IOB-CACHE

User Guide, 0.1 , Build 9f239e9

March 4, 2022

Contents

List of Tables

List of Figures

1 Introduction

The IObundle CACHE is ....

2 Symbol

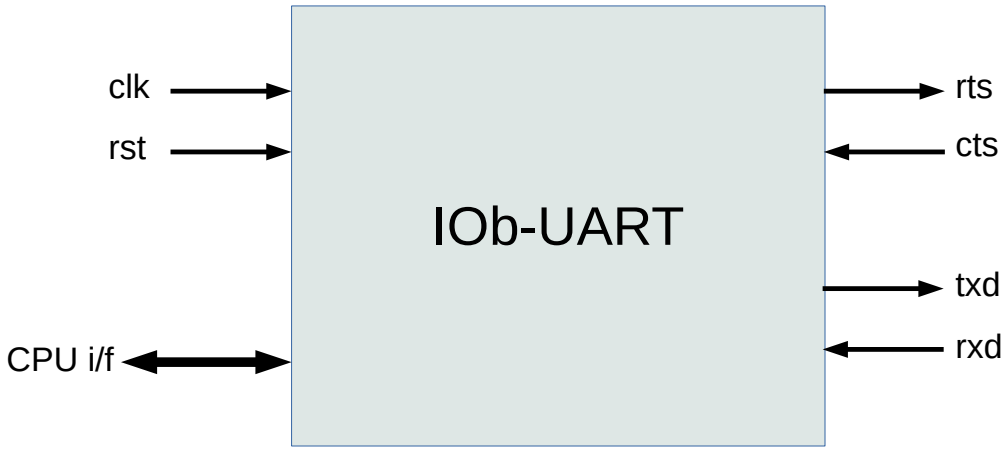

Figure 1: IP core symbol.

3 Features

- feature1
- feature2

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.

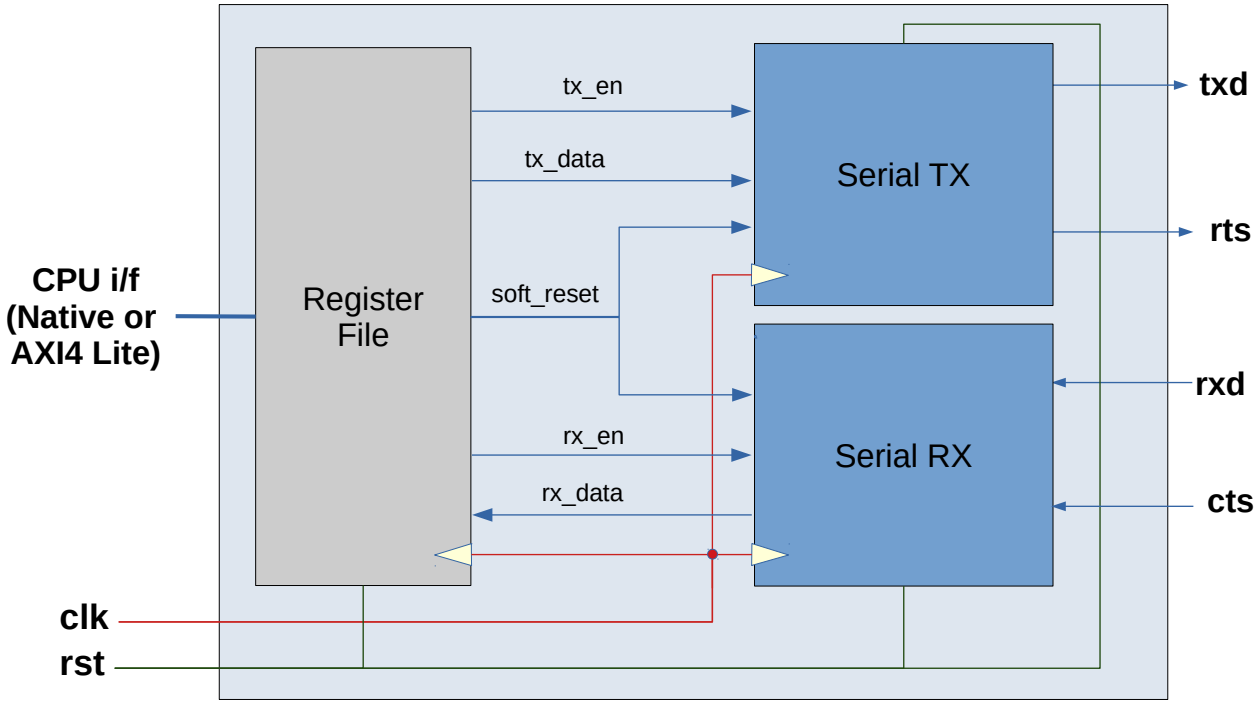

Figure 2: High-level block diagram.

FRONT-END Front-end block.

CACHE MEMORY Cache memory block.

BACK-END Back-end block.

CACHE CONTROL Cache control block.

7 Interface Signals

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

Table 1: General interface signals.

Name	Direction	Width	Description
valid	INPUT	1	Native CPU interface valid signal
addr	INPUT	CTRL_CACHE + FE_ADDR_W - FE_BYTE_W	Native CPU interface address signal
addr	INPUT	CTRL_CACHE + FE_ADDR_W	Native CPU interface address signal
wdata	INPUT	FE_DATA_W	Native CPU interface data write signal
wstrb	INPUT	FE_NBYTES	Native CPU interface write strobe signal
rdata	OUTPUT	FE_DATA_W	Native CPU interface read data signal
ready	OUTPUT	1	Native CPU interface ready signal

Table 2: IObundle Master Interface Signals

Name	Direction	Width	Description
force_inv_in	INPUT	1	force 1'b0 if unused
wtb_empty_in	INPUT	1	force 1'b1 if unused

Table 3: RS232 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 Slave Interface Signals

8 Instantiation and External Circuitry

Figure 4 illustrates how to instantiate the IP core and, if apllicable, the required external blocks. A Verilog file describing this setup is provided.

bla bla bla...

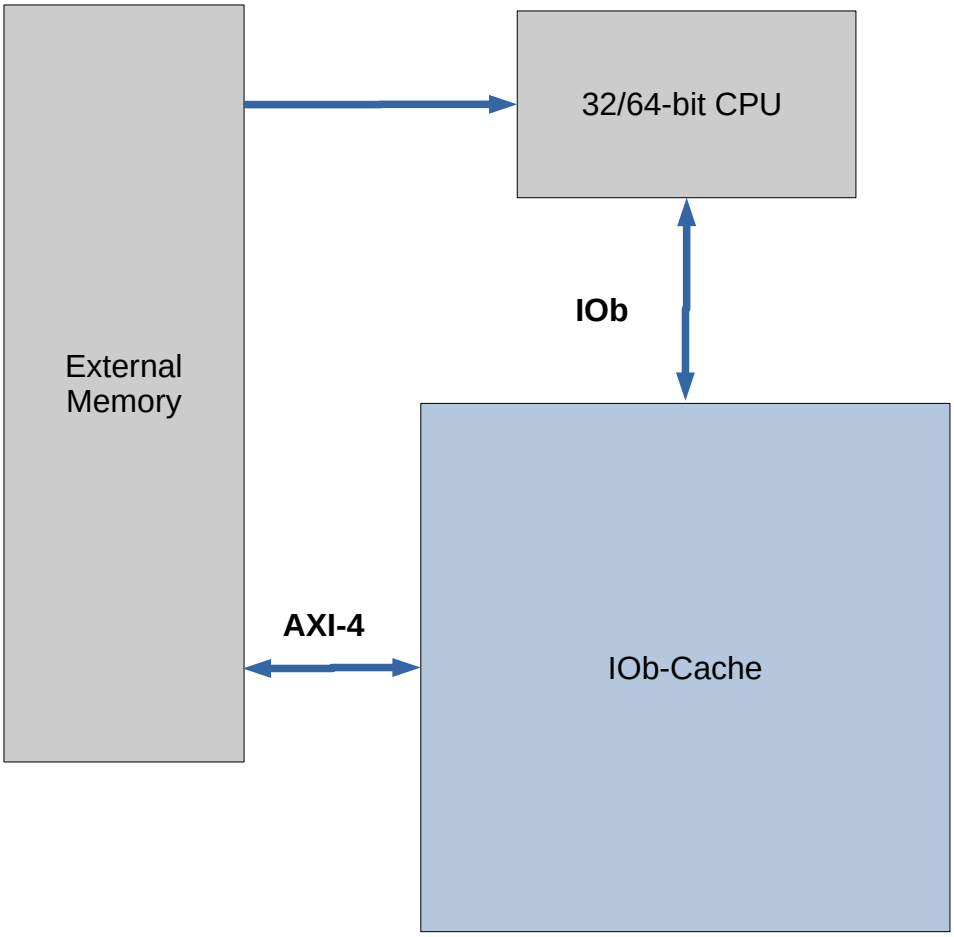

Figure 3: Core instance and required surrounding blocks

9 Simulation

The provided testbench uses the core instance described in Section 8. A high level block diagram of the testbench is shown in Figure 4. The testbench is organised in modular fashion with each test described in a separate file. The test suite consists of all the test case files, so that it becomes easy to add, modify or remove tests.

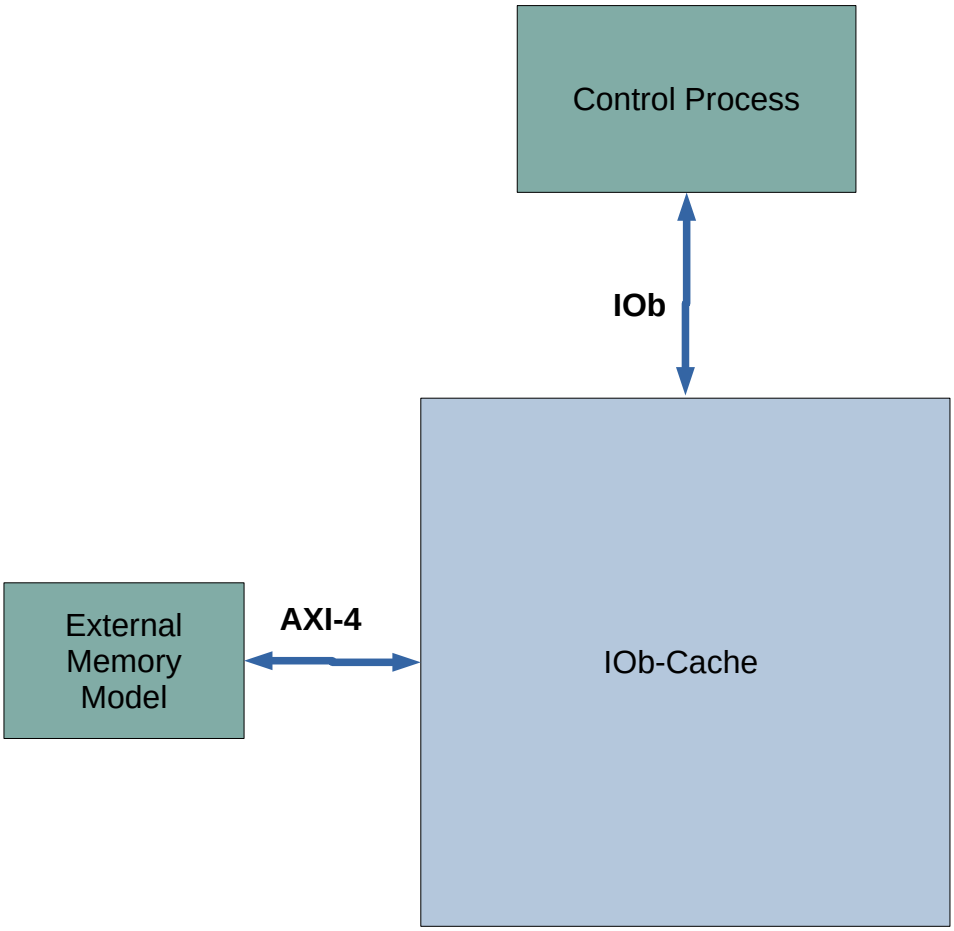

Figure 4: Testbench block diagram

In this preliminary version, simulation is not yet fully functional. The provided testbench merely allows compilation for simulation, and drives the clock and reset signals. Behavioural memory models to allow pre-synthesis simulation are already included. In the case of ROMs, their programming data is also included in the form of .hex files.

10 Synthesis

10.1 Synthesis Macros and Parameters

10.2 Synthesis Script and Timing Constraints

A simple `.tcl` script is provided for the Cadence Genus synthesis tool. The script reads the technology files, compiles and elaborates the design, and proceeds to synthesise it. The timing constraints are contained within the constraints file provided, or provided in a separate file.

After synthesis, reports on silicon area usage, power consumption, and timing closure are generated. A post-synthesis Verilog file is created, to be used in post-synthesis simulation.

In this preliminary version, synthesis of the IP core without the memories is functional. The memories are for now treated as black boxes.

It is important not to include the memory models provided in the simulation directory in synthesis, unless they are to be synthesised as logic. In a next version, the memories will be generated for the target technology and included.

11 Implementation Results

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

Resource	Used
LUTs	2168
Registers	1227
DSPs	0
BRAM	0

Resource	Used
ALM	826
FF	610
DSP	0
BRAM blocks	68
BRAM bits	72,768

Table 5: FPGA results for Kintex Ultrascale (left) and Cyclone V GT (right).