Simple UART

Sébastien Bourdeauducq

August 2010

1 Specifications

The UART is based on a very simple design from Das Labor. Its purpose is basically to provide a debug console.

The UART operates with 8 bits per character, no parity, and 1 stop bit. The default baudrate is configured during synthesis and can be modified at runtime using the divisor register.

The divisor is computed as follows :

$$\text{divisor} = \frac{\text{Clock frequency (Hz)}}{16 \cdot \text{Bitrate (bps)}}$$

2 Registers

Offset	Read/Write	Default	Description
0x0	RW	0x00	Data register. Received bytes and bytes to transmit are read/written from/to this register.
0x4	RW	for default bitrate	Divisor register (for bitrate selection).

3 Interrupts

The core has two active-high edge-sensitive interrupts outputs.

The "RX" interrupt is sent whenever a new character is received. The CPU should then read the data register immediately. If a new character is sent before the CPU has had time to read it, the first character will be lost.

The "TX" interrupt is sent as soon as the UART finished transmitting a character. When the CPU has written to the data register, it must wait for the interrupt before writing again.

4 Using the core

Connect the CSR signals and the interrupts to the system bus and the interrupt controller. The `uart_tx` and `uart_rx` signals should go to the FPGA pads. You must also provide the desired default baudrate and the system clock frequency in Hz using the parameters.

Copyright notice