# TMS320C28346, TMS320C28345, TMS320C28344
# TMS320C28343, TMS320C28342, TMS320C28341
## Delfino Microcontrollers

# Data Manual



Literature Number: SPRS516B
March 2009–Revised July 2010

# Contents

# List of Figures

# List of Tables

# Delfino Microcontrollers

Check for Samples: TMS320C28346, TMS320C28345, TMS320C28344, TMS320C28343, TMS320C28342, TMS320C28341

## 1 TMS320C2834x ( Delfino™) MCUs

### 1.1 Overview

The TMS320C2834x (C2834x) Delfino™ microcontroller (MCU) devices build on TI's existing F2833x high-performance floating-point microcontrollers. The C2834x delivers up to 300 MHz of floating-point performance, and has up to 516KB of on-chip RAM. Designed for real-time control applications, the C2834x is based on the C28x™ core, making it code-compatible with all C28x microcontrollers. The on-chip peripherals and low-latency core make the C2834x an excellent solution for performance-hungry real-time control applications.

### 1.2 Features

- **High-Performance Static CMOS Technology**
  - **Up to 300 MHz (3.33-ns Cycle Time)**
  - **1.1-V/1.2-V Core, 3.3-V I/O , 1.8-V PLL/Oscillator Design**
- **High-Performance 32-Bit CPU (TMS320C28x)**
  - **IEEE-754 Single-Precision Floating-Point Unit (FPU)**
  - **16 x 16 and 32 x 32 MAC Operations**
  - **16 x 16 Dual MAC**
  - **Harvard Bus Architecture**
  - **Fast Interrupt Response and Processing**
  - **Code-Efficient (in C/C++ and Assembly)**
- **Six-Channel DMA Controller (for McBSP, XINTF, and SARAM)**
- **16-Bit or 32-Bit External Interface (XINTF)**
  - **Over 2M x 16 Address Reach**
- **On-Chip Memory**
  - **Up to 258K x 16 SARAM**
  - **8K x 16 Boot ROM**
- **Clock and System Control**
  - **Dynamic PLL Ratio Changes Supported**
  - **On-Chip Oscillator**
  - **Watchdog Timer Module**
- **Peripheral Interrupt Expansion (PIE) Block That Supports All 64 Peripheral Interrupts**
- **Enhanced Control Peripherals**
  - **Eighteen Enhanced Pulse Width Modulator (ePWM) Outputs**
    - **Dedicated 16-Bit Time-Based Counter With Period and Frequency Control**
    - **Single-Edge, Dual-Edge Symmetric, or Dual-Edge Asymmetric Outputs**
    - **Dead-Band Generation**
    - **PWM Chopping by High-Frequency Carrier**
    - **Trip Zone Input**
    - **Up to 9 HRPWM Outputs With 55-ps MEP Resolution at $V_{DD}$ = 1.1 V (65 ps at 1.2 V)**
  - **Six 32-Bit Enhanced Capture (eCAP) Modules**
    - **Configurable as 3 Capture Inputs or 3 Auxiliary Pulse Width Modulator Outputs**
    - **Single-Shot Capture of up to Four Event Time-Stamps**
  - **Three 32-Bit Quadrature Encoder Pulse (QEP) Modules**
  - **Six 32-Bit Timers/Nine 16-Bit Timers**
- **Three 32-Bit CPU Timers**
- **Serial Port Peripherals**
  - **Up to 2 CAN Modules**
  - **Up to 3 SCI (UART) Modules**
  - **Up to 2 McBSP Modules (Configurable as SPI)**
  - **Up to 2 SPI Module s**
  - **One Inter-Integrated-Circuit (I2C) Bus**
- **External ADC Interface**
- **Up to 88 Individually Programmable, Multiplexed GPIO Pins With Input Filtering**
- **Advanced Emulation Features**
  - **Analysis and Breakpoint Functions**
  - **Real-Time Debug via Hardware**

- **2834x Package Options:**
  - **MicroStar BGA™ (ZHH)**
  - **Plastic BGA (ZFE)**
- **Community Resources**
  - **TI E2E Community**
  - **TI Embedded Processors Wiki**

## 1.3 Getting Started

This section gives a brief overview of the steps to take when first developing for a C28x device. For more detail on each of these steps, see the following:

- *Getting Started With TMS320C28x Digital Signal Controllers* (literature number SPRAAM0).
- C2000 Getting Started Website (http://www.ti.com/c2000getstarted)
- TMS320F28x Development and Experimenter's Kits (http://www.ti.com/f28xkits)

# 2 Introduction

The TMS320C28346, TMS320C28345, TMS320C28344, TMS320C28343, TMS320C28342, and TMS320C28341 devices, members of the Delfino™ MCU generation, are highly integrated, high-performance solutions for demanding control applications.

Throughout this document, the devices are abbreviated as C28346, C28345, C28344, C28343, C28342, and C28341, respectively. Table 2-1 provides a summary of features for each device.

## Table 2-1. C2834x Hardware Features

| FEATURE | TYPE(1) | C28346 (300 MHz) | C28345 (200 MHz) | | C28344 (300 MHz) | C28343 (200 MHz) | | C28342 (300 MHz) | C28341 (200 MHz) | |
|---|---|---|---|---|---|---|---|---|---|---|
| Package Type | – | 256-Ball ZFE PBGA(2) | 256-Ball ZFE PBGA(2) | 179-Ball ZHH BGA | 256-Ball ZFE PBGA(2) | 256-Ball ZFE PBGA(2) | 179-Ball ZHH BGA | 256-Ball ZFE PBGA(2) | 256-Ball ZFE PBGA(2) | 179-Ball ZHH BGA |
| Instruction cycle | – | 3.33 ns | 5 ns | | 3.33 ns | 5 ns | | 3.33 ns | 5 ns | |
| Floating-point unit | – | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| Single-access RAM (SARAM) (16-bit word) | – | 258K | 258K | | 130K | 130K | | 98K | 98K | |
| Code security for on-chip SARAM blocks | – | No(3) | No(3) | | No(3) | No(3) | | No(3) | No(3) | |
| Boot ROM (8K x 16) | – | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| 16-/32-bit External Interface (XINTF) | 1 | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| 6-channel Direct Memory Access (DMA) | 0 | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| PWM outputs | 0 | ePWM1/2/3/ 4/5/6/7/8/9 | ePWM1/2/3/ 4/5/6/7/8/9 | | ePWM1/2/3/ 4/5/6/7/8/9 | ePWM1/2/3/ 4/5/6/7/8/9 | | ePWM1/2/3/ 4/5/6 | ePWM1/2/3/ 4/5/6 | |
| HRPWM channels | 0 | ePWM1A/2A/ 3A/4A/5A/6A/ 7A/8A/9A | ePWM1A/2A/ 3A/4A/5A/6A/ 7A/8A/9A | | ePWM1A/2A/ 3A/4A/5A/6A/ 7A/8A/9A | ePWM1A/2A/ 3A/4A/5A/6A/ 7A/8A/9A | | ePWM1A/2A/ 3A/4A/5A/6A | ePWM1A/2A/ 3A/4A/5A/6A | |
| 32-bit Capture inputs or auxiliary PWM outputs | 0 | 6 | 6 | | 6 | 6 | | 4 | 4 | |
| 32-bit QEP channels (four inputs/channel) | 0 | 3 | 3 | | 3 | 3 | | 2 | 2 | |
| Watchdog timer | – | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| External ADC Interface | – | Yes | Yes | | Yes | Yes | | Yes | Yes | |
| 32-bit CPU timers | – | 3 | 3 | | 3 | 3 | | 3 | 3 | |
| Multichannel Buffered Serial Port (McBSP)/SPI | 1 | 2 | 2 | | 2 | 2 | | 1 | 1 | |
| Serial Peripheral Interface (SPI) | 0 | 2 | 2 | | 2 | 2 | | 2 | 2 | |
| Serial Communications Interface (SCI) | 0 | 3 | 3 | | 3 | 3 | | 3 | 3 | |
| Enhanced Controller Area Network (eCAN) | 0 | 2 | 2 | | 2 | 2 | | 2 | 2 | |
| Inter-Integrated Circuit (I2C) | 0 | 1 | 1 | | 1 | 1 | | 1 | 1 | |
| General-Purpose Input/Output (GPIO) pins (shared) | – | 88 | 88 | | 88 | 88 | | 88 | 88 | |
| External interrupts | – | 8 | 8 | | 8 | 8 | | 8 | 8 | |

(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. These device-specific differences are listed in the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (literature number SPRU566) and in the peripheral reference guides.
(2) TMX samples will come with the ZEP designator. The designator will change to ZFE after TMS.
(3) Custom secure versions of these devices are available. See Section 3.2.9, Security, for more details.

**Table 2-1. C2834x Hardware Features (continued)**

| FEATURE | | TYPE(1) | C28346 (300 MHz) | C28345 (200 MHz) | | C28344 (300 MHz) | C28343 (200 MHz) | | C28342 (300 MHz) | C28341 (200 MHz) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Temperature options | T: –40°C to 105°C | – | ZFE | ZFE | ZHH | ZFE | ZFE | ZHH | ZFE | ZFE | ZHH |
| | S: –40°C to 125°C | – | ZFE | ZFE | – | ZFE | ZFE | – | ZFE | ZFE | – |
| | Q: –40°C to 125°C (Q100 qualification) | – | ZFE | ZFE | – | ZFE | ZFE | – | ZFE | ZFE | – |
| Product status(1) | | – | TMS | TMS | | TMS | TMS | | TMS | TMS | |

(1) See Section 5.1 for descriptions of device stages.

## 2.1 Pin Assignments

The 179-ball ZHH ball grid array (BGA) terminal assignments are shown in Figure 2-1 through Figure 2-4. The 256-ball ZFE plastic ball grid array (PBGA) terminal assignments are shown in Figure 2-5 through Figure 2-8. Table 2-2 describes the function(s) of each pin.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| P | EXTSOC2B | EXTSOC3B | GPIO19/ $\overline{\text{SPISTEA}}$/ SCIRXDB/ CANTXA | GPIO23/ EQEP1I/ MFSXA/ SCIRXDB | GPIO24/ ECAP1/ EQEP2A/ MDXB | GPIO32/ SDAA/ EPWMSYNCI/ $\overline{\text{ADCSOCAO}}$ | **$V_{DD}$** | P |
| N | EXTSOC1A | EXTSOC3A | EXTADCCLK | GPIO22/ EQEP1S/ MCLKXA/ SCITXDB | **$V_{DD}$** | GPIO33/ SCLA/ EPWMSYNCO/ $\overline{\text{ADCSOCBO}}$ | TDO | N |
| M | **$V_{DD}$** | EXTSOC2A | EXTSOC1B | GPIO21/ EQEP1B/ MDRA/ CANRXB | GPIO25/ ECAP2/ EQEP2B/ MDRB | GPIO27/ ECAP4/ EQEP2S/ MFSXB | $\overline{\text{TRST}}$ | M |
| L | GPIO18/ SPICLKA/ SCITXDB/ CANRXA | **$V_{DDIO}$** | **$V_{SS}$** | GPIO20/ EQEP1A/ MDXA/ CANTXB | **$V_{SS}$** | TDI | **$V_{SS}$** | L |
| K | **$V_{SS}$** | GPIO15/ $\overline{\text{TZ4}}$/XHOLDA/ SCIRXDB/ MFSXB | **$V_{DD}$** | GPIO16/ SPISIMOA/ CANTXB/ $\overline{\text{TZ5}}$ | **$V_{DDIO}$** | GPIO26/ ECAP3/ EQEP2I/ MCLKXB | **$V_{DDIO}$** | K |
| J | **$V_{DDIO}$** | **$V_{SS}$** | **$V_{DD}$** | **$V_{DD}$** | GPIO17/ SPISOMIA/ CANRXB/ $\overline{\text{TZ6}}$ | | | J |
| H | **$V_{SS}$** | GPIO12/ $\overline{\text{TZ1}}$/ CANTXB/ MDXB | GPIO11/ EPWM6B/ SCIRXDB/ ECAP4 | GPIO13/ $\overline{\text{TZ2}}$/ CANRXB/ MDRB | GPIO14/ $\overline{\text{TZ3}}$/XHOLD/ SCITXDB/ MCLKXB | | | H |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |

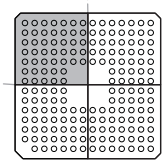


**Figure 2-1. C2834x 179-Ball ZHH MicroStar BGA™ Upper Left Quadrant (Bottom VIew)**

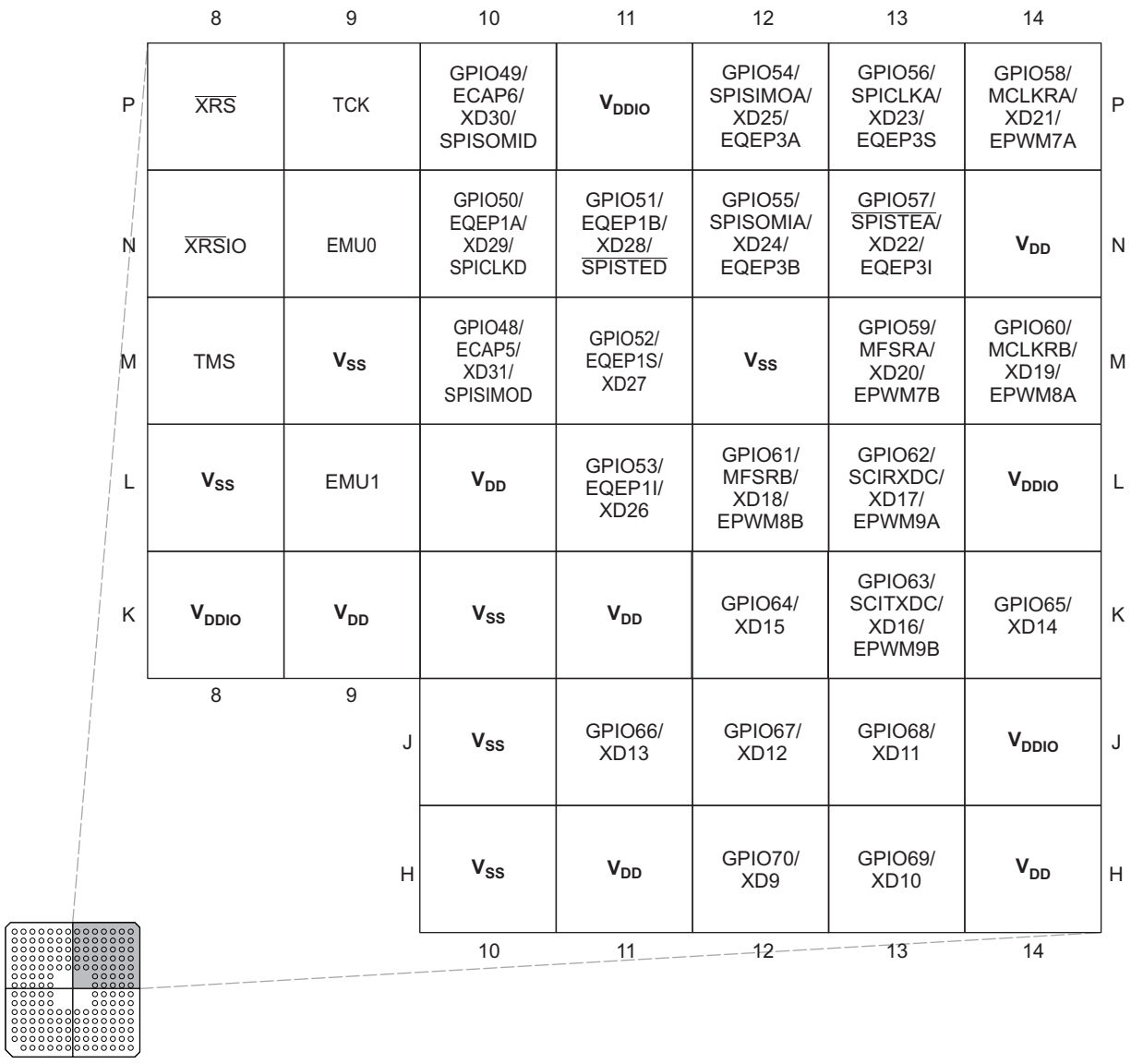
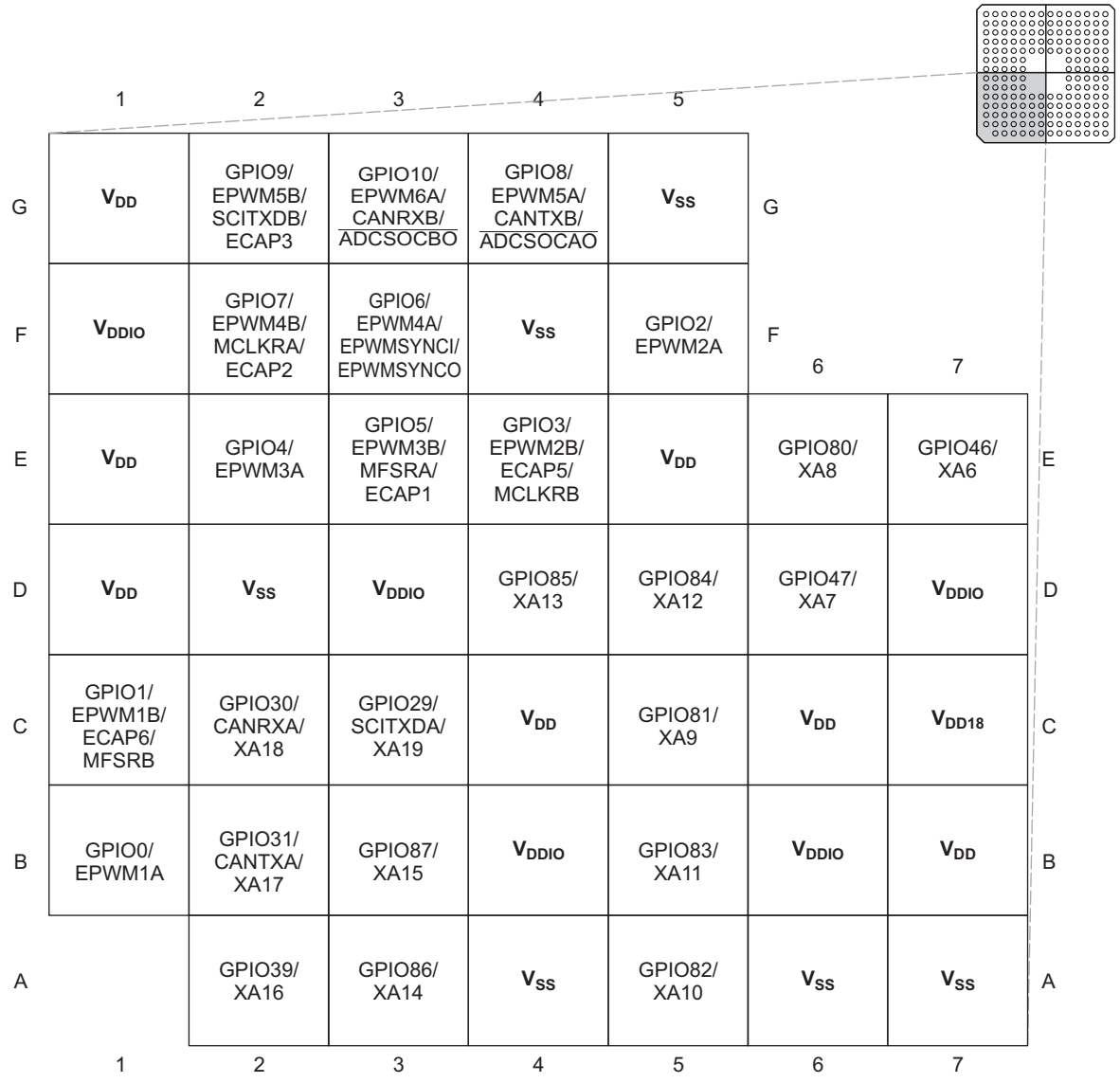



**Figure 2-2. C2834x 179-Ball ZHH MicroStar BGA™ Upper Right Quadrant (Bottom View)**



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| G | $V_{DD}$ | GPIO9/ EPWM5B/ SCITXDB/ ECAP3 | GPIO10/ EPWM6A/ CANRXB/ ADCSOCBO | GPIO8/ EPWM5A/ CANTXB/ ADCSOCAO | $V_{SS}$ | | | G |
| F | $V_{DDIO}$ | GPIO7/ EPWM4B/ MCLKRA/ ECAP2 | GPIO6/ EPWM4A/ EPWMSYNCI/ EPWMSYNCO | $V_{SS}$ | GPIO2/ EPWM2A | | | F |
| E | $V_{DD}$ | GPIO4/ EPWM3A | GPIO5/ EPWM3B/ MFSRA/ ECAP1 | GPIO3/ EPWM2B/ ECAP5/ MCLKRB | $V_{DD}$ | GPIO80/ XA8 | GPIO46/ XA6 | E |
| D | $V_{DD}$ | $V_{SS}$ | $V_{DDIO}$ | GPIO85/ XA13 | GPIO84/ XA12 | GPIO47/ XA7 | $V_{DDIO}$ | D |
| C | GPIO1/ EPWM1B/ ECAP6/ MFSRB | GPIO30/ CANRXA/ XA18 | GPIO29/ SCITXDA/ XA19 | $V_{DD}$ | GPIO81/ XA9 | $V_{DD}$ | $V_{DD18}$ | C |
| B | GPIO0/ EPWM1A | GPIO31/ CANTXA/ XA17 | GPIO87/ XA15 | $V_{DDIO}$ | GPIO83/ XA11 | $V_{DDIO}$ | $V_{DD}$ | B |
| A | | GPIO39/ XA16 | GPIO86/ XA14 | $V_{SS}$ | GPIO82/ XA10 | $V_{SS}$ | $V_{SS}$ | A |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |

**Figure 2-3. C2834x 179-Ball ZHH MicroStar BGA™ Lower Left Quadrant (Bottom View)**

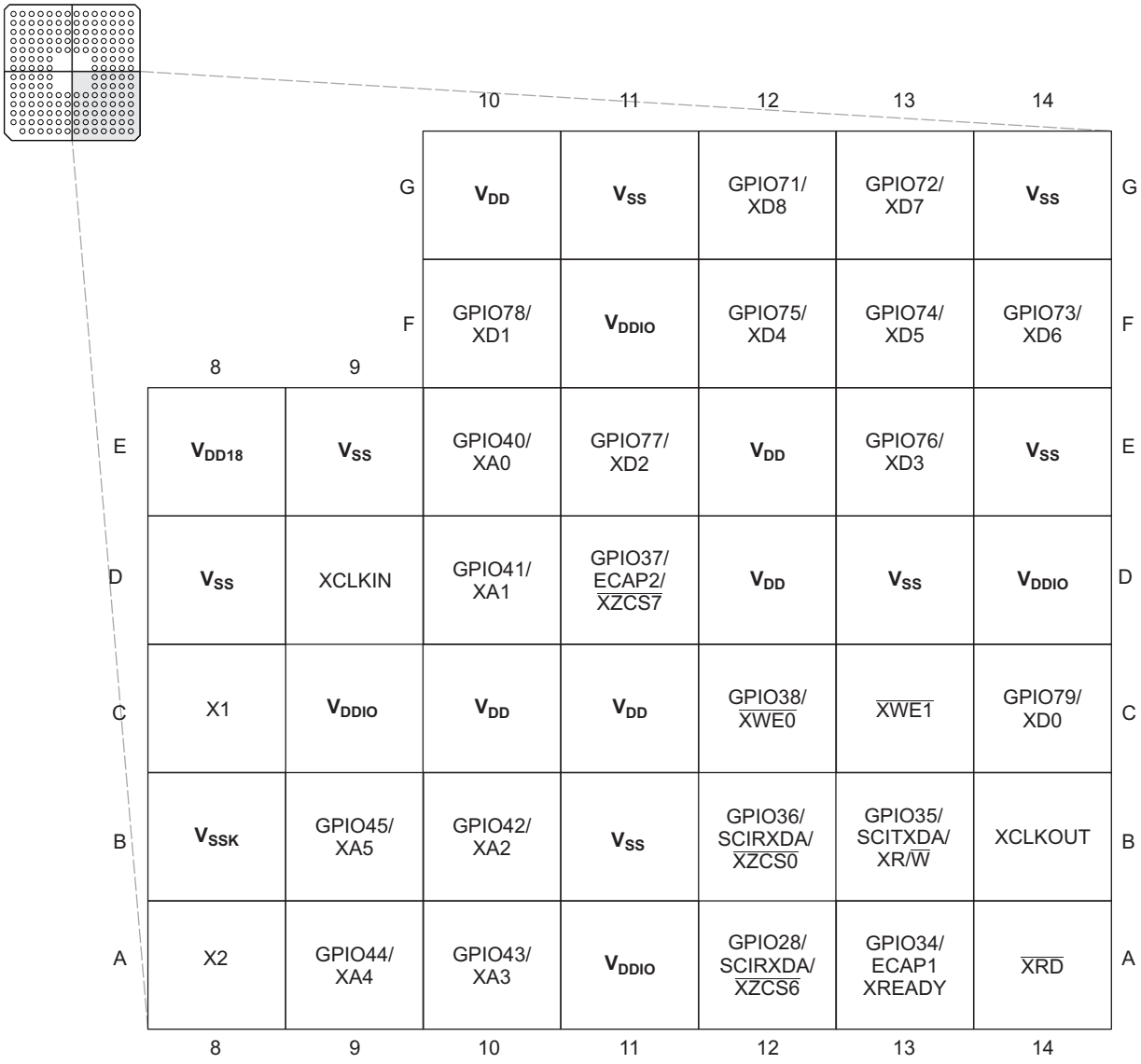


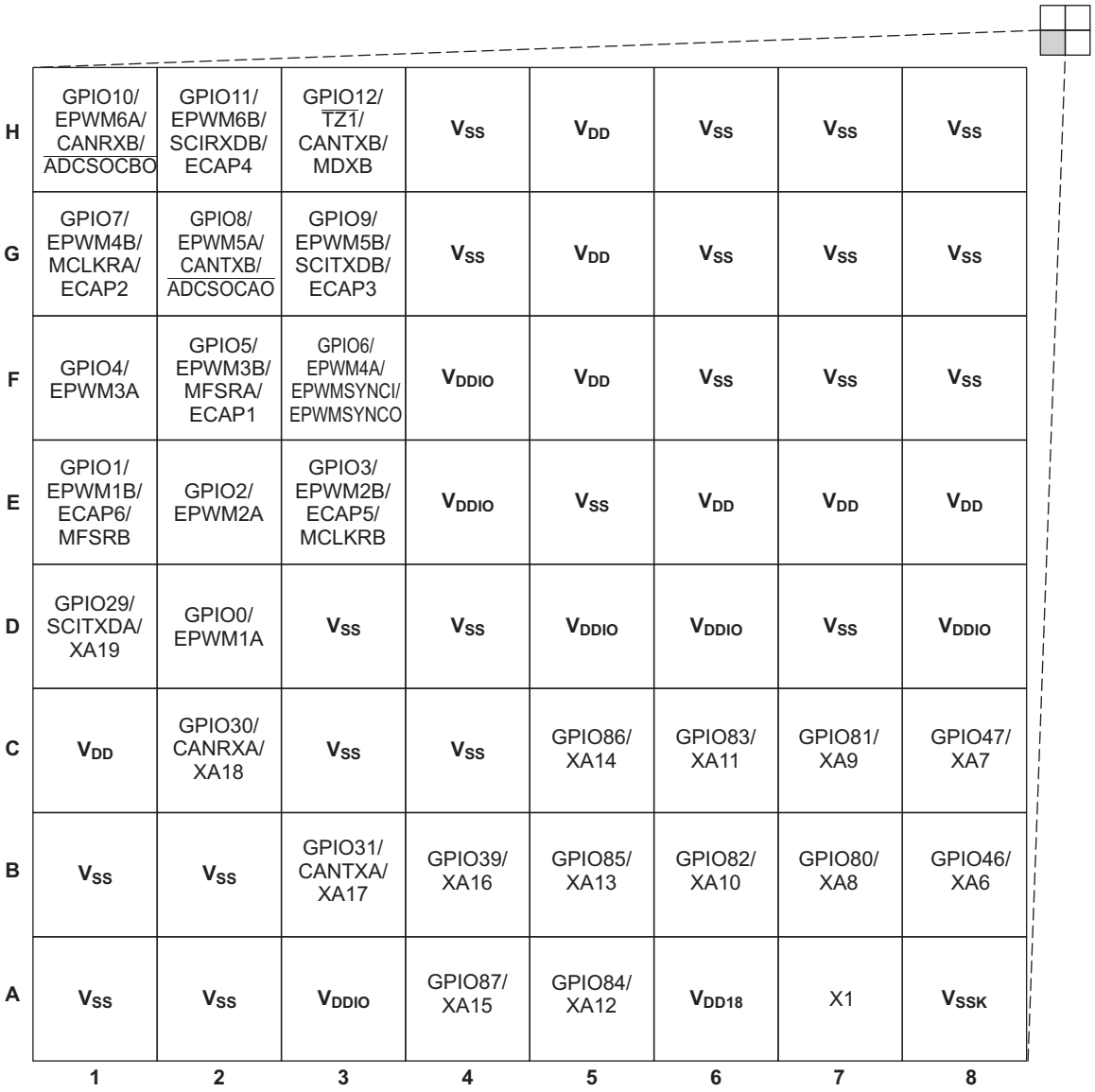

**Figure 2-4. C2834x 179-Ball ZHH MicroStar BGA™ Lower Right Quadrant (Bottom View)**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| T | $V_{SS}$ | $V_{SS}$ | $V_{DDIO}$ | GPIO19/ SPISTEA/ SCIRXDB/ CANTXA | GPIO21/ EQEP1B/ MDRA/ CANRXB | GPIO24/ ECAP1/ EQEP2A/ MDXB | GPIO27/ ECAP4/ EQEP2S/ MFSXB | TDI |
| R | $V_{SS}$ | $V_{SS}$ | EXTADCCLK | GPIO20/ EQEP1A/ MDXA/ CANTXB | GPIO22/ EQEP1S/ MCLKXA/ SCITXDB | GPIO25/ ECAP2/ EQEP2B/ MDRB | GPIO32/ SDAA/ EPWMSYNCI/ ADCSOCAO | TRST |
| P | $V_{DD}$ | EXTSOC3B | $V_{SS}$ | $V_{SS}$ | GPIO23/ EQEP1I/ MFSXA/ SCIRXDB | GPIO26/ ECAP3/ EQEP2I/ MCLKXB | GPIO33/ SCLA/ EPWMSYNCO/ ADCSOCBO | TDO |
| N | EXTSOC2A | EXTSOC2B | EXTSOC3A | $V_{SS}$ | $V_{DDIO}$ | $V_{DDIO}$ | $V_{SS}$ | $V_{DDIO}$ |
| M | GPIO18/ SPICLKA/ SCITXDB/ CANRXA | EXTSOC1A | EXTSOC1B | $V_{DDIO}$ | $V_{SS}$ | $V_{DD}$ | $V_{DD}$ | $V_{DD}$ |
| L | GPIO16/ SPISIMOA/ CANTXB/ TZ5 | GPIO17/ SPISOMIA/ CANRXB/ TZ6 | $V_{DD}$ | $V_{DDIO}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |
| K | $V_{SS}$ | GPIO15/ TZ4/XHOLDA/ SCIRXDB/ MFSXB | $V_{DD}$ | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |
| J | $V_{DDIO}$ | GPIO13/ TZ2/ CANRXB/ MDRB | GPIO14/ TZ3/XHOLD/ SCITXDB/ MCLKXB | $V_{DDIO}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |

**Figure 2-5. C2834x 256-Ball ZFE Plastic BGA Upper Left Quadrant (Bottom View)**

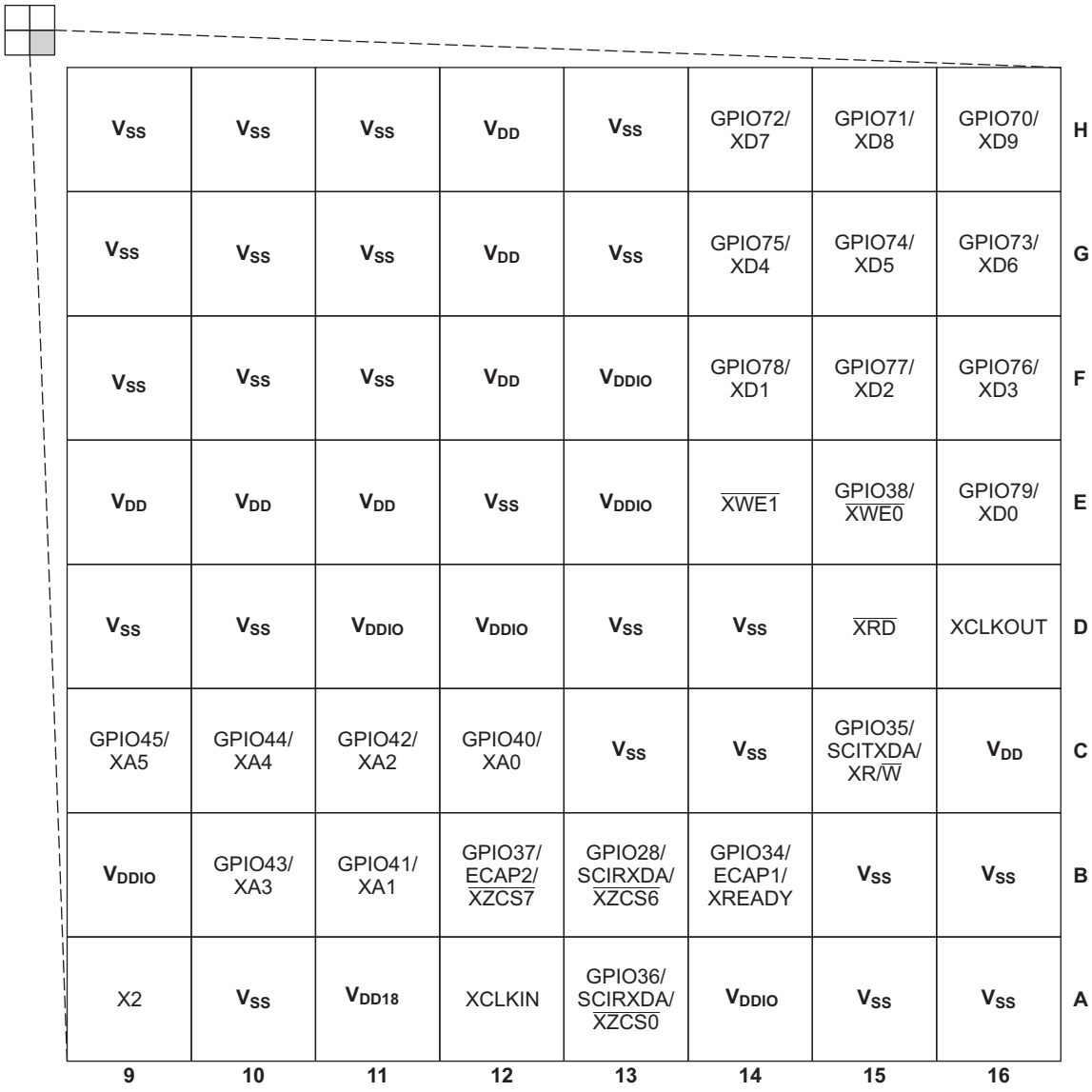
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|---|---|---|---|---|---|---|---|---|
| $\overline{\text{XRS}}$IO | $\overline{\text{XRS}}$ | TCK | GPIO50/ EQEP1A/ XD29/ SPICLKD | GPIO53/ EQEP1I/ XD26 | $V_{DDIO}$ | $V_{SS}$ | $V_{SS}$ | T |
| $V_{DDIO}$ | EMU1 | GPIO48/ ECAP5/ XD31/ SPISIMOD | GPIO51/ EQEP1B/ XD28/ $\overline{\text{SPISTED}}$ | GPIO54/ SPISIMOA/ XD25/ EQEP3A | GPIO56/ SPICLKA/ XD23/ EQEP3S | $V_{SS}$ | $V_{SS}$ | R |
| TMS | EMU0 | GPIO49/ ECAP6/ XD30/ SPISOMID | GPIO52/ EQEP1S/ XD27 | GPIO55/ SPISOMIA/ XD24/ EQEP3B | $V_{SS}$ | GPIO57/ $\overline{\text{SPISTEA}}$/ XD22/ EQEP3I | $V_{DD}$ | P |
| $V_{SS}$ | $V_{SS}$ | $V_{DDIO}$ | $V_{DDIO}$ | $V_{SS}$ | $V_{SS}$ | GPIO59/ MFSRA/ XD20/ EPWM7B | GPIO58/ MCLKRA/ XD21/ EPWM7A | N |
| $V_{DD}$ | $V_{DD}$ | $V_{DD}$ | $V_{SS}$ | $V_{DDIO}$ | GPIO62/ SCIRXDC/ XD17/ EPWM9A | GPIO61/ MFSRB/ XD18/ EPWM8B | GPIO60/ MCLKRB/ XD19/ EPWM8A | M |
| $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{DDIO}$ | GPIO65/ XD14 | GPIO64/ XD15 | GPIO63/ SCITXDC/ XD16/ EPWM9B | L |
| $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | GPIO67/ XD12 | GPIO66/ XD13 | $V_{SS}$ | K |
| $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{DDIO}$ | GPIO69/ XD10 | GPIO68/ XD11 | $V_{DDIO}$ | J |

**Figure 2-6. C2834x 256-Ball ZFE Plastic BGA Upper Right Quadrant (Bottom View)**



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| H | GPIO10/ EPWM6A/ CANRXB/ $\overline{\text{ADCSOCBO}}$ | GPIO11/ EPWM6B/ SCIRXDB/ ECAP4 | GPIO12/ $\overline{\text{TZ1}}$/ CANTXB/ MDXB | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |
| G | GPIO7/ EPWM4B/ MCLKRA/ ECAP2 | GPIO8/ EPWM5A/ CANTXB/ $\overline{\text{ADCSOCAO}}$ | GPIO9/ EPWM5B/ SCITXDB/ ECAP3 | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |
| F | GPIO4/ EPWM3A | GPIO5/ EPWM3B/ MFSRA/ ECAP1 | GPIO6/ EPWM4A/ EPWMSYNCI/ EPWMSYNCO | $V_{DDIO}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ |
| E | GPIO1/ EPWM1B/ ECAP6/ MFSRB | GPIO2/ EPWM2A | GPIO3/ EPWM2B/ ECAP5/ MCLKRB | $V_{DDIO}$ | $V_{SS}$ | $V_{DD}$ | $V_{DD}$ | $V_{DD}$ |
| D | GPIO29/ SCITXDA/ XA19 | GPIO0/ EPWM1A | $V_{SS}$ | $V_{SS}$ | $V_{DDIO}$ | $V_{DDIO}$ | $V_{SS}$ | $V_{DDIO}$ |
| C | $V_{DD}$ | GPIO30/ CANRXA/ XA18 | $V_{SS}$ | $V_{SS}$ | GPIO86/ XA14 | GPIO83/ XA11 | GPIO81/ XA9 | GPIO47/ XA7 |
| B | $V_{SS}$ | $V_{SS}$ | GPIO31/ CANTXA/ XA17 | GPIO39/ XA16 | GPIO85/ XA13 | GPIO82/ XA10 | GPIO80/ XA8 | GPIO46/ XA6 |
| A | $V_{SS}$ | $V_{SS}$ | $V_{DDIO}$ | GPIO87/ XA15 | GPIO84/ XA12 | $V_{DD18}$ | X1 | $V_{SSK}$ |

**Figure 2-7. C2834x 256-Ball ZFE Plastic BGA Lower Left Quadrant (Bottom View)**



| | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|---|---|---|---|---|---|---|---|---|---|
| | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | GPIO72/ XD7 | GPIO71/ XD8 | GPIO70/ XD9 | H |
| | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | GPIO75/ XD4 | GPIO74/ XD5 | GPIO73/ XD6 | G |
| | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{DDIO}$ | GPIO78/ XD1 | GPIO77/ XD2 | GPIO76/ XD3 | F |
| | $V_{DD}$ | $V_{DD}$ | $V_{DD}$ | $V_{SS}$ | $V_{DDIO}$ | $\overline{XWE1}$ | GPIO38/ $\overline{XWE0}$ | GPIO79/ XD0 | E |
| | $V_{SS}$ | $V_{SS}$ | $V_{DDIO}$ | $V_{DDIO}$ | $V_{SS}$ | $V_{SS}$ | $\overline{XRD}$ | XCLKOUT | D |
| | GPIO45/ XA5 | GPIO44/ XA4 | GPIO42/ XA2 | GPIO40/ XA0 | $V_{SS}$ | $V_{SS}$ | GPIO35/ SCITXDA/ XR/$\overline{W}$ | $V_{DD}$ | C |
| | $V_{DDIO}$ | GPIO43/ XA3 | GPIO41/ XA1 | GPIO37/ ECAP2/ $\overline{XZCS7}$ | GPIO28/ SCIRXDA/ $\overline{XZCS6}$ | GPIO34/ ECAP1/ XREADY | $V_{SS}$ | $V_{SS}$ | B |
| | X2 | $V_{SS}$ | $V_{DD18}$ | XCLKIN | GPIO36/ SCIRXDA/ $\overline{XZCS0}$ | $V_{DDIO}$ | $V_{SS}$ | $V_{SS}$ | A |

**Figure 2-8. C2834x 256-Ball ZFE Plastic BGA Lower Right Quadrant (Bottom View)**

## 2.2 Signal Descriptions

Table 2-2 describes the signals. The GPIO function (shown in Italics) is the default at reset. The peripheral signals that are listed under them are alternate functions. Some peripheral functions may not be available in all devices. See Table 2-1 for details. Inputs are not 5-V tolerant. All XINTF pins have a drive strength of 4 mA (typical). All GPIO pins are I/O/Z, 4-mA drive typical and have an internal pullup, which can be selectively enabled/disabled on a per-pin basis. This feature only applies to the GPIO pins. The pullups on GPIO0–GPIO11 and GPIO58–GPIO63 pins are not enabled at reset. The pullups on GPIO12–GPIO57 and GPIO64–GPIO87 are enabled upon reset.

**Table 2-2. Signal Descriptions**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| **JTAG** | | | |
| $\overline{\text{TRST}}$ | M7 | R8 | JTAG test reset with internal pulldown. $\overline{\text{TRST}}$, when driven high, gives the scan system control of the operations of the device. If this signal is not connected or driven low, the device operates in its functional mode, and the test reset signals are ignored.<br>NOTE: $\overline{\text{TRST}}$ is an active high test pin and must be maintained low at all times during normal device operation. An external pulldown resistor is recommended on this pin. The value of this resistor should be based on drive strength of the debugger pods applicable to the design. A 2.2-kΩ resistor generally offers adequate protection. Since this is application-specific, it is recommended that each target board be validated for proper operation of the debugger and the application. (I, ↓) |
| TCK | P9 | T11 | JTAG test clock. An external pullup resistor is required on this pin. A 2.2-kΩ resistor generally offers adequate protection.(I) |
| TMS | M8 | P9 | JTAG test-mode select (TMS) with internal pullup. This serial control input is clocked into the TAP controller on the rising edge of TCK. (I, ↑) |
| TDI | L6 | T8 | JTAG test data input (TDI) with internal pullup. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK. (I, ↑) |
| TDO | N7 | P8 | JTAG scan out, test data output (TDO). The contents of the selected register (instruction or data) are shifted out of TDO on the falling edge of TCK. |
| EMU0 | N9 | P10 | Emulator pin 0. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through the JTAG scan. This pin is also used to put the device into boundary-scan mode. With the EMU0 pin at a logic-high state and the EMU1 pin at a logic-low state, a rising edge on the $\overline{\text{TRST}}$ pin would latch the device into boundary-scan mode.<br>**NOTE:** An external pullup resistor is recommended on this pin. The value of this resistor should be based on the drive strength of the debugger pods applicable to the design. A 2.2-kΩ to 4.7-kΩ resistor is generally adequate. Since this is application-specific, it is recommended that each target board be validated for proper operation of the debugger and the application. |
| EMU1 | L9 | R10 | Emulator pin 1. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through the JTAG scan. This pin is also used to put the device into boundary-scan mode. With the EMU0 pin at a logic-high state and the EMU1 pin at a logic-low state, a rising edge on the $\overline{\text{TRST}}$ pin would latch the device into boundary-scan mode.<br>**NOTE:** An external pullup resistor is recommended on this pin. The value of this resistor should be based on the drive strength of the debugger pods applicable to the design. A 2.2-kΩ to 4.7-kΩ resistor is generally adequate. Since this is application-specific, it is recommended that each target board be validated for proper operation of the debugger and the application. |
| **Clock** | | | |
| XCLKOUT | B14 | D16 | Output clock derived from SYSCLKOUT. XCLKOUT is either the same frequency, one-half the frequency, one-fourth the frequency, or one-eighth the frequency of SYSCLKOUT. This is controlled by bit 19 (BY4CLKMODE), bits 18:16 (XTIMCLK), and bit 2 (CLKMODE) in the XINTCNF2 register. At reset, XCLKOUT = SYSCLKOUT/8. The XCLKOUT signal can be turned off by setting XINTCNF2[CLKOFF] to 1. Unlike other GPIO pins, the XCLKOUT pin is not placed in high-impedance state during a reset. |
| XCLKIN | D9 | A12 | External Oscillator Input. This pin is to feed a clock from an external 3.3-V oscillator. In this case, the X1 pin must be tied to $V_{SSK}$. If a crystal/resonator is used (or if an external 1.8-V oscillator is used to feed clock to X1 pin), this pin must be tied to $V_{SS}$. (I) |
| X1 | C8 | A7 | Internal/External Oscillator Input. To use the internal oscillator, a quartz crystal may be connected across X1 and X2. The X1 pin is referenced to the 1.8-V core digital power supply. A 1.8-V external oscillator may be connected to the X1 pin. In this case, the XCLKIN pin must be connected to $V_{SS}$. If a 3.3-V external oscillator is used with the XCLKIN pin, X1 must be tied to $V_{SSK}$. (I) |
| X2 | A8 | A9 | Internal Oscillator Output. A quartz crystal may be connected across X1 and X2. If X2 is not used it must be left unconnected. (O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| **Reset** | | | |
| $\overline{\text{XRS}}$ | P8 | T10 | Device Reset (in) and Watchdog Reset (out).<br>Device reset. $\overline{\text{XRS}}$ causes the device to terminate execution. The PC will point to the address contained at the location 0x3FFFC0. When $\overline{\text{XRS}}$ is brought to a high level, execution begins at the location pointed to by the PC. This pin is driven low by the MCU when a watchdog reset occurs. During watchdog reset, the $\overline{\text{XRS}}$ pin is driven low for the watchdog reset duration of 512 OSCCLK cycles. (I/OD, ↑)<br>The output buffer of this pin is an open-drain with an internal pullup. It is recommended that this pin be driven by an open-drain device. |
| $\overline{\text{XRS}}$IO | N8 | T9 | $\overline{\text{XRS}}$ I/O Control (I) - This pin must be connected to the $\overline{\text{XRS}}$ pin on the target board. When $\overline{\text{XRS}}$ is low (reset), the level detected on this pin puts all output buffers on the device in high-impedance mode. |
| **External ADC Interface Signals** | | | |
| EXTSOC1A | N1 | M2 | External ADC SOC Group 1 A Output. Trigger for external ADC, this signal is logical OR of ePWM1/2/3 SOCA internal signals (O) |
| EXTSOC1B | M3 | M3 | External ADC SOC Group 1 B Output. Trigger for external ADC, this signal is logical OR of ePWM1/2/3 SOCB internal signals (O) |
| EXTSOC2A | M2 | N1 | External ADC SOC Group 2 A Output. Trigger for external ADC, this signal is logical OR of ePWM4/5/6 SOCA internal signals (O) |
| EXTSOC2B | P1 | N2 | External ADC SOC Group 2 B Output. Trigger for external ADC, this signal is logical OR of ePWM4/5/6 SOCB internal signals (O) |
| EXTSOC3A | N2 | N3 | External ADC SOC Group 3 A Output. Trigger for external ADC, this signal is logical OR of ePWM7/8/9 SOCA internal signals (O) |
| EXTSOC3B | P2 | P2 | External ADC SOC Group3 B Output. Trigger for external ADC, this signal is logical OR of ePWM7/8/9 SOCB internal signals (O) |
| EXTADCCLK | N3 | R3 | External ADC Clock Signal. Clock for external ADC support, derived from SYSCLK (O) |
| **GPIO and Peripheral Signals** | | | |
| *GPIO0*<br>EPWM1A<br>-<br>- | B1 | D2 | General purpose input/output 0 (I/O/Z)<br>Enhanced PWM1 Output A and HRPWM channel (O)<br>-<br>- |
| *GPIO1*<br>EPWM1B<br>ECAP6<br>MFSRB | C1 | E1 | General purpose input/output 1 (I/O/Z)<br>Enhanced PWM1 Output B (O)<br>Enhanced Capture 6 input/output (I/O)<br>McBSP-B receive frame synch (I/O) |
| *GPIO2*<br>EPWM2A<br>-<br>- | F5 | E2 | General purpose input/output 2 (I/O/Z)<br>Enhanced PWM2 Output A and HRPWM channel (O)<br>-<br>- |
| *GPIO3*<br>EPWM2B<br>ECAP5<br>MCLKRB | E4 | E3 | General purpose input/output 3 (I/O/Z)<br>Enhanced PWM2 Output B (O)<br>Enhanced Capture 5 input/output (I/O)<br>McBSP-B receive clock (I/O) |
| *GPIO4*<br>EPWM3A<br>-<br>- | E2 | F1 | General purpose input/output 4 (I/O/Z)<br>Enhanced PWM3 output A and HRPWM channel (O)<br>-<br>- |
| *GPIO5*<br>EPWM3B<br>MFSRA<br>ECAP1 | E3 | F2 | General purpose input/output 5 (I/O/Z)<br>Enhanced PWM3 output B (O)<br>McBSP-A receive frame synch (I/O)<br>Enhanced Capture input/output 1 (I/O) |
| *GPIO6*<br>EPWM4A<br>EPWMSYNCI<br>EPWMSYNCO | F3 | F3 | General purpose input/output 6 (I/O/Z)<br>Enhanced PWM4 output A and HRPWM channel (O)<br>External ePWM sync pulse input (I)<br>External ePWM sync pulse output (O) |
| *GPIO7*<br>EPWM4B<br>MCLKRA<br>ECAP2 | F2 | G1 | General purpose input/output 7 (I/O/Z)<br>Enhanced PWM4 output B (O)<br>McBSP-A receive clock (I/O)<br>Enhanced capture input/output 2 (I/O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| *GPIO8*<br>EPWM5A<br>CANTXB<br>ADCSOCAO | G4 | G2 | General Purpose Input/Output 8 (I/O/Z)<br>Enhanced PWM5 output A and HRPWM channel (O)<br>Enhanced CAN-B transmit (O)<br>ADC start-of-conversion A (O) |
| *GPIO9*<br>EPWM5B<br>SCITXDB<br>ECAP3 | G2 | G3 | General purpose input/output 9 (I/O/Z)<br>Enhanced PWM5 output B (O)<br>SCI-B transmit data(O)<br>Enhanced capture input/output 3 (I/O) |
| *GPIO10*<br>EPWM6A<br>CANRXB<br>ADCSOCBO | G3 | H1 | General purpose input/output 10 (I/O/Z)<br>Enhanced PWM6 output A and HRPWM channel (O)<br>Enhanced CAN-B receive (I)<br>ADC start-of-conversion B (O) |
| *GPIO11*<br>EPWM6B<br>SCIRXDB<br>ECAP4 | H3 | H2 | General purpose input/output 11 (I/O/Z)<br>Enhanced PWM6 output B (O)<br>SCI-B receive data (I)<br>Enhanced CAP Input/Output 4 (I/O) |
| *GPIO12*<br>TZ1<br>CANTXB<br>MDXB | H2 | H3 | General purpose input/output 12 (I/O/Z)<br>Trip Zone input 1 (I)<br>Enhanced CAN-B transmit (O)<br>McBSP-B transmit serial data (O) |
| *GPIO13*<br>TZ2<br>CANRXB<br>MDRB | H4 | J2 | General purpose input/output 13 (I/O/Z)<br>Trip Zone input 2 (I)<br>Enhanced CAN-B receive (I)<br>McBSP-B receive serial data (I) |
| *GPIO14*<br><br><br><br>TZ3/XHOLD<br><br><br><br>SCITXDB<br>MCLKXB | H5 | J3 | General purpose input/output 14 (I/O/Z)<br>Trip Zone input 3/External Hold Request. XHOLD, when active (low), requests the external interface (XINTF) to release the external bus and place all buses and strobes into a high-impedance state. To prevent this from happening when TZ3 signal goes active, disable this function by writing XINTCNF2[HOLD] = 1. If this is not done, the XINTF bus will go into high impedance anytime TZ3 goes low. On the ePWM side, TZn signals are ignored by default, unless they are enabled by the code. The XINTF will release the bus when any current access is complete and there are no pending accesses on the XINTF. (I)<br>SCI-B Transmit (O)<br>McBSP-B transmit clock (I/O) |
| *GPIO15*<br><br><br><br>TZ4/XHOLDA<br><br><br><br>SCIRXDB<br>MFSXB | K2 | K2 | General purpose input/output 15 (I/O/Z)<br>Trip Zone input 4/External Hold Acknowledge. The pin function for this option is based on the direction chosen in the GPADIR register. If the pin is configured as an input, then TZ4 function is chosen. If the pin is configured as an output, then XHOLDA function is chosen. XHOLDA is driven active (low) when the XINTF has granted an XHOLD request. All XINTF buses and strobe signals will be in a high-impedance state. XHOLDA is released when the XHOLD signal is released. External devices should only drive the external bus when XHOLDA is active (low). (I/O)<br>SCI-B receive (I)<br>McBSP-B transmit frame synch (I/O) |
| *GPIO16*<br>SPISIMOA<br>CANTXB<br>TZ5 | K4 | L1 | General purpose input/output 16 (I/O/Z)<br>SPI slave in, master out (I/O)<br>Enhanced CAN-B transmit (O)<br>Trip Zone input 5 (I) |
| *GPIO17*<br>SPISOMIA<br>CANRXB<br>TZ6 | J5 | L2 | General purpose input/output 17 (I/O/Z)<br>SPI-A slave out, master in (I/O)<br>Enhanced CAN-B receive (I)<br>Trip zone input 6 (I) |
| *GPIO18*<br>SPICLKA<br>SCITXDB<br>CANRXA | L1 | M1 | General purpose input/output 18 (I/O/Z)<br>SPI-A clock input/output (I/O)<br>SCI-B transmit (O)<br>Enhanced CAN-A receive (I) |
| *GPIO19*<br>SPISTEA<br>SCIRXDB<br>CANTXA | P3 | T4 | General purpose input/output 19 (I/O/Z)<br>SPI-A slave transmit enable input/output (I/O)<br>SCI-B receive (I)<br>Enhanced CAN-A transmit (O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| *GPIO20*<br>EQEP1A<br>MDXA<br>CANTXB | L4 | R4 | General purpose input/output 20 (I/O/Z)<br>Enhanced QEP1 input A (I)<br>McBSP-A transmit serial data (O)<br>Enhanced CAN-B transmit (O) |
| *GPIO21*<br>EQEP1B<br>MDRA<br>CANRXB | M4 | T5 | General purpose input/output 21 (I/O/Z)<br>Enhanced QEP1 input B (I)<br>McBSP-A receive serial data (I)<br>Enhanced CAN-B receive (I) |
| *GPIO22*<br>EQEP1S<br>MCLKXA<br>SCITXDB | N4 | R5 | General purpose input/output 22 (I/O/Z)<br>Enhanced QEP1 strobe (I/O)<br>McBSP-A transmit clock (I/O)<br>SCI-B transmit (O) |
| *GPIO23*<br>EQEP1I<br>MFSXA<br>SCIRXDB | P4 | P5 | General purpose input/output 23 (I/O/Z)<br>Enhanced QEP1 index (I/O)<br>McBSP-A transmit frame synch (I/O)<br>SCI-B receive (I) |
| *GPIO24*<br>ECAP1<br>EQEP2A<br>MDXB | P5 | T6 | General purpose input/output 24 (I/O/Z)<br>Enhanced capture 1 (I/O)<br>Enhanced QEP2 input A (I)<br>McBSP-B transmit serial data (O) |
| *GPIO25*<br>ECAP2<br>EQEP2B<br>MDRB | M5 | R6 | General purpose input/output 25 (I/O/Z)<br>Enhanced capture 2 (I/O)<br>Enhanced QEP2 input B (I)<br>McBSP-B receive serial data (I) |
| *GPIO26*<br>ECAP3<br>EQEP2I<br>MCLKXB | K6 | P6 | General purpose input/output 26 (I/O/Z)<br>Enhanced capture 3 (I/O)<br>Enhanced QEP2 index (I/O)<br>McBSP-B transmit clock (I/O) |
| *GPIO27*<br>ECAP4<br>EQEP2S<br>MFSXB | M6 | T7 | General purpose input/output 27 (I/O/Z)<br>Enhanced capture 4 (I/O)<br>Enhanced QEP2 strobe (I/O)<br>McBSP-B transmit frame synch (I/O) |
| *GPIO28*<br>SCIRXDA<br>$\overline{\text{XZCS6}}$ | A12 | B13 | General purpose input/output 28 (I/O/Z)<br>SCI receive data (I)<br>External Interface zone 6 chip select (O) |
| *GPIO29*<br>SCITXDA<br>XA19 | C3 | D1 | General purpose input/output 29. (I/O/Z)<br>SCI transmit data (O)<br>External Interface Address Line 19 (O) |
| *GPIO30*<br>CANRXA<br>XA18 | C2 | C2 | General purpose input/output 30 (I/O/Z)<br>Enhanced CAN-A receive (I)<br>External Interface Address Line 18 (O) |
| *GPIO31*<br>CANTXA<br>XA17 | B2 | B3 | General purpose input/output 31 (I/O/Z)<br>Enhanced CAN-A transmit (O)<br>External Interface Address Line 17 (O) |
| *GPIO32*<br>SDAA<br>$\overline{\text{EPWMSYNCI}}$<br>$\overline{\text{ADCSOCAO}}$ | P6 | R7 | General purpose input/output 32 (I/O/Z)<br>I2C data open-drain bidirectional port (I/OD)<br>Enhanced PWM external sync pulse input (I)<br>ADC start-of-conversion A (O) |
| *GPIO33*<br>SCLA<br>$\overline{\text{EPWMSYNCO}}$<br>$\overline{\text{ADCSOCBO}}$ | N6 | P7 | General-Purpose Input/Output 33 (I/O/Z)<br>I2C clock open-drain bidirectional port (I/OD)<br>Enhanced PWM external synch pulse output (O)<br>ADC start-of-conversion B (O) |
| *GPIO34*<br>ECAP1<br>XREADY | A13 | B14 | General-Purpose Input/Output 34 (I/O/Z)<br>Enhanced Capture input/output 1 (I/O)<br>External Interface Ready signal |
| *GPIO35*<br>SCITXDA<br>XR/$\overline{\text{W}}$ | B13 | C15 | General-Purpose Input/Output 35 (I/O/Z)<br>SCI-A transmit data (O)<br>External Interface read, not write strobe |
| *GPIO36*<br>SCIRXDA<br>$\overline{\text{XZCS0}}$ | B12 | A13 | General-Purpose Input/Output 36 (I/O/Z)<br>SCI-A receive data (I)<br>External Interface zone 0 chip select (O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| *GPIO37*<br>ECAP2<br>XZCS7 | D11 | B12 | General-Purpose Input/Output 37 (I/O/Z)<br>Enhanced Capture input/output 2 (I/O)<br>External Interface zone 7 chip select (O) |
| *GPIO38*<br>-<br>XWE0 | C12 | E15 | General-Purpose Input/Output 38 (I/O/Z)<br>-<br>External Interface Write Enable 0 (O) |
| GPIO39<br>-<br>XA16 | A2 | B4 | General-Purpose Input/Output 39 (I/O/Z)<br>-<br>External Interface Address Line 16 (O) |
| GPIO40<br>-<br>XA0 | E10 | C12 | General-Purpose Input/Output 40 (I/O/Z)<br>-<br>External Interface Address Line 0 |
| GPIO41<br>-<br>XA1 | D10 | B11 | General-Purpose Input/Output 41 (I/O/Z)<br>-<br>External Interface Address Line 1 (O) |
| GPIO42<br>-<br>XA2 | B10 | C11 | General-Purpose Input/Output 42 (I/O/Z)<br>-<br>External Interface Address Line 2 (O) |
| GPIO43<br>-<br>XA3 | A10 | B10 | General-Purpose Input/Output 43 (I/O/Z)<br>-<br>External Interface Address Line 3 (O) |
| GPIO44<br>-<br>XA4 | A9 | C10 | General-Purpose Input/Output 44 (I/O/Z)<br>-<br>External Interface Address Line 4 (O) |
| GPIO45<br>-<br>XA5 | B9 | C9 | General-Purpose Input/Output 45 (I/O/Z)<br>-<br>External Interface Address Line 5 (O) |
| GPIO46<br>-<br>XA6 | E7 | B8 | General-Purpose Input/Output 46 (I/O/Z)<br>-<br>External Interface Address Line 6 (O) |
| GPIO47<br>-<br>XA7 | D6 | C8 | General-Purpose Input/Output 47 (I/O/Z)<br>-<br>External Interface Address Line 7 (O) |
| GPIO48<br>ECAP5<br>XD31<br>SPISIMOD | M10 | R11 | General-Purpose Input/Output 48 (I/O/Z)<br>Enhanced Capture input/output 5 (I/O)<br>External Interface Data Line 31 (O)<br>SPI-D slave in, master out (I/O) |
| GPIO49<br>ECAP6<br>XD30<br>SPISOMID | P10 | P11 | General-Purpose Input/Output 49 (I/O/Z)<br>Enhanced Capture input/output 6 (I/O)<br>External Interface Data Line 30 (O)<br>SPI-D slave out, master in (I/O) |
| GPIO50<br>EQEP1A<br>XD29<br>SPICLKD | N10 | T12 | General-Purpose Input/Output 50 (I/O/Z)<br>Enhanced QEP 1input A (I)<br>External Interface Data Line 29 (O)<br>SPI-D Clock input/output (I/O) |
| GPIO51<br>EQEP1B<br>XD28<br>SPISTED | N11 | R12 | General-Purpose Input/Output 51 (I/O/Z)<br>Enhanced QEP 1input B (I)<br>External Interface Data Line 28 (O)<br>SPI-D slave transmit enable input/output (I/O) |
| GPIO52<br>EQEP1S<br>XD27 | M11 | P12 | General-Purpose Input/Output 52 (I/O/Z)<br>Enhanced QEP 1Strobe (I/O)<br>External Interface Data Line 27 (O) |
| GPIO53<br>EQEP1I<br>XD26 | L11 | T13 | General-Purpose Input/Output 53 (I/O/Z)<br>Enhanced QEP1 Index (I/O)<br>External Interface Data Line 26 (O) |
| GPIO54<br>SPISIMOA<br>XD25<br>EQEP3A | P12 | R13 | General-Purpose Input/Output 54 (I/O/Z)<br>SPI-A slave in, master out (I/O)<br>External Interface Data Line 25 (O)<br>Enhanced QEP3 input A (I) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| GPIO55<br>SPISOMIA<br>XD24<br>EQEP3B | N12 | P13 | General-Purpose Input/Output 55 (I/O/Z)<br>SPI-A slave out, master in (I/O)<br>External Interface Data Line 24 (O)<br>Enhanced QEP3 input B (I) |
| GPIO56<br>SPICLKA<br>XD23<br>EQEP3S | P13 | R14 | General-Purpose Input/Output 56 (I/O/Z)<br>SPI-A clock (I/O)<br>External Interface Data Line 23 (O)<br>Enhanced QEP3 strobe (I/O) |
| GPIO57<br>SPISTEA<br>XD22<br>EQEP3I | N13 | P15 | General-Purpose Input/Output 57 (I/O/Z)<br>SPI-A slave transmit enable (I/O)<br>External Interface Data Line 22 (O)<br>Enhanced QEP3 index (I/O) |
| GPIO58<br>MCLKRA<br>XD21<br>EPWM7A | P14 | N16 | General-Purpose Input/Output 58 (I/O/Z)<br>McBSP-A receive clock (I/O)<br>External Interface Data Line 21 (O)<br>Enhanced PWM 7 output A and HRPWM channel (O) |
| GPIO59<br>MFSRA<br>XD20<br>EPWM7B | M13 | N15 | General-Purpose Input/Output 59 (I/O/Z)<br>McBSP-A receive frame synch (I/O)<br>External Interface Data Line 20 (O)<br>Enhanced PWM 7 output B (O) |
| GPIO60<br>MCLKRB<br>XD19<br>EPWM8A | M14 | M16 | General-Purpose Input/Output 60 (I/O/Z)<br>McBSP-B receive clock (I/O)<br>External Interface Data Line 19 (O)<br>Enhanced PWM 8 output A and HRPWM channel (O) |
| GPIO61<br>MFSRB<br>XD18<br>EPWM8B | L12 | M15 | General-Purpose Input/Output 61 (I/O/Z)<br>McBSP-B receive frame synch (I/O)<br>External Interface Data Line 18 (O)<br>Enhanced PWM8 output B (O) |
| GPIO62<br>SCIRXDC<br>XD17<br>EPWM9A | L13 | M14 | General-Purpose Input/Output 62 (I/O/Z)<br>SCI-C receive data (I)<br>External Interface Data Line 17 (O)<br>Enhanced PWM9 output A and HRPWM channel (O) |
| GPIO63<br>SCITXDC<br>XD16<br>EPWM9B | K13 | L16 | General-Purpose Input/Output 63 (I/O/Z)<br>SCI-C transmit data (O)<br>External Interface Data Line 16 (O)<br>Enhanced PWM9 output B (O) |
| GPIO64<br>-<br>XD15 | K12 | L15 | General-Purpose Input/Output 64 (I/O/Z)<br>-<br>External Interface Data Line 15 (O) |
| GPIO65<br>-<br>XD14 | K14 | L14 | General-Purpose Input/Output 65 (I/O/Z)<br>-<br>External Interface Data Line 14 (O) |
| GPIO66<br>-<br>XD13 | J11 | K15 | General-Purpose Input/Output 66 (I/O/Z)<br>-<br>External Interface Data Line 13 (O) |
| GPIO67<br>-<br>XD12 | J12 | K14 | General-Purpose Input/Output 67 (I/O/Z)<br>-<br>External Interface Data Line 12 (O) |
| GPIO68<br>-<br>XD11 | J13 | J15 | General-Purpose Input/Output 68 (I/O/Z)<br>-<br>External Interface Data Line 11 (O) |
| GPIO69<br>-<br>XD10 | H13 | J14 | General-Purpose Input/Output 69 (I/O/Z)<br>-<br>External Interface Data Line 10 (O) |
| GPIO70<br>-<br>XD9 | H12 | H16 | General-Purpose Input/Output 70 (I/O/Z)<br>-<br>External Interface Data Line 9 (O) |
| GPIO71<br>-<br>XD8 | G12 | H15 | General-Purpose Input/Output 71 (I/O/Z)<br>-<br>External Interface Data Line 8 (O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| GPIO72<br>-<br>XD7 | G13 | H14 | General-Purpose Input/Output 72 (I/O/Z)<br>-<br>External Interface Data Line 7 (O) |
| GPIO73<br>-<br>XD6 | F14 | G16 | General-Purpose Input/Output 73 (I/O/Z)<br>-<br>External Interface Data Line 6 (O) |
| GPIO74<br>-<br>XD5 | F13 | G15 | General-Purpose Input/Output 74 (I/O/Z)<br>-<br>External Interface Data Line 5 (O) |
| GPIO75<br>-<br>XD4 | F12 | G14 | General-Purpose Input/Output 75 (I/O/Z)<br>-<br>External Interface Data Line 4 (O) |
| GPIO76<br>-<br>XD3 | E13 | F16 | General-Purpose Input/Output 76 (I/O/Z)<br>-<br>External Interface Data Line 3 (O) |
| GPIO77<br>-<br>XD2 | E11 | F15 | General-Purpose Input/Output 77 (I/O/Z)<br>-<br>External Interface Data Line 2 (O) |
| GPIO78<br>-<br>XD1 | F10 | F14 | General-Purpose Input/Output 78 (I/O/Z)<br>-<br>External Interface Data Line 1 (O) |
| GPIO79<br>-<br>XD0 | C14 | E16 | General-Purpose Input/Output 79 (I/O/Z)<br>-<br>External Interface Data Line 0 (O) |
| GPIO80<br>-<br>XA8 | E6 | B7 | General-Purpose Input/Output 80 (I/O/Z)<br>-<br>External Interface Address Line 8 (O) |
| GPIO81<br>-<br>XA9 | C5 | C7 | General-Purpose Input/Output 81 (I/O/Z)<br>-<br>External Interface Address Line 9 (O) |
| GPIO82<br>-<br>XA10 | A5 | B6 | General-Purpose Input/Output 82 (I/O/Z)<br>-<br>External Interface Address Line 10 (O) |
| GPIO83<br>-<br>XA11 | B5 | C6 | General-Purpose Input/Output 83 (I/O/Z)<br>-<br>External Interface Address Line 11 (O) |
| GPIO84<br>-<br>XA12 | D5 | A5 | General-Purpose Input/Output 84 (I/O/Z)<br>-<br>External Interface Address Line 12 (O) |
| GPIO85<br>-<br>XA13 | D4 | B5 | General-Purpose Input/Output 85 (I/O/Z)<br>-<br>External Interface Address Line 13 (O) |
| GPIO86<br>-<br>XA14 | A3 | C5 | General-Purpose Input/Output 86 (I/O/Z)<br>-<br>External Interface Address Line 14 (O) |
| GPIO87<br>-<br>XA15 | B3 | A4 | General-Purpose Input/Output 87 (I/O/Z)<br>-<br>External Interface Address Line 15 (O) |
| $\overline{\text{XRD}}$ | A14 | D15 | External Interface Read Enable (O) |
| $\overline{\text{XWE1}}$ | C13 | E14 | External Memory Interface Write Enable for Upper 16-bits (O) |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| **CPU and I/O Power Pins** | | | |
| $V_{DD18}$ | E8 | A6 | Oscillator and PLL Power Pin (1.8 V) |
| $V_{DD18}$ | C7 | A11 | |
| $V_{SSK}$ | B8 | A8 | Oscillator Kelvin Reference Ground. This pin should not be connected to Vss. See Figure 3-10 through Figure 3-12 for proper application board connections. |
| $V_{DD}$ | D1 | C1 | CPU and logic digital power pins (1.1 V/1.2 V) |
| $V_{DD}$ | E1 | C16 | |
| $V_{DD}$ | G1 | E6 | |
| $V_{DD}$ | K3 | E7 | |
| $V_{DD}$ | M1 | E8 | |
| $V_{DD}$ | N5 | E9 | |
| $V_{DD}$ | P7 | E10 | |
| $V_{DD}$ | J3 | E11 | |
| $V_{DD}$ | J4 | F5 | |
| $V_{DD}$ | K9 | F12 | |
| $V_{DD}$ | L10 | G5 | |
| $V_{DD}$ | N14 | G12 | |
| $V_{DD}$ | K11 | H5 | |
| $V_{DD}$ | H11 | H12 | |
| $V_{DD}$ | H14 | J5 | |
| $V_{DD}$ | G10 | J12 | |
| $V_{DD}$ | E12 | K3 | |
| $V_{DD}$ | D12 | K5 | |
| $V_{DD}$ | C11 | K12 | |
| $V_{DD}$ | C10 | L3 | |
| $V_{DD}$ | B7 | L5 | |
| $V_{DD}$ | C6 | L12 | |
| $V_{DD}$ | E5 | M6 | |
| $V_{DD}$ | C4 | M7 | |
| $V_{DD}$ | | M8 | |
| $V_{DD}$ | | M9 | |
| $V_{DD}$ | | M10 | |
| $V_{DD}$ | | M11 | |
| $V_{DD}$ | | P1 | |
| $V_{DD}$ | | P16 | |
| $V_{DDIO}$ | D3 | A3 | Digital I/O power pins (3.3 V) |
| $V_{DDIO}$ | F1 | A14 | |
| $V_{DDIO}$ | J1 | B9 | |
| $V_{DDIO}$ | L2 | D5 | |
| $V_{DDIO}$ | K5 | D6 | |
| $V_{DDIO}$ | K7 | D8 | |
| $V_{DDIO}$ | K8 | D11 | |
| $V_{DDIO}$ | P11 | D12 | |
| $V_{DDIO}$ | L14 | E4 | |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| $V_{DDIO}$ | J14 | E13 | |
| $V_{DDIO}$ | F11 | F4 | |
| $V_{DDIO}$ | D14 | F13 | |
| $V_{DDIO}$ | A11 | J1 | |
| $V_{DDIO}$ | C9 | J4 | |
| $V_{DDIO}$ | D7 | J13 | |
| $V_{DDIO}$ | B6 | J16 | |
| $V_{DDIO}$ | B4 | L4 | |
| $V_{DDIO}$ | | L13 | |
| $V_{DDIO}$ | | M4 | Digital I/O power pins |
| $V_{DDIO}$ | | M13 | |
| $V_{DDIO}$ | | N5 | |
| $V_{DDIO}$ | | N6 | |
| $V_{DDIO}$ | | N8 | |
| $V_{DDIO}$ | | N11 | |
| $V_{DDIO}$ | | N12 | |
| $V_{DDIO}$ | | R9 | |
| $V_{DDIO}$ | | T3 | |
| $V_{DDIO}$ | | T14 | |
| $V_{SS}$ | D2 | A1 | |
| $V_{SS}$ | F4 | A2 | |
| $V_{SS}$ | G5 | A10 | |
| $V_{SS}$ | H1 | A15 | |
| $V_{SS}$ | J2 | A16 | |
| $V_{SS}$ | K1 | B1 | |
| $V_{SS}$ | L3 | B2 | |
| $V_{SS}$ | L5 | B15 | |
| $V_{SS}$ | L7 | B16 | |
| $V_{SS}$ | L8 | C3 | |
| $V_{SS}$ | M9 | C4 | |
| $V_{SS}$ | K10 | C13 | |
| $V_{SS}$ | M12 | C14 | |
| $V_{SS}$ | J10 | D3 | Digital ground pins |
| $V_{SS}$ | H10 | D4 | |
| $V_{SS}$ | G14 | D7 | |
| $V_{SS}$ | G11 | D9 | |
| $V_{SS}$ | E14 | D10 | |
| $V_{SS}$ | D13 | D13 | |
| $V_{SS}$ | B11 | D14 | |
| $V_{SS}$ | E9 | E5 | |
| $V_{SS}$ | D8 | E12 | |
| $V_{SS}$ | A7 | F6 | |
| $V_{SS}$ | A6 | F7 | |
| $V_{SS}$ | A4 | F8 | |
| $V_{SS}$ | | F9 | |
| $V_{SS}$ | | F10 | |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| $V_{SS}$ | | F11 | Digital ground pins |
| $V_{SS}$ | | G4 | |
| $V_{SS}$ | | G6 | |
| $V_{SS}$ | | G7 | |
| $V_{SS}$ | | G8 | |
| $V_{SS}$ | | G9 | |
| $V_{SS}$ | | G10 | |
| $V_{SS}$ | | G11 | |
| $V_{SS}$ | | G13 | |
| $V_{SS}$ | | H4 | |
| $V_{SS}$ | | H6 | |
| $V_{SS}$ | | H7 | |
| $V_{SS}$ | | H8 | |
| $V_{SS}$ | | H9 | |
| $V_{SS}$ | | H10 | |
| $V_{SS}$ | | H11 | |
| $V_{SS}$ | | H13 | |
| $V_{SS}$ | | J6 | |
| $V_{SS}$ | | J7 | |
| $V_{SS}$ | | J8 | |
| $V_{SS}$ | | J9 | |
| $V_{SS}$ | | J10 | |
| $V_{SS}$ | | J11 | |
| $V_{SS}$ | | K1 | |
| $V_{SS}$ | | K4 | |
| $V_{SS}$ | | K6 | |
| $V_{SS}$ | | K7 | |
| $V_{SS}$ | | K8 | |
| $V_{SS}$ | | K9 | |
| $V_{SS}$ | | K10 | |
| $V_{SS}$ | | K11 | |
| $V_{SS}$ | | K13 | |
| $V_{SS}$ | | K16 | |
| $V_{SS}$ | | L6 | |
| $V_{SS}$ | | L7 | |
| $V_{SS}$ | | L8 | |
| $V_{SS}$ | | L9 | |
| $V_{SS}$ | | L10 | |
| $V_{SS}$ | | L11 | |
| $V_{SS}$ | | M5 | |
| $V_{SS}$ | | M12 | |
| $V_{SS}$ | | N4 | |
| $V_{SS}$ | | N7 | |
| $V_{SS}$ | | N9 | |
| $V_{SS}$ | | N10 | |
| $V_{SS}$ | | N13 | |

**Table 2-2. Signal Descriptions (continued)**

| NAME | ZHH BALL # | ZFE BALL # | DESCRIPTION |
|---|---|---|---|
| $V_{SS}$ | | N14 | Digital ground pins |
| $V_{SS}$ | | P3 | |
| $V_{SS}$ | | P4 | |
| $V_{SS}$ | | P14 | |
| $V_{SS}$ | | R1 | |
| $V_{SS}$ | | R2 | |
| $V_{SS}$ | | R15 | |
| $V_{SS}$ | | R16 | |
| $V_{SS}$ | | T1 | |
| $V_{SS}$ | | T2 | |
| $V_{SS}$ | | T15 | |
| $V_{SS}$ | | T16 | |

# 3 Functional Overview

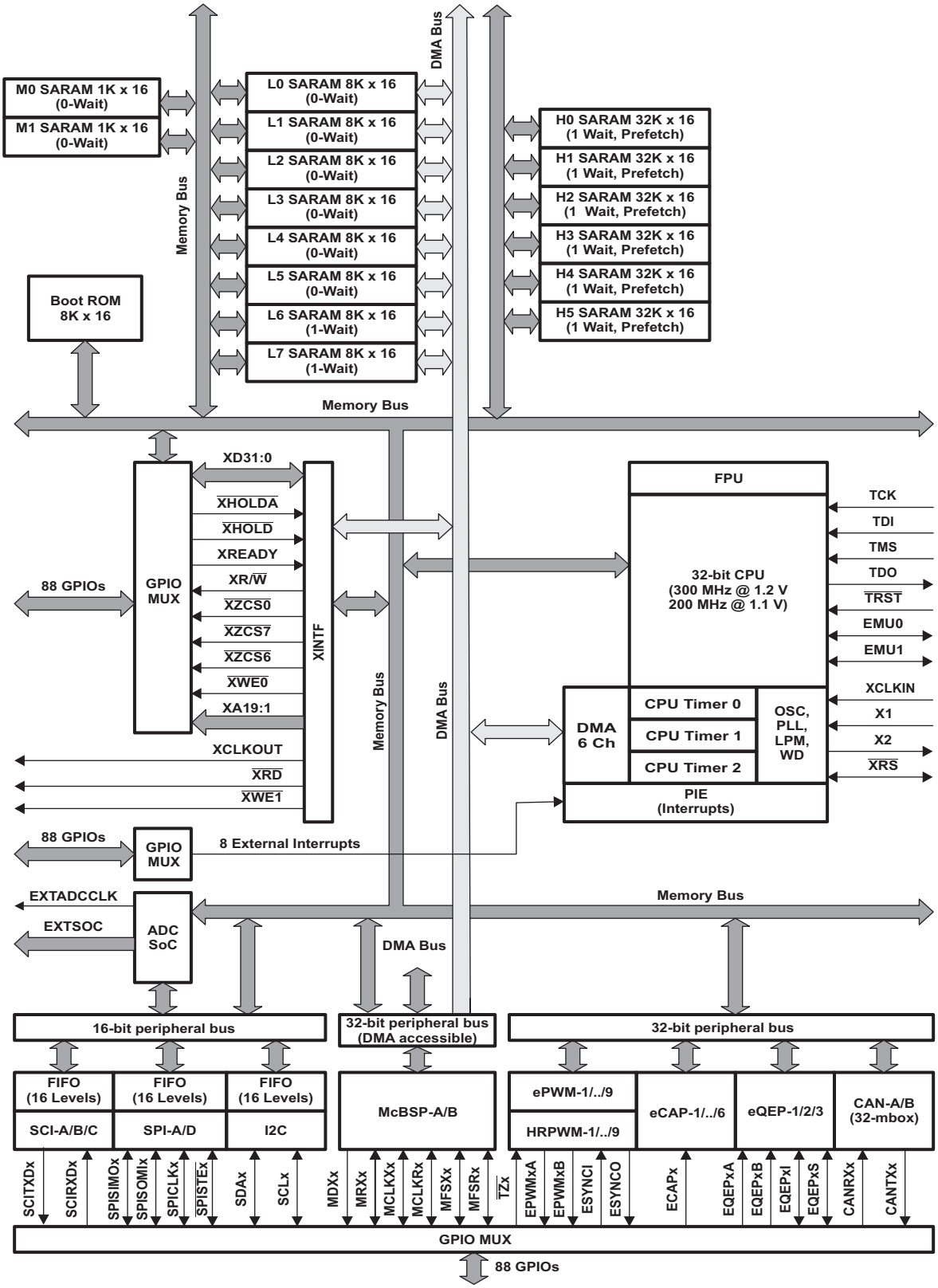
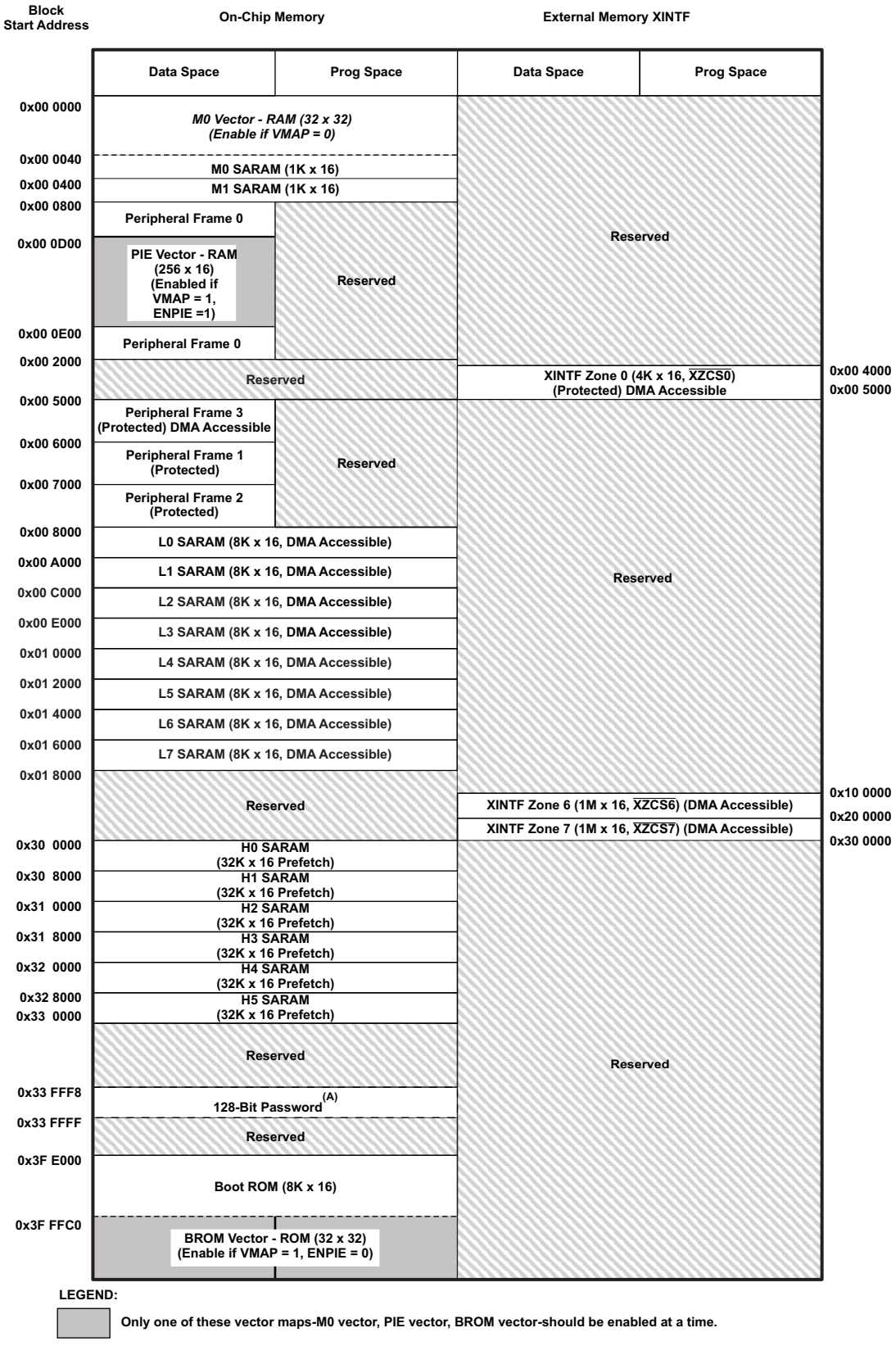



**Figure 3-1. Functional Block Diagram**

## 3.1 Memory Maps

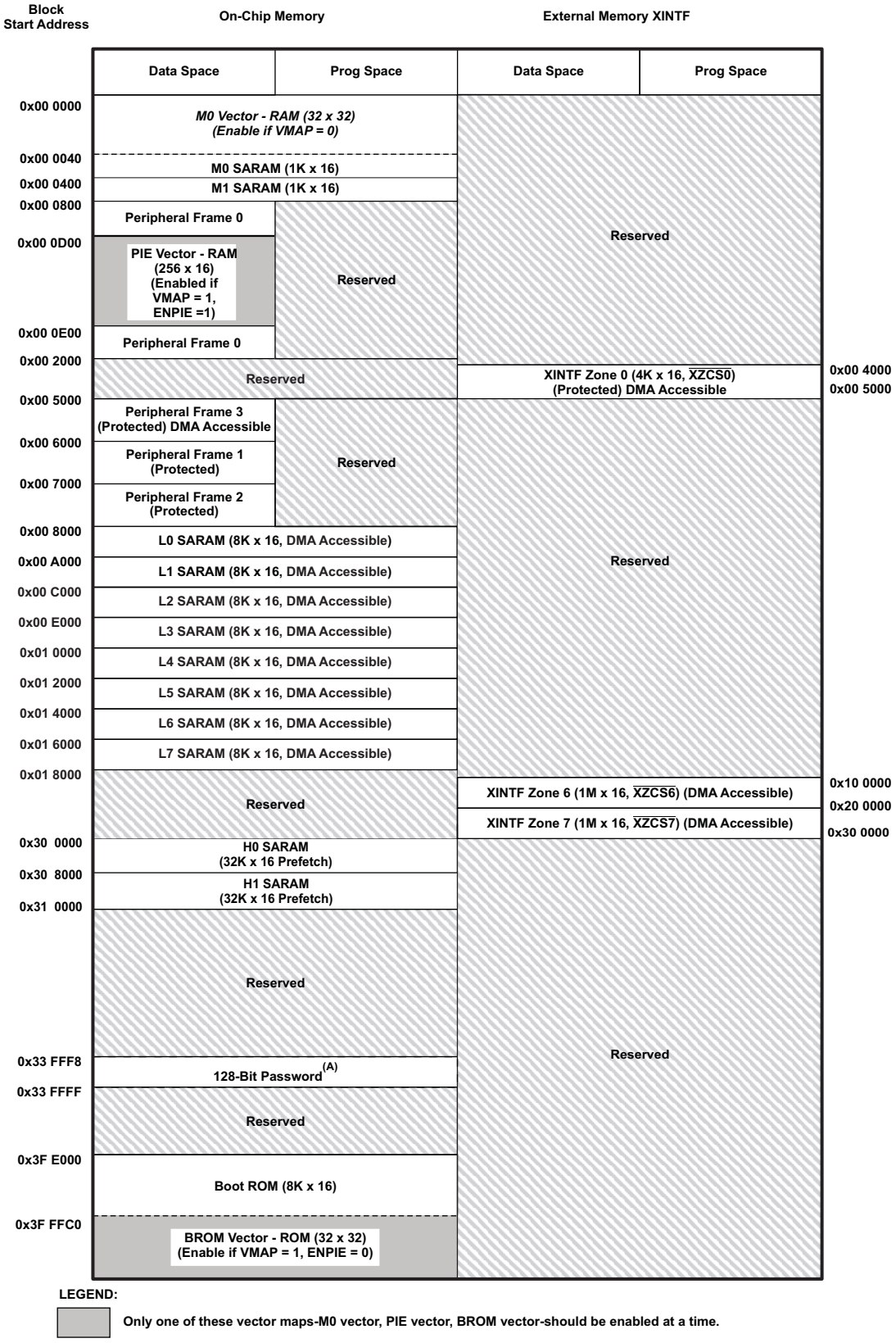
In Figure 3-2 through Figure 3-4, the following apply:

- Memory blocks are not to scale.
- Peripheral Frame 0, Peripheral Frame 1, Peripheral Frame 2, and Peripheral Frame 3 memory maps are restricted to data memory only. A user program cannot access these memory maps in program space.
- *Protected* means the order of "Write followed by Read" operations is preserved rather than the pipeline order. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1) for more details.
- Certain memory ranges are EALLOW protected against spurious writes after configuration.
- If the eCAN module is not used in an application, the RAM available (LAM, MOTS, MOTO, and mailbox RAM) can be used as general-purpose RAM. The CAN module clock should be enabled for this.



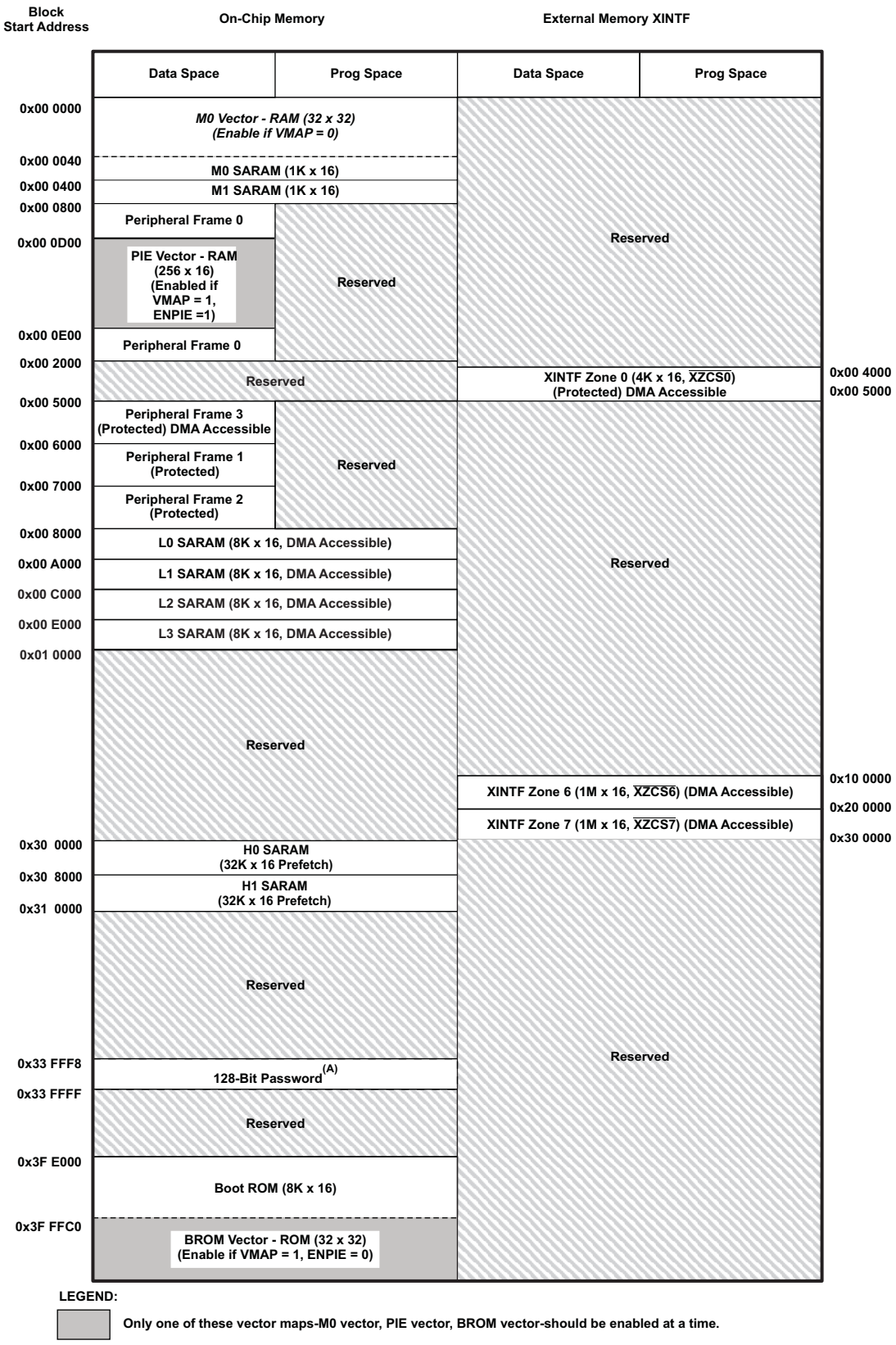

A. These locations support compatibility with legacy C28x designs only. See Section 3.2.9.

**Figure 3-2. C28346/C28345 Memory Map**




A. These locations support compatibility with legacy C28x designs only. See Section 3.2.9.

**Figure 3-3. C28344/C28343 Memory Map**




A. These locations support compatibility with legacy C28x designs only. See Section 3.2.9.

**Figure 3-4. C28342, C28341 Memory Map**

Peripheral Frame 1, Peripheral Frame 2, and Peripheral Frame 3 are grouped together to enable these blocks to be write/read peripheral block protected. The protected mode ensures that all accesses to these blocks happen as written. Because of the C28x pipeline, a write immediately followed by a read, to different memory locations, will appear in reverse order on the memory bus of the CPU. This can cause problems in certain peripheral applications where the user expected the write to occur first (as written). The C28x CPU supports a block protection mode where a region of memory can be protected so as to make sure that operations occur as written (the penalty is extra cycles are added to align the operations). This mode is programmable and by default, it will protect the selected zones.

The wait-states for the various spaces in the memory map area are listed in Table 3-1 .

**Table 3-1. Wait-states**

| AREA | WAIT-STATES (CPU) | WAIT-STATES (DMA)(1) | COMMENTS |
|---|---|---|---|
| M0 and M1 SARAMs | 0-wait | No access | Fixed |
| Peripheral Frame 0 | 0-wait (writes)<br>1-wait (reads) | No access (writes)<br>0-wait (reads) | |
| Peripheral Frame 3 | 0-wait (writes)<br>2-wait (reads) | 0-wait (writes)<br>1-wait (reads) | Assumes no conflicts between CPU and DMA. |
| Peripheral Frame 1 | 0-wait (writes)<br>2-wait (reads) | No access | Cycles can be extended by peripheral generated ready.<br>Consecutive writes to the CAN will experience a 1-cycle pipeline hit. |
| Peripheral Frame 2 | 0-wait (writes)<br>2-wait (reads) | No access | Fixed. Cycles cannot be extended by the peripheral. |
| L0 SARAM | 0-wait data and program | 1-wait | Assumes no CPU conflicts |
| L1 SARAM | | | |
| L2 SARAM | | | |
| L3 SARAM | | | |
| L4 SARAM | | | Assumes no conflicts between CPU and DMA |
| L5 SARAM | | | |
| L6 SARAM | 1-wait | | |
| L7 SARAM | | | |
| XINTF | Programmable | | Programmed via the XTIMING registers or extendable via external XREADY signal. |
| | 1-wait minimum | | 1-wait is minimum wait states allowed on external waveforms for both reads and writes on XINTF. |
| | 0-wait minimum writes with write buffer enabled | 0-wait data (write)<br>0-wait data (read) | 0-wait minimum for writes assumes write buffer enabled and not full.<br>Assumes no conflicts between CPU and DMA. When DMA and CPU attempt simultaneous conflict, 1-cycle delay is added for arbitration. |
| H0 SARAM | 1-wait | No access | A program-access prefetch mechanism is enabled on these memories to improve instruction fetch performance for linear code execution. |
| H1 SARAM | | | |
| H2 SARAM | | | |
| H3 SARAM | | | |
| H4 SARAM | | | |
| H5 SARAM | | | |
| Boot-ROM | 1-wait | No access | |

(1) The DMA has a base of 4 cycles/word.

## 3.2 Brief Descriptions

### 3.2.1 C28x CPU

The C2834x (C28x+FPU) family is a member of the TMS320C2000™ microcontroller (MCU) platform. The C28x+FPU based controllers have the same 32-bit fixed-point architecture as TI's existing C28x MCUs, but also include a single-precision (32-bit) IEEE 754 floating-point unit (FPU). It is a very efficient C/C++ engine, enabling users to develop their system control software in a high-level language. It also enables math algorithms to be developed using C/C++. The device is as efficient at DSP math tasks as it is at system control tasks. This efficiency removes the need for a second processor in many systems. The 32 x 32-bit MAC 64-bit processing capabilities enable the controller to handle higher numerical resolution problems efficiently. Add to this the fast interrupt response with automatic context save of critical registers, resulting in a device that is capable of servicing many asynchronous events with minimal latency. The device has an 8-level-deep protected pipeline with pipelined memory accesses. This pipelining enables it to execute at high speeds without resorting to expensive high-speed memories. Special branch-look-ahead hardware minimizes the latency for conditional discontinuities. Special store conditional operations further improve performance.

### 3.2.2 Memory Bus (Harvard Bus Architecture)

As with many MCU type devices, multiple busses are used to move data between the memories and peripherals and the CPU. The C28x memory bus architecture contains a program read bus, data read bus and data write bus. The program read bus consists of 22 address lines and 32 data lines. The data read and write busses consist of 32 address lines and 32 data lines each. The 32-bit-wide data busses enable single cycle 32-bit operations. The multiple bus architecture, commonly termed Harvard Bus, enables the C28x to fetch an instruction, read a data value and write a data value in a single cycle. All peripherals and memories attached to the memory bus will prioritize memory accesses. Generally, the priority of memory bus accesses can be summarized as follows:

| | | |
|---|---|---|
| Highest: | Data Writes | (Simultaneous data and program writes cannot occur on the memory bus.) |
| | Program Writes | (Simultaneous data and program writes cannot occur on the memory bus.) |
| | Data Reads | |
| | Program Reads | (Simultaneous program reads and fetches cannot occur on the memory bus.) |
| Lowest: | Fetches | (Simultaneous program reads and fetches cannot occur on the memory bus.) |

### 3.2.3 Peripheral Bus

To enable migration of peripherals between various Texas Instruments (TI) MCU family of devices, the C2834x devices adopt a peripheral bus standard for peripheral interconnect. The peripheral bus bridge multiplexes the various busses that make up the processor Memory Bus into a single bus consisting of 16 address lines and 16 or 32 data lines and associated control signals. Three versions of the peripheral bus are supported. One version supports only 16-bit accesses (called peripheral frame 2). Another version supports both 16- and 32-bit accesses (called peripheral frame 1). The third version supports DMA access and both 16- and 32-bit accesses (called peripheral frame 3).

### 3.2.4 Real-Time JTAG and Analysis

The C2834x devices implement the standard IEEE 1149.1 JTAG interface. Additionally, the devices support real-time mode of operation whereby the contents of memory, peripheral and register locations can be modified while the processor is running and executing code and servicing interrupts. The user can also single step through non-time critical code while enabling time-critical interrupts to be serviced without interference. The device implements the real-time mode in hardware within the CPU. This is a feature unique to the C2834x device, requiring no software monitor. Additionally, special analysis hardware is provided that allows setting of hardware breakpoint or data/address watch-points and generate various user-selectable break events when a match occurs.

### 3.2.5 External Interface (XINTF)

This asynchronous interface consists of 20 address lines, 32 data lines, and three chip-select lines. The chip-select lines are mapped to three external zones, Zones 0, 6, and 7. Each of the three zones can be programmed with a different number of wait states, strobe signal setup and hold timing and each zone can be programmed for extending wait states externally or not. The programmable wait-state, chip-select and programmable strobe timing enables glueless interface to external memories and peripherals.

### 3.2.6 M0, M1 SARAMs

All C2834x devices contain these two blocks of single access memory, each 1K × 16 in size. The stack pointer points to the beginning of block M1 on reset. The M0 and M1 blocks, like all other memory blocks on C28x devices, are mapped to both program and data space. Hence, the user can use M0 and M1 to execute code or for data variables. The partitioning is performed within the linker. The C28x device presents a unified memory map to the programmer. This makes for easier programming in high-level languages.

### 3.2.7 L0, L1, L2, L3, L4, L5, L6, L7 , H0, H1, H2, H3, H4, H5 SARAMs

The 2834x has up to 256K × 16 single-access RAM (SARAM) divided up into the following categories:

| | |
|---|---|
| L0/1/2/3/4/5 SARAM Blocks | Up to 48K × 16 of SARAM at all frequencies. Each block is 8K × 16. |
| L6/L7 SARAM Blocks | These 8K × 16 SARAM blocks are single-wait state at all frequencies. |
| H0/1/2/3/4/5 SARAM Blocks | H0–H5 are each 32K × 16 and 1-wait state at all frequencies. A program-access prefetch buffer is used to improve performance of linear code. |

All SARAM blocks are mapped to both program and data space. L0–L7 are accessible by both the CPU and the DMA (1 wait state).

### 3.2.8 Boot ROM

The Boot ROM is factory-programmed with boot-loading software. Boot-mode signals are provided to tell the bootloader software what boot mode to use on power up. The user can select to boot normally or to download new software from an external connection or to select boot software that is programmed in the internal ROM. The Boot ROM also contains standard tables, such as SIN/COS waveforms, for use in math related algorithms.

**Table 3-2. Boot Mode Selection**

| MODE | GPIO87/XA15 | GPIO86/XA14 | GPIO85/XA13 | GPIO84/XA12 | MODE (1) |
|---|---|---|---|---|---|
| F | 1 | 1 | 1 | 1 | Secure boot (2) |
| E | 1 | 1 | 1 | 0 | SCI-A boot |
| D | 1 | 1 | 0 | 1 | SPI-A boot |
| C | 1 | 1 | 0 | 0 | I2C-A boot Timing 1 |
| B | 1 | 0 | 1 | 1 | eCAN-A boot Timing 1 |
| A | 1 | 0 | 1 | 0 | McBSP-A boot |
| 9 | 1 | 0 | 0 | 1 | Jump to XINTF x16 |
| 8 | 1 | 0 | 0 | 0 | Reserved |
| 7 | 0 | 1 | 1 | 1 | eCAN-A boot Timing 2 |
| 6 | 0 | 1 | 1 | 0 | Parallel GPIO I/O boot |
| 5 | 0 | 1 | 0 | 1 | Parallel XINTF boot |
| 4 | 0 | 1 | 0 | 0 | Jump to SARAM |
| 3 | 0 | 0 | 1 | 1 | Branch to check boot mode |
| 2 | 0 | 0 | 1 | 0 | I2C-A boot Timing 2 |
| 1 | 0 | 0 | 0 | 1 | Reserved |
| 0 | 0 | 0 | 0 | 0 | TI Test Only |

(1) All four GPIO pins have an internal pullup.
(2) This mode is available on secure devices only. See Section 3.2.9, Security.

### 3.2.9 Security

The 128-bit password locations on these devices will always read back 0xFFFF. To preserve compatibility with other C28x designs with code security, the password locations at 0x33FFF8–0x33FFFF must be read after a device reset; otherwise, certain memory locations will be inaccessible. The Boot ROM code performs this read during startup. If during debug the Boot ROM is bypassed, then it is the responsibility of the application software to read the password locations after a reset.

**Custom Encryption: Activating the Code Security Module (CSM) and Emulation Code Security Logic (ECSL)**

Custom secure versions of these devices are available which enable the CSM and ECSL logic on these devices. In the custom version, the 128-bit password locations are set to a customer-chosen value, activating the Code Security Module (CSM), which protects the Hx RAM memories from unauthorized access. Additionally, a TI-generated AES decryption routine is embedded into an on-chip secure ROM, providing a method to secure application code that is stored externally. Contact TI at support@ti.com for more details.

### 3.2.10 Peripheral Interrupt Expansion (PIE) Block

The PIE block serves to multiplex numerous interrupt sources into a smaller set of interrupt inputs. The PIE block can support up to 96 peripheral interrupts. On the C2834x, 64 of the possible 96 interrupts are used by peripherals. The 96 interrupts are grouped into blocks of 8 and each group is fed into 1 of 12 CPU interrupt lines (INT1 to INT12). Each of the 96 interrupts is supported by its own vector stored in a dedicated RAM block that can be overwritten by the user. The vector is automatically fetched by the CPU on servicing the interrupt. It takes 8 CPU clock cycles to fetch the vector and save critical CPU registers. Hence the CPU can quickly respond to interrupt events. Prioritization of interrupts is controlled in hardware and software. Each individual interrupt can be enabled/disabled within the PIE block.

### 3.2.11 External Interrupts (XINT1–XINT7, XNMI)

The devices support eight masked external interrupts (XINT1–XINT7, XNMI). XNMI can be connected to the INT13 or NMI interrupt of the CPU. Each of the interrupts can be selected for negative, positive, or both negative and positive edge triggering and can also be enabled/disabled (including the XNMI). XINT1, XINT2, and XNMI also contain a 16-bit free running up counter, which is reset to zero when a valid interrupt edge is detected. This counter can be used to accurately time stamp the interrupt. Unlike the 281x devices, there are no dedicated pins for the external interrupts. XINT1 XINT2, and XNMI interrupts can accept inputs from GPIO0–GPIO31 pins. XINT3–XINT7 interrupts can accept inputs from GPIO32–GPIO63 pins.

### 3.2.12 Oscillator and PLL

The device can be clocked by an external oscillator or by a crystal attached to the on-chip oscillator circuit. A PLL is provided supporting up to 31 input-clock-scaling ratios. The PLL ratios can be changed on-the-fly in software, enabling the user to scale back on operating frequency if lower power operation is desired. Refer to the Electrical Specification section for timing details. The PLL block can be set in bypass mode.

### 3.2.13 Watchdog

The devices contain a watchdog timer. The user software must regularly reset the watchdog counter within a certain time frame; otherwise, the watchdog will generate a reset to the processor. The watchdog can be disabled if necessary.

### 3.2.14 Peripheral Clocking

The clocks to each individual peripheral can be enabled/disabled so as to reduce power consumption when a peripheral is not in use. Additionally, the system clock to the serial ports (except I2C and eCAN) blocks can be scaled relative to the CPU clock. This enables the timing of peripherals to be decoupled from increasing CPU clock speeds.

### 3.2.15 Low-Power Modes

The devices are full static CMOS devices. Three low-power modes are provided:

- IDLE: Place CPU into low-power mode. Peripheral clocks may be turned off selectively and only those peripherals that need to function during IDLE are left operating. An enabled interrupt from an active peripheral or the watchdog timer will wake the processor from IDLE mode.
- STANDBY: Turns off clock to CPU and peripherals. This mode leaves the oscillator and PLL functional. An external interrupt event will wake the processor and the peripherals. Execution begins on the next valid cycle after detection of the interrupt event
- HALT: Turns off the internal oscillator. This mode basically shuts down the device and places it in the lowest possible power consumption mode. A reset or external signal can wake the device from this mode.

### 3.2.16 Peripheral Frames 0, 1, 2, 3 (PFn)

The device segregates peripherals into three sections. The mapping of peripherals is as follows:

| | | |
|---|---|---|
| PF0: | PIE: | PIE Interrupt Enable and Control Registers Plus PIE Vector Table |
| | XINTF: | External Interface Registers |
| | DMA | DMA Registers |
| | Timers: | CPU-Timers 0, 1, 2 Registers |
| PF1: | eCAN: | eCAN Mailbox and Control Registers |
| | GPIO: | GPIO MUX Configuration and Control Registers |
| | ePWM: | Enhanced Pulse Width Modulator Module and Registers |
| | eCAP: | Enhanced Capture Module and Registers |
| | eQEP: | Enhanced Quadrature Encoder Pulse Module and Registers |
| PF2: | SYS: | System Control Registers |
| | SCI: | Serial Communications Interface (SCI) Control and RX/TX Registers |
| | SPI: | Serial Port Interface (SPI) Control and RX/TX Registers |
| | ADC: | External ADC Interface |
| | I2C: | Inter-Integrated Circuit Module and Registers |
| | XINT | External Interrupt Registers |
| PF3: | McBSP | Multichannel Buffered Serial Port Registers |

### 3.2.17 General-Purpose Input/Output (GPIO) Multiplexer

Most of the peripheral signals are multiplexed with general-purpose input/output (GPIO) signals. This enables the user to use a pin as GPIO if the peripheral signal or function is not used. On reset, GPIO pins are configured as inputs. The user can individually program each pin for GPIO mode or peripheral signal mode. For specific inputs, the user can also select the number of input qualification cycles. This is to filter unwanted noise glitches. The GPIO signals can also be used to bring the device out of specific low-power modes.

### 3.2.18 32-Bit CPU-Timers (0, 1, 2)

CPU-Timers 0, 1, and 2 are identical 32-bit timers with presettable periods and with 16-bit clock prescaling. The timers have a 32-bit count down register, which generates an interrupt when the counter reaches zero. The counter is decremented at the CPU clock speed divided by the prescale value setting. When the counter reaches zero, it is automatically reloaded with a 32-bit period value. CPU-Timer 2 is reserved for Real-Time OS (RTOS)/BIOS applications. It is connected to INT14 of the CPU. If DSP/BIOS is not being used, CPU-Timer 2 is available for general use. CPU-Timer 1 is for general use and can be connected to INT13 of the CPU. CPU-Timer 0 is also for general use and is connected to the PIE block.

### 3.2.19 Control Peripherals

The C2834x devices support the following peripherals which are used for embedded control and communication:

ePWM: The enhanced PWM peripheral supports independent/complementary PWM generation, adjustable dead-band generation for leading/trailing edges, latched/cycle-by-cycle trip mechanism. Some of the PWM pins support HRPWM features.

eCAP: The enhanced capture peripheral uses a 32-bit time base and registers up to four programmable events in continuous/one-shot capture modes.
This peripheral can also be configured to generate an auxiliary PWM signal.

eQEP: The enhanced QEP peripheral uses a 32-bit position counter, supports low-speed measurement using capture unit and high-speed measurement using a 32-bit unit timer.
This peripheral has a watchdog timer to detect motor stall and input error detection logic to identify simultaneous edge transition in QEP signals.

### 3.2.20 Serial Port Peripherals

The devices support the following serial communication peripherals:

eCAN: This is the enhanced version of the CAN peripheral. It supports 32 mailboxes, time stamping of messages, and is CAN 2.0B-compliant.

McBSP: The multichannel buffered serial port (McBSP) connects to E1/T1 lines, phone-quality codecs for modem applications or high-quality stereo audio DAC devices. The McBSP receive and transmit registers are supported by the DMA to significantly reduce the overhead for servicing this peripheral. Each McBSP module can be configured as an SPI as required.

SPI: The SPI is a high-speed, synchronous serial I/O port that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmable bit-transfer rate. Normally, the SPI is used for communications between the MCU and external peripherals or another processor. Typical applications include external I/O or peripheral expansion through devices such as shift registers, display drivers, and ADCs. Multi-device communications are supported by the master/slave operation of the SPI. The SPI contains a 16-level receive and transmit FIFO for reducing interrupt servicing overhead.

SCI: The serial communications interface is a two-wire asynchronous serial port, commonly known as UART. The SCI contains a 16-level receive and transmit FIFO for reducing interrupt servicing overhead.

I2C: The inter-integrated circuit (I2C) module provides an interface between an MCU and other devices compliant with Philips Semiconductors Inter-IC bus (I2C-bus) specification version 2.1 and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the MCU through the I2C module. The I2C contains a 16-level receive and transmit FIFO for reducing interrupt servicing overhead.

## 3.3 Register Map

The devices contain four peripheral register spaces. The spaces are categorized as follows:

Peripheral Frame 0: These are peripherals that are mapped directly to the CPU memory bus. See Table 3-3.

Peripheral Frame 1 These are peripherals that are mapped to the 32-bit peripheral bus. See Table 3-4.

Peripheral Frame 2: These are peripherals that are mapped to the 16-bit peripheral bus. See Table 3-5.

Peripheral Frame 3: These are peripherals that are mapped to the 32-bit DMA-accessible peripheral bus. See Table 3-6.

**Table 3-3. Peripheral Frame 0 Registers(1)**

| NAME | ADDRESS RANGE | SIZE (x16) | ACCESS TYPE(2) |
|---|---|---|---|
| Device Emulation Registers | 0x00 0880 – 0x00 09FF | 384 | EALLOW protected |
| Code Security Module Registers | 0x00 0AE0 – 0x00 0AEF | 16 | EALLOW protected |
| XINTF Registers | 0x00 0B20 – 0x00 0B3F | 32 | Not EALLOW protected |
| CPU–TIMER0/1/2 Registers | 0x00 0C00 – 0x00 0C3F | 64 | Not EALLOW protected |
| PIE Registers | 0x00 0CE0 – 0x00 0CFF | 32 | Not EALLOW protected |
| PIE Vector Table | 0x00 0D00 – 0x00 0DFF | 256 | EALLOW protected |
| DMA Registers | 0x00 1000 – 0x00 11FF | 512 | EALLOW protected |

(1) Registers in Frame 0 support 16-bit and 32-bit accesses.
(2) If registers are EALLOW protected, then writes cannot be performed until the EALLOW instruction is executed. The EDIS instruction disables writes to prevent stray code or pointers from corrupting register contents.

**Table 3-4. Peripheral Frame 1 Registers**

| NAME | ADDRESS RANGE | SIZE (x16) |
|---|---|---|
| eCAN-A Registers | 0x00 6000 – 0x00 61FF | 512 |
| eCAN-B Registers | 0x00 6200 – 0x00 63FF | 512 |
| ePWM1 + HRPWM1 registers | 0x00 6800 – 0x00 683F | 64 |
| ePWM2 + HRPWM2 registers | 0x00 6840 – 0x00 687F | 64 |
| ePWM3 + HRPWM3 registers | 0x00 6880 – 0x00 68BF | 64 |
| ePWM4 + HRPWM4 registers | 0x00 68C0 – 0x00 68FF | 64 |
| ePWM5 + HRPWM5 registers | 0x00 6900 – 0x00 693F | 64 |
| ePWM6 + HRPWM6 registers | 0x00 6940 – 0x00 697F | 64 |
| ePWM7 + HRPWM7 registers | 0x00 6980 – 0x00 69BF | 64 |
| ePWM8 + HRPWM8 registers | 0x00 69C0 – 0x00 69FF | 64 |
| ePWM9 + HRPWM9 registers | 0x00 6600 – 0x00 663F | 64 |
| eCAP1 registers | 0x00 6A00 – 0x00 6A1F | 32 |
| eCAP2 registers | 0x00 6A20 – 0x00 6A3F | 32 |
| eCAP3 registers | 0x00 6A40 – 0x00 6A5F | 32 |
| eCAP4 registers | 0x00 6A60 – 0x00 6A7F | 32 |
| eCAP5 registers | 0x00 6A80 – 0x00 6A9F | 32 |
| eCAP6 registers | 0x00 6AA0 – 0x00 6ABF | 32 |
| eQEP1 registers | 0x00 6B00 – 0x00 6B3F | 64 |
| eQEP2 registers | 0x00 6B40 – 0x00 6B7F | 64 |
| eQEP3 registers | 0x00 6B80 – 0x00 6BBF | 64 |
| GPIO registers | 0x00 6F80 – 0x00 6FFF | 128 |

**Table 3-5. Peripheral Frame 2 Registers**

| NAME | ADDRESS RANGE | SIZE (x16) |
|---|---|---|
| System Control Registers | 0x00 7010 – 0x00 702F | 32 |
| SPI-A Registers | 0x00 7040 – 0x00 704F | 16 |
| SCI-A Registers | 0x00 7050 – 0x00 705F | 16 |
| External Interrupt Registers | 0x00 7070 – 0x00 707F | 16 |
| SCI-B Registers | 0x00 7750 – 0x00 775F | 16 |
| SCI-C Registers | 0x00 7770 – 0x00 777F | 16 |
| SPI-D Registers | 0x00 7780 – 0x00 778F | 16 |
| I2C-A Registers | 0x00 7900 – 0x00 793F | 64 |

**Table 3-6. Peripheral Frame 3 Registers**

| NAME | ADDRESS RANGE | SIZE (x16) |
|---|---|---|
| McBSP-A Registers | 0x00 5000 – 0x00 503F | 64 |
| McBSP-B Registers | 0x00 5040 – 0x00 507F | 64 |

## 3.4 Device Emulation Registers

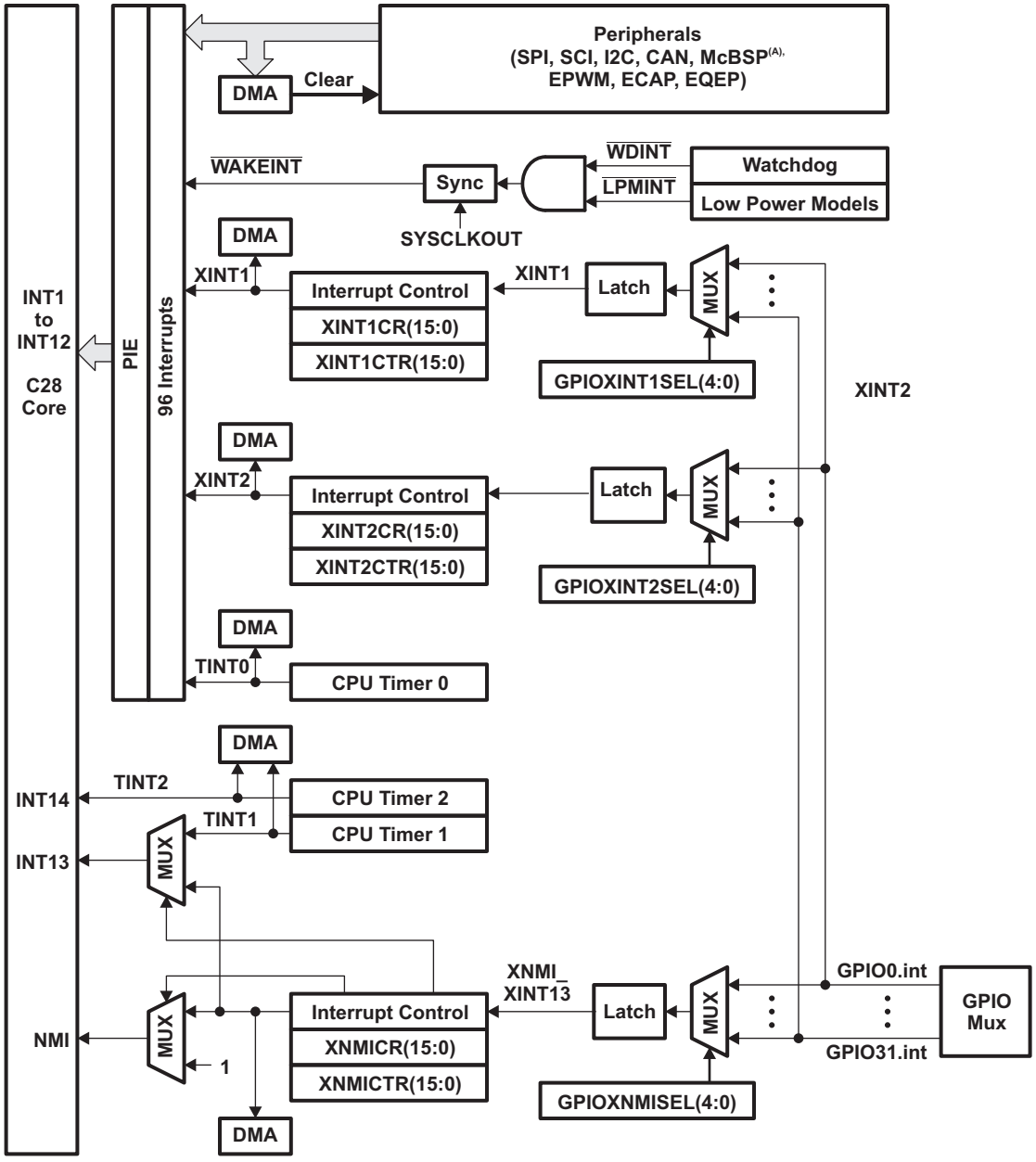
These registers are used to control the protection mode of the C28x CPU and to monitor some critical device signals. The registers are defined in Table 3-7.

**Table 3-7. Device Emulation Registers**

| NAME | ADDRESS RANGE | SIZE (x16) | DESCRIPTION | | |
|---|---|---|---|---|---|
| DEVICECNF | 0x0880<br>0x0881 | 2 | Device Configuration Register | | |
| PARTID | 0x0882 | 1 | Part ID Register | TMS320C28346 | 0xFFD0 |
| | | | | TMS320C28345 | 0xFFD1 |
| | | | | TMS320C28344 | 0xFFD2 |
| | | | | TMS320C28343 | 0xFFD3 |
| | | | | TMS320C28342 | 0xFFD4 |
| | | | | TMS320C28341 | 0xFFD5 |
| REVID | 0x0883 | 1 | Revision ID Register | 0x0000 - Silicon Rev. 0 - TMS | |
| PROTSTART | 0x0884 | 1 | Block Protection Start Address Register | | |
| PROTRANGE | 0x0885 | 1 | Block Protection Range Address Register | | |

## 3.5 Interrupts

Figure 3-5 shows how the various interrupt sources are multiplexed.




A. DMA-accessible

**Figure 3-5. External and PIE Interrupt Sources**

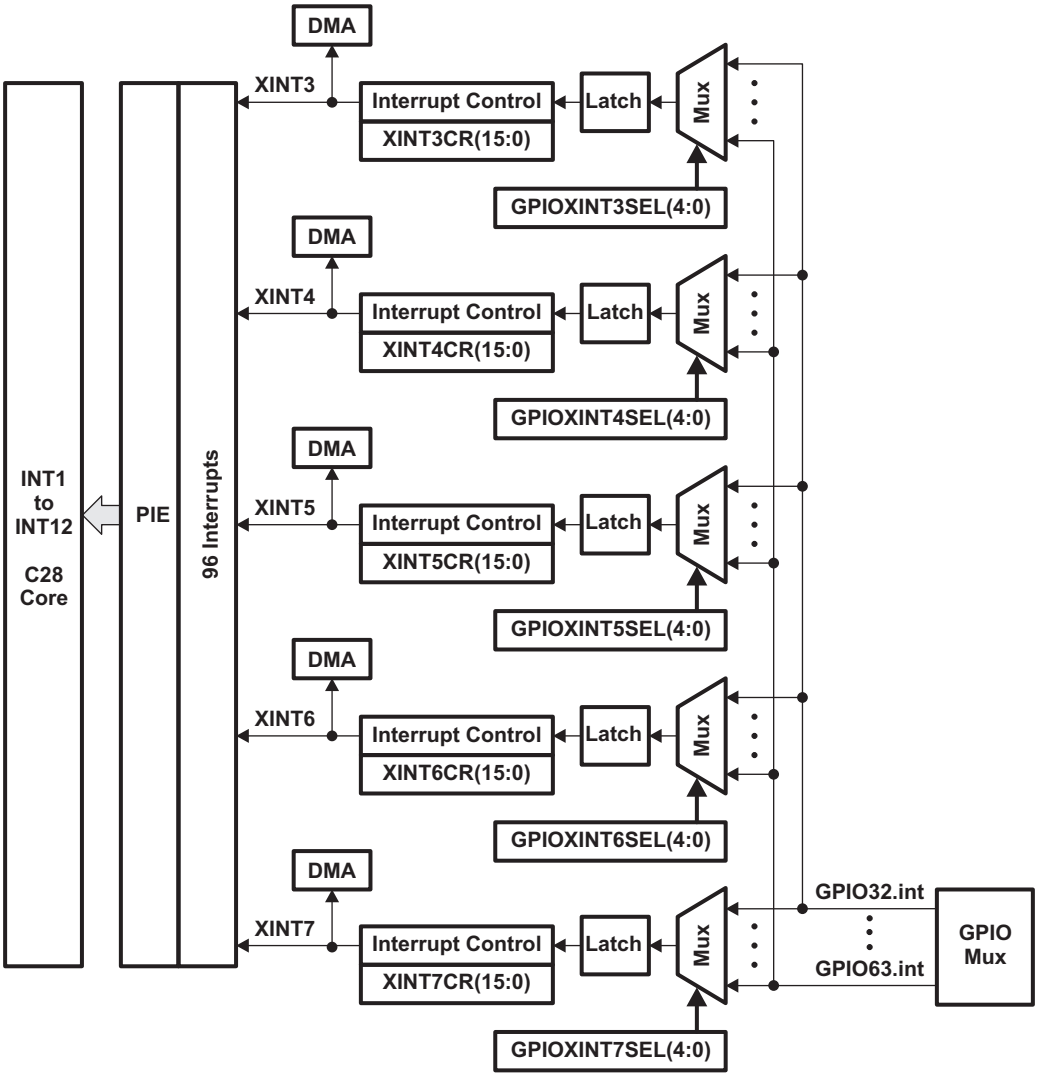



**Figure 3-6. External Interrupts**

Eight PIE block interrupts are grouped into one CPU interrupt. In total, 12 CPU interrupt groups, with 8 interrupts per group equals 96 possible interrupts. On the C2834x devices, 64 of these are used by peripherals as shown in Table 3-8.

The TRAP #VectorNumber instruction transfers program control to the interrupt service routine corresponding to the vector specified. TRAP #0 attempts to transfer program control to the address pointed to by the reset vector. The PIE vector table does not, however, include a reset vector. Therefore, TRAP #0 should not be used when the PIE is enabled. Doing so will result in undefined behavior.

When the PIE is enabled, TRAP #1 through TRAP #12 will transfer program control to the interrupt service routine corresponding to the first vector within the PIE group. For example: TRAP #1 fetches the vector from INT1.1, TRAP #2 fetches the vector from INT2.1, and so forth.

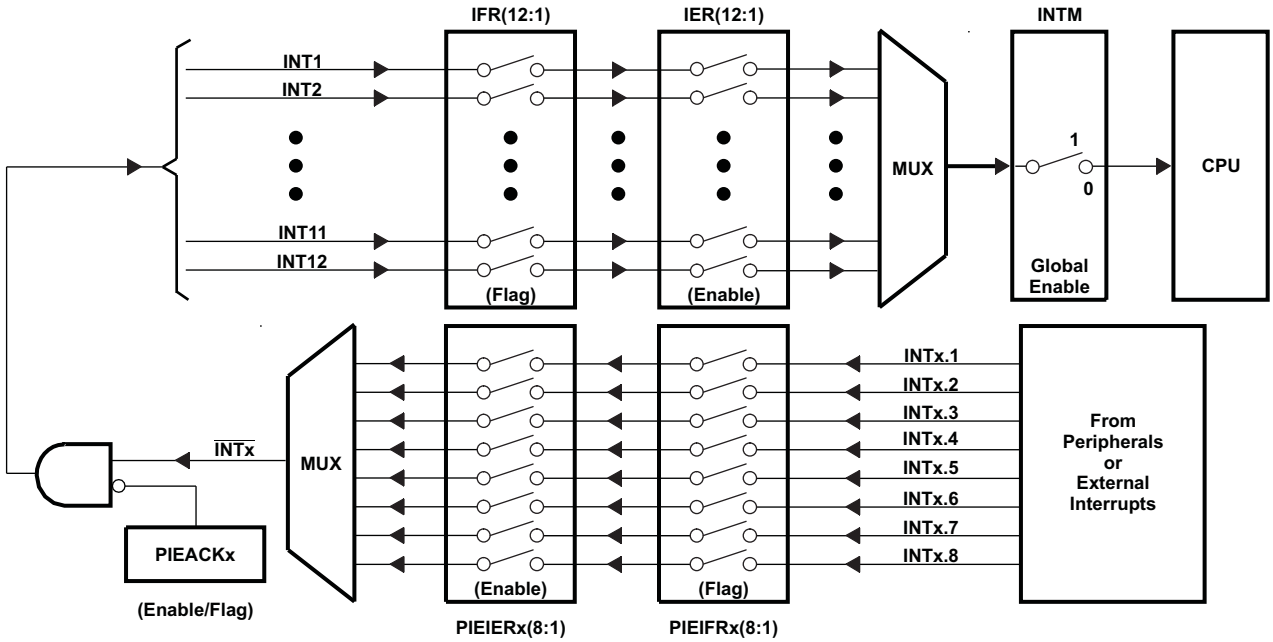



**Figure 3-7. Multiplexing of Interrupts Using the PIE Block**

**Table 3-8. PIE Peripheral Interrupts(1)**

| CPU INTERRUPTS | PIE INTERRUPTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | INTx.8 | INTx.7 | INTx.6 | INTx.5 | INTx.4 | INTx.3 | INTx.2 | INTx.1 |
| INT1 | WAKEINT (LPM/WD) | TINT0 (TIMER 0) | Reserved | XINT2 | XINT1 | Reserved | Reserved | Reserved |
| INT2 | EPWM8_TZINT (ePWM8) | EPWM7_TZINT (ePWM7) | EPWM6_TZINT (ePWM6) | EPWM5_TZINT (ePWM5) | EPWM4_TZINT (ePWM4) | EPWM3_TZINT (ePWM3) | EPWM2_TZINT (ePWM2) | EPWM1_TZINT (ePWM1) |
| INT3 | EPWM8_INT (ePWM8) | EPWM7_INT (ePWM7) | EPWM6_INT (ePWM6) | EPWM5_INT (ePWM5) | EPWM4_INT (ePWM4) | EPWM3_INT (ePWM3) | EPWM2_INT (ePWM2) | EPWM1_INT (ePWM1) |
| INT4 | Reserved | Reserved | ECAP6_INT (eCAP6) | ECAP5_INT (eCAP5) | ECAP4_INT (eCAP4) | ECAP3_INT (eCAP3) | ECAP2_INT (eCAP2) | ECAP1_INT (eCAP1) |
| INT5 | Reserved | Reserved | Reserved | Reserved | Reserved | EQEP3_INT (eQEP3) | EQEP2_INT (eQEP2) | EQEP1_INT (eQEP1) |
| INT6 | SPITXINTD (SPI-D) | SPIRXINTD (SPI-D) | MXINTA (McBSP-A) | MRINTA (McBSP-A) | MXINTB (McBSP-B) | MRINTB (McBSP-B) | SPITXINTA (SPI-A) | SPIRXINTA (SPI-A) |
| INT7 | Reserved | Reserved | DINTCH6 (DMA) | DINTCH5 (DMA) | DINTCH4 (DMA) | DINTCH3 (DMA) | DINTCH2 (DMA) | DINTCH1 (DMA) |
| INT8 | Reserved | Reserved | SCITXINTC (SCI-C) | SCIRXINTC (SCI-C) | Reserved | Reserved | I2CINT2A (I2C-A) | I2CINT1A (I2C-A) |
| INT9 | ECAN1_INTB (CAN-B) | ECAN0_INTB (CAN-B) | ECAN1_INTA (CAN-A) | ECAN0_INTA (CAN-A) | SCITXINTB (SCI-B) | SCIRXINTB (SCI-B) | SCITXINTA (SCI-A) | SCIRXINTA (SCI-A) |
| INT10 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | EPWM9_TZINT (ePWM9) |
| INT11 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | EPWM9_INT (ePWM9) |
| INT12 | LUF (FPU) | LVF (FPU) | Reserved | XINT7 | XINT6 | XINT5 | XINT4 | XINT3 |

(1) Out of the 96 possible interrupts, 64 interrupts are currently used. The remaining interrupts are reserved for future devices. These interrupts can be used as software interrupts if they are enabled at the PIEIFRx level, provided none of the interrupts within the group is being used by a peripheral. Otherwise, interrupts coming in from peripherals may be lost by accidentally clearing their flag while modifying the PIEIFR. To summarize, there is one sage case when the reserved interrupts could be used as software interrupts:
1) No peripheral within the group is asserting interrupts.

**Table 3-9. PIE Configuration and Control Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION(1) |
|---|---|---|---|
| PIECTRL | 0x0CE0 | 1 | PIE, Control Register |
| PIEACK | 0x0CE1 | 1 | PIE, Acknowledge Register |
| PIEIER1 | 0x0CE2 | 1 | PIE, INT1 Group Enable Register |
| PIEIFR1 | 0x0CE3 | 1 | PIE, INT1 Group Flag Register |
| PIEIER2 | 0x0CE4 | 1 | PIE, INT2 Group Enable Register |
| PIEIFR2 | 0x0CE5 | 1 | PIE, INT2 Group Flag Register |
| PIEIER3 | 0x0CE6 | 1 | PIE, INT3 Group Enable Register |
| PIEIFR3 | 0x0CE7 | 1 | PIE, INT3 Group Flag Register |
| PIEIER4 | 0x0CE8 | 1 | PIE, INT4 Group Enable Register |
| PIEIFR4 | 0x0CE9 | 1 | PIE, INT4 Group Flag Register |
| PIEIER5 | 0x0CEA | 1 | PIE, INT5 Group Enable Register |
| PIEIFR5 | 0x0CEB | 1 | PIE, INT5 Group Flag Register |
| PIEIER6 | 0x0CEC | 1 | PIE, INT6 Group Enable Register |
| PIEIFR6 | 0x0CED | 1 | PIE, INT6 Group Flag Register |
| PIEIER7 | 0x0CEE | 1 | PIE, INT7 Group Enable Register |
| PIEIFR7 | 0x0CEF | 1 | PIE, INT7 Group Flag Register |
| PIEIER8 | 0x0CF0 | 1 | PIE, INT8 Group Enable Register |
| PIEIFR8 | 0x0CF1 | 1 | PIE, INT8 Group Flag Register |
| PIEIER9 | 0x0CF2 | 1 | PIE, INT9 Group Enable Register |
| PIEIFR9 | 0x0CF3 | 1 | PIE, INT9 Group Flag Register |
| PIEIER10 | 0x0CF4 | 1 | PIE, INT10 Group Enable Register |
| PIEIFR10 | 0x0CF5 | 1 | PIE, INT10 Group Flag Register |
| PIEIER11 | 0x0CF6 | 1 | PIE, INT11 Group Enable Register |
| PIEIFR11 | 0x0CF7 | 1 | PIE, INT11 Group Flag Register |
| PIEIER12 | 0x0CF8 | 1 | PIE, INT12 Group Enable Register |
| PIEIFR12 | 0x0CF9 | 1 | PIE, INT12 Group Flag Register |
| Reserved | 0x0CFA – 0x0CFF | 6 | Reserved |

(1) The PIE configuration and control registers are not protected by EALLOW mode. The PIE vector table is protected.

### 3.5.1 External Interrupts

**Table 3-10. External Interrupt Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| XINT1CR | 0x00 7070 | 1 | XINT1 configuration register |
| XINT2CR | 0x00 7071 | 1 | XINT2 configuration register |
| XINT3CR | 0x00 7072 | 1 | XINT3 configuration register |
| XINT4CR | 0x00 7073 | 1 | XINT4 configuration register |
| XINT5CR | 0x00 7074 | 1 | XINT5 configuration register |
| XINT6CR | 0x00 7075 | 1 | XINT6 configuration register |
| XINT7CR | 0x00 7076 | 1 | XINT7 configuration register |
| XNMICR | 0x00 7077 | 1 | XNMI configuration register |
| XINT1CTR | 0x00 7078 | 1 | XINT1 counter register |
| XINT2CTR | 0x00 7079 | 1 | XINT2 counter register |
| Reserved | 0x707A – 0x707E | 5 | |
| XNMICTR | 0x00 707F | 1 | XNMI counter register |

Each external interrupt can be enabled/disabled or qualified using positive, negative, or both positive and negative edge. For more information, see the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1).

## 3.6 System Control

This section describes the oscillator, PLL and clocking mechanisms, the watchdog function and the low power modes. shows the various clock and reset domains that will be discussed.

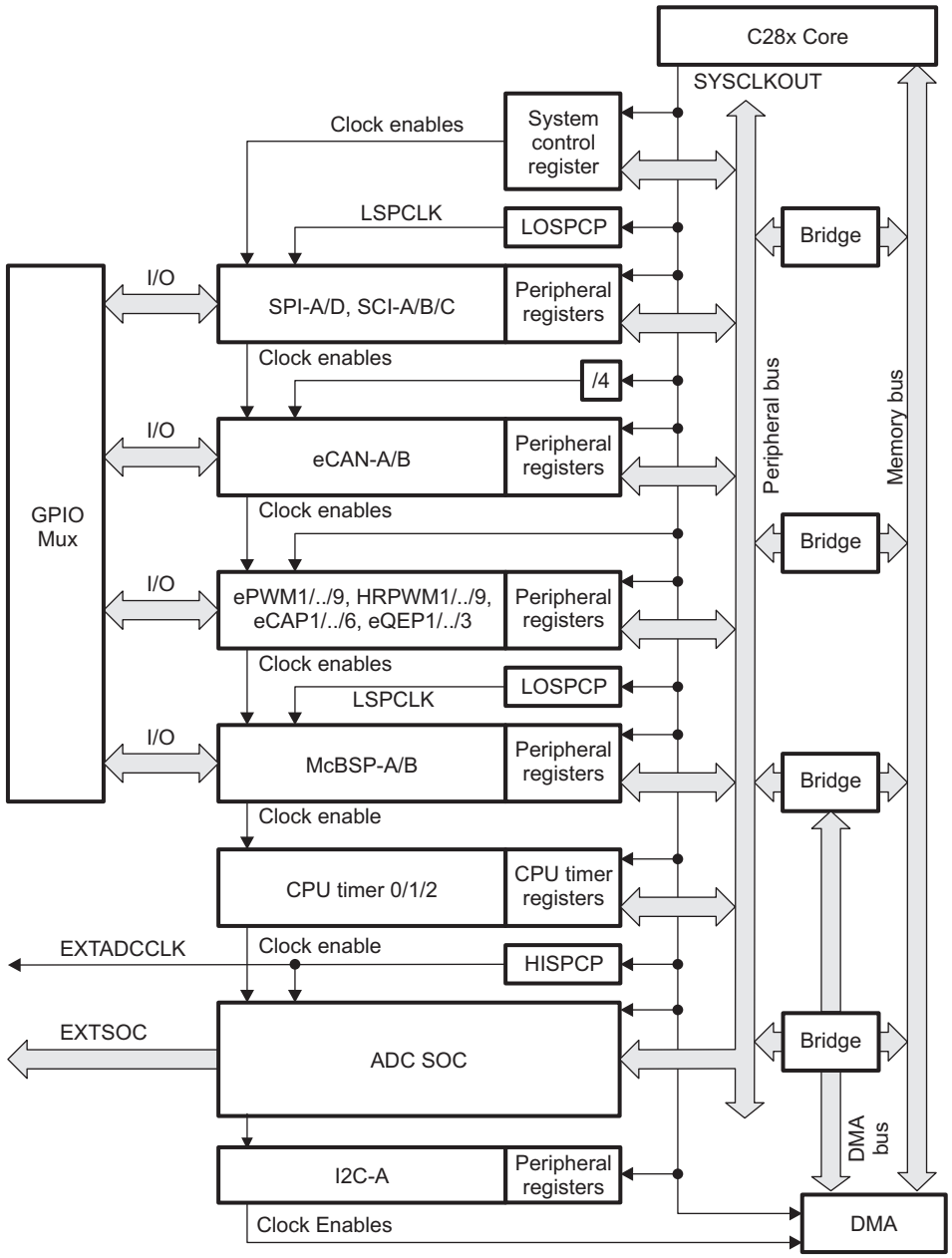



**Figure 3-8. Clock and Reset Domains**

**NOTE**

There is a 2-SYSCLKOUT cycle delay from when the write to PCLKCR0/1/2 registers (enables peripheral clocks) occurs to when the action is valid. This delay must be taken into account before attempting to access the peripheral configuration registers.

The PLL, clocking, watchdog and low-power modes, are controlled by the registers listed in Table 3-11.

**Table 3-11. PLL, Clocking, Watchdog, and Low-Power Mode Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| PLLSTS | 0x00 7011 | 1 | PLL Status Register |
| Reserved | 0x00 7012 – 0x00 7018 | 7 | Reserved |
| PCLKCR2 | 0x00 7019 | 1 | Peripheral Clock Control Register 2 |
| HISPCP | 0x00 701A | 1 | High-Speed Peripheral Clock Pre-Scaler Register |
| LOSPCP | 0x00 701B | 1 | Low-Speed Peripheral Clock Pre-Scaler Register |
| PCLKCR0 | 0x00 701C | 1 | Peripheral Clock Control Register 0 |
| PCLKCR1 | 0x00 701D | 1 | Peripheral Clock Control Register 1 |
| LPMCR0 | 0x00 701E | 1 | Low Power Mode Control Register 0 |
| Reserved | 0x00 701F | 1 | Reserved |
| PCLKCR3 | 0x00 7020 | 1 | Peripheral Clock Control Register 3 |
| PLLCR | 0x00 7021 | 1 | PLL Control Register |
| SCSR | 0x00 7022 | 1 | System Control and Status Register |
| WDCNTR | 0x00 7023 | 1 | Watchdog Counter Register |
| Reserved | 0x00 7024 | 1 | Reserved |
| WDKEY | 0x00 7025 | 1 | Watchdog Reset Key Register |
| Reserved | 0x00 7026 – 0x00 7028 | 3 | Reserved |
| WDCR | 0x00 7029 | 1 | Watchdog Control Register |
| Reserved | 0x00 702A – 0x00 702 C | 3 | Reserved |
| EXTSOCCFG | 0x00 702D | 1 | External ADC SOC Configuration Register |
| Reserved | 0x00 702E | 1 | Reserved |

### 3.6.1 OSC and PLL Block

Figure 3-9 shows the OSC and PLL block.

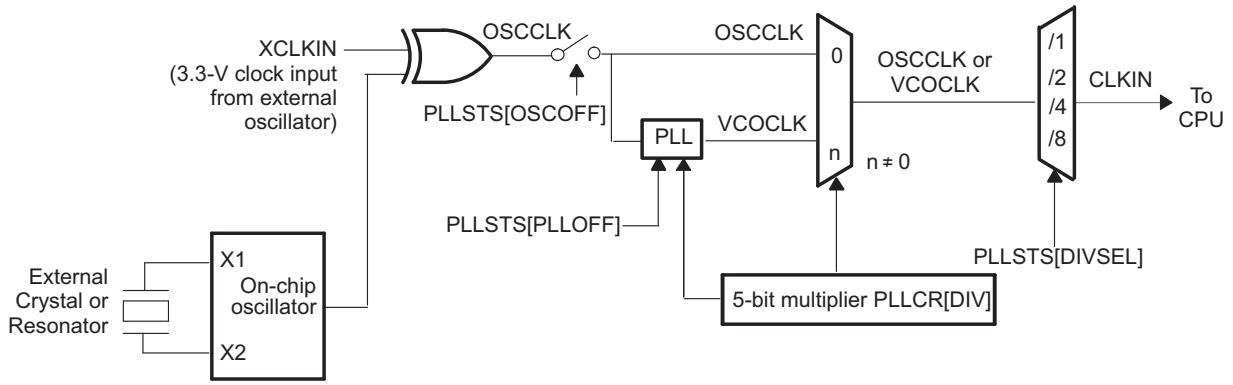



**Figure 3-9. OSC and PLL Block Diagram**

The on-chip oscillator circuit enables a crystal/resonator to be attached to the C2834x devices using the X1 and X2 pins. If the on-chip oscillator is not used, an external oscillator can be used in either one of the following configurations:

1. A 3.3-V external oscillator can be directly connected to the XCLKIN pin. The X2 pin should be left unconnected and the X1 pin tied to $V_{SSK}$ . The logic-high level in this case should not exceed $V_{DDIO}$.
2. A 1.8-V external oscillator can be directly connected to the X1 pin. The X2 pin should be left unconnected and the XCLKIN pin tied to $V_{SS}$ . The logic-high level in this case should not exceed $V_{DD18}$.

The three possible input-clock configurations are shown in Figure 3-10 through Figure 3-12.

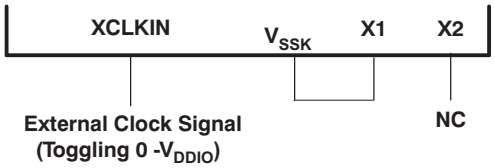



**Figure 3-10. Using a 3.3-V External Oscillator**

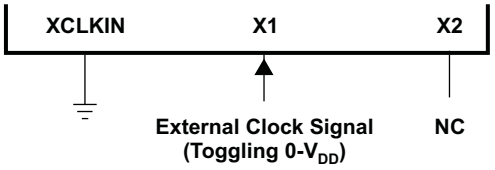



**Figure 3-11. Using a 1. 8-V External Oscillator**

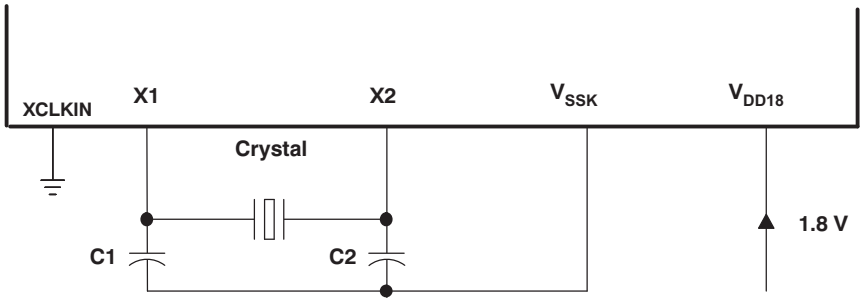



**Figure 3-12. Using the Internal Oscillator**

#### 3.6.1.1 External Reference Oscillator Clock Option

The on-chip oscillator requires an external crystal to be connected across the X1 and X2 pins.

The connection of the required circuit, consisting of the crystal and two load capacitors, is shown in Figure 3-12. The load capacitors, $C_1$ and $C_2$, must be chosen such that the equation below is satisfied (typical values are on the order of C1 = C2 = 10 pF). $C_L$ in the equation is the load specified for the crystal. All discrete components used to implement the oscillator circuit must be placed as close as possible to the associated oscillator pins (X1, X2, and $V_{SSK}$).

**NOTE**

The external crystal load capacitors must be connected only to the oscillator ground pin ($V_{SSK}$). Do not connect to board ground ($V_{SS}$).

$$C_L = \frac{C_1 C_2}{(C_1 + C_2)}$$

Where: $C_L$ equals the crystal load capacitance.

TI recommends that customers have the crystal vendor characterize the operation of their device with the MCU chip. The crystal vendor has the equipment and expertise to tune the crystal circuit. The vendor can also advise the customer regarding the proper component values that will produce proper start up and stability over the entire operating range.

### 3.6.1.2 PLL-Based Clock Module

The devices have an on-chip, PLL-based clock module. This module provides all the necessary clocking signals for the device, as well as control for low-power mode entry. The PLL has a 5-bit ratio control PLLCR[DIV] to select different CPU clock rates. The watchdog module should be disabled before writing to the PLLCR register. It can be re-enabled (if need be) after the PLL module has stabilized. The input clock and PLLCR[DIV] bits should be chosen in such a way that the output frequency of the PLL (VCOCLK) falls between 400 MHz and 600 MHz. The PLLSTS[DIVSEL] bit should be selected such that SYSCLKOUT(CLKIN) does not exceed the maximum operating frequency allowed for the device (300 MHz or 200 MHz). For example, suppose it is desired to operate a 300-MHz device at 100 MHz using a 20-MHz OSCCLK input (i.e., for power savings). The PLL should be configured for OSCCLK * 20, which produces VCOCLK = 400 MHz. PLLSTS[DIVSEL] should then be configured for /4 mode, resulting in the desired 100-MHz CLKIN to the CPU. The PLL should not be configured for OSCCLK * 10 with PLLSTS[DIVSEL] set for /2 mode. This combination would produce VCOCLK = 200 MHz, which does not fall within the required 400 MHz to 600 MHz range.

**Table 3-12. PLL Settings(1)**

| PLLCR[DIV] VALUE(2) (3) | PLLSTS[DIVSEL] = 0 | PLLSTS[DIVSEL] = 1 | SYSCLKOUT (CLKIN) | |
|---|---|---|---|---|
| | | | PLLSTS[DIVSEL] = 2 | PLLSTS[DIVSEL] = 3 (4) |
| 00000 (PLL bypass) | OSCCLK/8 (Default) | OSCCLK/4 | OSCCLK/2 | OSCCLK |
| 00001 | (OSCCLK * 2)/8 | (OSCCLK * 2)/4 | (OSCCLK * 2)/2 | – |
| 00010 | (OSCCLK * 3)/8 | (OSCCLK * 3)/4 | (OSCCLK * 3)/2 | – |
| 00011 | (OSCCLK * 4)/8 | (OSCCLK * 4)/4 | (OSCCLK * 4)/2 | – |
| 00100 | (OSCCLK * 5)/8 | (OSCCLK * 5)/4 | (OSCCLK * 5)/2 | – |
| 00101 | (OSCCLK * 6)/8 | (OSCCLK * 6)/4 | (OSCCLK * 6)/2 | – |
| 00110 | (OSCCLK * 7)/8 | (OSCCLK * 7)/4 | (OSCCLK * 7)/2 | – |
| 00111 | (OSCCLK * 8)/8 | (OSCCLK * 8)/4 | (OSCCLK * 8)/2 | – |
| 01000 | (OSCCLK * 9)/8 | (OSCCLK * 9)/4 | (OSCCLK * 9)/2 | – |
| 01001 | (OSCCLK * 10)/8 | (OSCCLK * 10)/4 | (OSCCLK * 10)/2 | – |
| 01010 | (OSCCLK * 11)/8 | (OSCCLK * 11)/4 | (OSCCLK * 11)/2 | – |
| 01011 – 11111 | (OSCCLK * 12)/8 – (OSCCLK * 32)/8 | (OSCCLK * 12)/4 – (OSCCLK * 32)/4 | (OSCCLK * 12)/2 – (OSCCLK * 32)/2 | – |

(1) PLLSTS[DIVSEL] must be 0 before writing to the PLLCR and must be set only to 1 or 2 after PLLSTS[PLLLOCKS] = 1. At reset, PLLSTS[DIVSEL] is configured for /8. The boot ROM changes this to /2 or /1, depending on the boot option.
(2) The PLL control register (PLLCR) and PLL Status Register (PLLSTS) are reset to their default state by the $\overline{\text{XRS}}$ signal or a watchdog reset only. A reset issued by the debugger or the missing clock detect logic have no effect.
(3) This register is EALLOW protected. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1) for more information.
(4) PLLSTS[DIVSEL] = 3 should be used only when the PLL is bypassed or off.

**Table 3-13. CLKIN Divide Options**

| PLLSTS [DIVSEL] | CLKIN DIVIDE |
|---|---|
| 0 | /8 |
| 1 | /4 |
| 2 | /2 |
| 3 | /1 |

The PLL-based clock module provides two modes of operation:

- Crystal-operation - This mode allows the use of an external crystal/resonator to provide the time base to the device.
- External clock source operation - This mode allows the internal oscillator to be bypassed. The device clocks are generated from an external clock source input on the X1 or the XCLKIN pin.

**Table 3-14. Possible PLL Configuration Modes**

| PLL MODE | REMARKS | PLLSTS[DIVSEL](1) | CLKIN AND SYSCLKOUT |
|---|---|---|---|
| PLL Off | Invoked by the user setting the PLLOFF bit in the PLLSTS register. The PLL block is disabled in this mode. This can be useful to reduce system noise and for low power operation. The PLLCR register must first be set to 0x0000 (PLL Bypass) before entering this mode. The CPU clock (CLKIN) is derived directly from the input clock on either X1/X2, X1 or XCLKIN. | 0<br>1<br>2<br>3 | OSCCLK/8<br>OSCCLK/4<br>OSCCLK/2<br>OSCCLK/1 |
| PLL Bypass | PLL Bypass is the default PLL configuration upon power-up or after an external reset (XRS). This mode is selected when the PLLCR register is set to 0x0000 or while the PLL locks to a new frequency after the PLLCR register has been modified. In this mode, the PLL itself is bypassed but the PLL is not turned off. | 0<br>1<br>2<br>3 | OSCCLK/8<br>OSCCLK/4<br>OSCCLK/2<br>OSCCLK/1 |
| PLL Enable | Achieved by writing a non-zero value n into the PLLCR register. Upon writing to the PLLCR the device will switch to PLL Bypass mode until the PLL locks. | 0<br>1<br>2<br>3 | OSCCLK*n/8<br>OSCCLK*n/4<br>OSCCLK*n/2<br>–(2) |

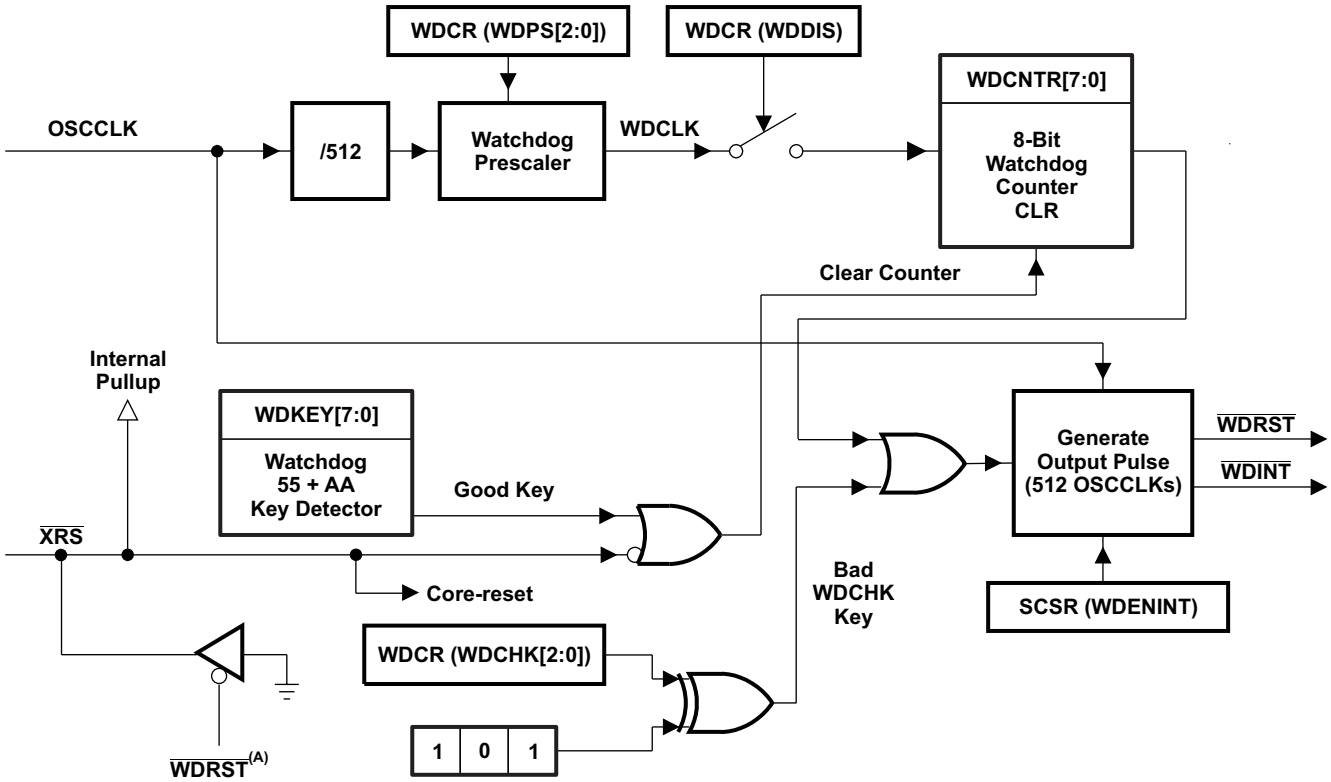
(1) PLLSTS[DIVSEL] must be 0 before writing to the PLLCR and must be set to 1 or 2 only after PLLSTS[PLLLOCKS] = 1. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1) for more information.
(2) PLLSTS[DIVSEL] should not be set to /1 mode while the PLL is enabled and not bypassed.

### 3.6.1.3 Loss of Input Clock

Applications in which the correct CPU operating frequency is absolutely critical should implement a mechanism by which the MCU will be held in reset, should the input clocks ever fail. For example, an R-C circuit may be used to trigger the XRS pin of the MCU, should the capacitor ever get fully charged. An I/O pin may be used to discharge the capacitor on a periodic basis to prevent it from getting fully charged.

### 3.6.2 Watchdog Block

The watchdog block on the C2834x device is similar to the one used on the 240x and 281x devices. The watchdog module generates an output pulse, 512 oscillator clocks wide (OSCCLK), whenever the 8-bit watchdog up counter has reached its maximum value. To prevent this, the user disables the counter or the software must periodically write a 0x55 + 0xAA sequence into the watchdog key register which will reset the watchdog counter. Figure 3-13 shows the various functional blocks within the watchdog module.




A. The WDRST signal is driven low for 512 OSCCLK cycles.

**Figure 3-13. Watchdog Module**

The WDINT signal enables the watchdog to be used as a wakeup from IDLE/STANDBY mode.

In STANDBY mode, all peripherals are turned off on the device. The only peripheral that remains functional is the watchdog. The WATCHDOG module will run off OSCCLK. The WDINT signal is fed to the LPM block so that it can wake the device from STANDBY (if enabled). See Section 3.7, Low-Power Modes Block, for more details.

In IDLE mode, the WDINT signal can generate an interrupt to the CPU, via the PIE, to take the CPU out of IDLE mode.

In HALT mode, this feature cannot be used because the oscillator (and PLL) are turned off and hence so is the WATCHDOG.

## 3.7 Low-Power Modes Block

The low-power modes on the C2834x devices are similar to the 240x devices. Table 3-15 summarizes the various modes.

**Table 3-15. Low-Power Modes**

| MODE | LPMCR0(1:0) | OSCCLK | CLKIN | SYSCLKOUT | EXIT(1) |
|---|---|---|---|---|---|
| IDLE | 00 | On | On | On(2) | $\overline{\text{XRS}}$, Watchdog interrupt, any enabled interrupt, XNMI |
| STANDBY | 01 | On (watchdog still running) | Off | Off | $\overline{\text{XRS}}$, Watchdog interrupt, GPIO Port A signal, debugger(3), XNMI |
| HALT | 1X | Off (oscillator and PLL turned off, watchdog not functional) | Off | Off | $\overline{\text{XRS}}$, GPIO Port A signal, XNMI, debugger(3) |

(1) The Exit column lists which signals or under what conditions the low power mode will be exited. A low signal, on any of the signals, will exit the low power condition. This signal must be kept low long enough for an interrupt to be recognized by the device. Otherwise, the low-power mode will not be exited and the device will go back into the indicated low power mode.
(2) The IDLE mode on the C28x behaves differently than on the 24x/240x. On the C28x, the clock output from the CPU (SYSCLKOUT) is still functional while on the 24x/240x the clock is turned off.
(3) On the C28x, the JTAG port can still function even if the CPU clock (CLKIN) is turned off.

The various low-power modes operate as follows:

IDLE Mode: This mode is exited by any enabled interrupt or an XNMI that is recognized by the processor. The LPM block performs no tasks during this mode as long as the LPMCR0(LPM) bits are set to 0,0.

STANDBY Mode: Any GPIO port A signal (GPIO[31:0]) can wake the device from STANDBY mode. The user must select which signal(s) will wake the device in the GPIOLPMSEL register. The selected signal(s) are also qualified by the OSCCLK before waking the device. The number of OSCCLKs is specified in the LPMCR0 register.

HALT Mode: Only the $\overline{\text{XRS}}$ and any GPIO port A signal (GPIO[31:0]) can wake the device from HALT mode. The user selects the signal in the GPIOLPMSEL register.

**NOTE**

The low-power modes do not affect the state of the output pins (PWM pins included). They will be in whatever state the code left them in when the IDLE instruction was executed. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1) for more details.

# 4 Peripherals

The integrated peripherals are described in the following subsections:

- 6-channel Direct Memory Access (DMA)
- Three 32-bit CPU-Timers
- Up to nine enhanced PWM modules (ePWM1, ePWM2, ePWM3, ePWM4, ePWM5, ePWM6, ePWM7, ePWM8, ePWM9)
- Up to six enhanced capture modules (eCAP1, eCAP2, eCAP3, eCAP4, eCAP5, eCAP6)
- Up to three enhanced QEP modules (eQEP1, eQEP2, eQEP3)
- External analog-to-digital converter (ADC) Interface
- Up to two enhanced controller area network (eCAN) modules (eCAN-A, eCAN-B)
- Up to three serial communications interface modules (SCI-A, SCI-B, SCI-C)
- Up to two serial peripheral interface (SPI) modules (SPI-A, SPI-D)
- Inter-integrated circuit module (I2C)
- Up to two multichannel buffered serial port (McBSP-A, McBSP-B) modules
- Digital I/O and shared pin functions
- External Interface (XINTF)

## 4.1 DMA Overview

Features:

- 6 Channels with independent PIE interrupts
- Trigger Sources:
  - McBSP-A and McBSP-B transmit and receive logic
  - XINT1–7 and XINT13
  - CPU Timers
  - Software
- Data Sources/Destinations:
  - L0–L7 64K × 16 SARAM
  - All XINTF zones
  - McBSP-A and McBSP-B transmit and receive buffers
- Word Size: 16-bit or 32-bit (McBSPs limited to 16-bit)
- Throughput: 4 cycles/word (5 cycles/word for McBSP reads)

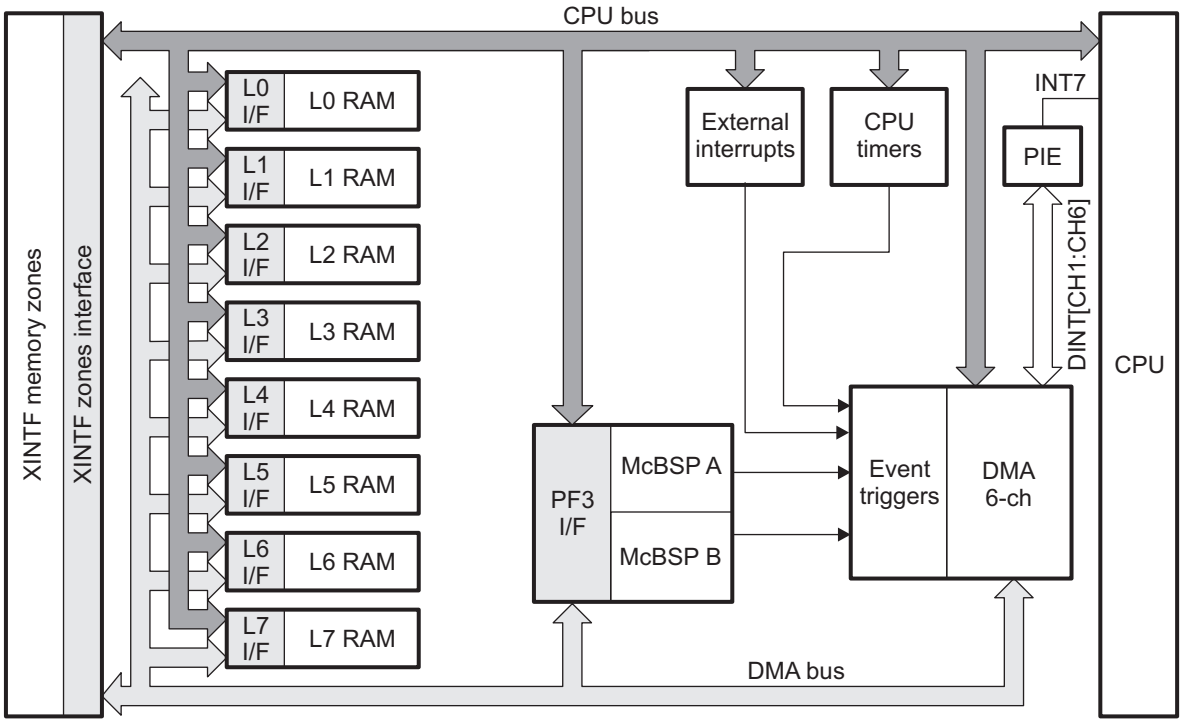



**Figure 4-1. DMA Functional Block Diagram**

## 4.2 32-Bit CPU-Timers 0/1/2

There are three 32-bit CPU-timers on the devices (CPU-TIMER0/1/2).

Timer 2 is reserved for DSP/BIOS™. CPU-Timer 0 and CPU-Timer 1 can be used in user applications. These timers are different from the timers that are present in the ePWM modules.

**NOTE**

NOTE: If the application is not using DSP/BIOS, then CPU-Timer 2 can be used in the application.

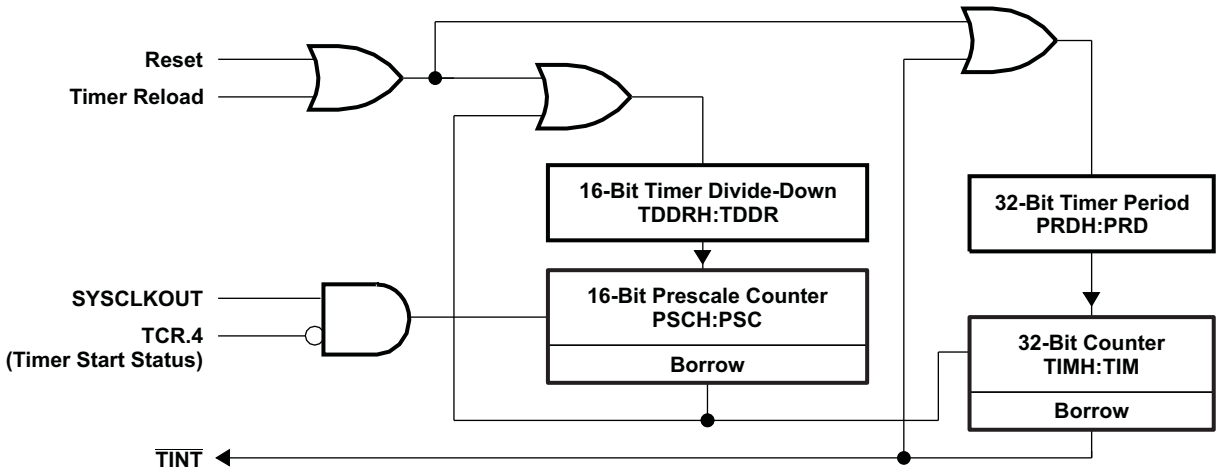
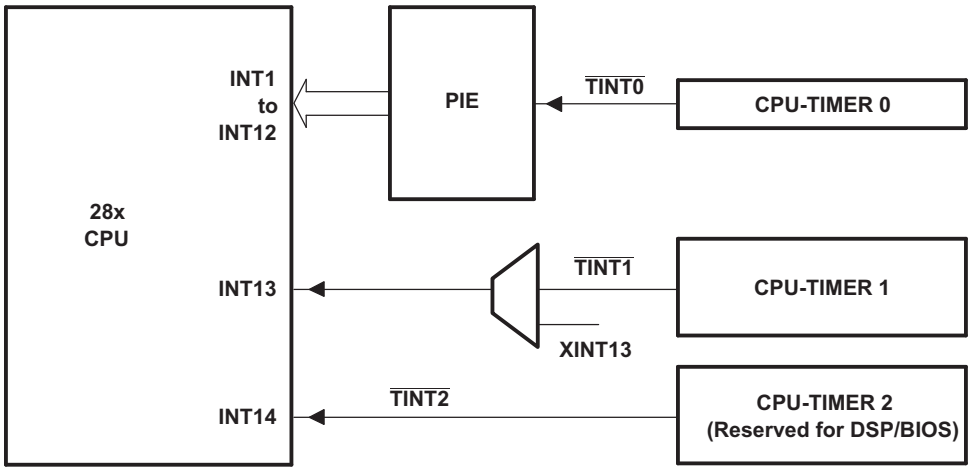



**Figure 4-2. CPU-Timers**

The timer interrupt signals (TINT0, TINT1, TINT2) are connected as shown in Figure 4-3.




A. The timer registers are connected to the memory bus of the C28x processor.
B. The timing of the timers is synchronized to SYSCLKOUT of the processor clock.

**Figure 4-3. CPU-Timer Interrupt Signals and Output Signal**

The general operation of the timer is as follows: The 32-bit counter register "TIMH:TIM" is loaded with the value in the period register "PRDH:PRD". The counter register decrements at the SYSCLKOUT rate of the C28x. When the counter reaches 0, a timer interrupt output signal generates an interrupt pulse. The registers listed in Table 4-1 are used to configure the timers. For more information, see the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1).

**Table 4-1. CPU-Timers 0, 1, 2 Configuration and Control Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| TIMER0TIM | 0x0C00 | 1 | CPU-Timer 0, Counter Register |
| TIMER0TIMH | 0x0C01 | 1 | CPU-Timer 0, Counter Register High |
| TIMER0PRD | 0x0C02 | 1 | CPU-Timer 0, Period Register |
| TIMER0PRDH | 0x0C03 | 1 | CPU-Timer 0, Period Register High |
| TIMER0TCR | 0x0C04 | 1 | CPU-Timer 0, Control Register |
| Reserved | 0x0C05 | 1 | |
| TIMER0TPR | 0x0C06 | 1 | CPU-Timer 0, Prescale Register |
| TIMER0TPRH | 0x0C07 | 1 | CPU-Timer 0, Prescale Register High |
| TIMER1TIM | 0x0C08 | 1 | CPU-Timer 1, Counter Register |
| TIMER1TIMH | 0x0C09 | 1 | CPU-Timer 1, Counter Register High |
| TIMER1PRD | 0x0C0A | 1 | CPU-Timer 1, Period Register |
| TIMER1PRDH | 0x0C0B | 1 | CPU-Timer 1, Period Register High |
| TIMER1TCR | 0x0C0C | 1 | CPU-Timer 1, Control Register |
| Reserved | 0x0C0D | 1 | |
| TIMER1TPR | 0x0C0E | 1 | CPU-Timer 1, Prescale Register |
| TIMER1TPRH | 0x0C0F | 1 | CPU-Timer 1, Prescale Register High |
| TIMER2TIM | 0x0C10 | 1 | CPU-Timer 2, Counter Register |
| TIMER2TIMH | 0x0C11 | 1 | CPU-Timer 2, Counter Register High |
| TIMER2PRD | 0x0C12 | 1 | CPU-Timer 2, Period Register |
| TIMER2PRDH | 0x0C13 | 1 | CPU-Timer 2, Period Register High |
| TIMER2TCR | 0x0C14 | 1 | CPU-Timer 2, Control Register |
| Reserved | 0x0C15 | 1 | |
| TIMER2TPR | 0x0C16 | 1 | CPU-Timer 2, Prescale Register |
| TIMER2TPRH | 0x0C17 | 1 | CPU-Timer 2, Prescale Register High |
| Reserved | 0x0C18 – 0x0C3F | 40 | |

## 4.3 Enhanced PWM Modules (ePWM1/2/3/4/5/6 /7/8/9)

The devices contain up to nine enhanced PWM Modules (ePWM). Figure 4-4 shows a block diagram of multiple ePWM modules. Figure 4-4 shows the signal interconnections with the ePWM.

Table 4-2 and Table 4-3 show the complete ePWM register set per module.

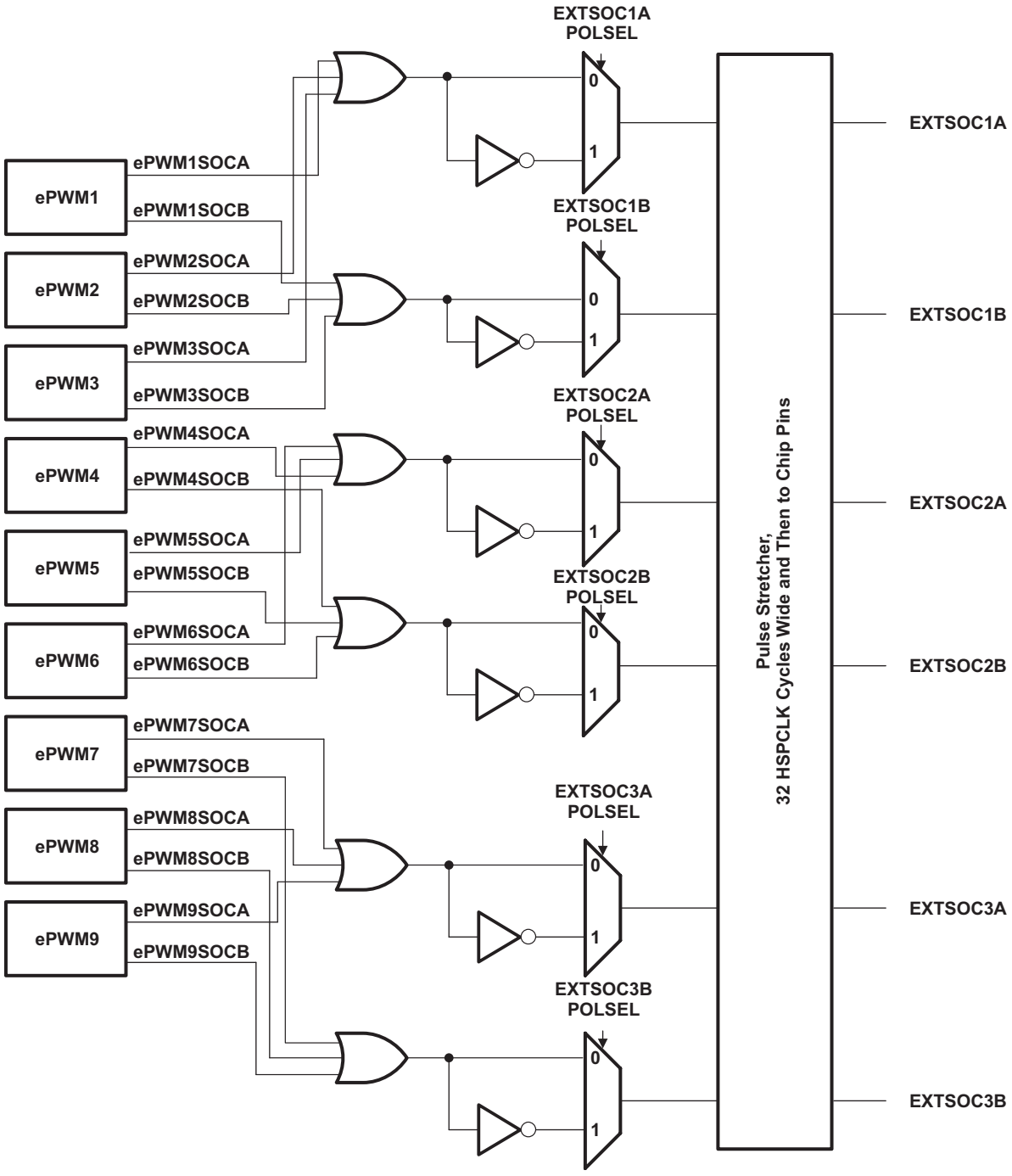



**Figure 4-4. Generation of SOC Pulses to the External ADC Module**

**Table 4-2. ePWM1-4 Control and Status Registers**

| NAME | ePWM1 | ePWM2 | ePWM3 | ePWM4 | SIZE (x16) / #SHADOW | DESCRIPTION |
|---|---|---|---|---|---|---|
| TBCTL | 0x6800 | 0x6840 | 0x6880 | 0x68C0 | 1 / 0 | Time Base Control Register |
| TBSTS | 0x6801 | 0x6841 | 0x6881 | 0x68C1 | 1 / 0 | Time Base Status Register |
| TBPHSHR | 0x6802 | 0x6842 | 0x6882 | 0x68C2 | 1 / 0 | Time Base Phase HRPWM Register |
| TBPHS | 0x6803 | 0x6843 | 0x6883 | 0x68C3 | 1 / 0 | Time Base Phase Register |
| TBCTR | 0x6804 | 0x6844 | 0x6884 | 0x68C4 | 1 / 0 | Time Base Counter Register |
| TBPRD | 0x6805 | 0x6845 | 0x6885 | 0x68C5 | 1 / 1 | Time Base Period Register Set |
| CMPCTL | 0x6807 | 0x6847 | 0x6887 | 0x68C7 | 1 / 0 | Counter Compare Control Register |
| CMPAHR | 0x6808 | 0x6848 | 0x6888 | 0x68C8 | 1 / 1 | Time Base Compare A HRPWM Register |
| CMPA | 0x6809 | 0x6849 | 0x6889 | 0x68C9 | 1 / 1 | Counter Compare A Register Set |
| CMPB | 0x680A | 0x684A | 0x688A | 0x68CA | 1 / 1 | Counter Compare B Register Set |
| AQCTLA | 0x680B | 0x684B | 0x688B | 0x68CB | 1 / 0 | Action Qualifier Control Register For Output A |
| AQCTLB | 0x680C | 0x684C | 0x688C | 0x68CC | 1 / 0 | Action Qualifier Control Register For Output B |
| AQSFRC | 0x680D | 0x684D | 0x688D | 0x68CD | 1 / 0 | Action Qualifier Software Force Register |
| AQCSFRC | 0x680E | 0x684E | 0x688E | 0x68CE | 1 / 1 | Action Qualifier Continuous S/W Force Register Set |
| DBCTL | 0x680F | 0x684F | 0x688F | 0x68CF | 1 / 1 | Dead-Band Generator Control Register |
| DBRED | 0x6810 | 0x6850 | 0x6890 | 0x68D0 | 1 / 0 | Dead-Band Generator Rising Edge Delay Count Register |
| DBFED | 0x6811 | 0x6851 | 0x6891 | 0x68D1 | 1 / 0 | Dead-Band Generator Falling Edge Delay Count Register |
| TZSEL | 0x6812 | 0x6852 | 0x6892 | 0x68D2 | 1 / 0 | Trip Zone Select Register |
| TZCTL | 0x6814 | 0x6854 | 0x6894 | 0x68D4 | 1 / 0 | Trip Zone Control Register |
| TZEINT | 0x6815 | 0x6855 | 0x6895 | 0x68D5 | 1 / 0 | Trip Zone Enable Interrupt Register |
| TZFLG | 0x6816 | 0x6856 | 0x6896 | 0x68D6 | 1 / 0 | Trip Zone Flag Register |
| TZCLR | 0x6817 | 0x6857 | 0x6897 | 0x68D7 | 1 / 0 | Trip Zone Clear Register |
| TZFRC | 0x6818 | 0x6858 | 0x6898 | 0x68D8 | 1 / 0 | Trip Zone Force Register |
| ETSEL | 0x6819 | 0x6859 | 0x6899 | 0x68D9 | 1 / 0 | Event Trigger Selection Register |
| ETPS | 0x681A | 0x685A | 0x689A | 0x68DA | 1 / 0 | Event Trigger Prescale Register |
| ETFLG | 0x681B | 0x685B | 0x689B | 0x68DB | 1 / 0 | Event Trigger Flag Register |
| ETCLR | 0x681C | 0x685C | 0x689C | 0x68DC | 1 / 0 | Event Trigger Clear Register |
| ETFRC | 0x681D | 0x685D | 0x689D | 0x68DD | 1 / 0 | Event Trigger Force Register |
| PCCTL | 0x681E | 0x685E | 0x689E | 0x68DE | 1 / 0 | PWM Chopper Control Register |
| HRCNFG | 0x6820 | 0x6860 | 0x68A0 | 0x68E0 | 1 / 0 | HRPWM Configuration Register(1) |

(1) Registers that are EALLOW protected.

**Table 4-3. ePWM5-9 Control and Status Registers**

| NAME | ePWM5 | ePWM6 | ePWM7 | ePWM8 | ePWM9 | SIZE (x16) / #SHADOW | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| TBCTL | 0x6900 | 0x6940 | 0x6980 | 0x69C0 | 0x6600 | 1 / 0 | Time Base Control Register |
| TBSTS | 0x6901 | 0x6941 | 0x6981 | 0x69C1 | 0x6601 | 1 / 0 | Time Base Status Register |
| TBPHSHR | 0x6902 | 0x6942 | 0x6982 | 0x69C2 | 0x6602 | 1 / 0 | Time Base Phase HRPWM Register |
| TBPHS | 0x6903 | 0x6943 | 0x6983 | 0x69C3 | 0x6603 | 1 / 0 | Time Base Phase Register |
| TBCTR | 0x6904 | 0x6944 | 0x6984 | 0x69C4 | 0x6604 | 1 / 0 | Time Base Counter Register |
| TBPRD | 0x6905 | 0x6945 | 0x6985 | 0x69C5 | 0x6605 | 1 / 1 | Time Base Period Register Set |
| CMPCTL | 0x6907 | 0x6947 | 0x6987 | 0x69C7 | 0x6607 | 1 / 0 | Counter Compare Control Register |
| CMPAHR | 0x6908 | 0x6948 | 0x6988 | 0x69C8 | 0x6608 | 1 / 1 | Time Base Compare A HRPWM Register |
| CMPA | 0x6909 | 0x6949 | 0x6989 | 0x69C9 | 0x6609 | 1 / 1 | Counter Compare A Register Set |
| CMPB | 0x690A | 0x694A | 0x698A | 0x69CA | 0x660A | 1 / 1 | Counter Compare B Register Set |
| AQCTLA | 0x690B | 0x694B | 0x698B | 0x69CB | 0x660B | 1 / 0 | Action Qualifier Control Register For Output A |
| AQCTLB | 0x690C | 0x694C | 0x698C | 0x69CC | 0x660C | 1 / 0 | Action Qualifier Control Register For Output B |
| AQSFRC | 0x690D | 0x694D | 0x698D | 0x69CD | 0x660D | 1 / 0 | Action Qualifier Software Force Register |
| AQCSFRC | 0x690E | 0x694E | 0x698E | 0x69CE | 0x660E | 1 / 1 | Action Qualifier Continuous S/W Force Register Set |
| DBCTL | 0x690F | 0x694F | 0x698F | 0x69CF | 0x660F | 1 / 1 | Dead-Band Generator Control Register |
| DBRED | 0x6910 | 0x6950 | 0x6990 | 0x69D0 | 0x6610 | 1 / 0 | Dead-Band Generator Rising Edge Delay Count Register |
| DBFED | 0x6911 | 0x6951 | 0x6991 | 0x69D1 | 0x6611 | 1 / 0 | Dead-Band Generator Falling Edge Delay Count Register |
| TZSEL | 0x6912 | 0x6952 | 0x6992 | 0x69D2 | 0x6612 | 1 / 0 | Trip Zone Select Register |
| TZCTL | 0x6914 | 0x6954 | 0x6994 | 0x69D4 | 0x6614 | 1 / 0 | Trip Zone Control Register |
| TZEINT | 0x6915 | 0x6955 | 0x6995 | 0x69D5 | 0x6615 | 1 / 0 | Trip Zone Enable Interrupt Register |
| TZFLG | 0x6916 | 0x6956 | 0x6996 | 0x69D6 | 0x6616 | 1 / 0 | Trip Zone Flag Register |
| TZCLR | 0x6917 | 0x6957 | 0x6997 | 0x69D7 | 0x6617 | 1 / 0 | Trip Zone Clear Register |
| TZFRC | 0x6918 | 0x6958 | 0x6998 | 0x69D8 | 0x6618 | 1 / 0 | Trip Zone Force Register |
| ETSEL | 0x6919 | 0x6959 | 0x6999 | 0x69D9 | 0x6619 | 1 / 0 | Event Trigger Selection Register |
| ETPS | 0x691A | 0x695A | 0x699A | 0x69DA | 0x661A | 1 / 0 | Event Trigger Prescale Register |
| ETFLG | 0x691B | 0x695B | 0x699B | 0x69DB | 0x661B | 1 / 0 | Event Trigger Flag Register |
| ETCLR | 0x691C | 0x695C | 0x699C | 0x69DC | 0x661C | 1 / 0 | Event Trigger Clear Register |
| ETFRC | 0x691D | 0x695D | 0x699D | 0x69DD | 0x661D | 1 / 0 | Event Trigger Force Register |
| PCCTL | 0x691E | 0x695E | 0x699E | 0x69DE | 0x661E | 1 / 0 | PWM Chopper Control Register |
| HRCNFG | 0x6920 | 0x6960 | 0x69A0 | 0x69E0 | 0x6620 | 1 / 0 | HRPWM Configuration Register(1) |

(1) Registers that are EALLOW protected.

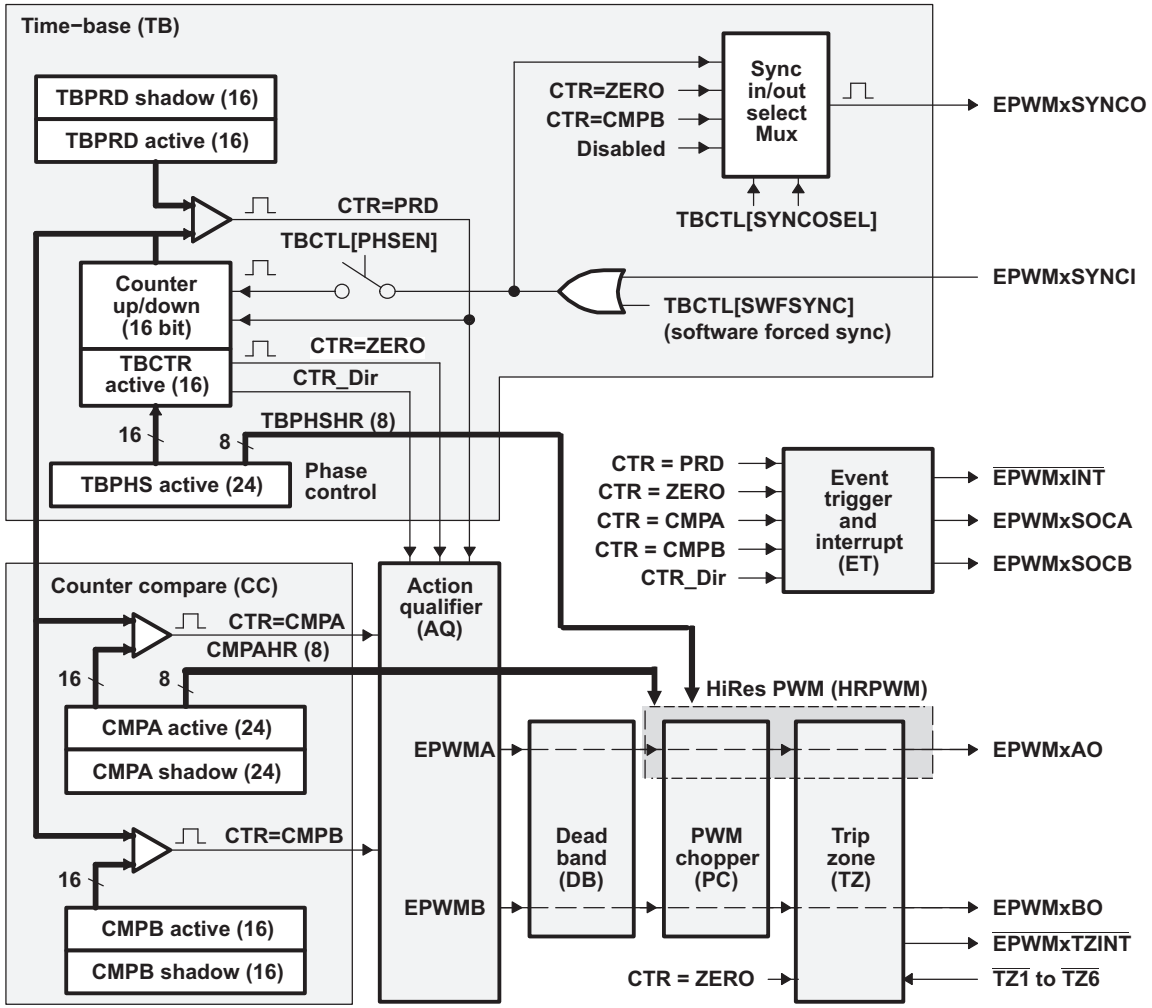



**Figure 4-5. ePWM Submodules Showing Critical Internal Signal Interconnections**

## 4.4 High-Resolution PWM (HRPWM)

The HRPWM module offers PWM resolution (time granularity) which is significantly better than what can be achieved using conventionally derived digital PWM methods. The key points for the HRPWM module are:

- Significantly extends the time resolution capabilities of conventionally derived digital PWM
- Typically used when effective PWM resolution falls below ~ 9–10 bits. This occurs at PWM frequencies greater than ~500 kHz when using a CPU/System clock of 300 MHz or ~375 kHz when using a CPU/system clock of 200 MHz.
- This capability can be utilized in both duty cycle and phase-shift control methods.
- Finer time granularity control or edge positioning is controlled via extensions to the Compare A and Phase registers of the ePWM module.
- HRPWM capabilities are offered only on the A signal path of an ePWM module (i.e., on the EPWMxA output). EPWMxB output has conventional PWM capabilities.

## 4.5 Enhanced CAP Modules (eCAP1/2/3/4/5/6)

The device contains up to six enhanced capture (eCAP) modules. Figure 4-6 shows a functional block diagram of a module.

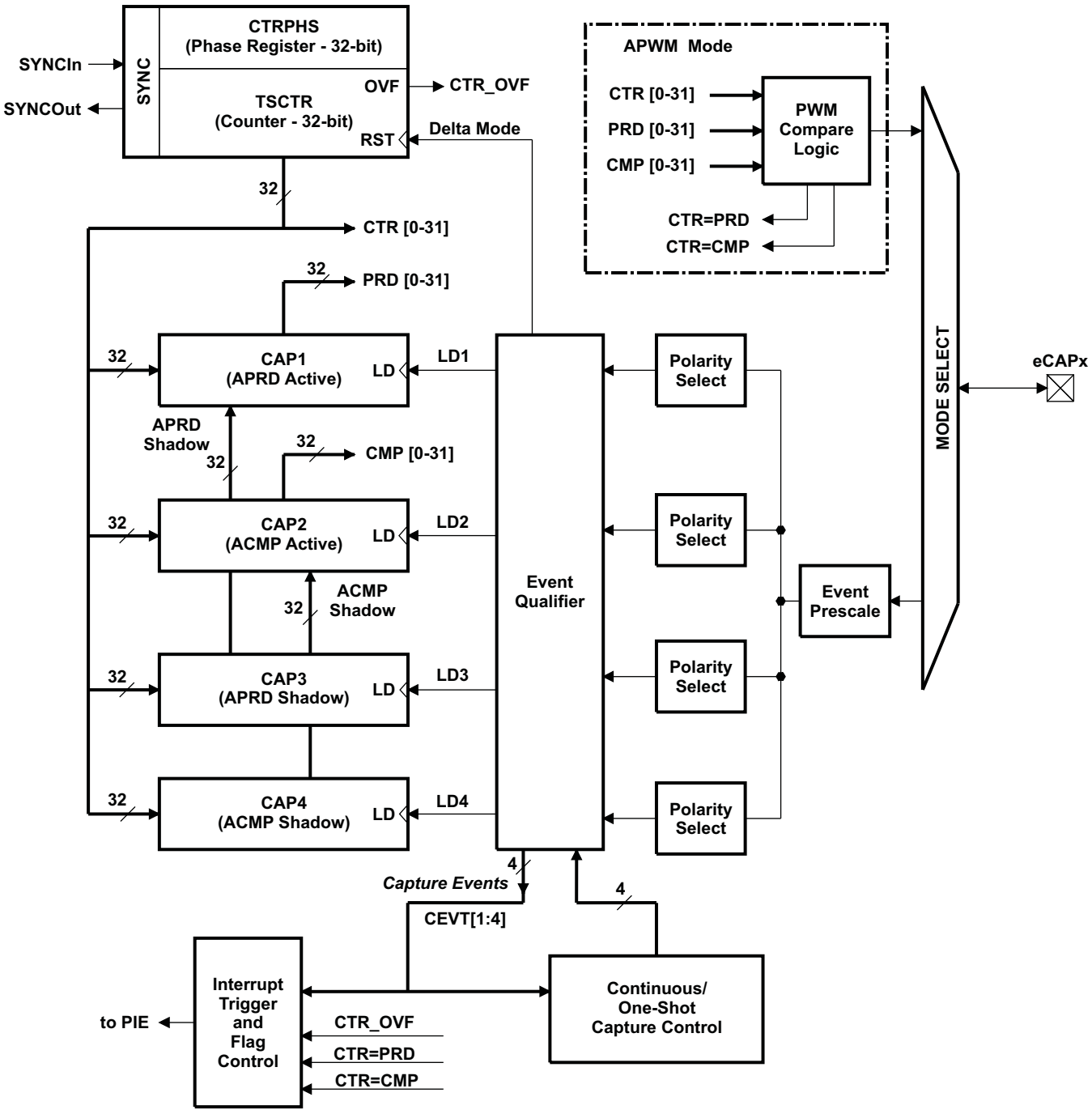



**Figure 4-6. eCAP Functional Block Diagram**

The eCAP modules are clocked at the SYSCLKOUT rate.

The clock enable bits (ECAP1/2/3/4/5/6ENCLK) in the PCLKCR1 register are used to turn off the eCAP modules individually (for low power operation). Upon reset, ECAP1ENCLK, ECAP2ENCLK, ECAP3ENCLK, ECAP4ENCLK, ECAP5ENCLK, and ECAP6ENCLK are set to low, indicating that the peripheral clock is off.

**Table 4-4. eCAP Control and Status Registers**

| NAME | eCAP1 | eCAP2 | eCAP3 | eCAP4 | eCAP5 | eCAP6 | SIZE (x16) | DESCRIPTION |
|---|---|---|---|---|---|---|---|---|
| TSCTR | 0x6A00 | 0x6A20 | 0x6A40 | 0x6A60 | 0x6A80 | 0x6AA0 | 2 | Time-Stamp Counter |
| CTRPHS | 0x6A02 | 0x6A22 | 0x6A42 | 0x6A62 | 0x6A82 | 0x6AA2 | 2 | Counter Phase Offset Value Register |
| CAP1 | 0x6A04 | 0x6A24 | 0x6A44 | 0x6A64 | 0x6A84 | 0x6AA4 | 2 | Capture 1 Register |
| CAP2 | 0x6A06 | 0x6A26 | 0x6A46 | 0x6A66 | 0x6A86 | 0x6AA6 | 2 | Capture 2 Register |
| CAP3 | 0x6A08 | 0x6A28 | 0x6A48 | 0x6A68 | 0x6A88 | 0x6AA8 | 2 | Capture 3 Register |
| CAP4 | 0x6A0A | 0x6A2A | 0x6A4A | 0x6A6A | 0x6A8A | 0x6AAA | 2 | Capture 4 Register |
| Reserved | 0x6A0C-0x6A12 | 0x6A2C-0x6A32 | 0x6A4C-0x6A52 | 0x6A6C-0x6A72 | 0x6A8C-0x6A92 | 0x6AAC-0x6AB2 | 8 | Reserved |
| ECCTL1 | 0x6A14 | 0x6A34 | 0x6A54 | 0x6A74 | 0x6A94 | 0x6AB4 | 1 | Capture Control Register 1 |
| ECCTL2 | 0x6A15 | 0x6A35 | 0x6A55 | 0x6A75 | 0x6A95 | 0x6AB5 | 1 | Capture Control Register 2 |
| ECEINT | 0x6A16 | 0x6A36 | 0x6A56 | 0x6A76 | 0x6A96 | 0x6AB6 | 1 | Capture Interrupt Enable Register |
| ECFLG | 0x6A17 | 0x6A37 | 0x6A57 | 0x6A77 | 0x6A97 | 0x6AB7 | 1 | Capture Interrupt Flag Register |
| ECCLR | 0x6A18 | 0x6A38 | 0x6A58 | 0x6A78 | 0x6A98 | 0x6AB8 | 1 | Capture Interrupt Clear Register |
| ECFRC | 0x6A19 | 0x6A39 | 0x6A59 | 0x6A79 | 0x6A99 | 0x6AB9 | 1 | Capture Interrupt Force Register |
| Reserved | 0x6A1A-0x6A1F | 0x6A3A-0x6A3F | 0x6A5A-0x6A5F | 0x6A7A-0x6A7F | 0x6A9A-0x6A9F | 0x6ABA-0x6ABF | 6 | Reserved |

## 4.6 Enhanced QEP Modules (eQEP1/2 /3)

The device contains up to three enhanced quadrature encoder (eQEP) modules with 32-bit resolution. Figure 4-7 shows the block diagram of the eQEP module.

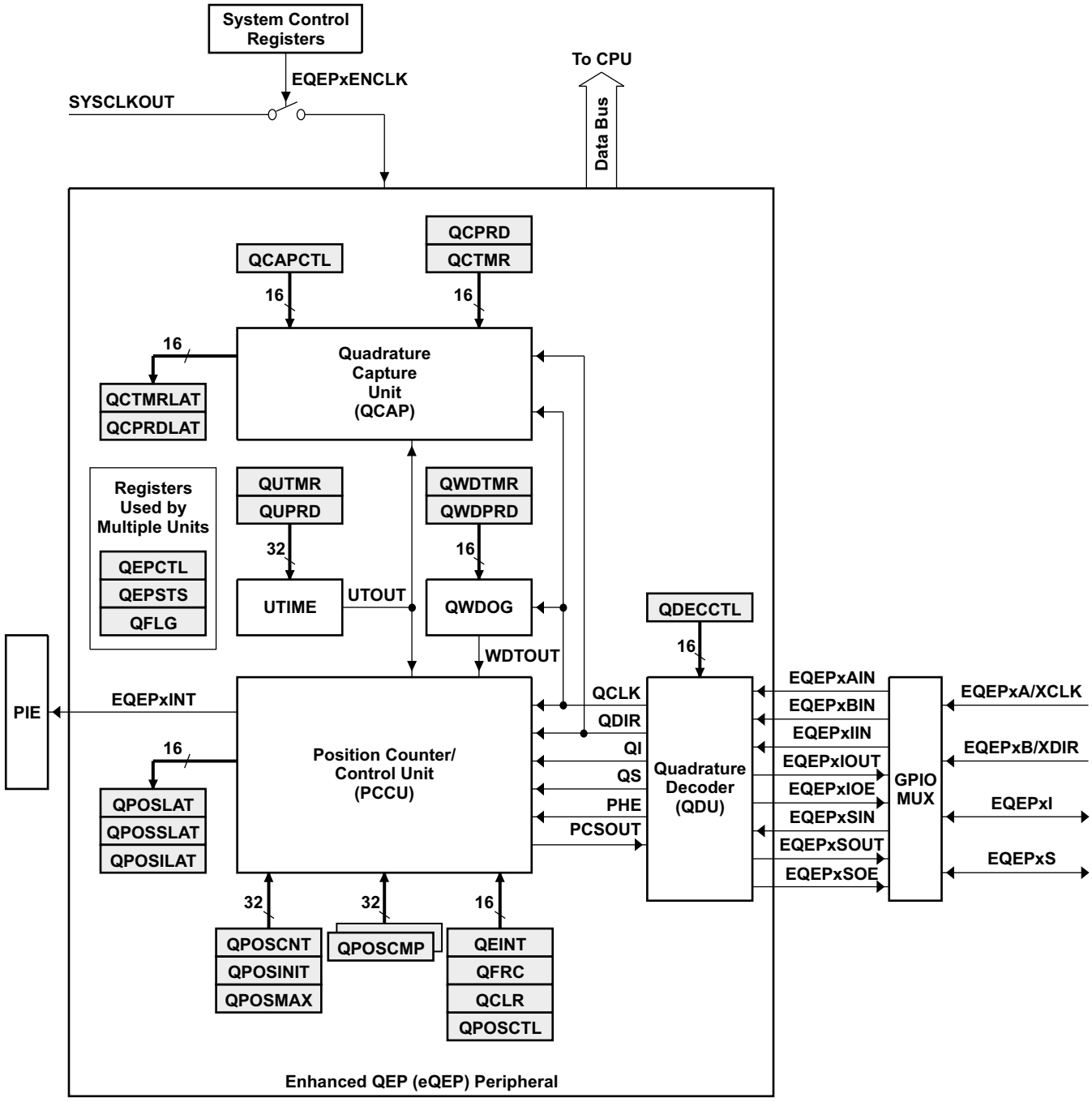



**Figure 4-7. eQEP Functional Block Diagram**

Table 4-5 provides a summary of the eQEP registers.

**Table 4-5. eQEP Control and Status Registers**

| NAME | eQEP1 ADDRESS | eQEP2 ADDRESS | eQEP3 ADDRESS | eQEPx SIZE(x16)/ #SHADOW | REGISTER DESCRIPTION |
|---|---|---|---|---|---|
| QPOSCNT | 0x6B00 | 0x6B40 | 0x6B80 | 2/0 | eQEP Position Counter |
| QPOSINIT | 0x6B02 | 0x6B42 | 0x6B82 | 2/0 | eQEP Initialization Position Count |
| QPOSMAX | 0x6B04 | 0x6B44 | 0x6B84 | 2/0 | eQEP Maximum Position Count |
| QPOSCMP | 0x6B06 | 0x6B46 | 0x6B86 | 2/1 | eQEP Position-compare |
| QPOSILAT | 0x6B08 | 0x6B48 | 0x6B88 | 2/0 | eQEP Index Position Latch |
| QPOSSLAT | 0x6B0A | 0x6B4A | 0x6B8A | 2/0 | eQEP Strobe Position Latch |
| QPOSLAT | 0x6B0C | 0x6B4C | 0x6B8C | 2/0 | eQEP Position Latch |
| QUTMR | 0x6B0E | 0x6B4E | 0x6B8E | 2/0 | eQEP Unit Timer |
| QUPRD | 0x6B10 | 0x6B50 | 0x6B90 | 2/0 | eQEP Unit Period Register |
| QWDTMR | 0x6B12 | 0x6B52 | 0x6B92 | 1/0 | eQEP Watchdog Timer |
| QWDPRD | 0x6B13 | 0x6B53 | 0x6B93 | 1/0 | eQEP Watchdog Period Register |
| QDECCTL | 0x6B14 | 0x6B54 | 0x6B94 | 1/0 | eQEP Decoder Control Register |
| QEPCTL | 0x6B15 | 0x6B55 | 0x6B95 | 1/0 | eQEP Control Register |
| QCAPCTL | 0x6B16 | 0x6B56 | 0x6B96 | 1/0 | eQEP Capture Control Register |
| QPOSCTL | 0x6B17 | 0x6B57 | 0x6B97 | 1/0 | eQEP Position-compare Control Register |
| QEINT | 0x6B18 | 0x6B58 | 0x6B98 | 1/0 | eQEP Interrupt Enable Register |
| QFLG | 0x6B19 | 0x6B59 | 0x6B99 | 1/0 | eQEP Interrupt Flag Register |
| QCLR | 0x6B1A | 0x6B5A | 0x6B9A | 1/0 | eQEP Interrupt Clear Register |
| QFRC | 0x6B1B | 0x6B5B | 0x6B9B | 1/0 | eQEP Interrupt Force Register |
| QEPSTS | 0x6B1C | 0x6B5C | 0x6B9C | 1/0 | eQEP Status Register |
| QCTMR | 0x6B1D | 0x6B5D | 0x6B9D | 1/0 | eQEP Capture Timer |
| QCPRD | 0x6B1E | 0x6B5E | 0x6B9E | 1/0 | eQEP Capture Period Register |
| QCTMRLAT | 0x6B1F | 0x6B5F | 0x6B9F | 1/0 | eQEP Capture Timer Latch |
| QCPRDLAT | 0x6B20 | 0x6B60 | 0x6BA0 | 1/0 | eQEP Capture Period Latch |
| Reserved | 0x6B21 - 0x6B3F | 0x6B61 - 0x6B7F | 0x6BBA1 - 0x6BBF | 31/0 | |

## 4.7 External ADC Interface

The external ADC interface operation is configured, controlled, and monitored by the External SoC Configuration Register (EXTSOCCFG) at address 0x702E.

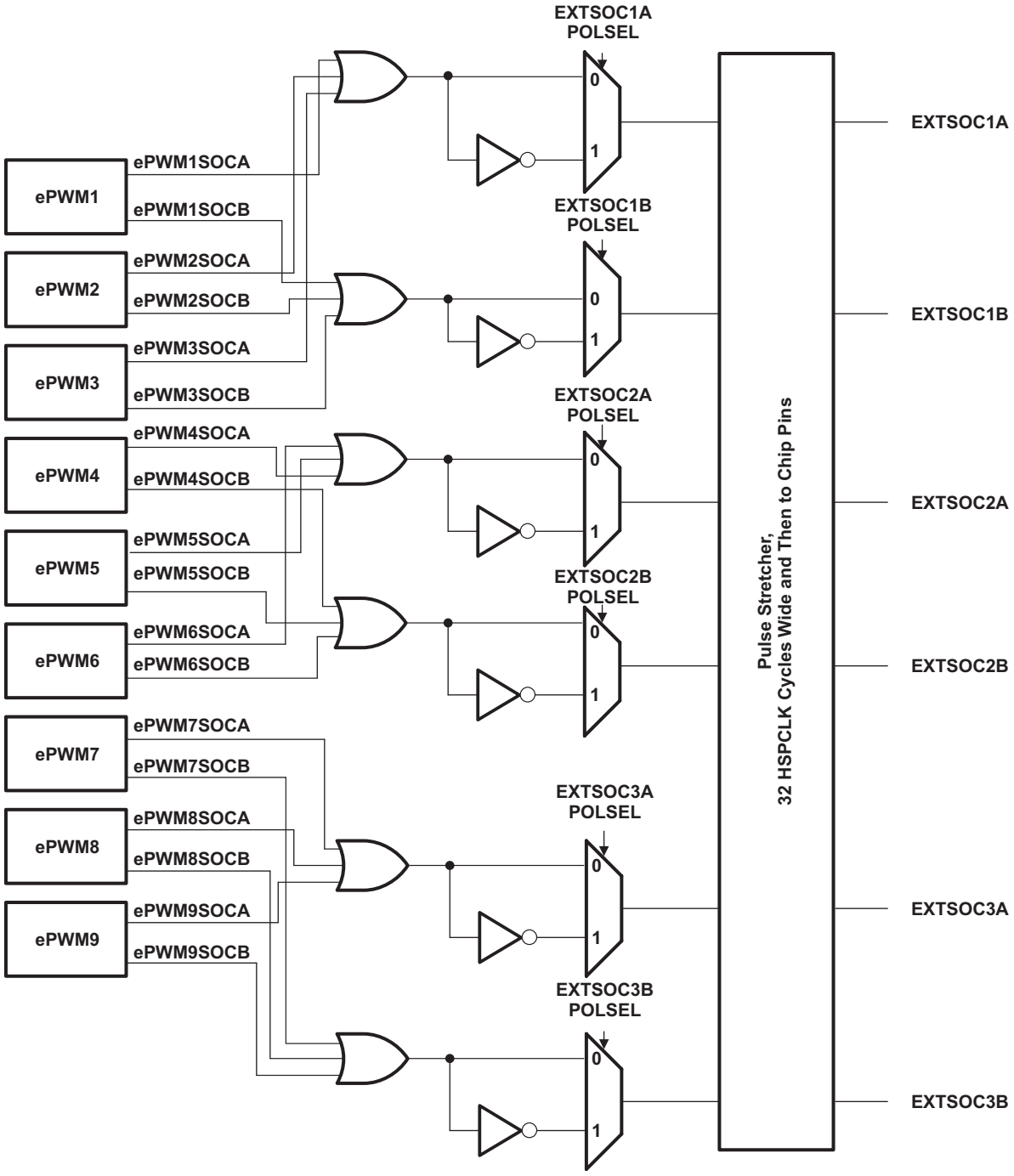



**Figure 4-8. External ADC Interface**

**Table 4-6. External ADC Interface Registers**

| NAME | DESCRIPTION | ADDRESS |
|---|---|---|
| EXTSOCCFG | External SoC Configuration Register | 0x00 702E |

## 4.8 Multichannel Buffered Serial Port (McBSP) Module

The McBSP module has the following features:

- Compatible to McBSP in TMS320C54x™/ TMS320C55x™ DSP devices
- Full-duplex communication
- Double-buffered data registers that allow a continuous data stream
- Independent framing and clocking for receive and transmit
- External shift clock generation or an internal programmable frequency shift clock
- A wide selection of data sizes including 8-, 12-, 16-, 20-, 24-, or 32-bits
- 8-bit data transfers with LSB or MSB first
- Programmable polarity for both frame synchronization and data clocks
- Highly programmable internal clock and frame generation
- Direct interface to industry-standard CODECs, Analog Interface Chips (AICs), and other serially connected A/D and D/A devices
- Works with SPI-compatible devices
- The following application interfaces can be supported on the McBSP:
  - T1/E1 framers
  - IOM-2 compliant devices
  - AC97-compliant devices (the necessary multiphase frame synchronization capability is provided.)
  - IIS-compliant devices
  - SPI
- McBSP clock rate,

$$CLKG = \frac{CLKSRG}{(1 + CLKGDV)}$$

where CLKSRG source could be LSPCLK, CLKX, or CLKR. Serial port performance is limited by I/O buffer switching speed. Internal prescalers must be adjusted such that the peripheral speed is less than the I/O buffer speed limit.

**NOTE**

See Section 6 for maximum I/O pin toggling speed.

Figure 4-9 shows the block diagram of the McBSP module.

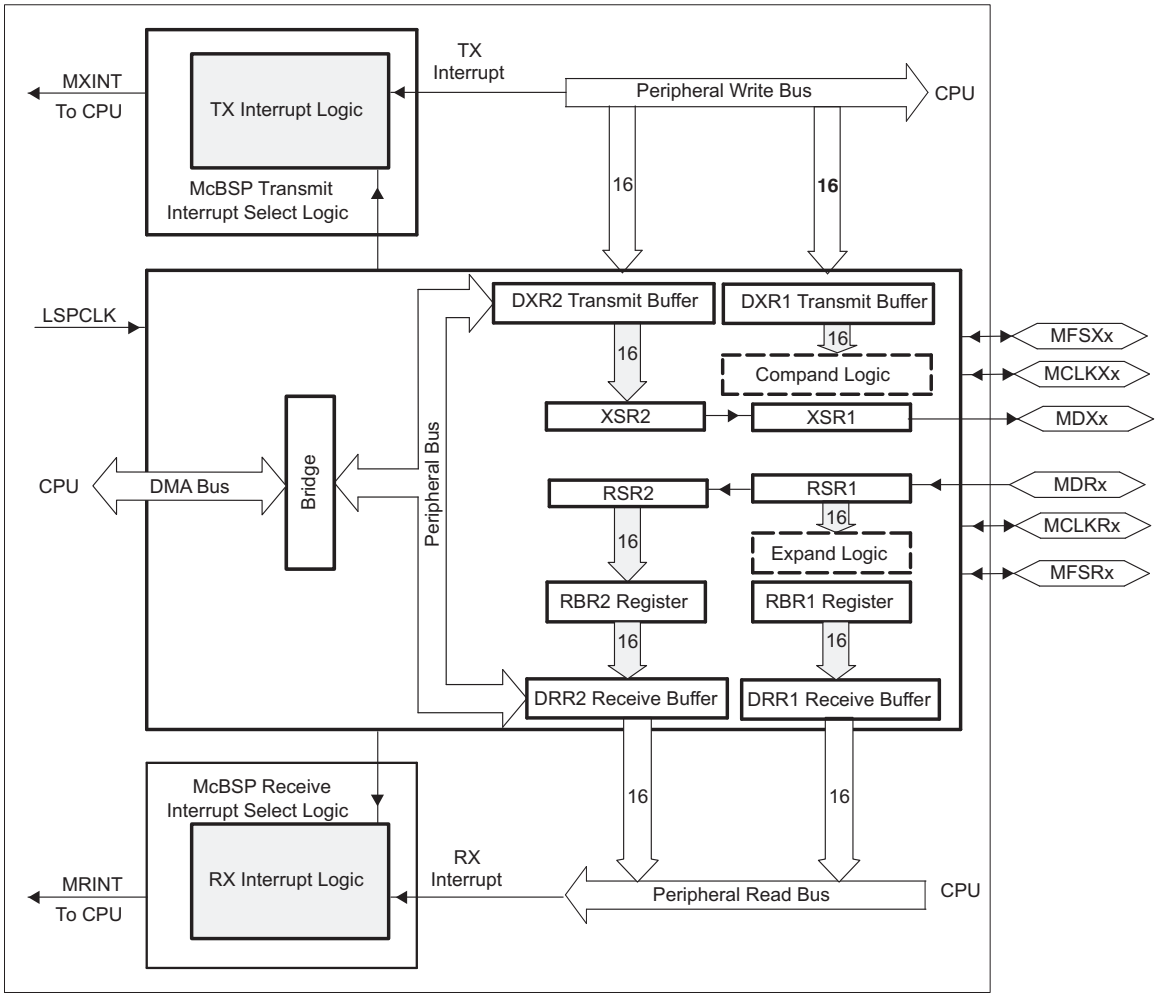



**Figure 4-9. McBSP Module**

Table 4-7 provides a summary of the McBSP registers.

**Table 4-7. McBSP Register Summary**

| NAME | McBSP-A ADDRESS | McBSP-B ADDRESS | TYPE | RESET VALUE | DESCRIPTION |
|---|---|---|---|---|---|
| **Data Registers, Receive, Transmit** | | | | | |
| DRR2 | 0x5000 | 0x5040 | R | 0x0000 | McBSP Data Receive Register 2 |
| DRR1 | 0x5001 | 0x5041 | R | 0x0000 | McBSP Data Receive Register 1 |
| DXR2 | 0x5002 | 0x5042 | W | 0x0000 | McBSP Data Transmit Register 2 |
| DXR1 | 0x5003 | 0x5043 | W | 0x0000 | McBSP Data Transmit Register 1 |
| **McBSP Control Registers** | | | | | |
| SPCR2 | 0x5004 | 0x5044 | R/W | 0x0000 | McBSP Serial Port Control Register 2 |
| SPCR1 | 0x5005 | 0x5045 | R/W | 0x0000 | McBSP Serial Port Control Register 1 |
| RCR2 | 0x5006 | 0x5046 | R/W | 0x0000 | McBSP Receive Control Register 2 |
| RCR1 | 0x5007 | 0x5047 | R/W | 0x0000 | McBSP Receive Control Register 1 |
| XCR2 | 0x5008 | 0x5048 | R/W | 0x0000 | McBSP Transmit Control Register 2 |
| XCR1 | 0x5009 | 0x5049 | R/W | 0x0000 | McBSP Transmit Control Register 1 |
| SRGR2 | 0x500A | 0x504A | R/W | 0x0000 | McBSP Sample Rate Generator Register 2 |
| SRGR1 | 0x500B | 0x504B | R/W | 0x0000 | McBSP Sample Rate Generator Register 1 |
| **Multichannel Control Registers** | | | | | |
| MCR2 | 0x500C | 0x504C | R/W | 0x0000 | McBSP Multichannel Register 2 |
| MCR1 | 0x500D | 0x504D | R/W | 0x0000 | McBSP Multichannel Register 1 |
| RCERA | 0x500E | 0x504E | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition A |
| RCERB | 0x500F | 0x504F | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition B |
| XCERA | 0x5010 | 0x5050 | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition A |
| XCERB | 0x5011 | 0x5051 | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition B |
| PCR | 0x5012 | 0x5052 | R/W | 0x0000 | McBSP Pin Control Register |
| RCERC | 0x5013 | 0x5053 | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition C |
| RCERD | 0x5014 | 0x5054 | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition D |
| XCERC | 0x5015 | 0x5055 | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition C |
| XCERD | 0x5016 | 0x5056 | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition D |
| RCERE | 0x5017 | 0x5057 | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition E |
| RCERF | 0x5018 | 0x5058 | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition F |
| XCERE | 0x5019 | 0x5059 | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition E |
| XCERF | 0x501A | 0x505A | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition F |
| RCERG | 0x501B | 0x505B | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition G |
| RCERH | 0x501C | 0x505C | R/W | 0x0000 | McBSP Receive Channel Enable Register Partition H |
| XCERG | 0x501D | 0x505D | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition G |
| XCERH | 0x501E | 0x505E | R/W | 0x0000 | McBSP Transmit Channel Enable Register Partition H |
| MFFINT | 0x5023 | 0x5063 | R/W | 0x0000 | McBSP Interrupt Enable Register |
| MFFST | 0x5024 | 0x5064 | R/W | 0x0000 | McBSP Pin Status Register |

## 4.9 Enhanced Controller Area Network (eCAN) Modules (eCAN-A and eCAN-B)

The CAN module has the following features:

- Fully compliant with CAN protocol, version 2.0B
- Supports data rates up to 1 Mbps
- Thirty-two mailboxes, each with the following properties:
  - Configurable as receive or transmit
  - Configurable with standard or extended identifier
  - Has a programmable receive mask
  - Supports data and remote frame
  - Composed of 0 to 8 bytes of data
  - Uses a 32-bit time stamp on receive and transmit message
  - Protects against reception of new message
  - Holds the dynamically programmable priority of transmit message
  - Employs a programmable interrupt scheme with two interrupt levels
  - Employs a programmable alarm on transmission or reception time-out
- Low-power mode
- Programmable wake-up on bus activity
- Automatic reply to a remote request message
- Automatic retransmission of a frame in case of loss of arbitration or error
- 32-bit local network time counter synchronized by a specific message (communication in conjunction with mailbox 16)
- Self-test mode
  - Operates in a loopback mode receiving its own message. A "dummy" acknowledge is provided, thereby eliminating the need for another node to provide the acknowledge bit.

**NOTE**

For a SYSCLKOUT of 300 MHz, the smallest bit rate possible is 11.719 kbps.

For a SYSCLKOUT of 200 MHz, the smallest bit rate possible is 7.8125 kbps.

The CAN has passed the conformance test per ISO/DIS 16845. Contact TI for test report and exceptions.

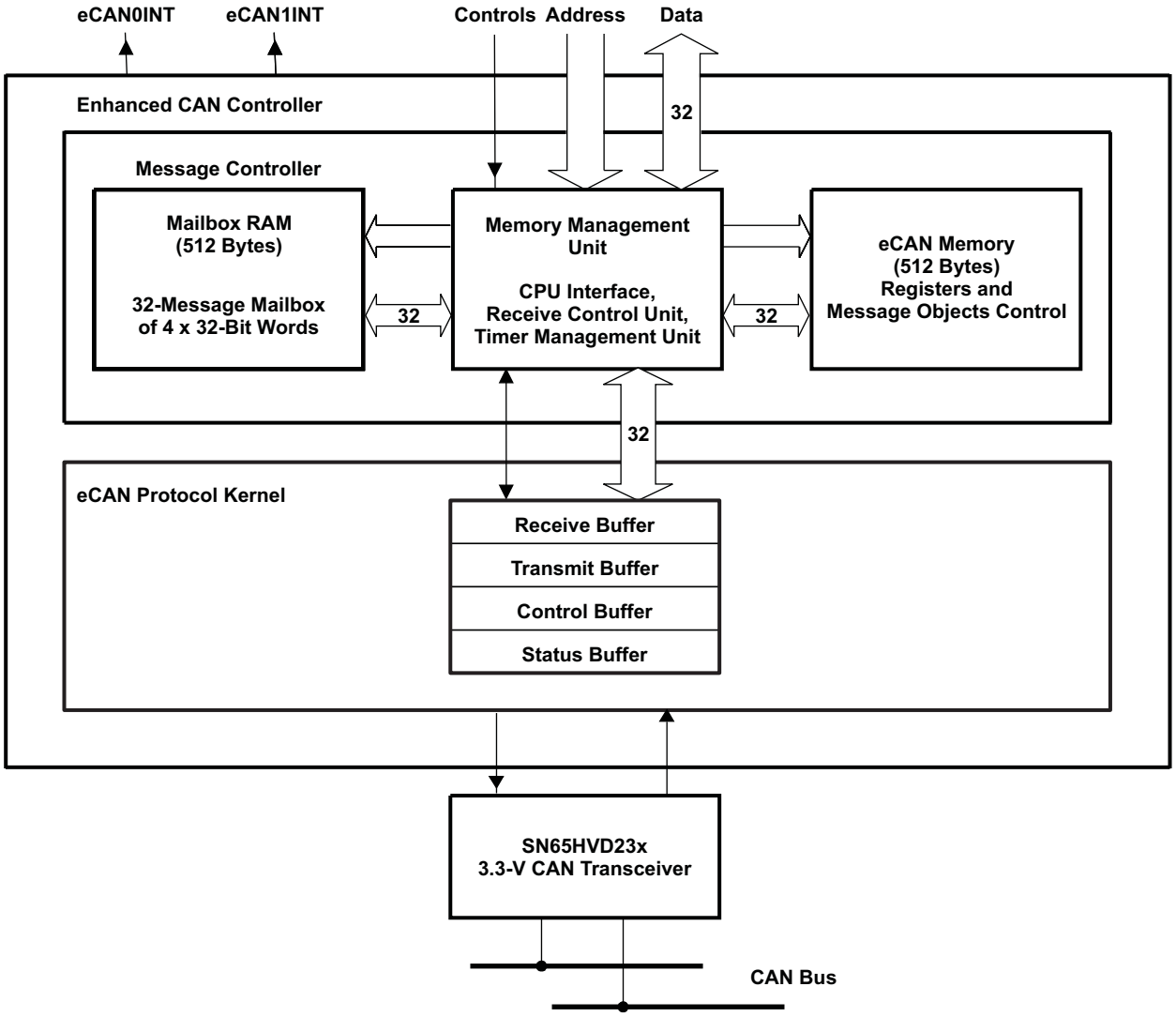



**Figure 4-10. eCAN Block Diagram and Interface Circuit**

**Table 4-8. 3.3-V eCAN Transceivers**

| PART NUMBER | SUPPLY VOLTAGE | LOW-POWER MODE | SLOPE CONTROL | VREF | OTHER | $T_A$ |
|---|---|---|---|---|---|---|
| SN65HVD230Q | 3.3 V | Standby | Adjustable | Yes | – | –40°C to 125°C |
| SN65HVD231Q | 3.3 V | Sleep | Adjustable | Yes | – | –40°C to 125°C |
| SN65HVD232Q | 3.3 V | None | None | None | – | –40°C to 125°C |
| SN65HVD233 | 3.3 V | Standby | Adjustable | None | Diagnostic Loopback | –40°C to 125°C |
| SN65HVD234 | 3.3 V | Standby and Sleep | Adjustable | None | – | –40°C to 125°C |
| SN65HVD235 | 3.3 V | Standby | Adjustable | None | Autobaud Loopback | –40°C to 125°C |
| ISO1050 | 3–5.5 V | None | None | None | • Built-in isolation<br>• Low-prop delay<br>• Thermal shutdown<br>• Failsafe operation<br>• Dominant time-out | –55°C to 105°C |

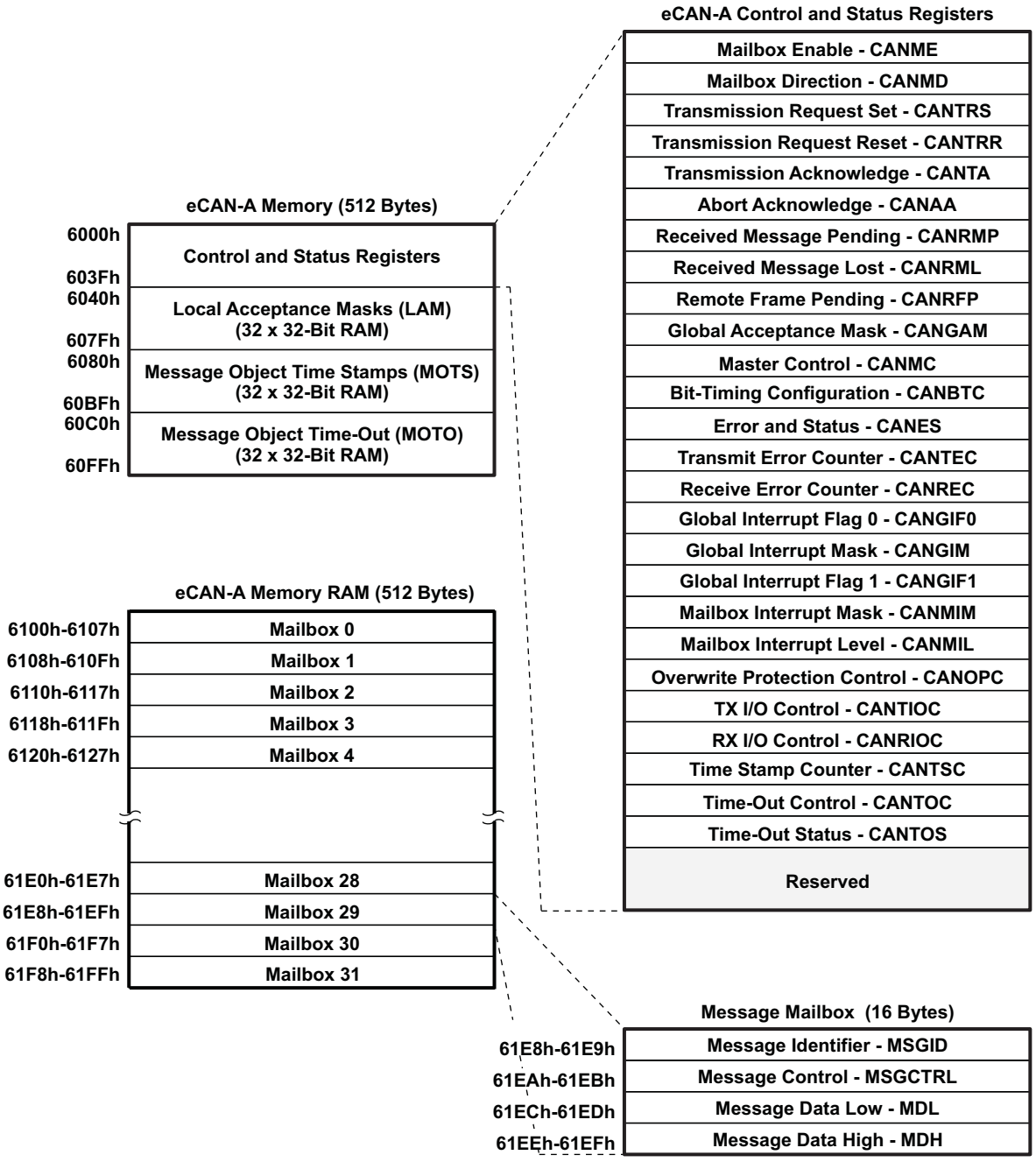



**Figure 4-11. eCAN-A Memory Map**

**NOTE**

If the eCAN module is not used in an application, the RAM available (LAM, MOTS, MOTO, and mailbox RAM) can be used as general-purpose RAM. The CAN module clock should be enabled for this.

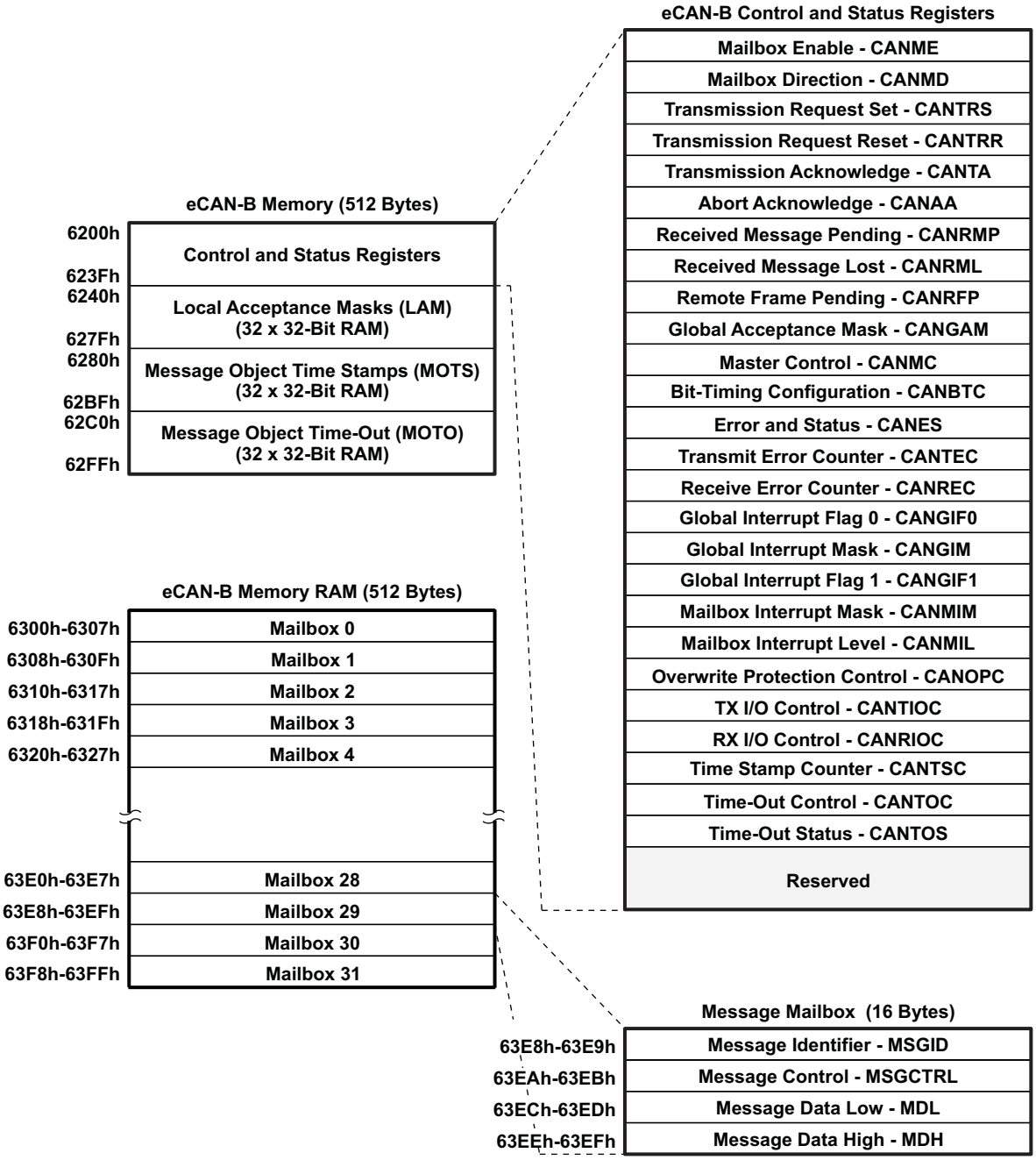



**Figure 4-12. eCAN-B Memory Map**

The CAN registers listed in Table 4-9 are used by the CPU to configure and control the CAN controller and the message objects. eCAN control registers only support 32-bit read/write operations. Mailbox RAM can be accessed as 16 bits or 32 bits. 32-bit accesses are aligned to an even boundary.

**Table 4-9. CAN Register Map(1)**

| REGISTER NAME | eCAN-A ADDRESS | eCAN-B ADDRESS | SIZE (x32) | DESCRIPTION |
|---|---|---|---|---|
| CANME | 0x6000 | 0x6200 | 1 | Mailbox enable |
| CANMD | 0x6002 | 0x6202 | 1 | Mailbox direction |
| CANTRS | 0x6004 | 0x6204 | 1 | Transmit request set |
| CANTRR | 0x6006 | 0x6206 | 1 | Transmit request reset |
| CANTA | 0x6008 | 0x6208 | 1 | Transmission acknowledge |
| CANAA | 0x600A | 0x620A | 1 | Abort acknowledge |
| CANRMP | 0x600C | 0x620C | 1 | Receive message pending |
| CANRML | 0x600E | 0x620E | 1 | Receive message lost |
| CANRFP | 0x6010 | 0x6210 | 1 | Remote frame pending |
| CANGAM | 0x6012 | 0x6212 | 1 | Global acceptance mask |
| CANMC | 0x6014 | 0x6214 | 1 | Master control |
| CANBTC | 0x6016 | 0x6216 | 1 | Bit-timing configuration |
| CANES | 0x6018 | 0x6218 | 1 | Error and status |
| CANTEC | 0x601A | 0x621A | 1 | Transmit error counter |
| CANREC | 0x601C | 0x621C | 1 | Receive error counter |
| CANGIF0 | 0x601E | 0x621E | 1 | Global interrupt flag 0 |
| CANGIM | 0x6020 | 0x6220 | 1 | Global interrupt mask |
| CANGIF1 | 0x6022 | 0x6222 | 1 | Global interrupt flag 1 |
| CANMIM | 0x6024 | 0x6224 | 1 | Mailbox interrupt mask |
| CANMIL | 0x6026 | 0x6226 | 1 | Mailbox interrupt level |
| CANOPC | 0x6028 | 0x6228 | 1 | Overwrite protection control |
| CANTIOC | 0x602A | 0x622A | 1 | TX I/O control |
| CANRIOC | 0x602C | 0x622C | 1 | RX I/O control |
| CANTSC | 0x602E | 0x622E | 1 | Time stamp counter (Reserved in SCC mode) |
| CANTOC | 0x6030 | 0x6230 | 1 | Time-out control (Reserved in SCC mode) |
| CANTOS | 0x6032 | 0x6232 | 1 | Time-out status (Reserved in SCC mode) |

(1) These registers are mapped to Peripheral Frame 1.

## 4.10 Serial Communications Interface (SCI) Modules (SCI-A, SCI-B, SCI-C)

The devices include three serial communications interface (SCI) modules. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format. The SCI receiver and transmitter are double-buffered, and each has its own separate enable and interrupt bits. Both can be operated independently or simultaneously in the full-duplex mode. To ensure data integrity, the SCI checks received data for break detection, parity, overrun, and framing errors. The bit rate is programmable to over 65000 different speeds through a 16-bit baud-select register.

Features of each SCI module include:

- Two external pins:
  - SCITXD: SCI transmit-output pin
  - SCIRXD: SCI receive-input pin

    NOTE: Both pins can be used as GPIO if not used for SCI.
  - Baud rate programmable to 64K different rates:

$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{BRR}+1)*8} \qquad \text{when BRR} \neq 0$$

$$\text{Baud rate} = \frac{\text{LSPCLK}}{16} \qquad \text{when BRR} = 0$$

**NOTE**

See Section 6 for maximum I/O pin toggling speed.

- Data-word format
  - One start bit
  - Data-word length programmable from one to eight bits
  - Optional even/odd/no parity bit
  - One or two stop bits
- Four error-detection flags: parity, overrun, framing, and break detection
- Two wake-up multiprocessor modes: idle-line and address bit
- Half- or full-duplex operation
- Double-buffered receive and transmit functions
- Transmitter and receiver operations can be accomplished through interrupt-driven or polled algorithms with status flags.
  - Transmitter: TXRDY flag (transmitter-buffer register is ready to receive another character) and TX EMPTY flag (transmitter-shift register is empty)
  - Receiver: RXRDY flag (receiver-buffer register is ready to receive another character), BRKDT flag (break condition occurred), and RX ERROR flag (monitoring four interrupt conditions)
- Separate enable bits for transmitter and receiver interrupts (except BRKDT)
- NRZ (non-return-to-zero) format

**NOTE**

All registers in this module are 8-bit registers that are connected to Peripheral Frame 2. When a register is accessed, the register data is in the lower byte (7-0), and the upper byte (15-8) is read as zeros. Writing to the upper byte has no effect.

Enhanced features:

- Auto baud-detect hardware logic
- 16-level transmit/receive FIFO

The SCI port operation is configured and controlled by the registers listed in Table 4-10, Table 4-11, and Table 4-12.

**Table 4-10. SCI-A Registers(1)**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| SCICCRA | 0x7050 | 1 | SCI-A Communications Control Register |
| SCICTL1A | 0x7051 | 1 | SCI-A Control Register 1 |
| SCIHBAUDA | 0x7052 | 1 | SCI-A Baud Register, High Bits |
| SCILBAUDA | 0x7053 | 1 | SCI-A Baud Register, Low Bits |
| SCICTL2A | 0x7054 | 1 | SCI-A Control Register 2 |
| SCIRXSTA | 0x7055 | 1 | SCI-A Receive Status Register |
| SCIRXEMUA | 0x7056 | 1 | SCI-A Receive Emulation Data Buffer Register |
| SCIRXBUFA | 0x7057 | 1 | SCI-A Receive Data Buffer Register |
| SCITXBUFA | 0x7059 | 1 | SCI-A Transmit Data Buffer Register |
| SCIFFTXA(2) | 0x705A | 1 | SCI-A FIFO Transmit Register |
| SCIFFRXA(2) | 0x705B | 1 | SCI-A FIFO Receive Register |
| SCIFFCTA(2) | 0x705C | 1 | SCI-A FIFO Control Register |
| SCIPRIA | 0x705F | 1 | SCI-A Priority Control Register |

(1) Registers in this table are mapped to Peripheral Frame 2 space. This space only allows 16-bit accesses. 32-bit accesses produce undefined results.
(2) These registers are new registers for the FIFO mode.

**Table 4-11. SCI-B Registers(1) (2)**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| SCICCRB | 0x7750 | 1 | SCI-B Communications Control Register |
| SCICTL1B | 0x7751 | 1 | SCI-B Control Register 1 |
| SCIHBAUDB | 0x7752 | 1 | SCI-B Baud Register, High Bits |
| SCILBAUDB | 0x7753 | 1 | SCI-B Baud Register, Low Bits |
| SCICTL2B | 0x7754 | 1 | SCI-B Control Register 2 |
| SCIRXSTB | 0x7755 | 1 | SCI-B Receive Status Register |
| SCIRXEMUB | 0x7756 | 1 | SCI-B Receive Emulation Data Buffer Register |
| SCIRXBUFB | 0x7757 | 1 | SCI-B Receive Data Buffer Register |
| SCITXBUFB | 0x7759 | 1 | SCI-B Transmit Data Buffer Register |
| SCIFFTXB(2) | 0x775A | 1 | SCI-B FIFO Transmit Register |
| SCIFFRXB(2) | 0x775B | 1 | SCI-B FIFO Receive Register |
| SCIFFCTB(2) | 0x775C | 1 | SCI-B FIFO Control Register |
| SCIPRIB | 0x775F | 1 | SCI-B Priority Control Register |

(1) Registers in this table are mapped to Peripheral Frame 2 space. This space only allows 16-bit accesses. 32-bit accesses produce undefined results.
(2) These registers are new registers for the FIFO mode.

**Table 4-12. SCI-C Registers(1) (2)**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| SCICCRC | 0x7770 | 1 | SCI-C Communications Control Register |
| SCICTL1C | 0x7771 | 1 | SCI-C Control Register 1 |
| SCIHBAUDC | 0x7772 | 1 | SCI-C Baud Register, High Bits |
| SCILBAUDC | 0x7773 | 1 | SCI-C Baud Register, Low Bits |
| SCICTL2C | 0x7774 | 1 | SCI-C Control Register 2 |
| SCIRXSTC | 0x7775 | 1 | SCI-C Receive Status Register |
| SCIRXEMUC | 0x7776 | 1 | SCI-C Receive Emulation Data Buffer Register |
| SCIRXBUFC | 0x7777 | 1 | SCI-C Receive Data Buffer Register |
| SCITXBUFC | 0x7779 | 1 | SCI-C Transmit Data Buffer Register |
| SCIFFTXC(2) | 0x777A | 1 | SCI-C FIFO Transmit Register |
| SCIFFRXC(2) | 0x777B | 1 | SCI-C FIFO Receive Register |
| SCIFFCTC(2) | 0x777C | 1 | SCI-C FIFO Control Register |
| SCIPRC | 0x777F | 1 | SCI-C Priority Control Register |

(1) Registers in this table are mapped to Peripheral Frame 2 space. This space only allows 16-bit accesses. 32-bit accesses produce undefined results.
(2) These registers are new registers for the FIFO mode.

Figure 4-13 shows the SCI module block diagram.

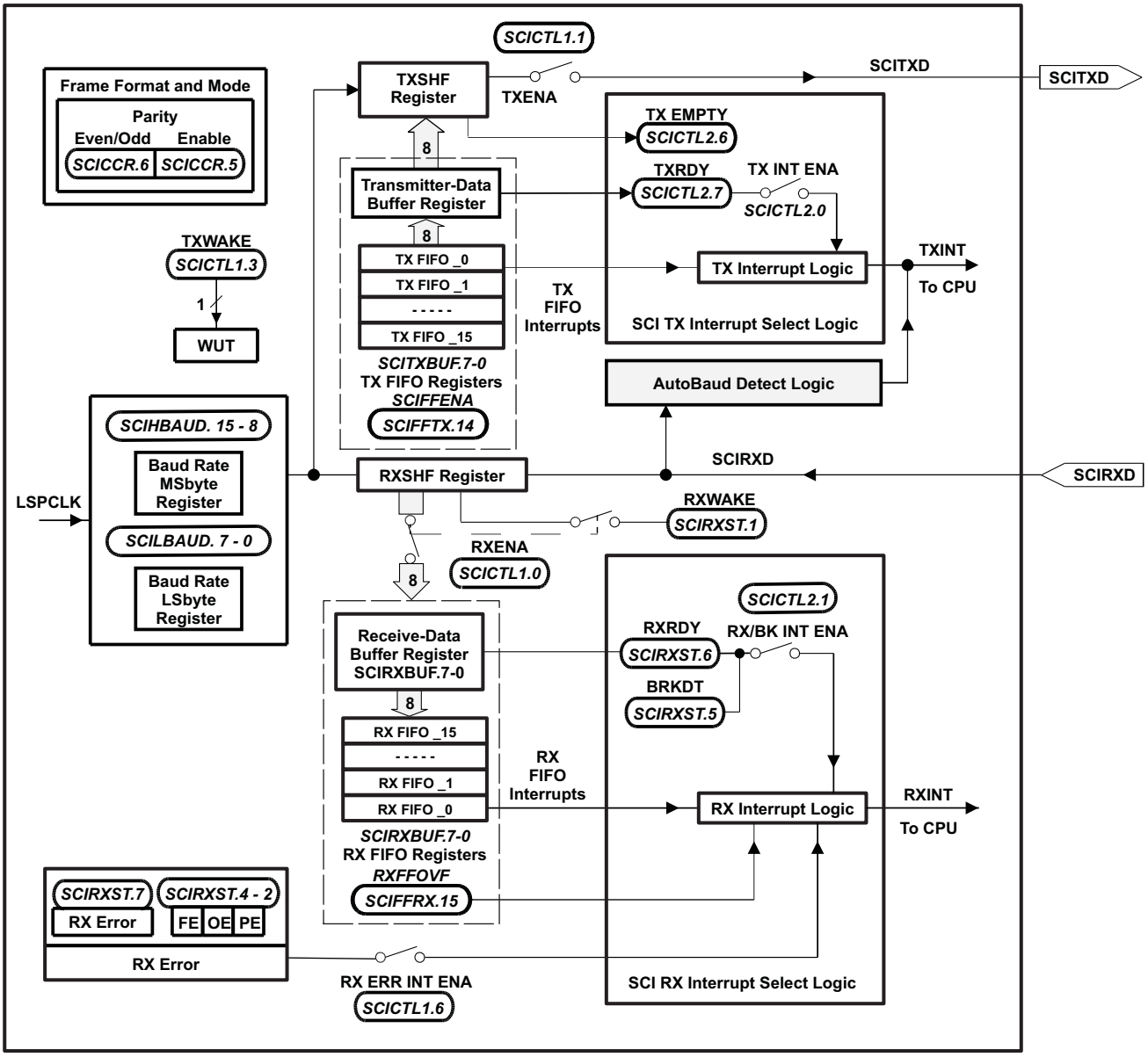



**Figure 4-13. Serial Communications Interface (SCI) Module Block Diagram**

## 4.11 Serial Peripheral Interface (SPI) Module (SPI-A , SPI-D)

The device includes the four-pin serial peripheral interface (SPI) module. Two SPI modules (SPI-A and SPI-D) are available. The SPI is a high-speed, synchronous serial I/O port that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmable bit-transfer rate. Normally, the SPI is used for communications between the MCU controller and external peripherals or another processor. Typical applications include external I/O or peripheral expansion through devices such as shift registers, display drivers, and ADCs. Multidevice communications are supported by the master/slave operation of the SPI.

The SPI module features include:

- Four external pins:
  - SPISOMI: SPI slave-output/master-input pin
  - SPISIMO: SPI slave-input/master-output pin
  - $\overline{\text{SPISTE}}$: SPI slave transmit-enable pin
  - SPICLK: SPI serial-clock pin

NOTE: All four pins can be used as GPIO if the SPI module is not used.

- Two operational modes: master and slave

  Baud rate: 125 different programmable rates.

$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{SPIBRR} + 1)} \qquad \text{when SPIBRR} = 3 \text{ to } 127$$

$$\text{Baud rate} = \frac{\text{LSPCLK}}{4} \qquad \text{when SPIBRR} = 0, 1, 2$$

**NOTE**

See Section 6 for maximum I/O pin toggling speed.

- Data word length: one to sixteen data bits
- Four clocking schemes (controlled by clock polarity and clock phase bits) include:
  - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
  - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive and transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt-driven or polled algorithms.
- Nine SPI module control registers: Located in control register frame beginning at address 7040h.

**NOTE**

All registers in this module are 16-bit registers that are connected to Peripheral Frame 2. When a register is accessed, the register data is in the lower byte (7–0), and the upper byte (15–8) is read as zeros. Writing to the upper byte has no effect.

Enhanced features:

- 16-level transmit/receive FIFO
- Delayed transmit control

The SPI port operation is configured and controlled by the registers listed in Table 4-13 and Table 4-14 .

**Table 4-13. SPI-A Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION(1) |
|---|---|---|---|
| SPICCR | 0x7040 | 1 | SPI-A Configuration Control Register |
| SPICTL | 0x7041 | 1 | SPI-A Operation Control Register |
| SPISTS | 0x7042 | 1 | SPI-A Status Register |
| SPIBRR | 0x7044 | 1 | SPI-A Baud Rate Register |
| SPIRXEMU | 0x7046 | 1 | SPI-A Receive Emulation Buffer Register |
| SPIRXBUF | 0x7047 | 1 | SPI-A Serial Input Buffer Register |
| SPITXBUF | 0x7048 | 1 | SPI-A Serial Output Buffer Register |
| SPIDAT | 0x7049 | 1 | SPI-A Serial Data Register |
| SPIFFTX | 0x704A | 1 | SPI-A FIFO Transmit Register |
| SPIFFRX | 0x704B | 1 | SPI-A FIFO Receive Register |
| SPIFFCT | 0x704C | 1 | SPI-A FIFO Control Register |
| SPIPRI | 0x704F | 1 | SPI-A Priority Control Register |

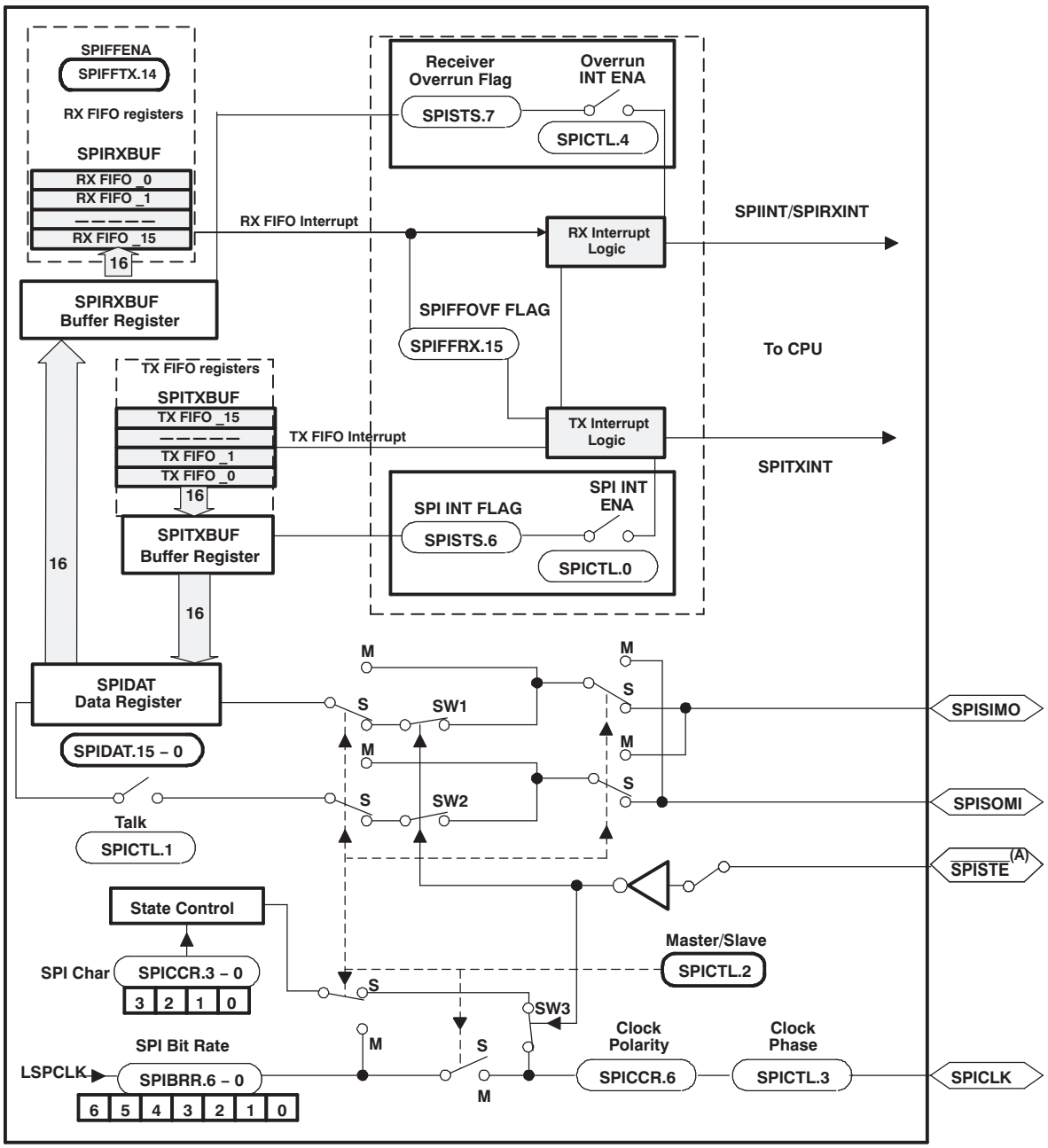
(1) Registers in this table are mapped to Peripheral Frame 2. This space only allows 16-bit accesses. 32-bit accesses produce undefined results.

**Table 4-14. SPI-D Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION(1) |
|---|---|---|---|
| SPICCR | 0x7780 | 1 | SPI-D Configuration Control Register |
| SPICTL | 0x7781 | 1 | SPI-D Operation Control Register |
| SPISTS | 0x7782 | 1 | SPI-D Status Register |
| SPIBRR | 0x7784 | 1 | SPI-D Baud Rate Register |
| SPIRXEMU | 0x7786 | 1 | SPI-D Receive Emulation Buffer Register |
| SPIRXBUF | 0x7787 | 1 | SPI-D Serial Input Buffer Register |
| SPITXBUF | 0x7788 | 1 | SPI-D Serial Output Buffer Register |
| SPIDAT | 0x7789 | 1 | SPI-D Serial Data Register |
| SPIFFTX | 0x778A | 1 | SPI-D FIFO Transmit Register |
| SPIFFRX | 0x778B | 1 | SPI-D FIFO Receive Register |
| SPIFFCT | 0x778C | 1 | SPI-D FIFO Control Register |
| SPIPRI | 0x778F | 1 | SPI-D Priority Control Register |

(1) Registers in this table are mapped to Peripheral Frame 2. This space only allows 16-bit accesses. 32-bit accesses produce undefined results.

Figure 4-14 is a block diagram of the SPI in slave mode.

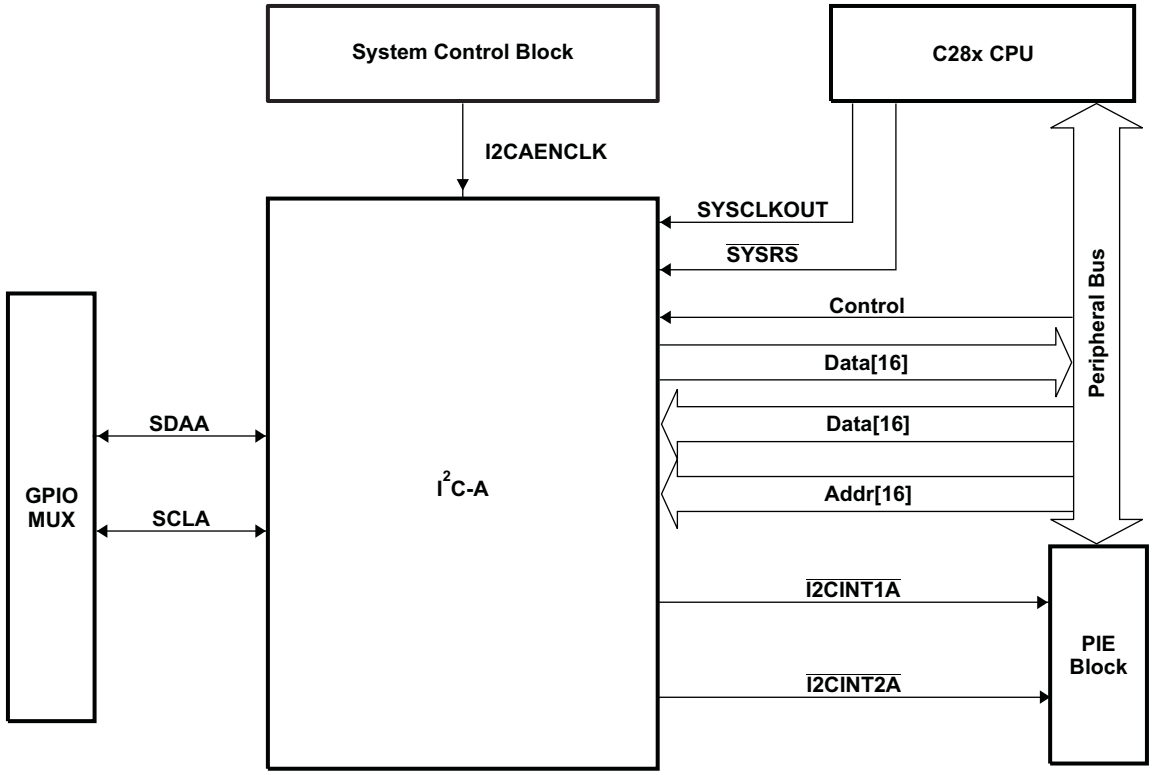



A. $\overline{\text{SPISTE}}$ is driven low by the master for a slave device.

**Figure 4-14. SPI Module Block Diagram (Slave Mode)**

## 4.12 Inter-Integrated Circuit (I2C)

The device contains one I2C Serial Port. Figure 4-15 shows how the I2C peripheral module interfaces within the device.




A. The I2C registers are accessed at the SYSCLKOUT rate. The internal timing and signal waveforms of the I2C port are also at the SYSCLKOUT rate.

B. The clock enable bit (I2CAENCLK) in the PCLKCR0 register turns off the clock to the I2C port for low power operation. Upon reset, I2CAENCLK is clear, which indicates the peripheral internal clocks are off.

**Figure 4-15. I2C Peripheral Module Interfaces**

The I2C module has the following features:

- Compliance with the Philips Semiconductors I2C-bus specification (version 2.1):
  - Support for 1-bit to 8-bit format transfers
  - 7-bit and 10-bit addressing modes
  - General call
  - START byte mode
  - Support for multiple master-transmitters and slave-receivers
  - Support for multiple slave-transmitters and master-receivers
  - Combined master transmit/receive and receive/transmit mode
  - Data transfer rate from 10 kbps up to 400 kbps (I2C Fast-mode rate)
- One 16-word receive FIFO and one 16-word transmit FIFO
- One interrupt that can be used by the CPU. This interrupt can be generated as a result of one of the following conditions:
  - Transmit-data ready
  - Receive-data ready
  - Register-access ready
  - No-acknowledgment received

  - Arbitration lost
  - Stop condition detected
  - Addressed as slave
- An additional interrupt that can be used by the CPU when in FIFO mode
- Module enable/disable capability
- Free data format mode

The registers in Table 4-15 configure and control the I2C port operation.

**Table 4-15. I2C-A Registers**

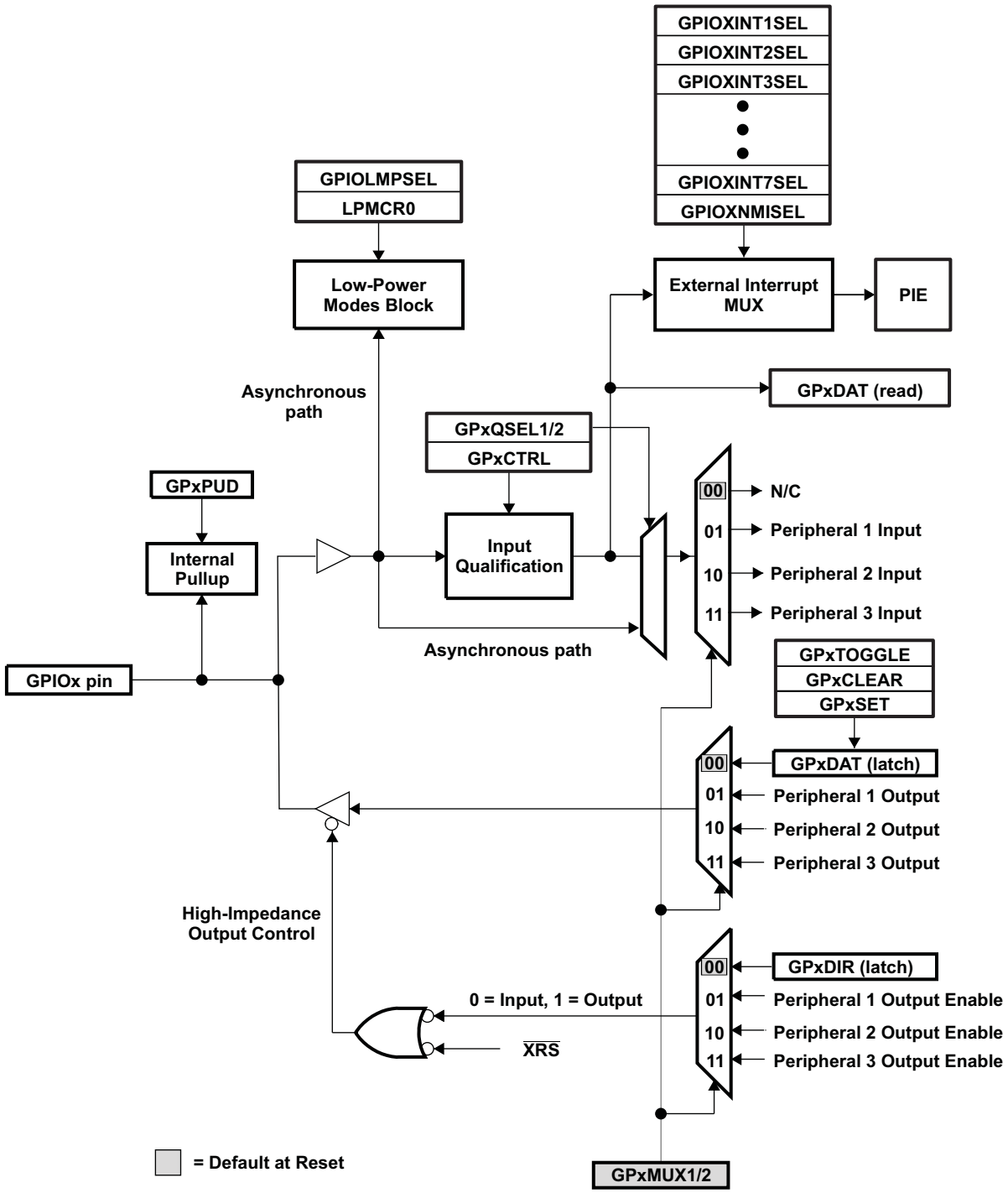
| NAME | ADDRESS | DESCRIPTION |
|---|---|---|
| I2COAR | 0x7900 | I2C own address register |
| I2CIER | 0x7901 | I2C interrupt enable register |
| I2CSTR | 0x7902 | I2C status register |
| I2CCLKL | 0x7903 | I2C clock low-time divider register |
| I2CCLKH | 0x7904 | I2C clock high-time divider register |
| I2CCNT | 0x7905 | I2C data count register |
| I2CDRR | 0x7906 | I2C data receive register |
| I2CSAR | 0x7907 | I2C slave address register |
| I2CDXR | 0x7908 | I2C data transmit register |
| I2CMDR | 0x7909 | I2C mode register |
| I2CISRC | 0x790A | I2C interrupt source register |
| I2CPSC | 0x790C | I2C prescaler register |
| I2CFFTX | 0x7920 | I2C FIFO transmit register |
| I2CFFRX | 0x7921 | I2C FIFO receive register |
| I2CRSR | – | I2C receive shift register (not accessible to the CPU) |
| I2CXSR | – | I2C transmit shift register (not accessible to the CPU) |

## 4.13 GPIO MUX

On the 2834x devices, the GPIO MUX can multiplex up to three independent peripheral signals on a single GPIO pin in addition to providing individual pin bit-banging I/O capability. The GPIO MUX block diagram per pin is shown in Figure 4-16. Because of the open drain capabilities of the I2C pins, the GPIO MUX block diagram for these pins differ. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1 ) for details.

**NOTE**

There is a 2-SYSCLKOUT cycle delay from when the write to the GPxMUXn and GPxQSELn registers occurs to when the action is valid.




A. x stands for the port, either A or B. For example, GPxDIR refers to either the GPADIR and GPBDIR register depending on the particular GPIO pin selected.

B. GPxDAT latch/read are accessed at the same memory location.

C. This is a generic GPIO MUX block diagram. Not all options may be applicable for all GPIO pins. See the *TMS320x2834x Delfino System Control and Interrupts Reference Guide* (literature number SPRUFN1 ) for pin-specific variations.

**Figure 4-16. GPIO MUX Block Diagram**

The device supports 88 GPIO pins. The GPIO control and data registers are mapped to Peripheral Frame 1 to enable 32-bit operations on the registers (along with 16-bit operations). Table 4-16 shows the GPIO register mapping.

**Table 4-16. GPIO Registers**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| **GPIO CONTROL REGISTERS (EALLOW PROTECTED)** | | | |
| GPACTRL | 0x6F80 | 2 | GPIO A Control Register (GPIO0 to 31) |
| GPAQSEL1 | 0x6F82 | 2 | GPIO A Qualifier Select 1 Register (GPIO0 to 15) |
| GPAQSEL2 | 0x6F84 | 2 | GPIO A Qualifier Select 2 Register (GPIO16 to 31) |
| GPAMUX1 | 0x6F86 | 2 | GPIO A MUX 1 Register (GPIO0 to 15) |
| GPAMUX2 | 0x6F88 | 2 | GPIO A MUX 2 Register (GPIO16 to 31) |
| GPADIR | 0x6F8A | 2 | GPIO A Direction Register (GPIO0 to 31) |
| GPAPUD | 0x6F8C | 2 | GPIO A Pull Up Disable Register (GPIO0 to 31) |
| Reserved | 0x6F8E – 0x6F8F | 2 | |
| GPBCTRL | 0x6F90 | 2 | GPIO B Control Register (GPIO32 to 63) |
| GPBQSEL1 | 0x6F92 | 2 | GPIO B Qualifier Select 1 Register (GPIO32 to 47) |
| GPBQSEL2 | 0x6F94 | 2 | GPIOB Qualifier Select 2 Register (GPIO48 to 63) |
| GPBMUX1 | 0x6F96 | 2 | GPIO B MUX 1 Register (GPIO32 to 47) |
| GPBMUX2 | 0x6F98 | 2 | GPIO B MUX 2 Register (GPIO48 to 63) |
| GPBDIR | 0x6F9A | 2 | GPIO B Direction Register (GPIO32 to 63) |
| GPBPUD | 0x6F9C | 2 | GPIO B Pull Up Disable Register (GPIO32 to 63) |
| Reserved | 0x6F9E – 0x6FA5 | 8 | |
| GPCMUX1 | 0x6FA6 | 2 | GPIO C MUX1 Register (GPIO64 to 79) |
| GPCMUX2 | 0x6FA8 | 2 | GPIO C MUX2 Register (GPIO80 to 87) |
| GPCDIR | 0x6FAA | 2 | GPIO C Direction Register (GPIO64 to 87) |
| GPCPUD | 0x6FAC | 2 | GPIO C Pull Up Disable Register (GPIO64 to 87) |
| Reserved | 0x6FAE – 0x6FBF | 18 | |
| **GPIO DATA REGISTERS (NOT EALLOW PROTECTED)** | | | |
| GPADAT | 0x6FC0 | 2 | GPIO A Data Register (GPIO0 to 31) |
| GPASET | 0x6FC2 | 2 | GPIO A Data Set Register (GPIO0 to 31) |
| GPACLEAR | 0x6FC4 | 2 | GPIO A Data Clear Register (GPIO0 to 31) |
| GPATOGGLE | 0x6FC6 | 2 | GPIO A Data Toggle Register (GPIO0 to 31) |
| GPBDAT | 0x6FC8 | 2 | GPIO B Data Register (GPIO32 to 63) |
| GPBSET | 0x6FCA | 2 | GPIO B Data Set Register (GPIO32 to 63) |
| GPBCLEAR | 0x6FCC | 2 | GPIO B Data Clear Register (GPIO32 to 63) |
| GPBTOGGLE | 0x6FCE | 2 | GPIOB Data Toggle Register (GPIO32 to 63) |
| GPCDAT | 0x6FD0 | 2 | GPIO C Data Register (GPIO64 to 87) |
| GPCSET | 0x6FD2 | 2 | GPIO C Data Set Register (GPIO64 to 87) |
| GPCCLEAR | 0x6FD4 | 2 | GPIO C Data Clear Register (GPIO64 to 87) |
| GPCTOGGLE | 0x6FD6 | 2 | GPIO C Data Toggle Register (GPIO64 to 87) |
| Reserved | 0x6FD8 – 0x6FDF | 8 | |
| **GPIO INTERRUPT AND LOW POWER MODES SELECT REGISTERS (EALLOW PROTECTED)** | | | |
| GPIOXINT1SEL | 0x6FE0 | 1 | XINT1 GPIO Input Select Register (GPIO0 to 31) |
| GPIOXINT2SEL | 0x6FE1 | 1 | XINT2 GPIO Input Select Register (GPIO0 to 31) |
| GPIOXNMISEL | 0x6FE2 | 1 | XNMI GPIO Input Select Register (GPIO0 to 31) |
| GPIOXINT3SEL | 0x6FE3 | 1 | XINT3 GPIO Input Select Register (GPIO32 to 63) |
| GPIOXINT4SEL | 0x6FE4 | 1 | XINT4 GPIO Input Select Register (GPIO32 to 63) |
| GPIOXINT5SEL | 0x6FE5 | 1 | XINT5 GPIO Input Select Register (GPIO32 to 63) |

**Table 4-16. GPIO Registers (continued)**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| GPIOXINT6SEL | 0x6FE6 | 1 | XINT6 GPIO Input Select Register (GPIO32 to 63) |
| GPIOINT7SEL | 0x6FE7 | 1 | XINT7 GPIO Input Select Register (GPIO32 to 63) |
| GPIOLPMSEL | 0x6FE8 | 2 | LPM GPIO Select Register (GPIO0 to 31) |
| Reserved | 0x6FEA – 0x6FFF | 22 | |

**Table 4-17. GPIO-A Mux Peripheral Selection Matrix**

| REGISTER BITS | | | PERIPHERAL SELECTION | | | |
|---|---|---|---|---|---|---|
| GPADIR GPADAT GPASET GPACLR GPATOGGLE | | GPAMUX1 GPAQSEL1 | GPIOx GPAMUX1 = 0,0 | PER1 GPAMUX1 = 0, 1 | PER2 GPAMUX1 = 1, 0 | PER3 GPAMUX1 = 1, 1 |
| QUALPRD0 | 0 | 1, 0 | GPIO0 (I/O) | EPWM1A (O) | Reserved | Reserved |
| | 1 | 3, 2 | GPIO1 (I/O) | EPWM1B (O) | ECAP6 (I/O) | MFSRB (I/O) |
| | 2 | 5, 4 | GPIO2 (I/O) | EPWM2A (O) | Reserved | Reserved |
| | 3 | 7, 6 | GPIO3 (I/O) | EPWM2B (O) | ECAP5 (I/O) | MCLKRB (I/O) |
| | 4 | 9, 8 | GPIO4 (I/O) | EPWM3A (O) | Reserved | Reserved |
| | 5 | 11, 10 | GPIO5 (I/O) | EPWM3B (O) | MFSRA (I/O) | ECAP1 (I/O) |
| | 6 | 13, 12 | GPIO6 (I/O) | EPWM4A (O) | EPWMSYNCI (I) | EPWMSYNCO (O) |
| | 7 | 15, 14 | GPIO7 (I/O) | EPWM4B (O) | MCLKRA (I/O) | ECAP2 (I/O) |
| QUALPRD1 | 8 | 17, 16 | GPIO8 (I/O) | EPWM5A (O) | CANTXB (O) | $\overline{\text{ADCSOCAO}}$ (O) |
| | 9 | 19, 18 | GPIO9 (I/O) | EPWM5B (O) | SCITXDB (O) | ECAP3 (I/O) |
| | 10 | 21, 20 | GPIO10 (I/O) | EPWM6A (O) | CANRXB (I) | $\overline{\text{ADCSOCBO}}$ (O) |
| | 11 | 23, 22 | GPIO11 (I/O) | EPWM6B (O) | SCIRXDB (I) | ECAP4 (I/O) |
| | 12 | 25, 24 | GPIO12 (I/O) | $\overline{\text{TZ1}}$ (I) | CANTXB (O) | MDXB (O) |
| | 13 | 27, 26 | GPIO13 (I/O) | $\overline{\text{TZ2}}$ (I) | CANRXB (I) | MDRB (I) |
| | 14 | 29, 28 | GPIO14 (I/O) | $\overline{\text{TZ3}}$ (I)/$\overline{\text{XHOLD}}$ (I) | SCITXDB (O) | MCLKXB (I/O) |
| | 15 | 31, 30 | GPIO15 (I/O) | $\overline{\text{TZ4}}$ (I)/$\overline{\text{XHOLDA}}$ (O) | SCIRXDB (I) | MFSXB (I/O) |
| | | GPAMUX2 GPAQSEL2 | GPAMUX2 = 0, 0 | GPAMUX2 = 0, 1 | GPAMUX2 = 1, 0 | GPAMUX2 = 1, 1 |
| QUALPRD2 | 16 | 1, 0 | GPIO16 (I/O) | SPISIMOA (I/O) | CANTXB (O) | $\overline{\text{TZ5}}$ (I) |
| | 17 | 3, 2 | GPIO17 (I/O) | SPISOMIA (I/O) | CANRXB (I) | $\overline{\text{TZ6}}$ (I) |
| | 18 | 5, 4 | GPIO18 (I/O) | SPICLKA (I/O) | SCITXDB (O) | CANRXA (I) |
| | 19 | 7, 6 | GPIO19 (I/O) | $\overline{\text{SPISTEA}}$ (I/O) | SCIRXDB (I) | CANTXA (O) |
| | 20 | 9, 8 | GPIO20 (I/O) | EQEP1A (I) | MDXA (O) | CANTXB (O) |
| | 21 | 11, 10 | GPIO21 (I/O) | EQEP1B (I) | MDRA (I) | CANRXB (I) |
| | 22 | 13, 12 | GPIO22 (I/O) | EQEP1S (I/O) | MCLKXA (I/O) | SCITXDB (O) |
| | 23 | 15, 14 | GPIO23 (I/O) | EQEP1I (I/O) | MFSXA (I/O) | SCIRXDB (I) |
| QUALPRD3 | 24 | 17, 16 | GPIO24 (I/O) | ECAP1 (I/O) | EQEP2A (I) | MDXB (O) |
| | 25 | 19, 18 | GPIO25 (I/O) | ECAP2 (I/O) | EQEP2B (I) | MDRB (I) |
| | 26 | 21, 20 | GPIO26 (I/O) | ECAP3 (I/O) | EQEP2I (I/O) | MCLKXB (I/O) |
| | 27 | 23, 22 | GPIO27 (I/O) | ECAP4 (I/O) | EQEP2S (I/O) | MFSXB (I/O) |
| | 28 | 25, 24 | GPIO28 (I/O) | SCIRXDA (I) | $\overline{\text{XZCS6}}$ (O) | |
| | 29 | 27, 26 | GPIO29 (I/O) | SCITXDA (O) | XA19 (O) | |
| | 30 | 29, 28 | GPIO30 (I/O) | CANRXA (I) | XA18 (O) | |
| | 31 | 31, 30 | GPIO31 (I/O) | CANTXA (O) | XA17 (O) | |

**Table 4-18. GPIO-B Mux Peripheral Selection Matrix**

| REGISTER BITS | | | PERIPHERAL SELECTION | | | |
|---|---|---|---|---|---|---|
| GPBDIR GPBDAT GPBSET GPBCLR GPBTOGGLE | | GPBMUX1 GPBQSEL1 | GPIOx GPBMUX1 = 0, 0 | PER1 GPBMUX1 = 0, 1 | PER2 GPBMUX1 = 1, 0 | PER3 GPBMUX1 = 1, 1 |
| QUALPRD0 | 0 | 1, 0 | GPIO32 (I/O) | SDAA (I/OC)(1) | EPWMSYNCI (I) | $\overline{\text{ADCSOCAO}}$ (O) |
| | 1 | 3, 2 | GPIO33 (I/O) | SCLA (I/OC)(1) | EPWMSYNCO (O) | $\overline{\text{ADCSOCBO}}$ (O) |
| | 2 | 5, 4 | GPIO34 (I/O) | ECAP1 (I/O) | XREADY (I) | |
| | 3 | 7, 6 | GPIO35 (I/O) | SCITXDA (O) | XR/$\overline{\text{W}}$ (O) | |
| | 4 | 9, 8 | GPIO36 (I/O) | SCIRXDA (I) | $\overline{\text{XZCS0}}$ (O) | |
| | 5 | 11, 10 | GPIO37 (I/O) | ECAP2 (I/O) | $\overline{\text{XZCS7}}$ (O) | |
| | 6 | 13, 12 | GPIO38 (I/O) | Reserved | $\overline{\text{XWE0}}$ (O) | |
| | 7 | 15, 14 | GPIO39 (I/O) | | XA16 (O) | |
| QUALPRD1 | 8 | 17, 16 | GPIO40 (I/O) | | XA0 (O) | |
| | 9 | 19, 18 | GPIO41 (I/O) | | XA1 (O) | |
| | 10 | 21, 20 | GPIO42 (I/O) | | XA2 (O) | |
| | 11 | 23, 22 | GPIO43 (I/O) | | XA3 (O) | |
| | 12 | 25, 24 | GPIO44 (I/O) | | XA4 (O) | |
| | 13 | 27, 26 | GPIO45 (I/O) | | XA5 (O) | |
| | 14 | 29, 28 | GPIO46 (I/O) | | XA6 (O) | |
| | 15 | 31, 30 | GPIO47 (I/O) | | XA7 (O) | |
| | | GPBMUX2 GPBQSEL2 | GPBMUX2 = 0, 0 | GPBMUX2 = 0, 1 | GPBMUX2 = 1, 0 | GPBMUX2 = 1, 1 |
| QUALPRD2 | 16 | 1, 0 | GPIO48 (I/O) | ECAP5 (I/O) | XD31 (I/O) | SPISIMOD (I/O) |
| | 17 | 3, 2 | GPIO49 (I/O) | ECAP6 (I/O) | XD30 (I/O) | SPISOMID (I/O) |
| | 18 | 5, 4 | GPIO50 (I/O) | EQEP1A (I) | XD29 (I/O) | SPICLKD (I/O) |
| | 19 | 7, 6 | GPIO51 (I/O) | EQEP1B (I) | XD28 (I/O) | SPISTED (I/O) |
| | 20 | 9, 8 | GPIO52 (I/O) | EQEP1S (I/O) | XD27 (I/O) | Reserved |
| | 21 | 11, 10 | GPIO53 (I/O) | EQEP1I (I/O) | XD26 (I/O) | Reserved |
| | 22 | 13, 12 | GPIO54 (I/O) | SPISIMOA (I/O) | XD25 (I/O) | EQEP3A (I) |
| | 23 | 15, 14 | GPIO55 (I/O) | SPISOMIA (I/O) | XD24 (I/O) | EQEP3B (I) |
| QUALPRD3 | 24 | 17, 16 | GPIO56 (I/O) | SPICLKA (I/O) | XD23 (I/O) | EQEP3S (I/O) |
| | 25 | 19, 18 | GPIO57 (I/O) | $\overline{\text{SPISTEA}}$ (I/O) | XD22 (I/O) | EQEP3I (I/O) |
| | 26 | 21, 20 | GPIO58 (I/O) | MCLKRA (I/O) | XD21 (I/O) | EPWM7A (O) |
| | 27 | 23, 22 | GPIO59 (I/O) | MFSRA (I/O) | XD20 (I/O) | EPWM7B (O) |
| | 28 | 25, 24 | GPIO60 (I/O) | MCLKRB (I/O) | XD19 (I/O) | EPWM8A (O) |
| | 29 | 27, 26 | GPIO61 (I/O) | MFSRB (I/O) | XD18 (I/O) | EPWM8B (O) |
| | 30 | 29, 28 | GPIO62 (I/O) | SCIRXDC (I) | XD17 (I/O) | EPWM9A (O) |
| | 31 | 31, 30 | GPIO63 (I/O) | SCITXDC (O) | XD16 (I/O) | EPWM9B (O) |

(1) Open drain

**Table 4-19. GPIO-C Mux Peripheral Selection Matrix**

| REGISTER BITS | | | PERIPHERAL SELECTION | |
|---|---|---|---|---|
| GPCDIR GPCDAT GPCSET GPCCLR GPCTOGGLE | | GPCMUX1 | GPIOx or PER1 GPCMUX1 = 0, 0 or 0, 1 | PER2 or PER3 GPCMUX1 = 1, 0 or 1, 1 |
| no qual | 0 | 1, 0 | GPIO64 (I/O) | XD15 (I/O) |
| | 1 | 3, 2 | GPIO65 (I/O) | XD14 (I/O) |
| | 2 | 5, 4 | GPIO66 (I/O) | XD13 (I/O) |
| | 3 | 7, 6 | GPIO67 (I/O) | XD12 (I/O) |
| | 4 | 9, 8 | GPIO68 (I/O) | XD11 (I/O) |
| | 5 | 11, 10 | GPIO69 (I/O) | XD10 (I/O) |
| | 6 | 13, 12 | GPIO70 (I/O) | XD9 (I/O) |
| | 7 | 15, 14 | GPIO71 (I/O) | XD8 (I/O) |
| no qual | 8 | 17, 16 | GPIO72 (I/O) | XD7 (I/O) |
| | 9 | 19, 18 | GPIO73 (I/O) | XD6 (I/O) |
| | 10 | 21, 20 | GPIO74 (I/O) | XD5 (I/O) |
| | 11 | 23, 22 | GPIO75 (I/O) | XD4 (I/O) |
| | 12 | 25, 24 | GPIO76 (I/O) | XD3 (I/O) |
| | 13 | 27, 26 | GPIO77 (I/O) | XD2 (I/O) |
| | 14 | 29, 28 | GPIO78 (I/O) | XD1 (I/O) |
| | 15 | 31, 30 | GPIO79 (I/O) | XD0 (I/O) |
| | | GPCMUX2 | GPCMUX2 = 0, 0 or 0, 1 | GPCMUX2 = 1, 0 or 1, 1 |
| no qual | 16 | 1, 0 | GPIO80 (I/O) | XA8 (O) |
| | 17 | 3, 2 | GPIO81 (I/O) | XA9 (O) |
| | 18 | 5, 4 | GPIO82 (I/O) | XA10 (O) |
| | 19 | 7, 6 | GPIO83 (I/O) | XA11 (O) |
| | 20 | 9, 8 | GPIO84 (I/O) | XA12 (O) |
| | 21 | 11, 10 | GPIO85 (I/O) | XA13 (O) |
| | 22 | 13, 12 | GPIO86 (I/O) | XA14 (O) |
| | 23 | 15, 14 | GPIO87 (I/O) | XA15 (O) |

The user can select the type of input qualification for each GPIO pin via the GPxQSEL1/2 registers from four choices:

- Synchronization To SYSCLKOUT Only (GPxQSEL1/2 = 0, 0): This is the default mode of all GPIO pins at reset and it simply synchronizes the input signal to the system clock (SYSCLKOUT).
- Qualification Using Sampling Window (GPxQSEL1/2 = 0, 1 and 1, 0): In this mode the input signal, after synchronization to the system clock (SYSCLKOUT), is qualified by a specified number of cycles before the input is allowed to change.

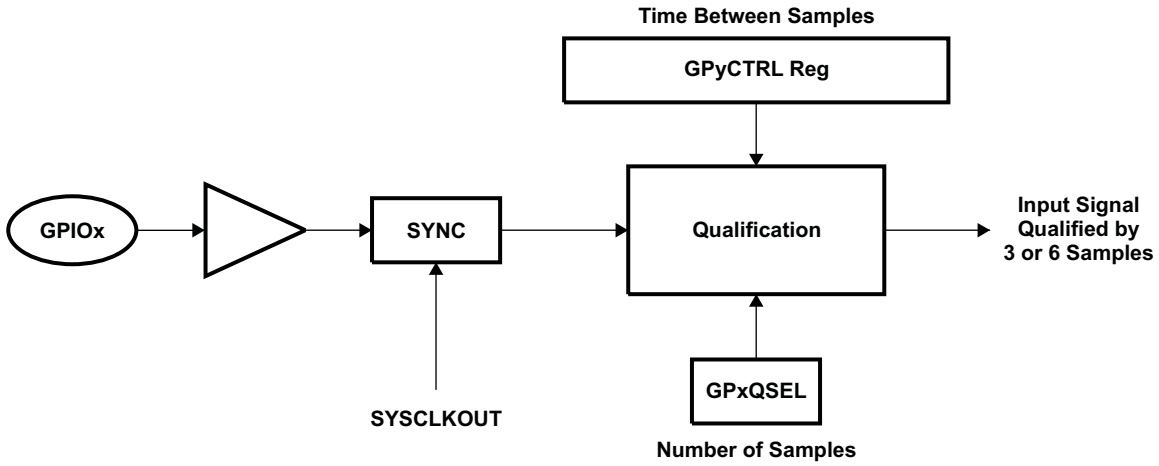



**Figure 4-17. Qualification Using Sampling Window**

- The sampling period is specified by the QUALPRD bits in the GPxCTRL register and is configurable in groups of 8 signals. It specifies a multiple of SYSCLKOUT cycles for sampling the input signal. The sampling window is either 3-samples or 6-samples wide and the output is only changed when ALL samples are the same (all 0s or all 1s) as shown in Figure 4-17 (for 6-sample mode).
- No Synchronization (GPxQSEL1/2 = 1,1): This mode is used for peripherals where synchronization is not required (synchronization is performed within the peripheral).

Due to the multi-level multiplexing that is required on the device, there may be cases where a peripheral input signal can be mapped to more then one GPIO pin. Also, when an input signal is not selected, the input signal will default to either a 0 or 1 state, depending on the peripheral.

## 4.14 External Interface (XINTF)

This section gives a top-level view of the external interface (XINTF) that is implemented on the C2834x devices.

The XINTF is a non-multiplexed asynchronous bus, similar to the 2812 XINTF. The XINTF is mapped into three fixed zones shown in Figure 4-18.

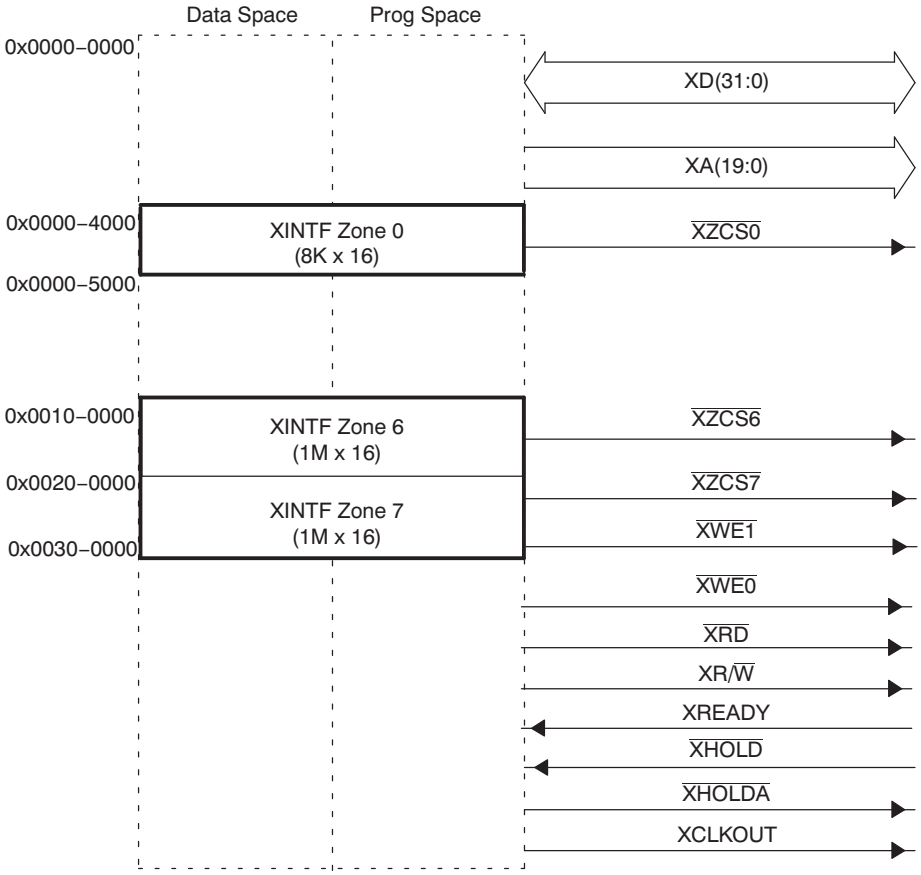



**Figure 4-18. External Interface Block Diagram**

Figure 4-19 and Figure 4-20 show typical 16-bit and 32-bit data bus XINTF connections, illustrating how the functionality of the XA0 and XWE1 signals change, depending on the configuration. Table 4-20 defines XINTF configuration and control registers.

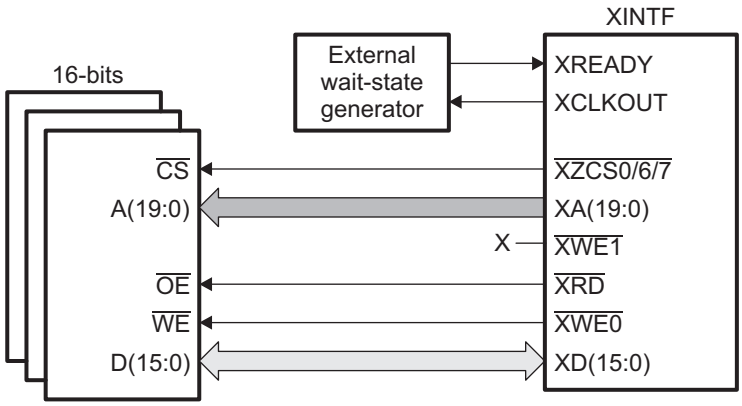



**Figure 4-19. Typical 16-bit Data Bus XINTF Connections**

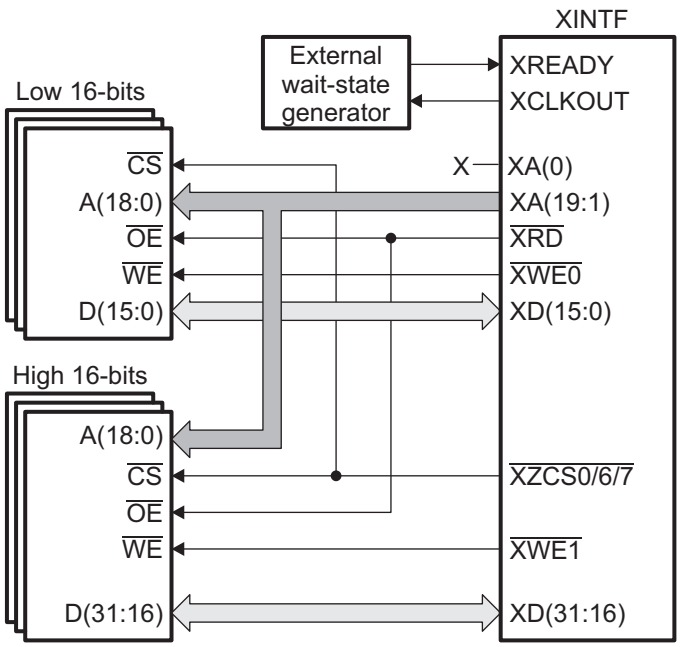



**Figure 4-20. Typical 32-bit Data Bus XINTF Connections**

**Table 4-20. XINTF Configuration and Control Register Mapping**

| NAME | ADDRESS | SIZE (x16) | DESCRIPTION |
|---|---|---|---|
| XTIMING0 | 0x00−0B20 | 2 | XINTF Timing Register, Zone 0 |
| XTIMING6(1) | 0x00−0B2C | 2 | XINTF Timing Register, Zone 6 |
| XTIMING7 | 0x00−0B2E | 2 | XINTF Timing Register, Zone 7 |
| XINTCNF2(2) | 0x00−0B34 | 2 | XINTF Configuration Register |
| XBANK | 0x00−0B38 | 1 | XINTF Bank Control Register |
| XREVISION | 0x00−0B3A | 1 | XINTF Revision Register |
| XRESET | 0x00−0B3D | 1 | XINTF Reset Register |

(1) XTIMING1 - XTIMING5 are reserved for future expansion and are not currently used.
(2) XINTCNF1 is reserved and not currently used.

# 5 Device Support

Texas Instruments (TI) offers an extensive line of development tools for the C28x™ generation of MCUs, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

The following products support development of 2834x-based applications:

**Software Development Tools**

- Code Composer Studio™ Integrated Development Environment (IDE)
  - C/C++ Compiler
  - Code generation tools
  - Assembler/Linker
  - Cycle Accurate Simulator
- Application algorithms
- Sample applications code

**Hardware Development Tools**

- Development board
- Evaluation modules
- JTAG-based emulators - SPI515, XDS510PP, XDS510PP Plus, XDS510USB
- Universal 5-V dc power supply
- Documentation and cables

## 5.1 Device and Development Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ MCU devices and support tools. Each TMS320™ commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., **TMS**320C28345). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

**TMX** Experimental device that is not necessarily representative of the final device's electrical specifications

**TMP** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

**TMS** Fully qualified production device

Support tool development evolutionary flow:

**TMDX** Development-support product that has not yet completed Texas Instruments internal qualification testing

**TMDS** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped against the following disclaimer:
"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ZFE) and temperature range (for example, T). Figure 5-1 provides a legend for reading the complete device name for any family member.

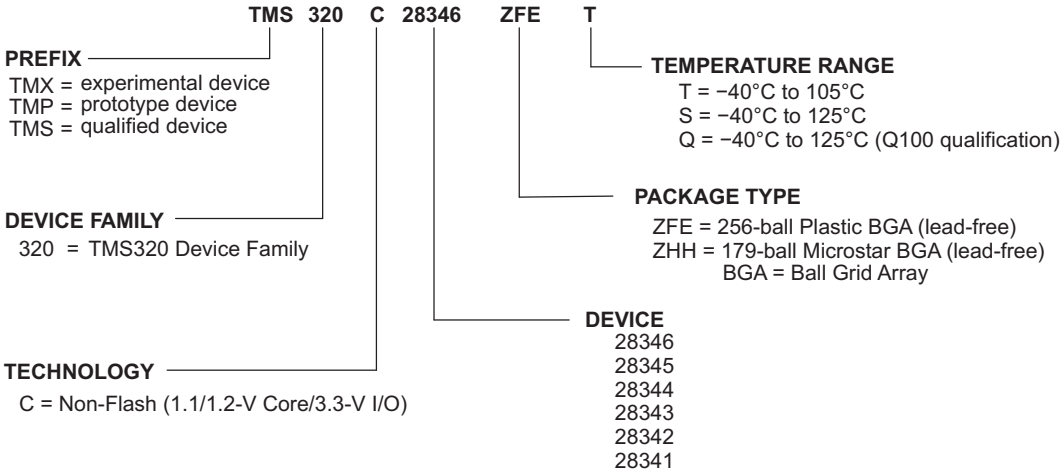



**Figure 5-1. Example of C2834x Device Nomenclature**

## 5.2 Documentation Support

Extensive documentation supports all of the TMS320™ DSP family generations of devices from product announcement through applications development. The types of documentation available include: data sheets and data manuals, with design specifications; and hardware and software applications.

Table 5-1 shows the peripheral reference guides appropriate for use with the devices in this data manual. See the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (literature number SPRU566) for more information on types of peripherals.

**Table 5-1. TMS320x2834x Delfino Peripheral Selection Guide**

| PERIPHERAL GUIDE | LITERATURE NUMBER | TYPE(1) | 28346, 28345, 28344, 28343, 28342, 28341 |
|---|---|---|---|
| TMS320x2834x Delfino System Control and Interrupts | SPRUFN1 | - | X |
| TMS320x2834x Delfino External Interface (XINTF) | SPRUFN4 | 1 | X |
| TMS320x2834x Delfino Enhanced Controller Area Network (eCAN) | SPRUEU4 | 0 | X |
| TMS320x2834x Delfino Multichannel Buffered Serial Port (McBSP) | SPRUG80 | 1 | X |
| TMS320x2834x Delfino Serial Communications Interface (SCI) | SPRUG75 | 0 | X |
| TMS320x2834x Delfino Serial Peripheral Interface (SPI) | SPRUG73 | 0 | X |
| TMS320x2834x Delfino Boot ROM | SPRUFN5 | - | X |
| TMS320x2834x Delfino Enhanced Quadrature Encoder Pulse (eQEP) Module | SPRUG74 | 0 | X |
| TMS320x2834x Delfino Enhanced Pulse Width Modulator (ePWM) Module | SPRUFZ6 | 0 | X |
| TMS320x2834x Delfino Enhanced Capture (eCAP) Module | SPRUG79 | 0 | X |
| TMS320x2834x Delfino Inter-Integrated Circuit (I2C) Module | SPRUG76 | 0 | X |
| TMS320x2834x Delfino High-Resolution Pulse-Width Modulator (HRPWM) | SPRUG77 | 0 | X |
| TMS320x2834x Delfino Direct Memory Access (DMA) Module | SPRUG78 | 0 | X |

(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. These device-specific differences are listed in the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (literature number SPRU566) and in the peripheral reference guides.

Useful reference documentation includes:

**CPU User's Guides**

**SPRU430** **TMS320C28x CPU and Instruction Set Reference Guide**. This document describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x fixed-point digital signal processors (DSPs). It also describes emulation features available on these DSPs.

**SPRUEO2** **TMS320C28x Floating Point Unit and Instruction Set Reference Guide.** This document describes the floating-point unit and includes the instructions for the FPU.

**Peripheral Guides**

**SPRU566** **TMS320x28xx, 28xxx DSP Peripheral Reference Guide**. This document describes the peripheral reference guides of the 28x digital signal processors (DSPs).

**SPRUFN1** **TMS320x2834x Delfino System Control and Interrupts Reference Guide**. This document describes the various interrupts and system control features of the x2834x microcontroller (MCUs).

**SPRUFN4** **TMS320x2834x Delfino External Interface (XINTF) Reference Guide**. This document describes the XINTF, which is a nonmultiplexed asynchronous bus, as it is used on the x2834x device.

**SPRUFN5** **TMS320x2834x Delfino Boot ROM Reference Guide**. This document describes the purpose and features of the bootloader (factory-programmed boot-loading software) and provides examples of code. It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.

**SPRUG80** **TMS320x2834x Delfino Multichannel Buffered Serial Port (McBSP) Reference Guide**. This document describes the McBSP available on the x2834x devices. The McBSPs allow direct interface between a microcontroller (MCU) and other devices in a system.

**SPRUG78** **TMS320x2834x Delfino Direct Memory Access (DMA) Reference Guide**. This document describes the DMA on the x2834x microcontroller (MCUs).

**SPRUFZ6** **TMS320x2834x Delfino Enhanced Pulse Width Modulator (ePWM) Module Reference Guide**. This document describes the main areas of the enhanced pulse width modulator that include digital motor control, switch mode power supply control, UPS (uninterruptible power supplies), and other forms of power conversion.

**SPRUG77** **TMS320x2834x Delfino High-Resolution Pulse Width Modulator (HRPWM) Reference Guide**. This document describes the operation of the high-resolution extension to the pulse width modulator (HRPWM).

**SPRUG79** **TMS320x2834x Delfino Enhanced Capture (eCAP) Module Reference Guide**. This document describes the enhanced capture module. It includes the module description and registers.

**SPRUG74** **TMS320x2834x Delfino Enhanced Quadrature Encoder Pulse (eQEP) Module Reference Guide**. This document describes the eQEP module, which is used for interfacing with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine in high performance motion and position control systems. It includes the module description and registers.

**SPRUEU4** **TMS320x2834x Delfino Enhanced Controller Area Network (eCAN) Reference Guide**. This document describes the eCAN that uses established protocol to communicate serially with other controllers in electrically noisy environments.

**SPRUG75** **TMS320x2834x Delfino Serial Communication Interface (SCI) Reference Guide**. This document describes the SCI, which is a two-wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format.

**SPRUG73** **TMS320x2834x Delfino Serial Peripheral Interface (SPI) Reference Guide**. This document describes the SPI - a high-speed synchronous serial input/output (I/O) port - that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate.

**SPRUG76** **TMS320x2834x Delfino Inter-Integrated Circuit (I2C) Reference Guide**. This document describes the features and operation of the inter-integrated circuit (I2C) module.

**Tools Guides**

**SPRU513** **TMS320C28x Assembly Language Tools v5.0.0 User's Guide**. This document describes the assembly language tools (assembler and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS320C28x device.

**SPRU514** **TMS320C28x Optimizing C/C++ Compiler v5.0.0 User's Guide**. This document describes the TMS320C28x™ C/C++ compiler. This compiler accepts ANSI standard C/C++ source code and produces TMS320 DSP assembly language source code for the TMS320C28x device.

**SPRU608** **TMS320C28x Instruction Set Simulator Technical Overview**. This document describes the simulator, available within the Code Composer Studio for TMS320C2000 IDE, that simulates the instruction set of the C28x™ core.

**SPRU625** **TMS320C28x DSP/BIOS 5.32 Application Programming Interface (API) Reference Guide.** This document describes development using DSP/BIOS.

**Application Reports and Software**

Key Links Include:

1. C2000 Get Started - www.ti.com/c2000getstarted
2. C2000 Digital Motor Control Software Library - www.ti.com/c2000appsw
3. C2000 Digital Power Supply Software Library - www.ti.com/dpslib
4. DSP Power Management Reference Designs - www.ti.com/dsppower

**SPRAAN9** **C28x FPU Primer**

provides an overview of the floating-point unit (FPU) in the TMS320F28335, TMS320F28334, and TMS320F28332 Digital Signal Controller (DSC) devices.

**SPRAAM0** **Getting Started With TMS320C28x Digital Signal Controllers** is organized by development flow and functional areas to make your design effort as seamless as possible. Tips on getting started with C28x™ DSP software and hardware development are provided to aid in your initial design and debug efforts. Each section includes pointers to valuable information including technical documentation, software, and tools for use in each phase of design.

**SPRA958** **Running an Application from Internal Flash Memory on the TMS320F28xxx DSP** covers the requirements needed to properly configure application software for execution from on-chip flash memory. Requirements for both DSP/BIOS™ and non-DSP/BIOS projects are presented. Example code projects are included.

**SPRAA85** **Programming TMS320x28xx and 28xxx Peripherals in C/C++** explores a hardware abstraction layer implementation to make C/C++ coding easier on 28x DSPs. This method is compared to traditional #define macros and topics of code efficiency and special case registers are also addressed.

**SPRAA88** **Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller** presents a method for utilizing the on-chip pulse width modulated (PWM) signal generators on the TMS320F280x family of digital signal controllers as a digital-to-analog converter (DAC).

**SPRAA91** **TMS320F280x Digital Signal Controller USB Connectivity Using the TUSB3410 USB-to-UART Bridge Chip** presents hardware connections as well as software preparation and operation of the development system using a simple communication echo program.

**SPRAAH1** **Using the Enhanced Quadrature Encoder Pulse (eQEP) Module in TMS320x280x, 28xxx as a Dedicated Capture** provides a guide for the use of the eQEP module as a dedicated capture unit and is applicable to the TMS320x280x, 28xxx family of processors.

**SPRAAI1** **Using the ePWM Module for 0% – 100% Duty Cycle Control** provides a guide for the use of the ePWM module to provide 0% to 100% duty cycle control and is applicable to the TMS320x280x family of processors.

**SPRAAD5** **Power Line Communication for Lighting Applications Using Binary Phase Shift Keying (BPSK) with a Single DSP Controller** presents a complete implementation of a power line modem following CEA-709 protocol using a single DSP.

**SPRAB26** **TMS320x2833x/2823x to TMS320x2834x Delfino Migration Overview**. This application report describes differences between the Texas Instruments TMS320x2833x/2823x and the TMS320x2834x devices to assist in application migration.

**SPRA820** **Online Stack Overflow Detection on the TMS320C28x DSP** presents the methodology for online stack overflow detection on the TMS320C28x™ DSP. C-source code is provided that contains functions for implementing the overflow detection on both DSP/BIOS™ and non-DSP/BIOS applications.

**SPRA806** **An Easy Way of Creating a C-callable Assembly Function for the TMS320C28x DSP** provides instructions and suggestions to configure the C compiler to assist with understanding of parameter-passing conventions and environments expected by the C compiler.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320 DSP newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 DSP customers on product information.

Updated information on the TMS320 DSP controllers can be found on the worldwide web at: http://www.ti.com.

To send comments regarding this data manual (literature number SPRS516), click on the *Submit Documentation Feedback* link at the bottom of the page. For questions and support, contact the Product Information Center listed at the http://www.ti.com/sc/docs/pic/home.htm site.

# 6 Electrical Specifications

This section provides the absolute maximum ratings and the recommended operating conditions.

## 6.1 Absolute Maximum Ratings(1) (2)

| | | |
|---|---|---|
| Supply voltage range, $V_{DDIO}$ | with respect to $V_{SS}$ | –0.3 V to 4.0 V |
| Supply voltage range, $V_{DD}$ | with respect to $V_{SS}$ | –0.3 V to 1.5 V |
| Supply voltage range, $V_{DD18}$ | with respect to $V_{SS}$ | –0.3 V to 2.4 V |
| Input voltage range, $V_{IN}$ (3.3 V) | | –0.3 V to 4.0 V |
| Input voltage range, $V_{IN}$ (1.8 V) | | –0.3 V to 2.4 V |
| Output voltage range, $V_O$ | | –0.3 V to 4.0 V |
| Input clamp current, $I_{IK}$ ($V_{IN}$ < 0 or $V_{IN}$ > $V_{DDIO}$)(3) | | ±20 mA |
| Output clamp current, $I_{OK}$ ($V_O$ < 0 or $V_O$ > $V_{DDIO}$) | | ±20 mA |
| Junction temperature range, $T_J$ (4) | | –40°C to 150°C |
| Storage temperature range, $T_{stg}$ (4) | | –65°C to 150°C |

(1) Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Section 6.2 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to $V_{SS}$, unless otherwise noted.
(3) Continuous clamp current per pin is ±2 mA.
(4) Long-term high-temperature storage and/or extended use at maximum temperature conditions may result in a reduction of overall device life. For additional information, see *IC Package Thermal Metrics Application Report* (literature number SPRA953) and *Reliability Data for TMS320LF24xx and TMS320F28xx Devices Application Report* (literature number SPRA963).

## 6.2 Recommended Operating Conditions

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Device supply voltage, I/O, $V_{DDIO}$ | | 3.14 | 3.3 | 3.46 | V |
| Device supply voltage CPU, $V_{DD}$ | 300-MHz devices | 1.14 | 1.2 | 1.26 | V |
| | 200-MHz devices | 1.05 | 1.1 | 1.16 | V |
| Supply ground, $V_{SS}$, $V_{SSIO}$ | | | 0 | | V |
| Oscillator supply ground, $V_{SSK}$ | | | 0 | | V |
| PLL/oscillator supply, $V_{DD18}$ | | 1.71 | 1.8 | 1.89 | V |
| Device clock frequency (system clock), $f_{SYSCLKOUT}$ | C28346/C28344/C28342 ($V_{DD}$ = 1.2 V ± 5%) | 2 | | 300 | MHz |
| | C28345/C28343/C28341 ($V_{DD}$ = 1.1 V ± 5%) | 2 | | 200 | MHz |
| High-level input voltage, $V_{IH}$ (3.3 V) | | 2 | | $V_{DDIO}$ + 0.3 | V |
| High-level input voltage, $V_{IH}$ (1.8 V) | | 0.7 * $V_{DD18}$ | | | V |
| Low-level input voltage, $V_{IL}$ (3.3 V) | | $V_{SS}$ – 0.3 | | 0.8 | V |
| Low-level input voltage, $V_{IL}$ (1.8 V) | | | | 0.3 * $V_{DD18}$ | V |
| High-level output source current, $V_{OH}$ = 2.4 V, $I_{OH}$ | All I/Os | | | –4 | mA |
| Low-level output sink current, $V_{OL}$ = $V_{OL}$ MAX, $I_{OL}$ | All I/Os | | | 4 | mA |
| Junction temperature, $T_J$ (1) | T version | –40 | | 105 | °C |
| | S version | –40 | | 125 | °C |
| | Q version (Q100 Qualification) | –40 | | 125 | °C |

(1) $T_A$ (Ambient temperature) is product- and application-dependent and can go up to the specified $T_J$ max of the device. See Section 6.5, Thermal Design Considerations.

## 6.3 Electrical Characteristics

over recommended operating conditions (unless otherwise noted)

| PARAMETER | | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High-level output voltage | | $I_{OH}$ = $I_{OH}$ MAX | | 2.4 | | | V |
| | | | $I_{OH}$ = 50 µA | | $V_{DDIO}$ – 0.2 | | | |
| $V_{OL}$ | Low-level output voltage | | $I_{OL}$ = $I_{OL}$ MAX | | | | 0.4 | V |
| $I_{IL}$ | Input current (low level) | Pin with pullup enabled | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = 0 V | All I/Os (including $\overline{XRS}$) | –190 | | –130 | µA |
| | | Pin with pulldown enabled | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = 0 V | | | | ±15 | |
| $I_{IH}$ | Input current (high level) | Pin with pullup enabled | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = $V_{DDIO}$ | | | | ±3 | µA |
| | | Pin with pulldown enabled | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = $V_{DDIO}$ | | 100 | | 175 | |
| $I_{OZ}$ | Output current, pullup or pulldown disabled | | $V_O$ = $V_{DDIO}$ or 0 V | | | | ±15 | µA |
| $C_I$ | Input capacitance | | | | | 2 | | pF |

## 6.4 Current Consumption

**Table 6-1. TMS320C28346/C28344(1) Current Consumption by Power-Supply Pins at 300-MHz SYSCLKOUT**

| MODE | TEST CONDITIONS | $I_{DD}$ | | | $I_{DDIO}$ (2) | | | $I_{DD18}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 25°C | 105°C | 125°C | 25°C | 105°C | 125°C | 25°C | 105°C | 125°C |
| Typical Operational | The following peripheral clocks are enabled:<br>• ePWM1/2/3/4/5/6/7/8/9<br>• eCAP1/2/3<br>• eQEP1/2/3<br>• eCAN-A<br>• SCI-A/B (FIFO mode)<br>• SPI-A (FIFO mode)<br>• McBSP-A<br>• I2C<br>• XINTF<br>• DMA<br>• CPU Timer 0/1/2<br>All PWM pins are toggled at 300 kHz.<br>All I/O pins are left unconnected.<br>XCLKOUT is turned off. Pullups on output pins and XINTF pins are disabled.(3) | 335 mA | 555 mA | 740 mA | 75 mA | 75 mA | 80 mA | 50 mA | 47 mA | 45 mA |
| IDLE | XCLKOUT is turned off.<br>Peripheral clocks are off. | 205 mA | 425 mA | 610 mA | 15 mA | 15 mA | 18 mA | 50 mA | 47 mA | 45 mA |
| STANDBY | Peripheral clocks are off. | 140 mA | 360 mA | 545 mA | 15 mA | 15 mA | 18 mA | 50 mA | 47 mA | 45 mA |
| HALT | Peripheral clocks are off.<br>Input clock is disabled.(4) | 135 mA | 355 mA | 540 mA | 15 mA | 15 mA | 18 mA | 550 µA | 550 µA | 550 µA |

(1) The $I_{DD}$ numbers in this table are valid for the TMS320C28346 and TMS320C28344 devices only. For the TMS320C28342 device, subtract the $I_{DD}$ current numbers for those peripherals that do not exist on this device (see Table 6-3) from the $I_{DD}$ current numbers shown in this table.
(2) $I_{DDIO}$ current is dependent on the electrical loading on the I/O pins.
(3) The following is done in a loop:
- Data is continuously transmitted out of the SCI-A, SCI-B, SPI-A, McBSP-A, and eCAN-A ports.
- Floating-point multiplication and addition are performed.
- 32-bit read/write of the XINTF is performed.
- DMA channels 1 and 2 transfer data from SARAM to SARAM.
- GPIO19 is toggled.

(4) If a quartz crystal or ceramic resonator is used as the clock source, the HALT mode shuts down the internal oscillator.

**NOTE**

The $I_{DD}$ numbers in Table 6-1 are valid for the TMS320C28346 and TMS320C28344 devices only. For the TMS320C28342 device, subtract the $I_{DD}$ current numbers for those peripherals that do not exist on this device (see Table 6-3) from the $I_{DD}$ current numbers shown in Table 6-1.

**NOTE**

The peripheral - I/O multiplexing implemented in the device prevents all available peripherals from being used at the same time. This is because more than one peripheral function may share an I/O pin. It is, however, possible to turn on the clocks to all the peripherals at the same time, although such a configuration is not useful. If this is done, the current drawn by the device will be more than the numbers specified in the current consumption tables.

**Table 6-2. TMS320C28345/C28343[1] Current Consumption by Power-Supply Pins at 200-MHz SYSCLKOUT**

| MODE | TEST CONDITIONS | $I_{DD}$ | | | $I_{DDIO}$ [2] | | | $I_{DD18}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 25°C | 105°C | 125°C | 25°C | 105°C | 125°C | 25°C | 105°C | 125°C |
| Typical operation | The following peripheral clocks are enabled:<br>• ePWM1/2/3/4/5/6/7/8/9<br>• eCAP1/2/3<br>• eQEP1/2/3<br>• eCAN-A<br>• SCI-A/B (FIFO mode)<br>• SPI-A (FIFO mode)<br>• McBSP-A<br>• I2C<br>• XINTF<br>• DMA<br>• CPU TImers 0/1/2<br>All PWM pins are toggled at 200 kHz. All I/O pins are left unconnected. XCLKOUT is turned off. Pullups on output pins and XINTF pins are disabled.[3] | 200 mA | 380 mA | 500 mA | 45 mA | 45 mA | 45 mA | 45 mA | 43 mA | 40 mA |
| IDLE | Peripheral clocks are off. XCLKOUT is turned off. | 95 mA | 275 mA | 395 mA | 15 mA | 15 mA | 18 mA | 45 mA | 43 mA | 40 mA |
| STANDBY | Peripheral clocks are off. | 45 mA | 225 mA | 345 mA | 15 mA | 15 mA | 18 mA | 45 mA | 43 mA | 40 mA |
| HALT | Peripheral clocks are off. Input clock is disabled.[4] | 40 mA | 220 mA | 340 mA | 15 mA | 15 mA | 18 mA | 550 µA | 550 µA | 550 µA |

(1) The $I_{DD}$ numbers in this table are valid for the TMS320C28345 and TMS320C28343 devices only. For the TMS320C28341 device, subtract the $I_{DD}$ current numbers for those peripherals that do not exist on this device (see Table 6-3) from the $I_{DD}$ current numbers shown in this table.
(2) $I_{DDIO}$ current is dependent on the electrical loading on the I/O pins.
(3) The following is done in a loop:
- Data is continuously transmitted out of the SCI-A, SCI-B, SPI-A, McBSP-A, and eCAN-A ports.
- Floating-point multiplication and addition are performed.
- 32-bit read/write of the XINTF is performed.
- DMA channels 1 and 2 transfer data from SARAM to SARAM.
- GPIO19 is toggled.

(4) If a quartz crystal or ceramic resonator is used as the clock source, the HALT mode shuts down the internal oscillator.

**NOTE**

The $I_{DD}$ numbers in Table 6-2 are valid for the TMS320C28345 and TMS320C28343 devices only. For the TMS320C28341 device, subtract the $I_{DD}$ current numbers for those peripherals that do not exist on this device (see Table 6-3) from the $I_{DD}$ current numbers shown in Table 6-2.

### 6.4.1 Reducing Current Consumption

Methods of reducing current consumption include the following:

- Turn off the clock to any peripheral module that is not used in a given application since each peripheral unit has an individual clock-enable bit. Table 6-3 indicates the typical reduction in current consumption achieved by turning off the clocks.
- Use any one of the three low-power modes to reduce current even further.
- Turn off XCLKOUT, reducing $I_{DDIO}$ current consumption by 15 mA (typical).
- Disable the pullups on pins that assume an output function and on XINTF pins for significant savings in $I_{DDIO}$.

**NOTE**

The TMS320C2834x devices are manufactured in a high-performance process node. Compared to the previous generation of the C28x devices, this process has more leakage current. Leakage current is significantly impacted by the operating temperature, and the increase in current with temperature is non-linear. The total power for a given operating condition includes switching/active power plus leakage power. Low-power HALT mode power is due to the leakage current alone.

Figure 6-1 shows the typical leakage current across temperature.

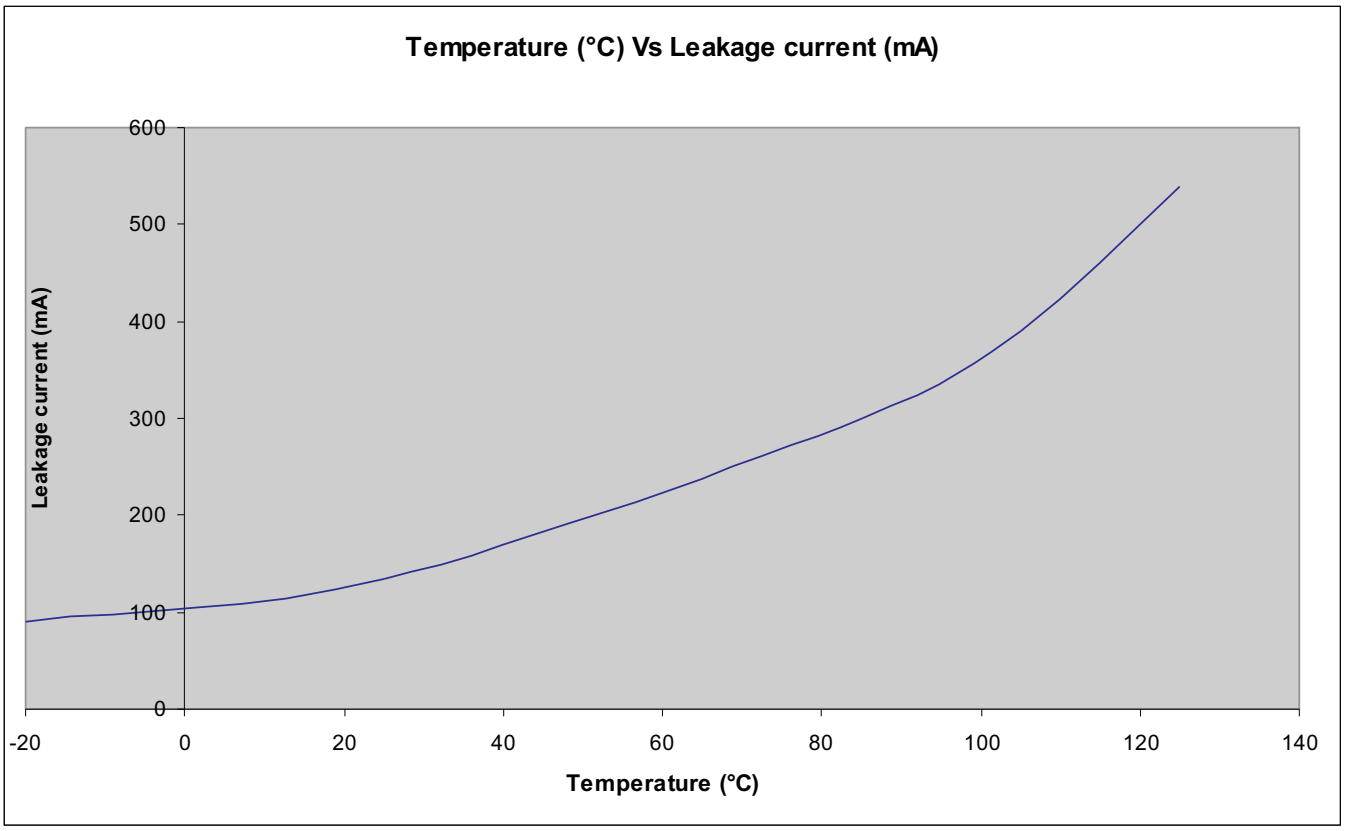



**Figure 6-1. Temperature Versus Leakage Current (Typical)**

**Table 6-3. Typical Current Consumption by Various Peripherals(1)**

| PERIPHERAL MODULE | $I_{DD}$ CURRENT REDUCTION (mA) |
|---|---|
| I2C | 5 |
| eQEP | 5 |
| ePWM | 3 |
| eCAP | 1 |
| SCI | 4 |
| SPI | 4 |
| eCAN | 2 |
| McBSP | 8 |
| CPU - Timer | 1 |
| XINTF | 4(2) |
| DMA | 7 |
| FPU | 8 |

(1) All peripheral clocks (except CPU timer clocks) are disabled upon reset. Writing to/reading from peripheral registers is possible only after the peripheral clocks are turned on.

(2) Operating the XINTF bus has a significant effect on IDDIO current. It will increase considerably based on the following:
- How many address/data pins toggle from one cycle to another
- How fast they toggle
- Whether 16-bit or 32-bit interface is used and
- The load on these pins.
- Whether internal pullups are enabled on the XINTF pins.

## 6.5 Thermal Design Considerations

Based on the end application design and operational profile, the $I_{DD}$ and $I_{DDIO}$ currents could vary. Systems that exceed the recommended maximum power dissipation in the end product may require additional thermal enhancements. Ambient temperature ($T_A$) varies with the end application and product design. The critical factor that affects reliability and functionality is $T_J$, the junction temperature, not the ambient temperature. Hence, care should be taken to keep $T_J$ within the specified limits. $T_{case}$ should be measured to estimate the operating junction temperature $T_J$. $T_{case}$ is normally measured at the center of the package top-side surface. The thermal application reports *IC Package Thermal Metrics* (literature number SPRA953) and *Reliability Data for TMS320LF24xx and TMS320F28xx Devices* (literature number SPRA963) help to understand the thermal metrics and definitions.

## 6.6 Emulator Connection Without Signal Buffering for the MCU

Figure 6-2 shows the connection between the MCU and JTAG header for a single-processor configuration. If the distance between the JTAG header and the MCU is greater than 6 inches, the emulation signals must be buffered. If the distance is less than 6 inches, buffering is typically not needed. Figure 6-2 shows the simpler, no-buffering situation. For the pullup/pulldown resistor values, see the pin description section. For details on buffering JTAG signals and multiple processor connections, see *TMS320F/C24x DSP Controllers CPU and Instruction Set Reference Guide* (literature number SPRU160).

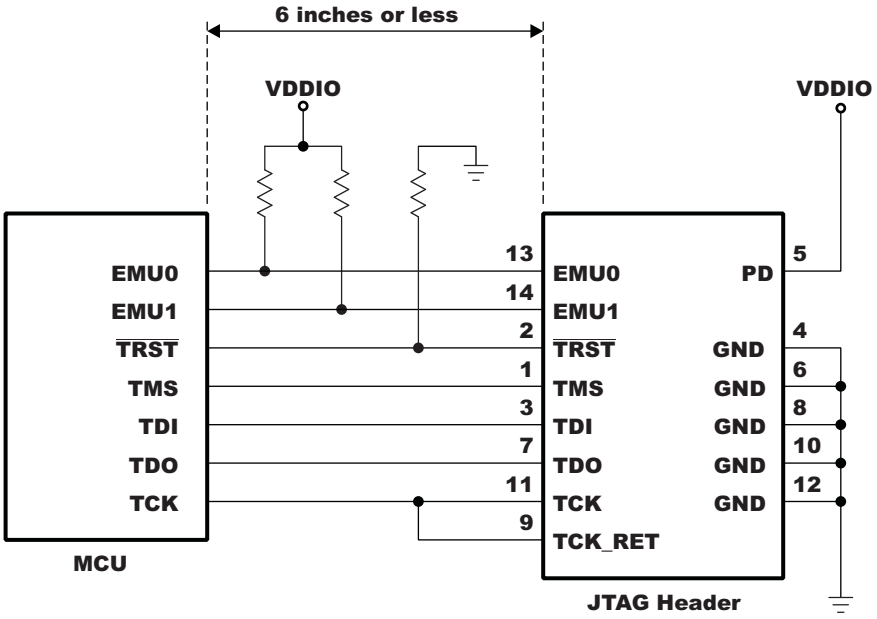



**Figure 6-2. Emulator Connection Without Signal Buffering for the MCU**

## 6.7 Timing Parameter Symbology

Timing parameter symbols used are created in accordance with JEDEC Standard 100. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

| Lowercase subscripts and their meanings: | | Letters and symbols and their meanings: | |
|---|---|---|---|
| a | access time | H | High |
| c | cycle time (period) | L | Low |
| d | delay time | V | Valid |
| f | fall time | X | Unknown, changing, or don't care level |
| h | hold time | Z | High impedance |
| r | rise time | | |
| su | setup time | | |
| t | transition time | | |
| v | valid time | | |
| w | pulse duration (width) | | |

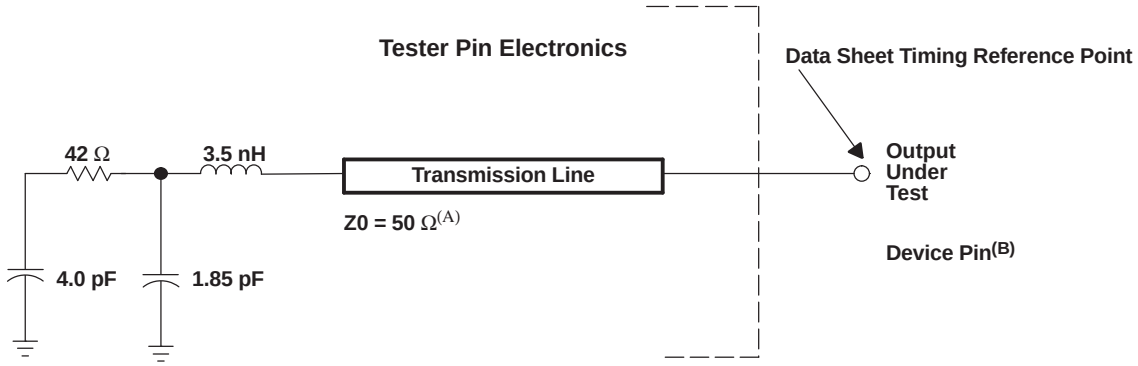
### 6.7.1 General Notes on Timing Parameters

All output signals from the 28x devices (including XCLKOUT) are derived from an internal clock such that all output transitions for a given half-cycle occur with a minimum of skewing relative to each other.

The signal combinations shown in the following timing diagrams may not necessarily represent actual cycles. For actual cycle examples, see the appropriate cycle description section of this document.

### 6.7.2 Test Load Circuit

This test load circuit is used to measure all switching characteristics provided in this document.




A. Input requirements in this data sheet are tested with an input slew rate of < 4 Volts per nanosecond (4 V/ns) at the device pin.

B. The data sheet provides timing at the device pin. For output timing analysis, the tester pin electronics and its transmission line effects must be taken into account. A transmission line with a delay of 2 ns or longer can be used to produce the desired transmission line effect. The transmission line is intended as a load only. It is not necessary to add or subtract the transmission line delay (2 ns or longer) from the data sheet timing.

**Figure 6-3. 3.3-V Test Load Circuit**

### 6.7.3 Device Clock Table

This section provides the timing requirements and switching characteristics for the various clock options available. Table 6-4 through Table 6-5 list the cycle times of various clocks.

**Table 6-4. Clocking and Nomenclature (300-MHz Devices)**

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| On-chip oscillator clock (crystal/resonator–X1/X2) | | $t_{c(OSC)}$, Cycle time | 33.3 | | 125 | ns |
| | | Frequency | 8 | | 30 | MHz |
| XCLKIN(1) | PLL enabled | $t_{c(CI)}$, Cycle time (C8) | 6.67 | | 50 | ns |
| | | Frequency | 2 | | 150 | MHz |
| | PLL disabled | $t_{c(CI)}$, Cycle time (C8) | 6.67 | | 250 | ns |
| | | Frequency | 4 | | 150 | MHz |
| X1(1) | PLL enabled | $t_{c(CI)}$, Cycle time (C8) | 10 | | 50 | ns |
| | | Frequency | 2 | | 100 | MHz |
| | PLL disabled | $t_{c(CI)}$, Cycle time (C8) | 10 | | 250 | ns |
| | | Frequency | 4 | | 100 | MHz |
| SYSCLKOUT | | $t_{c(SCO)}$, Cycle time | 3.33 | | 500 | ns |
| | | Frequency | 2 | | 300 | MHz |
| XCLKOUT | | $t_{c(XCO)}$, Cycle time | 13.3 | | 2000 | ns |
| | | Frequency | 0.5 | | 75(2) | MHz |
| HSPCLK/EXTADCCLK(3) | | $t_{c(HCO)}$, Cycle time | 25 | | | ns |
| | | Frequency | | | 40 | MHz |
| LSPCLK(4) | | $t_{c(LCO)}$, Cycle time | 6.67 | 13.3(5) | | ns |
| | | Frequency | | 75(5) | 150 | MHz |

(1) The input clock frequency and PLLCR[DIV] values should be chosen such that the output frequency of the PLL(VCOCLK) lies between 400 MHz to 600 MHz.
(2) Although the maximum XCLKOUT frequency is 75 MHz, this value may not be attainable depending on SYSCLKOUT and available prescalers.
(3) This frequency is limited by GPIO switching characteristics.
(4) Lower LSPCLK and HSPCLK will reduce device power consumption.
(5) This is the value if SYSCLKOUT = 300 MHz.

**Table 6-5. Clocking and Nomenclature (200-MHz Devices)**

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| On-chip oscillator clock (crystal/resonator–X1/X2) | | $t_{c(OSC)}$, Cycle time | 33.3 | | 125 | ns |
| | | Frequency | 8 | | 30 | MHz |
| XCLKIN[1] | PLL enabled | $t_{c(CI)}$, Cycle time (C8) | 6.67 | | 50 | ns |
| | | Frequency | 2 | | 150 | MHz |
| | PLL disabled | $t_{c(CI)}$, Cycle time (C8) | 6.67 | | 250 | ns |
| | | Frequency | 4 | | 150 | MHz |
| X1[1] | PLL enabled | $t_{c(CI)}$, Cycle time (C8) | 10 | | 50 | ns |
| | | Frequency | 2 | | 100 | MHz |
| | PLL disabled | $t_{c(CI)}$, Cycle time (C8) | 10 | | 250 | ns |
| | | Frequency | 4 | | 100 | MHz |
| SYSCLKOUT | | $t_{c(SCO)}$, Cycle time | 5 | | 500 | ns |
| | | Frequency | 2 | | 200 | MHz |
| XCLKOUT | | $t_{c(XCO)}$, Cycle time | 13.3 | | 2000 | ns |
| | | Frequency | 0.5 | | 75[2] | MHz |
| HSPCLK/EXTADCCLK[3] | | $t_{c(HCO)}$, Cycle time | 8 | | | ns |
| | | Frequency | | | 40 | MHz |
| LSPCLK[4] | | $t_{c(LCO)}$, Cycle time | 10 | 20[5] | | ns |
| | | Frequency | | 50[5] | 100 | MHz |

(1) The input clock frequency and PLLCR[DIV] values should be chosen such that the output frequency of the PLL(VCOCLK) lies between 400 MHz to 600 MHz.
(2) Although the maximum XCLKOUT frequency is 75 MHz, this value may not be attainable depending on SYSCLKOUT and available prescalers.
(3) This frequency is limited by GPIO switching characteristics.
(4) Lower LSPCLK and HSPCLK will reduce device power consumption.
(5) This is the value if SYSCLKOUT = 200 MHz.

## 6.8 Clock Requirements and Characteristics

**Table 6-6. XCLKIN/X1 Timing Requirements – PLL Enabled**

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| C9 | $t_{f(CI)}$ | Fall time, XCLKIN(1) | | 4 | ns |
| C10 | $t_{r(CI)}$ | Rise time, XCLKIN(1) | | 4 | ns |
| C11 | $t_{w(CIL)}$ | Pulse duration, XCLKIN low as a percentage of $t_{c(OSCCLK)}$ (1) | 40 | 60 | % |
| C12 | $t_{w(CIH)}$ | Pulse duration, XCLKIN high as a percentage of $t_{c(OSCCLK)}$ (1) | 40 | 60 | % |

(1) This applies to the X1 pin also.

**Table 6-7. XCLKIN/X1 Timing Requirements – PLL Disabled**

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| C9 | $t_{f(CI)}$ | Fall time, XCLKIN(1) | | 2 | ns |
| C10 | $t_{r(CI)}$ | Rise time, XCLKIN(1) | | 2 | ns |
| C11 | $t_{w(CIL)}$ | Pulse duration, XCLKIN low as a percentage of $t_{c(OSCCLK)}$ (1) | 45 | 55 | % |
| C12 | $t_{w(CIH)}$ | Pulse duration, XCLKIN high as a percentage of $t_{c(OSCCLK)}$ (1) | 45 | 55 | % |

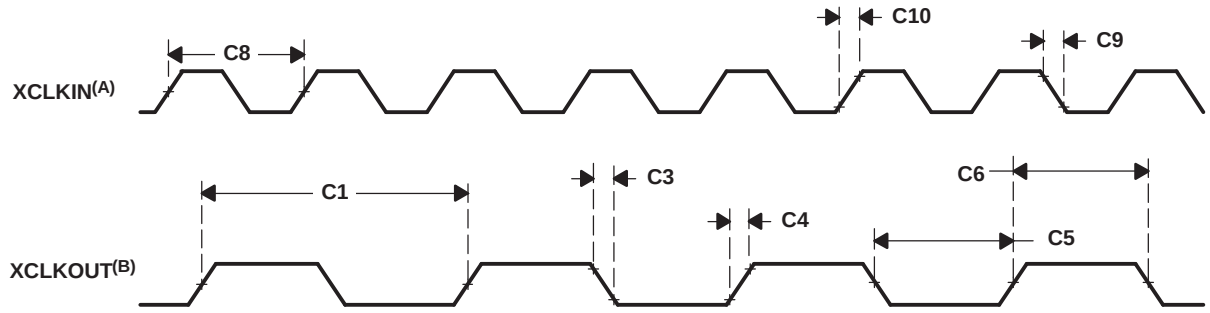
(1) This applies to the X1 pin also.

The possible configuration modes are shown in Table 3-14.

**Table 6-8. XCLKOUT Switching Characteristics (PLL Bypassed or Enabled)(1) (2)**

| NO. | PARAMETER | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| C1 | $t_{c(XCO)}$ | Cycle time, XCLKOUT | 13.3 | | | ns |
| C3 | $t_{f(XCO)}$ | Fall time, XCLKOUT | | 2 | | ns |
| C4 | $t_{r(XCO)}$ | Rise time, XCLKOUT | | 2 | | ns |
| C5 | $t_{w(XCOL)}$ | Pulse duration, XCLKOUT low | H – 2 | | H + 2 | ns |
| C6 | $t_{w(XCOH)}$ | Pulse duration, XCLKOUT high | H – 2 | | H + 2 | ns |
| | $t_p$ | PLL lock time | | | 2600 $t_{c(OSCCLK)}$ (3) | cycles |

(1) A load of 40 pF is assumed for these parameters.
(2) H = 0.5$t_{c(XCO)}$
(3) OSCCLK is either the output of the on-chip oscillator or the output from an external oscillator.




A. The relationship of XCLKIN to XCLKOUT depends on the divide factor chosen. The waveform relationship shown is intended to illustrate the timing parameters only and may differ based on actual configuration.
B. XCLKOUT configured to reflect SYSCLKOUT.

**Figure 6-4. Clock Timing**

## 6.9 Power Sequencing

No special requirements are placed on the power up/down sequence of the various power pins to ensure the correct reset state for all the modules. However, if the 3.3-V transistors in the level shifting output buffers of the I/O pins are powered prior to the 1.1-V/1.2-V transistors, it is possible for the output buffers to turn on, causing a glitch to occur on the pin during power up. To avoid this behavior, power the $V_{DD}$ pins prior to or simultaneously with the $V_{DDIO}$ pins, ensuring that the $V_{DD}$ pins have reached 0.7-V before the $V_{DDIO}$ pins reach 0.7 V. The 1.8-V rail for the PLL and oscillator logic can be powered up along with $V_{DD}/V_{DDIO}$ rails. The 1.8-V rail must be powered even if the PLL is not used. It should never be left unpowered. In any configuration, all the rails should ramp up within $t_{pup}$ (5 ms, typical) to allow early stability of clocks and IOs.

There is a requirement on the $\overline{XRS}$ pin:

- During power up, the $\overline{XRS}$ pin must be held low for $t_{w(RSL1)}$ after the input clock is stable. This is to enable the entire device to start from a known condition.

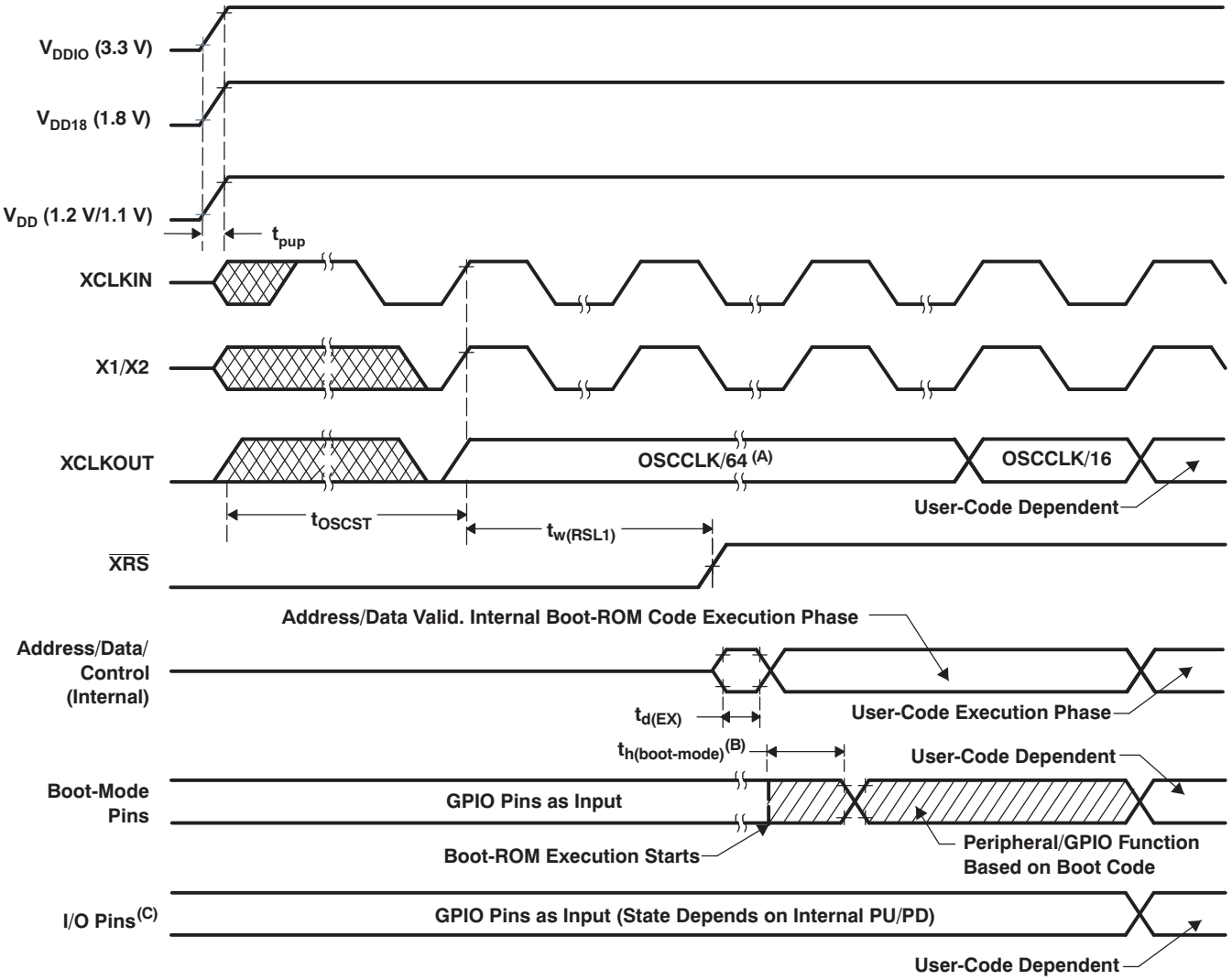
Additionally it is recommended that no voltage larger than a diode drop (0.7 V) should be applied to any pin prior to powering up the device. Voltages applied to pins on an unpowered device can bias internal P-N junctions in unintended ways and produce unpredictable results.

### 6.9.1 Power Management and Supervisory Circuit Solutions

Table 6-9 lists the power management and supervisory circuit solutions for the 2834x devices. LDO selection depends on the total power consumed in the end application. Go to www.ti.com and click on *Power Management* for a complete list of TI power ICs or select the *Power Management Selection Guide* link for specific power reference designs.

**Table 6-9. Power Management and Supervisory Circuit Solutions**

| SUPPLIER | TYPE | PART | DESCRIPTION |
|---|---|---|---|
| Texas Instruments | PMIC | TPS650061 | 3-channel power management IC with one DC/DC switcher, two low-dropout (LDO) regulators (fixed at 3.3 V and 1.8 V), a spread spectrum clock (SSC), and a supervisory circuit solution (SVS) |
| Texas Instruments | PMIC | TPS65001 | 3-channel power management IC with one DC/DC switcher, two externally adjustable low-dropout (LDO) regulators, a spread spectrum clock (SSC), and a supervisory circuit solution (SVS) |
| Texas Instruments | PMIC | TPS65053 | 5-channel power management IC with two step-down converters and three low-input voltage LDOs |
| Texas Instruments | DC/DC | TPS62290 | 1-A step-down DC/DC converter in 2 x 2 SON package |
| Texas Instruments | DC/DC | TPS62260 | 2.25-MHz, 600-mA step-down DC/DC converter in 2 x 2 SON/TSOT23 package |
| Texas Instruments | DC/DC | TPS62240 | 2.25-MHz, 300-mA step-down DC/DC converter in 2 x 2 SON/TSOT23 package |
| Texas Instruments | DC/DC | TPS62420 | Dual, adjustable, 600-mA and 1000-mA, 2.25-MHz step-down converter with 1-Wire® Interface in QFN |
| Texas Instruments | LDO | TPS71718 | Low-noise, high-bandwidth PSRR, low-dropout 150-mA linear regulator |
| Texas Instruments | LDO | TPS79601 | Ultra low-noise, high PSRR, fast, RF, 1-A, low-dropout linear regulator |
| Texas Instruments | LDO | TPS73701 | 1-A low-dropout regulator with reverse current protection |
| Texas Instruments | LDO | TPS73433 | 250-mA, low-quiescent current, ultra-low noise, high PSRR, low-dropout linear regulator |
| Texas Instruments | LDO | TPS71718 | Low-noise, high-bandwidth PSRR, low-dropout 150-mA linear regulator |
| Texas Instruments | LDO | TPS72118 | Low input voltage, cap-free 150-mA low-dropout regulators |
| Texas Instruments | SVS | TPS3808 | Open-drain SVS with programmable delay |
| Texas Instruments | SVS | TPS3803 | Low-cost open-drain SVS with 5-μS delay |




A. Upon power up, SYSCLKOUT is OSCCLK/8. Since the XTIMCLK, CLKMODE, and BY4CLKMODE bits in the XINTFCNF2 register come up with a reset state of 1, SYSCLKOUT is further divided by 8 before it applies to XCLKOUT. This explains why XCLKOUT = OSCCLK/64 during this phase. Subsequently, boot ROM changes SYSCLKOUT to OSCLK/2. Because the XTIMCLK register is unchanged by the boot ROM, XCLKOUT is OSCCLK/16 during this phase.

B. After reset, the boot ROM code samples Boot Mode pins. Based on the status of the Boot Mode pin, the boot code branches to destination memory or boot code function. If boot ROM code executes after power-on conditions (in debugger environment), the boot code execution time is based on the current SYSCLKOUT speed. The SYSCLKOUT will be based on user environment and could be with or without PLL enabled.

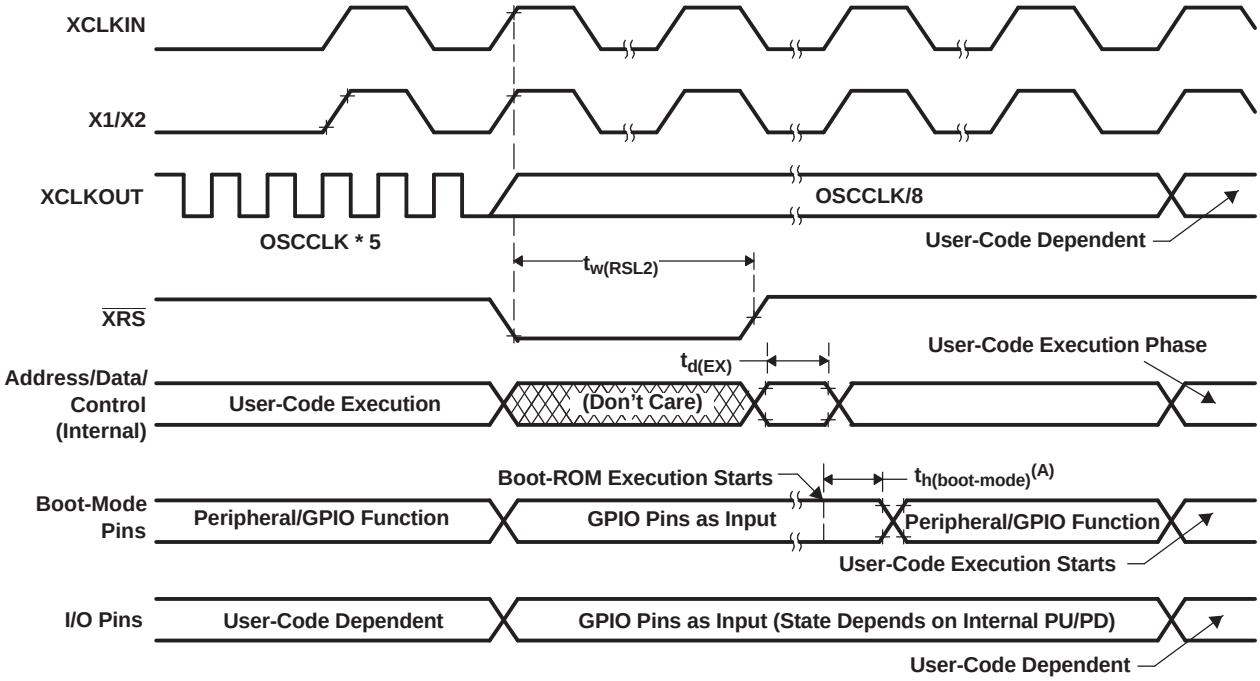
C. See Section 6.9 for requirements to ensure a high-impedance state for GPIO pins during power-up.

**Figure 6-5. Power-on Reset**

### Table 6-10. Reset ($\overline{XRS}$) Timing Requirements

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{w(RSL1)}$ (1) | Pulse duration, stable input clock to $\overline{XRS}$ high | | $64t_{c(OSCCLK)}$ | | | cycles |
| $t_{w(RSL2)}$ | Pulse duration, $\overline{XRS}$ low | Warm reset | $64t_{c(OSCCLK)}$ | | | cycles |
| $t_{w(WDRS)}$ | Pulse duration, reset pulse generated by watchdog | | | $512t_{c(OSCCLK)}$ | | cycles |
| $t_{d(EX)}$ | Delay time, address/data valid after $\overline{XRS}$ high | | | $32t_{c(OSCCLK)}$ | | cycles |
| $t_{OSCST}$ (2) | Oscillator start-up time | | 1 | 10 | | ms |
| $t_{h(boot-mode)}$ | Hold time for boot-mode pins | | $200t_{c(OSCCLK)}$ | | | cycles |
| $t_{pup}$ | Power-up time | | | 5 | | ms |

(1) In addition to the $t_{w(RSL1)}$ requirement, $\overline{XRS}$ has to be low until $V_{DD}$ has reached the minimum operating voltage.
(2) Dependent on crystal/resonator and board design.




A. After reset, the Boot ROM code samples BOOT Mode pins. Based on the status of the Boot Mode pin, the boot code branches to destination memory or boot code function. If Boot ROM code executes after power-on conditions (in debugger environment), the Boot code execution time is based on the current SYSCLKOUT speed. The SYSCLKOUT will be based on user environment and could be with or without PLL enabled.

**Figure 6-6. Warm Reset**

Figure 6-7 shows an example for the effect of writing into PLLCR register. In the first phase, PLLCR = 0x0003 and SYSCLKOUT = OSCCLK x 2. The PLLCR is then written with 0x0007 (setting for OSCCLK x 8). Right after the PLLCR register is written, the PLL lock-up phase begins. During this phase, SYSCLKOUT = OSCCLK/2. After the PLL lock-up is complete (which takes 2600 OSCCLK cycles), SYSCLKOUT reflects the new operating frequency, OSCCLK x 4.

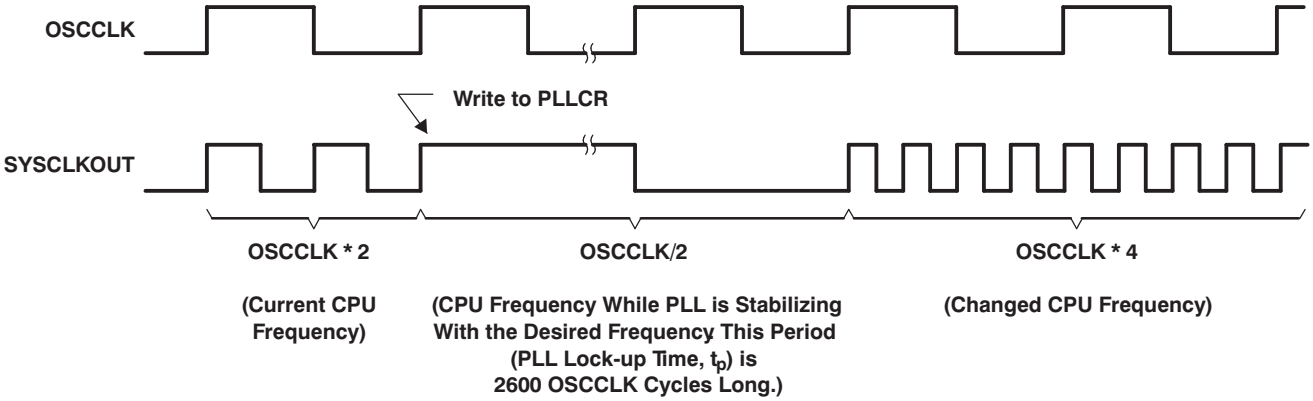



**Figure 6-7. Example of Effect of Writing Into PLLCR Register**

## 6.10 General-Purpose Input/Output (GPIO)

### *6.10.1 GPIO - Output Timing*

**Table 6-11. General-Purpose Output Switching Characteristics**

| PARAMETER | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{r(GPO)}$ | Rise time, GPIO switching low to high | All GPIOs | | 11 | ns |
| $t_{f(GPO)}$ | Fall time, GPIO switching high to low | All GPIOs | | 11 | ns |
| $t_{fGPO}$ | Toggling frequency, GPO pins | | | 40 | MHz |

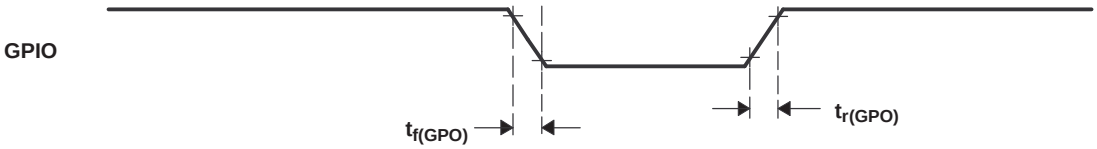
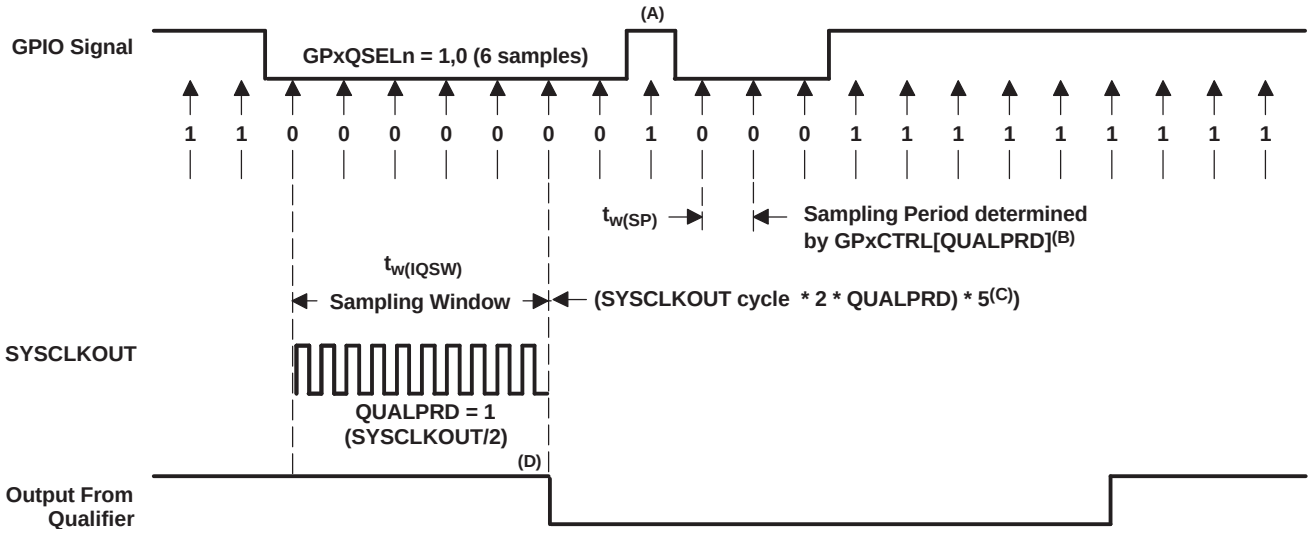



**Figure 6-8. General-Purpose Output Timing**

### 6.10.2 GPIO - Input Timing




A. This glitch will be ignored by the input qualifier. The QUALPRD bit field specifies the qualification sampling period. It can vary from 00 to 0xFF. If QUALPRD = 00, then the sampling period is 1 SYSCLKOUT cycle. For any other value "n", the qualification sampling period in 2n SYSCLKOUT cycles (i.e., at every 2n SYSCLKOUT cycles, the GPIO pin will be sampled).

B. The qualification period selected via the GPxCTRL register applies to groups of 8 GPIO pins.

C. The qualification block can take either three or six samples. The GPxQSELn Register selects which sample mode is used.

D. In the example shown, for the qualifier to detect the change, the input should be stable for 10 SYSCLKOUT cycles or greater. In other words, the inputs should be stable for (5 x QUALPRD x 2) SYSCLKOUT cycles. This would ensure 5 sampling periods for detection to occur. Since external signals are driven asynchronously, an 13-SYSCLKOUT-wide pulse ensures reliable recognition.

**Figure 6-9. Sampling Mode**

**Table 6-12. General-Purpose Input Timing Requirements**

| | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(SP)}$ | Sampling period | QUALPRD = 0 | $1t_{c(SCO)}$ | | cycles |
| | | QUALPRD ≠ 0 | $2t_{c(SCO)}$ * QUALPRD | | cycles |
| $t_{w(IQSW)}$ | Input qualifier sampling window | | $t_{w(SP)}$ * (n(1) – 1) | | cycles |
| $t_{w(GPI)}$ (2) | Pulse duration, GPIO low/high | Synchronous mode | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $t_{w(IQSW)} + t_{w(SP)} + 1t_{c(SCO)}$ | | cycles |

(1) "n" represents the number of qualification samples as defined by GPxQSELn register.
(2) For $t_{w(GPI)}$, pulse width is measured from $V_{IL}$ to $V_{IL}$ for an active low signal and $V_{IH}$ to $V_{IH}$ for an active high signal.

### 6.10.3 Sampling Window Width for Input Signals

The following section summarizes the sampling window width for input signals for various input qualifier configurations.

Sampling frequency denotes how often a signal is sampled with respect to SYSCLKOUT.

Sampling frequency = SYSCLKOUT/(2 * QUALPRD), if QUALPRD ≠ 0

Sampling frequency = SYSCLKOUT, if QUALPRD = 0

Sampling period = SYSCLKOUT cycle x 2 x QUALPRD, if QUALPRD ≠ 0

In the above equations, SYSCLKOUT cycle indicates the time period of SYSCLKOUT.

Sampling period = SYSCLKOUT cycle, if QUALPRD = 0

In a given sampling window, either 3 or 6 samples of the input signal are taken to determine the validity of the signal. This is determined by the value written to GPxQSELn register.

**Case 1:**

Qualification using 3 samples

Sampling window width = (SYSCLKOUT cycle x 2 x QUALPRD) x 2, if QUALPRD ≠ 0

Sampling window width = (SYSCLKOUT cycle) x 2, if QUALPRD = 0

**Case 2:**

Qualification using 6 samples

Sampling window width = (SYSCLKOUT cycle x 2 x QUALPRD) x 5, if QUALPRD ≠ 0

Sampling window width = (SYSCLKOUT cycle) x 5, if QUALPRD = 0

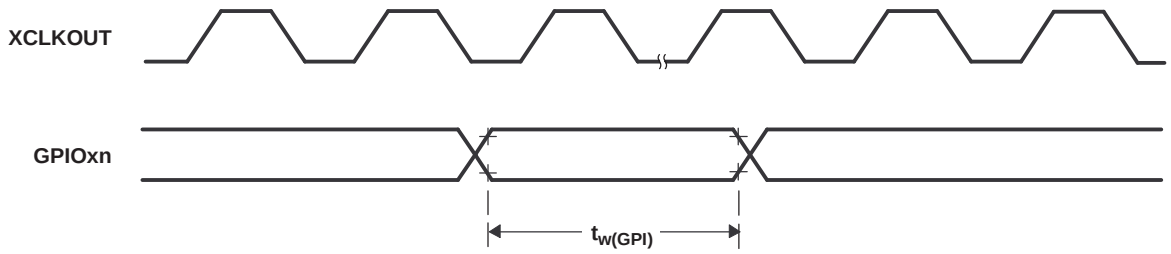



**Figure 6-10. General-Purpose Input Timing**

### 6.10.4 Low-Power Mode Wakeup Timing

The wakeup signal fed to a GPIO pin to wake up the device must meet the minimum pulse width requirement. Furthermore, this signal must be free of glitches. If a noisy signal is fed to a GPIO pin, the wakeup behavior of the device will not be deterministic and the device may not exit low-power mode for subsequent wakeup pulses.

Table 6-13 shows the timing requirements, Table 6-14 shows the switching characteristics, and Figure 6-11 shows the timing diagram for IDLE mode.

**Table 6-13. IDLE Mode Timing Requirements(1)**

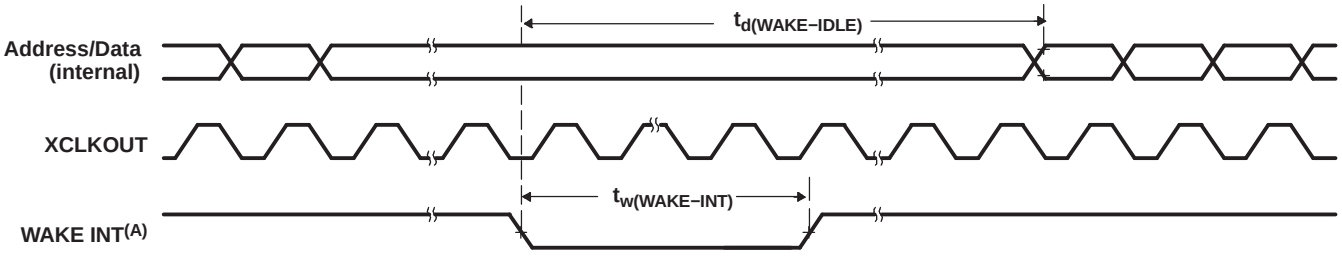
| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{w(WAKE-INT)}$ | Pulse duration, external wake-up signal | Without input qualifier | $2t_{c(SCO)}$ | | | cycles |
| | | With input qualifier | $5t_{c(SCO)} + t_{w(IQSW)}$ | | | |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

**Table 6-14. IDLE Mode Switching Characteristics(1)**

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{d(WAKE-IDLE)}$ | Delay time, external wake signal to program execution resume (2)<br>• Wake-up from SARAM | Without input qualifier | | | $20t_{c(SCO)}$ | cycles |
| | | With input qualifier | | | $20t_{c(SCO)} + t_{w(IQSW)}$ | |

(1) For an explanation of the input qualifier parameters, see Table 6-12.
(2) This is the time taken to begin execution of the instruction that immediately follows the IDLE instruction. execution of an ISR (triggered by the wake up) signal involves additional latency.




A. WAKE INT can be any enabled interrupt, $\overline{\text{WDINT}}$, XNMI, or $\overline{\text{XRS}}$.

**Figure 6-11. IDLE Entry and Exit Timing**

### Table 6-15. STANDBY Mode Timing Requirements

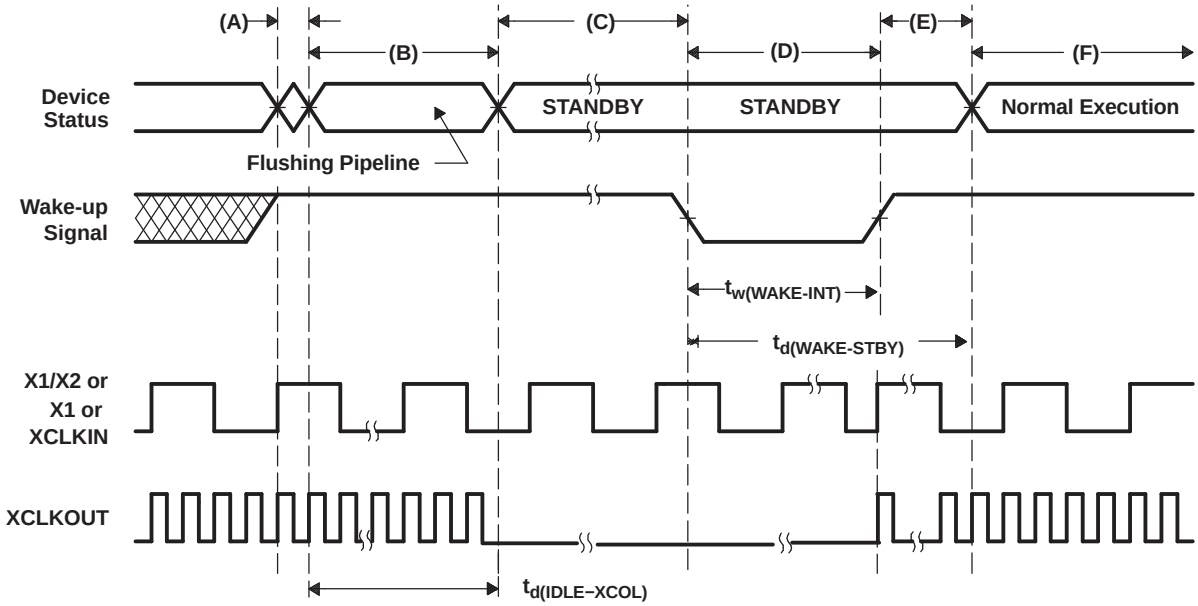
| | | TEST CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{w(WAKE-INT)}$ | Pulse duration, external wake-up signal | Without input qualification | $3t_{c(OSCCLK)}$ | | | cycles |
| | | With input qualification(1) | (2 + QUALSTDBY) * $t_{c(OSCCLK)}$ | | | |

(1) QUALSTDBY is a 6-bit field in the LPMCR0 register.

### Table 6-16. STANDBY Mode Switching Characteristics

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{d(IDLE-XCOL)}$ | Delay time, IDLE instruction executed to XCLKOUT low | | $32t_{c(SCO)}$ | | $45t_{c(SCO)}$ | cycles |
| $t_{d(WAKE-STBY)}$ | Delay time, external wake signal to program execution resume(1) | | | | | cycles |
| | • Wake up from SARAM | Without input qualifier | | | $100t_{c(SCO)}$ | cycles |
| | | With input qualifier | | | $100t_{c(SCO)} + t_{w(WAKE-INT)}$ | |

(1) This is the time taken to begin execution of the instruction that immediately follows the IDLE instruction. execution of an ISR (triggered by the wake up signal) involves additional latency.




A. IDLE instruction is executed to put the device into STANDBY mode.

B. The PLL block responds to the STANDBY signal. SYSCLKOUT is held for 32 cycles before being turned off. This delay enables the CPU pipeline and any other pending operations to flush properly. If an access to XINTF is in progress and its access time is longer than this number then it will fail. It is recommended to enter STANDBY mode from SARAM without an XINTF access in progress.

C. Clock to the peripherals are turned off. However, the PLL and watchdog are not shut down. The device is now in STANDBY mode.

D. The external wake-up signal is driven active.

E. After a latency period, the STANDBY mode is exited.

F. Normal execution resumes. The device will respond to the interrupt (if enabled).

### Figure 6-12. STANDBY Entry and Exit Timing Diagram

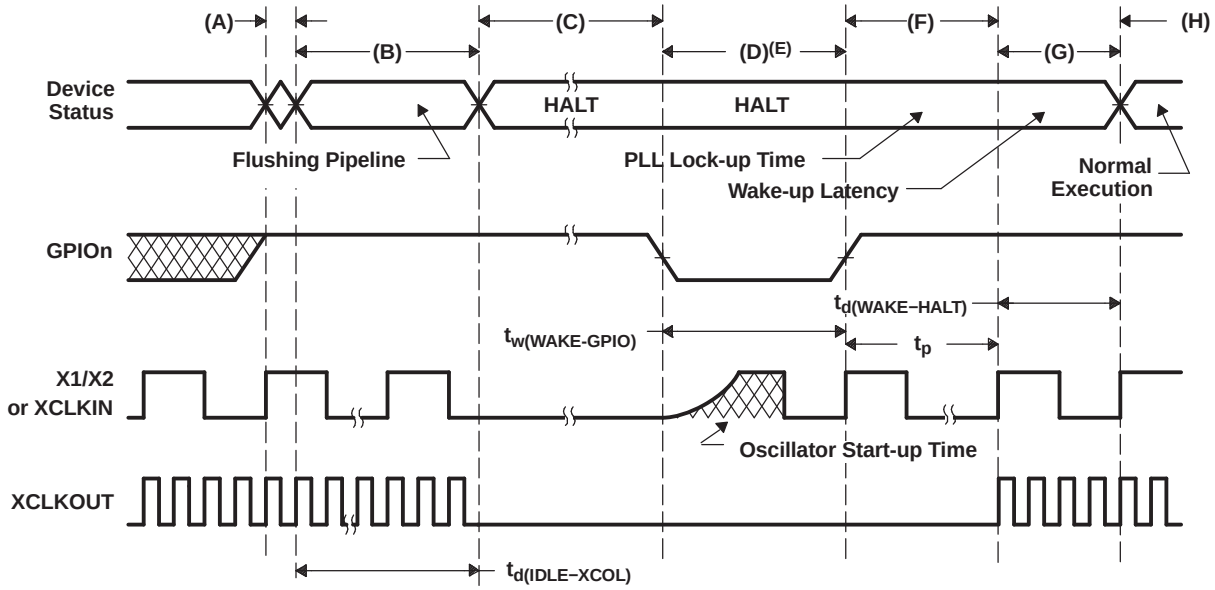
### Table 6-17. HALT Mode Timing Requirements

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(WAKE-GPIO)}$ | Pulse duration, GPIO wake-up signal | $t_{oscst} + 2t_{c(OSCCLK)}$ (1) | | | cycles |
| $t_{w(WAKE-XRS)}$ | Pulse duration, $\overline{XRS}$ wakeup signal | $t_{oscst} + 8t_{c(OSCCLK)}$ | | | cycles |

(1) See Table 6-10 for an explanation of $t_{oscst}$.

### Table 6-18. HALT Mode Switching Characteristics

| PARAMETER | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{d(IDLE-XCOL)}$ | Delay time, IDLE instruction executed to XCLKOUT low | $32t_{c(SCO)}$ | | $45t_{c(SCO)}$ | cycles |
| $t_p$ | PLL lock-up time | | | $2600t_{c(OSCCLK)}$ | cycles |
| $t_{d(WAKE-HALT)}$ | Delay time, PLL lock to program execution resume<br>• Wake up from SARAM | | | $35t_{c(SCO)}$ | cycles |




A. IDLE instruction is executed to put the device into HALT mode.

B. The PLL block responds to the HALT signal. SYSCLKOUT is held for 32 cycles before oscillator is turned off and the CLKIN to the core is stopped. This delay enables the CPU pipeline and any other pending operations to flush properly. If an access to XINTF is in progress and its access time is longer than this number then it will fail. It is recommended to enter HALT mode from SARAM without an XINTF access in progress.

C. Clocks to the peripherals are turned off and the PLL is shut down. If a quartz crystal or ceramic resonator is used as the clock source, the internal oscillator is shut down as well. The device is now in HALT mode and consumes absolute minimum power.

D. When the GPIOn pin (used to bring the device out of HALT) is driven low, the oscillator is turned on and the oscillator wake-up sequence is initiated. The GPIO pin should be driven high only after the oscillator has stabilized. This enables the provision of a clean clock signal during the PLL lock sequence. Since the falling edge of the GPIO pin asynchronously begins the wakeup process, care should be taken to maintain a low noise environment prior to entering and during HALT mode.

E. The wake-up signal fed to a GPIO pin to wake up the device must meet the minimum pulse width requirement. Furthermore, this signal must be free of glitches. If a noisy signal is fed to a GPIO pin, the wake-up behavior of the device will not be deterministic and the device may not exit low-power mode for subsequent wake-up pulses.

F. Once the oscillator has stabilized, the PLL lock sequence is initiated, which takes 2,600 OSCCLK (X1/X2 or X1 or XCLKIN) cycles.

G. Clocks to the core and peripherals are enabled. The HALT mode is now exited. The device will respond to the interrupt (if enabled), after a latency.

H. Normal operation resumes.

### Figure 6-13. HALT Wake-Up Using GPIOn

## 6.11 Enhanced Control Peripherals

### 6.11.1 Enhanced Pulse Width Modulator (ePWM) Timing

PWM refers to PWM outputs on ePWM1–6. Table 6-19 shows the PWM timing requirements and Table 6-20, switching characteristics.

**Table 6-19. ePWM Timing Requirements(1)**

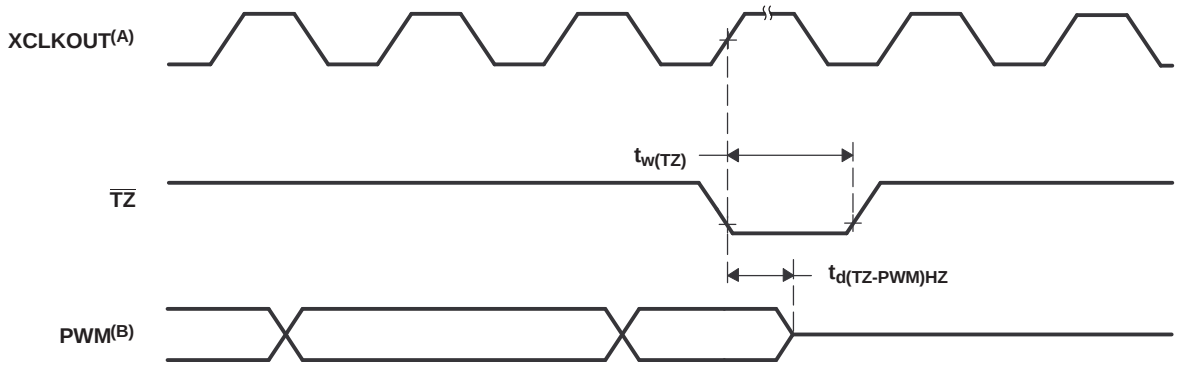
| | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(SYCIN)}$ | Sync input pulse width | Asynchronous | $2t_{c(SCO)}$ | | cycles |
| | | Synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $1t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

**Table 6-20. ePWM Switching Characteristics**

| PARAMETER | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(PWM)}$ | Pulse duration, PWMx output high/low | | 20 | | ns |
| $t_{w(SYNCOUT)}$ | Sync output pulse width | | $8t_{c(SCO)}$ | | cycles |
| $t_{d(PWM)tza}$ | Delay time, trip input active to PWM forced high<br>Delay time, trip input active to PWM forced low | no pin load | | 25 | ns |
| $t_{d(TZ-PWM)HZ}$ | Delay time, trip input active to PWM Hi-Z | | | 20 | ns |

### 6.11.2 Trip-Zone Input Timing




A. $\overline{TZ}$ - $\overline{TZ1}$, $\overline{TZ2}$, $\overline{TZ3}$, $\overline{TZ4}$, $\overline{TZ5}$, $\overline{TZ6}$

B. PWM refers to all the PWM pins in the device. The state of the PWM pins after $\overline{TZ}$ is taken high depends on the PWM recovery software.

**Figure 6-14. PWM Hi-Z Characteristics**

**Table 6-21. Trip-Zone Input Timing Requirements(1)**

| | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(TZ)}$ | Pulse duration, $\overline{TZx}$ input low | Asynchronous | $1t_{c(SCO)}$ | | cycles |
| | | Synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $1t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

Table 6-22 shows the high-resolution PWM switching characteristics.

**Table 6-22. High-Resolution PWM Characteristics at SYSCLKOUT = ( 150– 300 MHz)**

| | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| Micro Edge Positioning (MEP) step size(1) | $V_{DD}$ = 1.2 V | | 55 | 120 | ps |
| | $V_{DD}$ = 1.1 V | | 65 | 140 | ps |

(1) Maximum MEP step size is based on worst-case process, maximum temperature and maximum voltage. MEP step size will increase with low voltage and high temperature and decrease with voltage and cold temperature.
Applications that use the HRPWM feature should use MEP Scale Factor Optimizer (SFO) estimation software functions. See the TI software libraries for details of using SFO function in end applications. SFO functions help to estimate the number of MEP steps per SYSCLKOUT period dynamically while the HRPWM is in operation.

### 6.11.3 Enhanced Capture (eCAP) Timing

Table 6-23 shows the eCAP timing requirement and Table 6-24 shows the eCAP switching characteristics.

**Table 6-23. Enhanced Capture (eCAP) Timing Requirement(1)**

| | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(CAP)}$ | Capture input pulse width | Asynchronous | $2t_{c(SCO)}$ | | cycles |
| | | Synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $1t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

**Table 6-24. eCAP Switching Characteristics**

| PARAMETER | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(APWM)}$ | Pulse duration, APWMx output high/low | | 20 | | ns |

### 6.11.4 Enhanced Quadrature Encoder Pulse (eQEP) Timing

Table 6-25 shows the eQEP timing requirement and Table 6-26 shows the eQEP switching characteristics.

**Table 6-25. Enhanced Quadrature Encoder Pulse (eQEP) Timing Requirements(1)**

| | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(QEPP)}$ | QEP input period | Asynchronous/synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $2[1t_{c(SCO)} + t_{w(IQSW)}]$ | | cycles |
| $t_{w(INDEXH)}$ | QEP Index Input High time | Asynchronous/synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $2t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |
| $t_{w(INDEXL)}$ | QEP Index Input Low time | Asynchronous/synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $2t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |
| $t_{w(STROBH)}$ | QEP Strobe High time | Asynchronous/synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $2t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |
| $t_{w(STROBL)}$ | QEP Strobe Input Low time | Asynchronous/synchronous | $2t_{c(SCO)}$ | | cycles |
| | | With input qualifier | $2t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

**Table 6-26. eQEP Switching Characteristics**

| PARAMETER | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{d(CNTR)xin}$ | Delay time, external clock to counter increment | | | $4t_{c(SCO)}$ | cycles |
| $t_{d(PCS-OUT)QEP}$ | Delay time, QEP input edge to position compare sync output | | | $6t_{c(SCO)}$ | cycles |

### 6.11.5 ADC Start-of-Conversion Timing

**Table 6-27. External ADC Start-of-Conversion Switching Characteristics**

| PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{w(ADCSOCL)}$ | Pulse duration, $\overline{ADCSOCxO}$ low | $32t_{c(HCO)}$ | | cycles |

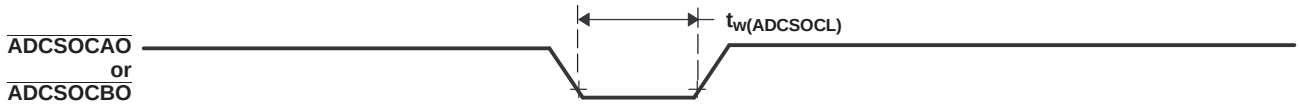



**Figure 6-15. $\overline{ADCSOCAO}$ or $\overline{ADCSOCBO}$ Timing**

## 6.12 External Interrupt Timing

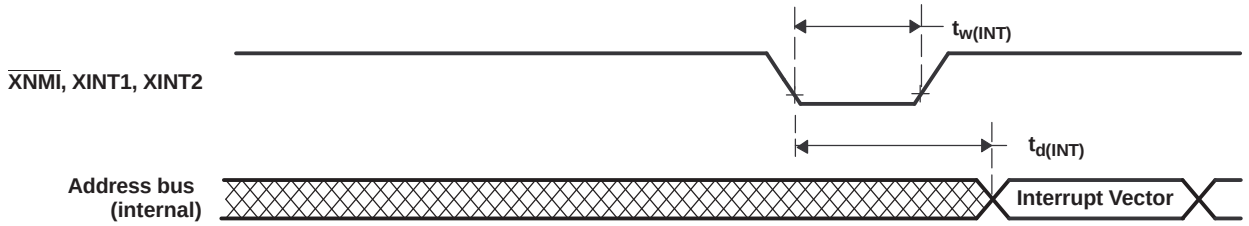



**Figure 6-16. External Interrupt Timing**

**Table 6-28. External Interrupt Timing Requirements(1)**

| | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{w(INT)}$ (2) | Pulse duration, INT input low/high | Synchronous | $1t_{c(SCO)}$ | | cycles |
| | | With qualifier | $1t_{c(SCO)} + t_{w(IQSW)}$ | | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.
(2) This timing is applicable to any GPIO pin configured for ADCSOC functionality.

**Table 6-29. External Interrupt Switching Characteristics(1)**

| PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(INT)}$ | Delay time, INT low/high to interrupt-vector fetch | | $t_{w(IQSW)} + 12t_{c(SCO)}$ | cycles |

(1) For an explanation of the input qualifier parameters, see Table 6-12.

## 6.13 I2C Electrical Specification and Timing

**Table 6-30. I2C Timing**

| | | TEST CONDITIONS | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $f_{SCL}$ | SCL clock frequency | I2C clock module frequency is between 7 MHz and 12 MHz and I2C prescaler and clock divider registers are configured appropriately | | 400 | kHz |
| $v_{il}$ | Low level input voltage | | | 0.3 $V_{DDIO}$ | V |
| $V_{ih}$ | High level input voltage | | 0.7 $V_{DDIO}$ | | V |
| $V_{hys}$ | Input hysteresis | | 0.05 $V_{DDIO}$ | | V |
| $V_{ol}$ | Low level output voltage | 3-mA sink current | 0 | 0.4 | V |
| $t_{LOW}$ | Low period of SCL clock | I2C clock module frequency is between 7 MHz and 12 MHz and I2C prescaler and clock divider registers are configured appropriately | 1.3 | | μs |
| $t_{HIGH}$ | High period of SCL clock | I2C clock module frequency is between 7 MHz and 12 MHz and I2C prescaler and clock divider registers are configured appropriately | 0.6 | | μs |
| $I_I$ | Input current with an input voltage between 0.1 $V_{DDIO}$ and 0.9 $V_{DDIO}$ MAX | | –10 | 10 | μA |

## 6.14 Serial Peripheral Interface (SPI) Timing

This section contains both Master Mode and Slave Mode timing data.

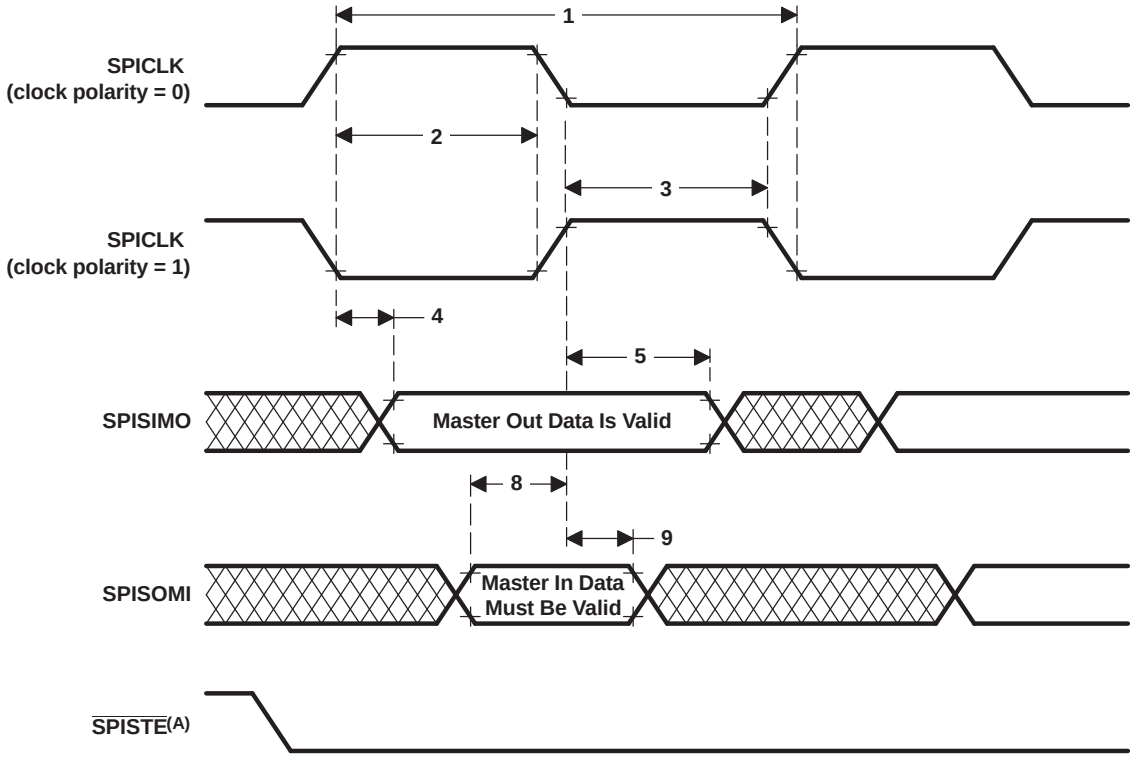
### *6.14.1 Master Mode Timing*

Table 6-31 lists the master mode timing (clock phase = 0) and Table 6-32 lists the timing (clock phase = 1). Figure 6-17 and Figure 6-18 show the timing waveforms.

**Table 6-31. SPI Master Mode External Timing (Clock Phase = 0)(1) (2) (3) (4) (5)**

| NO. | | | SPI WHEN (SPIBRR + 1) IS EVEN OR SPIBRR = 0 OR 2 | | SPI WHEN (SPIBRR + 1) IS ODD AND SPIBRR > 3 | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(SPC)M}$ | Cycle time, SPICLK | $4t_{c(LCO)}$ | $128t_{c(LCO)}$ | $5t_{c(LCO)}$ | $127t_{c(LCO)}$ | ns |
| 2 | $t_{w(SPCH)M}$ | Pulse duration, SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)}$ | ns |
| | $t_{w(SPCL)M}$ | Pulse duration, SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)}$ | |
| 3 | $t_{w(SPCL)M}$ | Pulse duration, SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | $0.5_{tc(SPC)M}$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)}$ | ns |
| | $t_{w(SPCH)M}$ | Pulse duration, SPICLK high (clock polarity = 1) | $0.5_{tc(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)}$ | |
| 4 | $t_{d(SPCH-SIMO)M}$ | Delay time, SPICLK high to SPISIMO valid (clock polarity = 0) | | 10 | | 10 | ns |
| | $t_{d(SPCL-SIMO)M}$ | Delay time, SPICLK low to SPISIMO valid (clock polarity = 1) | | 10 | | 10 | |
| 5 | $t_{v(SPCL-SIMO)M}$ | Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)} - 10$ | | ns |
| | $t_{v(SPCH-SIMO)M}$ | Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)} - 10$ | | |
| 8 | $t_{su(SOMI-SPCL)M}$ | Setup time, SPISOMI before SPICLK low (clock polarity = 0) | 20 | | 20 | | ns |
| | $t_{su(SOMI-SPCH)M}$ | Setup time, SPISOMI before SPICLK high (clock polarity = 1) | 20 | | 20 | | |
| 9 | $t_{v(SPCL-SOMI)M}$ | Valid time, SPISOMI data valid after SPICLK low (clock polarity = 0) | $0.25t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | | ns |
| | $t_{v(SPCH-SOMI)M}$ | Valid time, SPISOMI data valid after SPICLK high (clock polarity = 1) | $0.25t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | | |

(1) The MASTER / SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is cleared.
(2) $t_{c(SPC)}$ = SPI clock cycle time = LSPCLK/4 or LSPCLK/(SPIBRR +1)
(3) $t_{c(LCO)}$ = LSPCLK cycle time
(4) Internal clock prescalers must be adjusted such that the SPI clock speed is limited to the following SPI clock rate:
Master mode transmit 25-MHz MAX, master mode receive 12.5-MHz MAX
Slave mode transmit 20-MHz MAX, slave mode receive 20-MHz MAX.
(5) The active edge of the SPICLK signal referenced is controlled by the clock polarity bit (SPICCR.6).



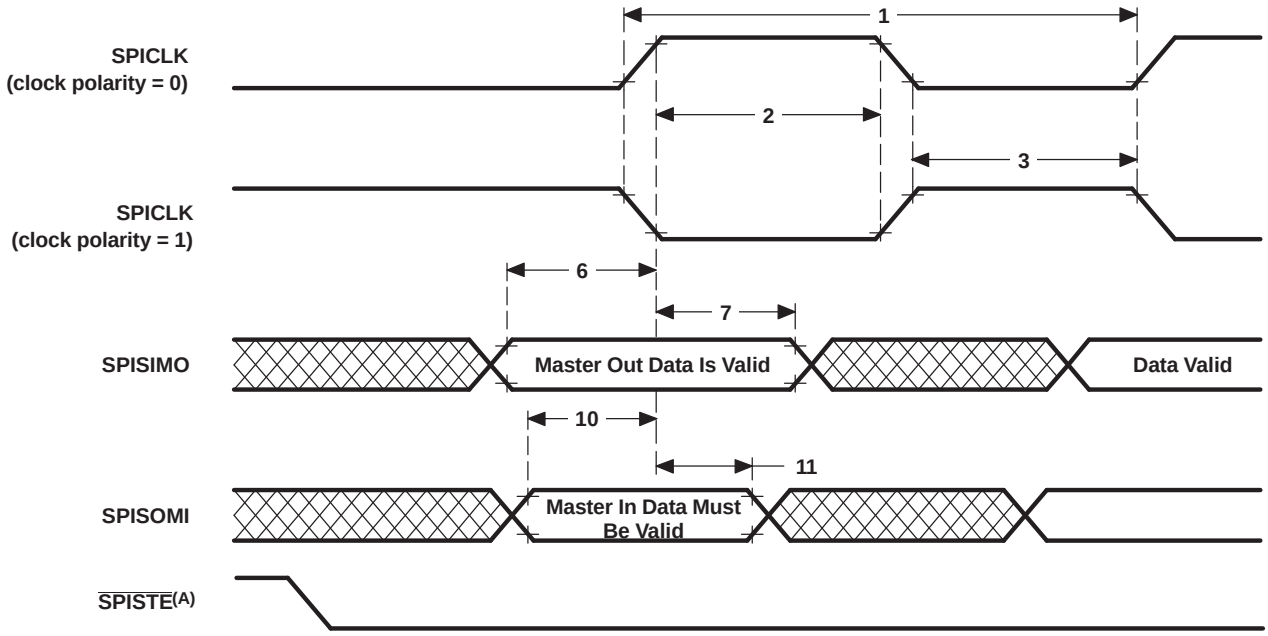

A. In the master mode, SPISTE goes active 1 $t_{c(SPC)}$ (minimum) before valid SPI clock edge. On the trailing end of the word, the SPISTE will go inactive 1 $t_{c(SPC)}$ after the receiving edge (SPICLK) of the last data bit, except that SPISTE stays active between back-to-back transmit words in both FIFO and non-FIFO modes.

**Figure 6-17. SPI Master Mode External Timing (Clock Phase = 0)**

**Table 6-32. SPI Master Mode External Timing (Clock Phase = 1)(1) (2) (3) (4) (5)**

| NO. | | | SPI WHEN (SPIBRR + 1) IS EVEN OR SPIBRR = 0 OR 2 | | SPI WHEN (SPIBRR + 1) IS ODD AND SPIBRR > 3 | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(SPC)M}$ | Cycle time, SPICLK | $4t_{c(LCO)}$ | $128t_{c(LCO)}$ | $5t_{c(LCO)}$ | $127t_{c(LCO)}$ | ns |
| 2 | $t_{w(SPCH)M}$ | Pulse duration, SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)}$ | ns |
| | $t_{w(SPCL))M}$ | Pulse duration, SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LCO)}$ | |
| 3 | $t_{w(SPCL)M}$ | Pulse duration, SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)}$ | ns |
| | $t_{w(SPCH)M}$ | Pulse duration, SPICLK high (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M}$ | $0.5_{tc(SPC)M} + 0.5t_{c(LCO)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LCO)}$ | |
| 6 | $t_{su(SIMO-SPCH)M}$ | Setup time, SPISIMO data valid before SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 10$ | | ns |
| | $t_{su(SIMO-SPCL)M}$ | Setup time, SPISIMO data valid before SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 10$ | | |
| 7 | $t_{v(SPCH-SIMO)M}$ | Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 10$ | | ns |
| | $t_{v(SPCL-SIMO)M}$ | Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 10$ | | |
| 10 | $t_{su(SOMI-SPCH)M}$ | Setup time, SPISOMI before SPICLK high (clock polarity = 0) | 20 | | 20 | | ns |
| | $t_{su(SOMI-SPCL)M}$ | Setup time, SPISOMI before SPICLK low (clock polarity = 1) | 20 | | 20 | | |
| 11 | $t_{v(SPCH-SOMI)M}$ | Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0) | $0.25t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 10$ | | ns |
| | $t_{v(SPCL-SOMI)M}$ | Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1) | $0.25_{tc(SPC)M} - 10$ | | $0.5_{tc(SPC)M} - 10$ | | |

(1) The MASTER/SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is set.
(2) $t_{c(SPC)}$ = SPI clock cycle time = LSPCLK/4 or LSPCLK/(SPIBRR + 1)
(3) Internal clock prescalers must be adjusted such that the SPI clock speed is limited to the following SPI clock rate:
Master mode transmit 25-MHz MAX, master mode receive 12.5 MHz MAX
Slave mode transmit 20-MHz MAX, slave mode receive 20 MHz MAX.
(4) $t_{c(LCO)}$ = LSPCLK cycle time
(5) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).




B. In the master mode, $\overline{\text{SPISTE}}$ goes active 1 $t_{c(SPC)}$ (minimum) before valid SPI clock edge. On the trailing end of the word, the $\overline{\text{SPISTE}}$ will go inactive 1 $t_{c(SPC)}$ after the receiving edge (SPICLK) of the last data bit, except that $\overline{\text{SPISTE}}$ stays active between back-to-back transmit words in both FIFO and non-FIFO modes.

**Figure 6-18. SPI Master Mode External Timing (Clock Phase = 1)**

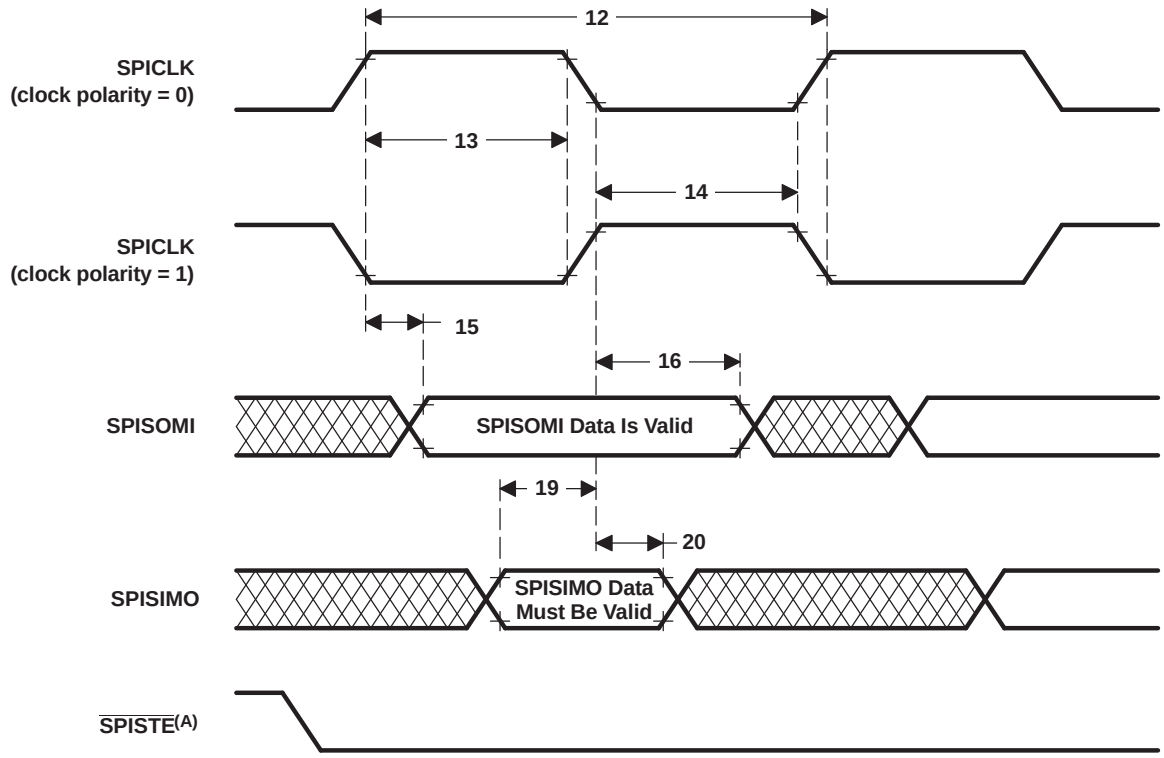
### 6.14.2 SPI Slave Mode Timing

Table 6-33 lists the slave mode external timing (clock phase = 0) and Table 6-34 (clock phase = 1). Figure 6-19 and Figure 6-20 show the timing waveforms.

**Table 6-33. SPI Slave Mode External Timing (Clock Phase = 0)(1) (2) (3) (4) (5)**

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 12 | $t_{c(SPC)S}$ | Cycle time, SPICLK | $4t_{c(LCO)}$ | | ns |
| 13 | $t_{w(SPCH)S}$ | Pulse duration, SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | ns |
| | $t_{w(SPCL)S}$ | Pulse duration, SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | |
| 14 | $t_{w(SPCL)S}$ | Pulse duration, SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | ns |
| | $t_{w(SPCH)S}$ | Pulse duration, SPICLK high (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | |
| 15 | $t_{d(SPCH-SOMI)S}$ | Delay time, SPICLK high to SPISOMI valid (clock polarity = 0) | | 20 | ns |
| | $t_{d(SPCL-SOMI)S}$ | Delay time, SPICLK low to SPISOMI valid (clock polarity = 1) | | 20 | |
| 16 | $t_{v(SPCL-SOMI)S}$ | Valid time, SPISOMI data valid after SPICLK low (clock polarity = 0) | $0.75t_{c(SPC)S}$ | | ns |
| | $t_{v(SPCH-SOMI)S}$ | Valid time, SPISOMI data valid after SPICLK high (clock polarity = 1) | $0.75t_{c(SPC)S}$ | | |
| 19 | $t_{su(SIMO-SPCL)S}$ | Setup time, SPISIMO before SPICLK low (clock polarity = 0) | 20 | | ns |
| | $t_{su(SIMO-SPCH)S}$ | Setup time, SPISIMO before SPICLK high (clock polarity = 1) | 20 | | |
| 20 | $t_{v(SPCL-SIMO)S}$ | Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | | ns |
| | $t_{v(SPCH-SIMO)S}$ | Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | | |

(1) The MASTER / SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is cleared.
(2) $t_{c(SPC)}$ = SPI clock cycle time = LSPCLK/4 or LSPCLK/(SPIBRR + 1)
(3) Internal clock prescalers must be adjusted such that the SPI clock speed is limited to the following SPI clock rate:
Master mode transmit 25-MHz MAX, master mode receive 12.5-MHz MAX
Slave mode transmit 20-MHz MAX, slave mode receive 20-MHz MAX.
(4) $t_{c(LCO)}$ = LSPCLK cycle time
(5) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).



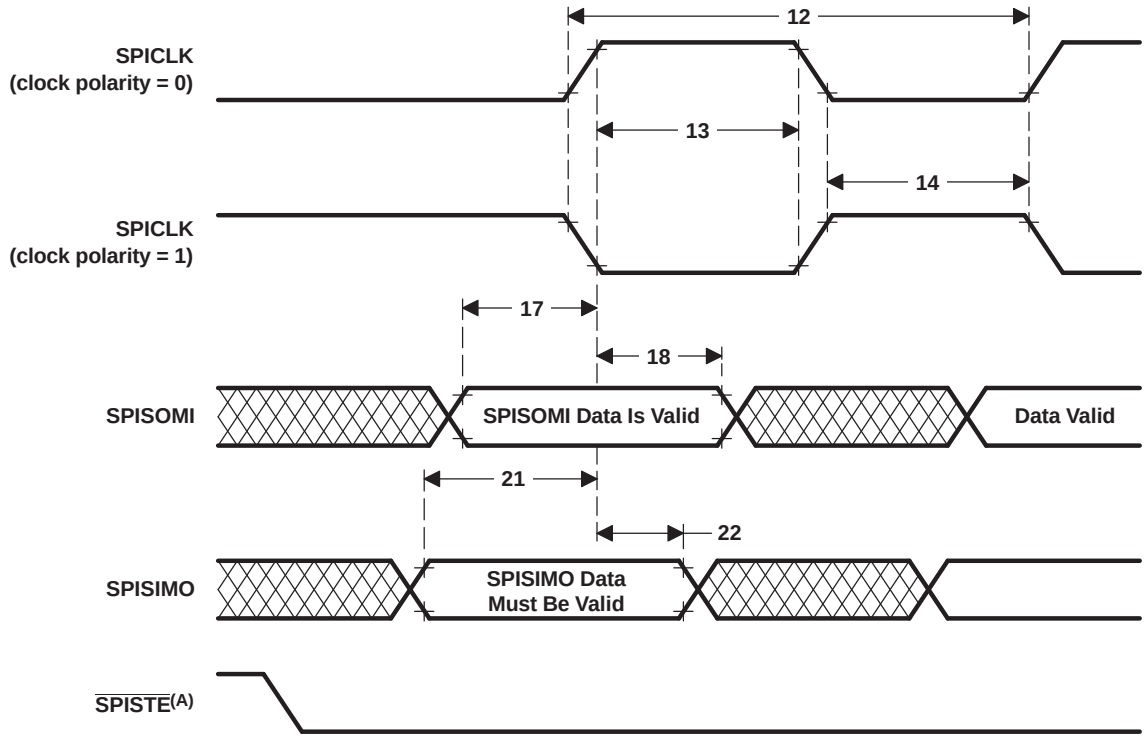

C. In the slave mode, the $\overline{\text{SPISTE}}$ signal should be asserted low at least 1 $t_{c(SPC)}$ (minimum) before the valid SPI clock edge and remain low for at least 1 $t_{c(SPC)}$ after the receiving edge (SPICLK) of the last data bit.

**Figure 6-19. SPI Slave Mode External Timing (Clock Phase = 0)**

**Table 6-34. SPI Slave Mode External Timing (Clock Phase = 1)(1) (2) (3) (4)**

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 12 | $t_{c(SPC)S}$ | Cycle time, SPICLK | $8t_{c(LCO)}$ | | ns |
| 13 | $t_{w(SPCH)S}$ | Pulse duration, SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | ns |
| | $t_{w(SPCL)S}$ | Pulse duration, SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | |
| 14 | $t_{w(SPCL)S}$ | Pulse duration, SPICLK low (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | ns |
| | $t_{w(SPCH)S}$ | Pulse duration, SPICLK high (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | $0.5t_{c(SPC)S}$ | |
| 17 | $t_{su(SOMI-SPCH)S}$ | Setup time, SPISOMI before SPICLK high (clock polarity = 0) | $0.125t_{c(SPC)S}$ | | ns |
| | $t_{su(SOMI-SPCL)S}$ | Setup time, SPISOMI before SPICLK low (clock polarity = 1) | $0.125t_{c(SPC)S}$ | | |
| 18 | $t_{v(SPCL-SOMI)S}$ | Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1) | $0.75t_{c(SPC)S}$ | | ns |
| | $t_{v(SPCH-SOMI)S}$ | Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0) | $0.75t_{c(SPC)S}$ | | |
| 21 | $t_{su(SIMO-SPCH)S}$ | Setup time, SPISIMO before SPICLK high (clock polarity = 0) | 20 | | ns |
| | $t_{su(SIMO-SPCL)S}$ | Setup time, SPISIMO before SPICLK low (clock polarity = 1) | 20 | | |
| 22 | $t_{v(SPCH-SIMO)S}$ | Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0) | $0.5t_{c(SPC)S} - 10$ | | ns |
| | $t_{v(SPCL-SIMO)S}$ | Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1) | $0.5t_{c(SPC)S} - 10$ | | |

(1) The MASTER / SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is cleared.
(2) $t_{c(SPC)}$ = SPI clock cycle time = LSPCLK/4 or LSPCLK/(SPIBRR + 1)
(3) Internal clock prescalers must be adjusted such that the SPI clock speed is limited to the following SPI clock rate:
Master mode transmit 25-MHz MAX, master mode receive 12.5-MHz MAX
Slave mode transmit 20-MHz MAX, slave mode receive 20-MHz MAX.
(4) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).




A. In the slave mode, the $\overline{\text{SPISTE}}$ signal should be asserted low at least 1 $t_{c(SPC)}$ before the valid SPI clock edge and remain low for at least 1 $t_{c(SPC)}$ after the receiving edge (SPICLK) of the last data bit.

**Figure 6-20. SPI Slave Mode External Timing (Clock Phase = 1)**

## 6.15 External Interface (XINTF) Timing

Each XINTF access consists of three parts: Lead, Active, and Trail. The user configures the Lead/Active/Trail wait states in the XTIMING registers. There is one XTIMING register for each XINTF zone. Table 6-35 shows the relationship between the parameters configured in the XTIMING register and the duration of the pulse in terms of XTIMCLK cycles.

**Table 6-35. Relationship Between Parameters Configured in XTIMING and Duration of Pulse**

| DESCRIPTION | | DURATION (ns)(1) (2) | |
|---|---|---|---|
| | | X2TIMING = 0 | X2TIMING = 1 |
| LR | Lead period, read access | XRDLEAD × $t_{c(XTIM)}$ | (XRDLEAD × 2) × $t_{c(XTIM)}$ |
| AR | Active period, read access | (XRDACTIVE + WS + 1) × $t_{c(XTIM)}$ | (XRDACTIVE × 2 + WS + 1) × $t_{c(XTIM)}$ |
| TR | Trail period, read access | XRDTRAIL × $t_{c(XTIM)}$ | (XRDTRAIL × 2) × $t_{c(XTIM)}$ |
| LW | Lead period, write access | XWRLEAD × $t_{c(XTIM)}$ | (XWRLEAD × 2) × $t_{c(XTIM)}$ |
| AW | Active period, write access | (XWRACTIVE + WS + 1) × $t_{c(XTIM)}$ | (XWRACTIVE × 2 + WS + 1) × $t_{c(XTIM)}$ |
| TW | Trail period, write access | XWRTRAIL × $t_{c(XTIM)}$ | (XWRTRAIL × 2) × $t_{c(XTIM)}$ |

(1) $t_{c(XTIM)}$ − Cycle time, XTIMCLK
(2) WS refers to the number of wait states inserted by hardware when using XREADY. If the zone is configured to ignore XREADY (USEREADY = 0), then WS = 0.

Minimum wait state requirements must be met when configuring each zone's XTIMING register. These requirements are in addition to any timing requirements as specified by that device's data sheet. No internal device hardware is included to detect illegal settings.

### *6.15.1 USEREADY = 0*

If the XREADY signal is ignored (USEREADY = 0), then:

Lead: $LR \geq 2 \times t_{c(XTIM)}$
$LW \geq 3 \times t_{c(XTIM)}$
Active: $AR \geq 6 \times t_{c(XTIM)}$
$AW \geq 1 \times t_{c(XTIM)}$
Trail: $TW \geq 3 \times t_{c(XTIM)}$

These requirements result in the following XTIMING register configuration restrictions:

| XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|
| ≥ 2 | ≥ 5 | ≥ 0 | ≥ 3(1) | ≥ 1 | ≥ 3(1) | 0(2) |

(1) Lead and trail write must be at least 7.5 ns.
(2) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

Examples of valid and invalid timing when not sampling XREADY:

| | XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|---|
| Invalid(1) | 0 | 0 | 0 | 0 | 0 | 0 | 0, 1 |
| Valid(2) | 2 | 5 | 0 | 3 | 1 | 3 | 0(3) |

(1) No hardware to detect illegal XTIMING configurations
(2) Based on 300-MHz system clock speed.
(3) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

### 6.15.2 Synchronous Mode (USEREADY = 1, READYMODE = 0)

If the XREADY signal is sampled in the synchronous mode (USEREADY = 1, READYMODE = 0), then:

1 Lead: $LR \geq 2 \times t_{c(XTIM)}$
$LW \geq 3 \times t_{c(XTIM)}$
2 Active: $AR \geq 6 \times t_{c(XTIM)}$
$AW \geq 2 \times t_{c(XTIM)}$
3 Trail: $TW \geq 3 \times t_{c(XTIM)}$

**NOTE**
Restriction does not include external hardware wait states.

These requirements result in the following XTIMING register configuration restrictions (based on 300-MHz system clock speed):

| XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|
| ≥ 2 | ≥ 5 | ≥ 0 | ≥ 3(1) | ≥ 1 | ≥ 3(1) | 0(2) |

(1) Lead and trail write must be at least 7.5 ns.
(2) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

Examples of valid and invalid timing when using synchronous XREADY:

| | XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|---|
| Invalid(1) | 0 | 0 | 0 | 0 | 0 | 0 | 0, 1 |
| Invalid(1) | 1 | 0 | 0 | 1 | 0 | 0 | 0, 1 |
| Valid(2) | 2 | 5 | 0 | 3 | 1 | 3 | 0(3) |

(1) No hardware to detect illegal XTIMING configurations
(2) Based on 300-MHz system clock speed
(3) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

### 6.15.3 Asynchronous Mode (USEREADY = 1, READYMODE = 1)

If the XREADY signal is sampled in the asynchronous mode (USEREADY = 1, READYMODE = 1), then:

1 Lead: $LR \geq 2 \times t_{c(XTIM)}$

$LW \geq 3 \times t_{c(XTIM)}$

2 Active: $AR \geq 6 \times t_{c(XTIM)}$

$AW \geq 4 \times t_{c(XTIM)}$

3 Trail: $TW \geq 3 \times t_{c(XTIM)}$

**NOTE**

Restrictions do not include external hardware wait states.

These requirements result in the following XTIMING register configuration restrictions (based on 300-MHz system clock speed):

| XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|
| ≥ 2 | ≥ 5 | 0 | ≥ 3(1) | ≥ 3 | 0(1) | 0(2) |

(1) Lead and trail write must be at least 7.5 ns.
(2) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

Examples of valid and invalid timing when using asynchronous XREADY:

| | XRDLEAD | XRDACTIVE | XRDTRAIL | XWRLEAD | XWRACTIVE | XWRTRAIL | X2TIMING |
|---|---|---|---|---|---|---|---|
| Invalid(1) | 0 | 0 | 0 | 0 | 0 | 0 | 0, 1 |
| Invalid(1) | 1 | 0 | 0 | 1 | 0 | 0 | 0, 1 |
| Invalid(1) | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| Valid(2) | 2 | 5 | 0 | 3 | 3 | 3 | 0(3) |

(1) No hardware to detect illegal XTIMING configurations
(2) Based on 300-MHz system clock speed
(3) If X2TIMCLK is enabled, specified Lead, Active, and Trail restrictions can be divided by 2 for values with even numbers.

Unless otherwise specified, all XINTF timing is applicable for the clock configurations shown in Table 6-36.

**Table 6-36. XINTF Clock Configurations for SYSCLKOUT = 300 MHz**

| MODE | SYSCLKOUT | XTIMCLK | XCLKOUT (1) |
|---|---|---|---|
| 1 | | SYSCLKOUT | SYSCLKOUT |
| Example: | 300 MHz | 300 MHz | 300 MHz |
| 2 | | SYSCLKOUT | 1/2 SYSCLKOUT |
| Example: | 300 MHz | 300 MHz | 150 MHz |
| 3 | | SYSCLKOUT | 1/2 SYSCLKOUT |
| Example: | 300 MHz | 300 MHz | 150 MHz |
| 4 | | SYSCLKOUT | 1/4 SYSCLKOUT |
| Example: | 300 MHz | 300 MHz | 75 MHz |
| 5 | | 1/2 SYSCLKOUT | 1/2 SYSCLKOUT |
| Example: | 300 MHz | 150 MHz | 150 MHz |
| 6 | | 1/2 SYSCLKOUT | 1/4 SYSCLKOUT |
| Example: | 300 MHz | 150 MHz | 75 MHz |
| 7 | | 1/2 SYSCLKOUT | 1/4 SYSCLKOUT |
| Example: | 300 MHz | 150 MHz | 75 MHz |
| 8 | | 1/2 SYSCLKOUT | 1/8 SYSCLKOUT |
| Example: | 300 MHz | 150 MHz | 37.5 MHz |

(1) The XCLKOUT signal is limited to a maximum frequency of 75 MHz.

The relationship between SYSCLKOUT and XTIMCLK is shown in Figure 6-21.

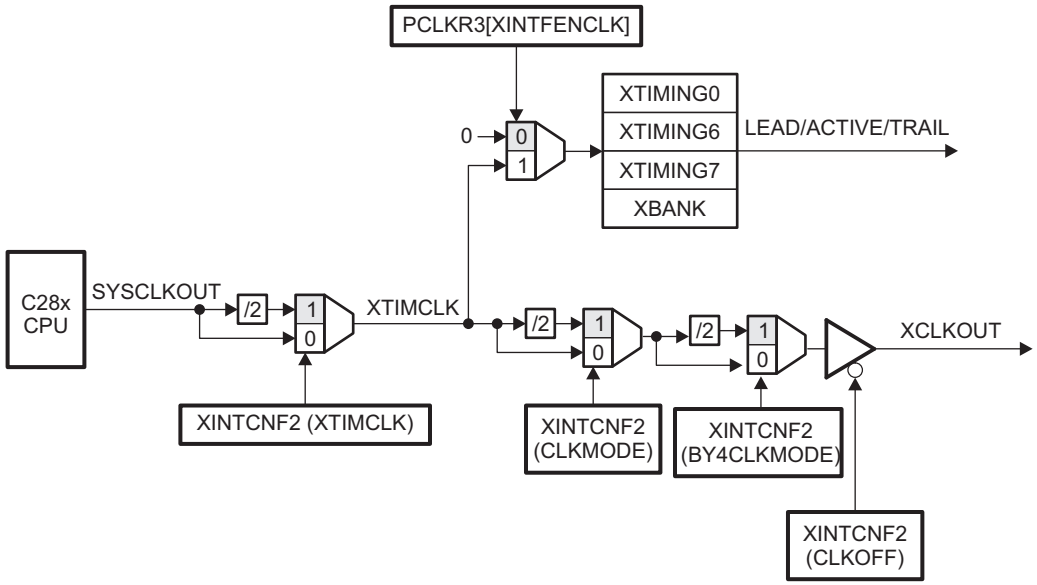



**Figure 6-21. Relationship Between XTIMCLK and SYSCLKOUT**

### 6.15.4 XINTF Signal Alignment to XCLKOUT

For each XINTF access, the number of lead, active, and trail cycles is based on the internal clock XTIMCLK. Strobes such as $\overline{\text{XRD}}$, $\overline{\text{XWE0}}$, $\overline{\text{XWE1}}$, and zone chip-select ($\overline{\text{XZCS}}$) change state in relationship to the rising edge of XTIMCLK. The external clock, XCLKOUT, can be configured to be either equal to or one-half the frequency of XTIMCLK.

For the case where XCLKOUT = XTIMCLK, all of the XINTF strobes will change state with respect to the rising edge of XCLKOUT. For the case where XCLKOUT = one-half or one-fourth XTIMCLK, some strobes will change state either on the rising edge of XCLKOUT or the falling edge of XCLKOUT. In the XINTF timing tables, the notation XCOHL is used to indicate that the parameter is with respect to either case; XCLKOUT rising edge (high) or XCLKOUT falling edge (low). If the parameter is always with respect to the rising edge of XCLKOUT, the notation XCOH is used.

For the case where XCLKOUT = one-half or one-fourth XTIMCLK, the XCLKOUT edge with which the change will be aligned can be determined based on the number of XTIMCLK cycles from the start of the access to the point at which the signal changes. If this number of XTIMCLK cycles is even, the alignment will be with respect to the rising edge of XCLKOUT. If this number is odd, then the signal will change with respect to the falling edge of XCLKOUT. Examples include the following:

- Strobes that change at the beginning of an access always align to the rising edge of XCLKOUT. This is because all XINTF accesses begin with respect to the rising edge of XCLKOUT.

| | | |
|---|---|---|
| Examples: | XZCSL | Zone chip-select active low |
| | XRNWL | XR/$\overline{\text{W}}$ active low |

- Strobes that change at the beginning of the active period will align to the rising edge of XCLKOUT if the total number of lead XTIMCLK cycles for the access is even. If the number of lead XTIMCLK cycles is odd, then the alignment will be with respect to the falling edge of XCLKOUT.

| | | |
|---|---|---|
| Examples: | XRDL | $\overline{\text{XRD}}$ active low |
| | XWEL | $\overline{\text{XWE1}}$ or $\overline{\text{XWE0}}$ active low |

- Strobes that change at the beginning of the trail period will align to the rising edge of XCLKOUT if the total number of lead + active XTIMCLK cycles (including hardware waitstates) for the access is even. If the number of lead + active XTIMCLK cycles (including hardware waitstates) is odd, then the alignment will be with respect to the falling edge of XCLKOUT.

| | | |
|---|---|---|
| Examples: | XRDH | $\overline{\text{XRD}}$ inactive high |
| | XWEH | $\overline{\text{XWE1}}$ or $\overline{\text{XWE0}}$ inactive high |

- Strobes that change at the end of the access will align to the rising edge of XCLKOUT if the total number of lead + active + trail XTIMCLK cycles (including hardware waitstates) is even. If the number of lead + active + trail XTIMCLK cycles (including hardware waitstates) is odd, then the alignment will be with respect to the falling edge of XCLKOUT.

| | | |
|---|---|---|
| Examples: | XZCSH | Zone chip-select inactive high |
| | XRNWH | XR/$\overline{\text{W}}$ inactive high |

### 6.15.5 External Interface Read Timing

**Table 6-37. External Interface Read Timing Requirements**

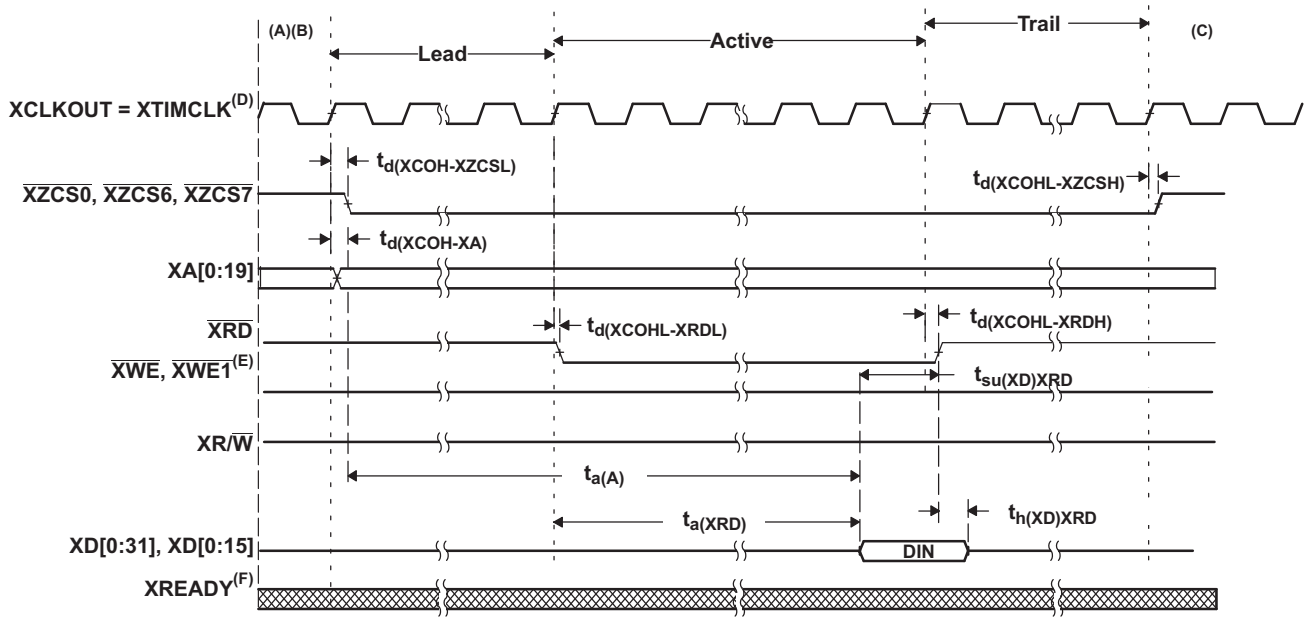
| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{a(A)}$ | Access time, read data from address valid | | (LR + AR) – 13.5 (1) | ns |
| $t_{a(XRD)}$ | Access time, read data valid from $\overline{XRD}$ active low | | AR – 13 (1) | ns |
| $t_{su(XD)XRD}$ | Setup time, read data valid before $\overline{XRD}$ strobe inactive high | 13 | | ns |
| $t_{h(XD)XRD}$ | Hold time, read data valid after $\overline{XRD}$ inactive high | 0 | | ns |

(1) LR = Lead period, read access. AR = Active period, read access. See Table 6-35.

**Table 6-38. External Interface Read Switching Characteristics**

| | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(XCOH-XZCSL)}$ | Delay time, XCLKOUT high to zone chip-select active low | 0 | 2 | ns |
| $t_{d(XCOHL-XZCSH)}$ | Delay time, XCLKOUT high/low to zone chip-select inactive high | – 0.2 | 0.9 | ns |
| $t_{d(XCOH-XA)}$ | Delay time, XCLKOUT high to address valid | | 1.5 | ns |
| $t_{d(XCOHL-XRDL)}$ | Delay time, XCLKOUT high/low to $\overline{XRD}$ active low | –0.2 | 0.8 | ns |
| $t_{d(XCOHL-XRDH)}$ | Delay time, XCLKOUT high/low to $\overline{XRD}$ inactive high | – 0.4 | 0.8 | ns |
| $t_{h(XA)XZCSH}$ | Hold time, address valid after zone chip-select inactive high | (1) | | ns |
| $t_{h(XA)XRD}$ | Hold time, address valid after $\overline{XRD}$ inactive high | (1) | | ns |

(1) During inactive cycles, the XINTF address bus always holds the last address put out on the bus. This includes alignment cycles.




A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When necessary, the device inserts an alignment cycle before an access to meet this requirement.
B. During alignment cycles, all signals transition to their inactive state.
C. XA[0:19] holds the last address put on the bus during inactive cycles, including alignment cycles except XA0, which remains high.
D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.
E. $\overline{\text{XWE1}}$ is used in 32-bit data bus mode.
F. For USEREADY = 0, the external XREADY input signal is ignored.

**Figure 6-22. Example Read Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| ≥ 2 | ≥ 5 | ≥ 0 | 0 | 0 | N/A(1) | N/A(1) | N/A(1) | N/A(1) |

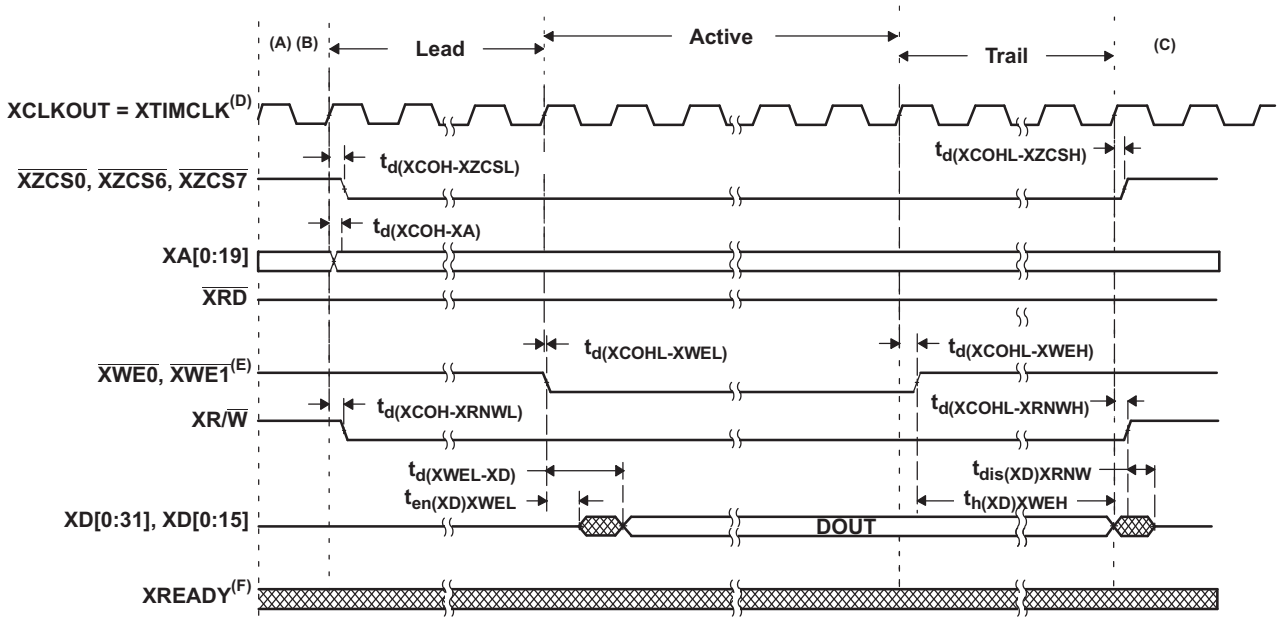
(1) N/A = Not applicable (or “Don’t care”) for this example

### 6.15.6 External Interface Write Timing

**Table 6-39. External Interface Write Switching Characteristics**

| | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(XCOH-XZCSL)}$ | Delay time, XCLKOUT high to zone chip-select active low | 0 | 2 | ns |
| $t_{d(XCOHL-XZCSH)}$ | Delay time, XCLKOUT high or low to zone chip-select inactive high | –0.2 | 0.9 | ns |
| $t_{d(XCOH-XA)}$ | Delay time, XCLKOUT high to address valid | | 1.5 | ns |
| $t_{d(XCOHL-XWEL)}$ | Delay time, XCLKOUT high/low to $\overline{XWE0}$, $\overline{XWE1}$ low | –0.3 | 0.7 | ns |
| $t_{d(XCOHL-XWEH)}$ | Delay time, XCLKOUT high/low to $\overline{XWE0}$, $\overline{XWE1}$ high | –0.5 | 0.5 | ns |
| $t_{d(XCOH-XRNWL)}$ | Delay time, XCLKOUT high to XR/$\overline{W}$ low | –0.2 | 1.5 | ns |
| $t_{d(XCOHL-XRNWH)}$ | Delay time, XCLKOUT high/low to XR/$\overline{W}$ high | 0.3 | 0.6 | ns |
| $t_{en(XD)XWEL}$ | Enable time, data bus driven from $\overline{XWE0}$, $\overline{XWE1}$ low | –7.5 | | ns |
| $t_{d(XWEL-XD)}$ | Delay time, data valid after $\overline{XWE0}$, $\overline{XWE1}$ active low | 0 | 4 | ns |
| $t_{h(XA)XZCSH}$ | Hold time, address valid after zone chip-select inactive high | (1) | | ns |
| $t_{h(XD)XWE}$ | Hold time, write data valid after $\overline{XWE0}$, $\overline{XWE1}$ inactive high | TW – 7.5 (2) | | ns |
| $t_{dis(XD)XRNW}$ | Maximum time for processor to release the data bus after XR/$\overline{W}$ inactive high | | 0 | ns |

(1) During inactive cycles, the XINTF address bus will always hold the last address put out on the bus except XA0, which remains high. This includes alignment cycles.
(2) TW = Trail period, write access. See Table 6-35.




A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When necessary, the device inserts an alignment cycle before an access to meet this requirement.
B. During alignment cycles, all signals transition to their inactive state.
C. XA[0:19] holds the last address put on the bus during inactive cycles, including alignment cycles except XA0, which remains high.
D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.
E. $\overline{\text{XWE1}}$ is used in 32-bit data bus mode.
F. For USEREADY = 0, the external XREADY input signal is ignored.

**Figure 6-23. Example Write Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| N/A(1) | N/A(1) | N/A(1) | 0 | 0 | ≥ 3 | ≥ 1 | ≥ 3 | N/A(1) |

(1) N/A = Not applicable (or “Don’t care”) for this example

### 6.15.7 External Interface Ready-on-Read Timing With One External Wait State

**Table 6-40. External Interface Read Switching Characteristics (Ready-on-Read, 1 Wait State)**

| | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(XCOH-XZCSL)}$ | Delay time, XCLKOUT high to zone chip-select active low | 0 | 2 | ns |
| $t_{d(XCOHL-XZCSH)}$ | Delay time, XCLKOUT high/low to zone chip-select inactive high | –0.2 | 0.9 | ns |
| $t_{d(XCOH-XA)}$ | Delay time, XCLKOUT high to address valid | | 1.5 | ns |
| $t_{d(XCOHL-XRDL)}$ | Delay time, XCLKOUT high/low to $\overline{XRD}$ active low | –0.2 | 0.8 | ns |
| $t_{d(XCOHL-XRDH)}$ | Delay time, XCLKOUT high/low to $\overline{XRD}$ inactive high | –0.4 | 0.8 | ns |
| $t_{h(XA)XZCSH}$ | Hold time, address valid after zone chip-select inactive high | (1) | | ns |
| $t_{h(XA)XRD}$ | Hold time, address valid after $\overline{XRD}$ inactive high | (1) | | ns |

(1) During inactive cycles, the XINTF address bus always holds the last address put out on the bus. This includes alignment cycles.

**Table 6-41. External Interface Read Timing Requirements (Ready-on-Read, 1 Wait State)**

| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{a(A)}$ | Access time, read data from address valid | | (LR + AR) – 13.5 (1) | ns |
| $t_{a(XRD)}$ | Access time, read data valid from $\overline{XRD}$ active low | | AR – 13 (1) | ns |
| $t_{su(XD)XRD}$ | Setup time, read data valid before $\overline{XRD}$ strobe inactive high | 13 | | ns |
| $t_{h(XD)XRD}$ | Hold time, read data valid after $\overline{XRD}$ inactive high | 0 | | ns |

(1) LR = Lead period, read access. AR = Active period, read access. See Table 6-35.

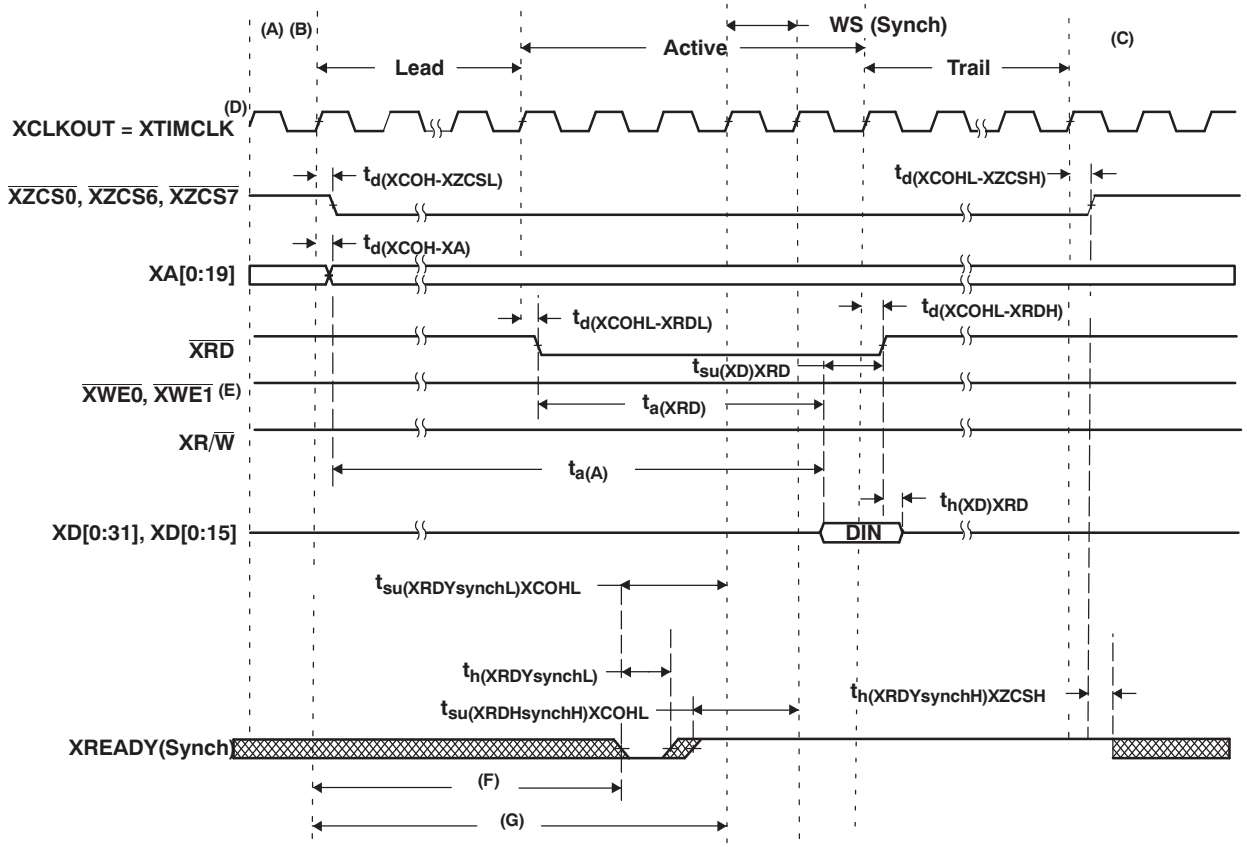
**Table 6-42. Synchronous XREADY Timing Requirements (Ready-on-Read, 1 Wait State)(1)**

| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{su(XRDYsynchL)XCOHL}$ | Setup time, XREADY (synchronous) low before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYsynchL)}$ | Hold time, XREADY (synchronous) low | $1t_{c(XTIM)}$ | | ns |
| $t_{su(XRDYsynchH)XCOHL}$ | Setup time, XREADY (synchronous) high before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYsynchH)XZCSH}$ | Hold time, XREADY (synchronous) held high after zone chip select high | 0 | | ns |

(1) The first XREADY (synchronous) sample occurs with respect to E in Figure 6-24:
E = (XRDLEAD + XRDACTIVE) $t_{c(XTIM)}$
When first sampled, if XREADY (synchronous) is found to be high, then the access will finish. If XREADY (synchronous) is found to be low, it is sampled again each $t_{c(XTIM)}$ until it is found to be high.
For each sample (n) the setup time (F) with respect to the beginning of the access can be calculated as:
F = (XRDLEAD + XRDACTIVE +n − 1) $t_{c(XTIM)}$ − $t_{su(XRDYsynchL)XCOHL}$
where n is the sample number: n = 1, 2, 3, and so forth.

**Table 6-43. Asynchronous XREADY Timing Requirements (Ready-on-Read, 1 Wait State)**

| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{su(XRDYAsynchL)XCOHL}$ | Setup time, XREADY (asynchronous) low before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYAsynchL)}$ | Hold time, XREADY (asynchronous) low | $1t_{c(XTIM)}$ | | ns |
| $t_{su(XRDYAsynchH)XCOHL}$ | Setup time, XREADY (asynchronous) high before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYasynchH)XZCSH}$ | Hold time, XREADY (asynchronous) held high after zone chip select high | 0 | | ns |




**Legend:**

▨ = Don't care. Signal can be high or low during this time.

A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When *necessary*, the device inserts an alignment cycle before an access to meet this requirement.

B. During alignment cycles, all signals transition to their inactive state.

C. During inactive cycles, the XINTF address bus always holds the last address put out on the bus except XA0, which remains high. This includes alignment cycles.

D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.

E. $\overline{\text{XWE1}}$ is valid only in 32-bit data bus mode.

F. For each sample, setup time from the beginning of the access (E) can be calculated as: D = (XRDLEAD + XRDACTIVE +n - 1) $t_{c(XTIM)}$ – $t_{su(XRDYsynchL)XCOHL}$

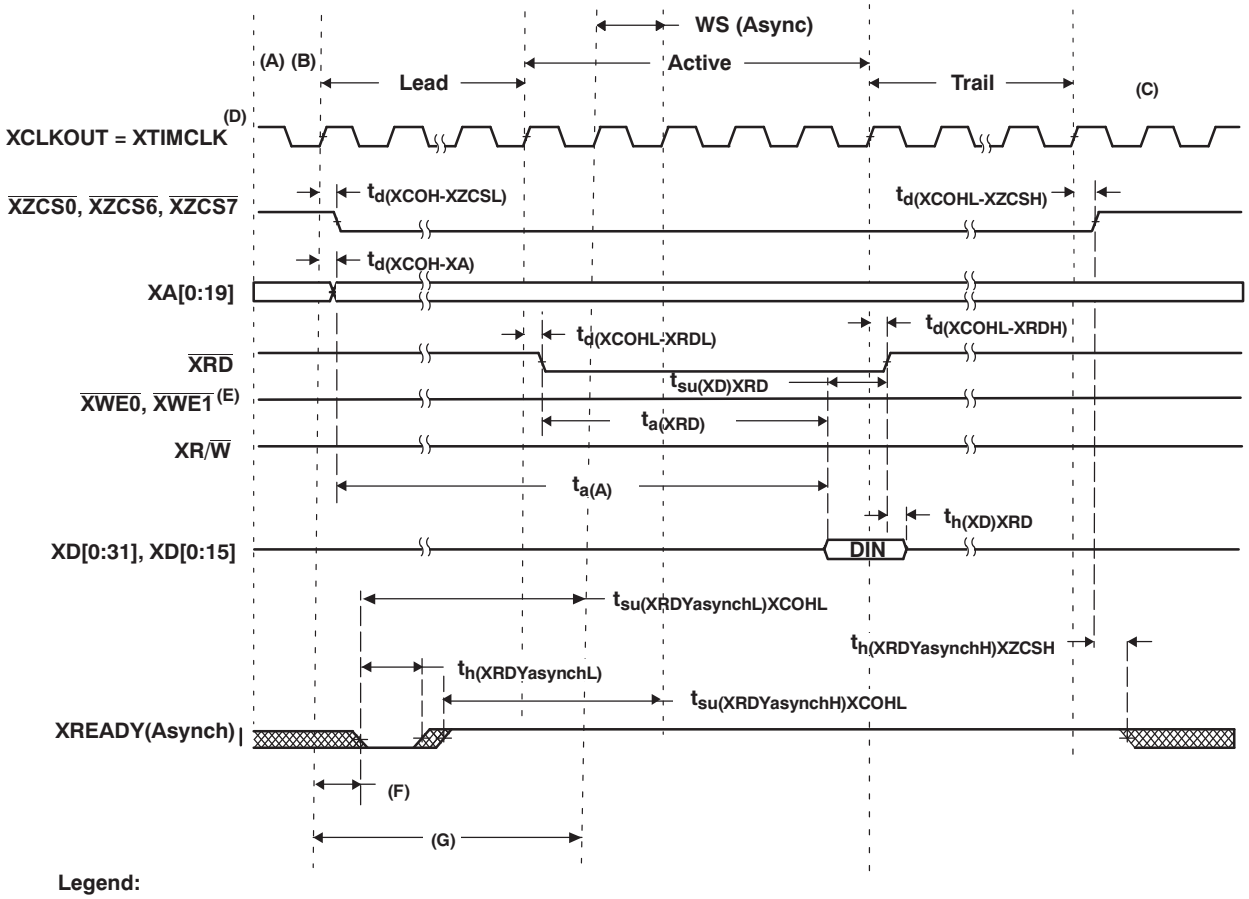
G. Reference for the first sample is with respect to this point: F = (XRDLEAD + XRDACTIVE) $t_{c(XTIM)}$ where n is the sample number: n = 1, 2, 3, and so forth.

**Figure 6-24. Example Read With Synchronous XREADY Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| ≥ 2 | 5 | ≥ 0 | 1 | 0 | N/A(1) | N/A(1) | N/A(1) | 0 = XREADY (Synch) |

(1) N/A = "Don't care" for this example




**Legend:**

= Don't care. Signal can be high or low during this time.

A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When necessary, the device will insert an alignment cycle before an access to meet this requirement.

B. During alignment cycles, all signals will transition to their inactive state.

C. During inactive cycles, the XINTF address bus will always hold the last address put out on the bus except XA0, which remains high. This includes alignment cycles.

D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.

E. $\overline{XWE1}$ is valid only in 32-bit data bus mode.

F. For each sample, setup time from the beginning of the access can be calculated as: E = (XRDLEAD + XRDACTIVE -3 +n) $t_{c(XTIM)}$ – $t_{su(XRDYasynchL)XCOHL}$ where n is the sample number: n = 1, 2, 3, and so forth.

G. Reference for the first sample is with respect to this point: F = (XRDLEAD + XRDACTIVE –2) $t_{c(XTIM)}$

**Figure 6-25. Example Read With Asynchronous XREADY Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| ≥ 2 | 5 | ≥ 0 | 1 | 0 | N/A(1) | N/A(1) | N/A(1) | 1 = XREADY (Async) |

(1) N/A = "Don't care" for this example

### 6.15.8 External Interface Ready-on-Write Timing With One External Wait State

**Table 6-44. External Interface Write Switching Characteristics (Ready-on-Write, 1 Wait State)**

| | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(XCOH-XZCSL)}$ | Delay time, XCLKOUT high to zone chip-select active low | 0 | 2 | ns |
| $t_{d(XCOHL-XZCSH)}$ | Delay time, XCLKOUT high or low to zone chip-select inactive high | –0.2 | 0.9 | ns |
| $t_{d(XCOH-XA)}$ | Delay time, XCLKOUT high to address valid | | 1.5 | ns |
| $t_{d(XCOHL-XWEL)}$ | Delay time, XCLKOUT high/low to $\overline{XWE0}$, $\overline{XWE1}$ low(1) | –0.3 | 0.7 | ns |
| $t_{d(XCOHL-XWEH)}$ | Delay time, XCLKOUT high/low to $\overline{XWE0}$, $\overline{XWE1}$ high(1) | –0.5 | 0.5 | ns |
| $t_{d(XCOH-XRNWL)}$ | Delay time, XCLKOUT high to XR/$\overline{W}$ low | –0.2 | 1.5 | ns |
| $t_{d(XCOHL-XRNWH)}$ | Delay time, XCLKOUT high/low to XR/$\overline{W}$ high | 0.3 | 0.6 | ns |
| $t_{en(XD)XWEL}$ | Enable time, data bus driven from $\overline{XWE0}$, $\overline{XWE1}$ low | –7.5 | | ns |
| $t_{d(XWEL-XD)}$ | Delay time, data valid after $\overline{XWE0}$, $\overline{XWE1}$ active low | 0 | 4 | ns |
| $t_{h(XA)XZCSH}$ | Hold time, address valid after zone chip-select inactive high | (2) | | ns |
| $t_{h(XD)XWE}$ | Hold time, write data valid after $\overline{XWE0}$, $\overline{XWE1}$ inactive high(1) | TW – 7.5 (3) | | ns |
| $t_{dis(XD)XRNW}$ | Maximum time for processor to release the data bus after XR/$\overline{W}$ inactive high | | 0 | ns |

(1) $\overline{XWE1}$ is used in 32-bit data bus mode only. In 16-bit, this signal is XA0.
(2) During inactive cycles, the XINTF address bus always holds the last address put out on the bus. This includes alignment cycles.
(3) TW = trail period, write access (see Table 6-35)

**Table 6-45. Synchronous XREADY Timing Requirements (Ready-on-Write, 1 Wait State)(1)**

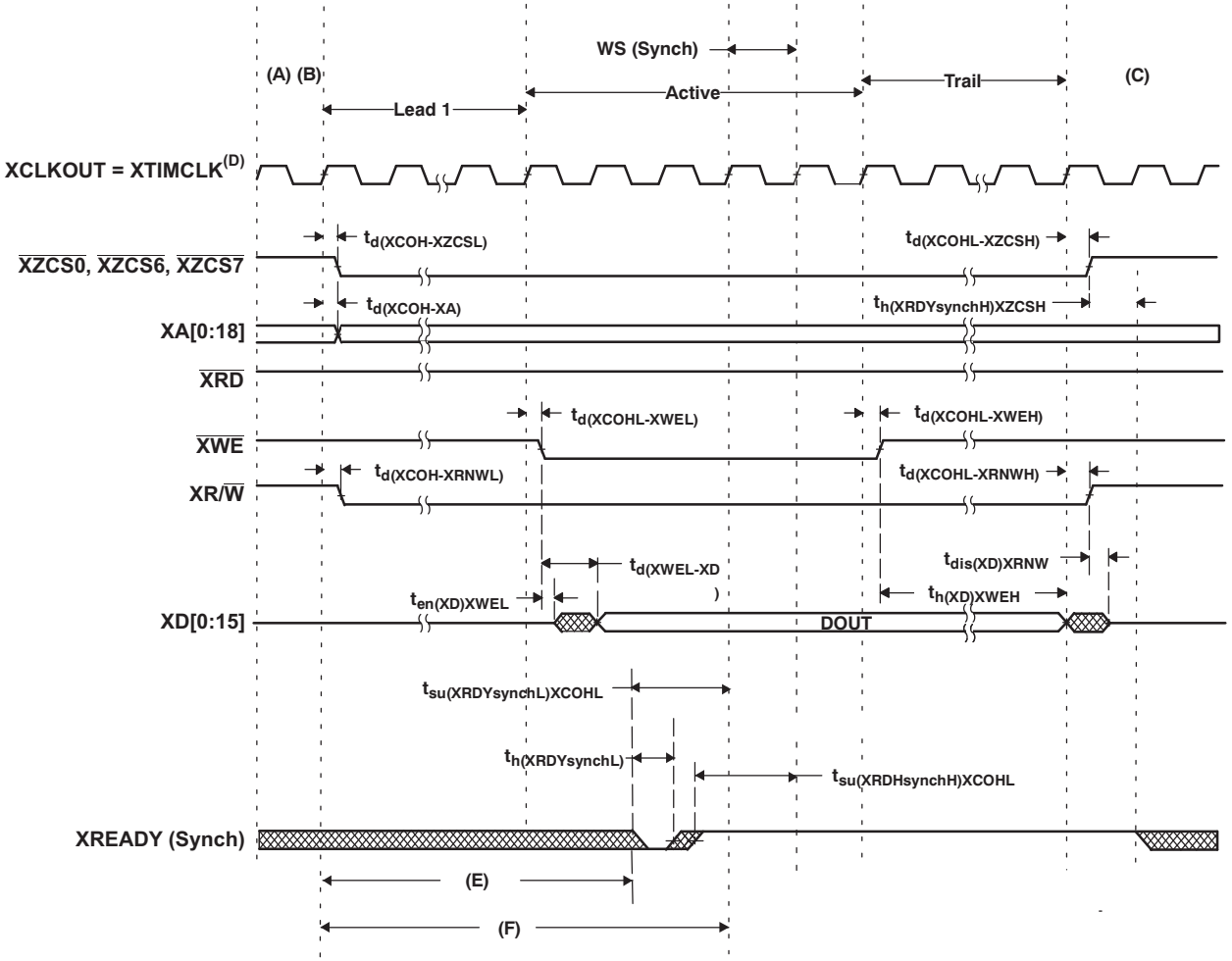
| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{su(XRDYsynchL)XCOHL}$ | Setup time, XREADY (synchronous) low before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYsynchL)}$ | Hold time, XREADY (synchronous) low | $1t_{c(XTIM)}$ | | ns |
| $t_{su(XRDYsynchH)XCOHL}$ | Setup time, XREADY (synchronous) high before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYsynchH)XZCSH}$ | Hold time, XREADY (synchronous) held high after zone chip select high | 0 | | ns |

(1) The first XREADY (synchronous) sample occurs with respect to E in Figure 6-26:
E =(XWRLEAD + XWRACTIVE) $t_{c(XTIM)}$
When first sampled, if XREADY (synchronous) is high, then the access will complete. If XREADY (synchronous) is low, it is sampled again each $t_{c(XTIM)}$ until it is high.
For each sample, setup time from the beginning of the access can be calculated as:
F = (XWRLEAD + XWRACTIVE +n –1) $t_{c(XTIM)}$ – $t_{su(XRDYsynchL)XCOHL}$
where n is the sample number: n = 1, 2, 3, and so forth.

**Table 6-46. Asynchronous XREADY Timing Requirements (Ready-on-Write, 1 Wait State)(1)**

| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{su(XRDYasynchL)XCOHL}$ | Setup time, XREADY (asynchronous) low before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYasynchL)}$ | Hold time, XREADY (asynchronous) low | $1t_{c(XTIM)}$ | | ns |
| $t_{su(XRDYasynchH)XCOHL}$ | Setup time, XREADY (asynchronous) high before XCLKOUT high/low | 8 | | ns |
| $t_{h(XRDYasynchH)XZCSH}$ | Hold time, XREADY (asynchronous) held high after zone chip select high | 0 | | ns |

(1) The first XREADY (synchronous) sample occurs with respect to E in Figure 6-26:
E = (XWRLEAD + XWRACTIVE –2) $t_{c(XTIM)}$. When first sampled, if XREADY (asynchronous) is high, then the access will complete. If XREADY (asynchronous) is low, it is sampled again each $t_{c(XTIM)}$ until it is high.
For each sample, setup time from the beginning of the access can be calculated as:
F = (XWRLEAD + XWRACTIVE –3 + n) $t_{c(XTIM)}$ – $t_{su(XRDYasynchL)XCOHL}$
where n is the sample number: n = 1, 2, 3, and so forth.




**Legend:**

= Don't care. Signal can be high or low during this time.

A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When necessary, the device inserts an alignment cycle before an access to meet this requirement.

B. During alignment cycles, all signals will transition to their inactive state.

C. During inactive cycles, the XINTF address bus always holds the last address put out on the bus except XA0, which remains high. This includes alignment cycles.

D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.

E. $\overline{XWE1}$ is used in 32-bit data bus mode only.

F. For each sample, setup time from the beginning of the access can be calculated as E = (XWRLEAD + XWRACTIVE + n –1) $t_{c(XTIM)}$ – $t_{su(XRDYsynchL)XCOH}$ where n is the sample number: n = 1, 2, 3, and so forth.

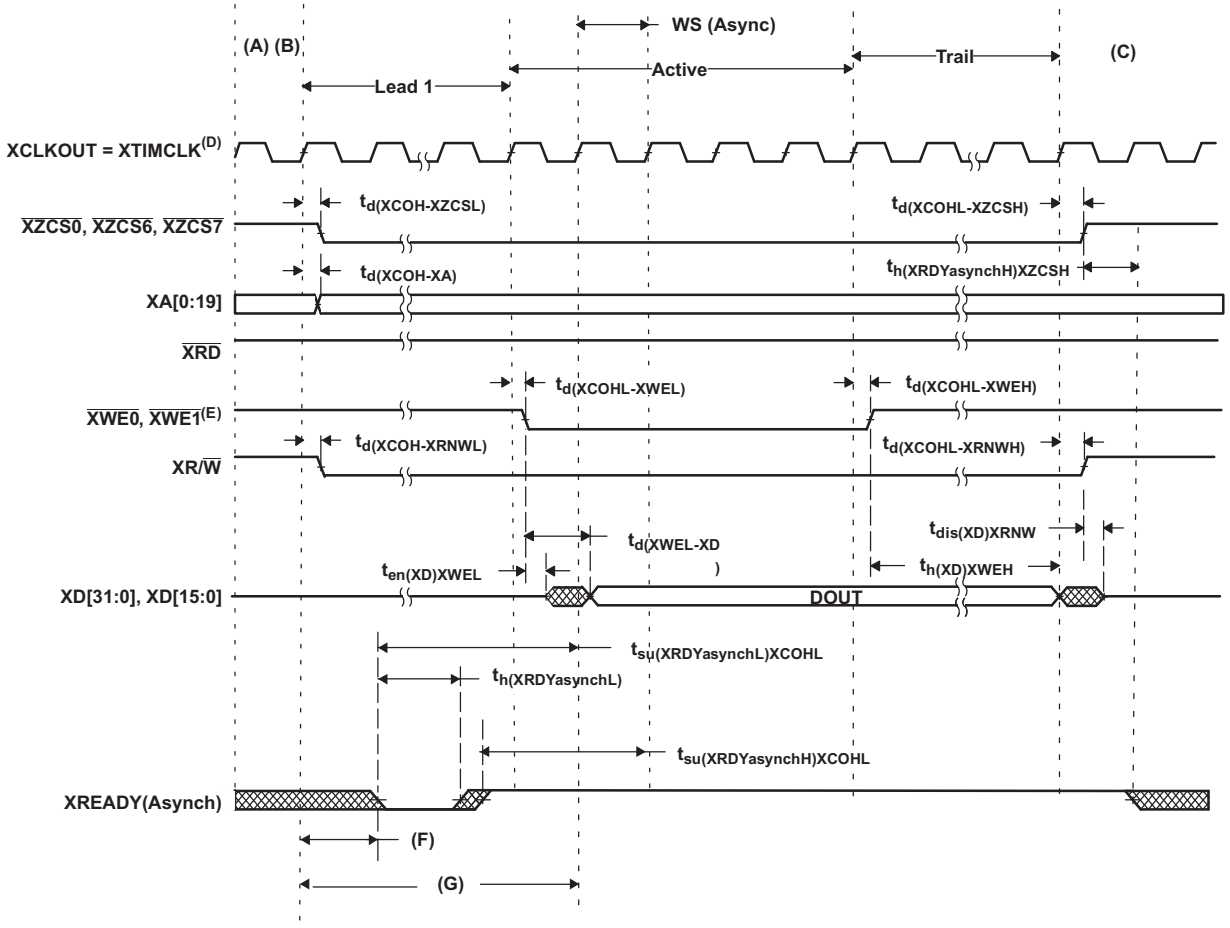
G. Reference for the first sample is with respect to this point: F = (XWRLEAD + XWRACTIVE) $t_{c(XTIM)}$

**Figure 6-26. Write With Synchronous XREADY Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| N/A(1) | N/A(1) | N/A(1) | 1 | 0 | ≥ 3 | 1 | ≥3 | 0 = XREADY (Synch) |

(1) N/A = "Don't care" for this example.




**Legend:**

▨ = Don't care. Signal can be high or low during this time.

A. All XINTF accesses (lead period) begin on the rising edge of XCLKOUT. When necessary, the device inserts an alignment cycle before an access to meet this requirement.

B. During alignment cycles, all signals transition to their inactive state.

C. During inactive cycles, the XINTF address bus always holds the last address put out on the bus except XA0, which remains high. This includes alignment cycles.

D. Timings are also relevant for XCLKOUT = 1/2 XTIMCLK and XCLKOUT = 1/4 XTIMCLK.

E. $\overline{\text{XWE1}}$ is used in 32-bit data bus mode only.

F. For each sample, set up time from the beginning of the access can be calculated as: E = (XWRLEAD + XWRACTIVE -3 + n) $t_{c(XTIM)}$ – $t_{su(XRDYasynchL)XCOHL}$ where n is the sample number: n = 1, 2, 3, and so forth.

G. Reference for the first sample is with respect to this point: F = (XWRLEAD + XWRACTIVE – 2) $t_{c(XTIM)}$

**Figure 6-27. Write With Asynchronous XREADY Access**

XTIMING register parameters used for this example (based on 300-MHz system clock):

| XRDLEAD | XRDACTIVE | XRDTRAIL | USEREADY | X2TIMING | XWRLEAD | XWRACTIVE | XWRTRAIL | READYMODE |
|---|---|---|---|---|---|---|---|---|
| N/A(1) | N/A(1) | N/A(1) | 1 | 0 | ≥ 3 | 3 | ≥ 3 | 1 = XREADY (Async) |

(1) N/A = "Don't care" for this example

### 6.15.9 $\overline{\text{XHOLD}}$ and $\overline{\text{XHOLDA}}$ Timing

If the HOLD mode bit is set while $\overline{\text{XHOLD}}$ and $\overline{\text{XHOLDA}}$ are both low (external bus accesses granted), the $\overline{\text{XHOLDA}}$ signal is forced high (at the end of the current cycle) and the external interface is taken out of high-impedance mode.

On a reset ($\overline{\text{XRS}}$), the HOLD mode bit is set to 0. If the $\overline{\text{XHOLD}}$ signal is active low on a system reset, the bus and all signal strobes must be in high-impedance mode, and the $\overline{\text{XHOLDA}}$ signal is also driven active low.

When HOLD mode is enabled and $\overline{\text{XHOLDA}}$ is active low (external bus grant active), the CPU can still execute code from internal memory. If an access is made to the external interface, the CPU is stalled until the $\overline{\text{XHOLD}}$ signal is removed.

An external DMA request, when granted, places the following signals in a high-impedance mode:

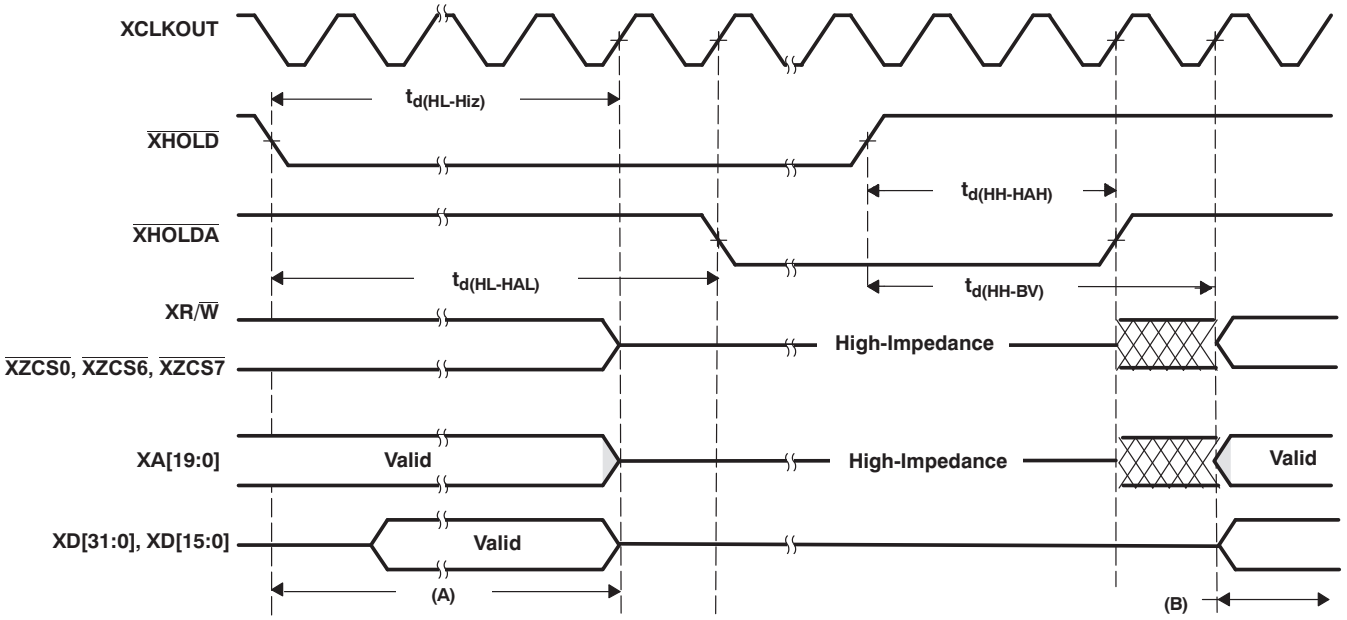
| | |
|---|---|
| XA[19:0] | $\overline{\text{XZCS0}}$ |
| XD[31:0], XD[15:0] | $\overline{\text{XZCS6}}$ |
| $\overline{\text{XWE0}}$, $\overline{\text{XWE1}}$, $\overline{\text{XRD}}$ | $\overline{\text{XZCS7}}$ |
| XR/$\overline{\text{W}}$ | |

All other signals not listed in this group remain in their default or functional operational modes during these signal events.

**Table 6-47. $\overline{\text{XHOLD}}$/$\overline{\text{XHOLDA}}$ Timing Requirements (1) (2) (3)**

| | | MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(HL-HiZ)}$ | Delay time, $\overline{\text{XHOLD}}$ low to Hi-Z on all address, data, and control | | $4t_{c(XTIM)} + t_{c(XCO)} + 20$ | ns |
| $t_{d(HL-HAL)}$ | Delay time, $\overline{\text{XHOLD}}$ low to $\overline{\text{XHOLDA}}$ low | | $4t_{c(XTIM)} + 2t_{c(XCO)} + 20$ | ns |
| $t_{d(HH-HAH)}$ | Delay time, $\overline{\text{XHOLD}}$ high to $\overline{\text{XHOLDA}}$ high | | $4t_{c(XTIM)} + 20$ | ns |
| $t_{d(HH-BV)}$ | Delay time, $\overline{\text{XHOLD}}$ high to bus valid | | $6t_{c(XTIM)} + 20$ | ns |

(1) When a low signal is detected on $\overline{\text{XHOLD}}$, all pending XINTF accesses will be completed before the bus is placed in a high-impedance state.
(2) The state of $\overline{\text{XHOLD}}$ is latched on the rising edge of XTIMCLK.
(3) After the $\overline{\text{XHOLD}}$ is detected low or high, all bus transitions and $\overline{\text{XHOLDA}}$ transitions occur with respect to the rising edge of XCLKOUT. Thus, for this mode where XCLKOUT = 1/2 XTIMCLK, the transitions can occur up to 1 XTIMCLK cycle earlier than the maximum value specified.




A. All pending XINTF accesses are completed.
B. Normal XINTF operation resumes.

**Figure 6-28. External Interface Hold Waveform**

## 6.16 Multichannel Buffered Serial Port (McBSP) Timing

### 6.16.1 McBSP Transmit and Receive Timing

**Table 6-48. McBSP Timing Requirements(1) (2)**

| NO. | | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|---|
| | | McBSP module clock (CLKG, CLKX, CLKR) range | | 1 | | kHz |
| | | | | | 40 (3) | MHz |
| | | McBSP module cycle time (CLKG, CLKX, CLKR) range | | 25 | | ns |
| | | | | | 1 | ms |
| M11 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X ext | 2P | | ns |
| M12 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | P – 4 | | ns |
| M13 | $t_{r(CKRX)}$ | Rise time, CLKR/X | CLKR/X ext | | 4 | ns |
| M14 | $t_{f(CKRX)}$ | Fall time, CLKR/X | CLKR/X ext | | 4 | ns |
| M15 | $t_{su(FRH-CKRL)}$ | Setup time, external FSR high before CLKR low | CLKR int | 20 | | ns |
| | | | CLKR ext | 2 | | |
| M16 | $t_{h(CKRL-FRH)}$ | Hold time, external FSR high after CLKR low | CLKR int | 0 | | ns |
| | | | CLKR ext | 6 | | |
| M17 | $t_{su(DRV-CKRL)}$ | Setup time, DR valid before CLKR low | CLKR int | 20 | | ns |
| | | | CLKR ext | 2 | | |
| M18 | $t_{h(CKRL-DRV)}$ | Hold time, DR valid after CLKR low | CLKR int | 0 | | ns |
| | | | CLKR ext | 6 | | |
| M19 | $t_{su(FXH-CKXL)}$ | Setup time, external FSX high before CLKX low | CLKX int | 20 | | ns |
| | | | CLKX ext | 2 | | |
| M20 | $t_{h(CKXL-FXH)}$ | Hold time, external FSX high after CLKX low | CLKX int | 0 | | ns |
| | | | CLKX ext | 6 | | |

(1) Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

(2) 2P = 1/CLKG in ns. CLKG is the output of sample rate generator mux. CLKG = $\frac{\text{CLKSRG}}{(1 + \text{CLKGDV})}$ CLKSRG can be LSPCLK, CLKX, CLKR as source. CLKSRG ≤ (SYSCLKOUT/2). McBSP performance is limited by I/O buffer switching speed.

(3) Internal clock prescalers must be adjusted such that the McBSP clock (CLKG, CLKX, CLKR) speeds are not greater than the I/O buffer speed limit ( 40 MHz).

### Table 6-49. McBSP Switching Characteristics[1] [2]

| NO. | PARAMETER | | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| M1 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | | CLKR/X int | 2P | | ns |
| M2 | $t_{w(CKRXH)}$ | Pulse duration, CLKR/X high | | CLKR/X int | D – 2 [3] | D + 2 [3] | ns |
| M3 | $t_{w(CKRXL)}$ | Pulse duration, CLKR/X low | | CLKR/X int | C – 2 [3] | C + 2 [3] | ns |
| M4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | | CLKR int | 0 | 4 | ns |
| | | | | CLKR ext | 3 | 20 | |
| M5 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | | CLKX int | 0 | 4 | ns |
| | | | | CLKX ext | 3 | 20 | |
| M6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, CLKX high to DX high impedance following last data bit | | CLKX int | | 8 | ns |
| | | | | CLKX ext | | 14 | |
| M7 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid. This applies to all bits except the first bit transmitted. | | CLKX int | | 4 | ns |
| | | | | CLKX ext | | 20 | |
| | | Delay time, CLKX high to DX valid | DXENA = 0 | CLKX int | | 4 | |
| | | | | CLKX ext | | 20 | |
| | | Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes | DXENA = 1 | CLKX int | | P + 4 | |
| | | | | CLKX ext | | P + 20 | |
| M8 | $t_{en(CKXH-DX)}$ | Enable time, CLKX high to DX driven | DXENA = 0 | CLKX int | 0 | | ns |
| | | | | CLKX ext | 10 | | |
| | | Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes | DXENA = 1 | CLKX int | P | | |
| | | | | CLKX ext | P + 10 | | |
| M9 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid | DXENA = 0 | FSX int | | 4 | ns |
| | | | | FSX ext | | 16 | |
| | | Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode. | DXENA = 1 | FSX int | | P + 4 | |
| | | | | FSX ext | | P + 16 | |
| M10 | $t_{en(FXH-DX)}$ | Enable time, FSX high to DX driven | DXENA = 0 | FSX int | 0 | | ns |
| | | | | FSX ext | 6 | | |
| | | Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode | DXENA = 1 | FSX int | P | | |
| | | | | FSX ext | P + 6 | | |

(1) Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) 2P = 1/CLKG in ns.
(3) C = CLKRX low pulse width = P
D = CLKRX high pulse width = P

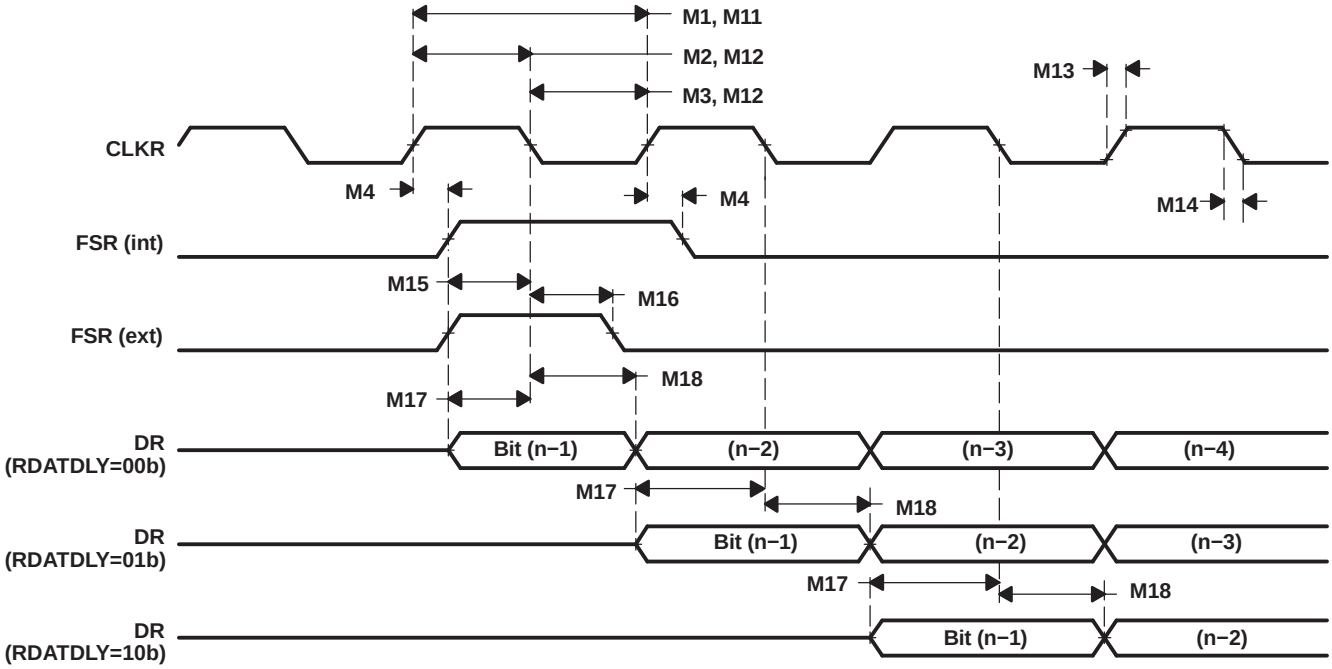



**Figure 6-29. McBSP Receive Timing**

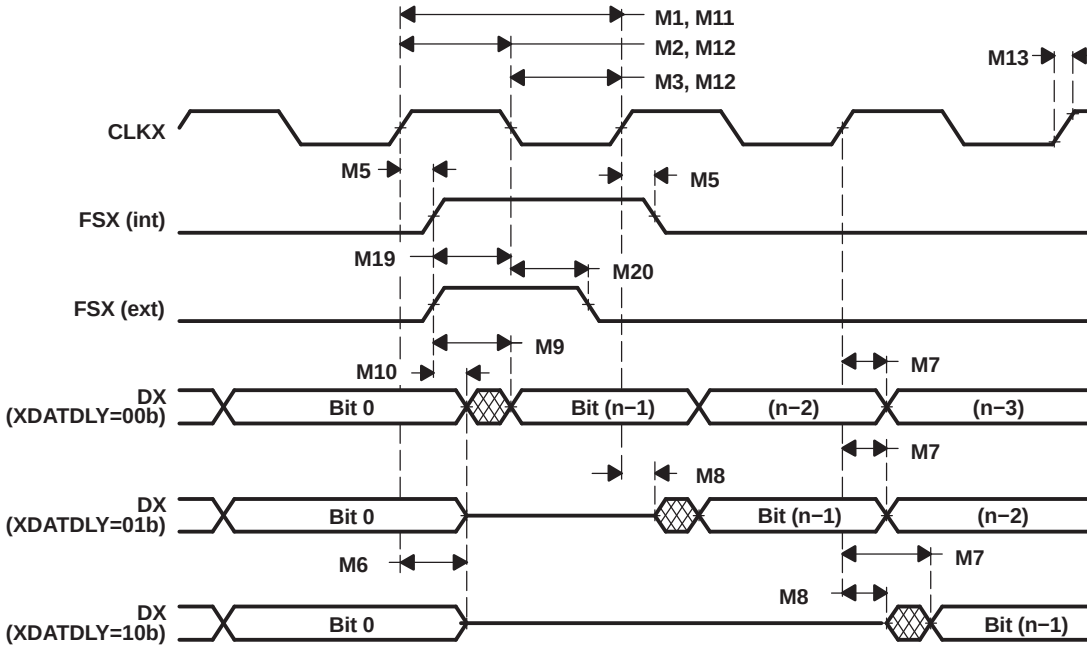



**Figure 6-30. McBSP Transmit Timing**

### 6.16.2 McBSP as SPI Master or Slave Timing

**Table 6-50. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 0)**

| NO. | | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M30 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | 30 | | 8P – 10 | | ns |
| M31 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | 1 | | 8P – 10 | | ns |
| M32 | $t_{su(BFXL-CKXH)}$ | Setup time, FSX low before CLKX high | | | 8P + 10 | | ns |
| M33 | $t_{c(CKX)}$ | Cycle time, CLKX | 2P(1) | | 16P | | ns |

(1) 2P = 1/CLKG

**Table 6-51. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 0)**

| NO. | PARAMETER | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M24 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low | 2P(1) | | | | ns |
| M25 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high | P | | | | ns |
| M26 | $t_{d(CLKXH-DXV)}$ | Delay time, CLKX low to DX valid | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M28 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | 6 | | 6P + 6 | | ns |
| M29 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

For all SPI slave modes, CLKX has to be minimum 8 CLKG cycles. Also CLKG should be LSPCLK/2 by setting CLKSM = CLKGDV = 1. With maximum LSPCLK speed of 150 MHz, CLKX maximum frequency will be LSPCLK/16 , that is 9.375 MHz and P = 6.67 ns.

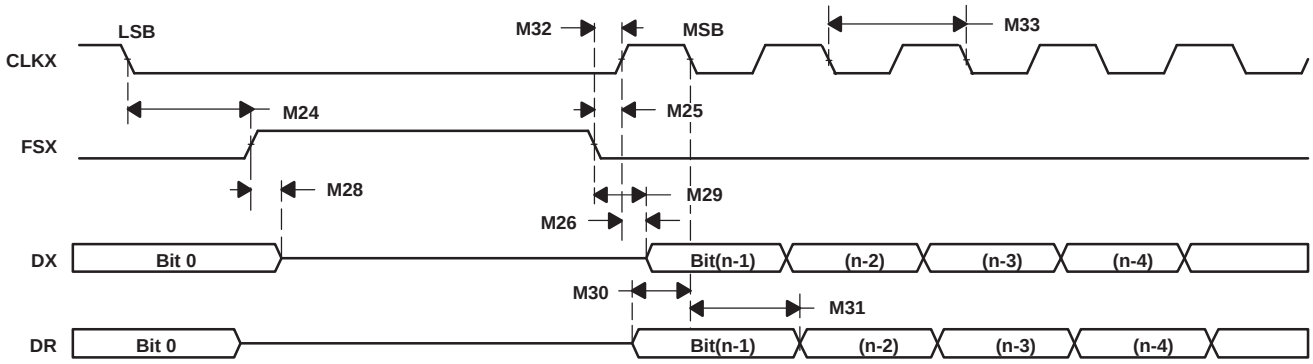



**Figure 6-31. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0**

**Table 6-52. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 0)**

| NO. | | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M39 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 30 | | 8P – 10 | | ns |
| M40 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 1 | | 8P – 10 | | ns |
| M41 | $t_{su(FXL-CKXH)}$ | Setup time, FSX low before CLKX high | | | 16P + 10 | | ns |
| M42 | $t_{c(CKX)}$ | Cycle time, CLKX | 2P (1) | | 16P | | ns |

(1) 2P = 1/CLKG

**Table 6-53. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 0)**

| NO. | PARAMETER | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M34 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low | P | | | | ns |
| M35 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high | 2P (1) | | | | ns |
| M36 | $t_{d(CLKXL-DXV)}$ | Delay time, CLKX low to DX valid | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M37 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | P + 6 | | 7P + 6 | | ns |
| M38 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

For all SPI slave modes, CLKX must be a minimum of 8 CLKG cycles. Also, CLKG should be LSPCLK/2 by setting CLKSM = CLKGDV = 1. With a maximum LSPCLK speed of 150 MHz, CLKX maximum frequency is LSPCLK/16; that is, 9.375 MHz and P = 6.67 ns.

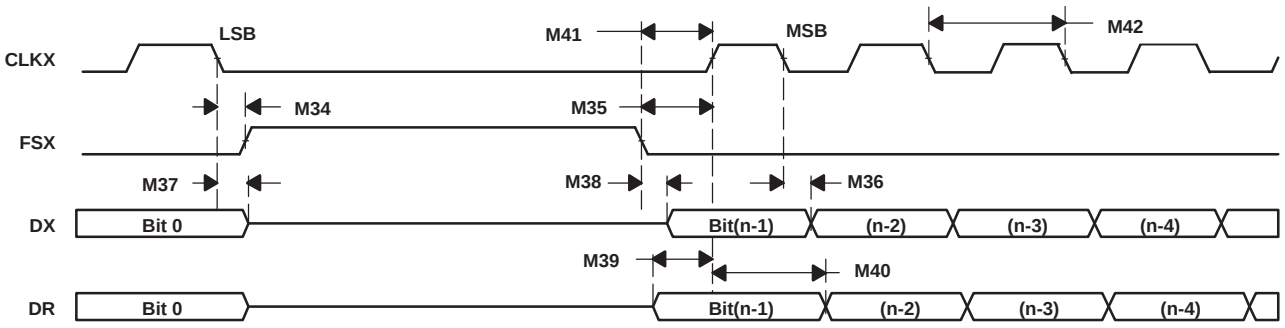



**Figure 6-32. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0**

**Table 6-54. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 1)**

| NO. | | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M49 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 30 | | 8P – 10 | | ns |
| M50 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 1 | | 8P – 10 | | ns |
| M51 | $t_{su(FXL-CKXL)}$ | Setup time, FSX low before CLKX low | | | 8P + 10 | | ns |
| M52 | $t_{c(CKX)}$ | Cycle time, CLKX | 2P (1) | | 16P | | ns |

(1) 2P = 1/CLKG

**Table 6-55. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 1)**

| NO. | PARAMETER | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M43 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high | 2P (1) | | | | ns |
| M44 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low | P | | | | ns |
| M45 | $t_{d(CLKXL-DXV)}$ | Delay time, CLKX low to DX valid | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M47 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | 6 | | 6P + 6 | | ns |
| M48 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

For all SPI slave modes, CLKX must be a minimum of 8 CLKG cycles. Also, CLKG should be LSPCLK/2 by setting CLKSM = CLKGDV = 1. With maximum LSPCLK speed of 150 MHz, CLKX maximum frequency will be LSPCLK/16; that is, 9.375 MHz and P = 6.67 ns.

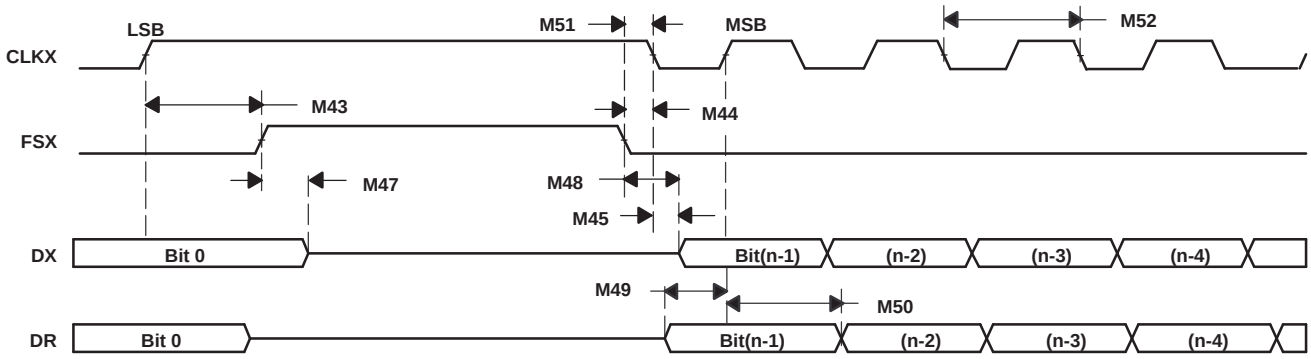



**Figure 6-33. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1**

**Table 6-56. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 1)**

| NO. | | | MASTER | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M58 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | 30 | | 8P – 10 | | ns |
| M59 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | 1 | | 8P – 10 | | ns |
| M60 | $t_{su(FXL-CKXL)}$ | Setup time, FSX low before CLKX low | | | 16P + 10 | | ns |
| M61 | $t_{c(CKX)}$ | Cycle time, CLKX | 2P(1) | | 16P | | ns |

(1) 2P = 1/CLKG

For all SPI slave modes, CLKX must be a minimum of 8 CLKG cycles. Also CLKG should be LSPCLK/2 by setting CLKSM = CLKGDV = 1. With maximum LSPCLK speed of 150 MHz, CLKX maximum frequency is LSPCLK/16 , that is 9.375 MHz and P = 6.67 ns.

**Table 6-57. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 1)(1)**

| NO. | PARAMETER | | MASTER(2) | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| M53 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high | P | | | | ns |
| M54 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low | 2P(1) | | | | ns |
| M55 | $t_{d(CLKXH-DXV)}$ | Delay time, CLKX high to DX valid | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M56 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | P + 6 | | 7P + 6 | | ns |
| M57 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG
(2) C = CLKX low pulse width = P
D = CLKX high pulse width = P

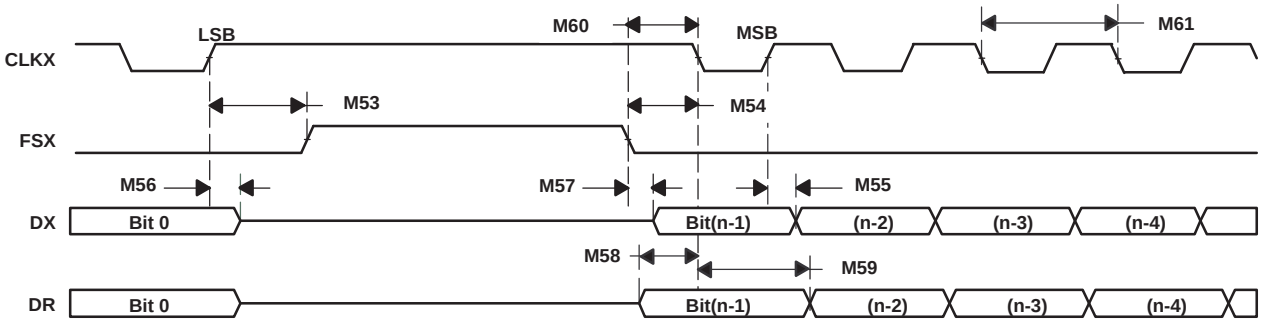



**Figure 6-34. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1**

# 7 Revision History

This data sheet revision history highlights the technical changes made to the SPRS516A device-specific data sheet to make it an SPRS516B revision.

**Scope:** See table below.

| LOCATION | ADDITIONS, DELETIONS, AND MODIFICATIONS |
|---|---|
| Section 1.2 | Features:<br>• Added "Community Resources" feature |
| Table 2-2 | Signal Descriptions:<br>• ZHH BALL # H5, ZFE BALL# J3: Updated DESCRIPTION of SCITXDB<br>• ZHH BALL # L11, ZFE BALL# T13: Updated DESCRIPTION of EQEP1I |
| Figure 4-5 | ePWM Submodules Showing Critical Internal Signal Interconnections:<br>• Changed TBCTL[CNTLDE] to TBCTL[PHSEN] |
| Table 6-6 | XCLKIN/X1 Timing Requirements – PLL Enabled:<br>• C9 [$t_{f(CI)}$]: Changed MAX value from 2 ns to 4 ns<br>• C10 [$t_{r(CI)}$]: Changed MAX value from 2 ns to 4 ns<br>• C11 [$t_{w(CIL)}$]:<br>– Changed MIN value from 45% to 40%<br>– Changed MAX value from 55% to 60%<br>• C12 [$t_{w(CIH)}$]:<br>– Changed MIN value from 45% to 40%<br>– Changed MAX value from 55% to 60% |

# 8 Thermal/Mechanical Data

Table 8-1 and Table 8-2 show the thermal data.

The mechanical package diagram(s) that follow the tables reflect the most current released mechanical data available for the designated device(s).

**Table 8-1. Thermal Model 179-Ball ZHH Results**

| AIR FLOW | | | | |
|---|---|---|---|---|
| **PARAMETER** | **0 lfm** | **150 lfm** | **250 lfm** | **500 lfm** |
| $\theta_{JA}$[°C/W] High k PCB | 40.8 | 32.4 | 31.0 | 29.1 |
| $\Psi_{JT}$[°C/W] | 0.4 | 0.5 | 0.6 | 0.8 |
| $\Psi_{JB}$ | 21.0 | 20.4 | 20.2 | 19.9 |
| $\theta_{JC}$ | 10.3 | | | |
| $\theta_{JB}$ | 21.2 | | | |

**Table 8-2. Thermal Model 256-Ball ZFE Results**

| AIR FLOW | | | | |
|---|---|---|---|---|
| **PARAMETER** | **0 lfm** | **150 lfm** | **250 lfm** | **500 lfm** |
| $\theta_{JA}$[°C/W] High k PCB | 30 | 21.8 | 20.6 | 19.1 |
| $\Psi_{JT}$[°C/W] | 1.24 | 2.63 | 3.15 | 4.05 |
| $\Psi_{JB}$ | 14 | 13.6 | 13.5 | 13.4 |
| $\theta_{JC}$ | 14 | | | |
| $\theta_{JB}$ | 13.9 | | | |

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead/ Ball Finish | MSL Peak Temp (3) | Samples (Requires Login) |
|---|---|---|---|---|---|---|---|---|---|
| TMS320C28341ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28341ZFET | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28341ZHHT | ACTIVE | BGA MICROSTAR | ZHH | 179 | 160 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28342ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28342ZFET | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Contact TI Distributor or Sales Office |
| TMS320C28343ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28343ZFET | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Contact TI Distributor or Sales Office |
| TMS320C28343ZHHT | ACTIVE | BGA MICROSTAR | ZHH | 179 | 160 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28344ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28344ZFET | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28345ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28345ZFET | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28345ZHHT | ACTIVE | BGA MICROSTAR | ZHH | 179 | 160 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28346ZFEQ | ACTIVE | BGA | ZFE | 256 | 90 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Purchase Samples |
| TMS320C28346ZFET | ACTIVE | BGA | ZFE | 256 | 1 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168 HR | Contact TI Distributor or Sales Office |

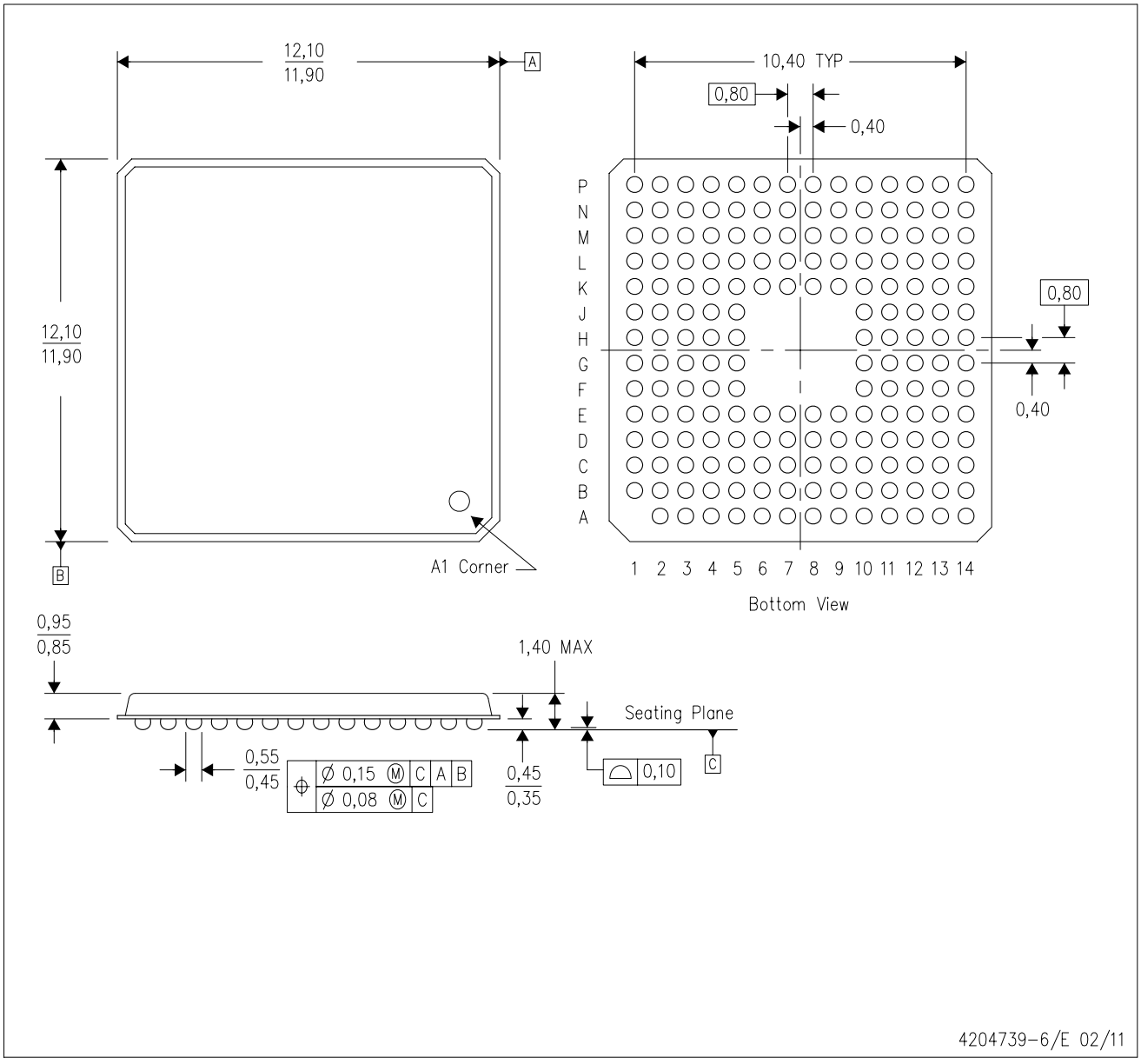
(1) The marketing status values are defined as follows:
**ACTIVE:** Product device recommended for new designs.
**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.
**OBSOLETE:** TI has discontinued the production of the device.

**(2)** Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
**TBD:** The Pb-Free/Green conversion plan has not been defined.
**Pb-Free (RoHS):** TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
**Pb-Free (RoHS Exempt):** This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
**Green (RoHS & no Sb/Br):** TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

**(3)** MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

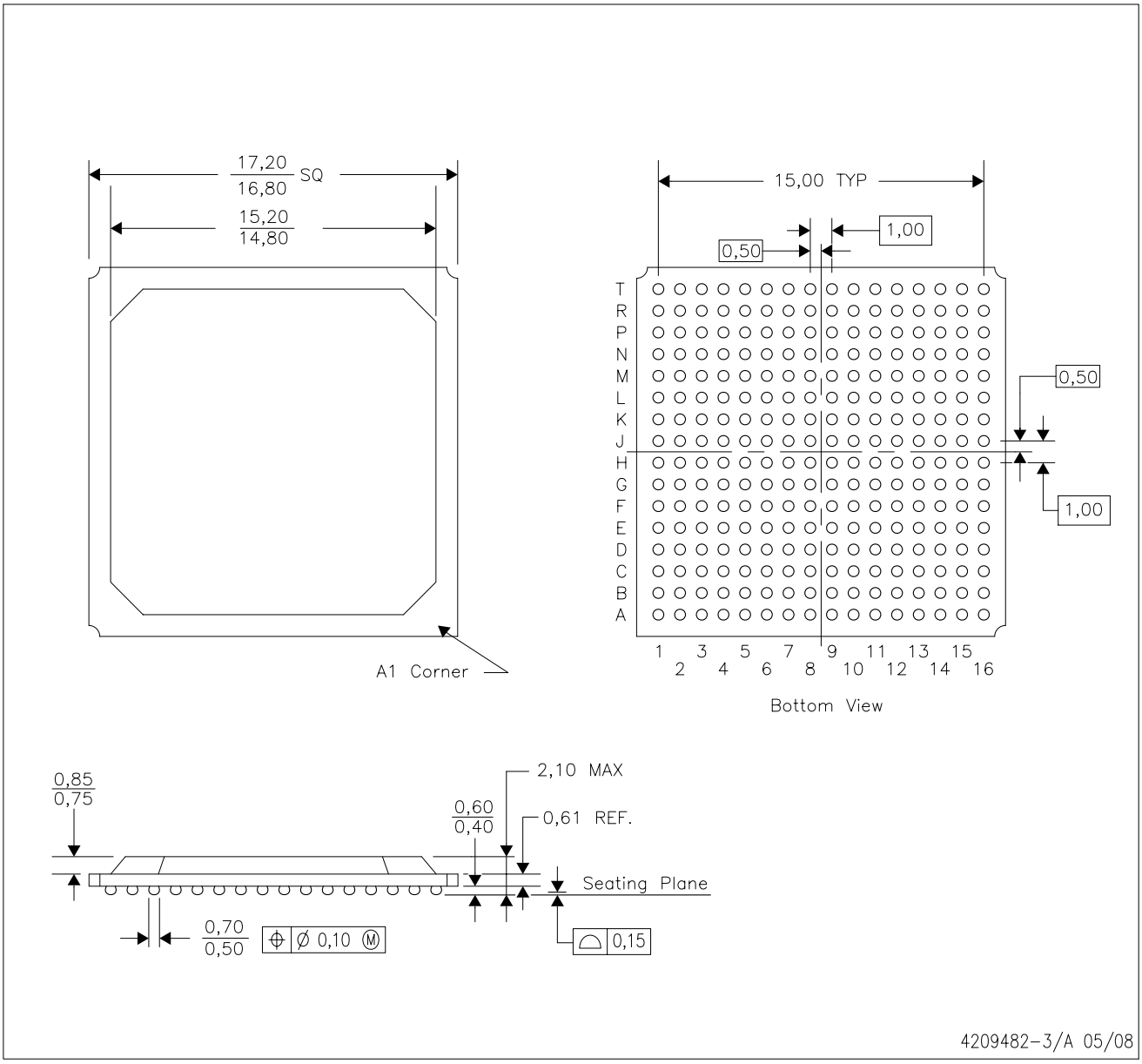
ZHH (S–PBGA–N179) PLASTIC BALL GRID ARRAY




NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar BGA configuration.
D. This is a Pb–free solder ball design.

**MicroStar is a trademark of Texas Instruments.**

ZFE (S–PBGA–N256) PLASTIC BALL GRID ARRAY




NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package.
D. This is a lead–free solder ball design.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

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

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