64-bit Timer

IOB-TIMER User Guide, V0.1 , Build 007125a

December 8, 2020

Contents

List of Tables

List of Figures

1 Introduction

The IObundle Timer core includes a 64-bit counter for returning the time in clock cycles. It is written in Verilog and includes a C software driver. With the knowledge of the clock frequency in its software driver, it is also possible to print the time in microseconds, milliseconds or seconds. The IP is currently supported for use in ASICs and FPGAs.

2 Symbol

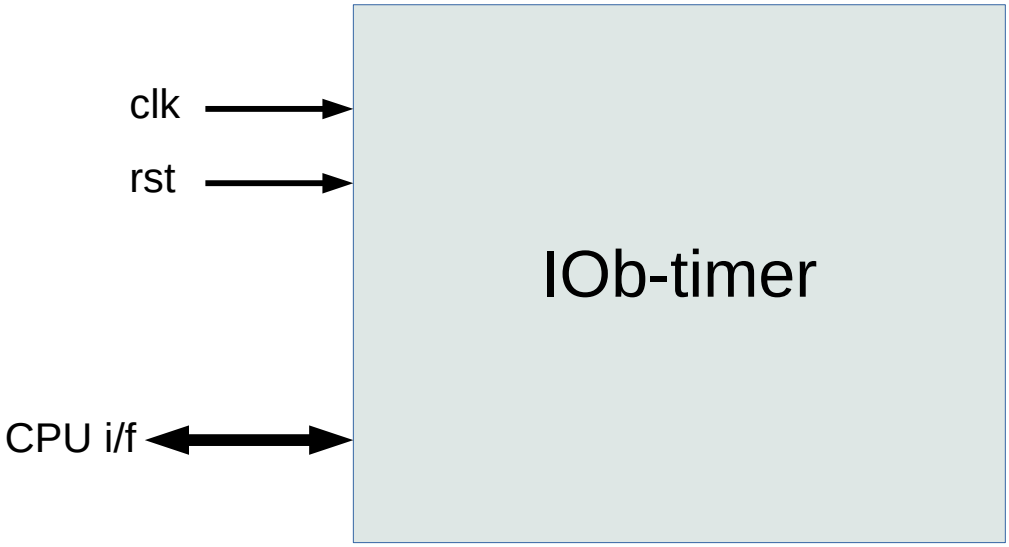

Figure 1: IP Core Symbol

3 Features

- Verilog 64-bit time counter in clock cycles.
- C software driver.
- Reset, enable and time read functions.
- IOb-SoC native CPU interface.
- AXI4 Lite CPU interface (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-TIMER core is presented in Figure 6 and a brief explanation of each block is given in Table 1.

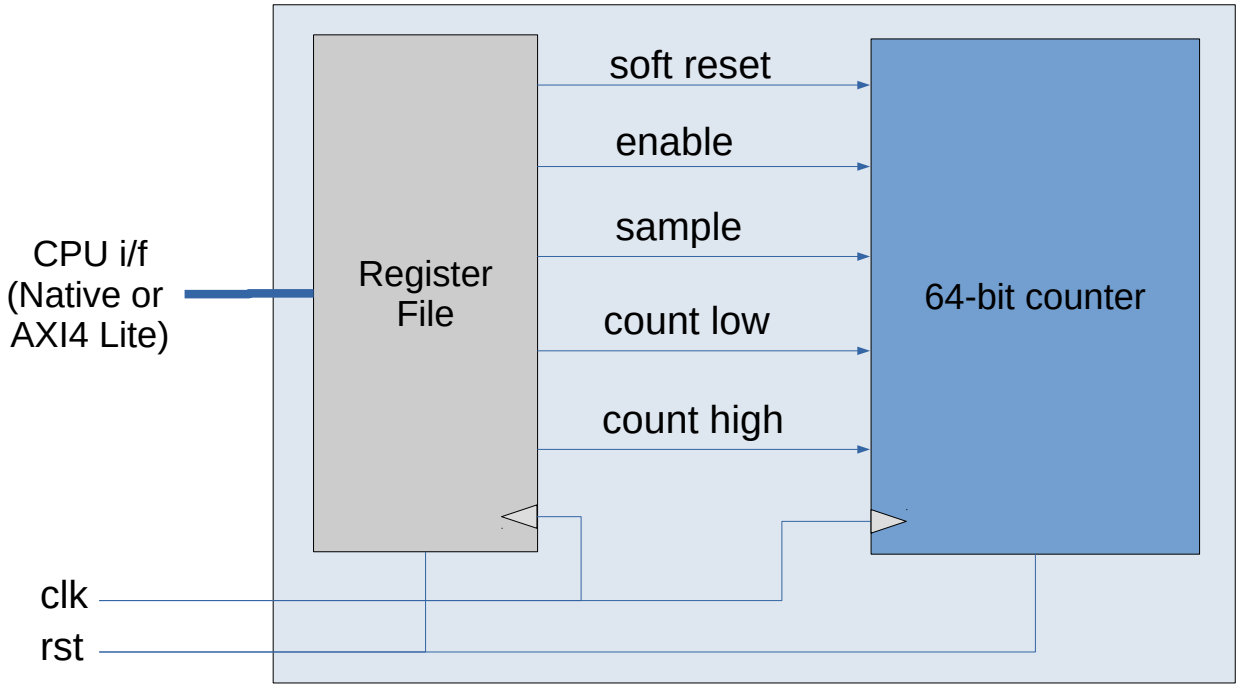

Figure 2: High-level block diagram

Block	Description
Register File	Configuration, control and status registers accessible by the sofware
64-bit time counter	Free-running 64-bit counter with enable and soft reset capabilities

Table 1: Block descriptions.

7 Interface Signals

The interface signals of the I^{2}S/TDM transceiver 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

8 Registers

The software accessible registers of the TIMER core are described in Table 5. 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
TIMER_RESET	W	0x00	0:0	0	Timer soft reset
TIMER_ENABLE	W	0x04	0:0	0	Timer enable
TIMER_SAMPLE	W	0x08	0:0	0	Sample time counter value into a readable register
TIMER_DATA_HIGH	R	0x0c	DATA_W-1:0	0	High part of the timer value which has twice the width of the data word width
TIMER_DATA_LOW	R	0x10	DATA_W-1:0	0	Low part of the timer value which has twice the width of the data word width

Table 5: Software accessible registers.

9 FPGA Results

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

Resource	Used
LUTs	37
Registers	132
DSPs	0
BRAM	0

Resource	Used
ALM	59
FF	165
DSP	0
BRAM blocks	0
BRAM bits	

Table 6: Kintex Ultrascale (left) and Cyclone V GT (right)