TMS320VC5501/5502 DSP Universal Asynchronous Receiver/Transmitter (UART) Reference Guide

SPRU597B
November 2002
Revised March 2004

TEXAS INSTRUMENTS

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Read This First

About This Manual

This manual describes the features and operation of the universal asynchronous receiver/transmitter (UART) that is on the TMS320VC5501 and TMS320VC5502 digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation.

Notational Conventions

This document uses the following conventions.

- The device number TMS320C55x is often abbreviated as C55x.
- In most cases, hexadecimal numbers are shown with the suffix h. For example, the following number is a hexadecimal 40 (decimal 64):

  40h

  Similarly, binary numbers often are shown with the suffix b. For example, the following number is the decimal number 4 shown in binary form:

  0100b
- If a signal or pin is active low, it has an overbar. For example, the $\overline{\text{RESET}}$ signal is active low.

Related Documentation From Texas Instruments

The following documents describe the C55x devices and related support tools. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

TMS320C55x Technical Overview (literature number SPRU393). This overview is an introduction to the TMS320C55x DSPs, the latest generation of fixed-point DSPs in the TMS320C5000™ DSP platform. Like the previous generations, this processor is optimized for high performance and low-power operation. This book describes the CPU architecture, low-power enhancements, and embedded emulation features.

TMS320C55x DSP CPU Reference Guide (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x DSPs.

TMS320C55x DSP Peripherals Reference Guide (literature number SPRU317) describes the peripherals, interfaces, and related hardware that are available on TMS320C55x DSPs.

TMS320C55x DSP Algebraic Instruction Set Reference Guide (literature number SPRU375) describes the TMS320C55x DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

TMS320C55x DSP Mnemonic Instruction Set Reference Guide (literature number SPRU374) describes the TMS320C55x DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

TMS320C55x Optimizing C/C++ Compiler User's Guide (literature number SPRU281) describes the TMS320C55x™ C/C++ Compiler. This C/C++ compiler accepts ISO standard C and C++ source code and produces assembly language source code for TMS320C55x devices.

TMS320C55x Assembly Language Tools User's Guide (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x devices.

TMS320C55x DSP Programmer's Guide (literature number SPRU376) describes ways to optimize C and assembly code for the TMS320C55x DSPs and explains how to write code that uses special features and instructions of the DSPs.

Trademarks

TMS320, TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

Trademarks are the property of their respective owners.

Contents

Figures

Tables

Universal Asynchronous Receiver/Transmitter (UART)

This chapter describes the features and operation of the universal asynchronous receiver/transmitter (UART) that is on the TMS320VC5501 and TMS320VC5502 digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation.

1 Introduction to the UART

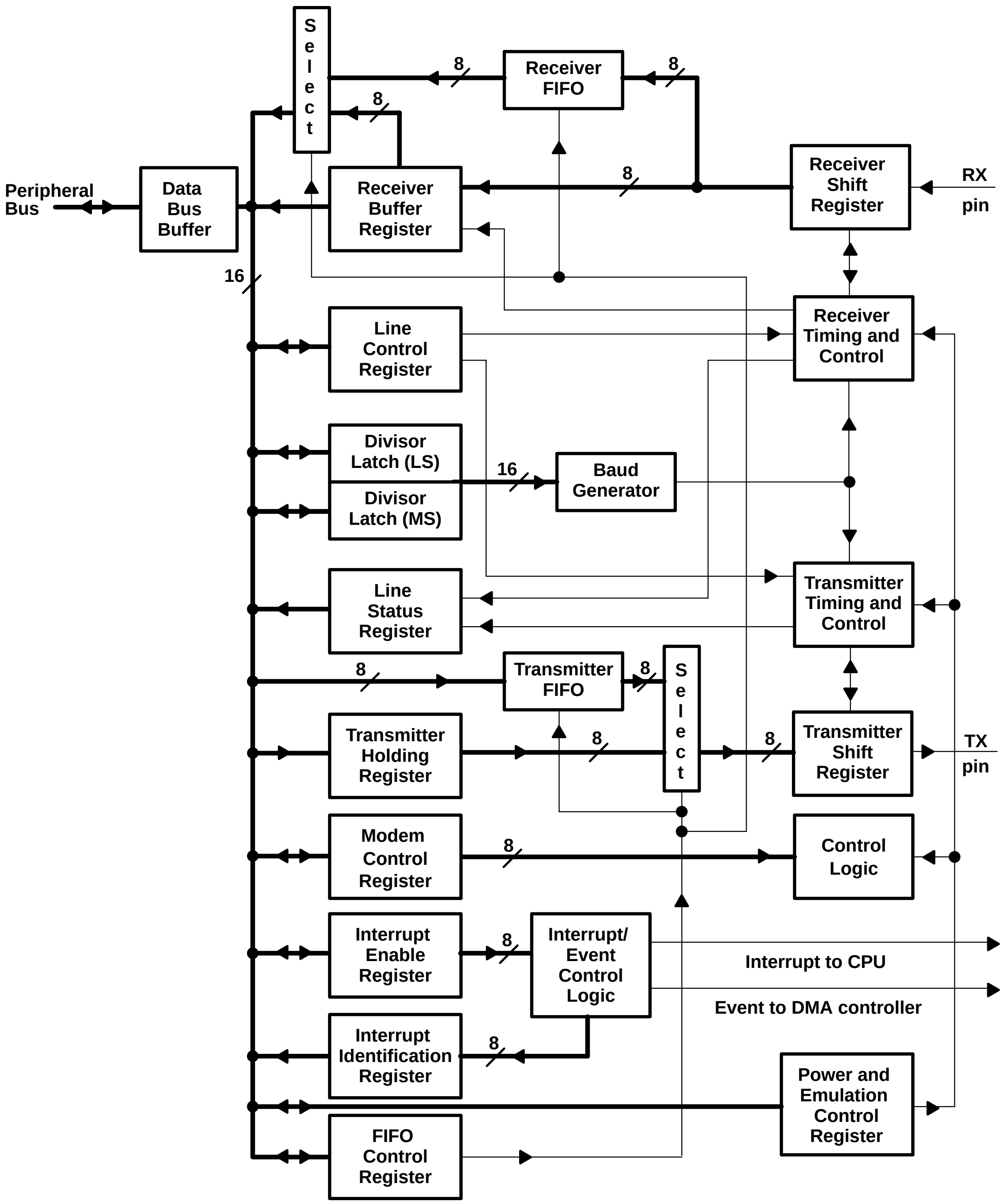
The UART peripheral is based on the industry-standard TL16C550 asynchronous communications element, which in turn is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single-character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the CPU of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link.

The UART includes a programmable baud generator capable of dividing the UART input clock by divisors from 1 to 65535 and producing a 16× reference clock for the internal transmitter and receiver logic. For detailed timing specifications for the UART, see the data manuals for the TMS320VC5501 and TMS320VC5502 devices.

Figure 1 is a block diagram of the UART.

Figure 1. UART Functional Block Diagram



1.1 Transmission

The UART transmitter section includes a transmitter hold register (URTHR) and a transmitter shift register (URTSR). When the UART is in the FIFO mode, URTHR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (URLCR). Based on the settings chosen in URLCR, the UART transmitter sends the following to the receiving device:

1 start bit
5, 6, 7, or 8 data bits
1 parity bit (optional)
1, 1.5, or 2 stop bits

URTHR receives data from the internal data bus, and when URTSR is ready, the UART moves the data from URTHR to URTSR. The UART serializes the data in URTSR and transmits the data on the TX pin.

In the non-FIFO mode, if URTHR is empty and the THR-empty interrupt is enabled in the interrupt enable register (URIER), an interrupt is generated. This interrupt is cleared when a character is loaded into URTHR. In the FIFO mode, the interrupt is generated when the transmitter FIFO is empty, and it is cleared when at least one byte is loaded into the FIFO.

1.2 Reception

The UART receiver section includes a receiver shift register (URRSR) and a receive buffer register (URRBR). When the UART is in the FIFO mode, URRBR is a 16-byte FIFO. Timing is supplied by the 16× receiver clock. Receiver section control is a function of the UART line control register (URLCR). Based on the settings chosen in URLCR, the UART receiver accepts the following from the transmitting device:

1 start bit
5, 6, 7, or 8 data bits
1 parity bit (optional)
1 stop bit (any other stop bits transferred with the above data are not detected)

URRSR receives the data bits from the RX pin. Then URRSR concatenates the data bits and moves the resulting value into URRBR (or the receiver FIFO). The UART also stores three bits of error status information next to each received character, to record a parity error, framing error, or break.

In the non-FIFO mode, when a character is placed in URRBR and the receiver data-ready interrupt is enabled in the interrupt enable register (URIER), an interrupt is generated. This interrupt is cleared when the character is read from URRBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (URFCR), and it is cleared when the FIFO contents drop below the trigger level.

2 Programmable Baud Generator

Figure 2 is a conceptual clock generation diagram for the UART. The DSP clock generator receives a signal from an external clock source and produces a **UART input clock** with a programmed frequency. Please refer to the device specific data manual for more information regarding the UART input clock. The UART contains a programmable baud generator that takes an input clock and divides it by a divisor in the range between 1 and (2^{16}–1) to produce a **baud clock (BCLK)**. The frequency of BCLK is sixteen times (16×) the baud rate; each received or transmitted bit lasts 16 BCLK cycles. When the UART is receiving, the bit is sampled in the 8th BCLK cycle. The formula to calculate the divisor is:

$$\text{Divisor} = \frac{\text{UART input clock frequency}}{\text{Desired baud rate} \times 16}$$

Two 8-bit register fields (DLM and DLL), called divisor latches, hold this 16-bit divisor. DLM holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. For information about these register fields, see section 6.3 on page 23.These divisor latches must be loaded during initialization of the UART in order to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

Figure 2. UART Clock Generation Diagram

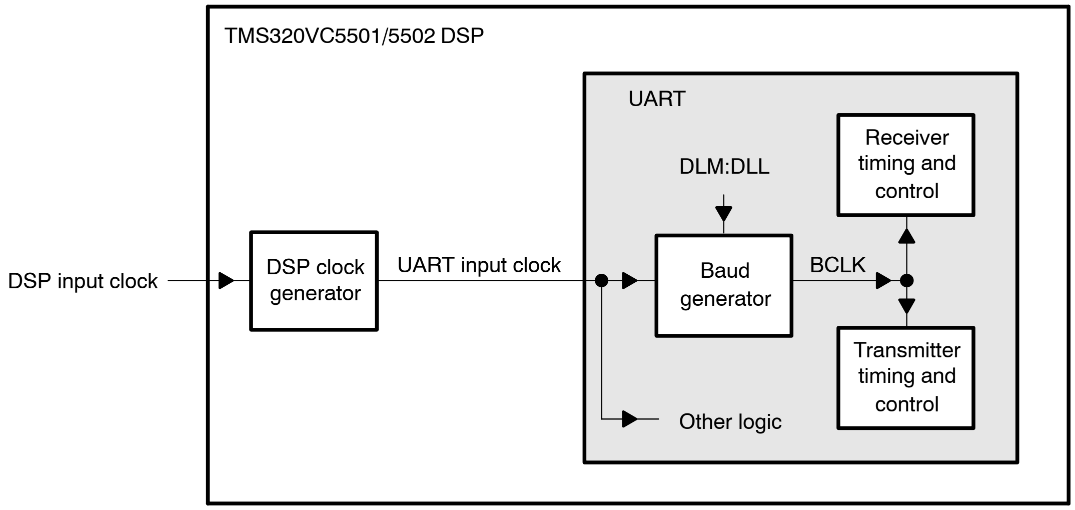


Figure 3 summarizes the relationships among the transferred data bit, BCLK, and the UART input clock.

Figure 3. Relationships Among Data Bit, BCLK, and UART Input Clock

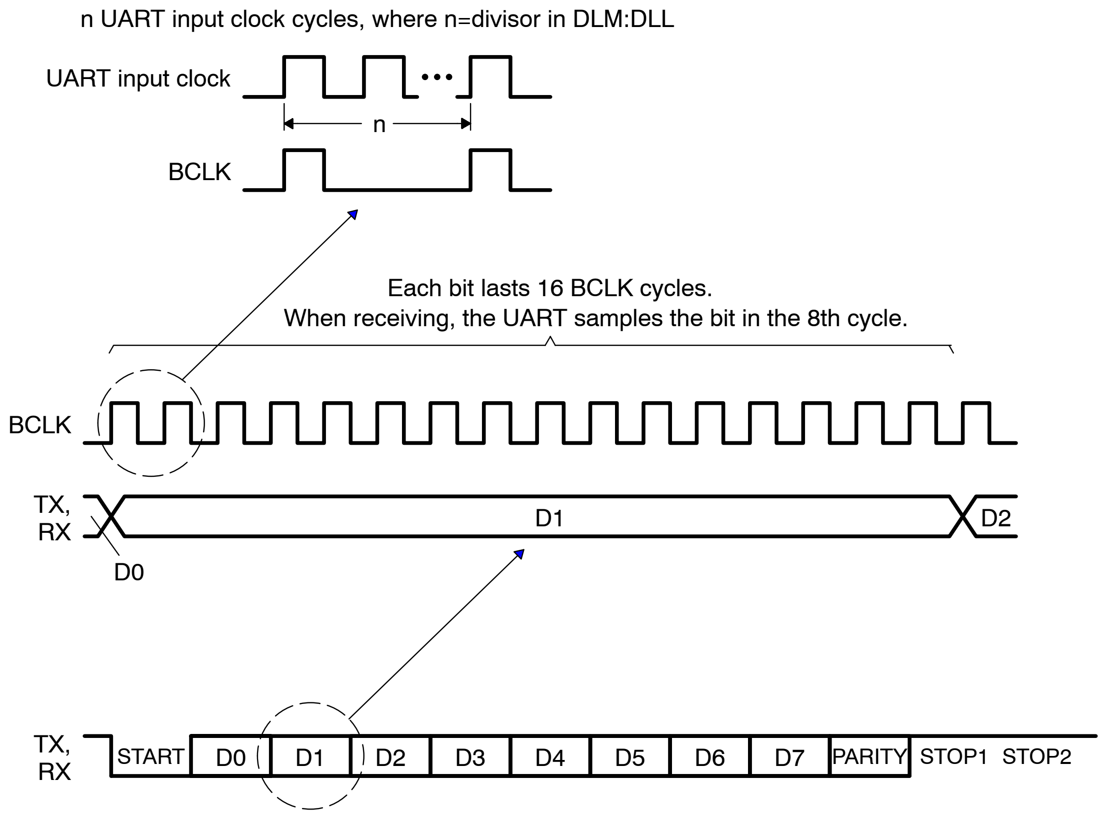

n UART input clock cycles, where n=divisor in DLM:DLL
UART input clock
n
BCLK
Each bit lasts 16 BCLK cycles.
When receiving, the UART samples the bit in the 8th cycle.
BCLK
TX,
RX
D1
D2
D0
TX,
RX
START
D0
D1
D2
D3
D4
D5
D6
D7
PARITY
STOP1
STOP2

Table 1 and Table 2 illustrate the use of the baud generator with UART input clock frequencies of 1.8432 MHz and 3.072 MHz respectively. For baud rates of 38.4 kbits/s and below, the error obtained is very small. The accuracy of the selected baud rate is dependent on the selected clock frequency.

Note:

The clock rates in Table 1 and Table 2 are shown as examples only, to illustrate the relationship of clock rate and divisor value to baud rate and baud rate error. Typically, higher clock rates will be used, and error values will differ accordingly.

Table 1. Baud Rates Using 1.8432 MHz for the UART Input Clock

Desired Baud Rate	Divisor Used to Generate 16 × Clock	Percent Error Difference Between Desired and Actual
50	2304	
75	1536	
110	1047	0.026
134.5	857	0.058
150	768	
300	384	
600	192	
1200	96	
1800	64	
2000	58	0.69
2400	48	
3600	32	
4800	24	
7200	16	
9600	12	
19200	6	
38400	3	
56000	2	2.86

Table 2. Baud Rates Using 3.072 MHz for the UART Input Clock

Desired Baud Rate	Divisor Used to Generate 16 × Clock	Percent Error Difference Between Desired and Actual
50	3840	
75	2560	
110	1745	0.026
134.5	1428	0.034
150	1280	
300	640	
600	320	
1200	160	
1800	107	0.312
2000	96	
2400	80	
3600	53	0.628
4800	40	
7200	27	1.23
9600	20	
19200	10	
38400	5	

3 Interrupt Requests and DMA Events Generated by the UART

The UART can generate the types of interrupt request described in section 3.1. Two of these can tell the CPU when to write transmit data and when to read receive data. If you want the DMA controller to handle transmit and receive data in the FIFO mode, you can use the two DMA events described in section 3.2.

3.1 UART Interrupt Requests

The UART generates the interrupt requests described in Table 3. As shown in Figure 4 (page 12), all requests are multiplexed through an arbiter to a single UART interrupt request to the CPU. Each of interrupt requests has an enable bit in the interrupt enable register (URIER) and can be recorded in the interrupt identification register (URIIR).

If an interrupt occurs and the corresponding enable bit is 1, the interrupt request is recorded in URIIR and is forwarded to the CPU. The recording in URIIR has two parts:

- ❑ The code for the interrupt request is written to the interrupt identification (IID) field. If multiple interrupt requests are enabled and they are generated simultaneously, the highest priority request is the one that is recorded in IID. A recorded code remains in IID until it is changed by one of the following events, whichever occurs first:
  - ■ The condition that caused the interrupt is cleared. IID is filled with the code 000b.
  - ■ Another interrupt occurs. IID is filled with the code for the new interrupt.
  - ■ A hardware reset occurs. IID is filled with the code 000b.
- ❑ The interrupt pending (IP) bit is made active (0) if it is not already active due to previous pending interrupts. IP remains 0 until all conditions that generated pending interrupts are cleared or until a hardware reset occurs.

If an interrupt occurs and the corresponding enable bit is 0, the interrupt request is blocked. The interrupt request is neither recorded in URIIR nor forwarded to the CPU.

The UART interrupt is one of the maskable interrupts of the CPU. As with any maskable interrupt, if it is properly enabled in the CPU, the CPU executes the corresponding interrupt service routine (ISR). The ISR for the UART interrupt can determine the interrupt source by reading the IID field of URIIR. Then the ISR can branch to the appropriate subroutine.

When multiple interrupts might be pending, the ISR can also store a copy of the line status register (URLSR). With only two exceptions, the operational events that generate interrupts also set status bits in URLSR. The exceptions are a receiver time-out event and the event of reaching the trigger level in the receiver FIFO; neither of these events has an associated status bit.

Table 3. Descriptions of the UART Interrupt Requests

UART Interrupt Request	**Interrupt Source**	**Comments**
THREINT	THR-empty condition: The transmitter holding register (URTHR) or the transmitter FIFO is empty. All of the data has been copied from URTHR to the transmitter shift register (URTSR).	If THREINT is enabled in the interrupt enable register (URIER), it is recorded in the interrupt identification register (URIIR). For more details about the THR-empty condition, including the event that clears it, see Table 13 on page 28. As an alternative to using THREINT, the CPU can poll the THRE bit of the line status register, URLSR (see page 34).
RDRINT	Receiver data-ready condition: This condition exists when data is ready to be read from the receiver buffer register (URRBR) or when the trigger level is reached in the receiver FIFO. If four character times[†] pass with no access of the FIFO, RDRINT is asserted again.	If RDRINT is enabled in URIER, it is recorded in URIIR. For more details about the receiver data-ready condition, including the event that clears it, see Table 13 on page 28. As an alternative to using RDRINT, the CPU can poll the DR bit of URLSR. In the FIFO mode, this is not a functionally equivalent alternative because the DR bit does not respond to the FIFO trigger level. The DR bit only indicates the presence or absence of unread characters.

[†] A character time is the time allotted for 1 start bit, n data bits, 1 parity bit, and 1 stop bit, where n depends on the word length selected with the WLS bits of URLCR:

Word Length	Character Time	Four Character Times
n = 5	Time for 8 bits	Time for 32 bits
n = 6	Time for 9 bits	Time for 36 bits
n = 7	Time for 10 bits	Time for 40 bits
n= 8	Time for 11 bits	Time for 44 bits

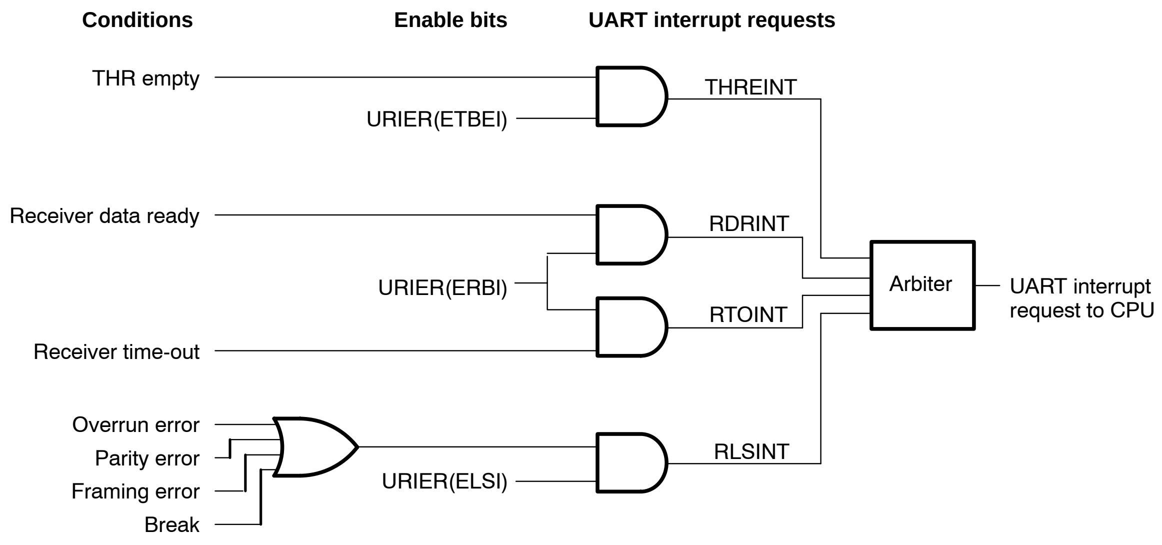
Table 3. Descriptions of the UART Interrupt Requests (Continued)

UART Interrupt Request	Interrupt Source	Comments
RTOINT	Receiver time-out condition (in the FIFO mode only): No characters have been removed from or input to the receiver FIFO during the last four character times[†], and there is at least one character in the receiver FIFO during this time.	The receiver time-out interrupt prevents the UART from waiting indefinitely in the case when the receiver FIFO level is below the trigger level and thus does not generate a receiver data-ready interrupt. If RTOINT is enabled in URIER, it is recorded in URIIR. For more details about the receiver time-out condition, including the events that clear it, see Table 13 on page 28. There is no status bit to reflect the occurrence of a time-out condition.
RLSINT	Receiver line status condition: An overrun error, parity error, framing error, or break has occurred.	If RLSINT is enabled in URIER, it is recorded in URIIR. For more details about the receiver line status condition, including the events that clear it, see Table 13 on page 28. As an alternative to using RLSINT, the CPU can poll the following bits of URLSR: overrun error indicator (OE), parity error indicator (PE), framing error indicator (FE), and break indicator (BI). **Note:** As soon as the CPU reads URLSR, OE is automatically cleared.

[†] A character time is the time allotted for 1 start bit, n data bits, 1 parity bit, and 1 stop bit, where n depends on the word length selected with the WLS bits of URLCR:

Word Length	Character Time	Four Character Times
n = 5	Time for 8 bits	Time for 32 bits
n = 6	Time for 9 bits	Time for 36 bits
n = 7	Time for 10 bits	Time for 40 bits
n= 8	Time for 11 bits	Time for 44 bits

Figure 4. Enable Paths of the UART Interrupt Requests



3.2 UART DMA Events

In the FIFO mode, the UART generates the following two DMA events. Activity in DMA channels can be synchronized to these events. In the non−FIFO mode, the UART generates no DMA events.

❑ Receive event (REVT): The trigger level for the receiver FIFO (1, 4, 8, or 14 characters) is set with the RFITR bits of URFCR (see section 6.6 on page 29). Every time the trigger level is reached or a receiver time−out occurs, the UART sends a receive event to the DMA controller. In response, the DMA controller can read the data from the receiver FIFO via URRBR.

❑ Transmit event (XEVT): When the transmitter FIFO is empty (when the last byte in the transmitter FIFO has been copied to the transmitter shift register), the UART sends an XEVT signal to the DMA controller. In response, the DMA controller can refill the transmitter FIFO via URTHR. The XEVT signal is also sent to the DMA controller when the UART is taken out of reset using the URST bit of the URPECR.

Any DMA channel synchronized to either of these events must be enabled at the time the UART event is generated. Otherwise, the DMA channel will miss the event and, unless the UART generates a new event, no data transfer will occur.

4 FIFO Modes

The following two modes can be used for servicing each of the FIFOs (transmitter FIFO and receiver FIFO):

- ❑ FIFO interrupt mode. The FIFO is enabled and the associated interrupts are enabled. Interrupts are sent to the CPU to indicate when specific events occur.
- ❑ FIFO poll mode. The FIFO is enabled but the associated interrupts are disabled. The CPU polls status bits to detect specific events.

Because the receiver FIFO and the transmitter FIFO are controlled separately, either one or both can be placed into the interrupt mode or the poll mode.

4.1 FIFO Interrupt Mode

When the receiver FIFO is enabled in URFCR and the receiver interrupts are enabled in URIER, the interrupt mode is selected for the receiver FIFO. The following are important points about the receiver interrupts:

- ❑ The receiver data-ready interrupt is issued to the CPU when the FIFO has reached the trigger level that is programmed in URFCR. It is cleared when the CPU or the DMA controller reads enough characters from the FIFO such that the FIFO drops below its programmed trigger level.
- ❑ The receiver line status interrupt is generated in response to an overrun error, a parity error, a framing error, or a break. This interrupt has higher priority than the receiver data-ready interrupt. For details, see Table 13 on page 28.
- ❑ The data-ready bit (DR) of URLSR indicates the presence or absence of characters in the receiver FIFO. The DR bit is set when a character is transferred from the receiver shift register (URRSR) to the empty receiver FIFO. The bit remains set until the FIFO is empty again.

- A receiver time-out interrupt occurs if all of the following conditions exist:
  - At least one character is in the FIFO,
  - The most recent character was received more than four continuous character times ago. A character time is the time allotted for 1 start bit, n data bits, 1 parity bit, and 1 stop bit, where n depends on the word length selected with the WLS bits of URLCR:

Word Length	Character Time	Four Character Times
n = 5	Time for 8 bits	Time for 32 bits
n = 6	Time for 9 bits	Time for 36 bits
n = 7	Time for 10 bits	Time for 40 bits
n= 8	Time for 11 bits	Time for 44 bits

  - The most recent DSP read of the FIFO has occurred more than four continuous character times before.
- Character times are calculated by using the baud rate.
- When a receiver time-out interrupt has occurred, it is cleared and the time-out timer is cleared when the CPU or the DMA controller reads one character from the receiver FIFO. The interrupt is also cleared if a new character is received in the FIFO or if the URST bit is cleared in the power and emulation control register (URPECR).
- If a receiver time-out interrupt has not occurred, the time-out timer is cleared after a new character is received or after the DSP reads the receiver FIFO.

When the transmitter FIFO is enabled in URFCR and the transmitter holding register empty interrupt is enabled in URIER, the interrupt mode is selected for the transmitter FIFO. The transmitter holding register empty interrupt occurs when the transmitter FIFO is empty. It is cleared when URTHR is loaded (1 to 16 characters may be written to the transmitter FIFO while servicing this interrupt).

4.2 FIFO Poll Mode

When the receiver FIFO is enabled in URFCR and the receiver interrupts are disabled in URIER, the poll mode is selected for the receiver FIFO. Similarly, when the transmitter FIFO is enabled and the transmitter interrupts are disabled, the transmitted FIFO is in the poll mode. In the poll mode, the CPU detects events by checking bits in the line status register (URLSR):

- ❑ The RFIER bit indicates whether there are any errors in the receiver FIFO.
- ❑ The TEMT bit indicates that both the transmitter holding register (URTHR) and the transmitter shift register (URTSR) are empty.
- ❑ The THRE bit indicates when URTHR is empty.
- ❑ The BI (break), FE (framing error), PE (parity error), and OE (overrun error) bits specify which error or errors have occurred.
- ❑ The DR (data-ready) bit is set as long as there is at least one byte in the receiver FIFO.

Also, in the FIFO poll mode:

- ❑ URIIR is not affected by any events because the interrupts are disabled.
- ❑ The UART does not indicate when the receiver FIFO trigger level is reached or when a receiver time-out occurs.

5 Power, Emulation, and Reset Considerations

5.1 Conserving Power

The DSP is divided into a number of idle domains. To minimize power consumption, you can choose which domains are active and which domains are idle at any given time. The current state of all domains is collectively called the *idle configuration*. If a particular idle configuration turns off the peripherals domain, the UART may enter an inactive, low-power mode, depending on other factors described in the device-specific data manual. When the UART is successfully requested to enter this idle mode:

- ❑ If a data transfer is in progress, the UART first completes the transfer and generates any interrupts and/or DMA events that it is programmed to generate. Then the UART enters its idle mode.
- ❑ If no data is being transferred, the UART enters its idle mode immediately.

Keep in mind that idle domains other than the peripherals domain can affect the UART. If the clock generator domain is idle, the UART has no clocks for operation, and if the DMA domain is idle, the DMA controller cannot respond to DMA events from the UART.

5.2 Emulation Mode

The FREE bit of the power and emulation control register (URPECR) determines how the UART responds to an emulator-related operation such as a debugger breakpoint. If FREE = 0 and a transmission is in progress, the UART stops after completing the one-word transmission. If FREE = 0 and a transmission is not in progress, the UART stops immediately. If FREE = 1, the UART does not stop.

5.3 Emulator Access Support

The UART supports non-intrusive emulator access; that is, an emulator can access the UART without affecting any of the UART register values.

5.4 Resetting the UART

The UART can be reset in two ways:

- ❑ UART reset. Write 0 to the UART reset bit (URST) in the power and emulation control register (URPECR). This action resets the UART state machine but does not affect the UART registers.
- ❑ DSP reset. Drive the $\overline{\text{RESET}}$ pin low. The entire DSP is reset and is held in the reset state until you drive the pin high. As part of a DSP reset, the UART state machine is reset, and the UART registers are forced to their default states. The default states of the registers and of the data transmit (TX) signal are listed in Table 4.

Notes:

1) The divisor latch registers (URDLM and URDLL) are not affected during a hardware reset or by a UART software reset and **must be loaded after power up**. Make sure the correct divisor is in these registers (to provide the desired baud rate) before initiating serial communications.
2) For proper operation of the UART, the 16-bit divisor in URDLM and URDLL must be greater than 0.
3) For proper communication between the UART and the DMA controller, the DMA mode (DMAMOD) bit of URFCR must be 1. Always write 1 to DMAMOD, and after a hardware reset, change DMAMOD from 0 to 1.

Table 4. Default State of Registers and TX Signal

Register/Signal	**Default State**
Interrupt enable register (URIER)	All bits cleared
Interrupt identification register (URIIR)	Bit 0 is set. All other bits cleared
FIFO control register (URFCR)	All bits cleared
Line control register (URLCR)	All bits cleared
Modem control register (URMCR)	All bits cleared
Line status register (URLSR)	Bit 5 and 6 are set, all other bits cleared
Receiver buffer register (URRBR)	Pointer logic for receiver FIFO cleared
Transmitter holding register (URTHR)	Pointer logic for transmitter FIFO cleared
Scratch register (URSCR)	(Not affected by reset)
Divisor latch registers (URDLL and URDLM)	(Not affected by reset)
Power and emulation control register (URPECR)	Bit 1 is set. All other bits cleared
TX signal	High

6 UART Registers

The system programmer has access to and control over any of the UART registers that are summarized in Table 5. A visual summary of the registers and their contents follows in Table 6. These registers, which control UART operations, receive data, and transmit data, are available at 16-bit addresses in the I/O space of the DSP. As on the industry-standard TL16C550 asynchronous communications element:

❑ URRBR, URTHR, and URDLL share one address. When the DLAB bit of URLCR is 0, reading from the address gives the content of URRBR, and writing to the address modifies URTHR. When DLAB = 1, all accesses at the address read or modify URDLL.

❑ URIER and URDLM share one address. When DLAB = 0, all accesses read or modify URIER. When DLAB = 1, all accesses read or modify URDLM.

❑ URIIR and URFCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of URIIR, and writing to the address modifies URFCR.

URDLL and URDLM also have dedicated addresses. If you use the dedicated addresses, you can keep the DLAB bit cleared, so that URRBR, URTHR, and URIER are always selected at the shared addresses.

Note:

Do not write to the control registers while a serial transfer is in progress. The UART expects the contents of the control registers to be static during a serial transfer.

Table 5. UART Registers

Register	Description	Access Information
URRBR/ URTHR/ URDLL[†]	Receiver buffer register (see page 21)/ Transmitter holding register (see page 22)/ Divisor latch register, least significant bits (see page 23)	If DLAB = 0: Read only: URRBR Write only: URTHR If DLAB = 1: Read/Write: URDLL
URIER/ URDLM[‡]	Interrupt enable register (see page 24)/ Divisor latch register, most significant bits (see page 23)	If DLAB = 0: Read/Write: URIER If DLAB = 1: Read/Write: URDLM
URIIR/ URFCR[§]	Interrupt identification register (see page 25)/ FIFO control register (see page 29)	Read only: URIIR Write only: URFCR
URLCR	Line control register (see page 30)	Read/Write: URLCR
URMCR	Modem control register (see page 34)	Read/Write: URMCR
URLSR	Line status register (see page 34)	Read only: URLSR
URSCR	Scratch register (see page 39)	Read/Write: URSCR
URDLL[†]	Divisor latch register, least significant bits (see page 23)	Read/Write: URDLL
URDLM[‡]	Divisor latch register, most significant bits (see page 23)	Read/Write: URDLM
URPECR	Power and emulation control register (see page 39)	Read/Write: URPECR

[†] The registers URRBR, URTHR, and URDLL share one address. URDLL also has a dedicated address. If you use the dedicated address, you can keep the DLAB bit cleared, so that URRBR and URTHR are always selected at the shared address.

[‡] The registers URIER and URDLM share one address. URDLM also has a dedicated address. If you use the dedicated address, you can keep the DLAB bit cleared, so that URIER is always selected at the shared address.

[§] The registers URIIR and URFCR share one address.

Table 6. Visual Summary of the UART Registers

	Register											
Bit	**URRBR[†] (DLAB = 0; Read only)**	**URTHR[†] (DLAB = 0); Write only)**	**URDLL[†]**	**URIER[‡] (DLAB = 0)**	**URDLM[‡]**	**URIIR[§] (Read only)**	**URFCR[§] (Write only)**	**URLCR**	**URMCR**	**URLSR**	**URSCR**	**URPECR**
15	—	—	—	—	—	—	—	—	—	—	—	URST
14	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	—	—	—	—	—	—	—	—	—	—
7	DATA	DATA	DLL	—	DLM	FIENR	RFITR	DLAB	—	RFIER	SCR	—
6				—				BC	—	TEMT		—
5				—		—	—	STPAR	—[¶]	THRE		—
4				—		—	—	EPS	LOOP	BI		—
3				—[¶]		IID	DMAMOD[#]	PEN	—[¶]	FE		—
2				ELSI			TFIRS	STB	—[¶]	PE		—
1				ETBEI			RFIRS	WLS	—[¶]	OE		—
0				ERBI		IP	FIEN		—[¶]	DR		FREE

Note: A dash (—) indicates a reserved bit.

[†] URRBR, URTHR, and URDLL share one address, and URDLL also has a dedicated address. To access URRBR and URTHR, you must clear the DLAB bit of URLCR and use the shared address. For URDLL, you can use the shared address when DLAB = 1, or you can use the dedicated address.

[‡] URIER and URDLM share one address, and URDLM also has a dedicated address. To access URIER, you must clear the DLAB bit of URLCR and use the shared address. For URDLM, you can use the shared address when DLAB = 1, or you can use the dedicated address.

[§] URIIR and URFCR share one address. Reading from the shared address returns the content of URIIR. Writing to the shared address modifies the content of URFCR.

[¶] Always write 0 to these reserved bits.

[#] Always write 1 to DMAMOD. After a hardware reset, change this bit from 0 to 1. DMAMOD = 1 is required for proper communication between the UART and the DMA controller.

6.1 Receiver Buffer Register (URRBR)

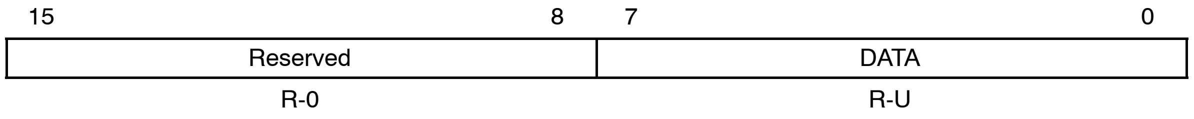
The UART receiver section consists of a receiver shift register (URRSR) and a receive buffer register (URRBR). The bit fields of URRBR are shown in Figure 5 and summarized in Table 7. When the UART is in the FIFO (TL16C550) mode, URRBR is a 16-byte FIFO. Timing is supplied by the 16× receiver clock. Receiver section control is a function of the line control register (URLCR).

URRSR receives serial data from the RX pin. Then URRSR concatenates the data and moves it into URRBR (or the receiver FIFO). In the non-FIFO (TL16C450) mode, when a character is placed in URRBR and the receiver data-ready interrupt is enabled (DR = 1 in URIER), an interrupt is generated. This interrupt is cleared when the character is read from URRBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (URFCR), and it is cleared when the FIFO contents drop below the trigger level.

Access considerations. URRBR, URTHR, and URDLL share one address. To read URRBR, write 0 to the DLAB bit of URLCR, and read from the shared address. When DLAB = 0, writing to the shared address modifies URTHR. When DLAB = 1, all accesses at the shared address read or modify URDLL.

URDLL also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that URRBR and URTHR are always selected at the shared address.

Figure 5. Receiver Buffer Register (URRBR)



Legend: R = Read; −*n* = Value after reset; −U = Value unchanged by reset

Table 7. Receiver Buffer Register (URRBR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7−0	DATA	00h−FFh	Received data. Once a data value has been shifted into the receiver shift register (URRSR), it is transferred to URRBR, where it can be read by the DSP.

6.2 Transmitter Holding Register (URTHR)

The UART transmitter section consists of a transmitter hold register (URTHR) and a transmitter shift register (URTSR). The bit fields of URTHR are shown in Figure 6 and summarized in Table 8. When the UART is in the FIFO (TL16C550) mode, URTHR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (URLCR).

URTHR receives data from the internal data bus and when URTSR is idle, the UART moves the data from URTHR to URTSR. The UART serializes the data in URTSR and transmits the data on the TX pin. In the non-FIFO (TL16C450) mode, if URTHR is empty and the THR-empty (THRE) interrupt is enabled (ETBEI = 1 in URIER), an interrupt is generated. This interrupt is cleared when a character is loaded into URTHR. In the FIFO mode, the interrupt is generated when the transmitter FIFO is empty, and it is cleared when at least one byte is loaded into the FIFO.

Access considerations. URRBR, URTHR, and URDLL share one address. To load URTHR, write 0 to the DLAB bit of URLCR, and write to the shared address. When DLAB = 0, reading from the shared address gives the content of URRBR. When DLAB = 1, all accesses at the address read or modify URDLL.

URDLL also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that URRBR and URTHR are always selected at the shared address.

Figure 6. Transmitter Holding Register (URTHR)

15 – 8	7 – 0
Reserved	DATA
R-0	W-U

Legend: R = Read; W = Write; −*n* = Value after reset; −U = Value unchanged by reset

Table 8. Transmitter Holding Register (URTHR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7−0	DATA	00h−FFh	Transmit data. To transmit data, the DSP must write the data to this field. When the transmitter shift register (URTSR) is idle, the content of URTHR is copied to URTSR.

6.3 Divisor Latch Registers (URDLL and URDLM)

Two 8-bit register fields (DLM and DLL), called divisor latches, store the 16-bit divisor for generation of the baud clock in the baud generator. The latches are in registers URDLM and URDLL (see Figure 7, Table 9, and Table 10). DLM holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. These divisor latches must be loaded during initialization of the UART in order to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

Notes:

1) The divisor latch registers are not affected during a hardware reset or by a UART software reset and **must be loaded after power up**. Make sure the correct divisor is in these registers (to provide the desired baud rate) before initiating serial communications.

2) For proper operation of the UART, the 16-bit divisor in the divisor latch registers must be greater than 0.

Access considerations. As on the industry standard TL16C550 asynchronous communications element:

- URRBR, URTHR, and URDLL share one address. When DLAB = 1 in URLCR, all accesses at the shared address are accesses to URDLL. When DLAB = 0, reading from the shared address gives the content of URRBR, and writing to the shared address modifies URTHR.
- URIER and URDLM share one address. When DLAB = 1 in URLCR, accesses to the shared address read or modify to URDLM. When DLAB = 0, all accesses at the shared address read or modify URIER.

URDLL and URDLM also have dedicated addresses. If you use the dedicated addresses, you can keep the DLAB bit cleared, so that URRBR, URTHR, and URIER are always selected at the shared addresses.

Figure 7. Divisor Latch Registers (URDLM and URDLL)

URDLM

15 – 8	7 – 0
Reserved	DLM
R-0	R/W-U

URDLL

15 – 8	7 – 0
Reserved	DLL
R-0	R/W-U

Legend: R = Read; W = Write; −*n* − Value after reset; −U = Value unchanged by reset

Table 9. Divisor Latch Register − Most Significant Bits (URDLM) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7−0	DLM	00h−FFh	Use DLM to program the most significant bits of the baud rate divisor.

Table 10. Divisor Latch Register − Least Significant Bits (URDLL) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7−0	DLL	00h−FFh	Use DLL to program the least significant bits of the baud rate divisor.

6.4 Interrupt Enable Register (URIER)

URIER is used to individually enable or disable each type of interrupt request that can be generated by the UART. Each interrupt request that is enabled in URIER is forwarded to the CPU. The content of URIER is shown in Figure 8 and summarized in Table 11.

Access considerations. URIER and URDLM share one address. To read or modify URIER, write 0 to the DLAB bit of URLCR. When DLAB = 1, all accesses at the shared address read or modify URDLM.

URDLM also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that URIER is always selected at the shared address.

Figure 8. Interrupt Enable Register (URIER)

15–4	3	2	1	0
Reserved	Reserved†	ELSI	ETBEI	ERBI
R-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read; W = Write; –*n* = Value after reset
† Write 0 to bit 3.

Table 11. Interrupt Enable Register (URIER) Field Descriptions

Bit	Field	Value	Description
15–4	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.
3	Reserved	0	Always write 0 to this bit.
2	ELSI		Enable bit for the receiver line status interrupt request
		0	Interrupt request disabled
		1	Interrupt request enabled
1	ETBEI		Enable bit for the THR-empty interrupt request
		0	Interrupt request disabled
		1	Interrupt request enabled
0	ERBI		Enable bit for the receiver data-ready interrupt request and the receiver time-out interrupt request
		0	Interrupt request disabled
		1	Interrupt request enabled

6.5 Interrupt Identification Register (URIIR)

The UART has an on-chip interrupt generation and prioritization capability that permits flexible communication with the CPU.

The UART provides three priority levels of interrupts:

- Priority 1 – Receiver line status (highest priority)
- Priority 2 – Receiver data ready or receiver time-out
- Priority 3 – Transmitter holding register empty

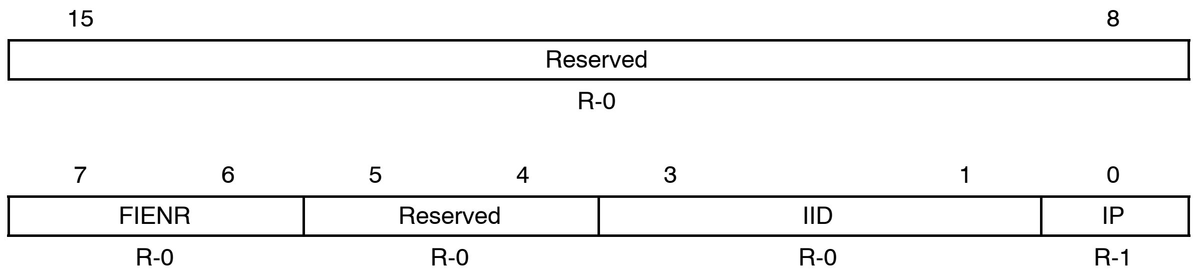
When an interrupt is generated and is enabled in URIER, the interrupt identification register (URIIR) indicates that an interrupt is pending in the IP bit and encodes the type of interrupt in the IID field.

The FIENR bit in this register can be checked to determine whether the UART is in the FIFO mode or the non-FIFO mode.

The content of URIIR is shown in Figure 9 and summarized in Table 12. Table 13 explains how to interpret bits 3−0 of URIIR.

Access consideration. URIIR and URFCR share one address. Regardless of the value of the DLAB bit of URLCR, reading from the address gives the content of URIIR, and writing to the address modifies URFCR.

Figure 9. Interrupt Identification Register (URIIR)



Legend: R = Read; −*n* = Value after reset

Table 12. Interrupt Identification Register (URIIR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.
7−6	FIENR		FIFO status bit
		00b	The non-FIFO (TL16C450) mode is selected (FIEN = 0 in URFCR).
		01b	Reserved
		10b	Reserved
		11b	The FIFO mode is selected (FIEN = 1 in URFCR).
5−4	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.

Table 12. Interrupt Identification Register (URIIR) Field Descriptions (Continued)

Bit	Field	Value	Description
3−1	IID		Interrupt identification field. When a UART interrupt is generated and is enabled in URIER, the type of interrupt is encoded in IID as shown in Table 13. If multiple interrupt requests are enabled and they are generated simultaneously, the highest priority request is the one that is recorded in IID. A recorded code remains in IID until it is changed by one of the following events, whichever occurs first: ❑ The condition that caused the interrupt is cleared. IID is filled with the code 000b. ❑ Another interrupt occurs. IID is filled with the code for the new interrupt. ❑ A hardware reset occurs. IID is filled with the code 000b.
		011b	Receiver line status interrupt (priority 1)
		010b	Receiver data-ready interrupt (priority 2)
		110b	Receiver time-out interrupt (priority 2)
		001b	Transmitter holding register empty interrupt (priority 3)
		000b	No interrupt
0	IP		Interrupt pending bit. When any UART interrupt is generated and is enabled in URIER, IP is forced to 0. IP remains 0 until all pending interrupts are cleared or until a hardware reset occurs. If no interrupts are enabled, IP is never forced to 0. **Note:** This bit has a negative polarity (0 = active).
		0	An interrupt is pending.
		1	No interrupt is pending.

Table 13. Interrupt Identification and Interrupt Clearing Information

Bits in URIIR 3	2	1	0	Priority Level	Interrupt Type	Interrupt Source	Event That Clears Interrupt
0	0	0	1	None	None	None	None
0	1	1	0	1	Receiver line status	Overrun error, parity error, framing error, or break detected	For an overrun error, reading the line status register clears the interrupt. For a parity error, framing error, or break, the interrupt is cleared only after all the erroneous data have been read.
0	1	0	0	2	Receiver data-ready	Non-FIFO mode: Receiver data ready FIFO mode: Trigger level reached. If four character times† pass with no access of the FIFO, the interrupt is asserted again.	Non-FIFO mode: The receiver buffer register is read. FIFO mode: The FIFO drops below the trigger level.‡
1	1	0	0	2	Receiver time-out	FIFO mode only: No characters have been removed from or input to the receiver FIFO during the last four character times†, and there is at least one character in the receiver FIFO during this time.	One of the following events: ❑ A character is read from the receiver FIFO.‡ ❑ A new character arrives in the receiver FIFO. ❑ URST is loaded with 0 in URPECR.
0	0	1	0	3	Transmitter holding register empty	Non-FIFO mode: Transmitter holding register empty FIFO mode: Transmitter FIFO empty	A character is written to the transmitter holding register.

† A character time is the time allotted for 1 start bit, n data bits, 1 parity bit, and 1 stop bit, where n depends on the word length selected with the WLS bits of URLCR:

Word Length	Character Time	Four Character Times
n = 5	Time for 8 bits	Time for 32 bits
n = 6	Time for 9 bits	Time for 36 bits
n = 7	Time for 10 bits	Time for 40 bits
n= 8	Time for 11 bits	Time for 44 bits

‡ In the FIFO mode, the receiver data-ready interrupt or receiver time-out interrupt is cleared by the CPU or by the DMA controller, whichever reads from the receiver FIFO first.

6.6 FIFO Control Register (URFCR)

URFCR is a write-only register at the same address as URIIR, which is a read-only register. Use URFCR to enable and clear the FIFOs and to select the receiver FIFO trigger level. Figure 10 and Table 14 describe the content of this register.

Access consideration. URIIR and URFCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of URIIR, and writing to the address modifies URFCR.

Note:

For proper communication between the UART and the DMA controller, the DMA mode (DMAMOD) bit of URFCR must be 1. Always write 1 to DMAMOD, and after a hardware reset, change DMAMOD from 0 to 1.

Figure 10. FIFO Control Register (URFCR)

15–8
Reserved
R-0

7–6	5–4	3	2	1	0
RFITR	Reserved	DMAMOD†	TFIRS	RFIRS	FIEN
W-0	R-0	W-0	W-0	W-0	W-0

Legend: R = Read; W = Write; –*n* = Value after reset

† Always write 1 to DMAMOD. After a hardware reset, change this bit from 0 to 1. DMAMOD = 1 is required for proper communication between the UART and the DMA controller.

Table 14. FIFO Control Register (URFCR) Field Descriptions

Bit	Field	Value	Description
15–8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7–6	RFITR		Receiver FIFO trigger level bits. These two bits set the trigger level for the receiver FIFO. When the trigger level is reached, a receiver data-ready interrupt is generated (if the interrupt request is enabled). Once the FIFO drops below the trigger level, the interrupt is cleared.
		00b	1 byte
		01b	4 bytes
		10b	8 bytes
		11b	14 bytes

Table 14. FIFO Control Register (URFCR) Field Descriptions (Continued)

Bit	Field	Value	Description
5−4	Reserved		Reading this field returns 0s. Writing to this field has no effect.
3	DMAMOD	1	DMA mode bit. Always write 1 to this bit. After a hardware reset, change this bit from 0 to 1. DMAMOD = 1 is a requirement for proper communication between the UART and the DMA controller.
2	TFIRS		Transmitter FIFO reset bit
		0	The bit has been cleared.
		1	This bit, when set, clears the transmitter FIFO pointer. Existing characters in the transmitter FIFO will not be transmitted and will be overwritten as new characters are loaded into the transmitter FIFO. The shift register is not cleared. The 1 that is written to this bit position is self clearing.
1	RFIRS		Receive FIFO reset bit
		0	The bit has been cleared.
		1	This bit, when set, clears the receiver FIFO pointer. Existing characters in the receiver FIFO can no longer be read and will be overwritten as new characters arrive in the receiver FIFO. The shift register is not cleared. The 1 that is written to this bit position is self clearing.
0	FIEN		FIFO enable bit
		0	Non-FIFO mode. The transmitter and receiver FIFOs are disabled, and the FIFO pointers are cleared.
		1	FIFO mode. The transmitter and receiver FIFOs are enabled.

6.7 Line Control Register (URLCR)

The system programmer controls the format of the asynchronous data communication exchange by using URLCR. In addition, the programmer can retrieve, inspect, and modify the content of URLCR; this eliminates the need for separate storage of the line characteristics in system memory. The content of this register is shown in Figure 11 and summarized in Table 15.

Figure 11. Line Control Register (URLCR)

15–8
Reserved
R-0

7	6	5	4	3	2	1–0
DLAB	BC	STPAR	EPS	PEN	STB	WLS
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read; W = Write; −*n* = Value after reset

Table 15. Line Control Register (URLCR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.
7	DLAB		Divisor latch access bit. The divisor latch registers (URDLL and URDLM) can be accessed at dedicated addresses or at addresses shared by the registers URRBR, URTHR, and URIER. Using the shared addresses requires toggling DLAB to change which registers are selected. If you use the dedicated addresses, you can keep DLAB = 0.
		0	URRBR, URTHR, and URIER selected. At the address shared by URRBR, URTHR, and URDLL, the DSP can read from URRBR and write to URTHR. At the address shared by URIER and URDLM, the DSP can read from and write to URIER.
		1	URDLL and URDLM selected. At the address shared by URRBR, URTHR, and URDLL, the DSP can read from and write to URDLL. At the address shared by URIER and URDLM, the DSP can read from and write to URDLM.
6	BC		Break control bit. Bit 6 is set to transmit a break condition to the receiving UART. A break condition is a condition where the TX signal is forced to the spacing (cleared) state. When bit 6 is cleared, the break condition is disabled.
		0	Break condition disabled
		1	Break condition enabled

Table 15. Line Control Register (URLCR) Field Descriptions (Continued)

Bit	Field	Value	Description
5	STPAR		Stick parity bit. When bits 3, 4, and 5 are set, the parity bit is transmitted and checked as cleared. When bits 3 and 5 are set and bit 4 is cleared, the parity bit is transmitted and checked as set. If bit 5 is cleared, stick parity is disabled. The relationship among bits 3, 4, and 5 is summarized in Table 16.
		0	Stick parity disabled
		1	Stick parity enabled
4	EPS		Even parity select bit. When parity is enabled (bit 3 is set) and bit 4 is set, even parity (an even number of logic 1s in the data and parity bits) is selected. When parity is enabled and bit 4 is cleared, odd parity (an odd number of logic 1s) is selected. The relationship among bits 3, 4, and 5 is summarized in Table 16.
		0	Odd parity selected
		1	Even parity selected
3	PEN		Parity enable bit. When bit 3 is set, a parity bit is generated in transmitted data between the last data word bit and the first stop bit. In received data, if bit 3 is set, parity is checked. When bit 3 is cleared, no parity is generated or checked. The relationship among bits 3, 4, and 5 is summarized in Table 16.
		0	Parity disabled
		1	Parity enabled
2	STB		Stop-bit mode bit. This bit specifies either one, one and one-half, or two stop bits with each transmitted character. The receiver clocks only the first stop bit regardless of the number of stop bits selected. The number of stop bits generated in relation to character word length and the STB bit is shown in Table 17.
		0	One stop bit is generated in the data.
		1	The number of stop bits generated is dependent on the word length selected with bits 0 and 1 (WLS).

Table 15. Line Control Register (URLCR) Field Descriptions (Continued)

Bit	Field	Value	Description
1−0	WLS		Word length select bits. These two bits specify the number of bits in each transmitted or received character.
		00b	5 bits
		01b	6 bits
		10b	7 bits
		11b	8 bits

Table 16. Relationship Among STPAR, EPS, and PEN bits of URLCR

STPAR (Bit 5)	EPS (Bit 4)	PEN (Bit 3)	Parity Option
X	X	0	Parity disabled: No parity bit transmitted or checked
0	0	1	Odd parity selected: Odd number of logic 1s
0	1	1	Even parity selected: Even number of logic 1s
1	0	1	Stick parity selected with parity bit transmitted and checked as set
1	1	1	Stick parity selected with parity bit transmitted and checked as cleared

Note: X indicates a don't care.

Table 17. Number of Stop Bits Generated

STB Bit	Word Length Selected with WLS Bits	Number of Stop Bits Generated	Baud Clock (BCLK) Cycles
0	Any word length	1	16
1	5 bits	1 1/2	24
1	6 bits	2	32
1	7 bits	2	32
1	8 bits	2	32

6.8 Modem Control Register (URMCR)

The LOOP bit of URMCR is used to enable or disable the loopback mode of the UART. The register's bit fields are shown in Figure 12 and described in Table 18.

Figure 12. Modem Control Register (URMCR)

15 12 11 8 7 6	5	4	3 0
Reserved	—†	LOOP	Reserved†
R-0	R/W-0	R/W-0	R/W-0

Legend: R = Read; W = Write; -*n* = Value after reset
† Write 0 to bit 5 and to bits 3−0.

Table 18. Modem Control Register (URMCR) Field Descriptions

Bit	Field	Value	Description
15−6	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.
5	Reserved	0	Write 0 to this bit.
4	LOOP		Loopback enable bit. This bit provides a local loopback feature for diagnostic testing of the UART.
		0	Loopback disabled
		1	Loopback enabled. When LOOP is set, the following occur: ❑ The TX signal is set high. ❑ The RX pin is disconnected. ❑ The output of the transmitter shift register (URTSR) is looped back into the receiver shift register (URRSR) input.
3−0	Reserved	0	Write 0s to these bits.

6.9 Line Status Register (URLSR)

URLSR provides information to the CPU concerning the status of data transfers. The content of this register is shown in Figure 13 and summarized in Table 19. The line status register is intended for read operations only; do not write to this register. Bits 1 through 4 record the error conditions that produce a receiver line status interrupt.

Figure 13. Line Status Register (URLSR)

15–8
Reserved
R-0

7	6	5	4	3	2	1	0
RFIER	TEMT	THRE	BI	FE	PE	OE	DR
R-0	R-1	R-1	R-0	R-0	R-0	R-0	R-0

Legend: R = Read; −*n* = Value after reset

Table 19. Line Status Register (URLSR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved		Reading this field returns 0s. Writing to this field has no effect.
7	RFIER		Receiver FIFO error indicator
			Non-FIFO mode:
		0	There has been no error, or RFIER was cleared because the CPU read the erroneous character from the receiver buffer register (URRBR).
		1	There is a parity error, framing error, or break indicator in URRBR.
			FIFO mode:
		0	There has been no error, or RFIER was cleared because the CPU read the erroneous character from the receiver FIFO and there are no more errors in the receiver FIFO.
		1	There is at least one parity error, framing error, or break indicator in the receiver FIFO.
6	TEMT		Transmitter empty indicator
			Non-FIFO mode:
		0	Either the transmitter holding register (URTHR) or the transmitter shift register (URTSR) contains a character.
		1	Both URTHR and URTSR are empty.
			FIFO mode:
		0	Either the transmitter FIFO or the transmitter shift register (URTSR) contains a character.
		1	Both the transmitter FIFO and URTSR are empty.

Table 19. Line Status Register (URLSR) Field Descriptions (Continued)

Bit	Field	Value	Description
5	THRE		Transmitter holding register empty indicator. If THRE is set and the corresponding interrupt enable bit is set (ETBEI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	URTHR is not empty. URTHR has been loaded by the CPU.
		1	URTHR is empty (ready to accept a new character). The content of URTHR has been transferred to the transmitter shift register (URTSR).
			FIFO mode:
		0	The transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. You can write to the transmitter FIFO if it is not full.
		1	The transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (URTSR).
4	BI		Break indicator. This bit is set whenever the receive data input (RX) was held low for longer than a full-word transmission time. A full-word transmission time is defined as the total time to transmit the start, data, parity, and stop bits. If BI is set and the corresponding interrupt enable bit is set (ELSI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	No break has been detected, or BI was cleared because the CPU read the erroneous character from the receiver buffer register (URRBR).
		1	A break has been detected with the character in URRBR.
			FIFO mode:
		0	No break has been detected, or BI was cleared because the CPU read the erroneous character from the receiver FIFO and the next character to be read from the FIFO has no break indicator.
		1	A break has been detected with the character at the top of the receiver FIFO.

Table 19. Line Status Register (URLSR) Field Descriptions (Continued)

Bit	Field	Value	Description
3	FE		Framing error (FE) indicator. A framing error occurs when the received character does not have a valid stop bit. In response to a framing error, the UART sets the FE bit and waits until the signal on the RX pin goes high. Once the RX signal goes high, the receiver is ready to detect a new start bit and receive new data. If FE is set and the corresponding interrupt enable bit is set (ELSI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	No framing error has been detected, or FE was cleared because the CPU read the erroneous data from the receiver buffer register (URRBR).
		1	A framing error has been detected with the character in URRBR.
			FIFO mode:
		0	No framing error has been detected, or FE was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no framing error.
		1	A framing error has been detected with the character at the top of the receiver FIFO.
2	PE		Parity error (PE) indicator. A parity error occurs when the parity of the received character does not match the parity selected with the EPS bit of URLCR. If PE is set and the corresponding interrupt enable bit is set (ELSI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	No parity error has been detected, or PE was cleared because the CPU read the erroneous data from the receiver buffer register (URRBR).
		1	A parity error has been detected with the character in URRBR.
			FIFO mode:
		0	No parity error has been detected, or PE was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no parity error.
		1	A parity error has been detected with the character at the top of the receiver FIFO.

Table 19. Line Status Register (URLSR) Field Descriptions (Continued)

Bit	Field	Value	Description
1	OE		Overrun error (OE) indicator. An overrun error in the non-FIFO mode is different from an overrun error in the FIFO mode. If OE is set and the corresponding interrupt enable bit is set (ELSI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	No overrun error has been detected, or OE was cleared because the CPU read the content of the line status register (URLSR).
		1	An overrun error has been detected. Before the character in URRBR could be read, it was overwritten by the next character arriving in URRBR.
			FIFO mode:
		0	No overrun error has been detected, or OE was cleared because the CPU read the content of the line status register (URLSR).
		1	An overrun error has been detected. If data continues to fill the FIFO beyond the trigger level, an overrun error occurs only after the FIFO is full and the next character has been completely received in the shift register. An overrun error is indicated to the CPU as soon as it happens. The new character overwrites the character in the shift register, but it is not transferred to the FIFO.
0	DR		Data-ready (DR) indicator for the receiver. If DR is set and the corresponding interrupt enable bit is set (ERBI = 1 in URIER), an interrupt request is generated (see section 3.1 on page 9).
			Non-FIFO mode:
		0	Data is not ready, or DR was cleared because the character was read from the receiver buffer register (URRBR).
		1	Data is ready. A complete incoming character has been received and transferred into URRBR.
			FIFO mode:
		0	Data is not ready, or DR was cleared because all of the characters in the receiver FIFO have been read.
		1	Data is ready. There is at least one unread character in the receiver FIFO. If the FIFO is empty, DR is set as soon as a complete incoming character has been received and transferred into the FIFO. DR remains set until the FIFO is empty again.

6.10 Scratch Register (URSCR)

The scratch register (see Figure 14 and Table 20) contains an 8-bit field that is intended for the programmer’s use as a scratch pad, in that it temporarily holds data without affecting any other UART operation.

Figure 14. Scratch Register (URSCR)

15–8	7–0
Reserved	SCR
R-0	R/W-U

Legend: R = Read; W = Write; –*n* = Value after reset

Table 20. Scratch Register (URSCR) Field Descriptions

Bit	Field	Value	Description
15−8	Reserved	0	Reading this field returns 0s. Writing to this field has no effect.
7−0	SCR	00h−FFh	This field can be used as a temporary data storage location.

6.11 Power and Emulation Control Register (URPECR)

As shown in Figure 15 and described in Table 21, URPECR contains the URST bit for putting the UART into its reset state and the FREE bit for selecting the emulation mode.

Figure 15. Power and Emulation Control Register (URPECR)

15	14–8
URST	Reserved
R/W-0	R−0

7–1	0
Reserved	FREE
R-01h	R/W−0

Legend: R = Read; W = Write; –*n* = Value after reset

Table 21. Power and Emulation Control Register (URPECR) Field Descriptions

Bit	Field	Value	Description
15	URST		UART software reset bit
		0	The receiver and transmitter are disabled and in the reset state. The UART generates no interrupts or DMA events. However, URTHR can still be loaded, and URRBR can still be read. When 0 is written to URST, the UART state machine is reset, and if there is a pending receiver time-out interrupt, this interrupt is cleared. The UART registers are not affected.
		1	The receiver and transmitter are enabled.
14−1	Reserved	0001h	Reading this field returns 0001h. Writing to this field has no effect.
0	FREE		This emulation mode bit determines the functionality of the UART in the case of an emulator-related operation such as a debugger breakpoint.
		0	If a character transfer is not in progress, the UART stops immediately. If a character transfer is in progress, the UART stops after completion of the transfer.
		1	The UART does not stop in response to the emulator-related operation.

Appendix A

Revision History

Table 22 lists the changes made since the previous version of this document. This document has been reviewed for accuracy and there are no changes since the previous version (November 2002) of this document.

Table 22. Document Revision History

Page	Additions/Modifications/Deletions
	None

Index