Application Report
SLAA380B–December 2007–Revised September 2018

Migrating From MSP430F16x MCUs to MSP430F261x MCUs

MSP430 Applications

ABSTRACT

This application report helps you migrate designs based on the MSP430F167, MSP430F168, MSP430F169, MSP430F1610, MSP430F1611, or MSP430F1612 microcontrollers (MCUs) to the MSP430F261x device family. This application report describes the main differences between the two device families and provides migration solutions for both software and hardware.

Contents

1 Comparison of MSP430F1xx and MSP430F2xx Families 2
2 Hardware Considerations for MSP430F16x to MSP430F261x Migration 2
2.1 Device Package and Pinout 2
2.2 Current Consumption 3
2.3 Operating Frequency and Supply Voltage 3
2.4 Device Errata 3
3 MSP430F16x to MSP430F261x Migration – Firmware Considerations 4
3.1 CPU and Memory Considerations 4
3.2 Serial Communication – USART Versus USCI 6
3.3 Clock System 7
3.4 Bootloader 8
3.5 Interrupt Vectors 8
3.6 Beware of Reserved Bits! 9
3.7 Timers 9
3.8 Analog Comparator 9
4 References 9

List of Figures

1 Program Counter Storage on the Stack for CALLA 4
2 CACTL2 Bit Description, F1xx Devices 9
3 CACTL2 Bit Description, F2xx Devices 9

List of Tables

1 Comparison of F1xx and F2xx Families 2
2 I^2C Module Connections Migration 3
3 Changed Interrupt Vector Locations 8

Trademarks

MSP430 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Comparison of MSP430F1xx and MSP430F2xx Families

The MSP430F2xx family of microcontrollers provides an upgrade path for the MSP430F1xx family. The MSP430F2xx MCUs offer more performance, lower power, and more built-in features. This enables an improved and more cost-optimized system design. Table 1 contains a general high-level comparison of the two device families, providing an overview of reasons why one should consider migrating.

Table 1. Comparison of F1xx and F2xx Families

Feature	F1xx MCUs	F2xx MCUs
Maximum CPU clock speed	8 MHz	16 MHz
Wake up from LPM3 or LPM4	<6 µs	<1 µs
Standby current consumption (LPM3)	<2 µA	<1 µA
Brownout reset	Selected devices only	All devices
Minimum voltage for flash ISP	2.7 V	2.2 V
Integrated port pullup or pulldown resistors	–	On all ports
Internal oscillator (DCO)	Large voltage and temperature drift (±20%)	Very small voltage and temperature drift (±2%), Factory calibrated
Oscillator fault detection	High-frequency crystal	High-frequency and low-frequency crystal
Additional built-in low-power low-frequency oscillator	–	12-kHz VLO
Additional oscillator features	–	Minimum pulse clock filter for increased system robustness, Configurable built-in crystal load capacitors
Additional watchdog timer features	–	Invalid address detection, Fail-safe clock source
Bootloader (BSL)	Protected through 256-bit password	Hack proof
Flash memory configurations	Up to 60 KB	Up to 120 KB (as of 4Q07)
RAM	Up to 10 KB	Up to 8 KB (as of 4Q07)
Operating temperature (T_A)	–40°C to 85°C	–40°C to 105°C

While the MSP430F261x can mostly be considered as a direct pin-to-pin compatible drop-in into existing MSP430F16x designs, there are some important details that require attention. This application report helps identify potential issues. After migration, the application benefits from all the MSP430F2xx family enhancements as indicated in Table 1. This enables further cost savings or other system-level optimizations. This document focuses on transitioning existing designs and leaves it to the engineer to make use of additional MSP430F2xx features during migration as applicable for a given system (for example, the use of internal pullup and pulldown resistors, making changes to the clock configuration, and so on).

2 Hardware Considerations for MSP430F16x to MSP430F261x Migration

This section provides information on differences between the MSP430F16x and MSP430F261x MCUs that should be considered during migration. Fortunately, the hardware migration process is straightforward, with only a few items to watch out for.

2.1 Device Package and Pinout

The good news is that a 64-pin LQFP MSP430F261x device directly drops into an existing MSP430F16x-based 64-pin LQFP PCB footprint. The package and the PCB footprint are identical. However, in case of a hardware-level redesign of the application to be migrated, the use of the 80-pin LQFP (PN) MSP430F261x device can be considered as an alternative option, as it offers more I/O pins. No direct drop-in replacement is available for a 64-pin MSP430F16x in the QFN package.

While almost all MSP430F261x pins can be used for the same purpose as the pins on their MSP430F16x counterparts (which includes all analog and digital modules, as well as power supply and JTAG pins), there is one exception.

When migrating a design that uses the F16x USART hardware module in I²C mode, special care must be taken, as the pin assignments are different on the MSP430F261x. This is the only pin mismatch that would prevent a direct drop-in of an MSP430F261x into an existing MSP430F16x design (see Table 2).

Table 2. I²C Module Connections Migration

I²C Signal Name	Pin and Signal on F16x (USART0)	Pin and Signal on F261x (USCI_B0)	Comment
SDA	Pin 29 (P3.1/SIMO0/SDA)	Pin 29 (P3.1/UCB0SIMO/UCB0SDA)	No conflict
SCL	Pin 31 (P3.3/UCLK0/SCL)	Pin 30 (P3.2/UCB0SOMI/UCB0SCL)	Pin changed

Details regarding packaging and pinout can be found in the device-specific data sheets. [3][4]

2.2 Current Consumption

When migrating to an MSP430F261x, the difference in current consumption of the devices should be considered. For example, in LFXT1 standby mode (LPM3 using a 32-kHz watch crystal), the standby current consumption of an MSP430F261x is in the 1.1-μA range (typical data sheet value at 3 V, 25°C), which is much lower than the current consumption of an MSP430F16x device, which is in the 2-μA to 2.6-μA range when performing the same function. This is a great benefit for applications that operate in standby mode most of the time. The active current consumption of the devices are comparable when operating at the same frequency, temperature, and voltage conditions. See the device-specific data sheets for the exact specifications. Note that when using LFXT1 in high-frequency mode or when using XT2, the current consumption component caused by the oscillator of an MSP430F261x device is slightly higher as compared to an MSP430F16x device due to differences in the oscillator design to support higher frequencies.

One additional point the designer should be aware of is that when taking advantage of the increased maximum operating frequency the MSP430F261x offers, additional current must be supplied by the system's power supply, because active mode current consumption scales linearly with operating frequency.

2.3 Operating Frequency and Supply Voltage

For the MSP430™ MCUs, the maximum frequency at which the CPU can operate depends on the supply voltage. This specification can be found in the recommended operating conditions of each device-specific data sheet. In general, it can be said that this specification differs for the MSP430F16x and MSP430F261x family of microcontrollers. An MSP430F261x device can always operate under the same operating conditions in terms of supply voltage and CPU clock frequency (MCLK) as an MSP430F16x device. However, if a designer who migrates an existing design to an MSP430F261x device wants to take advantage of the increased maximum clock frequency, it is important to closely review the recommended operating conditions in the MSP430F261x device data sheet. [4]

It is of extreme importance that this relationship is also observed during power-ramp scenarios. Violating this maximum frequency and voltage dependency can result in unpredictable code execution. Note that both MSP430F16x and MSP430F261x MCUs have a built-in SVS module that can be used to ensure that this operating condition is not violated.

2.4 Device Errata

In the course of migrating an existing application to the MSP430F261x, it is recommended that the user review and carefully consider the latest device errata sheets to ensure the application is not affected by a known device issue. Furthermore, the errata sheets typically outline workarounds along with the bug descriptions. For all MSP430 MCUs, the device errata sheets can be found in the product folders of each product on the MSP430 website.

3 MSP430F16x to MSP430F261x Migration – Firmware Considerations

This section outlines important steps to consider when transitioning existing software routines or an entire application to an MSP430F261x device. Even though MSP430F16x and MSP430F261x are code compatible and share many of the same peripherals, it can be said that, in most cases, migration is not as simple as programming the MSP430F16x binary image into an MSP430F261x device. In general, an application needs to be rebuilt on a source-code level (including all referenced code libraries) using the appropriate MSP430F261x device support files such as the header file and the respective linker command file. Doing this is the first step towards a successful migration to an MSP430F261x. The following sections provide more details regarding certain key aspects that need to be considered.

3.1 CPU and Memory Considerations

3.1.1 Extended Memory Architecture

MSP430F261x MCUs feature an upgraded memory and CPU architecture called MSP430X. The main reason for the architectural upgrade is introduction of an internal 20-bit wide address bus, allowing direct access and branching throughout a 1MB wide memory range without paging. The MSP430X CPU is backward compatible with the MSP430 CPU found on the MSP430F16x.

In general, the MSP430F261x CPU directly executes any MSP430F16x binary code as is. When taking advantage of the extended memory space by placing application code at address 0x10000 and beyond (beyond the first 64KB), one detail needs to be carefully considered. As the program counter (PC) is now 20 bit wide, it is important that the entire code makes use of the CALLA and RETA instructions (instead of CALL and RET). This is to ensure that return addresses are correctly stored and retrieved as 20-bit values. In case of recompiling higher level language code such as C, the compiler manages this automatically. However, in the case of transitioning assembler functions or pre-compiled libraries, this change needs to be incorporated on a source-code level.

An additional detail related to the 20-bit wide PC is that any instruction that directly modifies the PC does so according to the used addressing mode. For example, `MOV.W #value,PC` clears the upper 4 bits of the PC, because it is a word-wide instruction. This may or may not be desired. Therefore, assembler routines or precompiled libraries directly accessing the PC need to be reviewed and modified as necessary. Again, in case of higher-level language code, a simple recompile automatically takes care of these modifications.

3.1.2 Subroutine Parameter Passing and Stack Frame

This consideration regards passing parameters to and from assembler functions or precompiled library functions from a higher-level language such as C. For data exchange with a function, a compiler typically allocates and uses certain CPU registers. However, in the case that the parameters cannot all fit into CPU registers, additional data is pushed onto the stack before the subroutine is called. Programs that take advantage of the larger memory available on a MSP430X device (MSP430F261x) perform the subroutine call via the CALLA instruction, thus placing an additional 16-bit word onto the stack caused by the 20-bit wide program counter (see Figure 1).

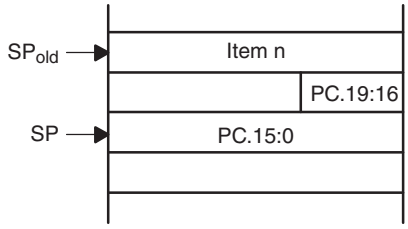

Figure 1. Program Counter Storage on the Stack for CALLA

When the subroutine wants to access the data placed on the stack by the caller, it is important to adjust all the offsets used for accessing the parameters using stack pointer (SP) offset indirect addressing to take the additional word on the stack into account. Therefore, assembler routines ported from the MSP430F16x need to be reviewed and changed if required, and precompiled libraries need to be rebuilt.

As a side note, if a different or newer version compiler is used during migration to an MSP430F261x to take better advantage of the larger memory, it should be verified that the subroutine calling conventions are compatible. This affects the porting of assembler routines as well as the reuse of precompiled libraries. More information on subroutine parameter passing and stack frame can be found in the compiler documentation.

3.1.3 MSP430X Instruction Cycle Count Optimizations

If the code being migrated to the MSP430F261x contains any hand-optimized timing-critical code, keep in mind that the cycle count for certain CPU instructions has been reduced by one or two cycles. However, this consideration typically applies to only assembler-coded software routines, which is the preferred method of coding anything that requires an exact prediction of CPU cycles being consumed.

Another architectural improvement that the MSP430X devices (MSP430F261x) bring is the reduced interrupt response time (five MCLK cycles instead of six), and the reduced time to return from an interrupt service function (three cycles instead of five).

Most applications being migrated transparently benefit from these cycle count reductions.

3.1.4 Device Memory Map

The memory maps of the MSP430F16x and MSP430F261x devices are different. This applies both to the location and size of RAM and to flash memory. Therefore, most applications need to be rebuilt to accommodate for the difference in the memory map. The build process makes use of the memory map information stored in the IDE linker command file and automatically accommodates these changes. The linker command files are found within the folder where the IDE was installed and have the file name extension CMD (for TI Code Composer Essentials) and XCL (for IAR Embedded Workbench). While an MSP430F261x device memory map can be considered as a superset of the MSP430F16x (it can fit any MSP430F16x program), the following two exceptions apply.

The MSP430F1611 has 10 KB of RAM, and the largest MSP430F261x devices at the time of publishing this application report have only 8KB. This requires software modifications if all of the 10KB of RAM are used in an MSP430F1611-based design.

In addition to that, the interrupt vector table of MSP430F261x devices spans 32 memory word locations, and the table in MSP430F16x devices spans 16 memory word locations. Furthermore, the word memory location 0xFFBE on MSP430F261x devices is reserved for special bootloader purposes. See Section 3.5 for more details regarding the interrupt vector table.

Further details regarding the devices' memory maps can be found in device data sheets. [3][4]

3.1.5 Information Flash Memory

Both MSP430F16x and MSP430F261x have 256 bytes of information flash memory located in the memory range of 0x1000 to 0x10FF. While the total memory size is the same, the memory is organized differently. The MSP430F16x device information memory consists of two flash segments (INFOA and INFOB) that are 128 bytes each, and the MSP430F261x has four segments (INFOA, INFOB, INFOC, and INFOD) that are 64 bytes each.

Applications storing data in the information memory need to consider the different segment sizes. Each information flash memory segment must be erased individually, resulting in four write accesses on an MSP430F261x instead of two on the MSP430F16x. Also, note that the MSP430F261x INFOA segment is protected by a lock feature and requires special treatment to be erased or written to. However, in general it is not recommended to erase INFOA or store any user data in it. INFOA comes with factory-provided device-specific calibration data, such as calibration to generate specific frequencies using the DCO. The majority of applications can benefit from those constants.

See the MSP430x2xx Family User's Guide for more details on the organization of the 2xx information flash memory, the INFOA lock feature, and the factory-provided calibration constants. [2]

3.2 Serial Communication – USART Versus USCI

One of the major differences between MSP430F16x and MSP430F261x devices is the serial communication module. On the MSP430F261x, the USCI module is implemented. It is the next generation MSP430 communication module offering more features and functionality to the user. The USART (MSP430F16x) and USCI modules are not software compatible and, therefore, MSP430F16x software using the USART module needs to be adapted to make use of the USCI module.

The MSP430F261x features two independent and identical USCI modules, whereby both provide two communication channels that operate simultaneously. With the MSP430F261x, for example, it is possible to service four SPI communication channels or two I^2C plus two UART channels simultaneously.

It is not in the scope of this application report to discuss all possible aspects regarding migrating application code to use the USCI interface; however, a few items are outlined in the following sections to highlight major differences between the devices (and the modules). In general, it is strongly recommended to carefully review both module descriptions in the appropriate device family user's guide [1][2], as well as to use the USCI code examples provided in the product folders on the MSP430 website as a starting point for any code that is newly created.

3.2.1 UART Mode

The operation of the MSP430F261x USCI in UART mode and that of the MSP430F16x USART is almost identical. The major differences are:

- The MSP430F261x USCI uses a different baud rate generator. It utilizes a new modulation scheme, provides a two-stage modulator, and can be used to implement an oversampling baud rate generation scheme. During application migration, the baud rate register settings must be recalculated. However, it is safe to say that the USCI module can be used to generate the same target baud rate using the same clock source that the MSP430F16x USART would be able to provide.
- The start edge detection and clock activation schemes differ between the two devices. The MSP430F261x features a simplified scheme, in which the USCI module automatically activates the USCI module clock source upon start edge detection, and then provides an interrupt to wake up the CPU after the entire character has been received. On the MSP430F16x UART, an interrupt is generated at start edge detection; therefore, the application must handle the clock source activation itself and then, as a second step, the character reception.
- On the MSP430F261x USCI, interrupt flags are no longer cleared automatically upon entering the interrupt service routine.

3.2.2 SPI Mode

The operation of the MSP430F261x USCI in SPI mode and the MSP430F16x USART is almost identical. The major differences are:

- The MSP430F16x USART supports two channels of simultaneous SPI communication (USART0 and USART1), and the MSP430F261x USCI supports four channels (USCI_A0, USCI_B0, USCI_A1, and USCI_B1).
- On the MSP430F16x, each of the four SPI communication endpoints has a dedicated interrupt vector. On the MSP430F261x, each USCI module has a two shared interrupt vectors, combining transmit and receive events for each module. On both devices, four interrupt vectors are available in total.
- The MSP430F261x USCI defaults to an LSB-first SPI bit order. The bit order can be configured with the UCMSB bit in the UCAxCTL0/UCBxCTL0 control registers. This is different compared to the UART module, where the bit order is MSB first and cannot be configured.
- The maximum MSP430F261x USCI bit clock frequency in SPI master mode is BRCLK, and on the MSP430F16x USART module, it is BRCLK/2.

3.2.3 I²C Mode

The major differences between the MSP430F261x USCI in I²C mode and the MSP430F16x USART are:

- On the MSP430F16x, the I²C feature is exclusively available on USART0, while on the MSP430F261x, it is available on both USCI_B0 and USCI_B1 (two channels simultaneously).
- On the MSP430F16x USART, all I²C activity (data exchange, status flag events) is routed through a single interrupt vector (USART0 TX vector). On the MSP430F261x USCI, both I²C capable USCI channels (USCI_B0 and USCI_B1) have two dedicated interrupt vectors each. One of the two vectors is used for data exchange, the other one for status events.
- Arrangement of I²C status interrupt flags

Furthermore, the MSP430F261x USCI in I²C mode does not support the following features that were available with the MSP430F16x USART:

- Automatic byte counting (I2CNDAT register)
- Word-wide access to the I²C data register (I2CDRW register)
- Decoding of the I²C interrupt source via an interrupt vector generator (I2CIV register)

3.3 Clock System

3.3.1 LFXT1 and XT2 Oscillators

The MSP430F261x oscillator blocks supersede the ones found on MSP430F16x devices. The MSP430F261x oscillators can operate with the same low- and high-frequency oscillators and clock sources but consume less power while providing increased robustness. In addition, built-in software configurable crystal load capacitors are provided in low-frequency (LF) mode. The power-on default for the effective load capacitance in LF mode is 6 pF, which is in line with the MSP430F16x LF oscillator.

When migrating designs that use external crystals or clock sources, items to keep in mind are:

- The capability of MSP430F261x devices to detect low-frequency oscillator failures and indicate them by setting the LFXT1OF flag results in another path for the global oscillator fault flag to become set (OFIFG). This may prevent the CPU from being clocked by a crystal or an external clock source in certain scenarios.
- In case the existing MSP430F16x design uses an external 32-kHz crystal for low-power mode operation and periodic wakeup (LPM3), and crystal-accurate precision is not required, the MSP430F261x built-in VLO oscillator can be used instead, resulting in the elimination of the external crystal and a reduced LPM3 power consumption. The VLO frequency is 12 kHz (data sheet typical value) but can be measured and virtually calibrated. For more details, see reference [5].
- In case an external digital clock source is used, the MSP430F261x newly available direct digital clock input mode should be used (by setting the LFXT1S1 and LFXT1S0 control bits).
- In case the existing MSP430F16x design uses a high-frequency crystal or resonator on LFXT1 or XT2, the appropriate frequency range must be configured in the MSP430F261x clock system control register BCSCTL3. The default range setting is for use with 0.4-MHz to 1-MHz crystals or resonators. See the Basic Clock Module+ user's guide chapter for further details. [2]

3.3.2 Digitally Controlled Oscillator (DCO)

The MSP430F16x and MSP430F261x have different DCO modules. The MSP430F261x DCO offers higher accuracy, an extended frequency range allowing operation of the device up to the maximum operating frequency, and factory-provided calibration constants to facilitate the design of systems that operate without external clock sources.

The key points that should be considered during migration are:

- The default DCO frequency of an MSP430F16x device is in the 800-kHz range, but is in the 1.2-MHz range for an MSP430F261x device. This needs to be considered for applications that run the device using the default DCO settings.
- On an MSP430F261x, consider loading any of the factory-provided DCO calibration constants into the DCO to achieve a deterministic and stable output frequency. The use of the DCO calibration constants may omit the need for software FLL algorithms used on an MSP430F16x device in combination with an external clock source to derive a stable high-speed system clock.
- The MSP430F16x has three bits to control the fundamental frequency range (RSELx in the BCSCTL1 register), and the MSP430F261x has four control bits. Care must be taken when porting algorithms such as a software FLL that modify these bits.
- In case an MSP430F16x application applies hard-coded DCOx, MODx, and RSELx values to the DCO control registers, they result in a different frequency range on an MSP430F261x.
- When enabling the external resistor DCO bias feature (by setting DCOR in the BCSCTL2 register), the MSP430F261x DCO start behaving like an MSP430F16x DCO. In this mode, the same bit settings and external bias resistors result in the same frequency being generated. See the device-specific data sheets for further details. [3][4]

3.4 Bootloader

MSP430F261x devices have a new bootloader (BSL) firmware with enhanced security features. Both MSP430F16x and MSP430F261x device memory access is protected by a 256-bit password. However only MSP430F261x devices erase the entire device flash memory contents (including the factory-provided calibration constants stored in the INFOA flash segment) on the first attempt to access the device with an incorrect password. This behavior is configurable and needs to be considered for applications that use the BSL interface to provide in-field software upgrade capability.

3.5 Interrupt Vectors

The interrupt vector arrangement of MSP430F16x and MSP430F261x devices are different, and application code using interrupt-controlled program flow needs to be migrated. Migrating to an MSP430F261x device involves ensuring that the new interrupt vector locations are used. See Table 3 for a list of module-associated interrupt vectors that require attention.

Table 3. Changed Interrupt Vector Locations

Module	MSP430F16x	MSP430F261x	Comments
Timer_A3	0xFFEA, 0xFFEC	0xFFF0, 0xFFF2	
ADC12	0xFFEE	0xFFEA	
Port 1	0xFFE8	0xFFE4	
Port 2	0xFFE2	0xFFE6	
USART0, USCI_A0/B0	0xFFF0, 0xFFF2	0xFFEC, 0xFFEE	The USCI interrupt vectors are multiplexed between RX, TX, I^2C data, and I^2C status events.
USART1, USCI_A1/B1	0xFFE4, 0xFFE6	0xFFE0, 0xFFE2	
DAC12	0xFFE0 (shared vector)	0xFFDC	F261x has dedicated vectors for DAC12 and DMA, thus simplifying software design.
DMA		0xFFDE	

In general, recompiling the MSP430F16x application code using MSP430F261x device support files automatically populates the interrupt vector table according to the device-specific requirements (for example, for Timer_A or ADC12). However, in some cases, the interrupt vector routines themselves also need to be modified to accommodate a different interrupt flag demultiplexing scheme (for example, USART compared to USCI, DAC12, and DMA).

Also, the memory range that is reserved for interrupt vectors (interrupt vector table) differs between MSP430F16x and MSP430F261x devices. For MSP430F16x devices, this memory ranges from address 0xFFE0 to 0xFFFF (16 words), and for MSP430F261x devices it ranges from 0xFFC0 to 0xFFFF (32 words). In addition, the word memory location 0xFFBE is reserved on MSP430F261x devices and used as the bootloader (BSL) security key (see Section 3.4).

3.6 Beware of Reserved Bits!

The MSP430F261x features a range of upgraded peripherals compared to the MSP430F16x, such as the BCS+ and the Comparator+. This added functionality is partially achieved through the use of bits that were previously marked as reserved on the corresponding MSP430F1xx peripheral. Newer generation MSP430s such as the MSP430F261x make use of these bits to implement additional functionality. If left in the default state, the peripheral usually behaves identical to its MSP430F1xx counterpart. However, care must be taken not to unintentionally switch some of these bits, which can be caused by migrated MSP430F16x firmware. For example, consider the following comparison of CACTL2 control register of Comparator_A and Comparator_A+:

Figure 2. CACTL2 Bit Description, F1xx Devices

7	6	5	4	3	2	1	0
Unused				P2CA1	P2CA0	CAF	CAOUT
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

Figure 3. CACTL2 Bit Description, F2xx Devices

7	6	5	4	3	2	1	0
CASHORT	P2CA4	P2CA3	P2CA2	P2CA1	P2CA0	CAF	CAOUT
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

When firmware that uses the comparator module sets bit 7 and runs fine on an MSP430F16x is executed on an MSP430F261x device, it results in the comparator inputs being shorted together internally.

3.7 Timers

An undocumented feature on the MSP430F16x allows the Timer_A and Timer_B modules to be used in capture mode to generate interrupts on input signal transitions with the timer in stop mode (MCx in TACTL/TBCTL is set to 00h). This feature is no longer available on MSP430F261x devices. To generate capture interrupts, the respective MSP430F261x timer must be running. In this specific use case, consider clocking the timer using a low frequency (for example, ACLK) to minimize power consumption.

3.8 Analog Comparator

On the Comparator_A of MSP430F16x devices, disabling the digital port functionality for an I/O pin by setting the associated bit in the Port Disable Register CAPD to prevent parasitic cross currents during analog measurements disables the digital CMOS input buffer. However, on MSP430F261x devices with Comparator_A+, setting a CAPDx bit disables both input and output buffer for that pin.

4 References

1. MSP430x1xx Family User's Guide
2. MSP430x2xx Family User's Guide
3. MSP430F15x, MSP430F16x, MSP430F161x Mixed-Signal Microcontrollers
4. MSP430F241x, MSP430F261x Mixed-Signal Microcontrollers
5. Using the VLO Library

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from November 12, 2008 to September 26, 2018	Page
• Formatting and editorial changes throughout document	1

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