IOB-UART, a RISC-V UART

User Guide, V0.1 , Build c5bd435

March 17, 2021

`www.iobundle.com`
Confidential

Contents

List of Tables

List of Figures

`www.iobundle.com` **Confidential**

1   Introduction

The IObundle UART is a RISC-V-based Peripheral written in Verilog, which users can download for free, modify, simulate and implement in FPGA or ASIC. It is written in Verilog and includes a C software driver. The IObundle UART is a very compact IP, that works at high clock rates if needed. It supports full-duplex operation and a configurable baud rate. The IObundle UART has a fixed configuration for the start and stop bits. More flexible licensable commercial versions are available upon request.
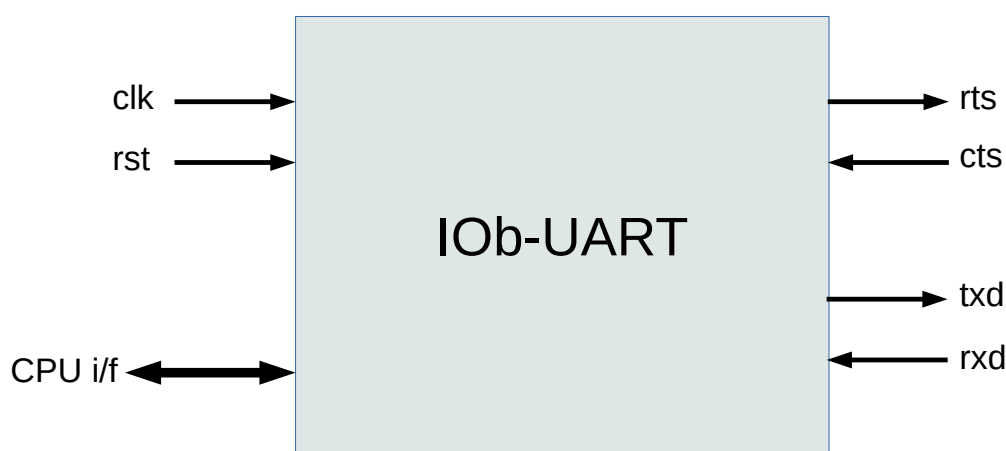
2   Symbol

Figure 1: IP Core Symbol

3   Features

- Supported in IObundle's RISC-V IOb-SoC open-source and free of charge template.
- IObundle's IOb-SoC native CPU interface.
- Verilog basic UART implementation.
- Reset, enable and time read functions.
- Runtime configurable baud rate
- C software driver at the bare-metal level.
- Simple Verilog testbench for the IP's *nucleus*.
- System-level Verilog testbench available when simulating the IP embedded in IOb-SoC.
- Simulation Makefile for the open-source and free of charge Icarus Verilog simulator.
- FPGA synthesis and implementation scripts for two FPGA families from two FPGA vendors.
- Automated creation of FPGA netlists

- Automated production of documentation using the open-source and free Latex framework.

- IP data automatically extracted from FPGA tool logs to include in documents.

- Makefile tree for full automation of simulation, FPGA implementation and document production.

- AXI4 Lite CPU interface (premium option).

- Parity bits (premium option).

4 Benefits

- Easy hardware and software integration

- Compact hardware implementation

- Can fit many instances in low cost FPGAs

- Can fit many instances in small ASICs

- Low power consumption

5 Deliverables

- ASIC or FPGA synthesized netlist or Verilog source code

- ASIC or FPGA synthesis and implementation scripts or

- ASIC or FPGA verification environment

- Software driver and example user software

- User documentation for easy system integration

- Example integration in IOb-SoC (optional)

6 Block Diagram and Description

A high-level block diagram of the IOB-UART core is presented in Figure 6 and a brief explanation of each block is given in Table 1.
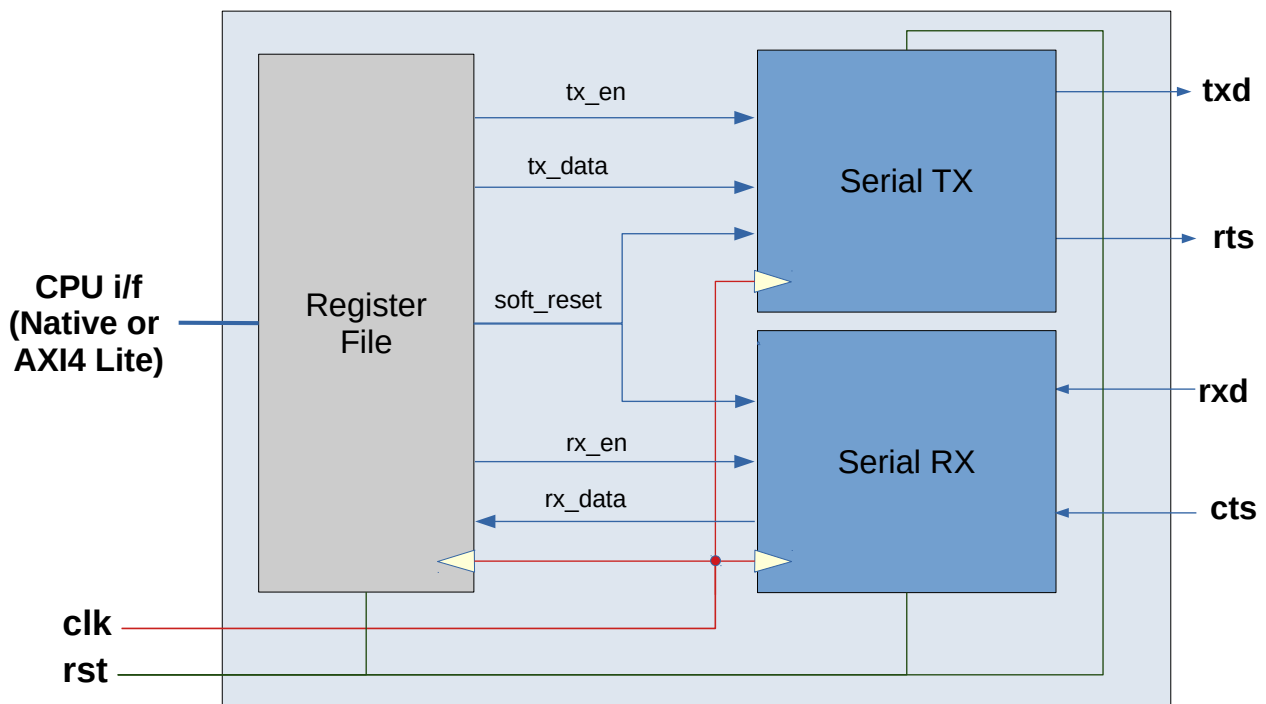
www.iobundle.com   **Confidential**

Figure 2: High-level block diagram

Block	Description
Register File	Holds the current configuration of the UART as well as internal parameters. Data to be sent or that has been received is stored here temporarily.
Serial TX	After enabled, this block serializes the data previously written to the `tx_data` register by the CPU, and sends the data word over the single transmit line connected to output `txd`.
Serial RX	After enabled, this block deserializes the data in the incoming single transmit line connected to pin `txd`, and writes a data word to the `rx_data` register for the CPU to read.

Table 1: Block descriptions.

7 Interface Signals

The interface signals of the UART core are described in the following tables.

Name	Direction	Width	Description
clk	input	1	System clock input
rst	input	1	System reset asynchronous and active high

Table 2: General Interface Signals

Name	Direction	Width	Description
valid	input	1	Native CPU interface valid signal
address	input	ADDR_W	Native CPU interface address signal
wdata	input	WDATA_W	Native CPU interface data write signal
wstrb	input	DATA_W/8	Native CPU interface write strobe signal
rdata	output	DATA_W	Native CPU interface read data signal
ready	output	1	Native CPU interface ready signal

Table 3: CPU Native Slave Interface Signals

Name	Direction	Width	Description
s_axil_awaddr	input	ADDR_W	Address write channel address
s_axil_awcache	input	4	Address write channel memory type. Transactions set with Normal Non-cacheable Modifiable and Bufferable (0011).
s_axil_awprot	input	3	Address write channel protection type. Transactions set with Normal Secure and Data attributes (000).
s_axil_awvalid	input	1	Address write channel valid
s_axil_awready	output	1	Address write channel ready
s_axil_wdata	input	DATA_W	Write channel data
s_axil_wstrb	input	DATA_W/8	Write channel write strobe
s_axil_wvalid	input	1	Write channel valid
s_axil_wready	output	1	Write channel ready
s_axil_bresp	output	2	Write response channel response
s_axil_bvalid	output	1	Write response channel valid
s_axil_bready	input	1	Write response channel ready
s_axil_araddr	input	ADDR_W	Address read channel address
s_axil_arcache	input	4	Address read channel memory type. Transactions set with Normal Non-cacheable Modifiable and Bufferable (0011).
s_axil_arprot	input	3	Address read channel protection type. Transactions set with Normal Secure and Data attributes (000).
s_axil_arvalid	input	1	Address read channel valid
s_axil_arready	output	1	Address read channel ready
s_axil_rdata	output	DATA_W	Read channel data
s_axil_rresp	output	2	Read channel response
s_axil_rvalid	output	1	Read channel valid
s_axil_rready	input	1	Read channel ready

Table 4: CPU AXI4 Lite Slave Interface Signals

Name	Direction	Width	Description
interrupt	output	1	to be done
txd	output	1	Serial transmit line
rxd	input	1	Serial receive line
cts	input	1	Clear to send the destination is ready to receive a transmission sent by the UART
rts	output	1	Ready to send the UART is ready to receive a transmission from the sender.

Table 5: RS232 Interface Signals

     www.iobundle.com   **Confidential**

8 Registers

The software accessible registers of the UART core are described in Table 6. The table gives information on the name, read/write capability, word aligned addresses, used word bits, and a textual description.

Name	R/W	Addr	Bits	Initial Value	Description
UART_SOFTRESET	W	0x00	0:0	0	Bit duration in system clock cycles.
UART_DIV	W	0x04	DATA_W/2-1:0	0	Bit duration in system clock cycles.
UART_TXDATA	W	0x08	DATA_W/4-1:0	0	TX data
UART_TXEN	W	0x0c	0:0	0	TX enable.
UART_TXREADY	R	0x10	0:0	0	TX ready to receive data
UART_RXDATA	R	0x14	DATA_W/4-1:0	0	RX data
UART_RXEN	W	0x18	0:0	0	RX enable.
UART_RXREADY	R	0x1c	0:0	0	RX data is ready to be read.

Table 6: Software accessible registers.

9 FPGA Results

The following are FPGA implementation results for two FPGA device families.

Resource	Used
LUTs	106
Registers	104
DSPs	0
BRAM	0

Resource	Used
ALM	85
FF	113
DSP	0
BRAM blocks	0
BRAM bits	

Table 7: FPGA results for Kintex Ultrascale (left) and Cyclone V GT (right)