KeyStone Architecture
Universal Asynchronous Receiver/Transmitter (UART)

User Guide

Literature Number: SPRUGP1
November 2010

Release History

Release	Date	Chapter/Topic	Description/Comments
1.0	November 2010	All	Initial Release

Contents

Chapter 3

List of Tables

List of Figures

Preface

About This Manual

The Universal Asynchronous Receiver/Transmitter (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 UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link.

Notational Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in `screen font`.
- Information you must enter is in **`boldface screen font`**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

Note—Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

CAUTION—Indicates the possibility of service interruption if precautions are not taken.

WARNING—Indicates the possibility of damage to equipment if precautions are not taken.

Related Documentation from Texas Instruments

Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide	SPRUGS5
Packet Accelerator (PA) for KeyStone Devices User Guide	SPRUGS4
Power Sleep Controller (PSC) for KeyStone Devices User Guide	SPRUGV4

Trademarks

Chapter 1

Introduction

The following sections provide an overview of the main components and features of the Universal Asynchronous Receiver/Transmitter (UART) peripheral.

1.1 Purpose of the Peripheral

The Universal Asynchronous Receiver/Transmitter (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 or a 13× reference clock for the internal transmitter and receiver logic. For detailed timing and electrical specifications for the UART, see the device-specific data manual.

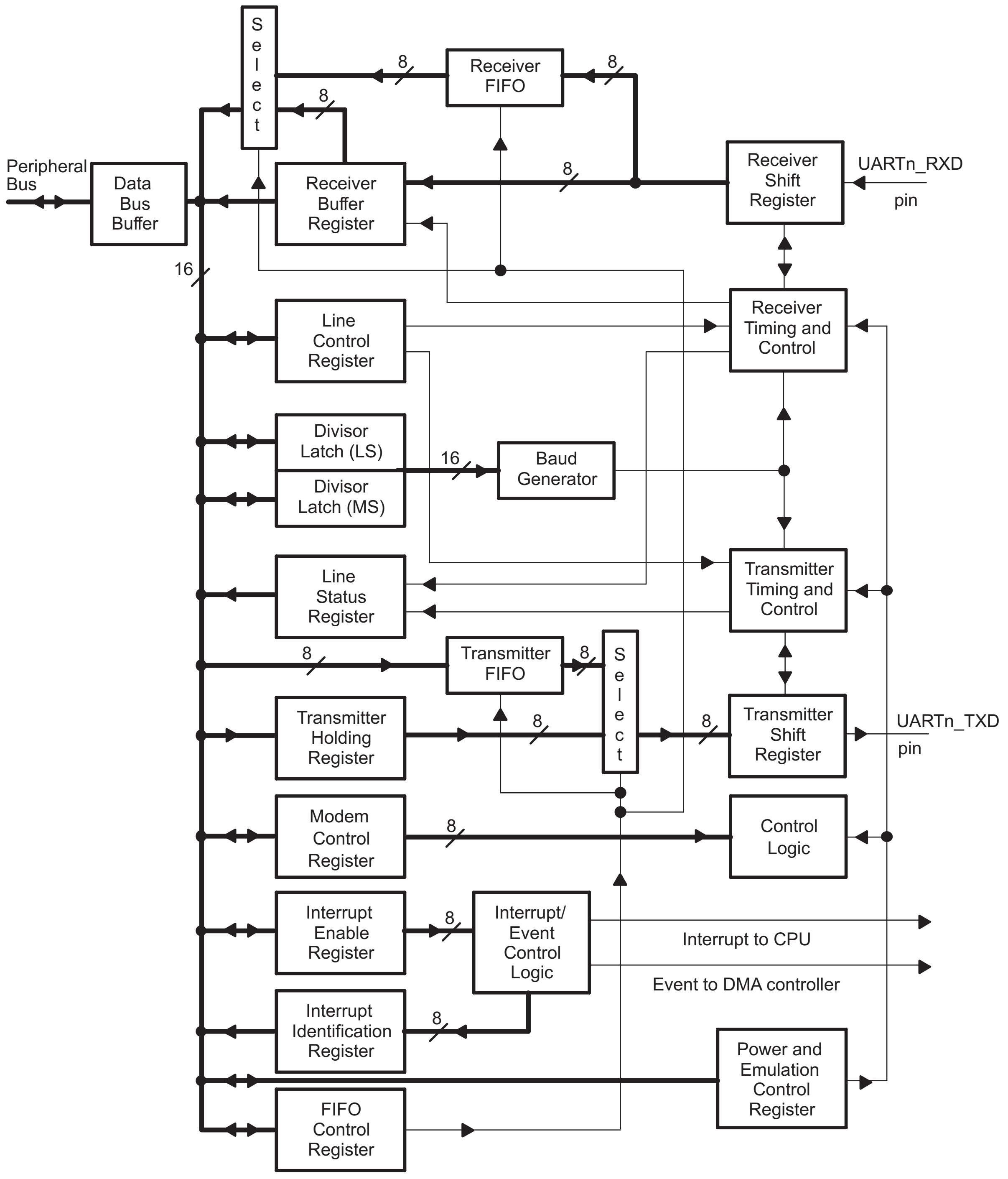
1.2 Features

Check the device-specific data manual to see the list of features that are supported and that are not supported by the UART.

1.3 Functional Block Diagram

A functional block diagram of the UART is shown in Figure 1-1.

Figure 1-1 KeyStone Device Universal Asynchronous Receiver/Transmitter (UART) Block Diagram



1.4 Industry Standard(s) Compliance Statement

The UART peripheral is based on the industry standard TL16C550 asynchronous communications element, which is a functional upgrade of the TL16C450. The information in this document assumes familiarity with these standards.

Chapter 2

Architecture

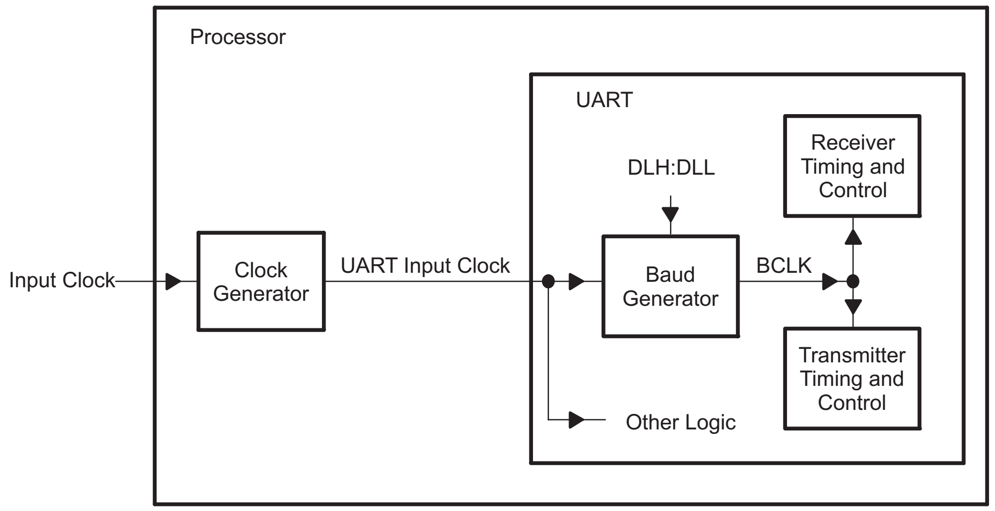
The following sections give an overview of the main components and features of the Universal Asynchronous Receiver/Transmitter (UART).

2.1 Clock Generation and Control

The UART bit clock is derived from an input clock to the UART. See the device-specific manual to check the maximum data rate supported by the UART.

Figure 2-1 is a conceptual clock generation diagram for the UART. The processor clock generator receives a signal from an external clock source and produces a UART input clock with a programmed frequency. 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) or thirteen times (13×) the baud rate (each received or transmitted bit lasts 13 BCLK cycles). When the UART is receiving, the bit is sampled in the 8th BCLK cycle for 16× over sampling mode and on the 6th BCLK cycle for 13× oversampling mode.

Figure 2-1 UART Clock Generation Diagram



The 16× or 13× reference clock is selected by configuring the OSM_SEL bit in the mode definition register (MDR). The formula to calculate the divisor is:

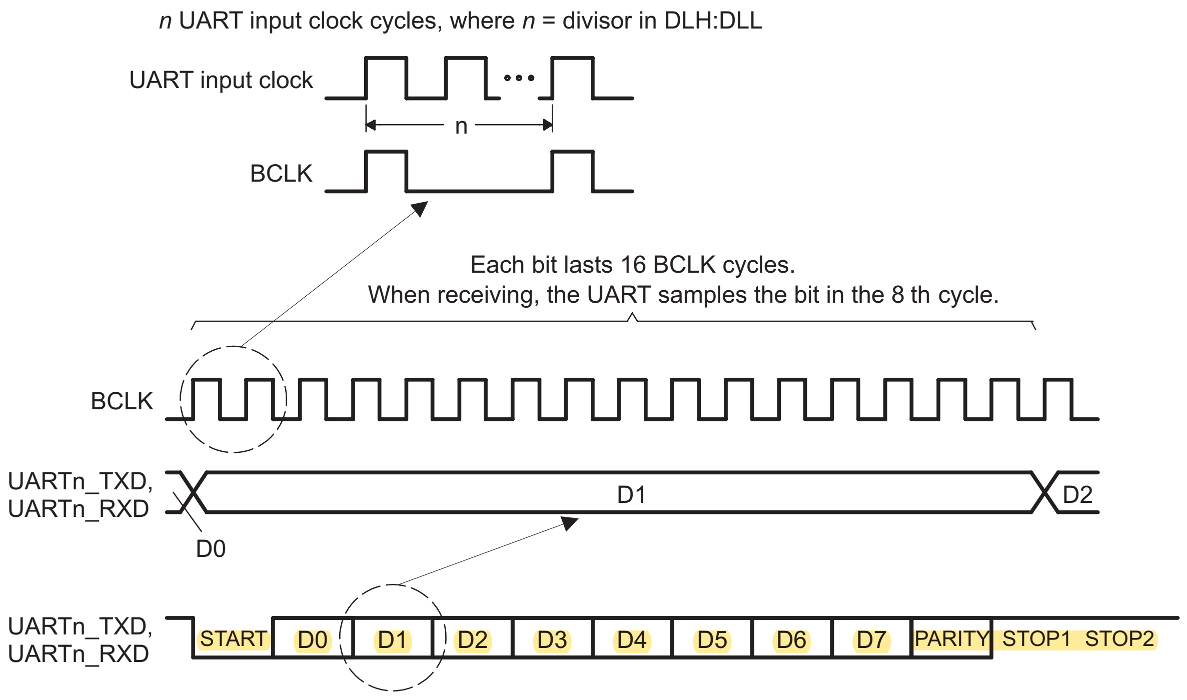
$$\text{Divisor} = \frac{\text{UART Input Clock Frequency}}{\text{Desired Baud Rate} \times 16} \qquad \text{[MDR.OSM\_SEL = 0]}$$

$$\text{Divisor} = \frac{\text{UART Input Clock Frequency}}{\text{Desired Baud Rate} \times 13} \qquad \text{[MDR.OSM\_SEL = 1]}$$

Two 8-bit register fields (DLH and DLL), called divisor latches, hold this 16-bit divisor. DLH 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 3.11. These divisor latches must be loaded during initialization of the UART 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 2-2 summarizes the relationship between the transferred data bit, BCLK, and the UART input clock. Note that the timing relationship in Figure 2-2 shows that each bit lasts for 16 BCLK cycles. This is in case of 16× oversampling mode. For 13× oversampling mode each bit lasts for 13 BCLK cycles.

Figure 2-2 Relationships Between Data Bit, BCLK, and UART Input Clock


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

Example baud rates and divisor values relative to a 150-MHz UART input clock and 16× oversampling mode are shown in Table 2-1.

Table 2-1 Baud Rate Examples for 150-MHz UART Input Clock and 16× Oversampling Mode

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	3906	2400.154	0.01
4800	1953	4800.372	0.01
9600	977	9595.701	–0.04
19200	488	19211.066	0.06
38400	244	38422.131	0.06
56000	167	56137.725	0.25
128000	73	129807.7	0.33
3000000	3	3125000	4.00

Example baud rates and divisor values relative to a 150-MHz UART input clock and 16× oversampling mode are shown in Table 2-2.

Table 2-2 Baud Rate Examples for 150-MHz UART Input Clock and 13× Oversampling Mode

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	4808	2399	–0.01
4800	2404	4799.646	–0.01
9600	1202	9599.386	–0.01
19200	601	19198.771	–0.01
38400	300	38461.538	0.16
56000	206	56011.949	0.02
128000	90	128205.128	0.16
3000000	4	2884615.385	–4.00

2.2 Signal Descriptions

UARTs use a minimal number of signal connections to interface with external devices. The UART signal descriptions are included in Table 2-3. Note that the number of UARTs and their supported features vary on each device, see the device-specific data manual for more details.

Table 2-3 UART Signal Descriptions

Signal Name [1]	Signal Type	Function
UARTn_TXD	Output	Serial data transmit
UARTn_RXD	Input	Serial data receive
$\overline{\text{UARTn\_CTS}}$ [2]	Input	Clear-to-Send handshaking signal
$\overline{\text{UARTn\_RTS}}$ [2]	Output	Request-to-Send handshaking signal

1. The value n indicates the applicable UART; that is, UART0, UART1, etc.
2. This signal is not supported in all UARTs. See the device-specific data manual to check if it is supported.

2.3 Pin Multiplexing

Extensive pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings. See the device-specific data manual to determine how pin multiplexing affects the UART.

2.4 Protocol Description

2.4.1 Transmission

The UART transmitter section includes a transmitter hold register (THR) and a transmitter shift register (TSR). When the UART is in the FIFO mode, THR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (LCR). Based on the settings chosen in the LCR, 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

2.4.2 Reception

The UART receiver section includes a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Receiver section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, 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)

2.4.3 Data Format

The UART transmits in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + STOP bit (1, 1.5, 2)

It transmits 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1, 1.5, or 2 STOP bits, depending on the STOP bit selection.

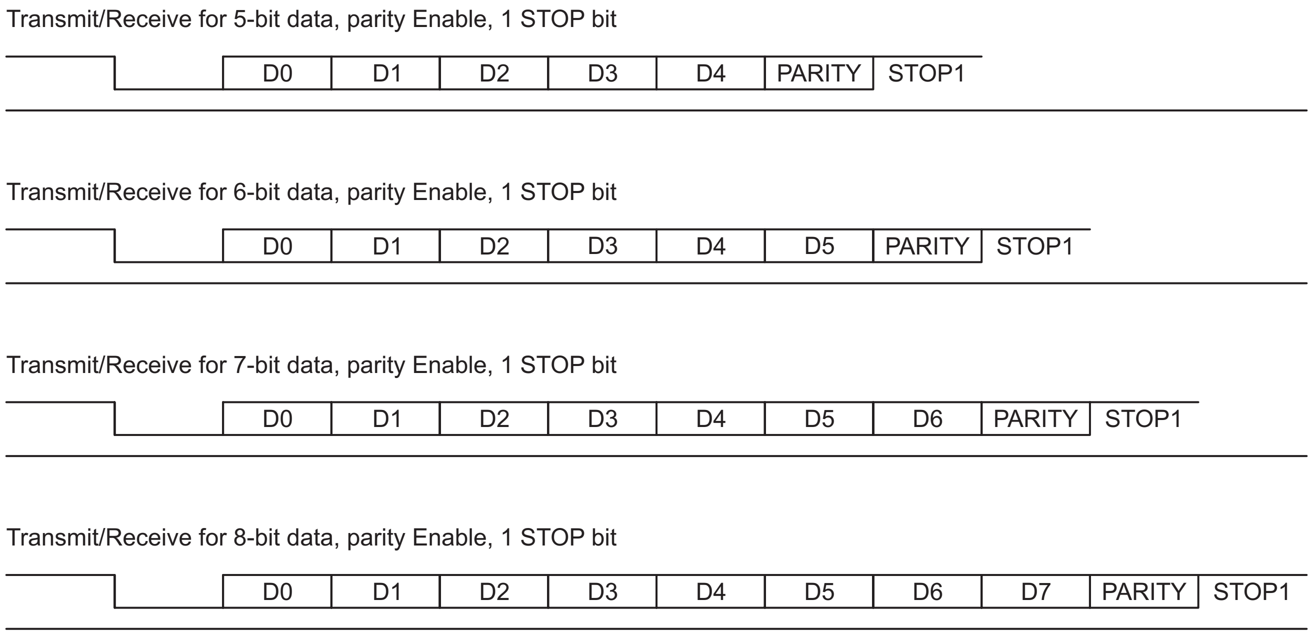
The UART receives in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + 1 STOP bit

It receives 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1 STOP bit.

The protocol formats are shown in Figure 2-3.

Figure 2-3 UART Protocol Formats


Transmit/Receive for 5-bit data, parity Enable, 1 STOP bit
D0
D1
D2
D3
D4
PARITY
STOP1
Transmit/Receive for 6-bit data, parity Enable, 1 STOP bit
D0
D1
D2
D3
D4
D5
PARITY
STOP1
Transmit/Receive for 7-bit data, parity Enable, 1 STOP bit
D0
D1
D2
D3
D4
D5
D6
PARITY
STOP1
Transmit/Receive for 8-bit data, parity Enable, 1 STOP bit
D0
D1
D2
D3
D4
D5
D6
D7
PARITY
STOP1

2.5 Operation

2.5.1 Transmission

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

1. 1 START bit
2. 5, 6, 7, or 8 data bits
3. 1 PARITY bit (optional)
4. 1, 1.5, or 2 STOP bits

THR receives data from the internal data bus, and when TSR is ready, the UART moves the data from THR to TSR. The UART serializes the data in TSR and transmits the data on the UARTn_TXD pin. In the non-FIFO mode, if THR is empty and the THR empty interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when a character is loaded into THR. 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.

2.5.2 Reception

The UART receiver section includes a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Timing is supplied by the 16Þ receiver clock. Receiver section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, 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)

RSR receives the data bits from the UARTn_RXD pin. Then RSR concatenates the data bits and moves the resulting value into RBR (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 RBR and the receiver data-ready interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when the character is read from RBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (FCR), and it is cleared when the FIFO contents drop below the trigger level.

2.5.3 FIFO Modes

The following two modes can be used for servicing the receiver and transmitter FIFOs:

- 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.

2.5.3.1 FIFO Interrupt Mode

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are enabled in the interrupt enable register (IER), 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 FCR. 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 Section 2.8 "Interrupt Support".
- The data-ready (DR) bit in the line status register (LSR) 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 (RSR) to the empty receiver FIFO. The DR 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 in the line control register (LCR). See Table 2-4.
  - The most recent 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 EDMA 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 URRST bit is cleared in the power and emulation management register (PWREMU_MGMT).
- If a receiver time-out interrupt has not occurred, the time-out timer is cleared after a new character is received or after the CPU or EDMA reads the receiver FIFO.

When the transmitter FIFO is enabled in FCR and the transmitter holding register empty interrupt is enabled in IER, 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 the transmitter hold register (THR) is loaded (1 to 16 characters may be written to the transmitter FIFO while servicing this interrupt).

Table 2-4 Character Time for Word Lengths

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

2.5.3.2 FIFO Poll Mode

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are disabled in the interrupt enable register (IER), 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 (LSR):

- The RXFIFOE bit indicates whether there are any errors in the receiver FIFO.
- The TEMT bit indicates that both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty.
- The THRE bit indicates when THR 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.

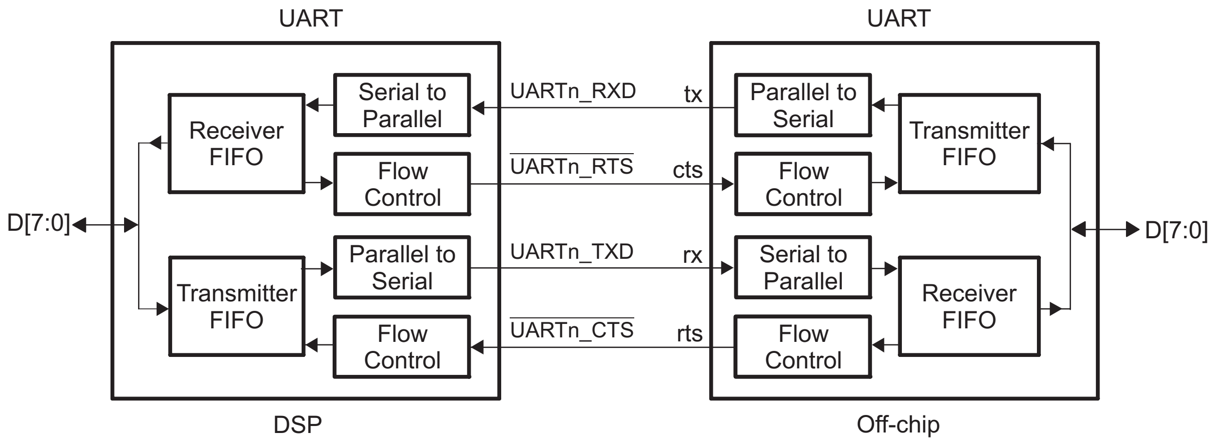
Also, in the FIFO poll mode:

- The interrupt identification register (IIR) 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.

2.5.4 Autoflow Control

The UART can employ autoflow control by connecting the UARTn_CTS and UARTn_RTS signals. Note that all UARTs do not support autoflow control, see the device-specific data manual for supported features. The UARTn_CTS input must be active before the transmitter FIFO can transmit data. The UARTn_RTS becomes active when the receiver needs more data and notifies the sending device. When UARTn_RTS is connected to UARTn_CTS, data transmission does not occur unless the receiver FIFO has space for the data. Therefore, when two UARTs are connected as shown in Figure 2-4 with autoflow enabled, overrun errors are eliminated.

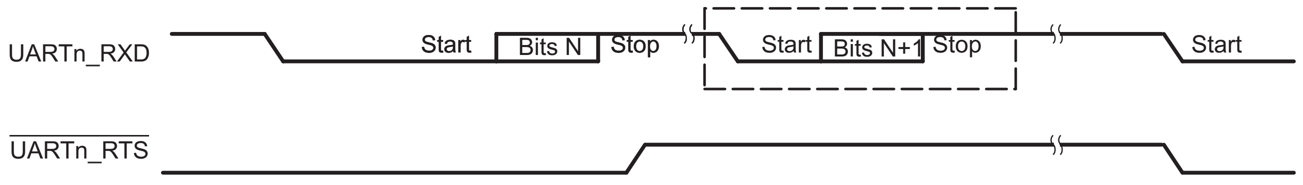
Figure 2-4 UART Interface Using Autoflow Diagram



2.5.4.1 UARTn_RTS Behavior

UARTn_RTS data flow control originates in the receiver block (Figure 1-1 on page 1-3). When the receiver FIFO level reaches a trigger level of 1, 4, 8, or 14 (see Figure 2-5), UARTn_RTS is deasserted. The sending UART may send an additional byte after the trigger level is reached (assuming the sending UART has another byte to send), because it may not recognize the deassertion of UARTn_RTS until after it has begun sending the additional byte. For trigger level 1, 4, and 8, UARTn_RTS is automatically reasserted after the receiver FIFO is emptied. For trigger level 14, UARTn_RTS is automatically reasserted when the receiver FIFO drops below the trigger level.

Figure 2-5 Autoflow Functional Timing Waveforms for UARTn_RTS

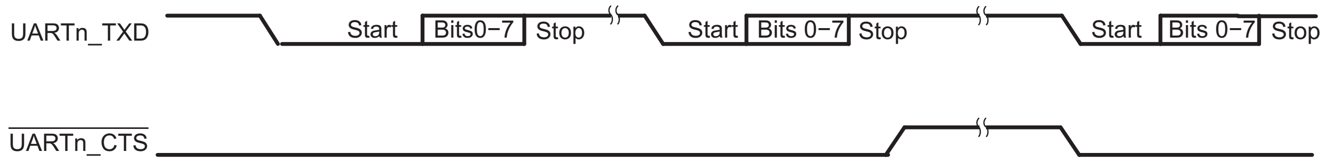


(A) The N = Receiver FIFI trigger level.
(B) The two blocks in dashed lines cover the case in which an additional byte is sent.

2.5.4.2 $\overline{\text{UARTn\_CTS}}$ Behavior

The transmitter checks UARTn_CTS before sending the next data byte. If UARTn_CTS is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, UARTn_CTS must be released before the middle of the last STOP bit that is currently being sent (see Figure 2-6). When flow control is enabled, UARTn_CTS level changes do not trigger interrupts because the device automatically controls its own transmitter. Without autoflow control, the transmitter sends any data present in the transmitter FIFO and a receiver overrun error may result.

Figure 2-6 Autoflow Functional Timing Waveforms for $\overline{\text{UARTn\_CTS}}$



(A) When $\overline{\text{UARTn\_CTS}}$ is active (low), the transmitter keeps sending serial data out.
(B) When $\overline{\text{UARTn\_CTS}}$ goes high before the middle of the last STOP bit of the current byte, the transmitter finishes sending the current byte but it does not send the next byte.
(C) When $\overline{\text{UARTn\_CTS}}$ goes from high to low, the transmitter begins sending data again.

2.5.5 Loopback Control

The UART can be placed in the diagnostic mode using the LOOP bit in the modem control register (MCR), which internally connects the UART output back to the UART input. In this mode, the transmit and receive data paths, the transmitter and receiver interrupts, and the modem control interrupts can be verified without connecting to another UART.

2.6 Reset Considerations

2.6.1 Software Reset Considerations

Two bits in the power and emulation management register (PWREMU_MGMT) control resetting the parts of the UART:

- The UTRST bit controls resetting the transmitter only. If UTRST = 1, the transmitter is active; if UTRST = 0, the transmitter is in reset.
- The URRST bit controls resetting the receiver only. If URRST = 1, the receiver is active; if URRST = 0, the receiver is in reset.

In each case, putting the receiver and/or transmitter in reset will reset the state machine of the affected portion but does not affect the UART registers.

2.6.2 Hardware Reset Considerations

When the processor RESET pin is asserted, the entire processor is reset and is held in the reset state until the RESET pin is released. As part of a device reset, the UART state machine is reset and the UART registers are forced to their default states. The default states of the registers are shown in Chapter 3 "Registers" on page 3-1.

2.7 Initialization

The following steps are required to initialize the UART:

1. Perform the necessary device pin multiplexing setup (see the device-specific data manual).
2. Set the desired baud rate by writing the appropriate clock divisor values to the divisor latch registers (DLL and DLH).
3. If the FIFOs will be used, select the desired trigger level and enable the FIFOs by writing the appropriate values to the FIFO control register (FCR). The FIFOEN bit in FCR must be set first, before the other bits in FCR are configured.
4. Choose the desired protocol settings by writing the appropriate values to the line control register (LCR).
5. If autoflow control is desired, write appropriate values to the modem control register (MCR). Note that all UARTs do not support autoflow control, see the device-specific data manual for supported features.
6. Choose the desired response to emulation suspend events by configuring the FREE bit and enable the UART by setting the UTRST and URRST bits in the power and emulation management register (PWREMU_MGMT).

2.8 Interrupt Support

2.8.1 Interrupt Events and Requests

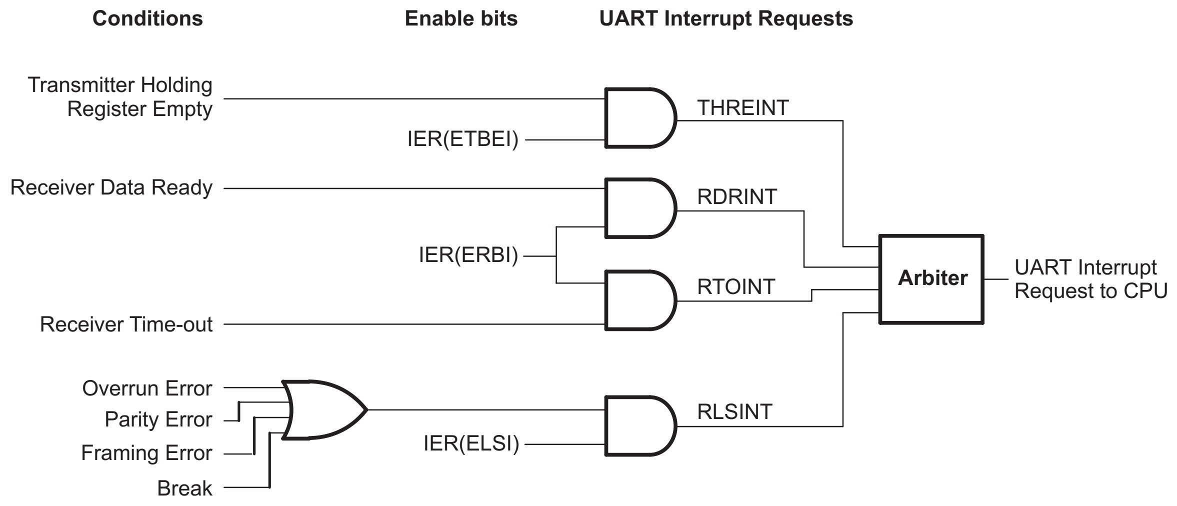
The UART generates the interrupt requests described in Table 2-5. All requests are multiplexed through an arbiter to a single UART interrupt request to the CPU, as shown in Figure 2-7. Each of the interrupt requests has an enable bit in the interrupt enable register (IER) and is recorded in the interrupt identification register (IIR).

If an interrupt occurs and the corresponding enable bit is set to 1, the interrupt request is recorded in IIR and is forwarded to the CPU. If an interrupt occurs and the corresponding enable bit is cleared to 0, the interrupt request is blocked. The interrupt request is neither recorded in IIR nor forwarded to the CPU.

Table 2-5 UART Interrupt Requests Descriptions

UART Interrupt Request	Interrupt Source	Comment
THREINT	THR-empty condition: The transmitter holding register (THR) or the transmitter FIFO is empty. All of the data has been copied from THR to the transmitter shift register (TSR).	If THREINT is enabled in IER, by setting the ETBEI bit, it is recorded in IIR. As an alternative to using THREINT, the CPU can poll the THRE bit in the line status register (LSR).
RDAINT	Receive data available in non-FIFO mode or trigger level reached in the FIFO mode.	If RDAINT is enabled in IER, by setting the ERBI bit, it is recorded in IIR. As an alternative to using RDAINT, the CPU can poll the DR bit in the line status register (LSR). 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 indicates only the presence or absence of unread characters.
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 (see Table 2-4), 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 when the receiver FIFO level is below the trigger level and, therefore, does not generate a receiver data-ready interrupt. If RTOINT is enabled in IER, by setting the ERBI bit, it is recorded in IIR. 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 IER, by setting the ELSI bit, it is recorded in IIR. • As an alternative to using RLSINT, the CPU can poll the following bits in the line status register (LSR): overrun error indicator (OE), parity error indicator (PE), framing error indicator (FE), and break indicator (BI).

Figure 2-7 UART Interrupt Request Enable Paths



2.8.2 Interrupt Multiplexing

The UARTs have dedicated interrupt signals to the CPU and the interrupts are not multiplexed with any other interrupt source.

2.9 DMA Event Support

In the FIFO mode, the UART generates the following two DMA events:

- **Receive event (URXEVT):** The trigger level for the receiver FIFO (1, 4, 8, or 14 characters) is set with the RXFIFTL bit in the FIFO control register (FCR). Every time the trigger level is reached or a receiver time-out occurs, the UART sends a receive event to the EDMA controller. In response, the EDMA controller reads the data from the receiver FIFO by way of the receiver buffer register (RBR).
- **Transmit event (UTXEVT):** 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 UTXEVT signal to the EDMA controller. In response, the EDMA controller refills the transmitter FIFO by way of the transmitter holding register (THR). The UTXEVT signal is also sent to the DMA controller when the UART is taken out of reset using the UTRST bit in the power and emulation management register (PWREMU_MGMT).

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

2.10 Power Management

The UART peripheral can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the UART peripheral is controlled by the processor Power and Sleep Controller (PSC). The PSC acts as a master controller for power management for all of the peripherals on the device. For detailed information on power management procedures using the PSC, see the *Power Sleep Controller (PSC) for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page ø-viii.

2.11 Emulation Considerations

The FREE bit in the power and emulation management register (PWREMU_MGMT) determines how the UART responds to an emulation suspend event such as an emulator halt or breakpoint. If FREE = 0 and a transmission is in progress, the UART halts after completing the one-word transmission; if FREE = 0 and a transmission is not in progress, the UART halts immediately. If FREE = 1, the UART does not halt and continues operating normally.

Note also that emulator accesses are essentially transparent to UART operation. Emulator read operations do not affect any register contents, status bits, or operating states. Emulator writes, however, may affect register contents and may affect UART operation, depending on what register is accessed and what value is written.

The UART registers can be read from or written to during emulation suspend events, even if the UART activity has stopped.

2.12 Exception Processing

2.12.1 Divisor Latch Not Programmed

Since the processor reset signal has no effect on the divisor latch, the divisor latch will have an unknown value after power up. If the divisor latch is not programmed after power up, the baud clock (BCLK) will not operate and will instead be set to a constant logic 1 state.

The divisor latch values should always be reinitialized following a processor reset.

2.12.2 Changing Operating Mode During Busy Serial Communication

Since the serial link characteristics are based on how the control registers are programmed, the UART will expect the control registers to be static while it is busy engaging in a serial communication. Therefore, changing the control registers while the module is still busy communicating with another serial device will most likely cause an error condition and should be avoided.

Chapter 3

Registers

The system programmer has access to and control over any of the UART registers that are listed in Table 3-1. These registers, which control UART operations, receive data, and transmit data, are available at 32-bit addresses in the device memory map. See the device-specific data manual for the memory address of these registers.

- RBR, THR, and DLL share one address. When the DLAB bit in LCR is 0, reading from the address gives the content of RBR, and writing to the address modifies THR. When DLAB = 1, all accesses at the address read or modify DLL. DLL can also be accessed with address offset 20h.
- IER and DLH share one address. When DLAB = 0, all accesses read or modify IER. When DLAB = 1, all accesses read or modify DLH. DLH can also be accessed with address offset 24h.
- IIR and FCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of IIR, and writing modifies FCR.

Table 3-1 UART Registers

Offset	Acronym	Register Description	Section
0h	RBR	Receiver Buffer Register (read only)	Section 3.1
0h	THR	Transmitter Holding Register (write only)	Section 3.2
4h	IER	Interrupt Enable Register	Section 3.3
8h	IIR	Interrupt Identification Register (read only)	Section 3.4
8h	FCR	FIFO Control Register (write only)	Section 3.5
Ch	LCR	Line Control Register	Section 3.6
10h	MCR	Modem Control Register	Section 3.7
14h	LSR	Line Status Register	Section 3.8
18h	MSR	Modem Status Register	Section 3.9
1Ch	SCR	Scratch Pad Register	Section 3.10
20h	DLL	Divisor LSB Latch	Section 3.11
24h	DLH	Divisor MSB Latch	Section 3.11
28h	REVID1	Revision Identification Register 1	Section 3.12
2Ch	REVID2	Revision Identification Register 2	Section 3.12
30h	PWREMU_MGMT	Power and Emulation Management Register	Section 3.13
34h	MDR	Mode Definition Register	Section 3.14
End of Table 3-1			

3.1 Receiver Buffer Register (RBR)

The UART receiver section consists of a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Timing is supplied by the 16× receiver clock or 13× receiver clock by programming OSM_SEL bit field of MDR register. Receiver section control is a function of the line control register (LCR).

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

Access considerations:

- The RBR, THR, and DLL share one address. To read the RBR, write 0 to the DLAB bit in the LCR, and read from the shared address. When DLAB = 0, writing to the shared address modifies thee THR. When DLAB = 1, all accesses at the shared address read or modify the DLL.
- The DLL also has a dedicated address. If the dedicated address is used, DLAB can = 0, so that the RBR and THR are always selected at the shared address.

The RBR is shown in Figure 3-1 and described in Table 3-2.

Figure 3-1 Receiver Buffer Register (RBR)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		DATA	
R-0		R-0	

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

Table 3-2 Receiver Buffer Register (RBR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DATA	0-FFh	Received data.
End of Table 3-2			

3.2 Transmitter Holding Register (THR)

The UART transmitter section consists of a transmitter hold register (THR) and a transmitter shift register (TSR). Transmitter control is a function of the line control register (LCR).

The THR receives data from the internal data bus. When the TSR is idle the UART then moves the data from the THR to the TSR. The UART serializes the data in the TSR and transmits the data on the TX pin.

In the non-FIFO mode, if the THR is empty and the TSR is empty, the THRE interrupt is enabled (ETBEI = 1 in the IER), and an interrupt is generated. This interrupt is cleared when a character is loaded into the THR.

When the UART is in the FIFO mode, the THR is a 16-byte FIFO. 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:

- The RBR, THR, and DLL share one address. To load the THR, write 0 to the DLAB bit of the LCR, and write to the shared address. When DLAB = 0, reading from the shared address gives the content of the RBR. When DLAB = 1, all accesses at the address read or modify the DLL.
- The DLL also has a dedicated address. If the dedicated address is used, DLAB can = 0, so that the RBR and the THR are always selected at the shared address.

The transmitter holding register (THR) is shown in Figure 3-2 and described in Table 3-3.

Figure 3-2 Transmitter Holding Register (THR)

31 16
Reserved
R-0

15 8	7 0
Reserved	DATA
R-0	R-0

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

Table 3-3 Transmitter Holding Register (THR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DATA	0-FFh	Data to transmit.
End of Table 3-3			

3.3 Interrupt Enable Register (IER)

The interrupt enable register (IER) 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 the IER is forwarded to the CPU.

Access considerations:

- The IER and DLH share one address. To read or modify the IER, write 0 to the DLAB bit in the LCR. When DLAB = 1, all accesses at the shared address read or modify the DLH.
- The DLH also has a dedicated address. If the dedicated address is used, DLAB can = 0, so that the IER is always selected at the shared address.

The IER is shown in Figure 3-3 and described in Table 3-4.

Figure 3-3 Interrupt Enable Register (IER)

31 — 16
Reserved
R-0

15 — 4	3			0
Reserved	EDSSI	ELSI	ETBEI	ERBI
R-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 3-4 Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	EDSSI	0	Enable modem status interrupt.
2	ELSI		Receiver line status interrupt enable. 0 = Receiver line status interrupt is disabled. 1 = Receiver line status interrupt is enabled.
1	ETBEI		Transmitter holding register empty interrupt enable. 0 = Transmitter holding register empty interrupt is disabled. 1 = Transmitter holding register empty interrupt is enabled.
0	ERBI		Receiver data available interrupt and character timeout indication interrupt enable. 0 = Receiver data available interrupt and character timeout indication interrupt is disabled. 1 = Receiver data available interrupt and character timeout indication interrupt is enabled.
End of Table 3-4			

3.4 Interrupt Identification Register (IIR)

The interrupt identification register (IIR) is a read-only register at the same address as the FIFO control register (FCR), which is a write-only register. When an interrupt is generated and enabled in the interrupt enable register (IER), the IIR indicates that an interrupt is pending in the IPEND bit and encodes the type of interrupt in the INTID bits.

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 timeout
- Priority 3 - Transmitter holding register empty

The FIFOEN bit in the IIR can be checked to determine whether the UART is in the FIFO mode or the non-FIFO mode.

Access consideration:
The IIR and FCR share one address. Regardless of the value of the DLAB bit in the LCR, reading from the address gives the content of the IIR, and writing to the address modifies the FCR.

The IIR is shown in Figure 3-4 and described in Table 3-5.

Figure 3-4 Interrupt Identification Register (IIR)

31 16
Reserved
R-0

15 8	7 6	5 4	3 1	0
Reserved	FIFOEN	Reserved	INTID	IPEND
R-0	R-0	R-0	R-0	R-1

Legend: R = Read only; -*n* = value after reset

Table 3-5 Interrupt Identification Register (IIR) Field Descriptions (Part 1 of 2)

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-6	FIFOEN	0-3h	FIFOs enabled. 0 = Non-FIFO mode. 1 = Reserved. 2 = Reserved. 3 = FIFOs are enabled. FIFOEN bit in the FIFO control register (FCR) is set to 1.
5-4	Reserved	0	Reserved

Table 3-5 Interrupt Identification Register (IIR) Field Descriptions (Part 2 of 2)

Bit	Field	Value	Description
3-1	INTID	0-7h	Interrupt type. See Table 3-6. 0 = Reserved. 1 = Transmitter holding register empty (priority 3). 2 = Receiver data available (priority 2). 3 = Receiver line status (priority 1, highest). 4 = Reserved. 5 = Reserved. 6 = Character timeout indication (priority 2). 7 = Reserved.
0	IPEND		Interrupt pending. When any UART interrupt is generated and is enabled in IER, IPEND is forced to 0. IPEND remains 0 until all pending interrupts are cleared or until a hardware reset occurs. If no interrupts are enabled, IPEND is never forced to 0. 0 = Interrupts pending. 1 = No interrupts pending.
End of Table 3-5			

Table 3-6 Interrupt Identification and Interrupt Clearing Information

Priority Level	IIR Bits				Interrupt Type	Interrupt Source	Event That Clears Interrupt
	3	2	1	0			
None	0	0	0	1	None	None	None
1	0	1	1	0	Receiver line status	Overrun error, parity error, framing error, or break is detected.	For an overrun error, reading the line status register (LSR) clears the interrupt. For a parity error, framing error, or break, the interrupt is cleared only after all the erroneous data have been read.
2	0	1	0	0	Receiver data-ready	Non-FIFO mode: Receiver data is ready.	Non-FIFO mode: The receiver buffer register (RBR) is read.
						FIFO mode: Trigger level reached. If four character times (see Table 2-4) pass with no access of the FIFO, the interrupt is asserted again.	FIFO mode: The FIFO drops below the trigger level. [1]
2	1	1	0	0	Receiver time-out	FIFO mode only: No characters have been removed from or input to the receiver FIFO during the last four character times (see Table 2-4), 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. [1] • A new character arrives in the receiver FIFO. • The URRST bit in the power and emulation management register (PWREMU_MGMT) is loaded with 0.
3	0	0	1	0	Transmitter holding register empty	• Non-FIFO mode: Transmitter holding register (THR) is empty. • FIFO mode: Transmitter FIFO is empty.	A character is written to the transmitter holding register (THR).
End of Table 3-6							

1. 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.

3.5 FIFO Control Register (FCR)

The FIFO control register (FCR) is a write-only register at the same address as the interrupt identification register (IIR), which is a read-only register. Use the FCR to enable and clear the FIFOs and to select the receiver FIFO trigger level. The FIFOEN bit must be set to 1 before other FCR bits are written to or the FCR bits are not programmed.

Access consideration:

The IIR and FCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of the IIR, and writing to the address modifies the FCR.

CAUTION—For proper communication between the UART and the EDMA controller, the DMAMODE1 bit must be set to 1. Always write a 1 to the DMAMODE1 bit, and after a hardware reset, change the DMAMODE1 bit from 0 to 1.

The FCR is shown in Figure 3-5 and described in Table 3-7

Figure 3-5 FIFO Control Register (FCR)

31–16
Reserved
R-0

15–8	7–6	5–4	3	2	1	0
Reserved	RXFIFTL	Reserved	DMAMODE1[1]	TXCLR	RXCLR	FIFOEN
R-0	R-0	R-0	W-0	W1C-0	W1C-0	W-0

Legend: R = Read only; W = Write only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

1. Always write 1 to the DMAMODE1 bit. After a hardware reset, change the DMAMODE1 bit from 0 to 1. DMAMODE = 1 is required for proper communication between the UART and the DMA controller.

Table 3-7 FIFO Control Register (FCR) Field Descriptions (Part 1 of 2)

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-6	RXFIFTL	0-3h	Receiver FIFO trigger level. RXFIFTL sets 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). When the FIFO drops below the trigger level, the interrupt is cleared. 0 = 1 byte. 1 = 4 bytes. 2 = 8 bytes. 3 = 14 bytes.
5-4	Reserved	0	Reserved
3	DMAMODE1		DMA MODE1 enable if FIFOs are enabled. Always write 1 to DMAMODE1. After a hardware reset, change DMAMODE1 from 0 to 1. DMAMOD1 = 1 is a requirement for proper communication between the UART and the EDMA controller. 0 = DMA MODE1 is disabled. 1 = DMA MODE1 is enabled.
2	TXCLR		Transmitter FIFO clear. Write a 1 to TXCLR to clear the bit. 0 = No effect. 1 = Clears transmitter FIFO and resets the transmitter FIFO counter. The shift register is not cleared.

Table 3-7 FIFO Control Register (FCR) Field Descriptions (Part 2 of 2)

Bit	Field	Value	Description
1	RXCLR		Receiver FIFO clear. Write a 1 to RXCLR to clear the bit. 0 = No effect. 1 = Clears receiver FIFO and resets the receiver FIFO counter. The shift register is not cleared.
0	FIFOEN		Transmitter and receiver FIFOs mode enable. FIFOEN must be set before other FCR bits are written to or the FCR bits are not programmed. Clearing this bit clears the FIFO counters. 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.
End of Table 3-7			

3.6 Line Control Register (LCR)

The system programmer controls the format of the asynchronous data communication exchange by using the line control register (LCR). In addition, the programmer can retrieve, inspect, and modify the content of the LCR. This eliminates the need for separate storage of the line characteristics in system memory.

The LCR is shown in Figure 3-6 and described in Table 3-8.

Figure 3-6 Line Control Register (LCR)

31	16
Reserved	
R-0	

15	8	7	6	5	4	3	2	1	0
Reserved		DLAB	BC	SP	EPS	PEN	STB	WLS	
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

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

Table 3-8 Line Control Register (LCR) Field Descriptions (Part 1 of 2)

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7	DLAB		Divisor latch access bit. The divisor latch registers (DLL and DLH) can be accessed at dedicated addresses or at addresses shared by RBR, THR, and IER. Using the shared addresses requires toggling DLAB to change which registers are selected. If the dedicated addresses are used, DLAB can = 0. 0 = Allows access to the receiver buffer register (RBR), the transmitter holding register (THR), and the interrupt enable register (IER) selected. At the address shared by RBR, THR, and DLL, the CPU can read from RBR and write to THR. At the address shared by IER and DLH, the CPU can read from and write to IER. 1 = Allows access to the divisor latches of the baud generator during a read or write operation (DLL and DLH). At the address shared by RBR, THR, and DLL, the CPU can read from and write to DLL. At the address shared by IER and DLH, the CPU can read from and write to DLH.
6	BC		Break control. 0 = Break condition is disabled. 1 = Break condition is transmitted to the receiving UART. A break condition is a condition in which the UARTn_TXD signal is forced to the spacing (cleared) state.
5	SP		Stick parity. The SP bit works in conjunction with the EPS and PEN bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 3-9. 0 = Stick parity is disabled. 1 = Stick parity is enabled. • When odd parity is selected (EPS = 0), the PARITY bit is transmitted and checked as set. • When even parity is selected (EPS = 1), the PARITY bit is transmitted and checked as cleared.
4	EPS		Even parity select. Selects the parity when parity is enabled (PEN = 1). The EPS bit works in conjunction with the SP and PEN bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 3-9. 0 = Odd parity is selected (an odd number of logic 1s is transmitted or checked in the data and PARITY bits). 1 = Even parity is selected (an even number of logic 1s is transmitted or checked in the data and PARITY bits).
3	PEN		Parity enable. The PEN bit works in conjunction with the SP and EPS bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 3-9. 0 = No PARITY bit is transmitted or checked. 1 = Parity bit is generated in transmitted data and is checked in received data between the last data word bit and the first STOP bit.

Table 3-8 Line Control Register (LCR) Field Descriptions (Part 2 of 2)

Bit	Field	Value	Description
2	STB		Number of STOP bits generated. STB specifies 1, 1.5, or 2 STOP bits in each transmitted character. When STB = 1, the WLS bit determines the number of STOP bits. The receiver clocks only the first STOP bit, regardless of the number of STOP bits selected. The number of STOP bits generated is summarized in Table 3-10. 0 = One STOP bit is generated. 1 = WLS bit determines the number of STOP bits: » When WLS = 0, 1.5 STOP bits are generated. » When WLS = 1h, 2h, or 3h, 2 STOP bits are generated.
1-0	WLS	0-3h	Word length select. Number of bits in each transmitted or received serial character. When STB = 1, the WLS bit determines the number of STOP bits. 0 = 5 bits 1 = 6 bits 2 = 7 bits 3 = 8 bits
End of Table 3-8			

Table 3-9 Relationship Between ST, EPS, and PEN Bits in LCR

ST Bit	EPS Bit	PEN Bit	Parity Option
x	x	0	Parity disabled: No PARITY bit is 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.
End of Table 3-9			

Table 3-10 Number of STOP Bits Generated

ST Bit	WLS Bit	Word Length Selected with WLS Bits	Number of STOP Bits Generated	Baud Clock (BCLK) Cycles
0	x	Any word length	1	16
1	0h	5 bits	1.5	24
1	1h	6 bits	2	32
1	2h	7 bits	2	32
1	3h	8 bits	2	32
End of Table 3-10				

3.7 Modem Control Register (MCR)

The modem control register (MCR) provides the ability to enable/disable the autoflow functions, and enable/disable the loopback function for diagnostic purposes.

The MCR is shown in Figure 3-7 and described in Table 3-11.

Figure 3-7 Modem Control Register (MCR)

31 – 16
Reserved
R-0

15 – 6	5	4	3	2	1	0
Reserved	AFE [1]	LOOP	OUT2	OUT1	RTS[1]	Reserved
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

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

1. All UARTs do not support this feature. See the device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0.

Table 3-11 Modem Control Register (MCR) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5	AFE		Autoflow control enable. Autoflow control allows the $\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ signals to provide handshaking between UARTs during data transfer. When AFE = 1, the RTS bit determines the autoflow control enabled. Note that all UARTs do not support this feature. See the device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0. 0 = Autoflow control is disabled. 1 = Autoflow control is enabled: » When RTS = 0, only $\overline{\text{UARTn\_CTS}}$ is enabled. » When RTS = 1, $\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ are enabled.
4	LOOP		Loopback mode enable. LOOP is used for the diagnostic testing using the loopback feature. 0 = Loopback mode is disabled. 1 = Loopback mode is enabled. When LOOP is set, the following occur: » The UARTn_TXD signal is set high. » The UARTn_RXD pin is disconnected » The output of the transmitter shift register (TSR) is looped back in to the receiver shift register (RSR) input.
3	OUT2	0	OUT2 Control Bit.
2	OUT1	0	OUT1 Control Bit.
1	RTS		RTS control. When AFE = 1, the RTS bit determines the autoflow control enabled. Note that all UARTs do not support this feature. See the device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0. 0 = $\overline{\text{UARTn\_RTS}}$ is disabled, only $\overline{\text{UARTn\_CTS}}$ is enabled. 1 = $\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ are enabled.
0	Reserved	0	Reserved
End of Table 3-11			

3.8 Line Status Register (LSR)

The line status register (LSR) provides information to the CPU concerning the status of data transfers. It 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.

The LSR is shown in Figure 3-8 and described in Table 3-12.

Figure 3-8 Line Status Register (LSR)

31–16
Reserved
R-0

15–8	7	6	5	4	3	2	1	0
Reserved	RXFIFOE	TEMT	THRE	BIS	FE	PE	OE	DR
R-0	R-0	R-1	R-1	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; -*n* = value after reset

Table 3-12 Line Status Register (LSR) Field Descriptions (Part 1 of 3)

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	RXFIFOE		Receiver FIFO error. **In non-FIFO mode:** 0 = There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver buffer register (RBR). 1 = There is a parity error, framing error, or break indicator in the receiver buffer register (RBR). **In FIFO mode:** 0 = There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver FIFO and there are no more errors in the receiver FIFO. 1 = At least one parity error, framing error, or break indicator in the receiver FIFO.
6	TEMT		Transmitter empty (TEMT) indicator. **In non-FIFO mode:** 0 = Either the transmitter holding register (THR) or the transmitter shift register (TSR) contains a data character. 1 = Both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty. **In FIFO mode:** 0 = Either the transmitter FIFO or the transmitter shift register (TSR) contains a data character. 1 = Both the transmitter FIFO and the transmitter shift register (TSR) are empty.
5	THRE		Transmitter holding register empty (THRE) indicator. If the THRE bit is set and the corresponding interrupt enable bit is set (ETBEI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = Transmitter holding register (THR) is not empty. THR has been loaded by the CPU. 1 = Transmitter holding register (THR) is empty (ready to accept a new character). The content of THR has been transferred to the transmitter shift register (TSR). **In FIFO mode:** 0 = Transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. The transmitter FIFO may be written to if it is not full. 1 = Transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (TSR).

Table 3-12 Line Status Register (LSR) Field Descriptions (Part 2 of 3)

Bit	Field	Value	Description
4	BI		Break indicator. The BI bit is set whenever the receive data input (UARTn_RXD) 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 the BI bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = No break has been detected, or the BI bit was cleared because the CPU read the erroneous character from the receiver buffer register (RBR). 1 = A break has been detected with the character in the receiver buffer register (RBR). **In FIFO mode:** 0 = No break has been detected, or the BI bit 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.
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. When the RX signal goes high, the receiver is ready to detect a new START bit and receive new data. If the FE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR). 1 = A framing error has been detected with the character in the receiver buffer register (RBR). **In FIFO mode:** 0 = No framing error has been detected, or the FE bit 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 in the line control register (LCR). If the PE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR). 1 = A parity error has been detected with the character in the receiver buffer register (RBR). **In FIFO mode:** 0 = No parity error has been detected, or the PE bit 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.
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 the OE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). 1 Overrun error has been detected. Before the character in the receiver buffer register (RBR) could be read, it was overwritten by the next character arriving in RBR. **In FIFO mode:** 0 = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). 1 = 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.

Table 3-12 Line Status Register (LSR) Field Descriptions (Part 3 of 3)

Bit	Field	Value	Description
0	DR		Data-ready (DR) indicator for the receiver. If the DR bit is set and the corresponding interrupt enable bit is set (ERBI = 1 in IER), an interrupt request is generated. **In non-FIFO mode:** 0 = Data is not ready, or the DR bit was cleared because the character was read from the receiver buffer register (RBR). 1 = Data is ready. A complete incoming character has been received and transferred into the receiver buffer register (RBR). **In FIFO mode:** 0 = Data is not ready, or the DR bit 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, the DR bit is set as soon as a complete incoming character has been received and transferred into the FIFO. The DR bit remains set until the FIFO is empty again.
End of Table 3-12			

3.9 Modem Status Register (MSR)

The modem status register (MSR) provides information to the CPU concerning the status of modem control signals. It is intended for read operations only — do not write to this register.

The MSR is shown in Figure 3-9 and described in Table 3-13.

Figure 3-9 Modem Status Register (MSR)

31–16
Reserved
R-0

15–8	7	6	5	4	3	2	1	0
Reserved	CD	RI	DSR	CTS	DCD	TERI	DDSR	DCTS
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; -*n* = value after reset

Table 3-13 Modem Status Register (MSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	CD	0	Complement of the Carrier Detect input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 3 (OUT2).
6	RI	0	Complement of the Ring Indicator input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 2 (OUT1).
5	DSR	0	Complement of the Data Set Ready input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 0 (DTR).
4	CTS	0	Complement of the Clear To Send input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 1 (RTS).
3	DCD	0	Change in DCD indicator bit. DCD indicates that the DCD input has changed state since the last time it was read by the CPU. When DCD is set and the modem status interrupt is enabled, a modem status interrupt is generated.
2	TERI	0	Trailing edge of RI (TERI) indicator bit. TERI indicates that the RI input has changed from a low to a high. When TERI is set and the modem status interrupt is enabled, a modem status interrupt is generated.
1	DDSR	0	Change in DSR indicator bit. DDSR indicates that the DSR input has changed state since the last time it was read by the CPU. When DDSR is set and the modem status interrupt is enabled, a modem status interrupt is generated.
0	DCTS	0	Change in CTS indicator bit. DCTS indicates that the CTS input has changed state since the last time it was read by the CPU. When DCTS is set (autoflow control is not enabled and the modem status interrupt is enabled), a modem status interrupt is generated. When autoflow control is enabled, no interrupt is generated.
End of Table 3-13			

3.10 Scratch Pad Register (SCR)

The scratch pad register (SCR) is intended for the programmer's use as a scratch pad. It temporarily holds the programmer's data without affecting UART operation.

The SCR is shown in Figure 3-10 and described in Table 3-14.

Figure 3-10 Scratch Pad Register (SCR)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		SCR	
R-0		R-0	

Legend: R = Read only; -*n* = value after reset

Table 3-14 Scratch Pad Register (MSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	SCR	0	These bits are intended for the programmer's use as a scratch pad in the sense that it temporarily holds the programmer's data without affecting any other UART operation.
End of Table 3-14			

3.11 Divisor Latches (DLL and DLH)

Two 8-bit register fields (DLL and DLH), called divisor latches, store the 16-bit divisor for generation of the baud clock in the baud generator. The latches are in the DLH and DLL. The DLH holds the most-significant bits of the divisor, and the DLL holds the least-significant bits of the divisor. These divisor latches must be loaded during initialization of the UART 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.

Access considerations:

- The RBR, THR, and DLL share one address. When DLAB = 1 in the LCR, all accesses at the shared address are accesses to the DLL. When DLAB = 0, reading from the shared address gives the content of the RBR, and writing to the shared address modifies the THR.
- The IER and DLH share one address. When DLAB = 1 in the LCR, accesses to the shared address read or modify the DLH. When DLAB = 0, all accesses at the shared address read or modify the IER.

The DLL and DLH also have dedicated addresses. If dedicated addresses are used, the DLAB bit can be kept cleared, so that the RBR, THR, and IER are always selected at the shared addresses.

The divisor LSB latch (DLL) is shown in Figure 3-11 and described in Table 3-15.

Figure 3-11 Divisor LSB Latch (DLL)

31–16
Reserved
R-0

15–8	7–0
Reserved	DLL
R-0	R/W-0

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

Table 3-15 Divisor LSB Latch (DLL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DLL	0-Fh	The 8 least-significant bits (LSBs) of the 16-bit divisor for generation of the baud clock in the baud rate generator.
End of Table 3-15			

The divisor MSB latch (DLH) is shown in Figure 3-12 and described in Table 3-16.

Figure 3-12 Divisor MSB Latch (DLH)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		DLH	
R-0		R-0	

Legend: R = Read only; -*n* = value after reset

Table 3-16 Divisor MSB Latch (DLH) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DLH	0-Fh	The 8 most-significant bits (MSBs) of the 16-bit divisor for generation of the baud clock in the baud rate generator. Maximum baud rate is 128 kbps.
End of Table 3-16			

3.12 Revision Identification Registers (REVID1 and REVID2)

Revision identification registers (REVID1 and REVID2) contain peripheral identification data for the peripheral.

The REVID1 is shown in Figure 3-13 and described in Table 3-17.

Figure 3-13 Revision Identification Register 1 (REVID1)

31 ... 0
REVID1
R-1102 0002h

Legend: R = Read only; -*n* = value after reset

Table 3-17 Revision Identification Register 1 (REVID1) Field Descriptions

Bit	Field	Value	Description
31-0	REVID1	1102 0002h	Peripheral identification number.
End of Table 3-17			

The REVID2 is shown in Figure 3-14 and described in Table 3-18.

Figure 3-14 Revision Identification Register 2 (REVID2)

31 ... 16
Reserved
R-0

15 ... 8	7 ... 0
Reserved	REVID2
R-0	R-0

Legend: R = Read only; -*n* = value after reset

Table 3-18 Revision Identification Register 2 (REVID2) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	REVID2	0	Peripheral identification number.
End of Table 3-18			

3.13 Power and Emulation Management Register (PWREMU_MGMT)

The power and emulation management register (PWREMU_MGMT) is shown in Figure 3-15 and described in Table 3-19.

Figure 3-15 Power and Emulation Management Register (PWREMU_MGMT)

31–16
Reserved
R-0

15	14	13	12–1	0
Reserved	UTRST	URRST	Reserved	FREE
R/W-0	R/W-0	R/W-0	R-1	R/W-0

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

Table 3-19 Power and Emulation Management Register (PWREMU_MGMT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	Reserved	0	Reserved. This bit must always be written as 0.
14	UTRST		UART transmitter reset. Resets and enables the transmitter. 0 = Transmitter is disabled and in reset state. 1 = Transmitter is enabled.
13	URRST		UART receiver reset. Resets and enables the receiver. 0 = Receiver is disabled and in reset state. 1 = Receiver is enabled.
12-1	Reserved	1	Reserved
0	FREE		Free-running enable mode bit. This bit determines the emulation mode functionality of the UART. When halted, the UART can handle register read/write requests, but does not generate any transmission/reception, interrupts or events. 0 = If a transmission is not in progress, the UART halts immediately. If a transmission is in progress, the UART halts after completion of the one-word transmission. 1 = Free-running mode is enabled. UART continues to run normally.
End of Table 3-19			

3.14 Mode Definition Register (MDR)

The Mode Definition register (MDR) determines the oversampling mode for the UART. The MDR is shown in Figure 3-16 and described in Table 3-20.

Figure 3-16 Mode Definition Register (MDR)

31 … 16
Reserved
R-0

15 … 1	0
Reserved	OSM_SEL
R-0	R/W-0

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

Table 3-20 Mode Definition Register (MDR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	OSM_SEL		Over-Sampling Mode Select. 0 = 16× oversampling. 1 = 13× oversampling.
End of Table 3-20			

Index

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	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps