July 2008
DE REF #318

iCE65 as UART Transciever

Overview

UART Transceiver provides a cost-effective solution for asynchronous serial data transmission in communication applications. This design example illustrates the implementation of UART Transciever using iCE65 FPGAs.

Description

The UART, or Universal Asynchronous Receiver / Transmitter, is a device useful for communicating serial data (text, numbers, etc.) to an Application Processor. The device converts incoming parallel information to serial data which can be sent on a communication line.

The asynchronous transmission in UARTs involves "start" bit, five to eight data bits, least-significant-bit first, an optional "parity" bit, and then one, one and a half, or two "stop" bits. The start bit is the opposite polarity of the data-line's idle state. The stop bit is the data-line's idle state, and provides a delay before the next character can start. (This is called asynchronous start-stop transmission). The parity bit can either make the number of "one" bits between any start/stop pair odd, or even, or it can be omitted. Odd parity is more reliable because it assures that there will always be at least one data transition, and this permits many UARTs to resynchronize. An asynchronous transmission sends no characters over the interconnection when the transmitting device has nothing to send, only idle stop bits are sent then.

This UART design example combines UART Transmitter module, Receiver module, a baud-rate generator, Rx start synchronization logic, Rx and Tx FIFOs.

Figure1 shows a typical data frame of UART Transceiver.

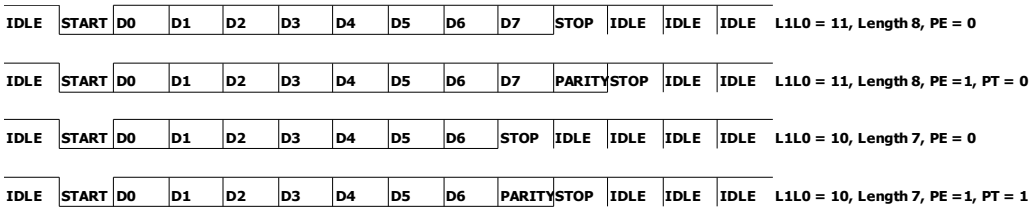

Fig 2: Typical UART Data frame

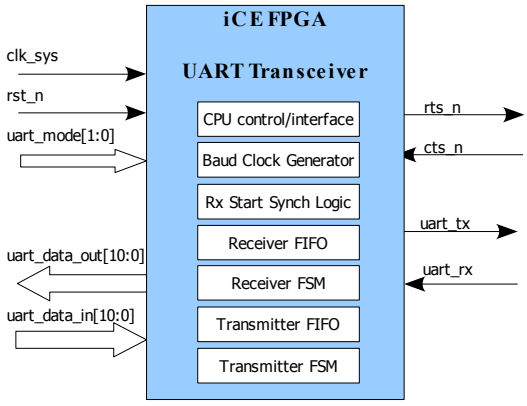

Fig 2: UART Transceiver

Implementation

Simplified block diagram of UART Transceiver is as shown in Figure 2 above. Figure 3 below lists the significance each bit in the UART data frame. The following summarizes the operation of the UART Transceiver:

- uart_mode signal selects 4 different UART operating modes as described below:
  when "00", reads data from Rx FIFO.
  when "01", reads status from Status Register.
  when "10", writes data to Tx FIFO.
  When "11", writes configuration register.
- Configure the UART by writing a 11-bit word to a write-configuration register, which contains the baud rate, data-word length, parity enable, and parity type.
- Transmit data by writing a 11-bit word to a write-data register and holding cts_n=0.
- The baud-rate generator determines the rate at which the transmitter and receiver operate. Bits B0 to B3 in the write-configuration register determine the baud-rate divisor.
- The receiver section receives data in serial form. The Rx FSM detects a start bit(a high-to-low Rx transition). An internal clock samples data at 16 times the data rate. Start is confirmed at the 8th clock pulse of 16x baud clock. Received data is stored in an 10-word FIFO, which includes parity and block error.

Table 1 below lists interface details of this Design Example.

Table 2 below shows the post P&R resource utilization summary of this design when implemented on an iCE65 FPGA.

UART Data Write – Mode 11:Config Register Write

DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
X	X	X	X	PT	PE	L1	L0	B2	B1	B0

UART Data Write – Mode 10: Tx FIFO Write

DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
X	X	X	D7	D6	D5	D4	D3	D2	D1	D0

UART Data Read – Mode 00: Read Rx FIFO

DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
X	PER	BER	D7	D6	D5	D4	D3	D2	D1	D0

UART Data Read – Mode 01: Status Register Read

DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
CTS	RTS	RXBSY	TXBSY	PER	BER	WFULL	RFULL	PE	L1	L0

Fig 2: UART Data Frame

Pin	Direction	Description
clk_sys	Input	System Clock
rst_n	Input	Active low system reset
uart_mode[1:0]	Input	Mode selection signal
uart_data_in[10:0]	input	UART Data input
uart_data_out[10:0]	output	UART Data output
rts_n	output	Active low, nRTS
cts_n	input	Active low, nCTS.
uart_tx	output	UART Transmit data
uart_rx	input	UART Receive data

Table1: Pin Description

Device	Logic Cells	IO Cells
iCE65L04-UCB284	371	27

Table2: Resource Utilization

Conclusion

This design example demonstrates the implementation of a UART Transciever interface using iCE FPGAs. iCE FPGA's very low power capabilities makes iCE FPGAs an obvious choice for implementing a UART Transeciver for battery operated compact and handheld devices like PDAs, cellular phones etc..

Disclaimer