32-bit KNN

IOB-KNN User Guide, V0.1 , Build 026adde

January 3, 2021

Contents

List of Tables

List of Figures

1 Introduction

The IObundle KNN core implements a 32-bit version of the k-nearest neighbour algorithm. It is written in Verilog and includes a C software driver.The IP is currently supported for use in ASICs and FPGAs.

2 Symbol

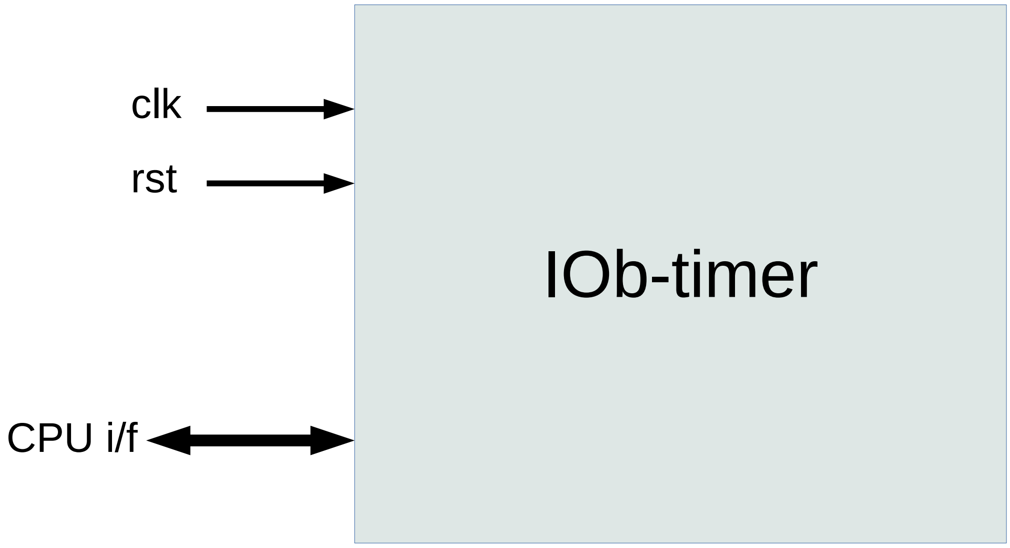

Figure 1: IP Core Symbol

3 Features

- Verilog 32-bit module to calculate the distances in the KNN algorithm.
- C software driver.
- Reset, enable and distance calculation functions.
- IOb-SoC native CPU interface.

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

- FPGA synthesized netlist or
- ASIC synthesized netlist or
- Verilog source code
- Example testbench
- User documentation for easy system integration
- Example integration in IOb-SoC (optional)
- FPGA synthesis and implementation scripts or
- ASIC synthesis and place and route scripts

6 Block Diagram and Description

A high-level block diagram of the IOB-KNN core is presented in Figure **??** and a brief explanation of each block is given in Table **??**.

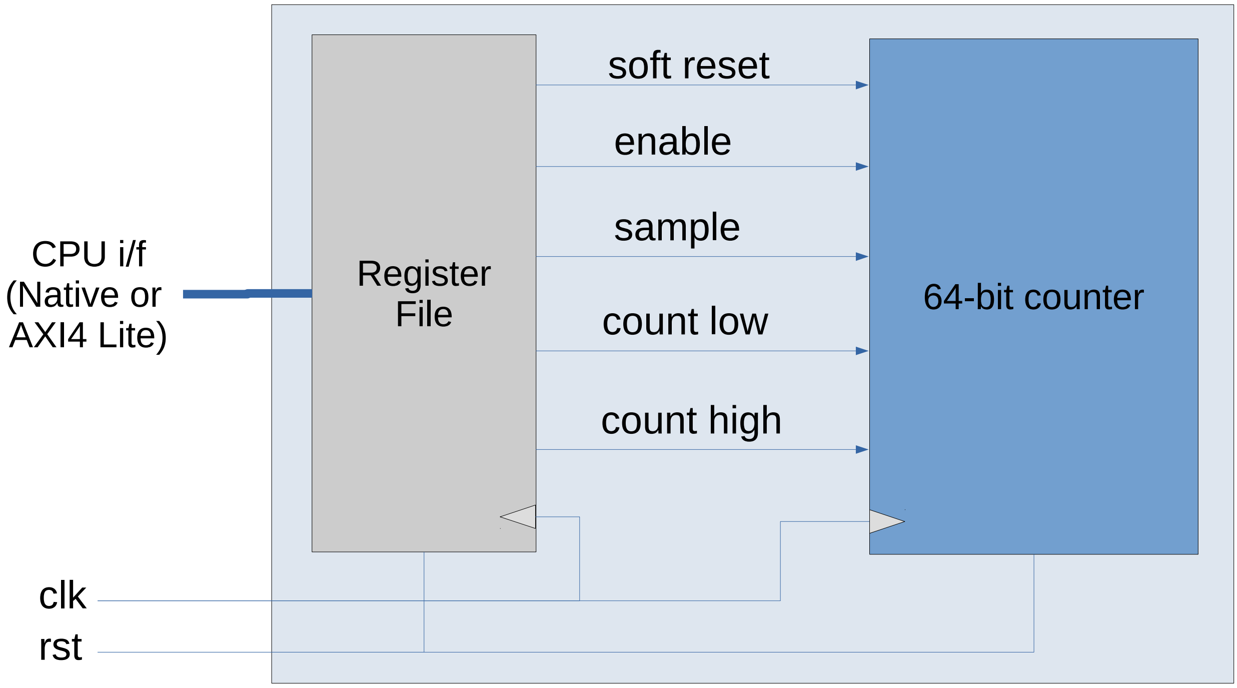

Figure 2: High-level block diagram

Block	Description
Register File	Configuration, control and status registers accessible by the sofware

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 KNN core are described in Table **??**. 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
KNN_RESET	W	0x00	0:0	0	KNN soft reset
KNN_ENABLE	W	0x04	0:0	0	KNN enable o
KNN_x1sw	W	0x08	‘DATA_W/2-1:0	0	
KNN_x2sw	W	0x0c	‘DATA_W/2-1:0	0	
KNN_y1sw	W	0x10	‘DATA_W/2-1:0	0	
KNN_y2sw	W	0x14	‘DATA_W/2-1:0	0	
KNN_zsw	R	0x18	‘DATA_W-1:0	0	
dummy	R	0x1c	‘DATA_W-1:0	0	

Table 5: Software accessible registers.

9 FPGA Results

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

Resource	Used
LUTs	68
Registers	65 Registers
65	
DSPs	2
BRAM	0

Table 6: Implementation Resources for Xilinx Kintex Ultrascale Devices