

# MiniUART IP Core Specification

*Author: Philippe Carton  
pc@microsystemes.com*

**Rev. 0.1  
January 17, 2002**

This page has been intentionally left blank

*Revision History*

Rev.	Date	Author	Description
0.1	28/12/01	Philippe Carton	First Draft

# Contents

Introduction .....	1
IO ports .....	2
Clocks .....	3
Registers .....	4
Operation .....	6
Architecture .....	8

# 1

---

## Introduction

This UART (Universal Asynchronous Receiver Transmitter) is designed to make an interface between a RS232 line and a wishbone bus, or a microcontroller, or an IP core. It works fine connected to the serial port of a PC for data exchange with custom electronic.

It was built in the perspective to be very small, but efficient. It had to fit in a small FPGA. It is not suited to interface a modem as there is no control handshaking (CTS/RTS). It integrate two separate clocks, one for wishbone bus, the other for bitstream generation. This has the advantage to let the user bring his own desired frequency for the baudrate.

**Support:**

- WISHBONE interface in 8-bit data bus
- Two clock: one for wishbone interface, one for RS232 bitstream generation
- Baudrate divisor from 1 to 65536 (generic parameter set at integration time)

**Does not support:**

- FIFO input/output
- Control handshaking

# 2

## IO ports

### 2.1 WISHBONE interface signals

Port	Width	Direction	Description
WB_CLK_I	1	Input	Block's clock input
WB_RST_I	1	Input	Asynchronous Reset
WB_ADDR_I	2	Input	Used for register selection
WB_DAT_I	8	Input	Data input
WB_DAT_O	8	Output	Data output
WB_WE_I	1	Input	Write or read cycle selection
WB_STB_I	1	Input	Specifies transfer cycle
WB_ACK_O	1	Output	Acknowledge of a transfer

### 2.2 Other internal signals

Port	Width	Direction	Description
IntTx_O	1	Output	Transmit Interrupt
IntRx_O	1	Output	Receive Interrupt
BR_CLK_I	1	Output	Clock for serialisation/unserialisation

### 2.3 External (off-chip) connections

Port	Width	Direction	Description
TxD_PAD_O	1	Output	The serial output signal
RxD_PAD_I	1	Input	The serial input signal

# 3

---

## Clocks

Clocks table:

Name	Source	Rates (MHz)			Remarks	Description
		Max	Min	Resolution		
WB_CLK_I	Wishbone bus	Limited by target host			None	Wishbone clock
BR_CLK_I	User				None	Baudrate clock

# 4

## Registers

### 4.1 Registers list

Name	Address	Width	Access	Description
Receive buffer	0	8	R	Contain byte received
Transmit buffer	0	8	W	Contain byte to transmit
Status	1	8	R	Receive buffer full / Transmitter busy
Reserved	2	8		
Reserved	3	8		



## 4.2 Status register

Bit #	Access	Description
0	R	Transmitter buffer state = IntTx_O pin '0' – Busy. Can't accept incoming byte '1' – Accept a byte to transmit
1	R	Receiver buffer state = IntRx_O pin '0' – Buffer empty '1' – Buffer contain a received byte

Reset Value: XXXXXX01b

# 5

---

## Operation

The UART Operation is very basic :

Upon a write to the data input bus WB\_DAT\_I, the core will automatically serialise and emit the byte on the TxD\_PAD\_O. It will hold IntTx\_O low as long as it cannot accept an incoming byte. Therefore a rising edge on IntTx\_O can trigger the interrupt line of a microcontroller to emit another byte.

Upon reception of a bitstream on RxD\_PAD\_O, the core will unserialise the information and assert IntRx\_O pin. This announce that the received byte can be read on the data output bus WB\_DAT\_O. As soon as the byte is read, IntRx\_O is negated.

### Wishbone bus

The core is 8 bit wishbone compatible. It doesn't use the WB\_CYC\_I pin as it will never insert wait states.

### Initialization

The core doesn't need to be reset, as it is ready to use upon power on.

However, a synchronous assertion of WB\_RST\_I will abort any pending transmit / receive and will set the core in idle state.

### Baudrate divisor

A generic VHDL parameter allow the user to introduce a divisor between BR\_CLK\_I clock and the bitstream frequency. In addition, the core insert a 4 divisor for sampling purpose for the receiver.

Thus:

$$\text{Baudrate} = \text{Freq}(\text{BR\_CLK\_I}) / \text{BRDIVISOR} / 4$$

Where BRDIVISOR is the generic parameter.

To instanciate the component in a VHDL unit, write:

```
U1 : MiniUART
  generic map (BRDIVISOR => 103)
  port map (clk, rst, adr, ...);
```

for a division of 103.

Below are given some divisor values for common baudrates:

BRDIVISOR	Baudrate
2080	1200
1040	2400
520	4800
260	9600
130	19200
65	38400

BR\_CLK\_I at 10MHz

BRDIVISOR	Baudrate
192	2400
96	4800
48	9600
32	14400
24	19200
16	28800
8	57600
4	115200
2	230400
1	460800

BR\_CLK\_I at 1.8432MHz

## Baudrate tolerance

The receiver will accept a light variation between the expected baudrate and the effective bitstream baudrate that is:

Min freq	Nominal Baudrate -5.58%
Max freq	Nominal Baudrate +2.60%

# 6

## Architecture

The block diagram of the core is given:

