IOB-ILA, a RISC-V ILA

User Guide, V0.1 , Build c06b7d8

June 11, 2021

Contents

1 Introduction **5**

2 Symbol **5**

3 Features **5**

4 Benefits **6**

5 Deliverables **6**

6 Block Diagram and Description **6**

7 Interface Signals **7**

8 Registers **9**

9 FPGA Results **9**

List of Tables

1 Block descriptions. 7

2 General Interface Signals 7

3 CPU Native Slave Interface Signals 8

4 CPU AXI4 Lite Slave Interface Signals 8

5 RS232 Interface Signals 8

6 Software accessible registers. 9

7 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 ILA 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 ILA 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 ILA has a fixed configuration for the Start and Stop bits. More flexible licensable commercial versions are available upon request.

2 Symbol

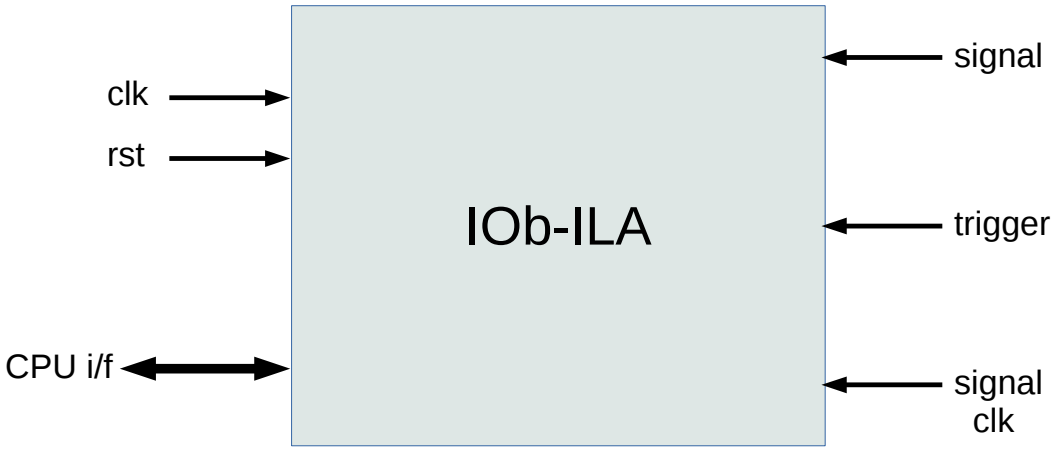

Figure 1: IP Core Symbol

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 ILA implementation.
- Soft reset and enable functions.
- Runtime configurable
- 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.

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-ILA core is presented in Figure 6 and a brief explanation of each block is given in Table 1.

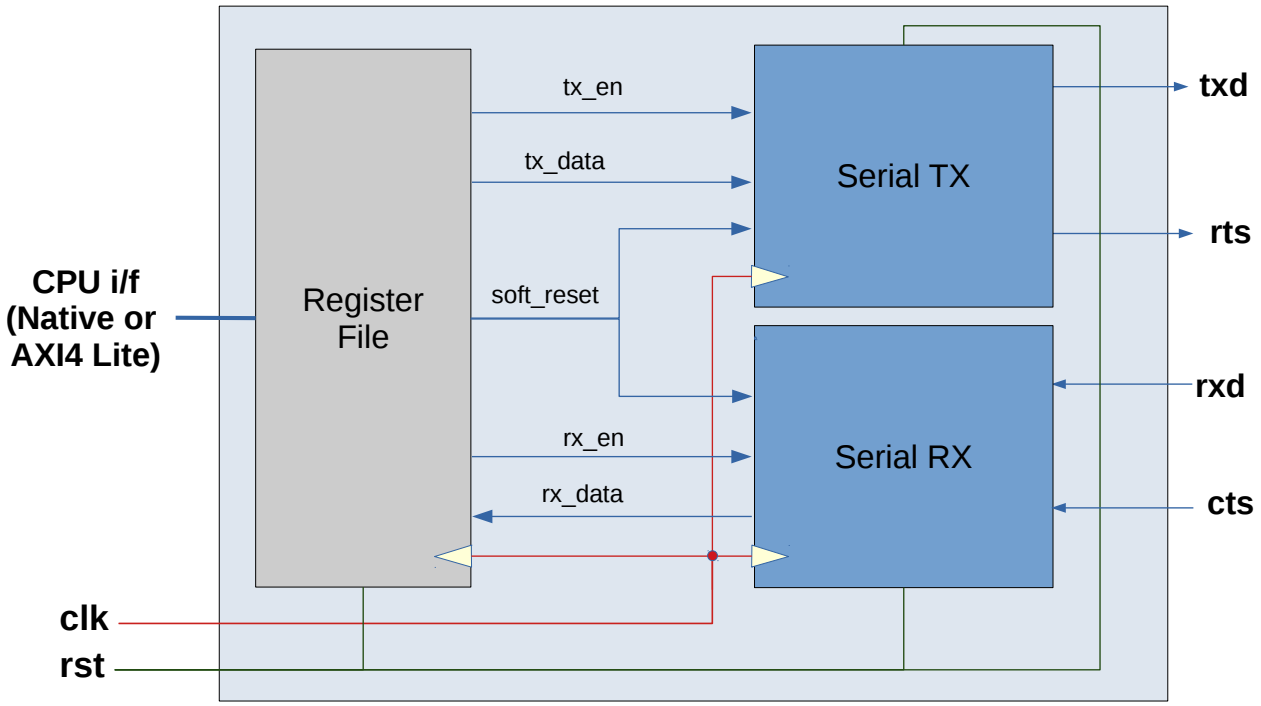

Figure 2: High-level block diagram

Block	Description
Register File	Holds the current configuration of the ILA as well as internal parameters. Data to be sent or that has been received is stored here temporarily.

Table 1: Block descriptions.

7 Interface Signals

The interface signals of the ILA 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	'AXI_CACHE_W	Address write channel memory type. Transactions set with Normal Non-cacheable Modifiable and Bufferable (0011).
s_axil_awprot	input	'AXI_PROT_W	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	'AXI_RESP_W	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	'AXI_CACHE_W	Address read channel memory type. Transactions set with Normal Non-cacheable Modifiable and Bufferable (0011).
s_axil_arprot	input	'AXI_PROT_W	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	'AXI_RESP_W	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
signal	input	DATA_W	
trigger	input	1	
sampling_clk	input	1	

Table 5: RS232 Interface Signals

8 Registers

The software accessible registers of the ILA 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
ILA_SOFTRESET	W	0x00	0:0	0	
ILA_TRIGGER_TYPE	W	0x04	0:0	0	
ILA_ENABLED	W	0x08	0:0	1	Starts enabled
ILA_INDEX	W	0x0c	'ILA_MAX_SAMPLES_W-1:0	0	Since it is a debug core samples are accessed by first setting the index and then reading the value of ILA_DATA
ILA_SAMPLES	R	0x10	'ILA_MAX_SAMPLES_W-1:0	0	
ILA_DATA	R	0x14	'ILA_RDATA_W-1:0	0	

Table 6: Software accessible registers.

9 FPGA Results

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

Resource	Used
LUTs	43
Registers	27
DSPs	0
BRAM	4

Resource	Used
ALM	28
FF	27
DSP	0
BRAM blocks	16
BRAM bits	131,072

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