Application Report
SPRAAQ8B–July 2009

TMS320x280x to TMS320x2833x or 2823x Migration Overview

Lori Heustess *AEC C2000*

ABSTRACT

This application report describes differences between the Texas Instruments TMS320x280x/2801x/2804x and TMS320x2833x/2823x microcontrollers to assist in application migration. While the main focus of this document is migration from 280x/2801x/2804x to 2833x/2823x, you will also find this document useful if you are considering migrating in the reverse direction. 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 groups in the 28x generation.

Contents

List of Figures

List of Tables

1 Introduction

The TMS320x280x, TMS320x2801x, TMS320x2804x, TMS320x2833x and TMS320x2823x devices are members of the C2000™ MCU platform for use within embedded control applications. The TMS320x2833x and TMS320x2823x features the same enhanced control peripherals available on the TMS230x280x, 2801x and 2804x devices. In addition, the 2833x/2823x features direct memory access (DMA), and an external interface (XINTF). The 2833x devices also feature the C28x plus floating-point unit CPU. These new peripherals enable the firmware engineer to solve challenging control problems effectively.

For purposes of migration, these devices can be thought of in two groups:

- TMS320x280x, TMS320x2801x and TMS320x2804x. This group will be referenced as 280x/2801x/2804x.
- TMS320x2833x and TMS320x2823x. This group will be referenced as 2833x/2823x.

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 data manuals and user guides available on the TI website at http://www.ti.com/. This report does not cover the silicon exceptions or advisories that may be present on each device. Consult the following silicon errata for specific advisories and workarounds:

- *TMS320F280x, TMS320C280x, and TMS320F2801x DSC Silicon Errata* (SPRZ171)
- *TMS320F28044 DSP Silicon Errata* (SPRZ255)
- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 DSC Silicon Errata* (SPRZ272)

NOTE: Always refer to the TMS data manual for information regarding any electrical specifications.

1.1 Abbreviations

The following abbreviations are used in this document:

- 2833x and 2823x: Refers to the TMS320x2833x and TMS320x2823x devices. For example, TMS320F28335, TMS320F28334, TMS320F28332 TMS320F28235, TMS320F28234 and TMS320F28232. For migration purposes these devices are identical, except the 2823x does not include the floating-point unit. As a group these will be referred to as 2833x/2823x.
- 280x: Refers to the TMS320x280x devices. For example, TMS320F2809, TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, TMS320C2801 and TMS320C2802. The individual parts in this group are abbreviated 2809, 2808, 2806, 2802 and 2801.
- 2804x: Refers to the TMS320x2804x devices. For example, TMS320F28044.
- 2801x: Refers to the TMS320x2801x devices. For example, TMS320F28015 and TMS320F28016. Individual parts are abbreviated 28015 and 28016.
- 280x/2801x/2804x: Refers to the group of devices made up of TMS320x280x, TMS320x2801x and TMS320x2804x. For the purpose of migration, these three device families are very similar and can be thought of as one group.

For a full list of devices currently available within the 2833x, 2823x, 280x, 2801x and 2804x family, see the TI website.

2 Central Processing Unit (CPU)

The 2833x devices are the first to include the C28x plus floating point unit CPU (C28x+FPU). C28x+FPU-based controllers have the same 32-bit fixed-point architecture as TI's existing C28x™ DSCs, but also include a single-precision (32-bit) IEEE 754 floating point unit (FPU). It is a very efficient C/C++ engine, enabling you to develop system control software and math algorithms using C/C++.

No changes have been made to the existing:

- C28x Instructions
- C28x Pipeline
- C28x Emulation
- Memory Bus Architecture

New instructions to support floating point operations have been added as an extension to the standard C28x instruction set. This means code written for the C28x fixed-point CPU is 100% compatible with the C28x+FPU. The C28x+FPU latched overflow and underflow (LVF, LUF) flags are connected to the peripheral interrupt expansion (PIE) block. This makes debugging overflow and underflow issues much easier. For an introduction to the C28x+FPU, see the *C28x FPU Primer* (SPRAAN9) and the *TMS320C28x Digital Signal Controller Plus Floating Point Unit* online training from the TI website.

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

- *TMS320C28x DSP CPU and Instruction Set Reference Guide* (SPRU430). This document also applies to the C28x+FPU.
- *TMS320C28x Floating Point Unit and Instruction Set Reference Guide* (SPRUEO2). This is a supplement to SPRU430.

The 2823x devices are identical to the 2833x devices except 2823x does not include the floating-point unit.

3 Development Tools

A new set of header files and peripheral examples are available for the 2833x and 2823x with the same structure as the 280x/2801x/2804x header files. Refer to *C2833x/C2823x C/C++ Header Files and Peripheral Examples* (SPRC530). Since the peripherals on the 2833x and 2823x are identical, the header files are shared by the two devices. In the header file package, the examples have been duplicated with one set enabling native floating point (2833x) and the other group setup to only generate fixed point code (2823x).

The C28x+FPU on the 2833x is supported with a patch to Code Composer Studio™ 3.3 software. Currently the floating-point unit (FPU) registers can be viewed within a Code Composer Studio watch window only. The 2833x GEL files include a function to automatically populate the watch window with the FPU registers. In future updates to Code Composer Studio, the FPU registers will be added to a register window. Current C2000 emulation pods will work with the C28x+FPU.

The compiler, assembler, and linker must be V5.0 or greater to take advantage of the hardware FPU resources on the 2833x. As of this writing, the latest compiler is V5.2.1. Check the Code Composer Studio update advisor for future updates. When building for native floating-point, you must use the correct run-time support library. For example, rts2800_fpu32.lib for C or rts2800_fpu32_eh.lib for C++. These libraries are supplied with the compiler.

NOTE: To get the best native floating-point performance on 2833x for math routines, consider using the C28x FPU Fast RTS Library (SPRC664). The C28x Fast RTS is a collection of optimized floating-point math functions for C programmers of the C28x with floating-point unit. Designers of computationally intensive real-time applications can achieve execution speeds considerably faster than what are currently available without having to rewrite existing code. The functions listed in the features section are specifically optimized for the C28x + FPU controllers.

NOTE: To enable the compiler to generate native FPU instructions, you must tell it that you have a C28x device with a floating point unit. To do this, use the following compiler switches:
-v28 --float_support=fpu32
In Code Composer Studio the fpu32 switch is under the advanced compiler options.

You can not mix code built without the --float_support=fpu32 switch with code built with it. This is because the compiler calling conventions changed for floating point numbers. If you try to mix the two, the linker will issue an error indicating the object files are not compatible. If you receive this error, check that all libraries have been built with the same switch. In particular the runtime support library that comes with the compiler must be correct. When compiling with --float_support=fpu32 use the rts2800_fpu32.lib or rts2800_fpu32_eh.lib.

3.1 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 0.000030518.0.
3. Do the following to take advantage of the on-chip floating point unit:
   - Use Code Composer Studio 3.3 with Service Release 9 or later along with the C28x codegen tools version 5.0.2 or later.
   - Tell the compiler 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 on the Advanced tab of the compiler options.
   - Use the correct run-time support library for native 32-bit floating-point. For C code, this is rts2800_fpu32.lib. For C++ code with exception handling, use rts2800_fpu32_eh.lib.
   - Consider using the *C28x FPU Fast RTS Library* (SPRC664) to get a performance boost from math functions such as sin, cos, div, sqrt, and atan. The Fast RTS library should be linked in before the normal run-time support library.

4 Package and Pinout

The two device groups are neither package nor pin-compatible. Any application being moved from one to the other will require a new board layout to accommodate the changes in pinout and package.

5 Operating Frequency and Power Supply

The 280x/2801x/2804x devices require a 1.8-V core voltage at all operating frequencies with a top operating frequency of 100 MHz. Some of the 280x and 2801x devices are also available in a 60 MHz version. The 2833x/2823x 150 MHz devices require a 1.9-V core voltage and the 100 MHz devices operate at a 1.8-V core voltage. Both device groups require a 3.3-V I/O supply.

NOTE: Always refer to the TMS data manual for information regarding any electrical specifications.

See the following data manuals for the most recent detailed electrical specifications:

- *TMS320F2809, TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, TMS320C2802, TMS320C2801, and TMS320F2801x DSPs Data Manual* (SPRS230)
- *TMS320F28044 Digital Signal Processor Data Manual* (SPRS357)

- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controllers (DSCs) Data Manual* (SPRS439)

6 Power Sequencing

Power sequencing requirements are identical for both device groups. That is, the V_{DDIO} and V_{DD} rail can ramp together.

See the appropriate data manual for each device for details related to power sequencing:

- *TMS320F2809, TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, TMS320C2802, TMS320C2801, and TMS320F2801x DSPs Data Manual* (SPRS230)
- *TMS320F28044 Digital Signal Processor Data Manual* (SPRS357)
- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controllers (DSCs) Data Manual* (SPRS439)

7 Memory Map

The memory maps are similar except for the changes described in this section.

7.1 SARAM

This section highlights the major differences in the SARAM memory subsystem.

- **Increased amount of SARAM**

  On the 280x/2801x/2804x up to 18K x 16 words of SARAM is available. On the 2833x/2823x up to 34K x 16 is available.

- **Maximum SARAM block size 4K x 16**

  The H0 SARAM block at 0x3F A000 was split into two smaller memory blocks: L2 and L3. The maximum size of an SARAM block is now 4K x 16. The smaller memory blocks make it easier to partition code and data. If your code uses multiply and accumulate operations (MAC), for example, you will want to partition the opcode and two operands into three different memory blocks. This allows for maximum efficiency.

- **SARAM blocks are dual-memory mapped**

  Memory blocks L0, L1, L2 and L3 are all dual mapped into both high memory and low memory. On the 280x/2801x/2804x, only L0 and L1 are mirrored. The dual mapping of the memory gives flexibility when partitioning code as required by the application. The memory region in the upper 64K range is required when running 24x compatible code, while the stack pointer (SP) can only access memory in the lower 64K. If the application is not porting 24x code, then either memory map location can be used for either data or code. Keep in mind that the stack pointer can still only access the lower 64K range.

- **Wait states**

  On 280x/2801x/2804x devices, all SARAM blocks are 0 wait state in both program and data space. On the 2833x/2823x devices, blocks L4, L5, L6 and L7 are 0 wait for accesses that use the DMA or data bus and 1 wait for accesses that use the program bus. A program bus access occurs when an instruction opcode is fetched or when program-space indirect or direct addressing is used. Instructions that use program-space addressing to access an operand include MAC, DMAC, QMACL, IMACL, PREAD and PWRITE. For example, MAC uses program-space addressing via the XAR7 register. This instruction sees a decrease in performance if the operand pointed to by XAR7 is in L4-L7. L4-L7 should be allocated for data space accesses before they are used for program code or data accessed via program-space addressing.

- **DMA accessible SARAM**

  The L4-L7 memory blocks can be used as a source and/or destination for each of the 6 DMA channels. DMA accesses to L4-L7 are 0 wait. On the 280x/2801x/2804x, DMA was not available.

Table 1. SARAM Addresses

Memory Address [1]	280x/2801x/2804x Memory Block	2833x/2823x Memory Block
0x00 8000 – 0x00 8FFF	L0	L0
0x00 9000 – 0x00 9FFF	L1	L1
0x00 A000 – 0x00 AFFF	N/A [2]	L2
0x00 B000 – 0x00 BFFF	N/A	L3
0x00 C000 – 0x00 CFFF	N/A	L4 [3]
0x00 D000 – 0x00 DFFF	N/A	L5 [3]
0x00 E000 – 0x00 EFFF	N/A	L6 [3]
0x00 F000 – 0x00 FFFF	N/A	L7 [3]
0x3F 8000 – 0x3F 8FFF	L0 Mirror	L0 Mirror
0x3F 9000 – 0x3F 9FFF	L1 Mirror	L1 Mirror
0x3F A000 – 0x3F AFFF	H0 [4]	L2 Mirror [4]
0x3F B000 – 0x3F BFFF		L3 Mirror [4]

(1) Some SARAM blocks may not be available on some family derivatives. Refer to the device specific data sheets.
(2) N/A = Not available.
(3) DMA accessible memory block. One wait state in program space.
(4) H0 is not protected by the CSM. L2 and L3 are protected by the CSM.

7.2 Flash and OTP

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 as shown in Table 2.

Table 2. Sector Configuration Per Device

	F2809	F2808 F28044	F2806 F2802	F2801 F28016 F28015	F28235 F28335	F28234 F28334	F28232 F28332
	8 Sectors 16K X 16	4 Sectors 16K X 16	4 Sectors 8K X 16	4 Sectors 4K X 16	8 Sectors 32K X 16	8 Sectors 16K X 16	4 Sectors 16K X 16
Total	128K X 16	64K X 16	32K X 16	16K X 16	256K x 16	128K x 16	64K x 16

7.2.2 Flash Access Time

The access time of the Flash has increased by 1ns. Depending on your CPU speed, you may need to update the waitstates accordingly.

NOTE: Always refer to the device-specific data manual timing information.

7.2.3 Entry Point Into Flash and CSM Password Locations

On both device groups, the boot ROM entry point and code security module password locations are located at the highest addresses of sector A. This address has shifted on the 2833x/2823x as the entire Flash array moved; they are still located at the end of sector A. To keep the CSM unlocked during debug, you can open a memory window to the new password locations at 0x33 FFF8 - 0x33 FFFF. If you use a Code Composer Studio GEL file to unlock the CSM, the GEL function needs to be modified with the new addresses. Table 3 in Section 7.3 shows a summary of the boot ROM entry point locations.

7.2.4 Entry Point Into OTP

On both device groups, the boot ROM entry point into the one-time programmable (OTP) is the first address within the OTP. This location has shifted as the memory mapping of the OTP has moved. Table 3 in Section 7.3 shows a summary of the boot ROM entry point locations.

7.2.5 Flash programming

The method for programming the device remains the same. TI supplies a Flash API per device that is used as the basis of all programming solutions. The Flash API and programming algorithms for the F280x/F2801x/F2804x devices cannot be used on the F2833x or F2823x. New Flash APIs are required to program these devices; however, the Flash API function prototypes remain compatible.

NOTE: The Flash API includes timing critical delay loops. These loops should always be run from 0 wait memory in order to be timing accurate. On the 2833x/2823x, the L4-L7 SARAM blocks have 1 wait state in program space so they are not suited for running the Flash API.

7.3 Boot ROM

The boot ROM loaders are similar on both device groups. This section highlights the differences.

7.4 Enhancements

The following enhancements have been made to the boot ROM:

- Added bootloader for the multichannel buffered serial port (McBSP-A) module.
- The SPI-A bootloader has been updated to support both serial 16-bit addressable EEPROMs and 24-bit addressable serial peripheral interface (SPI) Flash.
- A parallel bootloader for the XINTF has been added. This is similar to the general-purpose input/output (GPIO) parallel loader except the XINTF data lines are used to import data.
- The IQMath tables have been updated to include an exponent table.
- Floating-point tables for sin, cos and atan have been added.
- The boot ROM calibrates the analog-to-digital converter (ADC) as described in Section 9.2.

NOTE: The memory locations of the IQmath tables has shifted. Make sure to use the new addresses in your linker command file as shown in the *C2833x/C2823x C/C++ Header Files and Peripheral Examples* (SPRC530).

7.5 Entry Points Into Memory

The entry point into the Flash memory block is still the two words right before the password locations. The address has changed to 0x33 FFF6 to reflect the new memory mapping of the Flash. The entry point into the OTP is still the first word of the OTP block, but the address has changed to reflect the new location of the OTP. Table 3 summarizes the changes to the memory entry point locations.

Table 3. Boot Loader Entry Points

	280x/2801x/2804x Entry Point	2823x/2833x Entry Point	Notes
Jump to Flash	0x3F 7FF6	0x33 FFF6	Two words before the password locations
Jump to SARAM	0x00 0000	0x00 0000	First location in M0
Jump to OTP	0x3D 7800	0x38 0400	First word of the OTP
Jump to XINTF (x16 or x32)	N/A	0x10 0000	XINTF is not available on 280x/2801x/2804x

7.6 Boot-Mode Selection

The boot-mode pin configuration has changed to enable the different bootloaders and is shown in Table 4. You must configure each of the four pins before reset for the bootloader to properly start. The data stream format used is identical to that used on the 280x/2801x/2804x devices.

Table 4. Boot Mode Selection

Mode	GPIO87 XA15	GPIO86 XA14	GPIO85 XA13	GPIO84 XA12	Description
F	1	1	1	1	Jump to Flash
E	1	1	1	0	SCI-A
D	1	1	0	1	SPI-A
C	1	1	0	0	I2C-A
B	1	0	1	1	eCAN-A
A	1	0	1	0	McBSP-A
9	1	0	0	1	Jump to XINTF x16
8	1	0	0	0	Jump to XINTF x32
7	0	1	1	1	Jump to OTP
6	0	1	1	0	Parallel I/O
5	0	1	0	1	Parallel XINTF
4	0	1	0	0	Jump to SARAM
3	0	0	1	1	Branch to check boot mode
2	0	0	1	0	Jump to Flash, skip ADC calibration
1	0	0	0	1	Jump to SARAM, skip ADC calibration
0	0	0	0	0	SCI-A, skip ADC calibration

NOTE: Modes 0, 1 and 2 in Table 4 are for TI debug only. Skipping the ADC calibration function in an application causes the ADC to operate outside of specification. For more detailed information on the ADC_cal() function, see the *TMS320x2833x Analog-to-Digital (ADC) Module Reference Guide* (SPRU812)

See the following reference guides for more information on the boot ROM:

- *TMS320x280x, 2801x, 2804x Boot ROM Reference Guide* (SPRU722)
- *TMS320x2833x, 2823x Boot ROM Reference Guide* (SPRU963)

8 Clocks and System Control

This section describes changes that affect device clocking and system control. This includes new and renamed registers, pin functionality, new logic, and other enhancements. For more information on system control, see the following reference guides:

- *TMS320x280x, 2801x, 2804x DSP System Control and Interrupts Reference Guide* (SPRU712)
- *TMS320x2833x, 2823x System Control and Interrupts Reference Guide* (SPRUFB0)

8.1 Register Changes

Table 5 shows a summary of registers that were added, renamed, or modified. The following sections describe changes that were made. This section does not include the new DMA registers as they can be considered as all new.

Table 5. New, Updated, and Removed Registers

Register	Change	Description
XCLK	Removed	The XCLKOUT divider is now in the XINTF Configuration Register (XINTFCNF2) within the XINTF module.
PLLSTS	Updated	CLKINDIV (single bit) is now DIVSEL (two bits). Refer to Section 8.2.
PCLKCR0	Updated	Updated to support additional SCI and McBSP clock enables
PCLKCR1	Updated	Updated to support additional eCAP clock enables
PCLKCR2	Removed	2804x only: enable/disable clocks to the ePWM modules
PCLKCR3	New	New register to add clock enables for the XINTF, CPU Timers, DMA and input qualification.
GPAMCFG	Removed	2804x only: Selects the pinout configuration for the ePWM
MAPCNF	New	Selects the register mapping for the ePWM/HRPWM modules. If this bit is 0, then the modules are mapped as on 280x/2801x/28044x. If this bit is 1, then the modules are remapped to peripheral frame 3 where they can be accessed by the DMA module.

8.2 CLKIN Input Divider

On 280x/2801x/2804x, the clock into the CPU is either equal or one-half of the phase-locked loop (PLL) output. This divider is controlled by bit 1 (CLKINDIV) of the PLL Status Register (PLLSTS). By default, PLLSTS[CLKINDIV] is configured for divide-by-two (multiply by one-half) operation.

On 2833x/2823x, the divider is controlled by bits 7 & 8 (DIVSEL) of the PLL Status Register (PLLSTS). The PLLSTS[DIVSEL] divider can be set to equal, one-half or one-fourth. By default, at power-up, the divider is set for 1/4. The 1/4 divider can be used to allow a more gradual step up of clocks and to reduce inrush current.

NOTE: The boot ROM changes PLLSTS[DIVSEL] so the loaders will run at one-half the input clock. The boot ROM leaves the divider in this state which is compatible with 280x/2801x/2804x.

8.3 Peripheral Clock Enable Registers

Due to new peripherals and additional instances of old peripherals, the registers to enable and disable the clocks to individual peripherals have been updated. There is an additional register to enable and disable the clocks to the XINTF, CPU Timers, DMA and GPIO input logic: Peripheral Clock Control Register 3 (PCLKCR3). Peripheral Clock Control Register 2 (PCLKCR2) is reserved but not used on the 2833x/2823x devices.

8.4 XCLKOUT Control

On 280x/2801x/2804x, the ratio of XCLKOUT to SYSCKOUT is controlled in the XCLK register. On the 2833x/2823x, XCLKOUT is based off of the XINTF clock (XTIMCLK). Because of this change, the control for XTIMCLK as well as XCLKOUT is in the XINTFCNF2 register within the XINTF module. By default, XCLKOUT is 1/4 of SYSCLKOUT which is the same as on the 280x/2801x/2804x.

8.5 Low Power

The low-power modes remain the same: idle, halt and standby. What has changed is in the way of peripheral clocking. The CPU Timer clocks can now be disabled in the PCLKCR3 register. The CPU Timer clocks are enabled by default to be compatible with 280x/2801x/2804x.

Another new clock control is the GPIOINENCLK bit in PCLKCR3. When this bit is cleared, the input path of the GPIO will be disabled when the pin is configured as an output. This can be used to lower power when the pin is used as an output. Clearing the GPIOINENCLK bit resets the synchronization and qualification logic. By default, GPIOINENCLK is enabled which is compatible with the 280x/2801x/2804x.

NOTE: When GPIOINENCLK is disabled, the GPIO input logic is turned off for any pin configured as an output. This means the respective bit in the DAT register will be an undefined value and can not be used to read the current state of the pin.

9 Peripherals

New peripherals have been added and others have been updated. This section briefly describes the changes. For an overview of all peripherals available, see the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (SPRU566).

9.1 New Peripherals

The 2833x/2823x devices include new peripherals that are not available on the 280x/2801x/2804x devices.

9.1.1 Direct Memory Access (DMA)

The direct memory access module provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby, freeing up bandwidth for other system functions. Additionally, the DMA has the capability to orthogonally rearrange the data as it is transferred as well as *ping-pong* data between buffers. These features are useful for structuring data into blocks for optimal CPU processing.

The DMA module is an event-based machine requiring a peripheral interrupt trigger to start a transfer. The interrupt trigger source for each of the six DMA channels can be configured separately and each channel contains its own independent PIE interrupt to let the CPU know when a DMA transfers has either started or completed. Major features of the DMA are:

- Six channels. Each channel has its own interrupt in the PIE vector table.
- Trigger sources include:
  - ADC sequencer 1 and 2
  - McBSP transmit and receive
  - External interrupts 1-7 and 13
  - CPU timers
  - ePWM1-6 start of conversion (SOCA, SOCB)
  - Software
- Data sources and destinations: L4-L7 SARAM, all XINTF zones, ADC result registers, McBSP transmit and receive registers, ePWM registers.
- Word size can be configured for x16 or x32 bits.

For more information, see the *TMS320x2833x, 2823x Direct Memory Access (DMA) Reference Guide* (SPRUFB8).

9.1.2 External Interface (XINTF)

An external asynchronous interface, similar to that used on the TMS320x281x devices, is included. The XINTF on 2833x/2823x supports both 16- and 32-bit data bus mode. For more information, see the *TMS320x2833x, 2823x DSC External Interface (XINTF) Reference Guide* (SPRU949).

9.1.3 Multichannel Buffered Serial Port (McBSP)

The McBSP on the 2833x/2823x is the same as that on the TMS320x281x devices except that the FIFOs have been replaced with a connection to the DMA module. For more information, see the *TMS320x2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7) for more information.

9.2 Analog-to-Digital Converter (ADC)

The ADC, on the 2833x/2823x, is documented in the *TMS320x2833x Analog-to-Digital (ADC) Module Reference Guide* (SPRU812). This ADC is the same as that on the 280x/2801x/2804x devices with the following two exceptions:

- **ADC Calibration**

  On the 2833x/2823x, the ADC is calibrated by the boot ROM software at boot time. The ADC_cal() routine is programmed into TI reserved OTP memory by the factory. The boot ROM automatically calls the ADC_cal() routine to initialize the ADC Reference Select Register (ADCREFSEL0 and the ADC Offset Trim Register (ADCOFFTRIM) with device-specific calibration data. During normal operation, this process occurs automatically and no action is required by you. If the boot ROM is bypassed by Code Composer Studio during the development process, then ADCREFSEL and ADCOFFTRIM must be initialized by other methods, such as the application or a Code Composer Studio GEL file. For working examples and GEL files, see the ADC initialization in the *C2833x/C2823x C/C++ Header Files and Peripheral Examples* (SPRC530). For more detailed information on the ADC_cal() function, see the *TMS320x2833x Analog-to-Digital (ADC) Module Reference Guide* (SPRU812)

- **ADC Mirror Result Registers**

  The ADC result registers, located at 0xB00, are now accessible by both the CPU and the DMA. The registers have 1 wait state when accessed by the CPU, but are 0 wait when accessed using the DMA. On the 280x/2801x/2804x, these registers were 0 wait for CPU accesses.

9.3 Code Security Module (CSM)

This module protects the Flash, OTP, and L0/L1 SARAM blocks as it did on the 280x/2801x/2804x devices. On the 2833x/2823x, the module was extended to protect both mappings of the L0 and L1 as well as L2 and L3.

NOTE: On 280x/2801x/2804x, the H0 block is not protected by the CSM. On 2833x/2823x, this block has been replaced with L2 and L3 which are both protected by the CSM.

In addition to the CSM, the emulation code security logic (ECSL) has been implemented to prevent unauthorized users from stepping through secure code. Any code or data access to Flash, user OTP, L0, L1, L2 or L3 memory, while the emulator is connected, will trip the ECSL and break the emulation connection. To allow emulation of secure code, while maintaining the CSM protection against secure memory reads, you must write the correct value into the lower 64 bits of the KEY register, which matches the value stored in the lower 64 bits of the password locations within the Flash. Note that dummy reads of all 128 bits of the password in the Flash must still be performed. If the lower 64 bits of the password locations are all ones (unprogrammed), then the KEY value does not need to match. When initially debugging a device with the password locations in Flash programmed (i.e., secured), the emulator takes some time to take control of the CPU. During this time, the CPU will start running and may execute an instruction that performs an access to a protected ECSL area. If this happens, the ECSL trips and causes the emulator connection to be cut. Two solutions to this problem exist:

- Use the wait-in-reset emulation mode, which holds the device in reset until the emulator takes control. The emulator must support this mode for this option.
- Use the *Branch to check boot mode* boot option. This sits in a loop and continuously polls the boot-mode select pins. You can select this boot mode and then exit it once the emulator is connected by re-mapping the PC to another address or by changing the boot-mode selection pins to the desired boot mode.

The CSM password is still the last 128-bits within sector A of the Flash. On the 280x/2801x/2804x, the password is at 0x3F 7FF8 - 0x3F 7FFF. On the 2833x/2823x, it is at 0x33 FFF8 - 0x33 FFFF. If you have any Code Composer Studio GEL scripts that unlock the CSM during debug, they need to be modified to reflect the new password address. An alternative is to keep a memory window open to the password locations during debug.

CSM detailed information is included in the appropriate System Control and Interrupts Reference Guide:

- *TMS320x280x, 2801x, 2804x DSP System Control and Interrupts Reference Guide* (SPRU712)
- *TMS320x2833x, 2823x System Control and Interrupts Reference Guide* (SPRUFB0)

9.4 General-Purpose I/O (GPIO)

The GPIO multiplexing scheme is the same as the 280x/2801x/2804x devices. Additional ports have been added to support the additional GPIOs on the device. Register changes are shown in Table 6.

Table 6. New, Updated, and Removed Registers

Register	Change	Description
GPAMUX2	Updated	Added GPIO28-GPIO31
GPBCTRL	Updated	Added support for GPIO47-GPIO56
GPBQSEL1	Updated	Added support for GPIO34 to GPIO47
GPBQSEL2	New	Port B Qualification for GPIO48 to GPIO63
GPBMUX1	Updated	Added support for GPIO34 to GPIO47
GPBMUX2	New	Port B MUX for GPIO48 to GPIO63
GPBDIR	Updated	Added support for GPIO35 to GPIO63
GPBPUD	Updated	Added support for GPIO35 to GPIO63
GPBDAT	Updated	Added support for GPIO35 to GPIO63
GPBSET	Updated	Added support for GPIO35 to GPIO63
GPBCLEAR	Updated	Added support for GPIO35 to GPIO63
GPBTOGGLE	Updated	Added support for GPIO35 to GPIO63
GPCMUX1	New	Port C MUX for GPIO64 to GPIO79
GPCMUX2	New	Port C MUX for GPIO70 to GPIO87
GPCDIR	New	Port C direction for GPIO64 to GPIO87
GPCPUD	New	Port C pull-up disable for GPIO64 to GPIO87
GPCDAT	New	Port C data for GPIO64 to GPIO87
GPCSET	New	Port C set for GPIO64 to GPIO87
GPCCLEAR	New	Port C clear for GPIO64 to GPIO87
GPCTOGGLE	New	Port C toggle for GPIO64 to GPIO87
GPAMCFG	Removed	2804x only: Selects the pinout configuration for the ePWM

9.4.1 GPIO Ports and MUX

2833x/2823x has additional GPIO pins and, therefore, additional ports. Port A consists of GPIO0-GPIO31, Port B consists of GPIO32-GPIO63, and Port C consists of GPIO64-GPIO87. 280x/2801x/2804x devices only have Port A and Port B.

NOTE: On 2833x/2823x the $\overline{\text{TZ3}}$ and $\overline{\text{XHOLD}}$ input signals come from the same GPIO MUX selection for GPIO14. By default, the XINTF module responds to an XHOLD request. If you are using the pin as $\overline{\text{TZ3}}$ and do not want XINTF to respond to a XHOLD request, then you should disable the $\overline{\text{XHOLD}}$ input in the XINTF's XINTCNF2 register. Similarly, if you are using the XHOLD functionality, the ePWM modules can be configured to ignore the state of the $\overline{\text{TZ3}}$ signal.

9.4.2 GPIO Qualification

On 2833x/2823x devices, the type of qualification required for input signals on GPIO0-GPIO63 can be specified by you just as it is on 280x/2801x/2804x. This qualification applies whether the pin is configured as a GPIO or peripheral. GPIO64-GPIO87 does not have input qualification to reduce delays on the XINTF signals.

9.4.3 External Interrupt Signal Selection

The 2833x/2823x has five more external interrupts. GPIO signals from Port B (GPIO32 - GPIO63) can be assigned to be the interrupt source for XINT3 - XINT7. Refer to Section 10.2.

9.5 Communication Peripherals

The serial communications interface (SCI), serial peripheral interface (SPI), and inter-integrated circuit (I2C) modules remain functionally the same. The register sets are identical on the two device groups. The memory addresses of the registers for each instance of peripheral are also identical. For example, the SCI-A registers on 280x/2801x/2804x are at the same location as SCI-A registers on 2833x/2823x. In addition, the interrupt vector location in the PIE vector table is identical. New interrupt vectors have been added to the PIE vector table for the additional peripheral instances (i.e., SCI-C and eCAN-B). Code that has been written for the I2C, SCI, SPI, on the 280x/2801x/2804x can be directly targeted for the 2833x and 2823x.

The 2833x/2823x eCAN is now clocked at one-half the SYSCLKOUT frequency, unlike 280x/2801x/2804x which clocks the eCAN module at SYSCLKOUT. Code written for the eCAN needs to take this timing change into account. Otherwise, the eCAN module is functionally the same and the register sets are identical on the two device groups.

NOTE: In *C2833x/C2823x C/C++ Header Files and Peripheral Examples* (SPRC530) the AL bits in the I2C Interrupt Enable Register (I2CIER) and the I2C Status Register (I2CSTR) have been renamed ARBL bits. This is to allow the header files to an be used an assembly project where AL is a reserved word.

Table 7. Available Communication Peripherals

Device	McBSP	I2C	SCI Modules	SPI Modules	eCAN Modules
28335, 28334, 28235, 28234	McBSP-A, McBSP-B	I2C-A	SCI-A, SCI-B, SCI-C	SPI-A	eCAN-A, eCAN-B
28332, 28232	McBSP-A	I2C-A	SCI-A, SCI-B	SPI-A	eCAN-A, eCAN-B
2809, 2808	-	I2C-A	SCI-A, SCI-B	SPI-A, SPI-B, SPI-C, SPI-D	eCAN-A, eCAN-B
2806	-	I2C-A	SCI-A, SCI-B	SPI-A, SPI-B, SPI-C, SPI-D	eCAN-A
2802, 2801	-	I2C-A	SCI-A	SPI-A, SPI-B	eCAN-A
28044	-	I2C-A	SCI-A	SPI-A	-
28016	-	I2C-A	SCI-A	SPI-A	eCAN-A
28015	-	I2C-A	SCI-A	SPI-A	-

9.6 Enhanced Control Peripherals

The eCAP, ePWM, HRPWM, and eQEP modules remain functionally the same. The register sets are identical on the two device groups: the memory addresses of the registers for each instance of peripheral are also identical. For example, ePWM1 registers on 280x/2801x/2804x are at the same location as ePWM1 registers on 2833x/2823x. In addition, the interrupt vector location in the PIE vector table is identical. There are two notable differences to the control modules:

- **ePWM and eCAP Synchronization**

  The synchronization scheme varies slightly between 280x, 2801x, 28044 and 2833x/2823x. For more information, see the *TMS320x2833x, 2823x Enhanced Pulse Width Modulator (ePWM) Module Reference Guide* (SPRUG04) and the *TMS320x2833x, 2823x Enhanced Capture (eCAP) Module Reference Guide* (SPRUFG4). Otherwise, code that has been written for these modules on the 280x/2801x/2804x can be directly targeted for the 2833x/2823x.

- **ePWM Connection to DMA**

  On the 2833x/2823x, the ePWM/HRPWM module registers can also be re-mapped to peripheral frame 3 where they can be accessed by the DMA module. A new register called MAPCNF has been added in the system control space at address 0x702E. The MAPCNF[MAPEPWM] bit defines whether the ePWM/HRPWM registers are mapped as on the 280x or if they are mapped to peripheral frame 3.

Table 8. Available Control Peripherals

Device	ePWM	HRPWM	eCAP	eQEP
28335, 28235	ePWM1-ePWM6	ePWM1A-ePWM6A	eCAP1-eCAP6	eQEP1-eQEP2
28334, 28234	ePWM1-ePWM6	ePWM1A-ePWM6A	eCAP1-eCAP4	
28332, 28232	ePWM1-ePWM6	ePWM1A-ePWM4A	eCAP1-eCAP4	eQEP1-eQEP2
2809	ePWM1-ePWM6	ePWM1A-ePWM6A	eCAP1-eCAP4	eQEP1-eQEP2
2808	ePWM1-ePWM6	ePWM1A-ePWM4A	eCAP1-eCAP4	eQEP1-eQEP2
2806	ePWM1-ePWM6	ePWM1A-ePWM4A	eCAP1-eCAP4	eQEP1-eQEP2
2802, 2801	ePWM1-ePWM3	ePWM1A-ePWM3A	eCAP1-eCAP2	eQEP1
28044	ePWM1-ePWM16	ePWM1A-ePWM16A	-	-
28016, 28015	ePWM1-ePWM4	ePWM1A-ePWM4A	eCAP1-eCAP2	-

10 Interrupts

Changes to interrupts include updates to the peripheral interrupt expansion (PIE) module and handling of external interrupts.

10.1 Peripheral Interrupt Expansion (PIE) Module

The functionality of the PIE module and of the PIE configuration registers remains the same. The PIE vector table has been updated to accommodate the interrupts issued by the new peripheral blocks such as DMA, additional external interrupts, and the floating point unit overflow and underflow flags.

10.2 Additional External Interrupts

Five additional interrupts (XINT3 - XINT7) have been added. Each interrupt has its own vector within the PIE vector table. You select the source for each XINT3-XINT7 interrupt GPIO32-GPIO63. This selection is made in the following registers: XINT1 GPIO Input Select Register (GPIOXINT1SEL), XINT2 GPIO Input Select Register (GPIOXINT2SEL), XINT3 GPIO Input Select Register (GPIOXINT3SEL), XINT4 GPIO Input Select Register (GPIOXINT4SEL), XINT6 GPIO Input Select Register (GPIOXINT5SEL), GPIOXINT6SEL, XINT7 GPIO Input Select Register (GPIOXINT7SEL), and XNMI GPIO Input Select Register (GPIOXNMISEL). On the 280x/2801x/2804x devices, only XINT1, XINT2 and XNMI are available.

Table 9. New External Interrupt Registers

Register	Change	Description
XINT3CR	New	External interrupt 3 control
XINT4CR	New	External interrupt 4 control
XINT5CR	New	External interrupt 5 control
XINT6CR	New	External interrupt 6 control
XINT7CR	New	External interrupt 7 control
GPIOXINT3SEL	New	External interrupt 3 pin select (GPIO32-GPIO63)
GPIOXINT4SEL	New	External interrupt 4 pin select (GPIO32-GPIO63)

Table 9. New External Interrupt Registers (continued)

Register	Change	Description
GPIOXINT5SEL	New	External interrupt 5 pin select (GPIO32-GPIO63)
GPIOXINT6SEL	New	External interrupt 6 pin select (GPIO32-GPIO63)
GPIOXINT7SEL	New	External interrupt 7 pin select (GPIO32-GPIO63)

11 Errata Fixes

280x/2801x/2804x errata that have been fixed on 2833x/2823x include:

- When the CAN option is invoked in the boot ROM, the code may hang occasionally.
- SCI bootloader does not clear the ABD bit after auto-baud lock
- Incorrect operation of SCI in address bit mode
- ADC simultaneous sampling latency

For more information on the various advisories, see the TI website for the most recent device specific silicon errata.

- *TMS320F280x, TMS320C280x, and TMS320F2801x DSC Silicon Errata* (SPRZ171)
- *TMS320F28044 DSP Silicon Errata* (SPRZ255)
- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 DSC Silicon Errata* (SPRZ272)

12 References

- *TMS320F280x, TMS320C280x, and TMS320F2801x DSC Silicon Errata* (SPRZ171)
- *TMS320F28044 DSP Silicon Errata* (SPRZ255)
- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 DSC Silicon Errata* (SPRZ272)
- *C28x FPU Primer* (SPRAAN9)
- *TMS320C28x Digital Signal Controller Plus Floating Point Unit* online training available from the TI website at http://www.ti.com/
- *TMS320C28x CPU and Instruction Set Reference Guide* (SPRU430)
- *TMS320C28x Floating Point Unit and Instruction Set Reference Guide* (SPRUEO2)
- *C2833x/C2823x C/C++ Header Files and Peripheral Examples* (SPRC530)
- C28x FPU Fast RTS Library (SPRC664)
- *TMS320F2809, TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, TMS320C2802, TMS320C2801, and TMS320F2801x DSPs Data Manual* (SPRS230)
- *TMS320F28044 Digital Signal Processor Data Manual* (SPRS357)
- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controllers (DSCs) Data Manual* (SPRS439)
- *TMS320x2833x Analog-to-Digital (ADC) Module Reference Guide* (SPRU812)
- *TMS320x280x, 2801x, 2804x Boot ROM Reference Guide* (SPRU722)
- *TMS320x2833x, 2823x Boot ROM Reference Guide* (SPRU963)
- *TMS320x280x, 2801x, 2804x DSP System Control and Interrupts Reference Guide* (SPRU712)
- *TMS320x2833x, 2823x System Control and Interrupts Reference Guide* (SPRUFB0)
- *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (SPRU566)
- *TMS320x2833x, 2823x Direct Memory Access (DMA) Reference Guide* (SPRUFB8)
- *TMS320x2833x, 2823x DSC External Interface (XINTF) Reference Guide* (SPRU949)
- *TMS320x2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7)

- *TMS320x280x, 2801x, 2804x Enhanced Pulse Width Modulator (ePWM) Module Reference Guide* (SPRU791)
- *TMS320x280x, 2801x, 2804x Enhanced Capture (eCAP) Module Reference Guide* (SPRU807)

13 Version History

The following changes were made for Rev B of this application note:

- Corrected fixed errata list.
- Updated peripheral reference guide names and lit numbers

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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