TMS320x 2834x Delfino Direct Memory Access (DMA) Module

Reference Guide

Literature Number: SPRUG78A
March 2009–Revised September 2009

List of Figures

List of Tables

Read This First

This document describes the TMS320x2834x Delfino™ Direct Memory Access (DMA) Module.

The DMA module described in this reference guide is a Type 0 DMA. See the *TMS320C28xx, 28xxx DSP Peripheral Reference Guide* (SPRU566) for a list of all devices with a DMA module of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h or with a leading 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Docs

The following documents support the 2834x Delfino™ device and can be downloaded from the Texas Instruments web site: www.ti.com.

Data Manual—

SPRS516 — TMS320C28346, TMS320C28345, TMS320C28344, TMS320C28343, TMS320C28342, TMS320C28341 Delfino Microcontrollers Data Manual. This document contains the pinout, signal descriptions, as well as electrical and timing specifications for the C2834x devices.

SPRZ267 — TMS320C2834x Delfino MCU Silicon Errata. This document describes the advisories and usage notes for different versions of silicon.

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—

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.

TMS320x2833x Direct Memory Access (DMA) Module

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

1 Introduction

The strength of a microcontroller (MCU) is not measured purely in processor speed, but in total system capabilities. As a part of the equation, any time the CPU bandwidth for a given function can be reduced, the greater the system capabilities. Many times applications spend a significant amount of their bandwidth moving data, whether it is from off-chip memory to on-chip memory, or from a peripheral such as a McBSP to RAM, or even from one peripheral to another. Furthermore, many times this data comes in a format that is not conducive to the optimum processing powers of the CPU. The DMA module described in this reference guide has the ability to free up CPU bandwidth and rearrange the data into a pattern for more streamlined processing.

The DMA module is an event-based machine, meaning it requires a peripheral interrupt trigger to start a DMA transfer. Although it can be made into a periodic time-driven machine by configuring a timer as the interrupt trigger source, there is no mechanism within the module itself to start memory transfers periodically. The interrupt trigger source for each of the six DMA channels can be configured separately and each channel contains its own independent PIE interrupt to let the CPU know when a DMA transfers has either started or completed. Five of the six channels are exactly the same, while Channel 1 has one additional feature: the ability to be configured at a higher priority than the others. At the heart of the DMA is a state machine and tightly coupled address control logic. It is this address control logic that allows for rearrangement of the block of data during the transfer as well as the process of *ping-ponging* data between buffers. Each of these features, along with others will be discussed in detail in this document.

DMA Overview:

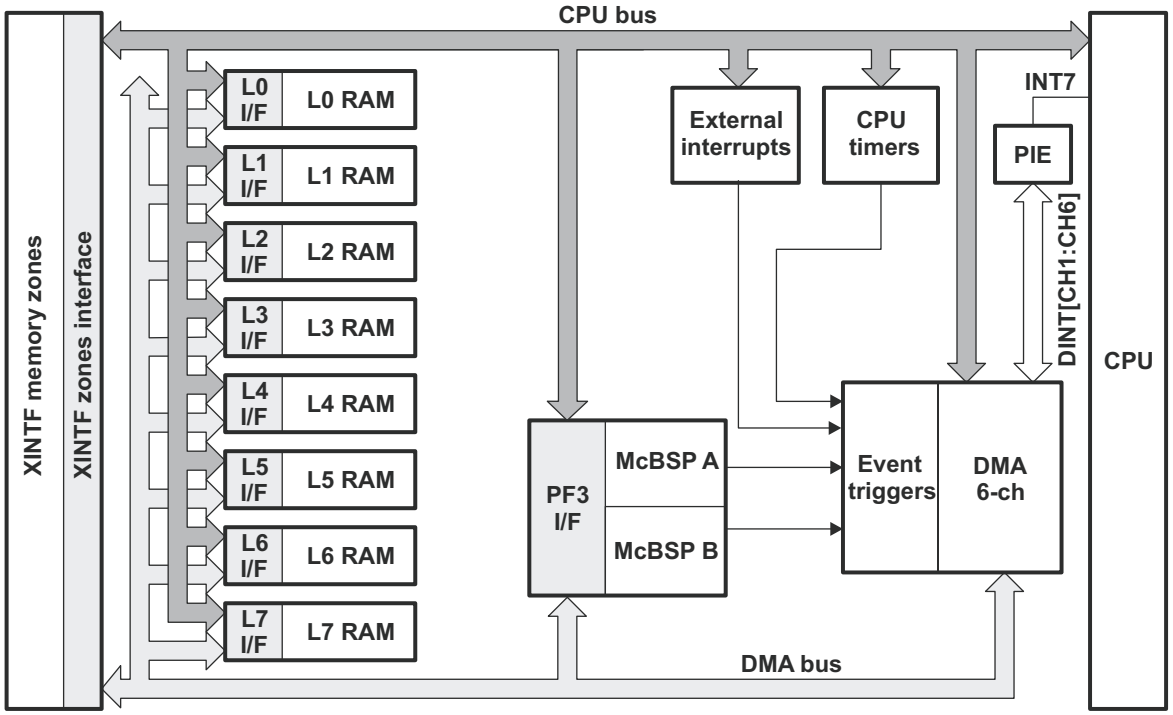
- 6 channels with independent PIE interrupts
- Peripheral interrupt trigger sources
  - Multichannel Buffered Serial Port A and B (McBSP-A, McBSP-B) transmit and receive
  - XINT1-7 and XINT13
  - CPU Timers
  - Software
- Data sources/destinations:
  - L0-L7 64K x 16 SARAM
  - All XINTF zones
  - McBSP-A 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)

2 Architecture

2.1 Block Diagram

Figure 1 shows a device level block diagram of the DMA.

Figure 1. DMA Block Diagram



2.2 Peripheral Interrupt Event Trigger Sources

The peripheral interrupt event trigger can be independently configured as one of eighteen different sources for each of the six DMA channels. Included in these sources are 8 external interrupt signals which can be connected to most of the general-purpose input/output (GPIO) pins on the device. This adds significant flexibility to the event trigger capabilities. A bit field called PERINTSEL in the MODE register of each channel is used to select that channels interrupt trigger source. An active peripheral interrupt trigger will be latched into the PERINTFLG bit of the CONTROL register, and if the respective interrupt and DMA channel is enabled (see the MODE.CHx[PERINTE] and CONTROL.CHx[RUNSTS] bits), it will be serviced by the DMA channel. Upon receipt of a peripheral interrupt event signal, the DMA will automatically send a clear signal to the interrupt source so that subsequent interrupt events will occur.

Regardless of the value of the MODE.CHx[PERINTSEL] bit field, software can always force a trigger by using the CONTROL.CHx[PERINTFRC] bit. Likewise, software can always clear a pending DMA trigger using the CONTROL.CHx[PERINTCLR] bit.

Once a particular interrupt trigger sets a channel's PERINTFLG bit, the bit stays pending until the priority logic of the state machine starts the burst transfer for that channel. Once the burst transfer starts, the flag is cleared. If a new interrupt trigger is generated while a burst is in progress, the burst will complete before responding to the new interrupt trigger (after proper prioritization). If a third interrupt trigger occurs before the pending interrupt is serviced, an error flag is set in the CONTROL.CHx[OVRFLG] bit. If a peripheral interrupt trigger occurs at the same time as the latched flag is being cleared, the peripheral interrupt trigger has priority and the PERINTFLG will remain set.

Figure 2 shows a diagram of the trigger select circuit. See the MODE.CHx[PERINTSEL] bit field description for the complete list of peripheral interrupt trigger sources.

Figure 2. Peripheral Interrupt Trigger Input Diagram

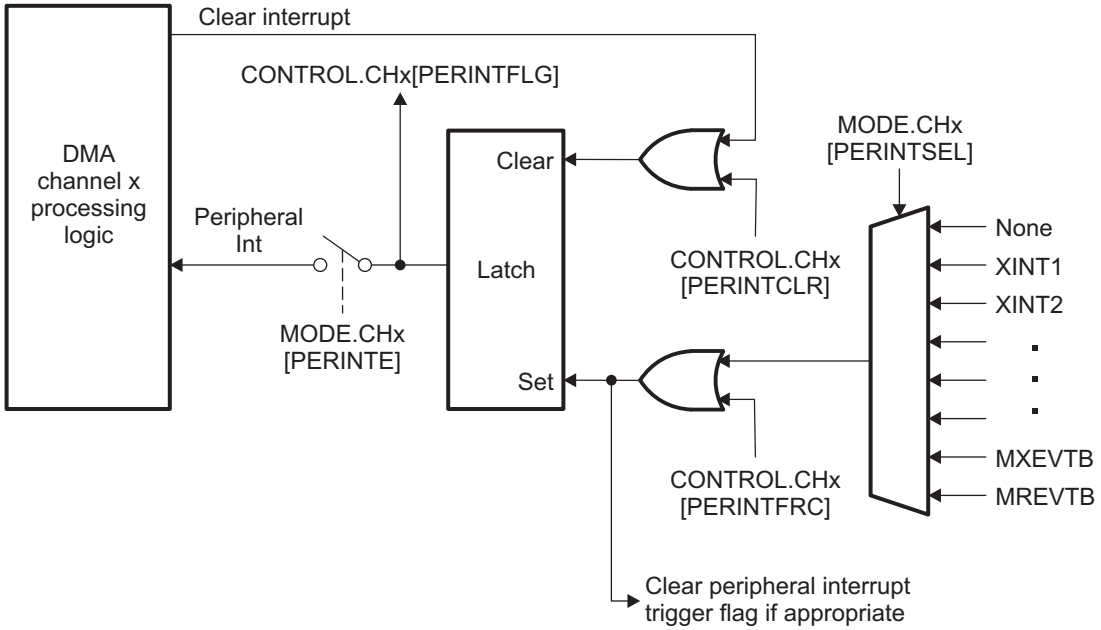


Table 1 shows the interrupt trigger source options that are available for each channel.

Table 1. Peripheral Interrupt Trigger Source Options

Peripheral	Interrupt Trigger Source
CPU	DMA Software bit (CHx.CONTROL.PERINTFRC) only
External Interrupts	External Interrupt 1
	External Interrupt 2
	External Interrupt 3
	External Interrupt 4
	External Interrupt 5
	External Interrupt 6
	External Interrupt 7
	External Interrupt 13
CPU Timers	Timer 0 Overflow
	Timer 1 Overflow
	Timer 2 Overflow
McBSP-A	McBSP-A Transmit Buffer Empty
	McBSP-A Receive Buffer Full
McBSP-B	McBSP-B Transmit Buffer Empty
	McBSP-B Receive Buffer Full

2.3 DMA Bus

The DMA bus architecture consists of a 22-bit address bus, a 32-bit data read bus, and a 32-bit data write bus. Memories and register locations connected to the DMA bus are via interfaces that sometimes share resources with the CPU memory or peripheral bus. Arbitration rules are defined in Section 4. The following resources are connected to the DMA bus:

- XINTF Zones 0, 6 & 7
- L0-L7 SARAM
- McBSP-A and McBSP-B Data Receive Registers (DRR2/DRR1) and Data Transmit Registers (DXR2/DXR1)

3 Pipeline Timing and Throughput

The DMA consists of a 4-stage pipeline as shown in Figure 3. The one exception to this is when a DMA channel is configured to have one of the McBSPs as its data source. A read of a McBSP DRR register stalls the DMA bus for one cycle during the read portion of the transfer, as shown in Figure 4.

Figure 3. 4-Stage Pipeline DMA Transfer

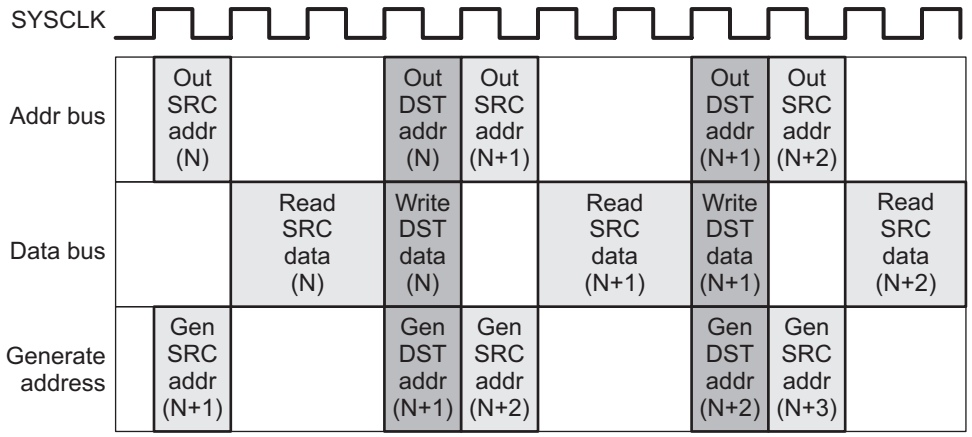
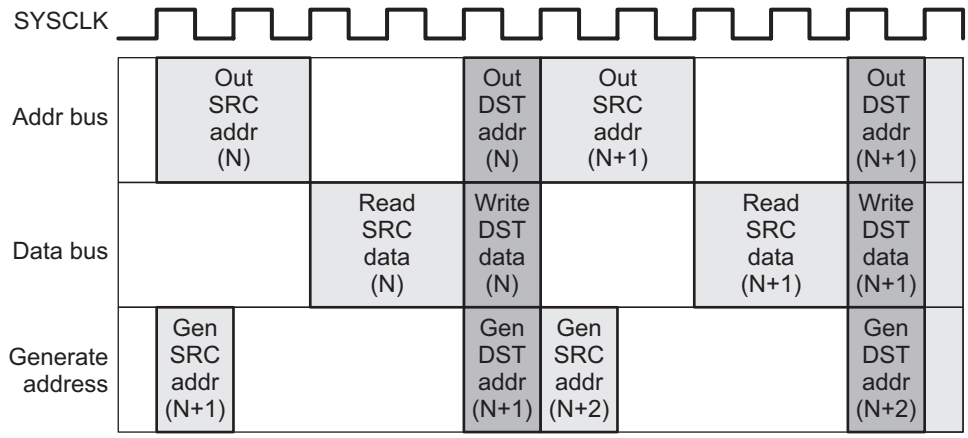


Figure 4. 4-Stage Pipeline With One Read Stall (McBSP as source)



In addition to the pipeline there are a few other behaviors of the DMA that affect it's total throughput

- A 1-cycle delay is added at the beginning of each burst
- A 1-cycle delay is added when returning from a CH1 high priority interrupt
- 32-bit transfers run at double the speed of a 16-bit transfer (i.e., it takes the same amount of time to transfer a 32-bit word as it does a 16-bit word)
- Collisions with the CPU may add delay slots (see Section 4)

For example, to transfer 128 16-bit words from L0 RAM to L1 RAM a channel can be configured to transfer 8 bursts of 16 words/burst. This will give:

8 bursts * [(4 cycles/word * 16 words/burst) + 1] = 520 cycles

If instead the channel were configured to transfer the same amount of data 32 bits at a time (the word size is configured to 32 bits) the transfer would take:

8 bursts * [(4 cycles/word * 8 words/burst) + 1] = 264 cycles

4 CPU Arbitration

Typically, DMA activity is independent of the CPU activity. Under the circumstance where both the DMA and the CPU are attempting to access memory or a peripheral register within the same interface concurrently, an arbitration procedure will occur. Any combined accesses between the different interfaces, or where the CPU access is outside of the interface that the DMA is accessing do not create a conflict.

The interfaces which internally contain conflicts are:

- XINTF Memory Zones 0, 6 and 7
- L0 RAM
- L1 RAM
- L2 RAM
- L3 RAM
- L4 RAM
- L5 RAM
- L6 RAM
- L7 RAM
- Peripheral Frame 3 (McBSP-A, McBSP-B)

4.1 For the External Memory Interface (XINTF) Zones

- If the CPU and the DMA attempt an access to any of the XINTF zones on the same cycle, the DMA is serviced first, followed by all the pending CPU accesses (in the proper priority order for CPU accesses: write → read → fetch).
- If CPU accesses to an XINTF zone are pending or being processed by the XINTF and a DMA access to an XINTF zone is attempted, the DMA access is stalled until all CPU pending accesses are completed. For example, if a CPU write and read access is pending and a fetch is in progress, first the fetch is completed, then the CPU write is performed, then the CPU read is performed, and then the DMA access is performed.
- There is a 1 cycle stall if simultaneous write accesses by the CPU and the DMA are attempted.

If the DMA or CPU is used to write to the XINTF zones, then the write buffer of the XINTF can help to avoid CPU or DMA stalls. If the CPU or DMA are performing reads from XINTF, then significant stalls can occur. The only concern here is if the DMA is stalled and the DMA misses other higher priority DMA events. In such situations, the DMA should not be used to transfer data on XINTF, if the stalls are too long that there is potential to miss other DMA events.

The DMA does not support abort mechanisms for DMA reads from XINTF. If the DMA is performing an access to one of the XINTF zones and the DMA access is stalled (XREADY not responding) then the CPU can issue a HARDRESET that would abort the access. HARDRESET behaves like a System Reset on the DMA. Likewise, a HARDRESET needs to be applied to the XINTF hence releasing the peripheral from the struck ready condition. Any data that is write buffered or pending on the XINTF or DMA will be lost.

4.2 For All Other Peripherals/Memories

- If the CPU and the DMA make an access to the same interface in the same cycle, the DMA has priority and the CPU is stalled.
- If a CPU access to an interface is in progress and another CPU access to the same interface is pending, for example, the CPU is performing a write operation and a read operation from the CPU is pending, then a DMA access to that same interface has priority over the pending CPU access when the current CPU access completes.

NOTE: If the CPU is performing a read-modify-write operation and the DMA performs a write to the same location, the DMA write may be lost if the operation occurs in between the CPU read and the CPU write. For this reason, it is advised not to mix such CPU accesses with DMA accesses to the same locations.

In the case of RAM, a ping-pong scheme can be implemented to avoid the CPU and the DMA accessing the same RAM block concurrently, thus avoiding any stalls or corruption issues.

5 Channel Priority

Two priority schemes exist when determining channel priority: Round-robin mode and Channel 1 high-priority mode.

5.1 Round-Robin Mode

In this mode, all channels have *equal* priority and each enabled channel is serviced in round-robin fashion as follows:

CH1 → CH2 → CH3 → CH4 → CH5 → CH6 → CH1 → CH2 → …

In the case above, after each channel has transferred a burst of words, the next channel is serviced. You can specify the size of the burst for each channel. Once CH6 (or the last enabled channel) has been serviced, and no other channels are pending, the round-robin state machine enters an idle state.

From the idle state, channel 1 (if enabled) is always serviced first. However, if the DMA is currently processing another channel x, all other pending channels between x and the end of the round are serviced before CH1. It is in this sense that all the channels are of *equal* priority. For instance, take an example where CH1, CH4, and CH5 are enabled in round-robin mode and CH4 is currently being processed. Then CH1 and CH5 both receive an interrupt trigger from their respective peripherals before CH4 completes. CH1 and CH5 are now both pending. When CH4 completes its burst, CH5 will be serviced next. Only after CH5 completes will CH1 be serviced. Upon completion of CH1, if there are no more channels pending, the round-robin state machine will enter an idle state.

A more complicated example is shown below:

- Assume all channels are enabled, and the DMA is in an idle state,
- Initially a trigger occurs on CH1, CH3, and CH5 on the same cycle,
- When the CH1 burst transfer starts, requests from CH3 and CH5 are pending,
- Before completion of the CH1 burst, the DMA receives a request from CH2. Now the pending requests are from CH2, CH3, and CH5,
- After completing the CH1 burst, CH2 will be serviced since it is next in the round-robin scheme after CH1.
- After the burst from CH2 is finished, the CH3 burst will be serviced, followed by CH5 burst.
- Now while the CH5 burst is being serviced, the DMA receives a request from CH1, CH3, and CH6.

- The burst from CH6 will start after the completion of the CH5 burst since it is the next channel after CH5 in the round-robin scheme.
- This will be followed by the CH1 burst and then the CH3 burst
- After the CH3 burst finishes, assuming no more triggers have occurred, the round-robin state machine will enter an idle state.

The round-robin state machine may be reset to the idle state via the DMACTRL[PRIORITYRESET] bit.

5.2 Channel 1 High Priority Mode

In this mode, if a CH1 trigger occurs, the current word transfer on any other channel is completed (not the complete burst), execution is halted, and CH1 is serviced for the complete burst count. When the CH1 burst is complete, execution returns to the channel that was active when the CH1 trigger occurred. All other channels have equal priority and each enabled channel is serviced in round-robin fashion as follows:

Higher Priority:	CH1
Lower priority:	CH2 → CH3 → CH4 → CH5 → CH6 → CH2 → …

Given an example where CH1, CH4 and CH5 are enabled in Channel 1 High Priority Mode and CH4 is currently being processed. Then CH1 and CH5 both receive an interrupt trigger from their respective peripherals before CH4 completes. CH1 and CH5 are now both pending. When the current CH4 word transfer is completed, regardless of whether the DMA has completed the entire CH4 burst, CH4 execution will be suspended and CH1 will be serviced. After the CH1 burst completes, CH4 will resume execution.

Upon completion of CH4, CH5 will be serviced. After CH5 completes, if there are no more channels pending, the round-robin state machine will enter an idle state.

Channel 1 High Priority Mode may be used in conjunction with any peripheral.

6 Address Pointer and Transfer Control

The DMA state machine is, at its most basic level, two nested loops. The inner loop transfers a burst of data when a peripheral interrupt trigger is received. A burst is the smallest amount of data that can be transferred at one time and its size is defined by the BURST_SIZE register for each channel. The BURST_SIZE register allows a maximum of 32 sixteen-bit words to be transferred in one burst. The outer loop, whose size is set by the TRANSFER_SIZE register for each channel, defines how many bursts are performed in the entire transfer. Since TRANSFER_SIZE is a 16-bit register, the total size of a transfer allowed is well beyond any practical requirement. One CPU interrupt is generated, if enabled, for each transfer. This interrupt can be configured to occur at the beginning or the end of the transfer via the MODE.CHx[CHINTMODE] bit.

In the default setting of the MODE.CHx[ONESHOT] bit, the DMA transfers one burst of data each time a peripheral interrupt trigger is received. After the burst is completed, the state machine moves on to the next pending channel in the priority scheme, even if another trigger for the channel just completed is pending. This feature keeps any single channel from monopolizing the DMA bus. If a transfer of more than the maximum number of words per burst is desired for a single trigger, the MODE.CHx[ONESHOT] bit can be set to complete the entire transfer when triggered. Care is advised when using this mode, since this can create a condition where one trigger uses up the majority of the DMA bandwidth.

Each DMA channel contains a shadowed address pointer for the source and the destination address. These pointers, SRC_ADDR and DST_ADDR, can be independently controlled during the state machine operation. At the beginning of each transfer, the shadowed version of each pointer is copied into its respective active register. During the burst loop, after each word is transferred, the signed value contained in the appropriate source or destination BURST_STEP register is added to the active SRC/DST_ADDR register. During the transfer loop, after each burst is complete, there are two methods that can be used to modify the active address pointer. The first, and default, method is by adding the signed value contained in the SRC/DST_TRANSFER_STEP register to the appropriate pointer. The second is via a process called wrapping, where a wrap address is loaded into the active address pointer. When a wrap procedure occurs, the associated SRC/DST_TRANSFER_STEP register has no effect.

Address wrapping occurs when a number of bursts specified by the appropriate SRC/DST_WRAP_SIZE register completes. Each DMA channel contains two shadowed wrap address pointers, SRC_BEG_ADDR and DST_BEG_ADDR, allowing the source and destination wrapping to be independent of each other. Like the SRC_ADDR and DST_ADDR registers, the active SRC/DST_BEG_ADDR registers are loaded from their shadow counterpart at the beginning of a transfer. When the specified number of bursts has occurred, a two part wrap procedure takes place:

- The appropriate active SRC/DST_BEG_ADDR register is incremented by the signed value contained in the SRC/DST_WRAP_STEP register, then
- The new active SRC/DST_BEG_ADDR register is loaded into the active SRC/DST_ADDR register.

Additionally the wrap counter (SRC/DST_WRAP_COUNT) register is reloaded with the SRC/DST_WRAP_SIZE value to setup the next wrap period. This allows the channel to wrap multiple times within a single transfer. Combined with the first bullet above, this allows the channel to address multiple buffers within a single transfer.

The DMA contains both an active and shadow set of the following address pointers. When a DMA transfer begins, the shadow register set is copied to the active working set of registers. This allows you to program the values of the shadow registers for the next transfer while the DMA works with the active set. It also allows you to implement Ping-Pong buffer schemes without disrupting the DMA channel execution.

Source/Destination Address Pointers (SRC/DST_ADDR)— The value written into the shadow register is the start address of the first location where data is read or written to.

At the beginning of a transfer the shadow register is copied into the active register. The active register performs as the current address pointer.

Source/Destination Begin Address Pointers (SRC/DST_BEG_ADDR)— This is the wrap pointer.

The value written into the shadow register will be loaded into the active register at the start of a transfer. On a wrap condition, the active register will be incremented by the signed value in the appropriate SRC/DST_WRAP_STEP register prior to being loaded into the active SRC/DST_ADDR register.

For each channel, the transfer process can be controlled with the following size values:

Source and Destination Burst Size (BURST_SIZE): — This specifies the number of words to be transferred in a burst.

This value is loaded into the BURST_COUNT register at the beginning of each burst. The BURST_COUNT decrements each word that is transferred and when it reaches a zero value, the burst is complete, indicating that the next channel can be serviced. The behavior of the current channel is defined by the ONE_SHOT bit in the MODE register. The maximum size of the burst is dictated by the type of peripheral. For a McBSP peripheral, the burst size is limited to 1 since there is no FIFO and the receive or transmit data register must be loaded or copied every word transferred. For RAM and XINTF the burst size can be up to the maximum allowed by the BURST_SIZE register, which is 32.

Source and Destination Transfer Size (TRANSFER_SIZE): — This specifies the number of bursts to be transferred before per CPU interrupt (if enabled).

Whether this interrupt is generated at the beginning or the end of the transfer is defined in the CHINTMODE bit in the MODE register. Whether the channel remains enabled or not after the transfer is completed is defined by the CONTINUOUS bit in the MODE register. The TRANSFER_SIZE register is loaded into the TRANSFER_COUNT register at the beginning of each transfer. The TRANSFER_COUNT register keeps track of how many bursts of data the channel has transferred and when it reaches zero, the DMA transfer is complete.

Source/Destination Wrap Size (SRC/DST_WRAP_SIZE)— This specifies the number of bursts to be transferred before the current address pointer wraps around to the beginning.

This feature is used to implement a circular addressing type function. This value is loaded into the appropriate SRC/DST_WRAP_COUNT register at the beginning of each transfer. The SRC/DST_WRAP_COUNT registers keep track of how many bursts of data the channel has transferred and when they reaches zero, the wrap procedure is performed on the appropriate source or destination address pointer. A separate size and count register is allocated for source and destination pointers. To *disable* the wrap function, assign the value of these registers to be larger than the TRANSFER_SIZE.

NOTE: The value written to the SIZE registers is one less than the intended size. So, to transfer three 16-bit words, the value 2 should be placed in the SIZE register.

Regardless of the state of the DATASIZE bit, the value specified in the SIZE registers are for 16-bit addresses. So, to transfer three 32-bit words, the value 5 should be placed in the SIZE register.

For each source/destination pointer, the address changes can be controlled with the following step values:

Source/Destination Burst Step (SRC/DST_BURST_STEP)— Within each burst transfer, the address source and destination step sizes are specified by these registers.

This value is a signed 2's compliment number so that the address pointer can be incremented or decremented as required. If no increment is desired, such as when accessing the McBSP data receive or transmit registers, the value of these registers should be set to zero.

Source/Destination Transfer Step (SRC/DST_TRANSFER_STEP)— This specifies the address offset to start the next burst transfer after completing the current burst transfer.

This is used in cases where registers or data memory locations are spaced at constant intervals. This value is a signed 2's compliment number so that the address pointer can be incremented or decremented as required.

Source/Destination Wrap Step (SRC/DST_WRAP_STEP): — When the wrap counter reaches zero, this value specifies the number of words to add/subtract from the BEG_ADDR pointer and hence sets the new start address.

This implements a circular type of addressing mode, useful in many applications. This value is a signed 2's compliment number so that the address pointer can be incremented or decremented as required.

NOTE: Regardless of the state of the DATASIZE bit, the value specified in the STEP registers are for 16-bit addresses. So, to increment one 32-bit address, a value of 2 should be placed in these registers.

Three modes are provided to control the way the state machine behaves during the burst loop and the transfer loop:

One Shot Mode (ONESHOT)— If one shot mode is enabled when an interrupt event trigger occurs, the DMA will continue transferring data in bursts until TRANSFER_COUNT is zero. If one shot mode is disabled, then an interrupt event trigger is required for each burst transfer and this will continue until TRANSFER_COUNT is zero.

NOTE: When ONESHOT mode is enabled, the DMA will continuously transfer bursts of data on the given channel until the TRANSFER_COUNT value is zero. This could potentially hog the bandwidth of a peripheral and cause long CPU stalls to occur. To avoid this, you could configure a CPU timer (or similar) and disable ONESHOT so as to avoid this situation.

Continuous Mode (CONTINUOUS)— If continuous mode is disabled the RUNSTS bit in the CONTROL register is cleared at the end of the transfer, disabling the DMA channel.

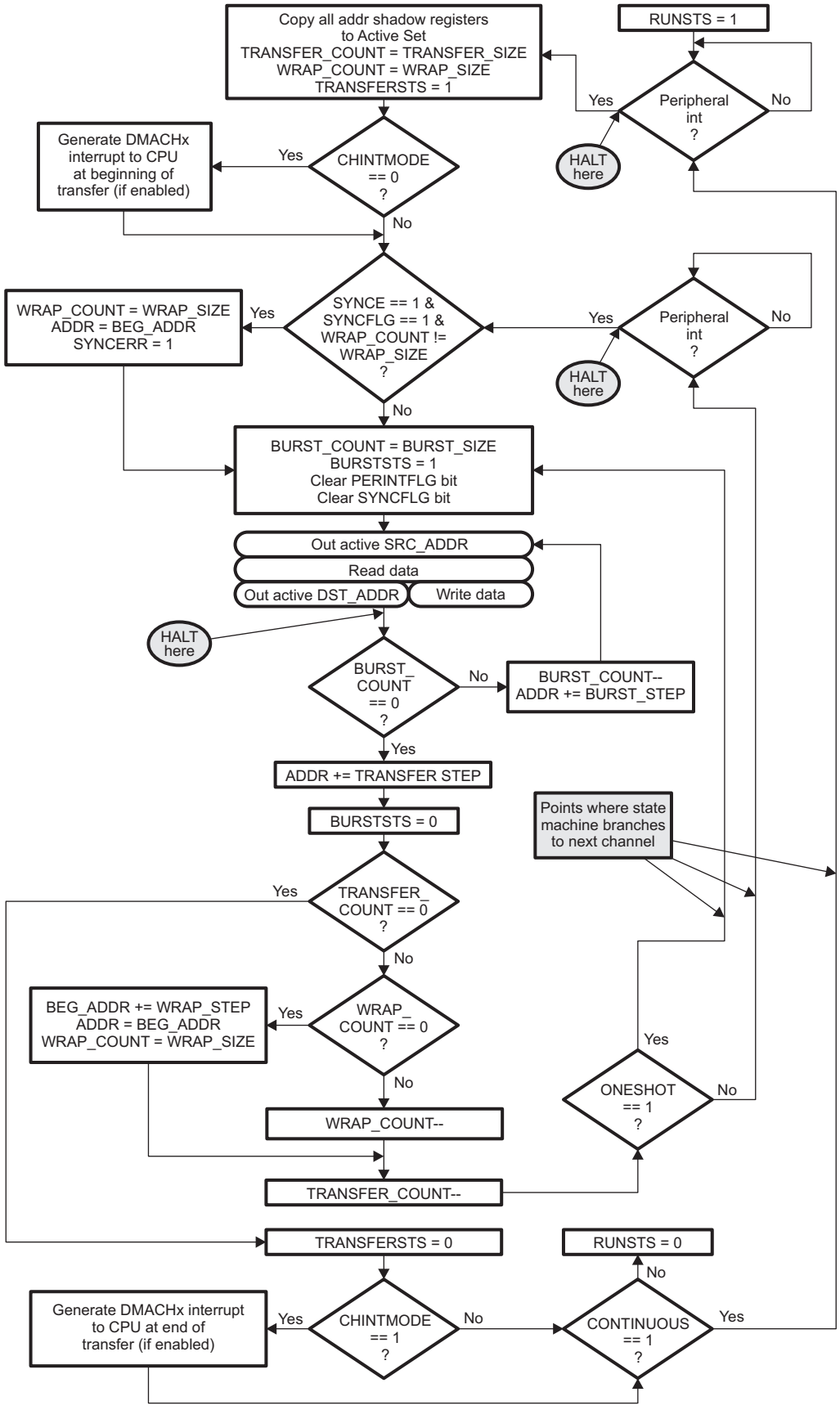
The channel must be re-enabled by setting the RUN bit in the CONTROL register before another transfer can be started on that channel. If the continuous mode is enabled the RUNSTS bit is not cleared at the end of the transfer.

Channel Interrupt Mode (CHINTMODE)— This mode bit selects whether the DMA interrupt from the respective channel is generated at the beginning of a new transfer or at the end of the transfer.

If implementing a ping-pong buffer scheme with continuous mode of operation, then the interrupt would be generated at the beginning, just after the working registers are copied to the shadow set. If the DMA does not operate in continuous mode, then the interrupt is typically generated at the end when the transfer is complete.

All of the above features and modes are shown in Figure 5.

Figure 5. DMA State Diagram



The following items are in reference to Figure 5.

- The *HALT* points represent where the channel halts operation when interrupted by a high priority channel 1 trigger, or when the HALT command is set, or when an emulation halt is issued and the FREE bit is cleared to 0.
- The ADDR registers are not affected by BEG_ADDR at the start of a transfer. BEG_ADDR only affects the ADDR registers on a wrap or sync error. Following is what happens to each of the ADDR registers when a transfer first starts:
  - BEG_ADDR_SHADOW remains unchanged.
  - ADDR_SHADOW remains unchanged.
  - BEG_ADDR = BEG_ADDR_SHADOW
  - ADDR = ADDR_SHADOW
- The active registers get updated when a wrap occurs. The shadow registers remain unchanged. Specifically:
  - BEG_ADDR_SHADOW remains unchanged.
  - ADDR_SHADOW remains unchanged.
  - BEG_ADDR += WRAP_STEP
  - ADDR = BEG_ADDR
- The active registers get updated when a sync error occurs. The shadow registers remain unchanged. Specifically:
  - BEG_ADDR_SHADOW remains unchanged.
  - ADDR_SHADOW remains unchanged.
  - BEG_ADDR remains unchanged.
  - ADDR = BEG_ADDR

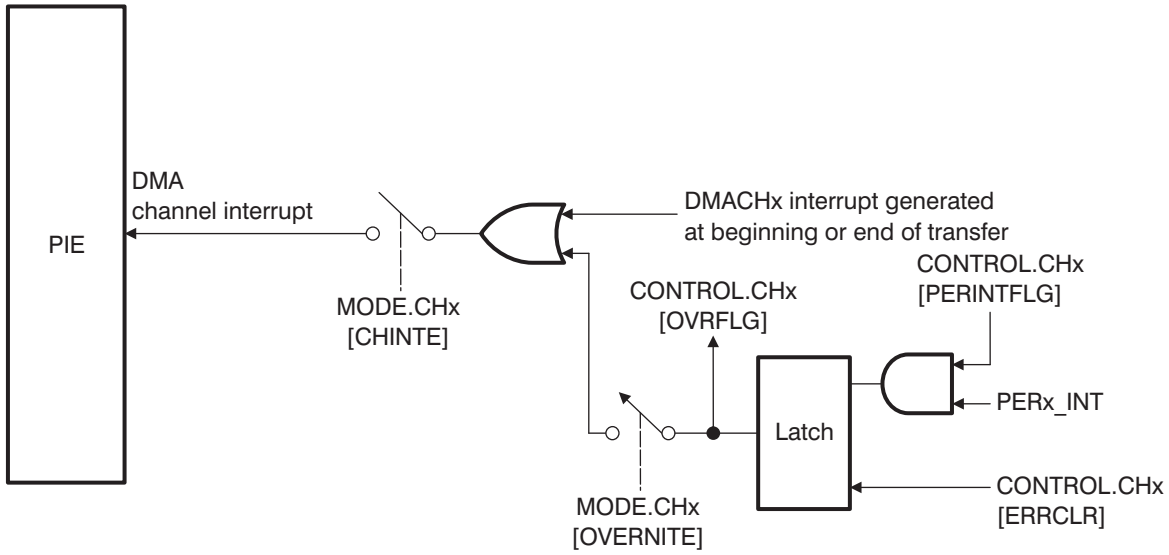
Probably the easiest way to remember all this is that:

- The shadow registers never change except by software.
- The active registers never change except by hardware, and a shadow register is only copied into its own active register, never an active register by another name.

7 Overrun Detection Feature

The DMA contains overrun detection logic. When a peripheral event trigger is received by the DMA, the PERINTFLG bit in the CONTROL register is set, pending the channel to the DMA state machine. When the burst for that channel is started, the PERINTFLG is cleared. If however, between the time that the PERINTFLG bit is set by an event trigger and cleared by the start of the burst, an additional event trigger arrives, the second trigger will be lost. This condition will set the OVRFLG bit in the CONTROL register as in Figure 6. If the overrun interrupt is enabled then the channel interrupt will be generated to the PIE module.

Figure 6. Overrun Detection Logic



8 Register Descriptions

The complete DMA register set is shown in Table 2.

Table 2. DMA Register Summary[1]

Address	Acronym	Description	Section
DMA Control, Mode and Status Registers			
0x1000	DMACTRL	DMA Control Register	Section 8.1
0x1001	DEBUGCTRL	Debug Control Register	Section 8.2
0x1002	REVISION	Peripheral Revision Register	Section 8.3
0x1003	Reserved	Reserved	
0x1004	PRIORITYCTRL1	Priority Control Register 1	Section 8.4
0x1005	Reserved	Reserved	
0x1006	PRIORITYSTAT	Priority Status Register	Table 7
0x1007 0x101F	Reserved	Reserved	
DMA Channel 1 Registers			
0x1020	MODE	Mode Register	Section 8.6
0x1021	CONTROL	Control Register	Section 8.7
0x1022	BURST_SIZE	Burst Size Register	Section 8.8
0x1023	BURST_COUNT	Burst Count Register	Section 8.9
0x1024	SRC_BURST_STEP	Source Burst Step Size Register	Section 8.10
0x1025	DST_BURST_STEP	Destination Burst Step Size Register	Section 8.11
0x1026	TRANSFER_SIZE	Transfer Size Register	Table 13
0x1027	TRANSFER_COUNT	Transfer Count Register	Section 8.13
0x1028	SRC_TRANSFER_STEP	Source Transfer Step Size Register	Section 8.14
0x1029	DST_TRANSFER_STEP	Destination Transfer Step Size Register	Section 8.15
0x102A	SRC_WRAP_SIZE	Source Wrap Size Register	Section 8.16
0x102B	SRC_WRAP_COUNT	Source Wrap Count Register	Section 8.17
0x102C	SRC_WRAP_STEP	Source Wrap Step Size Register	Section 8.18
0x102D	DST_WRAP_SIZE	Destination Wrap Size Register	Section 8.16
0x102E	DST_WRAP_COUNT	Destination Wrap Count Register	Section 8.17
0x102F	DST_WRAP_STEP	Destination Wrap Step Size Register	Section 8.18
0x1030	SRC_BEG_ADDR_SHADOW	Shadow Source Begin and Current Address Pointer Registers	Section 8.19
0x1032	SRC_ADDR_SHADOW		Section 8.19
0x1034	SRC_BEG_ADDR	Active Source Begin and Current Address Pointer Registers	Section 8.20
0x1036	SRC_ADDR		Section 8.20
0x1038	DST_BEG_ADDR_SHADOW	Shadow Destination Begin and Current Address Pointer Registers	Section 8.21
0x103A	DST_ADDR_SHADOW		Section 8.21
0x103C	DST_BEG_ADDR	Active Destination Begin and Current Address Pointer Registers	Section 8.22
0x103E	DST_ADDR		Section 8.22
0x103F	Reserved	Reserved	
DMA Channel 2 Registers			
0x1040 0x105F	Same as above		
DMA Channel 3 Registers			
0x1060 0x107F	Same as above		
DMA Channel 4 Registers			

(1) All DMA register writes are EALLOW protected.

Table 2. DMA Register Summary[(1)] (continued)

Address	Acronym	Description	Section
0x1080 0x109F	Same as above		
DMA Channel 5 Registers			
0x10A0 0x10BF	Same as above		
DMA Channel 6 Registers			
0x10C0 0x10DF	Same as above		

8.1 DMA Control Register (DMACTRL) — EALLOW Protected

The DMA control register (DMACTRL) is shown in Figure 7 and described in Table 3.

Figure 7. DMA Control Register (DMACTRL)

15	8
Reserved	
R-0	

7	2	1	0
Reserved		PRIORITY RESET	HARD RESET
R-0		R0/S-0	R0/S-0

LEGEND: R0/S = Read 0/Set; R = Read only; *-n* = value after reset

Table 3. DMA Control Register (DMACTRL) Field Descriptions

Bit	Field	Value	Description
15-2	Reserved	0	Reserved
1	PRIORITYRESET	0	The priority reset bit resets the round-robin state machine when a 1 is written. Service starts from the first enabled channel. Writes of 0 are ignored and this bit always reads back a 0. When a 1 is written to this bit, any pending burst transfer completes before resetting the channel priority machine. If CH1 is configured as a high priority channel, and this bit is written to while CH1 is servicing a burst, the CH1 burst is completed and then any lower priority channel burst is also completed (if CH1 interrupted in the middle of a burst), before the state machine is reset. In case CH1 is high priority, the *state machine* restarts from CH2 (or the next highest enabled channel).
0	HARDRESET	0	Writing a 1 to the hard reset bit resets the whole DMA and aborts any current access (similar to applying a device reset). Writes of 0 are ignored and this bit always reads back a 0. For a *soft* reset, a bit is provided for each channel to perform a gentler reset. Refer to the channel control registers. If the DMA was performing an access to the XINTF and the DMA access was stalled (XREADY not responding), then a HARDRESET would abort the access. The XINTF access would only complete if XREADY is released. When writing to this bit, there is a one cycle delay before it takes effect. Hence at least a one cycle delay (i.e., a NOP instruction) after writing to this bit should be introduced before attempting an access to any other DMA register.

8.2 Debug Control Register (DEBUGCTRL) — EALLOW Protected

The debug control register (DEBUGCTRL) is shown in Figure 8 and described in Table 4.

Figure 8. Debug Control Register (DEBUGCTRL)

15	14	0
FREE	Reserved	
R/W-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 4. Debug Control Register (DEBUGCTRL) Field Descriptions

Bit	Field	Value	Description
15	FREE		Emulation Control Bit: This bit specifies the action when an emulation halt event occurs.
		0	DMA runs until the current DMA read-write access is completed and the current status of a DMA is frozen. See the *HALT* points in Figure 5 for possible halt states.
		1	DMA is unaffected by emulation suspend (run free)
14-0	Reserved	0	Reserved

8.3 Revision Register (REVISION)

The revision register (REVISION) is shown in Figure 9 and described in Table 5.

Figure 9. Revision Register (REVISION)

15	8	7	0
TYPE		REV	
R		R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 5. Revision Register (REVISION) Field Descriptions

Bit	Field	Value	Description
15-8	TYPE		DMA Type Bits. A type change represents a major functional feature difference in a peripheral module. Within a peripheral type, there may be minor differences between devices which do not affect the basic functionality of the module. These device-specific differences are listed in the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (SPRU566).
		0x0000	This document describes a Type0 DMA.
7-0	REV		DMA Silicon Revision Bits: These bits specify the DMA revision and are changed if any bug fixes are performed.
		0x0000	First release

8.4 Priority Control Register 1 (PRIORITYCTRL1) — EALLOW Protected

The priority control register 1 (PRIORITYCTRL1) is shown in Figure 10 and described in Table 6.

Figure 10. Priority Control Register 1 (PRIORITYCTRL1)

15	1	0
Reserved		CH1 PRIORITY
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6. Priority Control Register 1 (PRIORITYCTRL1) Field Descriptions

Bit	Field	Value	Description
15-1	Reserved	0	Reserved
0	CH1PRIORITY		DMA Ch1 Priority: This bit selects whether channel 1 has higher priority or not:
		0	Same priority as all other channels
		1	Highest priority channel
			Channel priority can only be changed when all channels are disabled. A priority reset should be performed before restarting channels after changing priority.

8.5 Priority Status Register (PRIORITYSTAT)

The priority status register (PRIORITYSTAT) is shown in Figure 11 and described in Table 7.

Figure 11. Priority Status Register (PRIORITYSTAT)

15	8
Reserved	
R-0	

7	6	4	3	2	0
Reserved	ACTIVESTS_SHADOW		Reserved	ACTIVESTS	
R-0	R-0		R-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 7. Priority Status Register (PRIORITYSTAT) Field Descriptions

Bit	Field	Value	Description
15-7	Reserved	0	Reserved
6-4	ACTIVESTS_SHADOW		Active Channel Status Shadow Bits: These bits are only useful when CH1 is enabled as a higher priority channel. When CH1 is serviced, the ACTIVESTS bits are copied to the shadow bits and indicate which channel was interrupted by CH1. When CH1 service is completed, the shadow bits are copied back to the ACTIVESTS bits. If this bit field is zero or the same as the ACTIVESTS bit field, then no channel is pending due to a CH1 interrupt. When CH1 is not a higher priority channel, these bits should be ignored:
		0,0,0	No channel pending
		0,0,1	CH 1
		0,1,0	CH 2
		0,1,1	CH 3
		1,0,0	CH 4
		1,0,1	CH 5
		1,1,0	CH 6
3	Reserved		Reserved
2-0	ACTIVESTS		Active Channel Status Bits: These bits indicate which channel is currently active or performing a transfer:
		0,0,0	no channel active
		0,0,1	CH 1
		0,1,0	CH 2
		0,1,1	CH 3
		1,0,0	CH 4
		1,0,1	CH 5
		1,1,0	CH 6

8.6 Mode Register (MODE) — EALLOW Protected

The mode register (MODE) is shown in Figure 12 and described in Table 8.

Figure 12. Mode Register (MODE)

15	14	13	12	11	10	9	8
CHINTE	DATASIZE	SYNCSEL	SYNCE	CONTINUOUS	ONESHOT	CHINTMODE	PERINTE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4				0
OVRINTE	Reserved		PERINTSEL				
R/W-0	R-0		R/W-0				

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 8. Mode Register (MODE) Field Descriptions

Bit	Field	Value	Description
15	CHINTE		Channel Interrupt Enable Bit: This bit enables/disables the respective DMA channel interrupt to the CPU (via the PIE).
		0	Interrupt disabled
		1	Interrupt enabled
14	DATASIZE		Data Size Mode Bit: This bit selects if the DMA channel transfers 16-bits or 32-bits of data at a time.
		0	16-bit data transfer size
		1	32-bit data transfer size
			NOTE: Regardless of the value of this bit all of the registers in the DMA refer to 16-bit words. The only effect this bit causes is whether the databus width is 16 or 32 bits. It is up to you to configure the pointer step increment and size to accommodate 32-bit data transfers. See section Section 6 for details.
13	SYNCSEL		Sync Mode Select Bit: Not used in the family of devices
12	SYNCE		Sync Enable Bit: Not used in the family of devices.
11	CONTINUOUS		Continuous Mode Bit: If this bit is set to 1, then DMA re-initializes when TRANSFER_COUNT is zero and waits for the next interrupt event trigger. If this bit is 0, then the DMA stops and clears the RUNSTS bit to 0.
10	ONESHOT		One Shot Mode Bit: If this bit is set to 1, then subsequent burst transfers occur without additional event triggers after the first event trigger. If this bit is 0 then only one burst transfer is performed per event trigger.
9	CHINTMODE		Channel Interrupt Generation Mode Bit: This bit specifies when the respective DMA channel interrupt should be generated to the CPU (via the PIE).
		0	Generate interrupt at beginning of new transfer
		1	Generate interrupt at end of transfer.
8	PERINTE		Peripheral Interrupt Trigger Enable Bit: This bit enables/disables the selected peripheral interrupt trigger to the DMA.
		0	Interrupt trigger disabled. Neither the selected peripheral nor software can start a DMA burst.
		1	Interrupt trigger enabled.
7	OVRINTE		Overflow Interrupt Enable: This bit when set to 1 enables the DMA to generate an interrupt when an overflow event is detected.
		0	Overflow interrupt disabled
		1	Overflow interrupt enabled
			An overflow interrupt is generated when the PERINTFLG is set and another interrupt event occurs. The PERINTFLG being set indicates a previous peripheral event is latched and has not been serviced by the DMA.

Table 8. Mode Register (MODE) Field Descriptions (continued)

Bit	Field	Value	Description		
6-5	Reserved	0	Reserved		
4-0	PERINTSEL		Peripheral Interrupt Source Select Bits: These bits select which interrupt triggers a DMA burst for the given channel. Only one interrupt source can be selected. A DMA burst can also be forced via the PERINTFRC bit.		
		Value	**Interrupt**	**Sync**	**Peripheral**
		2:0	None	None	No Peripheral Connection
		3	XINT1	None	External Interrupts
		4	XINT2	None	
		5	XINT3	None	
		6	XINT4	None	
		7	XINT5	None	
		8	XINT6	None	
		9	XINT7	None	
		10	XINT13	None	
		11	TINT0	None	CPU Timers
		12	TINT1	None	
		13	TINT2	None	
		14	MXEVTA	None	McBSP-A
		15	MREVTA	None	
		16	MXEVTB	None	McBSP-B
		17	MREVTB	None	
		31: 18	Reserved		No peripheral connection

8.7 Control Register (CONTROL) — EALLOW Protected

The control register (CONTROL) is shown in Figure 13 and described in Table 9.

Figure 13. Control Register (CONTROL)

15	14	13	12	11	10	9	8
Reserved	OVRFLG	RUNSTS	BURSTSTS	TRANSFERSTS	SYNCERR	SYNCFLG	PERINTFLG
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
ERRCLR	SYNCCLR	SYNCFRC	PERINTCLR	PERINTFRC	SOFTRESET	HALT	RUN
R0/S-0	R0/S-0	R0/S-0	R0/S-0	R0/S-0	R0/S-0	R0/S-0	R0/S-0

LEGEND: R0/S = Read 0/Set; R = Read only; -*n* = value after reset

Table 9. Control Register (CONTROL) Field Descriptions

Bit	Field	Value	Description
15	Reserved	0	Reserved
14	OVRFLG		Overflow Flag Bit: This bit indicates if a peripheral interrupt event trigger is received from the selected peripheral and the PERINTFLG is already set.
		0	No overflow event
		1	Overflow event
			The ERRCLR bit can be used to clear the state of this bit to 0. The OVRFLG bit is not affected by the PERINTFRC event.
13	RUNSTS		Run Status Bit: This bit is set to 1 when the RUN bit is written to with a 1. This indicates the DMA channel is now ready to process peripheral interrupt event triggers. This bit is cleared to 0 when TRANSFER_COUNT reaches zero and CONTINUOUS mode bit is set to 0. This bit is also cleared to 0 when either the HARDRESET bit, the SOFTRESET bit, or the HALT bit is activated.
		0	Chanel is disabled.
		1	Channel is enabled.
12	BURSTSTS		Burst Status Bit: This bit is set to 1 when a DMA burst transfer begins and the BURST_COUNT is initialized with the BURST_SIZE. This bit is cleared to zero when BURST_COUNT reaches zero. This bit is also cleared to 0 when either the HARDRESET or the SOFTRESET bit is activated.
		0	No burst activity
		1	The DMA is currently servicing or suspending a burst transfer from this channel.
11	TRANSFERSTS		Transfer Status Bit: This bit is set to 1 when a DMA transfer begins and the address registers are copied to the shadow set and the TRANSFER_COUNT is initialized with the TRANSFER_SIZE. This bit is cleared to zero when TRANSFER_COUNT reaches zero. This bit is also cleared to 0 when either the HARDRESET or the SOFTRESET bit is activated.
		0	No transfer activity
		1	The channel is currently in the middle of a transfer regardless of whether a burst of data is actively being transferred or not.
10	SYNCERR		Sync ERR Bit: Not used in the family of devices.
9	SYNCFLG		Sync Flag Bit: Not used in the family of devices.
8	PERINTFLG		Peripheral Interrupt Trigger Flag Bit: This bit indicates if a peripheral interrupt event trigger has occurred. This flag is automatically cleared when the first burst transfer begins.
		0	No interrupt event trigger
		1	Interrupt event trigger
			The PERINTFRC bit can be used to set the state of this bit to 1 and force a software DMA event. The PERINTCLR bit can be used to clear the state of this bit to 0.
7	ERRCLR	0	Error Clear Bit: Writing a 1 to this bit will clear the OVRFLG bit. This bit would normally be used when initializing the DMA for the first time or if an overflow condition is detected. If an overflow event occurs at the same time as writing to this bit, the overrun has priority and the OVRFLG bit is set.
6	SYNCCLR	0	Sync Clear Bit: Not used in the family of devices.
5	SYNCFRC	0	Sync Force Bit: Not used in the family of devices.

Table 9. Control Register (CONTROL) Field Descriptions (continued)

Bit	Field	Value	Description
4	PERINTCLR	0	Peripheral Interrupt Clear Bit: Writing a 1 to this bit clears any latched peripheral interrupt event and clears the PERINTFLG bit. This bit would normally be used when initializing the DMA for the first time. If a peripheral event occurs at the same time as writing to this bit, the peripheral has priority and the PERINTFLG bit is set.
3	PERINTFRC	0	Peripheral Interrupt Force Bit: Writing a 1 to this bit latches a peripheral interrupt event trigger and sets the PERINTFLG bit. If the PERINTE bit is set, this bit can be used like a software force for a DMA burst transfer.
2	SOFTRESET	0	Channel Soft Reset Bit: Writing a 1 to this bit completes current read-write access and places the channel into a default state as follows: RUNSTS = 0 TRANSFERSTS = 0 BURSTSTS = 0 BURST_COUNT = 0 TRANSFER_COUNT = 0 SRC_WRAP_COUNT = 0 DST_WRAP_COUNT = 0 This is a *soft* reset that basically allows the DMA to complete the current read-write access and then places the DMA channel into the default reset state.
1	HALT	0	Channel Halt Bit: Writing a 1 to this bit halts the DMA at the current state and any current read-write access is completed. See Figure 5 for the various positions the state machine can be at when HALTED. The RUNSTS bit is set to 0. To take the device out of HALT, the RUN bit needs to be activated.
0	RUN	0	Channel Run Bit: Writing a 1 to this bit starts the DMA channel. The RUNSTS bit is set to 1. This bit is also used to take the device out of HALT. The RUN bit is typically used to start the DMA running after you have configured the DMA. It will then wait for the first interrupt event (PERINTFLG == 1) to start operation. The RUN bit can also be used to take the DMA channel out of a HALT condition See Figure 5 for the various positions the state machine can be at when HALTED.

8.8 Burst Size Register (BURST_SIZE) — EALLOW Protected

The burst size register (BURST_SIZE) is shown in Figure 14 and described in Table 10.

Figure 14. Burst Size Register (BURST_SIZE)

15	5	4	0
Reserved		BURSTSIZE	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10. Burst Size Register (BURST_SIZE) Field Descriptions

Bit	Field	Value	Description
15-5	Reserved	0	Reserved
4-0	BURSTSIZE		These bits specify the burst transfer size:
		0	Transfer 1 word in a burst
		1	Transfer 2 words in a burst
		...	...
		31	Transfer 32 words in a burst

8.9 BURST_COUNT Register

The burst count register (BURST_COUNT) is shown in Figure 15 and described in Table 11.

Figure 15. Burst Count Register (BURST_COUNT)

15	5	4	0
Reserved		BURSTCOUNT	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 11. Burst Count Register (BURST_COUNT) Field Descriptions

Bit	Field	Value	Description
15-5	Reserved	0	Reserved
4-0	BURSTCOUNT		These bits indicate the current burst counter value:
		0	0 word left in a burst
		1	1 word left in a burst
		2	2 words left in a burst
		...	...
		31	31 words left in a burst
			The above values represent the state of the counter at the HALT conditions.

8.10 Source Burst Step Register Size (SRC_BURST_STEP) — EALLOW Protected

The source burst step size register (SRC_BURST_STEP) is shown in Figure 16 and described in Table 12.

Figure 16. Source Burst Step Size Register (SRC_BURST_STEP)

15	0
SRCBURSTSTEP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 12. Source Burst Step Size Register (SRC_BURST_STEP) Field Descriptions

Bit	Field	Value	Description
15-0	SRCBURSTSTEP		These bits specify the source address post-increment/decrement step size while processing a burst of data:
		0x0FFF	Add 4095 to address
		...	...
		0x0002	Add 2 to address
		0x0001	Add 1 to address
		0x0000	No address change
		0xFFFF	Sub 1 from address
		0xFFFE	Sub 2 from address
		...	...
		0xF000	Sub 4096 from address
			Only values from -4096 to 4095 are valid.

8.11 Destination Burst Step Register Size (DST_BURST_STEP) — EALLOW Protected

The destination burst step register size (DST_BURST_STEP) is shown in Figure 17 and described in Table 13.

Figure 17. Destination Burst Step Register Size (DST_BURST_STEP)

15	0
DSTBURSTSTEP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 13. Destination Burst Step Register Size (DST_BURST_STEP) Field Descriptions

Bit	Field	Value	Description
15-0	DSTBURSTSTEP		These bits specify the destination address post-increment/decrement step size while processing a burst of data:
		0x0FFF	Add 4095 to address
		...	...
		0x0002	Add 2 to address
		0x0001	Add 1 to address
		0x0000	No address change
		0xFFFF	Sub 1 from address
		0xFFFE	Sub 2 from address
		...	...
		0xF000	Sub 4096 from address
			Only values from -4096 to 4095 are valid.

8.12 Transfer Size Register (TRANSFER_SIZE) — EALLOW Protected

The transfer size register (TRANSFER_SIZE) is shown in Figure 18 and described in Table 14.

Figure 18. Transfer Size Register (TRANSFER_SIZE)

15	0
TRANSFERSIZE	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14. Transfer Size Register (TRANSFER_SIZE) Field Descriptions

Bit	Field	Value	Description
15-0	TRANSFERSIZE		These bits specify the number of bursts to transfer:
		0x0000	Transfer 1 burst
		0x0001	Transfer 2 bursts
		0x0002	Transfer 3 bursts
		...	...
		0xFFFF	Transfer 65536 bursts

8.13 Transfer Count Register (TRANSFER_COUNT)

The transfer count register (TRANSFER_COUNT) is shown in Figure 19 and described in Table 15.

Figure 19. Transfer Count Register (TRANSFER_COUNT)

15	0
TRANSFERCOUNT	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15. Transfer Count Register (TRANSFER_COUNT) Field Descriptions

Bit	Field	Value	Description
15-0	TRANSFERCOUNT		These bits specify the current transfer counter value:
		0x0000	0 bursts left to transfer
		0x0001	1 burst left to transfer
		0x0002	2 bursts left to transfer
		...	...
		0xFFFF	65535 bursts left to transfer
			The above values represent the state of the counter at the HALT conditions.

8.14 Source Transfer Step Size Register (SRC_TRANSFER_STEP) — EALLOW Protected

The source transfer step size register (SRC_TRANSFER_STEP) is shown in Figure 20 and described in Table 16.

Figure 20. Source Transfer Step Size Register (SRC_TRANSFER_STEP)

15	0
SRCTRANSFERSTEP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16. Source Transfer Step Size Register (SRC_TRANSFER_STEP) Field Descriptions

Bit	Field	Value	Description
15-0	SRCTRANSFERSTEP		These bits specify the source address pointer post-increment/decrement step size after processing a burst of data:
		0x0FFF	Add 4095 to address
		...	...
		0x0002	Add 2 to address
		0x0001	Add 1 to address
		0x0000	No address change
		0xFFFF	Sub 1 from address
		0xFFFE	Sub 2 from address
		...	...
		0xF000	Sub 4096 from address
			Only values from -4096 to 4095 are valid.

8.15 Destination Transfer Step Size Register (DST_TRANSFER_STEP) — EALLOW Protected

The destination transfer step size register (DST_TRANSFER_STEP) is shown in Figure 21 and described in Table 17.

Figure 21. Destination Transfer Step Size Register (DST_TRANSFER_STEP)

15	0
DSTTRANSFERSTEP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 17. Destination Transfer Step Size Register (DST_TRANSFER_STEP) Field Descriptions

Bit	Field	Value	Description
15-0	DSTTRANSFERSTEP		These bits specify the destination address pointer post-increment/decrement step size after processing a burst of data:
		0x0FFF	Add 4095 to address
		...	...
		0x0002	Add 2 to address
		0x0001	Add 1 to address
		0x0000	No address change
		0xFFFF	Sub 1 from address
		0xFFFE	Sub 2 from address
		...	...
		0xF000	Sub 4096 from address
			Only values from -4096 to 4095 are valid.

8.16 Source/Destination Wrap Size Register (SRC/DST_WRAP_SIZE) — EALLOW protected)

The source/destination wrap size register is shown in Figure 22 and described in Table 18.

Figure 22. Source/Destination Wrap Size Register (SRC/DST_WRAP_SIZE)

15	0
WRAPSIZE	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18. Source/Destination Wrap Size Register (SRC/DST_WRAP_SIZE) Field Descriptions

Bit	Field	Value	Description
15-0	WRAPSIZE		These bits specify the number of bursts to transfer before wrapping back to begin address pointer:
		0x0000	Wrap after 1 burst
		0x0001	Wrap after 2 bursts
		0x0002	Wrap after 3 bursts
		...	...
		0xFFFF	Wrap after 65536 bursts
			To *disable* the wrap function, set the WRAPSIZE bit field to a number larger than the TRANSFERSIZE bit field.

8.17 Source/Destination Wrap Count Register (SCR/DST_WRAP_COUNT)

The source/destination wrap count register (SCR/DST_WRAP_COUNT) is shown in Figure 23 and described in Table 19.

Figure 23. Source/Destination Wrap Count Register (SCR/DST_WRAP_COUNT)

15	0
WRAPCOUNT	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19. Source/Destination Wrap Count Register (SCR/DST_WRAP_COUNT) Field Descriptions

Bit	Field	Value	Description
15-0	WRAPCOUNT		These bits indicate the current wrap counter value:
		0x0000	Wrap complete
		0x0001	1 burst left
		0x0002	2 burst left
		...	...
		0xFFFF	65535 burst left
			The above values represent the state of the counter at the HALT conditions.

8.18 Source/Destination Wrap Step Size Registers (SRC/DST_WRAP_STEP) — EALLOW Protected

The source/destination wrap step size register (SRC/DST_WRAP_STEP) are shown in Figure 24 and described in Table 20.

Figure 24. Source/Destination Wrap Step Size Registers (SRC/DST_WRAP_STEP)

15	0
WRAPSTEP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20. Source/Destination Wrap Step Size Registers (SRC/DST_WRAP_STEP) Field Descriptions

Bit	Field	Value	Description
15-0	WRAPSTEP		These bits specify the source begin address pointer post-increment/decrement step size after wrap counter expires:
		0x0FFF	Add 4095 to address
		...	...
		0x0002	Add 2 to address
		0x0001	Add 1 to address
		0x0000	No address change
		0xFFFF	Sub 1 from address
		0xFFFE	Sub 2 from address
		...	...
		0xF000	Sub 4096 from address
			Only values from -4096 to 4095 are valid.

8.19 Shadow Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR_SHADOW/DST_BEG_ADDR_SHADOW) — All EALLOW Protected

The shadow source begin and current address pointer registers (SRC_BEG_ADDR_SHADOW/DST_BEG_ADDR_SHADOW) are shown in Figure 25 and described in Table 21.

Figure 25. Shadow Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR_SHADOW/DST_BEG_ADDR_SHADOW)

31	22	21	0
Reserved		BEGADDR	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21. Shadow Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR_SHADOW/DST_BEG_ADDR_SHADOW) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-0	BEGADDR		22-bit address value

8.20 Active Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR/DST_BEG_ADDR)

The active source begin and current address pointer registers (SRC_BEG_ADDR/DST_BEG_ADDR) are shown in Table 22 and described in Table 22.

Figure 26. Active Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR/DST_BEG_ADDR)

31	22	21	0
Reserved		BEGADDR	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 22. Active Source Begin and Current Address Pointer Registers (SRC_BEG_ADDR/DST_BEG_ADDR) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-0	BEGADDR		22-bit address value

8.21 Shadow Destination Begin and Current Address Pointer Registers (SRC_ADDR_SHADOW/DST_ADDR_SHADOW) — All EALLOW Protected

The shadow destination begin and current address pointer registers (SRC_ADDR_SHADOW/DST_ADDR_SHADOW) are shown in Figure 27 and described in Table 23.

Figure 27. Shadow Destination Begin and Current Address Pointer Registers (SRC_ADDR_SHADOW/DST_ADDR_SHADOW)

31	22	21	0
Reserved		ADDR	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 23. Shadow Destination Begin and Current Address Pointer Registers (SRC_ADDR_SHADOW/DST_ADDR_SHADOW) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-0	ADDR		22-bit address value

8.22 Active Destination Begin and Current Address Pointer Registers (SRC_ADDR/DST_ADDR)

The active destination begin and current address pointer registers (SRC_ADDR/DST_ADDR) are shown in Figure 28 and described in Table 24.

Figure 28. Active Destination Begin and Current Address Pointer Registers (SRC_ADDR/DST_ADDR)

31	22	21	0
Reserved		ADDR	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24. Active Destination Begin and Current Address Pointer Registers (SRC_ADDR/DST_ADDR) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-0	ADDR		22-bit address value

Appendix A Revision History

Table 25 lists the changes made since the previous version of this document.

Table 25. Document Revision History

Reference	Additions/Modifications/Deletions
Figure 5	Revised the graphic.

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**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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