Counter specification

Company : Elsys Eastern Europe
Author : Andrea Erdeljan
E-Mail : andrea.erdeljan@elsys-eastern.com
Mentor : Darko Tomusilovic

Revision History

Rev.	Date	Description
0.1	14.9.2015.	Initial

Table of Contents

1. Introduction

This counter design is Verilog RTL model which implements a 16-bit up/down counter with AXI4 full protocol support and interrupt signals.

Highlights

- 16-bit up/down counter
- AXI4 protocol compliant
- contains RIS, IM and MIS registers
- LOAD register
- counter enable, direction
- MATCH register
- IIR register
- reset counter on edge, SWRESET
- two clock domains

2. Interface

The counter interface is described in this section.

2.1 Parameters

Name	Type	Description
ID_WIDTH	Integer	Width of ID for for write address, write data, read address and read data
DATA_WIDTH	Integer	Width of AXI data bus
ADDR_WIDTH	Integer	Width of AXI address bus
AWUSER_WIDTH	Integer	Width of optional user defined signal in write address channel
ARUSER_WIDTH	Integer	Width of optional user defined signal in read address channel
WUSER_WIDTH	Integer	Width of optional user defined signal in write data channel
RUSER_WIDTH	Integer	Width of optional user defined signal in read data channel
BUSER_WIDTH	Integer	Width of optional user defined signal in write response channel

2.2 Signals

Name	Width	In/Out	Description
Full AXI4			
FCLK	1	Input	Counter clock
IRQ_O	1	Output	Interrupt (high when any of the interrupt bits in MIS are set, else low)
DOUT_O	1	Output	Is counter > LOAD (high when counter > LOAD, else low)
RESET_I	1	Input	Resets counter on rising edge
SWRESET_O	1	Output	High for one clock cycle when SWRESET occurs (writing the code (0x5a) to SWRESET register resets all registers and activates this signal)

3. Internal Registers

Name	Width	Description		Address
RIS	[DATA_WIDTH-1 : 0]	Register interrupt status, set by hardware when an interrupt occurs bit 0 – overflow bit 1 – underflow bit 2 – match	Read-only	0
IM	[DATA_WIDTH-1 : 0]	Interrupt enable bit 0 – overflow bit 1 – underflow bit 2 – match	Read-write	2
MIS	[DATA_WIDTH-1 : 0]	Masked intrrupt status bit 0 – overflow bit 1 – underflow bit 2 – match additional functionality: writing 1 to a correct bit location clears the flag in RIS and MIS registers, but only if PROT = 1 (writing 0 does nothing, when PROT = 0 nothing)	Read_write	4
LOAD	[DATA_WIDTH-1 : 0]	Counter is being compared to this value and DOUT_O set accordingly	Read-write	6
CFG	[DATA_WIDTH-1 : 0]	Configuration register bit 0 – counter enable bit 1 – up (0) or down (1)	Read-write	8
SWRESET	[DATA_WIDTH-1 : 0]	Writing the code (0x5a) resets all registers other writes do nothing read always returns 0	Read-write	10
IIR	[DATA_WIDTH-1 : 0]	Interrupt index register – priority register MIS[2:0]=1xx – value = 3 MIS[2:0]=01x – value = 2 MIS[2:0]=001 – value = 1 reading it clears the highest priority interrupt	Read-only	12
MATCH	[DATA_WIDTH-1 : 0]	Counter is being compared to this value. When equal to it the match interrupt occurs and RIS[2] is set	Read-write	14
count_aclk	[DATA_WIDTH-1 : 0]	Value of the counter	Read-only	16

*note: all unimplemented bits are read-only, value = 0

4. Clock domains

Two clock domains are used for this design:

1. AXI_ACLK – controls all the registers and the AXI interface

2. FCLK – controls the counter

Crossing clock domains is done using synchronizers, as follows:

- Overflow or underflow in the counter sent to RIS[0] or RIS[1] via the edge detector
- Match interrupt in the counter is sent to RIS[2] via the edge detector
- changes in the CFG register are sent to the counter via a simple 2 flip-flop synchronizer
- nrst signal (based on SWRESET register) is sent to the counter via the edge detector
- RESET_I (input reset on edge) is sent to the counter via the edge detector
- the dout_o value is set via a simple 2 flip-flop synchronizer
- count value from counter is sent to the count_aclk register and LOAD and MATCH values to the counter using a two-phase handshaking protocol (counter sets the req and put the data on the data_bus, when the req is sensed the data is collected, ack is raised and the LOAD and MATCH values placed on the bus (from the AXI_ACLK domain); when the counter senses the ack signal it can start a new transaction by lowering the req and when the data is collected the ack is lowered; when the counter senses the low ack it can initiate a new transaction, and so on...)

5. AXI4 transactions

AXI read and write burst requests should be as follows:

- DATA_WIDTH should be 16
- valid AxADDR values are 0, 2, 4, 6, 8, 10, 12, 14 or 16
- value of AxLEN should be 0 (burst length = 1 transfer)
- size should match the DATA_WIDTH (AxSIZE = 1 (16 bits))
- burst type should be FIXED (AxBURST = 2'b00)
- AxID should be 0
- all other signals can have any value

Any deviation from these specified values will result in the counter returning a SLVERR