UART

Introduction

Universal Asynchronous Receiver/Transmitter (UART) is a standard serial communication protocol for exchanging data between two devices. In this communication protocol, data is transferred sequentially, one bit at a time. This implementation uses a frame consisting of 8 data bits, one start bit, one optional parity bit, and one or more stop bits. A parity bit is a bit, with a value of 0 or 1, that is added to a block of data for error detection purposes. This bit is optional; it may or may not be added to the data payload. It can also be set either to odd or even. These bits are often used in data transmission to ensure that data is not corrupted during the transfer process. If the data transmission protocol is set to an odd parity, each data packet must have an odd parity. If it is set to even, each packet must have an even parity. If a packet is received with the wrong parity, an error will be produced and the data will need to be retransmitted. The parity bit for each data packet is computed before the data is transmitted.

Because the UART protocol is asynchronous, it does not need a clock signal. In UART communication, speed is defined by the baud rate. The baud rate is equal to the number of bits transmitted per second including the start and stop bits. For reliable data transmission and reception without any loss of bits, both the transmitter and receiver should have the same baud rate. Mismatch in baud rates typically results in framing error. Common baud rates are 4800, 9600, 19200, 38400, 57600, and 115200 but other rates may also be used.

Features and Overview

- Asynchronous receiver and transmitter
- Data framing consists of start, optional parity, and stop bits
- Optional interrupt on receive register full and transmit buffer empty
- Parity, overrun, and framing error detection
- High level transmit and receive functions

Functional Description

The UART User Module implements a serial transmitter and receiver. The UART maps onto two different Digital Communication blocks designated TX and RX. The TX-block gives transmitter functionality and the RX-block gives receiver functionality. RX and TX operate independently. Each has their own Control and Status register, programmable interrupts, I/O, Buffer register, and Shift register. They share the same enable, clock, and data format.
Setting the Enable bit in the RX Control and TX Control registers enables the UART for operation. The UART User Module clock is shared by both the RX and TX components. The clock frequency selected must be eight times the frequency of the required data bit rate. Each received or transmitted data bit requires eight input clock cycles. The data received and transmitted is a bit stream that consists of a start bit, eight data bits, an optional parity bit, and a stop bit. The parity may be set to none, even, or odd, and both RX and TX are set to the same parity configuration.

TX - UART Transmitter

The transmitter uses the TX Buffer, TX Shift, and TX Control registers. When the Enable bit in the TX Control register is set, an internal divide-by-eight bit clock is generated. A data byte to transmit is written by an API routine into the TX Buffer register, clearing the Tx Buffer Empty status bit in the TX Control register. This status bit can be used to detect and prevent transmit overrun errors. The rising edge of the next bit clock transfers the data to the Shift register and sets the Tx Buffer Empty bit in the TX Control register. If the interrupt enable mask is enabled, an interrupt is triggered. This interrupt enables queuing the next byte to transmit. So when the current data byte is completely transmitted, the new byte is transmitted on the next available transmit clock. The start bit is transmitted at the same time that the data byte is transferred from the TX Buffer register to the TX Shift register. Successive bit clocks shift a serial bit stream to the output. The stream is composed of each bit of the data byte, least significant bit first, an optional parity bit, and a final stop bit. When the stop bit is completely transmitted, the TX Control register's Tx-Complete status bit is set. This bit remains valid until read. If a new data byte has been written to the TX Buffer register, the data byte is transferred to the TX Shift register and transmission of the data begins on the next rising edge of the bit clock.

RX - UART Receiver

The receiver uses the RX Buffer, RX Shift, and RX Control registers.
Initialization of the RX consists of setting the UART parity, optionally enabling the interrupt on the Rx Register Full condition, and then enabling the UART. When a start bit is detected on the RX input, a divide-by-eight bit clock is started and synchronized to sample the data in the center of the received bits. On the rising edge of the next eight-bit clock, the input data is sampled and shifted into the RX Shift register. If parity is enabled, the next bit clock samples the parity bit. The sampling of the stop bit, on the next clock, results in the received data byte transfer to the RX Buffer register and the triggering of one or more of the following events:

- Rx Register Full bit in the RX Control register is set, and if the interrupt for the RX is enabled, then the associated interrupt is triggered.
- If the stop bit is not detected at the expected bit position in the data stream, then the Framing Error bit in the RX Control register is set.
- If the Buffer register has not been read before the stop bit of the currently received data, then the Over-run Error bit in the RX Control register is set.
- If a parity error was detected, then the Parity Error bit is set in the RX Control register.
- For polling detection of a completely received data byte, the Rx Register Full bit in the RX Control register must be monitored. Data must be read out of the RX Buffer register before the next byte is completely received, to prevent the overrun error condition.