

IOB-UART, a RISC-V UART

User Guide, V0.1 , Build 74f7ee8



February 26, 2022



Contents

1	Introduction	5
2	Symbol	5
3	Features	5
4	Benefits	6
5	Deliverables	6
6	Block Diagram and Description	7
7	Synthesis Parameters and Macros	7
8	Interface Signals	8
9	Implementation Results	9

List of Tables

1	Synthesis Parameters.	7
2	Synthesis Macros.	7
3	General interface signals.	8
4	IObundle Interface Signals	8
5	RS232 Interface Signals	8
6	FPGA results for Kintex Ultrascale (left) and Cyclone V GT (right).	9

List of Figures

1	IP core symbol.	5
2	High-level block diagram.	7



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

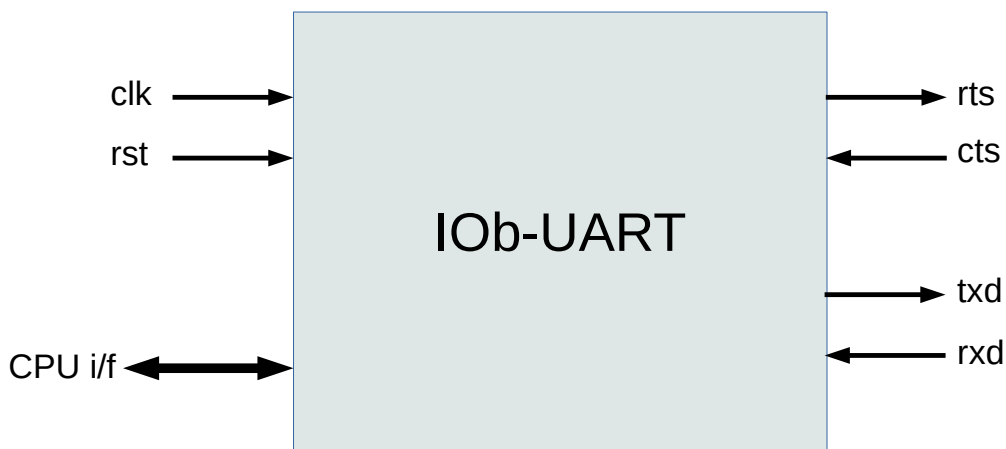


Figure 1: IP core symbol.

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.
- Soft reset and enable 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

- Compact and easy to integrate hardware and software implementation
- Can fit many instances in low cost FPGAs and ASICs
- Low power consumption

5 Deliverables

- ASIC or FPGA synthesized netlist or Verilog source code, and respective synthesis and implementation scripts
- ASIC or FPGA verification environment by simulation and emulation
- Bare-metal 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 core is presented in Figure 2, followed by a brief description of each of the blocks

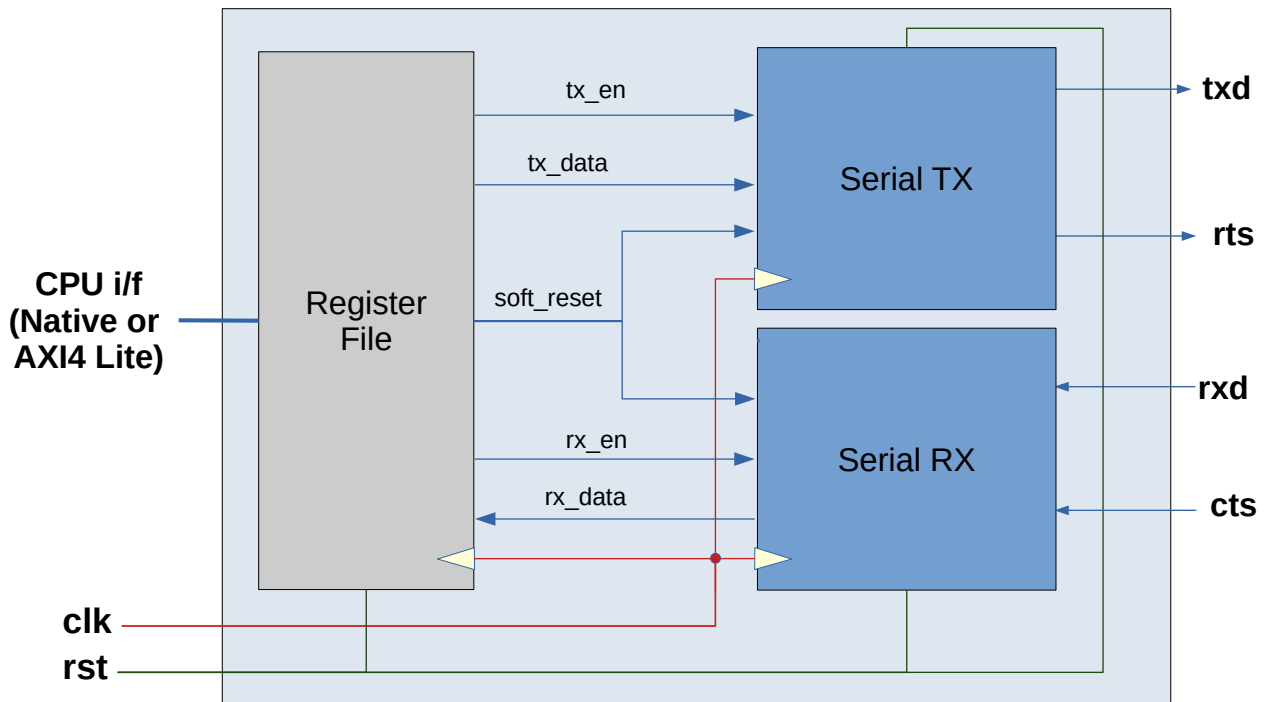


Figure 2: High-level block diagram.

REGISTER FILE Configuration control and status register file.

7 Synthesis Parameters and Macros

The generic synthesis parameters of the core are presented in Table 1. Generic parameters can vary from instance to instance.

Parameter	Default Value	Description
DATA_W	32	CPU data width

Table 1: Synthesis Parameters.

The synthesis macros of the core are presented in Table 2. Macros apply to all instances of the core.

Parameter	Default Value	Description
ADDR_W	3	CPU address width

Table 2: Synthesis Macros.

8 Interface Signals

The interface signals of the 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 3: 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	DATA_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 4: IObundle Interface Signals

Name	Direction	Width	Description
interrupt	OUTPUT	1	to be done
txd	OUTPUT	1	Serial transmit line
rxn	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

9 Implementation Results

The following are FPGA implementation results for two FPGA families.

Resource	Used
LUTs	100
Registers	112
DSPs	0
BRAM	0

Resource	Used
ALM	87
FF	121
DSP	0
BRAM blocks	0
BRAM bits	0

Table 6: FPGA results for Kintex Ultrascale (left) and Cyclone V GT (right).