AXI Event Counter V1.3

IP User Guide

1 Description

The AXI Event Counter is an FPGA IP Core for debugging and performance measurement purposes. It contains a simple clock cycle counter that can be triggered by external start / stop inputs. Access to the counter register is granted through an AXI Lite interface.

2 Applications

- Performance Measurements
- Latency Measurement

3 Features

- AXI Lite register interface
- Asynchronous start / stop trigger inputs
- Software start / stop

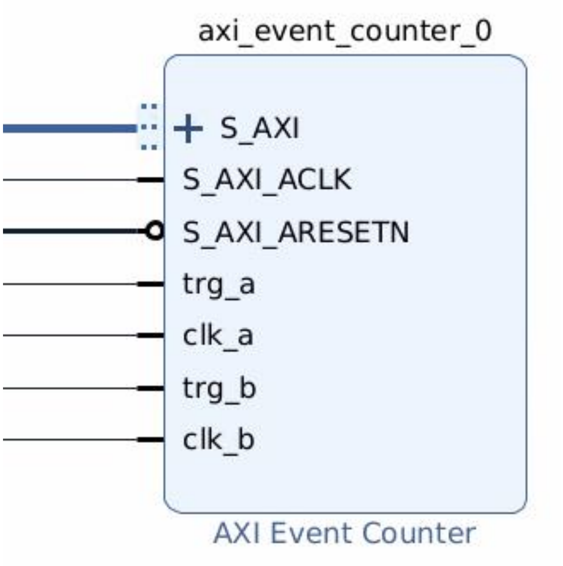

Figure 1: AXI Event Counter IP

Table of Contents

5 Designing with the Core

Clocking

The AXI Event Counter will measure the latency between a trg_a and a trg_b event in clock cycles of the axi_aclk clock domain. A finer time resolution is achieved with higher clock frequencies. The maximum recommended clock frequency is 300 MHz.

Asynchronous Trigger Inputs

Both trg_a and trg_b are asynchronous inputs which are internally connected to a pulse elongation and clock domain crossing synchronization to the axi_aclk clock. Pulses up to double the axi_aclk frequency are detectable. In the best case, both the start and stop pulse are delayed exactly the same number of clock cycles and the measured latency is accurate. In the worst case, the two pulses are delayed unequally which can lead to a maximum error of 2 clock cycles.

Table 1: Signal Descriptions

Signal Name	Direction	Clock Domain	Description
S_AXI_*	AXI Lite bus	s_axi_aclk	AXI Lite interface to access the internal register space.
trg_a	input	async	Counter start trigger. Uses double flip-flop synchronization to synch to s_axi_aclk domain.
trg_b	input	async	Counter stop trigger. Uses double flip-flop synchronization to synch to s_axi_aclk domain.
clk_a	input	-	Clock input of trg_a clock domain. This input is required to elongate the pulses at the trg_a input.
clk_b	input	-	Clock input of trg_b clock domain. This input is required to elongate the pulses at the trg_b input.

5 Software Flow

The IP is programmed through the AXI Lite interface. Table 2 shows the available registers and functions. In normal operation, a trg_a event starts the counter and a trg_b event stops the counter. A counter reset must be issued explicitly by writing all zeroes to the COUNTER_VALUE register. For some special cases, it may be required to start or stop the counter manually by writing a ‘1’ to bit 0 of the SW_TRIG_START or SW_TRIG_STOP register. The IP will clear any written value to the SW_TRIG_* registers when the trigger is issued.

Table 2: Configuration Registers

Address Offset	Name	Access	Description
0x00	COUNTER_VALUE	RW	Latency counter register which counts up at a rate of 1 per s_axi_aclk cycle when started by a trg_a event. Counter is stopped by a trg_y event.
0x04	SW_TRIG_START	RW	Software start trigger. By writing a '1' to this register a trg_a event is triggered.
0x08	SW_TRIG_STOP	RW	Software stop trigger. By writing a '1' to this register a trg_y event is triggered.
0x0C - 0x1C	–	RW	Unused, can be written / read by user application.

Latency Measurement Application Procedure

1. Reset the COUNTER_VALUE register by writing '0'
2. Start any user specific tasks that generate both the start and stop trigger pulses at the AXI Event Counter IP core
3. Read the COUNTER_VALUE register to get the measured latency in axi_aclk ticks.
    a. (optional) check that the counter has effectively stopped counting by reading the COUNTER_VALUE register again and compare the values.
    b. If the counter is still running, stop it by writing a '1' to the SW_TRIG_STOP register.
4. Repeat from Step 1 for another measurement.

5 Example Design

Figure 2 shows one possible application scenario where the latency of an AXI Direct Memory Access IP is measured.

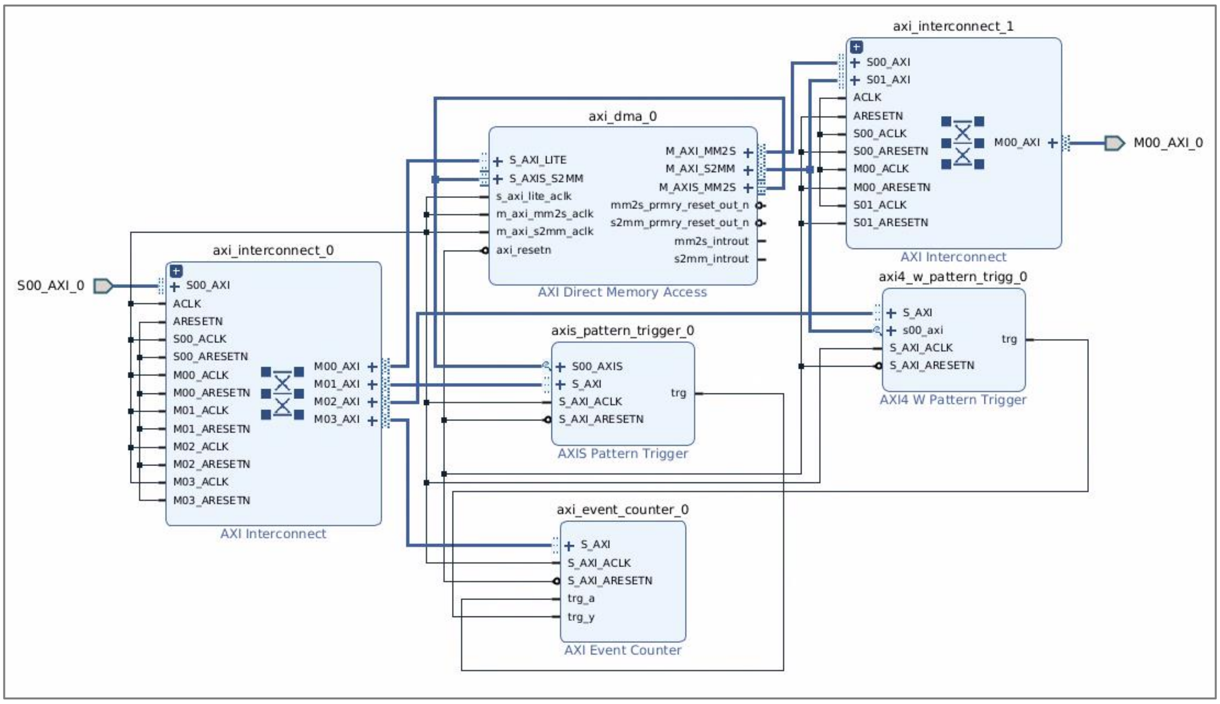

Figure 2: Example Design