TMS320DM647/DM648 DSP Universal Asynchronous Receiver/Transmitter (UART)

User's Guide

Literature Number: SPRUEL8A
October 2007

TEXAS INSTRUMENTS

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the universal asynchronous receiver/transmitter (UART) in the TMS320DM647/DM648 Digital Signal Processor (DSP).

Notational Conventions

This document uses the following conventions.

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

Note: Acronyms 3PSW, CPSW, CPSW_3G, and 3pGSw are interchangeable and all refer to the 3 port gigabit switch.

Related Documentation From Texas Instruments

The following documents describe the TMS320DM647/DM648 Digital Signal Processor (DSP). 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.

SPRS372 — ***TMS320DM647/DM648 Digital Media Processor Data Manual*** describes the signals, specifications and electrical characteristics of the device.

SPRU732 — ***TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide*** describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C64x and TMS320C64x+ digital signal processors (DSPs) of the TMS320C6000 DSP family. The C64x/C64x+ DSP generation comprises fixed-point devices in the C6000 DSP platform. The C64x+ DSP is an enhancement of the C64x DSP with added functionality and an expanded instruction set.

SPRUEK5 — ***TMS320DM647/DM648 DSP DDR2 Memory Controller User's Guide*** describes the DDR2 memory controller in the TMS320DM647/DM648 Digital Signal Processor (DSP). The DDR2/mDDR memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices and standard Mobile DDR SDRAM devices.

SPRUEK6 — ***TMS320DM647/DM648 DSP External Memory Interface (EMIF) User's Guide*** describes the operation of the asynchronous external memory interface (EMIF) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The EMIF supports a glueless interface to a variety of external devices.

SPRUEK7 — ***TMS320DM647/DM648 DSP General-Purpose Input/Output (GPIO) User's Guide*** describes the general-purpose input/output (GPIO) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.

SPRUEK8 — ***TMS320DM647/DM648 DSP Inter-Integrated Circuit (I2C) Module User's Guide*** describes the inter-integrated circuit (I2C) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The I2C peripheral provides an interface between the DSP and other devices compliant with the I2C-bus specification and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the DSP through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

SPRUEL0 — ***TMS320DM647/DM648 DSP 64-Bit Timer User's Guide*** describes the operation of the 64-bit timer in the TMS320DM647/DM648 Digital Signal Processor (DSP). The timer can be configured as a general-purpose 64-bit timer, dual general-purpose 32-bit timers, or a watchdog timer.

SPRUEL1 — ***TMS320DM647/DM648 DSP Multichannel Audio Serial Port (McASP) User's Guide*** describes the multichannel audio serial port (McASP) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The McASP functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT).

SPRUEL2 — ***TMS320DM647/DM648 DSP Enhanced DMA (EDMA) Controller User's Guide*** describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM647/DM648 Digital Signal Processor (DSP). The EDMA3 controller’s primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DSP.

SPRUEL4 — ***TMS320DM647/DM648 DSP Peripheral Component Interconnect (PCI) User's Guide*** describes the peripheral component interconnect (PCI) port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The PCI port supports connection of the C642x DSP to a PCI host via the integrated PCI master/slave bus interface. The PCI port interfaces to the DSP via the enhanced DMA (EDMA) controller. This architecture allows for both PCI master and slave transactions, while keeping the EDMA channel resources available for other applications.

SPRUEL5 — ***TMS320DM647/DM648 DSP Host Port Interface (UHPI) User's Guide*** describes the host port interface (HPI) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The HPI is a parallel port through which a host processor can directly access the CPU memory space. The host device functions as a master to the interface, which increases ease of access. The host and CPU can exchange information via internal or external memory. The host also has direct access to memory-mapped peripherals. Connectivity to the CPU memory space is provided through the enhanced direct memory access (EDMA) controller.

SPRUEL8 — ***TMS320DM647/DM648 DSP Universal Asynchronous Receiver/Transmitter (UART) User's Guide*** describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The UART peripheral performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data received from the CPU.

SPRUEL9 — ***TMS320DM647/DM648 DSP VLYNQ Port User's Guide*** describes the VLYNQ port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The VLYNQ port is a high-speed point-to-point serial interface for connecting to host processors and other VLYNQ compatible devices. It is a full-duplex serial bus where transmit and receive operations occur separately and simultaneously without interference.

SPRUEM1 — ***TMS320DM647/DM648 DSP Video Port/VCXO Interpolated Control (VIC) Port User's Guide*** discusses the video port and VCXO interpolated control (VIC) port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The video port can operate as a video capture port, video display port, or transport channel interface (TCI) capture port. The VIC port provides single-bit interpolated VCXO control with resolution from 9 bits to up to 16 bits. When the video port is used in TCI mode, the VIC port is used to control the system clock, VCXO, for MPEG transport channel.

SPRUEM2 — ***TMS320DM647/DM648 DSP Serial Port Interface (SPI) User's Guide*** discusses the Serial Port Interface (SPI) in the TMS320DM647/DM648 Digital Signal Processor (DSP). This reference guide provides the specifications for a 16-bit configurable, synchronous serial peripheral interface. The SPI is a programmable-length shift register, used for high speed communication between external peripherals or other DSPs.

SPRUEU6 — **TMS320DM647/DM648 DSP Subsystem User's Guide** describes the subsystem in the TMS320DM647/DM648 Digital Signal Processor (DSP). The subsystem is responsible for performing digital signal processing for digital media applications. The subsystem acts as the overall system controller, responsible for handling many system functions such as system-level initialization, configuration, user interface, user command execution, connectivity functions, and overall system control.

SPRUF57 — **TMS320DM647/DM648 DSP 3 Port Switch (3PSW) Ethernet Subsystem User's Guide** describes the operation of the 3 port switch (3PSW) ethernet subsystem in the TMS320DM647/DM648 Digital Signal Processor (DSP). The 3 port switch gigabit ethernet subsystem provides ethernet packet communication and can be configured as an ethernet switch (DM648 only). It provides the serial gigabit media independent interface (SGMII), the management data input output (MDIO) for physical layer device (PHY) management.

Trademarks

Universal Asynchronous Receiver/Transmitter (UART)

1 Introduction

This document describes the universal asynchronous receiver/transmitter (UART) peripheral.

1.1 Purpose of the Peripheral

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 16x or 13x 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

The UART peripheral has the following features:

- Programmable baud rates (frequency pre-scale values from 1 to 65535).
- Fully programmable serial interface characteristics:
  - 5, 6, 7, or 8-bit characters.
  - Even, odd, or no PARITY bit generation and detection.
  - 1, 1.5, or 2 STOP bit generation.
- 16-byte depth transmitter and receiver FIFOs:
  - The UART can be operated with or without the FIFOs.
  - 1, 4, 8, or 14 byte selectable receiver FIFO trigger level for autoflow control and DMA.
- DMA signaling capability for both received and transmitted data.
- CPU interrupt capability for both received and transmitted data.
- False START bit detection.
- Line break generation and detection.
- Internal diagnostic capabilities:
  - Loopback controls for communications link fault isolation.
  - Break, parity, overrun, and framing error simulation.
- Programmable autoflow control using RTS and CTS signals.
- Modem control functions (CTS, RTS).

Table 1 shows the capabilities supported on each instance.

Table 1. UART Supported Features/Characteristics by Instance

Feature	Support
5, 6, 7 or 8-bit characters	Yes
Even, odd, or no PARITY bit	Yes
1, 1.5, or 2 STOP bit generation	Yes
Line break generation and detection	Yes
Internal loop back	Yes
DMA sync events for both received and transmitted data	Yes
1, 4, 8, or 14 byte selectable receiver FIFO trigger level	Yes
Polling/Interrupt	Yes
Max speed 128 kbps	Yes
Modem control features (CTS, RTS)	Yes
Autoflow control using CTS/RTS	Yes
DTR and DSR	No
Ring indication	No
Carrier detection	No
Single-character transfer mode (mode 0) in DMA mode	No

1.3 Functional Block Diagram

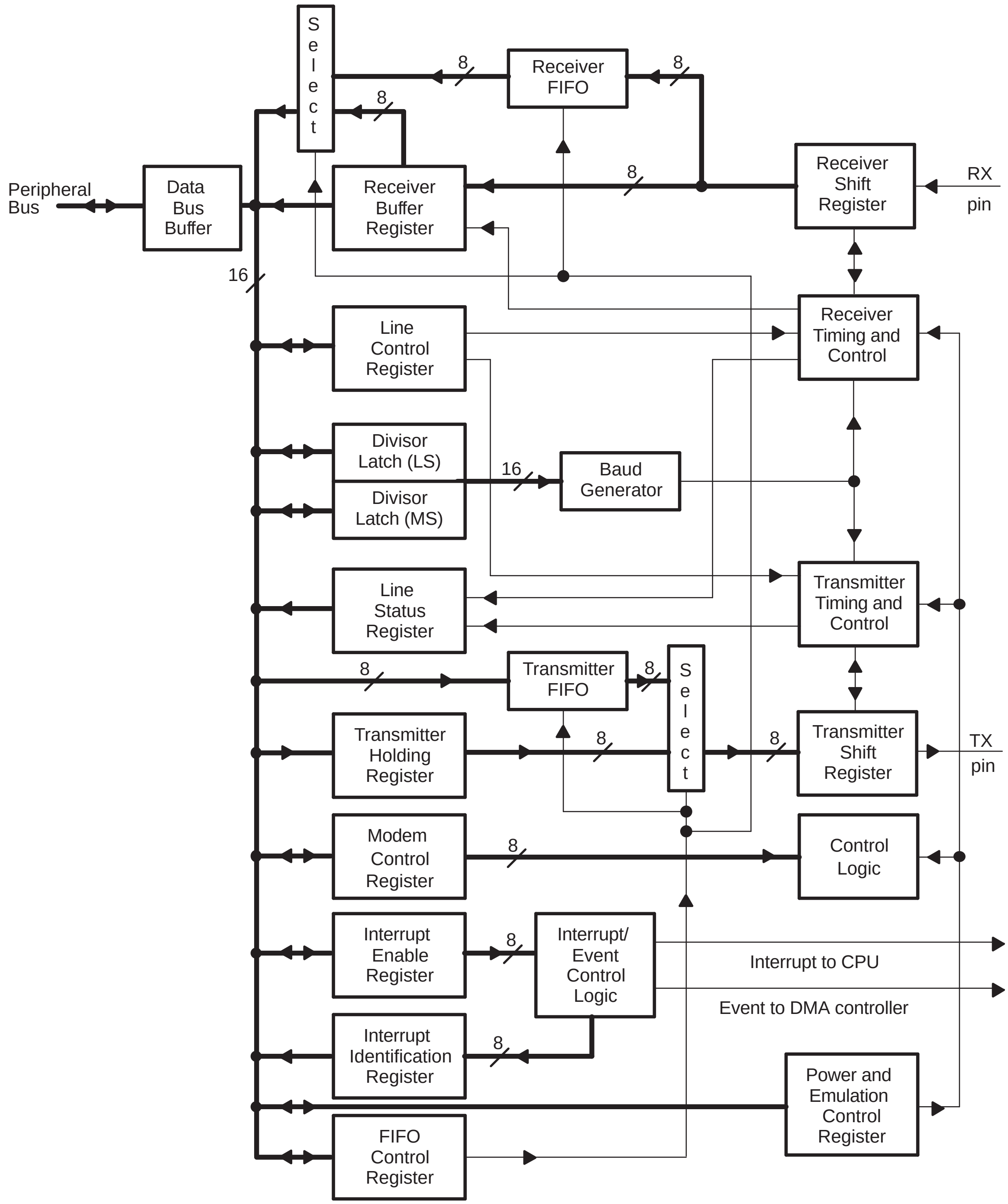
A functional block diagram of the UART is shown in Figure 1.

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. Any deviations in supported functions are indicated in Table 1.

The information in this document assumes the reader is familiar with these standards.

Figure 1. UART Block Diagram



2 Peripheral Architecture

2.1 Clock Generation and Control

The UART bit clock is equal to 1/6 of CPU clock. It supports up to 384 000 bps maximum data rate. By default, the clock to this module is disabled (in order to save power). To use UART user should enable the clock to this module by programming the appropriate PSC registers.

Figure 2 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 X) or thirteen times (13 X) the baud rate; each received or transmitted bit lasts 16 or 13 BCLK cycles. When the UART is receiving, the bit is sampled in the 8th BCLK cycle (for OSM_SEL=0). The formula to calculate the divisor is:

Divisor = (UART input clock frequency)/ (Desired baud rate X 16) [for osm_sel=0]

Divisor = (UART input clock frequency)/ (Desired baud rate X 13) [for 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. 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 3 summarizes the relationship between the transferred data bit, BCLK, and the UART input clock.

Example baud rates and divisor values relative to a 27-MHz UART input clock are shown in Table 2. In the said table, examples of 83MHz and 117 MHz input clock frequencies are considered. Since UART bit clock is equal to 1/6 of CPU clock, we derive for the examples in Table 2; CPU is running at 83 * 6= 500 MHz(approximately) and 117 *6 = 700 MHz(approx). So, for a CPU running at 'X' MHz, UART clock is equal to 1/6 of 'X'. And thus the divisor value to be programmed is different for different clock frequencies. The divisor calculation formula in Section 2.1 is to be used for the calculation of divisor value for the given UART clock frequency(which is 1/6 of CPU clock frequency), desired baud rate, and desired over sampling mode.

Figure 2. UART Clock Generation Diagram

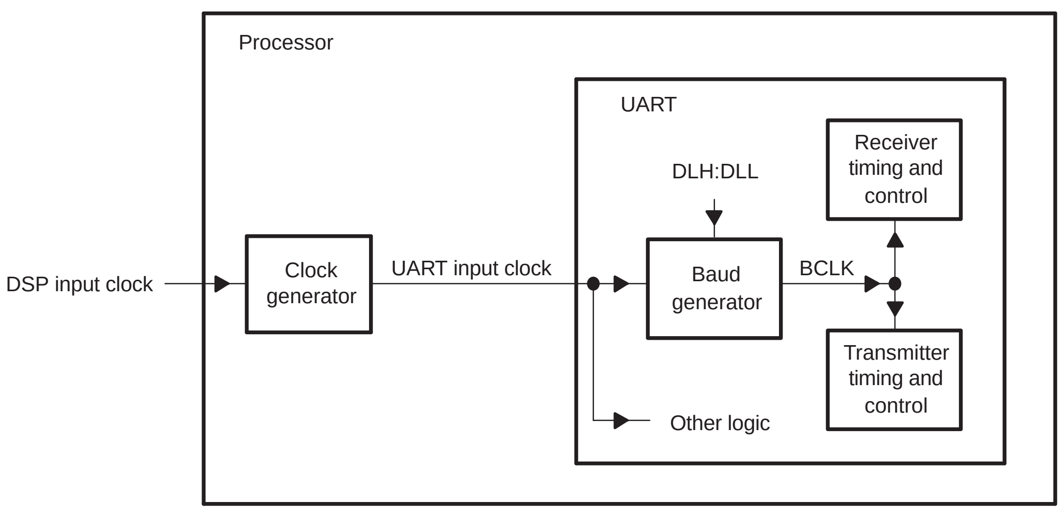


Figure 3. Relationships Between Data Bit, BCLK, and UART Input Clock(OSM_SEL=0)

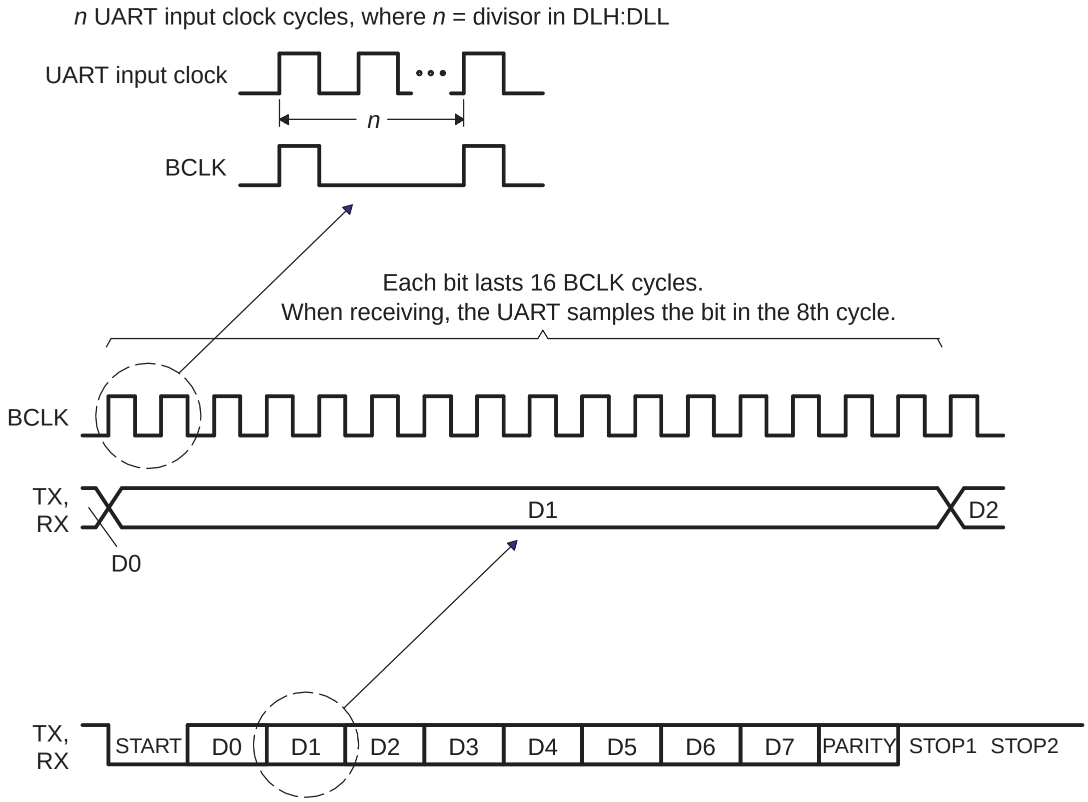


Table 2. UART baud rates with accuracy calculation with example frequencies of 83 and 117 MHz

Input clock	OSM_SEL	Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
83 MHz	0	2400	2170	2400.15361	0.0064
		4800	1085	4800.30722	0.0064
		9600	543	9591.774095	0.0857
		19200	271	19218.94219	0.0987
		38400	136	38296.56863	0.02694
		56000	93	56003.58423	0.0064
		128000	41	127032.5203	0.7558
	1	2400	2671	2399.946241	0.0022
		4800	1335	4801.690195	0.0352
		9600	668	9596.192231	0.0397
		19200	334	19192.38446	0.0397
		38400	167	38384.76892	0.0397
		56000	114	56230.31939	0.4113
		128000	50	128205.1282	0.1603

Table 2. UART baud rates with accuracy calculation with example frequencies of 83 and 117 MHz (continued)

Input clock	OSM_SEL	Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
117 MHz	0	2400	3038	2400.15361	0.0064
		4800	1519	4800.30722	0.0064
		9600	760	9594.298246	0.0594
		19200	380	19188.59649	0.0594
		38400	190	38377.19298	0.0594
		56000	130	56089.74359	0.1603
		128000	57	127923.9766	0.0594
		384000	19	383771.9298	0.0594
		115200	63	115740.7407	0.4694
	1	2400	3739	2400.202989	0.0085
		4800	1870	4799.122446	0.0183
		9600	935	9598.244892	0.0183
		19200	467	19217.04277	0.0888
		38400	234	38351.96143	0.1251
		56000	160	56089.74359	0.1603
		128000	70	128205.1282	0.1603
		384000	23	390189.5206	1.6119
		115200	78	115055.8843	0.1251

2.2 Signal Descriptions

UART supports flow control and the associated signals (UARTCTS and UARTRTS). The UART signal descriptions are included in Table 3.

Table 3. UART Signal Descriptions

Signal Name	Signal Type	Function
UARTTX	Output	Serial data transmit
UARTRX	Input	Serial data receive
UARTCTS	Input	Clear-to-Send handshaking signal
UARTRTS	Output	Request-to-Send handshaking signal

2.3 Pin Multiplexing

UART pins are multiplexed with SPI pins' functions.

For these signals to be used for UART functions, the pin multiplexing must be configured appropriately. For details on the pin multiplexing and configuration, see the device specific data manual.

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 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.

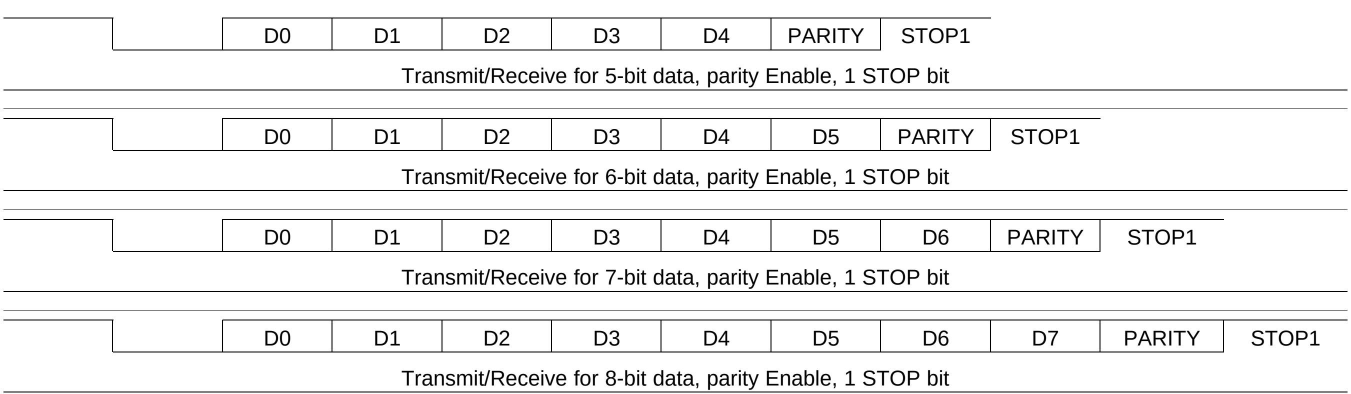
The UART receives in the following format:

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

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 4

Figure 4. UART Protocol Formats



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 START bit
- 5, 6, 7, or 8 data bits

- 1 PARITY bit (optional)
- 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 TX 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 16x 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 RX 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.
- 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 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 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 RCVR 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.

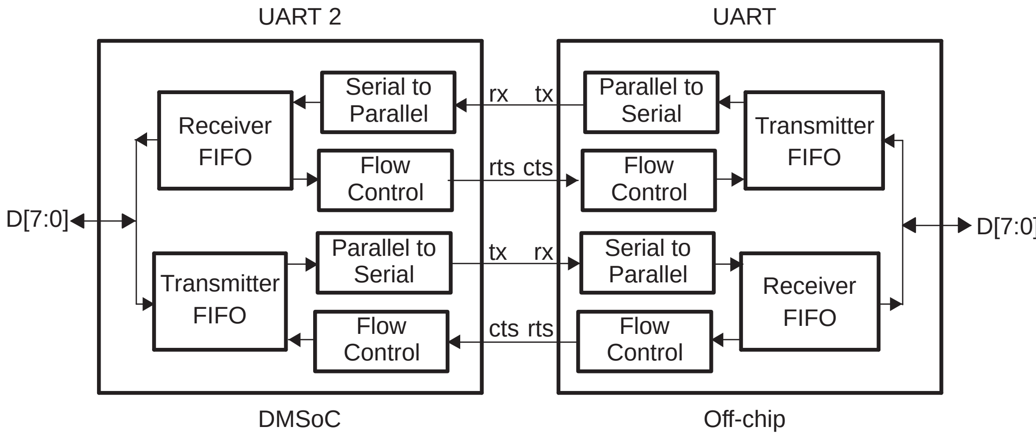
Also, in the FIFO poll mode:

- 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

UART can employ autoflow control by connecting the CTS and RTS signals. The CTS input must be active before the transmitter FIFO can transmit data. The RTS becomes active when the receiver needs more data and notifies the sending device. When RTS is connected to 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 5 with autoflow enabled, overrun errors are eliminated.

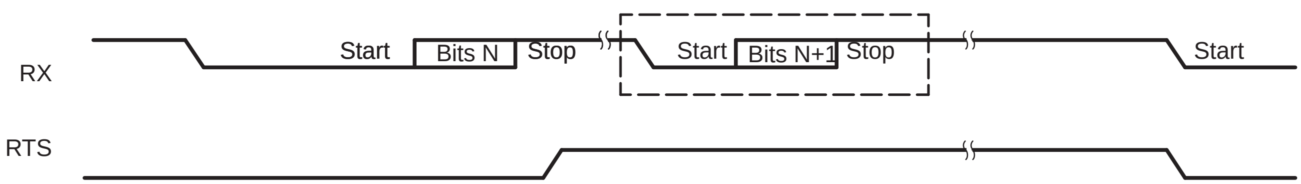
Figure 5. UART Interface Using Autoflow Diagram



2.5.4.1 RTS Behavior

RTS data flow control originates in the receiver block (see Figure 1). When the receiver FIFO level reaches a trigger level of 1, 4, 8, or 14 (see Figure 6), RTS is de-asserted. 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 de-assertion of RTS until after it has begun sending the additional byte. For trigger level 1, 4, and 8, RTS is automatically reasserted once the receiver FIFO is emptied. For trigger level 14, RTS is automatically reasserted once the receiver FIFO drops below the trigger level.

Figure 6. Autoflow Functional Timing Waveforms for RTS



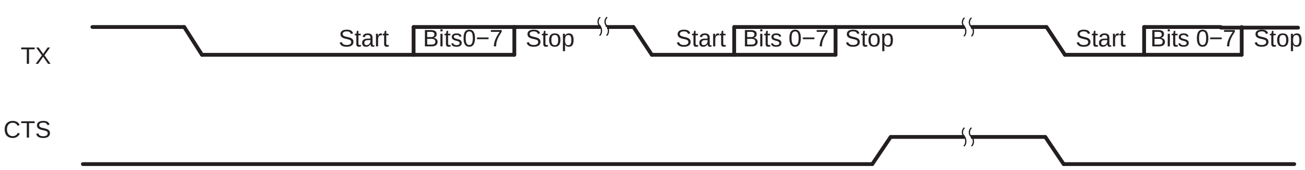
(1) N = Receiver FIFO trigger level.

(2) The two blocks in dashed lines cover the case where an additional byte is sent.

2.5.4.2 CTS Behavior

The transmitter checks CTS before sending the next data byte. If CTS is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, CTS must be released before the middle of the last STOP bit that is currently being sent (see Figure 7). When flow control is enabled, 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 7. Autoflow Functional Timing Waveforms for CTS



(1) When CTS is active (low), the transmitter keeps sending serial data out.

(2) When 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.

(3) When 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

Three 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.

These software bits affect only transmitter and receiver, and they don’t affect modem control signals. The control of modem signals are responsibility of software. RTS in auto-flow control mode must be taken care of especially. User must set RTS bit after setting URRST bit. Otherwise, RTS output goes to low indicating the UART is ready to receive data while the receiver of the UART is actually disabled.

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 Section 3.

Following a hardware reset, the UART reset bits (UTRST and URRST) in the power and emulation management register (PWREMU_MGMT) must be set to 1 to bring the UART out of reset. Since the divisor latch is not affected during reset, the BAUD rate clock, should be programmed before initiating serial communications.

2.7 Initialization

The following steps are required to initialize the UART:

- Set the desired baud rate by writing the appropriate clock divisor values to the divisor latch registers (DLL and DLH).
- 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.
- Choose the desired protocol settings by writing the appropriate values to the line control register (LCR).
- If autoflow control is desired, write appropriate values to the modem control register (MCR).
- 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 5. All requests are multiplexed through an arbiter to a single UART interrupt request to the CPU, as shown in Figure 8. Each of the interrupt requests has an enable bit in the interrupt enable register (IER) and is recorded in the interrupt identification register (IIR).

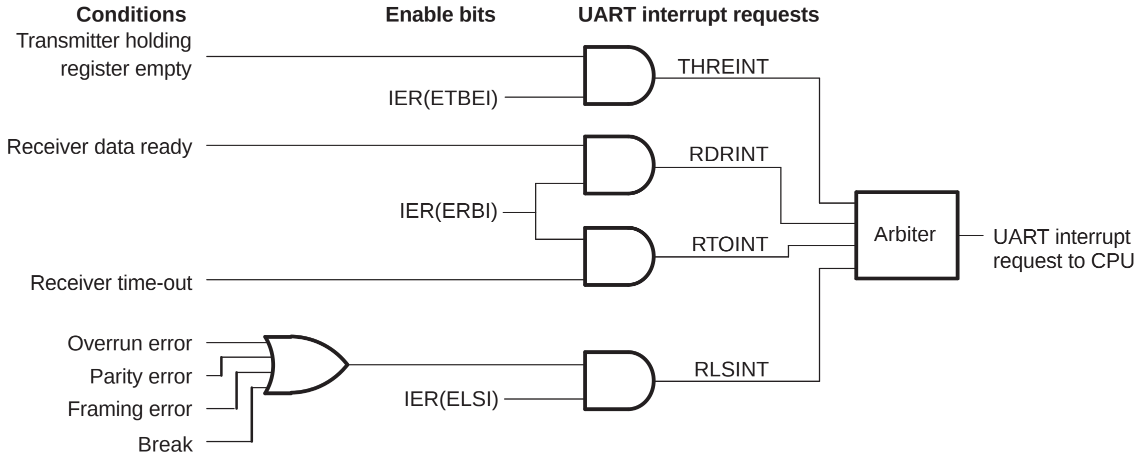
If an interrupt occurs and the corresponding enable bit is set to 1, the interrupt request is recorded in IIR and is forwarded to the CPU. If an interrupt occurs and the corresponding enable bit is cleared to 0, the interrupt request is blocked. The interrupt request is neither recorded in IIR nor forwarded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

If an interrupt source happens when other interrupt is pending, the interrupt signal is not asserted for the interrupt. That is, UART sends only one interrupt to CPU even if there are multiple interrupts pending. So when an interrupt is serviced, IIR[0] (Interrupt Pending Bit) should be read in order to check whether there is any other interrupt pending.

Table 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 only indicates 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 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, 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 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 8. UART Interrupt Request Enable Paths



2.9 DMA Event Support

There is only one mode of DMA signaling, the multi-character (or block) DMA transfer with the FIFO enabled. If the FIFOs are disabled (FCR[0] = 0), no DMA signals will be asserted. Please note that URXEVT is not asserted if the data at the top of the receiver FIFO is erroneous even if the trigger level has been reached. 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 RXFIFOTL 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). Please note that URXEVT is not asserted if the data at the top of the receiver FIFO is erroneous even if the trigger level has been reached.
- **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 device PSC document.

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 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 and continues operating normally.

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.

3 Registers

The system programmer has access to and control over any of the UART registers that are listed in Table 6. 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 6. 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
20h	DLL	Divisor LSB Latch	Section 3.9
24h	DLH	Divisor MSB Latch	Section 3.9
28h	PID	Peripheral Identification Register	Section 3.10
30h	PWREMU_MGMT	Power and Emulation Management Register	Section 3.11
34h	MDR	Mode Definition Register	Section 3.12

3.1 Receiver Buffer Register (RBR)

The receiver buffer register (RBR) is shown in Figure 9 and described in Table 7.

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 16x receiver clock. Receiver section control is a function of the line control register (LCR).

RSR receives serial data from the RX pin. Then RSR concatenates the data and moves it into 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 (ERBI = 1 in 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.

Access considerations:

RBR, THR, and DLL share one address. To read RBR, write 0 to the DLAB bit in LCR, and read from the shared address. When DLAB = 0, writing to the shared address modifies THR. When DLAB = 1, all accesses at the shared address read or modify DLL.

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

Figure 9. Receiver Buffer Register (RBR)

31–16
Reserved
R-0

15–8	7–0
Reserved	RBR
R-0	R-0

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

Table 7. Receiver Buffer Register (RBR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7-0	RBR	0-FFh	Received data.

3.2 *Transmitter Holding Register (THR)*

The transmitter holding register (THR) is shown in Figure 10 and described in Table 8.

The UART transmitter section consists of 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 line control register (LCR).

THR receives data from the internal data bus and when TSR is idle, the UART moves the data from THR to TSR. The UART serializes the data in TSR and transmits the data on the TX pin. In the non-FIFO mode, if THR is empty and the THR Empty (THRE) interrupt is enabled (ETBEI = 1 in 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.

Access considerations:

RBR, THR, and DLL share one address. To load THR, write 0 to the DLAB bit of LCR, and write to the shared address. When DLAB = 0, reading from the shared address gives the content of RBR. When DLAB = 1, all accesses at the address read or modify DLL.

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

Figure 10. Transmitter Holding Register (THR)

31–16
Reserved
R-0

15–8	7–0
Reserved	THR
R-0	W-0

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

Table 8. Transmitter Holding Register (THR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7-0	THR	0-FFh	Data to transmit.

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 IER is forwarded to the CPU. IER is shown in Figure 11 and described in Table 9.

Access considerations:

IER and DLH share one address. To read or modify IER, write 0 to the DLAB bit in LCR. When DLAB = 1, all accesses at the shared address read or modify DLH.

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

Figure 11. Interrupt Enable Register (IER)

31–16
Reserved
R-0

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

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

Table 9. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved.
3	Reserved	0	Reserved. This bit must always be written with a 0.
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.

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), IIR indicates that an interrupt is pending in the IPEND bit and encodes the type of interrupt in the INTID bits. IIR is shown in Figure 12 and described in Figure 12.

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

Access consideration:

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

Figure 12. 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 10. Interrupt Identification Register (IIR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7-6	FIFOEN	0-3h	FIFOs enabled.
		0	Non-FIFO mode.
		1h-2h	Reserved.
		3h	FIFOs are enabled. FIFOEN bit in the FIFO control register (FCR) is set to 1.
5-4	Reserved	0	Reserved.
3-1	INTID	0-7h	Interrupt type. See Table 11.
		0	Reserved.
		1h	Transmitter holding register empty (priority 3).
		2h	Receiver data available (priority 2).
		3h	Receiver line status (priority 1, highest).
		4h-5h	Reserved.
		6h	Character timeout indication (priority 2).
		7h	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.

Table 11. Interrupt Identification and Interrupt Clearing Information

Priority Level	IIR Bits 3	2	1	0	Interrupt Type	Interrupt Source	Event That Clears Interrupt
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 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 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).

[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 FCR to enable and clear the FIFOs and to select the receiver FIFO trigger level FCR is shown in Figure 13 and described in Table 12. The FIFOEN bit must be set to 1 before other FCR bits are written to or the FCR bits are not programmed.

Access consideration:

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 to the address modifies FCR.

Note: 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.

Figure 13. FIFO Control Register (FCR)

31–16
Reserved
R-0

15–8
Reserved
R-0

7–6	5–4	3	2	1	0
RXFIFOTL	Reserved	DMAMODE1(A)	TXCLR	RXCLR	FIFOEN
W-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

(A) 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 12. FIFO Control Register (FCR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7-6	RXFIFOTL	0-3h	Receiver FIFO trigger level. RXFIFOTL 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). Once the FIFO drops below the trigger level, the interrupt is cleared.
		0	1 byte.
		1h	4 bytes.
		2h	8 bytes.
		3h	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.
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.

3.6 Line Control Register (LCR)

The line control register (LCR) is shown in Figure 14 and described in Table 13.

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

Figure 14. 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/W = Read/Write; R = Read only; -*n* = value after reset

Table 13. Line Control Register (LCR) Field Descriptions

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 you use the dedicated addresses, you can keep DLAB = 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 where the UARTTX 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 14.
		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 14.
		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 14.
		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 13. Line Control Register (LCR) Field Descriptions (continued)

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 15.
		0	1 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.
		1h	6 bits.
		2h	7 bits.
		3h	8 bits.

Table 14. 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.

Table 15. Number of STOP Bits Generated

STB Bit	WLS Bits	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

3.7 Modem Control Register (MCR)

The modem control register (MCR) is shown in Figure 15 and described in Table 16. The modem control register provides the ability to enable/disable the autoflow functions, and enable/disable the loopback function for diagnostic purposes.

Figure 15. Modem Control Register (MCR)

31–16
Reserved
R-0

15–6	5	4	3–2	1	0
Reserved	AFE	LOOP	Reserved	RTS	Rsvd
R-0	R/W-0	R/W-0	R-0	R/W-0	R-0

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

Table 16. Modem Control Register (MCR) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved.
5	AFE		Autoflow control enable. Autoflow control allows the RTS and CTS signals to provide handshaking between UARTs during data transfer. When AFE = 1, the RTS bit determines the autoflow control enabled.
		0	Autoflow control is disabled.
		1	Autoflow control is enabled: • When RTS = 0, CTS is only enabled. • When RTS = 1, RTS and CTS are enabled.
4	LOOP		Loop back mode enable. LOOP is used for the diagnostic testing using the loop back feature.
		0	Loop back mode is disabled.
		1	Loop back mode is enabled. When LOOP is set, the following occur: • The UARTTX signal is set high. • The UARTRX pin is disconnected. • The output of the transmitter shift register (TSR) is lopped back in to the receiver shift register (RSR) input.
3-2	Reserved	0	Reserved
1	RTS		RTS control. When AFE = 1, the RTS bit determines the autoflow control enabled.
		0	RTS is disabled, CTS is only enabled.
		1	RTS and CTS are enabled.
0	Reserved	0	Reserved.

3.8 Line Status Register (LSR)

The line status register (LSR) is shown in Figure 16 and described in Table 17. LSR provides information to the CPU concerning the status of data transfers. LSR 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 16. Line Status Register (LSR)

31–16
Reserved
R-0

15–8	7	6	5	4	3	2	1	0
Reserved	RCVR	TEMT	THRE	BI	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 17. Line Status Register (LSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7	RCVR		Receiver FIFO error.
			In non-FIFO mode:
		0	There has been no error, or RCVR 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 RCVR 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. You can write to the transmitter FIFO 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 17. Line Status Register (LSR) Field Descriptions (continued)

Bit	Field	Value	Description
4	BI		Break indicator. The BI 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 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. Once 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 17. Line Status Register (LSR) Field Descriptions (continued)

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.

3.9 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 DLH and DLL. DLH 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.

Access considerations:

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

DLL and DLH also have dedicated addresses. If you use the dedicated addresses, you can keep the DLAB bit cleared, so that RBR, THR, and IER are always selected at the shared addresses.

The divisor LSB latch (DLL) is shown in Figure 17 and described in Table 18. The divisor MSB latch (DLH) is shown in Figure 18 and described in Table 19.

Figure 17. Divisor LSB Latch (DLL)

31–16
Reserved
R-0

15–8	7–0
Reserved	DLL
R-0	R/W-0

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

Table 18. 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.

Figure 18. Divisor MSB Latch (DLH)

31–16
Reserved
R-0

15–8	7–0
Reserved	DLH
R-0	R/W-0

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

Table 19. 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.

3.10 Peripheral Identification Register

This register is intended to store versioning information used to identify the specific UART peripheral implemented in a particular device. All bits within this register are read only. The values within this register are hard coded to match the reset states as shown below. PID is shown in Figure 19 and described in Table 20.

Figure 19. Peripheral Identification Register

31–30	29–28	27–16
SCHEME	Reserved	FUNC
R-01b	R-00b	R-414h

15–11	10–8	7–6	5–0
R	X	Z	Y
R-00000b	R-001b	R-00b	R-000010b

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

Table 20. Peripheral Identification Register Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	01b	UART follows the new PID scheme. This value is 0b01
29-28	Reserved	0	Reserved
27-16	FUNC	414h	UART's assigned FUNC value is set to 0x414
15-11	R	00000b	RTL version number
10-8	X	001b	UART major spec release
7-6	Z	00b	Custom number
5-0	Y	000010b	UART minor spec release

3.11 Power and Emulation Management Register (PWREMU_MGMT)

The power and emulation management register (PWREMU_MGMT) is shown in Figure 20 and described in Table 21.

Figure 20. Power and Emulation Management Register (PWREMU_MGMT)

31–16
Reserved
R-0

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

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

Table 21. Power and Emulation Management Register (PWREMU_MGMT) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved.
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	0	Reserved.
0	FREE		Free-running enable mode bit. This bit determines the emulation mode functionality of the UART. In suspended mode, 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 stops immediately. If a transmission is in progress, the UART stops after completion of the one word transmission.
		1	Free-running mode is enabled; UART continues to run normally.

3.12 Mode Definition Register (MDR)

The mode definition register (MDR) is shown in Figure 21 and described in Table 22.

Figure 21. Mode Definition Register (MDR)

31–16
Reserved
R-0

15–1	0
Reserved	OSM_SEL
R-0	R/W-0

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

Table 22. Mode Definition Register (MDR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved.
0	OSM_SEL		This bit determines over-sampling mode.
		0	16x over-sampling.
		1	13x over-sampling.