gmZPU Wishbone Controller Datasheet

Koen Martens
kmartens@sonologic.nl

March 29, 2014

1 Introduction

The gmZPU Wishbone Controller (ZWC) is a configurable WISHBONE B4 [1] compliant bus master. The ZWC connects to memory buses of a cpu as slave, and responds to read and write requests from the gmZPU. It supports classic WISHBONE B4 pipelined single cycles. Future revisions may support additional cycle types.

2 Notation

BUS(1:0) Bit 1 downto 0 of bus BUS.

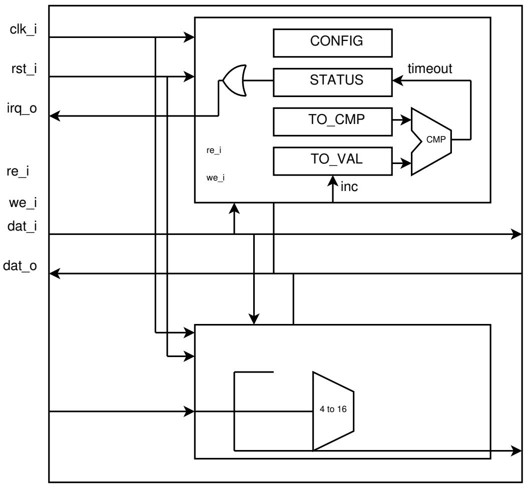

Figure 1: Block diagram

3 Overview

The top design unit is *zwishbone_controller*. Figure 1 contains the conceptual block diagram of the *ZWC*.

3.1 Design constraints

The *ZWC* can have a variable address and data width. To comply with the WISHBONE B4 specification [1], the data bus width must be 8, 16 or 32 bits.

The WISHBONE B4 [1] bus granularity is always equal to the data bus width.

The minimum address bus width configuration is shown in table 3.1. Any additional bits added to the width of the address bus extend the WISHBONE B4 [1] address bus width.

The MSB of the address bus decodes to the bus enable. If the bit is 1, the WISHBONE B4 [1] bus is selected. If the bit is 0, the controller registers are selected.

bit	address(6)=0	address(6)=1
6	0	1
5	reg_addr(5)	chipselect(3)
4	reg_addr(4)	chipselect(2)
3	reg_addr(3)	chipselect(1)
2	reg_addr(2)	chipselect(0)
1	reg_addr(1)	bus_addr(1)
0	reg_addr(0)	bus_addr(0)

Table 1: Minimal address bus width configuration

4 Registers

The CPU can control and check the operation of the ZWC by reading and writing to four registers:

CONFIG Controller configuration register

STATUS Controller status register

TO_CMP Timeout counter compare value

TO_VAL Timeout counter value

4.1 CONFIG

The configuration register selects the WISHBONE B4 [1] bus cycle type. The bits in this register are defined in table 4.1. The current revision only supports pipelined cycles. All bits except PIPELINE (bit 0) of the CONFIG register should be written as 0. The result writing 1 to one of these bits is undefined.

4.2 STATUS

The STATUS register reflects the following error conditions:

31 - 3	2	1	0
reserved (write 0)	RMW	BLOCK	PIPELINE

Table 2: CONFIG register bits

TO Time out - a WISHBONE B4 cycle was initiated but the strobed slave did not respond

RTY Retry - the WISHBONE B4 slave asserted wb_retry

ERR Error - the WISHBONE B4 slave asserted wb_error

The bits in this register are defined in table 4.2.

31 - 3	2	1	0
reserved	TO	RTY	ERR

Table 3: STATUS register bits

4.3 TO_CMP

The timeout compare register is compared with the TO_VAL register. If the registers are equal, a timeout signal is asserted.

4.4 TO_VAL

The timeout value register is an incrementing counter. It is reset to 0 at the start of a bus cycle. While the bus cycle is in progress, the register is incremented on each rising edge of the system clock.

The timeout value register is compared with the TO_CMP register. If the registers are equal, a timeout signal is assertd.

A timeout ends the bus cycle and sets the TO bit in the STATUS register. If the cycle was a read cycle, the value read by the cpu is undefined. If the cycle was a write cycle, no assumptions can be made about the current state of the slave. Specifically, no assumptions can be made regarding whether the value driven on the WISHBONE B4 bus was written to the destination within the slave or not.

5 Timing

This section contains timing diagrams.

5.1 Register timing

The CPU performs a register read in two cycles (see figure 2):

1. On the first rising clock edge, the cpu drives the address bus and asserts *re_en*. The controller asserts *busy*.

2. On the second rising clock edge, the cpu releases the address bus and de-asserts *re_en*. The controller asserts *ready* and drives the data bus.

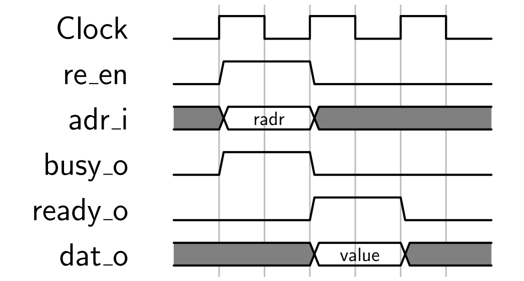

Figure 2: Register read timing

The CPU performs a register write in one cycle, the register will be loaded on the second cycle (see figure 3):

1. On the first rising clock edge, the cpu drives the address and data buses and asserts *we_en*. The controller prepares to latch the value on the data bus into the selected register.

2. On the second rising clock edge, the cpu releases the address and data buses and de-asserts *we_en*. The controller latches the value into the selected register.

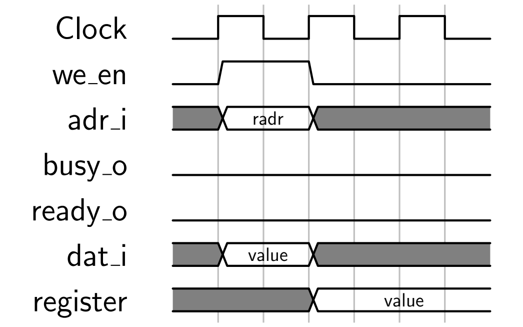

Figure 3: Register write timing

6 Application

The reference design is the gmZPU [2] System-on-Chip (SoC). The reference application of the design entity has a data width of 32 bits and an address width of 14 bits. A number of on-chip peripherals are connected to the gmZPU cpu by the ZWC.

References

[1] OpenCores *Wishbone B4, WISHBONE System-on-Chip (SoC)Interconnection Architecture for Portable IP Cores*

http://opencores.org/opencores,wishbone 2010.

[2] Sonologic *gmzpu github repository* http://github.com/sonologic/gmzpu.