlines the analog differences between the F2838x and F2837x devices. Most of the Analog modules on F2838x are exactly same as on F2837x. There are some minor enhancements in digital logic of some of the modules that are listed in Table 15.

Table 15. Analog Module Instances

Module	Feature	F2838x	F2837x
ADC	Programmable early interrupt	Can be configured via ADCINTCYCLE.OFFSET register	Fixed interrupt time.
ADC	Post Processing Block Cycle By Cycle Enable	Supported. This feature can be enable by setting ADCPPB1CONFIG.CBCEN register. This feature automatically clears the ADCEVTSTAT on a conversion if the event condition is no longer present.	Not supported
ADC	16-bit Single Ended mode	Supported	Not supported
DAC		No Change	
CMPSS	Blanking window	Yes, supported.	No
CMPSS	RAMPSTS Register	PWMSYNC takes precedence over COMPHSTS when both occur simultaneously.	COMPHSTS takes precedence over PWMSYNC when both occur simultaneously
CMPSS	LATCHCLR Signal	LATCHCLR signal goes to the sync block, filter and latch	LATCHCLR signal only goes to the latch

For more details on the Analog module, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.3 Other Device Changes

This section describes feature differences between the F2837x and F2838x devices that were not covered in the previous sections. The changes identified in the following sections must be considered when migrating applications between devices.

3.3.1 Bus Architecture

Basic Bus Architecture in the F2838x and F2837x devices is the same. As shown in Figure 1, F2838x has some additional modules along with the Connectivity Manager (CM) subsystem. Like F2837x, most of the modules available on the CPU1/CPU2 subsystem are shared between both the CPU with a few exceptions like USB and EtherCAT modules, which are not accessible from CPU2. There are communication modules that are accessible only by CM. Some communication modules like USB, EtherCAT and DCAN that are accessible by CPU1 (or CPU1 and CPU2) can also be accessed by CM.

Below are some key points about this architecture:

- CPU1 is master CPU and controls CPU1 application software controls the reset for CPU2 and CM as well as all the shared resources.
- All of the shared resources and modules are assigned to CPU1 after reset and CPU1 application software needs to re-allocate the assignment to CPU2 or CM (only modules), if needed.
- CPU1 application software can allocated shared peripheral to CPU2 by configuring CPUSELx register and to CM by configuring PALLOCATE0 register
- GPIO pinmux assignment can be done by CPU1 application software only. CPU2 and CM do not have access to it.
- ADC Result registers are always accessible by all of the masters except the CM subsystem without any arbitration.
- Clock configuration for all of the peripherals is controlled by the CPU, which has the master ownership based on CPUSELx or PALLOCATE0 register configuration.

3.3.1.1 CLA and DMA Access

On F2837x, CLA/DMA were known as secondary master for the peripherals allocated to their CPU. You can assign access to only one of them by configuring the SECMSEL register bits and the selection applicable to all of the peripherals connected to the specific peripheral frame. This had many limitations; therefore, on F2838x, simultaneous access from all the masters (CPU and their respective CLA and DMA) has been enabled instead of selecting between CLA and DMA. There is no use of the SECMSEL register on the F2838x devices. Also, the access protection feature has been provided on F2838x. You can configure the peripheral specific access control register (SPIA_AC register) to disable access from any of the masters.

Table 16. Access Control Register

Field Name	Reset Value	Definition
CPUx_ACC	0x3	0x3 : No protection. RD/WR access are allowed. 0x2 : No Write access, only RD access to CPU1 (or CPU2). Read in this case will not change any status bit e.g. FIFO empty, RD pointers etc. 0x1 : Reserved 0x0 : No RD/WR access to CPU1 (or CPU2).
CLA1_ACC	0x3	0x3 : No protection. RD/WR access are allowed. 0x2 : No Write access, only RD access to CPUx.CLA1. Read in this case will not change any status bit e.g. FIFO empty, RD pointers etc. 0x1 : Reserved 0x0 : No RD/WR access to CPUx.CLA1
DMA1_ACC	0x3	0x3 : No protection. RD/WR access are allowed. 0x2 : No Write access, only RD access to CPUx.DMA. Read in this case will not change any status bit e.g. FIFO empty, RD pointers etc. 0x1 : Reserved 0x0 : No RD/WR access to CPUx.DMA.

3.3.2 Control Law Accelerator (CLA)

CLA is an independent, fully-programmable, 32-bit floating-point math processor that brings concurrent control-loop execution to the C28x device family. CLA on the F2838x devices is of "Type 2", whereas F2837x has "Type 1" CLA.

Below is the list of enhancements made in "Type 2" CLA on the F2838x devices:

- "Type 2" CLA supports background task. The background task is enabled by setting the BGEN bit in the MCTLBGRND register. When enabled, Task 8 act as background task and once triggered, runs continuously until you either terminate it or reset the CLA or the device. The background task derives its interrupt vector from the MVECTBGRND register instead of MVECT8.
- On "Type 2" CLA instruction fetch access can be stalled. It was not supported on earlier version of CLA (CLA on F2837x).
- Below is the list of debug related enhancement:
  - Addition of MDEBUGSTOP1 instruction to support true software breakpoints on CLA
  - Addition of two hardware breakpoints as well

Additional trigger source was added to support the new module on F2838x. For more details on the Analog module, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.3.3 Direct Memory Access (DMA)

The DMA module on the F2838x devices is the same as on F2837x. Additional trigger source was added (same as CLA) to support the new module on F2838x. For more details on the Analog module, see the *TMS320F2838x Microcontrollers Technical Reference Manual*.

3.4 Power Management

F2837x and F2838x devices only support dual-rail (3.3 V and 1.2 V) supply except for the F2807x variant of the F2837x device that supports single-rail (3.3 V). This section describes the power management differences and similarities between the two devices.

3.4.1 LDO/VREG

F2838x supports external VREG only (no internal VREG).

3.4.2 POR/BOR

There are no functional changes for the POR and BOR.

3.5 Power Consumption

There is not a significant difference in power consumption between the F2838x and F2837x devices, if the same number of peripherals are being utilized.

3.6 GPIO

GPIO architecture on F2838x is very similar to that on F2837x - except some minor enhancements that are listed below:

- Addition of GPIOxDAT_R (GPIO Data Read) register that shows the value written to the GPIOxDAT register from CPU (or CLA) instead of pin value. In most of the cases, value written is reflected on pin also. Therefore, this new register has the same value as that of the data register. In cases where PIN is not driven with written value, this will help in debug.
- On F2837x, all of the masters who had access to the GPIO data registers have their own copy of the GPIO DATA register. This was creating an issue when the master ownership of the pin was changed from one master to other master. To avoid this on F2838x, only one copy of the data register is available and all masters have access to the same data registers. User code does not need any update due to this change.
- On F2838x GPIO, data registers are reset by CPU1 reset only, whereas, on F2837x, these are reset by respective CPU reset, which has master ownership of the GPIO pin (based on GPxCSELy register configuration). Due to this, on F2838x if a GPIO is assigned to CPU2 and the CPU2 application code drives the GPIO to value '1', when the CPU2 subsystem gets reset by CPU2 WD or NMIWD (or by any others means that only resets CPU2), the GPIO pin continues to drive '1'.

3.6.1 GPIO Multiplexing Diagram

Table 18 outlines the differences and similarities that exist in the GPIO mux between the F2838x and F2837x devices. The legend for this table is Table 17. The main changes highlighted in Table 18 are the absence of SDFM mux positions and the DCDC GPIO support pins from F2838x. The other notable change is the addition of HIC mux positions and the use of X1 as a GPIO pin in the F2838x device if external clock is not used.

Table 17. Mux Legend

Color	Description
	mux function common for both devices
🟥	mux function applicable only for F2837x
🟩	mux function applicable only for F2838x

Table 18. GPIO Mux Table Comparison

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO0	EPWM1A				I2CA_SDA		CM-I2CA_SDA	ESC_GPI0		FSITXA_D0			
GPIO1	EPWM1B		MFSRB		I2CA_SCL		CM-I2CA_SCL	ESC_GPI1		FSITXA_D1			
GPIO2	EPWM2A			OUTPUTXBAR1	I2CB_SDA			ESC_GPI2		FSITXA_CLK			
GPIO3	EPWM2B	OUTPUTXBAR2	MCLKRB	OUTPUTXBAR2	I2CB_SCL			ESC_GPI3		FSIRXA_D0			
GPIO4	EPWM3A			OUTPUTXBAR3	CANA_TX		MCAN_TX	ESC_GPI4		FSIRXA_D1			
GPIO5	EPWM3B	MFSRA	OUTPUTXBAR3		CANA_RX		MCAN_RX	ESC_GPI5		FSIRXA_CLK			
GPIO6	EPWM4A	OUTPUTXBAR4	EXTSYNCOUT	EQEP3_A	CANB_TX			ESC_GPI6		FSITXB_D0			
GPIO7	EPWM4B	MCLKRA	OUTPUTXBAR5	EQEP3_B	CANB_RX			ESC_GPI7		FSITXB_D1			
GPIO8	EPWM5A	CANB_TX	ADCSOCAO	EQEP3_STROBE	SCIA_TX		MCAN_TX	ESC_GPO0		FSITXB_CLK	FSITXA_D1	FSIRXA_D0	
GPIO9	EPWM5B	SCIB_TX	OUTPUTXBAR6	EQEP3_INDEX	SCIA_RX			ESC_GPO1		FSIRXB_D0	FSITXA_D0	FSIRXA_CLK	
GPIO10	EPWM6A	CANB_RX	ADCSOCBO	EQEP1_A	SCIB_TX		MCAN_RX	ESC_GPO2		FSIRXB_D1	FSITXA_CLK	FSIRXA_D1	
GPIO11	EPWM6B	SCIB_RX	OUTPUTXBAR7	EQEP1_B	SCIB_RX			ESC_GPO3		FSIRXB_CLK	FSIRXA_D1	UPP-STRT	
GPIO12	EPWM7A	CANB_TX	MDXB	EQEP1_STROBE	SCIC_TX			ESC_GPO4		FSIRXC_D0	FSIRXA_D0	UPP-ENA	
GPIO13	EPWM7B	CANB_RX	MDRB	EQEP1_INDEX	SCIC_RX			ESC_GPO5		FSIRXC_D1	FSIRXA_CLK	UPP-D7	
GPIO14	EPWM8A	SCIB_TX	MCLKXB		OUTPUTXBAR3			ESC_GPO6		FSIRXC_CLK		UPP-D6	
GPIO15	EPWM8B	SCIB_RX	MFSXB		OUTPUTXBAR4			ESC_GPO7		FSIRXD_D0		UPP-D5	
GPIO16	SPIA_SIMO	CANB_TX	OUTPUTXBAR7	EPWM9A		SD1_D1			SSIA_TX	FSIRXD_D1		UPP-D4	
GPIO17	SPIA_SOMI	CANB_RX	OUTPUTXBAR8	EPWM9B		SD1_C1			SSIA_RX	FSIRXD_CLK		UPP-D3	

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO18	SPIA_CLK	SCIB_TX	CANA_RX	EPWM10A		SD1_D2	MCAN_RX	EMIF1_CS2n	SSIA_CLK	FSIRXE_D0		UPP-D2	
GPIO19	SPIA_STEn	SCIB_RX	CANA_TX	EPWM10B		SD1_C2	MCAN_TX	EMIF1_CS3n	SSIA_FSS	FSIRXE_D1		UPP-D1	
GPIO20	EQEP1_A	MDXA	CANB_TX	EPWM11A		SD1_D3		EMIF1_BA0	TRACE_DATA0	FSIRXE_CLK	SPIC_SIMO	UPP-D0	
GPIO21	EQEP1_B	MDRA	CANB_RX	EPWM11B		SD1_C3		EMIF1_BA1	TRACE_DATA1	FSIRXF_D0	SPIC_SOMI	UPP-CLK	
GPIO22	EQEP1_STROBE	MCLKXA	SCIB_TX	EPWM12A	SPIB_CLK	SD1_D4	MCAN_TX	EMIF1_RAS	TRACE_DATA2	FSIRXF_D1	SPIC_CLK		
GPIO23	EQEP1_INDEX	MFSXA	SCIB_RX	EPWM12B	SPIB_STEn	SD1_C4	MCAN_RX	EMIF1_CAS	TRACE_DATA3	FSIRXF_CLK	SPIC_STEn		
GPIO24	OUTPUTXBAR1	EQEP2_A	MDXB		SPIB_SIMO	SD2_D1	PMBUSA_SCL	EMIF1_DQM0	TRACE_CLK	EPWM13A		FSIRXG_D0	
GPIO25	OUTPUTXBAR2	EQEP2_B	MDRB		SPIB_SOMI	SD2_C1	PMBUSA_SDA	EMIF1_DQM1	TRACE_SWO	EPWM13B	FSITXA_D1	FSIRXG_D1	
GPIO26	OUTPUTXBAR3	EQEP2_INDEX	MCLKXB	OUTPUTXBAR3	SPIB_CLK	SD2_D2	PMBUSA_ALERT	EMIF1_DQM2	ESC_MDIO_CLK	EPWM14A	FSITXA_D0	FSIRXG_CLK	
GPIO27	OUTPUTXBAR4	EQEP2_STROBE	MFSXB	OUTPUTXBAR4	SPIB_STEn	SD2_C2	PMBUSA_CTL	EMIF1_DQM3	ESC_MDIO_DATA	EPWM14B	FSITXA_CLK	FSIRXH_D0	
GPIO28	SCIA_RX	EMIF1_CS4n		OUTPUTXBAR5	EQEP3_A	SD2_D3	EMIF1_CS2n			EPWM15A		FSIRXH_D1	
GPIO29	SCIA_TX	EMIF1_SDCKE		OUTPUTXBAR6	EQEP3_B	SD2_C3	EMIF1_CS3n	ESC_LATCH0	ESC_I2C_SDA	EPWM15B	ESC_SYNC0	FSIRXH_CLK	
GPIO30	CANA_RX	EMIF1_CLK	MCAN_RX	OUTPUTXBAR7	EQEP3_STROBE	SD2_D4	EMIF1_CS4n	ESC_LATCH1	ESC_I2C_SCL	EPWM16A	ESC_SYNC1	SPID_SIMO	
GPIO31	CANA_TX	EMIF1_WEn	MCAN_TX	OUTPUTXBAR8	EQEP3_INDEX	SD2_C4	EMIF1_RNW	I2CA_SDA	CM-I2CA_SDA	EPWM16B		SPID_SOMI	
GPIO32	I2CA_SDA	EMIF1_CS0n	SPIA_SIMO			CLB_OUTPUTXBAR1	EMIF1_OEn	I2CA_SCL	CM-I2CA_SCL			SPID_CLK	
GPIO33	I2CA_SCL	EMIF1_RNW	SPIA_SOMI			CLB_OUTPUTXBAR2	EMIF1_BA0					SPID_STEn	
GPIO34	OUTPUTXBAR1	EMIF1_CS2n	SPIA_CLK		I2CB_SDA	CLB_OUTPUTXBAR3	EMIF1_BA1	ESC_LATCH0	ENET_MII_CRS	SCIA_TX	ESC_SYNC0		
GPIO35	SCIA_RX	EMIF1_CS3n	SPIA_STEn		I2CB_SCL	CLB_OUTPUTXBAR4	EMIF1_A0	ESC_LATCH1	ENET_MII_COL		ESC_SYNC1		
GPIO36	SCIA_TX	EMIF1_WAIT			CANA_RX	CLB_OUTPUTXBAR5	EMIF1_A1	MCAN_RX		SD1_D1			
GPIO37	OUTPUTXBAR2	EMIF1_OEn			CANA_TX	CLB_OUTPUTXBAR6	EMIF1_A2	MCAN_TX		SD1_D2			

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO38		EMIF1_A0		SCIC_TX	CANB_TX	CLB_OUTPUTXBAR7	EMIF1_A3	ENET_MII_RX_DV	ENET_MII_CRS	SD1_D3			
GPIO39		EMIF1_A1		SCIC_RX	CANB_RX	CLB_OUTPUTXBAR8	EMIF1_A4	ENET_MII_RX_ERR	ENET_MII_COL	SD1_D4			
GPIO40		EMIF1_A2			I2CB_SDA				ENET_MII_CRS		ESC_I2C_SDA		
GPIO41		EMIF1_A3			I2CB_SCL			ENET_REVMII_MDIO_RST	ENET_MII_COL		ESC_I2C_SCL		
GPIO42					I2CA_SDA			ENET_MDIO_CLK	UARTA_TX			SCIA_TX	USB0DM
GPIO43					I2CA_SCL			ENET_MDIO_DATA	UARTA_RX			SCIA_RX	USB0DP
GPIO44		EMIF1_A4							ENET_MII_TX_CLK		ESC_TX1_CLK		
GPIO45		EMIF1_A5							ENET_MII_TX_EN		ESC_TX1_ENA		
GPIO46		EMIF1_A6			SCID_RX				ENET_MII_TX_ERR		ESC_MDIO_CLK		
GPIO47		EMIF1_A7			SCID_TX				ENET_PPS0		ESC_MDIO_DATA		
GPIO48	OUTPUTXBAR3	EMIF1_A8			SCIA_TX	SD1_D1			ENET_PPS1		ESC_PHY_CLK		
GPIO49	OUTPUTXBAR4	EMIF1_A9			SCIA_RX	SD1_C1	EMIF1_A5		ENET_MII_RX_CLK	SD2_D1	FSITXA_D0		
GPIO50	EQEP1_A	EMIF1_A10			SPIC_SIMO	SD1_D2	EMIF1_A6		ENET_MII_RX_DV	SD2_D2	FSITXA_D1		
GPIO51	EQEP1_B	EMIF1_A11			SPIC_SOMI	SD1_C2	EMIF1_A7		ENET_MII_RX_ERR	SD2_D3	FSITXA_CLK		
GPIO52	EQEP1_STROBE	EMIF1_A12			SPIC_CLK	SD1_D3	EMIF1_A8		ENET_MII_RX_DATA0	SD2_D4	FSIRXA_D0		
GPIO53	EQEP1_INDEX	EMIF1_D31	EMIF2_D15		SPIC_STEn	SD1_C3	EMIF1_A9		ENET_MII_RX_DATA1	SD1_C1	FSIRXA_D1		
GPIO54	SPIA_SIMO	EMIF1_D30	EMIF2_D14	EQEP2_A	SCIB_TX	SD1_D4	EMIF1_A10		ENET_MII_RX_DATA2	SD1_C2	FSIRXA_CLK	SSIA_TX	
GPIO55	SPIA_SOMI	EMIF1_D29	EMIF2_D13	EQEP2_B	SCIB_RX	SD1_C4	EMIF1_D0		ENET_MII_RX_DATA3	SD1_C3	FSITXB_D0	SSIA_RX	
GPIO56	SPIA_CLK	EMIF1_D28	EMIF2_D12	EQEP2_STROBE	SCIC_TX	SD2_D1	EMIF1_D1	I2CA_SDA	ENET_MII_TX_EN	SD1_C4	FSITXB_CLK	SSIA_CLK	

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO57	SPIA_STEn	EMIF1_D27	EMIF2_D11	EQEP2_INDEX	SCIC_RX	SD2_C1	EMIF1_D2	I2CA_SCL	ENET_MII_TX_ERR		FSITXB_D1	SSIA_FSS	
GPIO58	MCLKRA	EMIF1_D26	EMIF2_D10	OUTPUTXBAR1	SPIB_CLK	SD2_D2	EMIF1_D3	ESC_LED_LINK0_ACTIVE	ENET_MII_TX_CLK	SD2_C2	FSIRXB_D0	SPIA_SIMO	
GPIO59	MFSRA	EMIF1_D25	EMIF2_D9	OUTPUTXBAR2	SPIB_STEn	SD2_C2	EMIF1_D4	ESC_LED_LINK1_ACTIVE	ENET_MII_TX_DATA0	SD2_C3	FSIRXB_D1	SPIA_SOMI	
GPIO60	MCLKRB	EMIF1_D24	EMIF2_D8	OUTPUTXBAR3	SPIB_SIMO	SD2_D3	EMIF1_D5	ESC_LED_ERR	ENET_MII_TX_DATA1	SD2_C4	FSIRXB_CLK	SPIA_CLK	
GPIO61	MFSRB	EMIF1_D23	EMIF2_D7	OUTPUTXBAR4	SPIB_SOMI	SD2_C3	EMIF1_D6	ESC_LED_RUN	ENET_MII_TX_DATA2		CANA_RX	SPIA_STEn	
GPIO62	SCIC_RX	EMIF1_D22	EMIF2_D6	EQEP3_A	CANA_RX	SD2_D4	EMIF1_D7	ESC_LED_STATE_RUN	ENET_MII_TX_DATA3		CANA_TX		
GPIO63	SCIC_TX	EMIF1_D21	EMIF2_D5	EQEP3_B	CANA_TX	SD2_C4	SSIA_TX		ENET_MII_RX_DATA0	SD1_D1	ESC_RX1_DATA0	SPIB_SIMO	
GPIO64		EMIF1_D20	EMIF2_D4	EQEP3_STROBE	SCIA_RX		SSIA_RX	ENET_MII_RX_DV	ENET_MII_RX_DATA1	SD1_C1	ESC_RX1_DATA1	SPIB_SOMI	
GPIO65		EMIF1_D19	EMIF2_D3	EQEP3_INDEX	SCIA_TX		SSIA_CLK	ENET_MII_RX_ERR	ENET_MII_RX_DATA2	SD1_D2	ESC_RX1_DATA2	SPIB_CLK	
GPIO66		EMIF1_D18	EMIF2_D2		I2CB_SDA		SSIA_FSS	ENET_MII_RX_DATA0	ENET_MII_RX_DATA3	SD1_C2	ESC_RX1_DATA3	SPIB_STEn	
GPIO67		EMIF1_D17	EMIF2_D1					ENET_MII_RX_CLK	ENET_REVMII_MDIO_RST	SD1_D3			
GPIO68		EMIF1_D16	EMIF2_D0						ENET_MII_INTR	SD1_C3	ESC_PHY1_LINKSTATUS		
GPIO69		EMIF1_D15			I2CB_SCL			ENET_MII_TX_EN	ENET_MII_RX_CLK	SD1_D4	ESC_RX1_CLK	SPIC_SIMO	
GPIO70		EMIF1_D14		CANA_RX	SCIB_TX		MCAN_RX		ENET_MII_RX_DV	SD1_C4	ESC_RX1_DV	SPIC_SOMI	
GPIO71		EMIF1_D13		CANA_TX	SCIB_RX		MCAN_TX	ENET_MII_RX_DATA0	ENET_MII_RX_ERR		ESC_RX1_ERR	SPIC_CLK	
GPIO72		EMIF1_D12		CANB_TX	SCIC_TX			ENET_MII_RX_DATA1	ENET_MII_TX_DATA3		ESC_TX1_DATA3	SPIC_STEn	
GPIO73		EMIF1_D11	XCLKOUT	CANB_RX	SCIC_RX			ENET_RMII_CLK	ENET_MII_TX_DATA2	SD2_D2	ESC_TX1_DATA2		
GPIO74		EMIF1_D10					MCAN_TX		ENET_MII_TX_DATA1	SD2_C2	ESC_TX1_DATA1		
GPIO75		EMIF1_D9					MCAN_RX		ENET_MII_TX_DATA0	SD2_D3	ESC_TX1_DATA0		
GPIO76		EMIF1_D8			SCID_TX			ENET_MII_RX_ERR		SD2_C3	ESC_PHY_RESETn		

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO77		EMIF1_D7			SCID_RX					SD2_D4	ESC_RX0_CLK		
GPIO78		EMIF1_D6			EQEP2_A					SD2_C4	ESC_RX0_DV		
GPIO79		EMIF1_D5			EQEP2_B					SD2_D1	ESC_RX0_ERR		
GPIO80		EMIF1_D4			EQEP2_STROBE					SD2_C1	ESC_RX0_DATA0		
GPIO81		EMIF1_D3			EQEP2_INDEX						ESC_RX0_DATA1		
GPIO82		EMIF1_D2									ESC_RX0_DATA2		
GPIO83		EMIF1_D1									ESC_RX0_DATA3		
GPIO84				SCIA_TX	MDXB				UARTA_TX		ESC_TX0_ENA	MDXA	
GPIO85		EMIF1_D0		SCIA_RX	MDRB				UARTA_RX		ESC_TX0_CLK	MDRA	
GPIO86		EMIF1_A13	EMIF1_CAS	SCIB_TX	MCLKXB						ESC_PHY0_LINKSTATUS	MCLKXA	
GPIO87		EMIF1_A14	EMIF1_RAS	SCIB_RX	MFSXB		EMIF1_DQM3				ESC_TX0_DATA0	MFSXA	
GPIO88		EMIF1_A15	EMIF1_DQM0				EMIF1_DQM1				ESC_TX0_DATA1		
GPIO89		EMIF1_A16	EMIF1_DQM1		SCIC_TX		EMIF1_CAS				ESC_TX0_DATA2		
GPIO90		EMIF1_A17	EMIF1_DQM2		SCIC_RX		EMIF1_RAS				ESC_TX0_DATA3		
GPIO91		EMIF1_A18	EMIF1_DQM3		I2CA_SDA		EMIF1_DQM2	PMBUSA_SCL	SSIA_TX	FSIRXF_D0	CLB_OUTPUTXBAR1	SPID_SIMO	
GPIO92		EMIF1_A19	EMIF1_BA1		I2CA_SCL		EMIF1_DQM0	PMBUSA_SDA	SSIA_RX	FSIRXF_D1	CLB_OUTPUTXBAR2	SPID_SOMI	
GPIO93			EMIF1_BA0		SCID_TX			PMBUSA_ALERT	SSIA_CLK	FSIRXF_CLK	CLB_OUTPUTXBAR3	SPID_CLK	
GPIO94					SCID_RX		EMIF1_BA1	PMBUSA_CTL	SSIA_FSS	FSIRXG_D0	CLB_OUTPUTXBAR4	SPID_STEn	
GPIO95			EMIF2_A12							FSIRXG_D1	CLB_OUTPUTXBAR5		
GPIO96			EMIF2_DQM1	EQEP1_A						FSIRXG_CLK	CLB_OUTPUTXBAR6		

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO97			EMIF2_DQM0	EQEP1_B						FSIRXH_D0	CLB_OUTPUTXBAR7		
GPIO98			EMIF2_A0	EQEP1_STROBE						FSIRXH_D1	CLB_OUTPUTXBAR8		
GPIO99			EMIF2_A1	EQEP1_INDEX						FSIRXH_CLK			
GPIO100			EMIF2_A2	EQEP2_A	SPIC_SIMO			ESC_GPI0		FSITXA_D0			
GPIO101			EMIF2_A3	EQEP2_B	SPIC_SOMI			ESC_GPI1		FSITXA_D1			
GPIO102			EMIF2_A4	EQEP2_STROBE	SPIC_CLK			ESC_GPI2		FSITXA_CLK			
GPIO103			EMIF2_A5	EQEP2_INDEX	SPIC_STEn			ESC_GPI3		FSIRXA_D0			
GPIO104	I2CA_SDA		EMIF2_A6	EQEP3_A	SCID_TX			ESC_GPI4	CM-I2CA_SDA	FSIRXA_D1			
GPIO105	I2CA_SCL		EMIF2_A7	EQEP3_B	SCID_RX			ESC_GPI5	CM-I2CA_SCL	FSIRXA_CLK	ENET_MDIO_CLK		
GPIO106			EMIF2_A8	EQEP3_STROBE	SCIC_TX			ESC_GPI6		FSITXB_D0	ENET_MDIO_DATA		
GPIO107			EMIF2_A9	EQEP3_INDEX	SCIC_RX			ESC_GPI7		FSITXB_D1	ENET_REVMII_MDIO_RST		
GPIO108			EMIF2_A10					ESC_GPI8		FSITXB_CLK	ENET_MII_INTR		
GPIO109			EMIF2_A11					ESC_GPI9			ENET_MII_CRS		
GPIO110			EMIF2_WAIT					ESC_GPI10		FSIRXB_D0	ENET_MII_COL		
GPIO111			EMIF2_BA0					ESC_GPI11		FSIRXB_D1	ENET_MII_RX_CLK		
GPIO112			EMIF2_BA1					ESC_GPI12		FSIRXB_CLK	ENET_MII_RX_DV		
GPIO113			EMIF2_CAS					ESC_GPI13			ENET_MII_RX_ERR		
GPIO114			EMIF2_RAS					ESC_GPI14			ENET_MII_RX_DATA0		
GPIO115			EMIF2_CS0n	OUTPUTXBAR5				ESC_GPI15		FSIRXC_D0	ENET_MII_RX_DATA1		
GPIO116			EMIF2_CS2n	OUTPUTXBAR6				ESC_GPI16		FSIRXC_D1	ENET_MII_RX_DATA2		
GPIO117			EMIF2_SDCKE					ESC_GPI17		FSIRXC_CLK	ENET_MII_RX_DATA3		

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO118			EMIF2_CLK					ESC_GPI18		FSIRXD_D0	ENET_MII_TX_EN		
GPIO119			EMIF2_RNW					ESC_GPI19		FSIRXD_D1	ENET_MII_TX_ERR		
GPIO120			EMIF2_WEn					ESC_GPI20		FSIRXD_CLK	ENET_MII_TX_CLK	USB0PFLT	
GPIO121			EMIF2_OEn					ESC_GPI21		FSIRXE_D0	ENET_MII_TX_DATA0	USB0EPEN	
GPIO122			EMIF2_D15		SPIC_SIMO	SD1_D1		ESC_GPI22			ENET_MII_TX_DATA1		
GPIO123			EMIF2_D14		SPIC_SOMI	SD1_C1		ESC_GPI23			ENET_MII_TX_DATA2		
GPIO124			EMIF2_D13		SPIC_CLK	SD1_D2		ESC_GPI24			ENET_MII_TX_DATA3		
GPIO125			EMIF2_D12		SPIC_STEn	SD1_C2		ESC_GPI25		FSIRXE_D1	ESC_LATCH0		
GPIO126			EMIF2_D11			SD1_D3		ESC_GPI26		FSIRXE_CLK	ESC_LATCH1		
GPIO127			EMIF2_D10			SD1_C3		ESC_GPI27			ESC_SYNC0		
GPIO128			EMIF2_D9			SD1_D4		ESC_GPI28			ESC_SYNC1		
GPIO129			EMIF2_D8			SD1_C4		ESC_GPI29			ESC_TX1_ENA		
GPIO130			EMIF2_D7			SD2_D1		ESC_GPI30			ESC_TX1_CLK		
GPIO131			EMIF2_D6			SD2_C1		ESC_GPI31			ESC_TX1_DATA0		
GPIO132			EMIF2_D5			SD2_D2		ESC_GPO0			ESC_TX1_DATA1		
GPIO133						SD2_C2							AUXCLKIN
GPIO134			EMIF2_D4			SD2_D3		ESC_GPO1			ESC_TX1_DATA2		
GPIO135			EMIF2_D3		SCIA_TX	SD2_C3		ESC_GPO2			ESC_TX1_DATA3		
GPIO136			EMIF2_D2		SCIA_RX	SD2_D4		ESC_GPO3			ESC_RX1_DV		
GPIO137	EPWM13A		EMIF2_D1		SCIB_TX	SD2_C4		ESC_GPO4			ESC_RX1_CLK		
GPIO138	EPWM13B		EMIF2_D0		SCIB_RX			ESC_GPO5			ESC_RX1_ERR		

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO139	EPWM14A				SCIC_RX			ESC_GPO6			ESC_RX1_DATA0		
GPIO140	EPWM14B				SCIC_TX			ESC_GPO7			ESC_RX1_DATA1		
GPIO141	EPWM15A				SCID_RX			ESC_GPO8			ESC_RX1_DATA2		
GPIO142	EPWM15B				SCID_TX			ESC_GPO9			ESC_RX1_DATA3		
GPIO143	EPWM16A							ESC_GPO10			ESC_LED_LINK0_ACTIVE		
GPIO144	EPWM16B							ESC_GPO11			ESC_LED_LINK1_ACTIVE		
GPIO145	EPWM1A							ESC_GPO12			ESC_LED_ERR		
GPIO146	EPWM1B							ESC_GPO13			ESC_LED_RUN		
GPIO147	EPWM2A							ESC_GPO14			ESC_LED_STATE_RUN		
GPIO148	EPWM2B							ESC_GPO15			ESC_PHY0_LINKSTATUS		
GPIO149	EPWM3A							ESC_GPO16			ESC_PHY1_LINKSTATUS		
GPIO150	EPWM3B							ESC_GPO17			ESC_I2C_SDA		
GPIO151	EPWM4A							ESC_GPO18			ESC_I2C_SCL		
GPIO152	EPWM4B							ESC_GPO19			ESC_MDIO_CLK		
GPIO153	EPWM5A							ESC_GPO20			ESC_MDIO_DATA		
GPIO154	EPWM5B							ESC_GPO21			ESC_PHY_CLK		
GPIO155	EPWM6A							ESC_GPO22			ESC_PHY_RESETn		
GPIO156	EPWM6B							ESC_GPO23			ESC_TX0_ENA		
GPIO157	EPWM7A							ESC_GPO24			ESC_TX0_CLK		
GPIO158	EPWM7B							ESC_GPO25			ESC_TX0_DATA0		
GPIO159	EPWM8A							ESC_GPO26			ESC_TX0_DATA1		

Table 18. GPIO Mux Table Comparison (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO160	EPWM8B							ESC_GPO27			ESC_TX0_DATA2		
GPIO161	EPWM9A							ESC_GPO28			ESC_TX0_DATA3		
GPIO162	EPWM9B							ESC_GPO29			ESC_RX0_DV		
GPIO163	EPWM10A							ESC_GPO30			ESC_RX0_CLK		
GPIO164	EPWM10B							ESC_GPO31			ESC_RX0_ERR		
GPIO165	EPWM11A							MDXA			ESC_RX0_DATA0		
GPIO166	EPWM11B							MDRA			ESC_RX0_DATA1		
GPIO167	EPWM12A							MCLKXA			ESC_RX0_DATA2		
GPIO168	EPWM12B							MFSXA			ESC_RX0_DATA3		

4 Application Code Migration From F2837x to F2838x

The following section describes different software components available for F2838x. For more details on software examples of the new features and changes from the F2837x to F2838x devices, see ***<C2000Ware>\device_support\f2838x\docs\Driverlib_F2837x_to_F2838x_Migration_Guide.pdf***.

4.1 C2000Ware Driverlib Files

Along with the driverlib for C28x core and peripherals, F2838x has a driverlib for the CM subsystem peripherals. This CM driverlib is a new component, but maintains similarity to the C28x driverlib in terms of API definition styles and code organization. Driverlib format is primarily supports for the F2838x software, whereas, the F2837x devices bitfield header file format was primarily support.

4.2 C2000Ware Header Files

Bit-field header files for both the F2838x and F2837x devices are available in C2000Ware under the device_support sub directory. Bit-field header file support on F2838x is very limited compared to the existing module from the F2837x devices that has very limited examples.

4.3 Linker command Files

Linker command files for both the F2838x and F2837x devices are available in C2000Ware under the device_support sub directory. Specific to F2838x, which have to be compiled to the Embedded Application Binary Interface (EABI) format, the section names would also need to conform to the EABI standard. For more details, see Table 19.

4.4 Minimum Compiler Version Requirement

Code Composer Studio™ (CCS) compiler version 18.12.0.LTS supports the new instruction sets on F2838x.

4.5 EABI Support

In the past, F2837x applications have always supported the Common Object File Format (COFF) binary executable output. COFF has several limitations, one of which is that the symbolic debugging information is not capable of supporting C/C++. There is also a limit on the maximum number of sections and length of section names and source files, among other things. COFF is also not an industry standard. For these reasons, C2000 is now migrating to the Embedded Application Binary Interface (EABI) format and F2838x is one of the first devices to support it. EABI and COFF are incompatible and conversion between the two formats is not possible. This section provides summary of COFF and EABI differences and useful links that provide more guidelines in migrating applications from COFF to EABI.

- EABI key differences with 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 which 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 should not 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 will make 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 will make 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 will be 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.

- Sections overview:

  Table 19 summarizes the section names for COFF and EABI. These are compiler-generated sections.

Table 19. 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
Red-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

- Resources:
  For more information regarding EABI and the migration process, see the resources on the links below:
  - COFF to EABI Migration: *C2000 EABI Migration*
  - C28 EABI Specifications: *C28x Embedded Application Binary Interface*

4.5.1 Flash API

The F2838x Flash API is enhanced to return an error when an invalid programming mode is provided for program operation. Also, it is enhanced to check the validity of the input address for different functions. Fapi_getLibraryInfo() in F2838x_C28x_FlashAPI.lib returns the Flash API minor version as 60 (F2837xD Flash API returns 54 as the API minor version). Note that the F2838x Flash API library is compiled for EABI format, whereas, the F2837x Flash API library is compiled for legacy COFF. These features are summarized in Table 20

Table 20. Flash API Differences

Feature	F2838x	F2837x
Library Name	F2838x_C28x_FlashAPI.lib (CPU1/CPU2) F2838x_CM_FlashAPI.lib (CM)	F021_API_F2837x_FPU32.lib
Library Executable Output	EABI	COFF (with future EABI support)
Flash Wait States	Same wait states on both devices	
FlashAPI Minor Version	60	54 (F2837xD), 55 (F2837xS)

For more details, see *TMS320F2838x Flash API Reference Guide*.

4.5.2 NoINIT Struct Fix (linker command)

With EABI, the SECTIONS area of a linker command file has to be modified as shown in the example below in order for the registers or memory areas to not be initialized to a zero value. This is important as failure to make this modification can result to unintended behavior when register bits are forced to zero during start up. By default, EABI initializes registers or memory areas defined in the SECTIONS part of the linker to zero.

Linker modification example:

```
SECTIONS
{
:
Regs1File :> REG1_ADDR, type=NOINIT
Regs2File :> REG2_ADDR, type=NOINIT
:
}
```

4.5.3 Pre-Compiled Libraries

All F2838x libraries supplied by TI will be released as EABI. Future F2838x libraries created by users should be generated and compiled as EABI as well.

5 References

- Texas Instruments: *C28x Embedded Application Binary Interface*
- Texas Instruments: *TMS320F2838x Flash API Reference Guide*
- Texas Instruments: *TMS320F2838x Microcontrollers Technical Reference Manual*
- Texas Instruments: *TMS320F2838x Microcontrollers With Connectivity Manager Data Sheet*

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