# UART Controller

## Introduction

The Universal Asynchronous Receiver Transmitter (UART) module is a controller for a full-duplex, asynchronous communication channel that communicates with peripheral devices and personal computers, using protocols such as RS-232, RS-485. The module also supports the hardware flow control option with CTS and RTS.

.

## Features

The primary features of the UART module are:

* Full-Duplex, 8-Bit Data Transmission through the TXD and RXD pins
* Even, Odd or No Parity options
* One or two Stop bits
* Hardware Flow Control option with CTS and RTS pins
* Fully Integrated Baud Rate Generator with 16-Bit Prescaler
* Baud Rates ranging from 1 Mbps to 15 bps at 16 MIPS
* 4-deep First-In-First-Out (FIFO) Transmit Data Buffer
* 4-deep FIFO Receive Data Buffer
* Parity, Framing and Buffer Overrun Error Detection
* Transmit and Receive Interrupts
* Loopback mode for Diagnostic Support

## Parameters

Top-level DMA controller parameters for integration:

* APB\_ADDR\_WIDTH – defines APB address bus width
* APB\_BASE\_ADDRESS – defines APB base address for registers access

# UART Configuration

The UART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight data bits and one or two Stop bits). Parity is supported by the hardware and may be configured by the user as even, odd or no parity.The number of Stop bits and the parity, are specified in the PDSEL[1:0] (MODE[1:0]) and STSEL (MODE[2]) bits. An on-chip, dedicated, 16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the input clock frequency . The UART transmits and receives the LSb first. The UART module’s transmitter and receiver are functionally independent but use the same data format and baud rate.

# Functional Description

## Block Diagram



**uart\_apb\_slave**

The module is responsible for bridging read/write CPU requests to operational registers located in uart\_regs through APB interface.

**uart\_regs**

The module contains software accessible registers, RxFIFO, TxFIFO, Baud Rate Generator, interrupt request generation and control interface logic with uart\_receiver and uart\_transmitter modules.

**uart\_transmitter**

The module implements UART transmit function. It contains shift register and transmit FSM logic.

**uart\_receiver**

The module implements UART receive function. It contains shift register and receive FSM logic.

## UART Baud Rate Generator (BRG)

The UART module includes a dedicated 16-bit Baud Rate Generator. The BRG register controls

the period of a free-running, 16-bit timer. Writing a new value to the BRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

Equation below shows the formula for computation of the baud rate with BRGH = ‘0’.



The maximum baud rate (BRGH = ‘0’) possible is FCY/16 (for BRG = 16’h0000), and the minimum baud rate possible is FCY/16 \* 65536).

Equation below shows the formula for computation of the baud rate with BRGH = ‘1’.



The maximum baud rate (BRGH = ‘1’) possible is FCY/4 (for BRG = 16’h0000), and the minimum baud rate possible is FCY/(4 \* 65536).

The example shows the calculation of the baud rate error for the following conditions:

* FCY = 4 MHz
* Desired Baud Rate = 9600



**Baud Rate table for BRGH = ‘0’**

## 

**Baud Rate table for BRGH = ‘1’**



## UART Transmitter

The heart of the transmitter is the Transmit Shift register (TSR). The Shift register obtains its data from the transmit FIFO buffer, TXBUF. The TXBUF register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXBUF register (if available).

The actual transmission will not occur until the TXBUF register has been loaded with data and the Baud Rate Generator (BRG) has produced a shift clock. Normally, when transmission is first started, the TSR register is empty, so a transfer to the TXBUF register will result in an immediate transfer to TSR.

The transmit buffer is 8 bits wide and 4 levels deep. Together with the Transmit Shift registers (TSR), the user effectively has a 5-level deep buffer. It is organized as First-In-First-Out (FIFO). Once the TXREG contents are transferred to the TSR register, the current buffer location becomes available for new data to be written and the next buffer location is sourced to the TSR register. The TXBF status bit is set whenever the buffer is full. If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO. The FIFO is reset during device Reset.

The Transmit Interrupt Flag (TXIF) is located in the Interrupt Flag (INTF) register. The TXISEL[1:0] control bits determine when the UART will generate a transmit interrupt.

1. TXISEL[1:0] = ‘00’ or ‘11’, the TXIF is set when a character is transferred from the transmit buffer to the Transmit Shift register (TSR). This implies at least one location is empty in the transmit buffer.
2. TXISEL[1:0] = ‘01’, the TXIF is set when the last character is shifted out of the Transmit Shift register (TSR). This implies that all the transmit operations are completed.
3. TXISEL[1:0] = ‘10’, the TXIF is set when the character is transferred to the Transmit Shift register (TSR) and the transmit buffer is empty.

Switching between the Interrupt modes during operation is possible.

While the TXIF flag bit indicates the status of the TXBUF register, the TRMT bit shows the status of the TSR. The TRMT status bit is a read-only bit, which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

A Break character transmit consists of a Start bit, followed by twelve bits of ‘0’ and a Stop bit. A

Frame Break character is sent whenever the TXBRK is set while the Transmit Shift register is loaded with data. A dummy write to the TXBUF register is necessary to initiate the Break character transmission. Note that the data value written to the TXBUF for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence – all ‘0’s will be transmitted.

The TXBRK bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break

character (typically, the Sync character in the LIN specification). The TRMT bit indicates when the Transmit Shift register is empty or full, just as it does during normal transmission.

## UART Receiver

The heart of the receiver is the Receive (Serial) Shift register (RSR). The data is received on the RXD pin and is sent to the data recovery block. The data recovery block operates at 16 times the baud rate, whereas the main receive serial shifter operates at the baud rate. After sampling the RXD pin for the Stop bit, the received data in RSR is transferred to the receive FIFO (if it is empty). The data on the RXD pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RXD pin.

The UART receiver has a 4-deep, 8-bit wide FIFO receive data buffer. RXBUF is a memory mapped register that provides access to the output of the FIFO. It is possible for 4 words of data to be received and transferred to the FIFO and a fifth word to begin shifting to the RSR register before a buffer overrun occurs.

If the FIFO is full (four characters) and a fifth character is fully received into the RSR register, the Overrun Error bit, OERR, will be set. The word in RSR will be kept, but further transfers to the receive FIFO are inhibited as long as the FIFO is full. The user must clear the OERR bit by reading the FIFO to empty state.

The Framing Error bit, FERR, is set if a Stop bit is detected at a logic low level.

The Parity Error bit, PERR, is set if a parity error has been detected in the data word at the top of the buffer (i.e., the current word). For example, a parity error would occur if the parity is set to be even, but the total number of ones in the data has been detected to be odd. The FERR and PERR bits are buffered along with the corresponding byte and should be read before reading the data word. An interrupt is generated if any of these (OERR, FERR and PERR) errors occur. The user will have to enable the corresponding Interrupt Enable Control bit ERIE to go to the corresponding interrupt vector location.

The UART Receive Interrupt Flag (RXIF) located in the Interrupt Flag (INTF) register. The RXISEL[1:0] control bits determine when the UART receiver generates an interrupt.

1. If RXISEL[1:0] = ‘00’ or ‘01’, an interrupt is generated each time a data word is transferred

from the Receive Shift register (RSR) to the receive buffer. There may be one or more characters in the receive buffer.

1. If RXISEL[1:0] = ‘10’, an interrupt is generated when a word is transferred from the Receive Shift register (RSR) to the receive buffer, and as a result, the receive buffer contains 3 or 4 characters.
2. If RXISEL[1:0] = ‘11’, an interrupt is generated when a word is transferred from the Receive Shift register (RSR) to the receive buffer, and as a result, the receive buffer contains 4 characters (i.e., becomes full).

Switching between the Interrupt modes during operation is possible.

The data in the receive FIFO should be read prior to clearing the OERR bit. The FIFO is reset when OERR is cleared which causes all data in the buffer to be lost.

While the RXDA and RXIF flag bits indicate the status of the RXBUF register, the RIDLE bit shows the status of the RSR register. The RIDLE status bit is a read-only bit which is set when the receiver is Idle (i.e., the RSR register is empty). No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the RSR is Idle. The URXDA bit indicates whether the receive buffer has data or whether the buffer is empty. This bit is set as long as there is at least one character to be read from the receive buffer. RXDA is a read-only bit.

## UART Loopback Mode

Setting the LPBACK bit enables this special mode in which the TXD and RTS outputs are internally connected to the RXD and CTS inputs respectively. When configured for the Loopback mode, the RXD, CTS pins are disconnected from the internal UART receive logic. However, the TXD and RTS pins still function normally.

To select this mode:

1. Configure UART for the desired mode of operation.

2. Enable transmission

3. Set LPBACK = ‘1’ to enable Loopback mode.

## UART Flow Control Mode

In the Flow Control mode, the RTS pins of two UART devices are connected to the CTS pins of as shown in the figure.

In the mode, (FCE bit = ‘1’) the RTS signal indicates that the device is ready to receive the data.

The RTS pin is asserted (driven low) whenever the receiver is ready to receive data.

When the FCE bit = ‘1’ (when the device is in Flow Control mode), the RTS pin is driven low whenever the receive buffer is not full or the OERR bit is not set. When the FCE bit = ‘1’, the RTS pin is driven high whenever the device is not ready to receive (i.e., when the receiver buffer is either full or in the process of shifting). Since the RTS is connected to the CTS of remote device, the RTS will drive the CTS low whenever it is ready to receive the data. Transmission of the data will begin when the CTS goes low. The user can also read the status of the CTS pin by reading the CTS bit.

.

## Interfaces

UART Controller has the following interfaces:

* APB slave interface for registers access from CPU side
* External interface to remote UART device

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal Name** | **Size** | **Direction** | **Description** |  |
| CLK | 1 | I | clock signal |  |
| RST\_N | 1 | I | low active reset signal, deasserted synchronously to rising edge of clock |  |
|  |  |  |  |  |
| TRIGGERS | 64 | I | Trigger bus |  |
|  |  |  |  |  |
| PADDR | 32 | I | APB address bus |  |
| PSEL | 1 | I | APB select |  |
| PENABLE | 1 | I | APB strobe |  |
| PWRITE | 1 | I | APB transfer direction |  |
| PWDATA | 32 | I | APB write data bus |  |
| PRDATA | 32 | O | APB read data bus |  |
| PREADY | 1 | O | APB ready signal |  |
| PSLVERR | 1 | O | APB transfer failure signal |  |
|  |  |  |  |  |
| TXD | 1 | O |  |  |
| RXD | 1 | I |  |  |
| RTS | 1 | O |  |  |
| CTS | 1 | I |  |  |

## Registers

Read or write access to the registers has to be of size 1, 2 or 4 bytes.

A device Reset forces all registers to their Reset state.

## INTF Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:3 | - | RO | Unimplemented bits. Write has no effect. Always reads as ’0’. | all ‘0’ |
| 2 | ERIF | R/W1C | Receive Error Interrupt Flag bit  ‘1’ – Receive error source is active  ‘0’ – Receive error source is not active | ‘0’ |
| 1 | RXIF | R/W1C | Receiver Interrupt Flag bit  ‘1’ – Receive Interrupt source is active  ‘0’ – Receive Interrupt source is not active | ‘0’ |
| 0 | TXIF | R/W1C | Transmit Interrupt Flag bit  ‘1’ – Transmit Interrupt source is active  ‘0’ – Transmit Interrupt source is not active | ‘0’ |

## INTC Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:6 | - | RO | Unimplemented bits. Write has no effect. Always reads as ’0’. | all ‘0’ |
| 2 | ERIE | R/W | Receive Error Interrupt Enable bit  ‘1’ - Setting ERIF to ‘1’ asserts channel interrupt line  ‘0’ - Setting ERIF to ‘1’ has no effect on channel interrupt line | ‘0’ |
| 1 | RXIE | R/W | Receive Interrupt Enable bit  ‘1’ - Setting RXIF to ‘1’ asserts channel interrupt line  ‘0’ - Setting RXIF to ‘1’ has no effect on channel interrupt line | ‘0’ |
| 0 | TXIE | R/W | Transmit Interrupt Enable bit  ‘1’ - Setting TXIF to ‘1’ asserts channel interrupt line  ‘0’ - Setting TXIF to ‘1’ has no effect on channel interrupt line | ‘0’ |

## MODE Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:16 | BRG | R/W | BRG Constant | all ‘0’ |
| 15 | TXINV | R/W | Transmit Polarity Inversion bit  ‘1’ – TXD idle state is ‘0’  ‘0’ – TXD idle state is ‘1’ | ‘0’ |
| 14 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | ‘0’ |
| 13:12 | TXISEL | R/W | Transmit Interrupt Mode Selection bits  ‘11’ – same as ‘00’  ‘10’ – Interrupt generated when a character is transferred to the Transmit  Shift register and the transmit buffer becomes empty  ‘01’ – Interrupt generated when the last transmission is over  (last character shifted out of Transmit Shift register) and all the  transmit operations are completed  ‘00’– Interrupt generated when any character is transferred to the  Transmit Shift Register  (this implies at least one location is empty in the transmit buffer) | ‘00’ |
| 11 | RXINV | R/W | Receive Polarity Inversion bit  ‘1’ – RXD idle state is ‘0’  ‘0’ – RXD idle state is ‘1’ | ‘0’ |
| 10 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | ‘0’ |
| 9:8 | RXISEL | R/W | Receive Interrupt Mode Selection bits  ‘11’ – Interrupt flag bit is set when receive buffer is full (4 bytes)  ‘10’ – Interrupt flag bit is set when receive buffer is 3/4 full (3 bytes) ‘0x’ – Interrupt flag bit is set when a character is received | ‘00’ |
| 7:6 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | all ‘0’ |
| 5 | LPBACK | R/W | Loopback Mode Select bit  ‘1’ – Enable Loopback Mode  ‘0’ – Loopback Mode is disabled | ‘0’ |
| 4 | FCE | R/W | flow control enable bit (use RTS and CTS pins)  ‘1’ – Flow control on  ‘0’ – Flow control off | ‘0’ |
| 3 | BRGH | R/W | High Baud Rate Select bit  ‘1’ – High speed  ‘0’ – Low speed | ‘0’ |
| 2 | STSEL | R/W | Stop Selection bit  ‘1’ – 2 Stop bits  ‘0’ – 1 Stop bit | ‘0’ |
| 1:0 | PDSEL | R/W | Parity and Data Selection bits  ‘11’ – 8 bit data, no parity  ‘10’ – 8 bit data, odd parity  ‘01’ – 8 bit data, even parity  ‘00’ – 8 bit data, no parity | ‘00’ |

## STATUS Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:9 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | all ‘0’ |
| 8 | TXBRK | R/W | Break bit  ‘1’ – TXD pin is driven low regardless of transmitter state  (Sync Break transmission – Start bit followed by twelve ‘0’s and  followed by a Stop bit)  ‘0’ – Sync Break transmission is disabled or completed | ‘0’ |
| 7 | CTS | RO | CTS pin Status bit.  ‘1’ – Not active, ‘0’ – Active |  |
| 6 | TRMT | RO | Transmit Shift Register is Empty bit  ‘1’ – Transmit Shift register is empty and transmit buffer is empty  (the last transmission has completed)  ‘0’ – Transmit Shift register is not empty, a transmission is in progress  or queued in the transmit buffer | ‘0’ |
| 5 | TXBF | RO | Transmit Buffer Full Status bit  ‘1’ – Transmit buffer is full  ‘0’ – Transmit buffer is not full | ‘0’ |
| 4 | RIDLE | RO | Receiver Idle bit  ‘1’ – Receiver is Idle  ‘0’ – Data is being received | ‘0’ |
| 3 | PERR | RO | Parity Error Status bit  ‘1’ – Parity error has been detected for the current character  ‘0’ – Parity error has not been detected | ‘0’ |
| 2 | FERR | RO | Framing Error Status bit  ‘1’ – Framing error has been detected for the current character  ‘0’ – Framing error has not been detected |  |
| 1 | OERR | RO | Receive Buffer Overrun Error Status bit  ‘1’ – Receive buffer has overflowed  ‘0’ – Receive buffer has not overflowed (reading RXBUF to empty state resets OERR bit) | ‘0’ |
| 0 | RXDA | RO | Receive Buffer Data Available bit  ‘1’ – Receive buffer has data, at least one more character can be read  ‘0’ – Receive buffer is empty | ‘0’ |

## RXBUF Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:8 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | all ‘0’ |
| 7:0 | RXBUF | R/W | Receive buffer | - |

## TXBUF Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits** | **Field Name** | **Type** | **Description** | **POR** |
| 31:8 | - | RO | Unimplemented bits. Write has no effect. Always reads as ‘0’. | all ‘0’ |
| 7:0 | TXBUF | R/W | Transmit buffer | - |

# Programming

Steps to follow when setting up a transmission:

1. Initialize the BRG register for the appropriate baud rate.
2. Set the number of Stop bits and parity selection by writing to the PDSEL[1:0] and STSEL bits.
3. If transmit interrupts are desired, set the TXIE control bit in the INTC register. Select the Transmit Interrupt mode by writing the TXISEL[1:0] bits.
4. Load data to the TXBUF register (starts transmission). Data can be loaded into the buffer until the TXBF status bit is set.

Steps to follow when setting up a reception:

1. Initialize the UxBRG register for the appropriate baud rate.
2. Set the number of Stop bits and parity selection by writing to the PDSEL[1:0] and STSEL bits.
3. If interrupts are desired, set the RXIE, ERIE control bits in the INTC register. Select the Transmit Interrupt mode by writing the RXISEL[1:0] bits.
4. Receive interrupts will depend on the RXISEL[1:0] control bit settings. If receive interrupts are not enabled, the user can poll the RXDA bit. The RXIF bit should be cleared in the software routine that services the UART receive interrupt.
5. Read data from the receive buffer. The RXDA status bit will be set whenever data is available in the buffer.