I2C Slave to Wishbone Master module

Carlos Gil Soriano
BE-CO-HT
carlos.gil.soriano@cern.ch

February 23, 2012

Abstract

An I2C Slave to Wishbone Master module is depicted in this document. The module is targeted for Spartan 6 devices and written in VHDL.
The following subjects are addressed:

- The registers to control the module.
- Step-by-step instructions for proper use. access.

Revision history		
HDL version	**Module**	**Date**
0.1	I2C slave	February 23, 2012

Contents

1 Structure

The i2c module contains several blocks related the following way:

– i2c_slave_top.vhd
—– i2c_regs.vhd
—– i2c_slave_core.vhd
———— FIFO_dispatcher.vhd
———— FIFO_stack.vhd
———— gc_counter.vhd
———— gc_ff.vhd
———— i2c_bit.vhd

2 Interrupting lines offered

2.1 ind_wb_addr

This interrupting signal issues when a *indirect wishbone address* has been received. It is notified right after the first CTR0[BIA] + 1 bytes upon the reception of the I2C byte address packet. This signal is vital for correctly prefetching when a I2C read operation is requested.

2.2 inst_rd

This signal is issued when a read operation directed by an external master over the HDL slave core is finished. That means it will be generated after the last data byte has been sent by the HDL core.

2.3 inst_wr

This signal is issued when a write operation directed by an external master over the HDL slave core is finished. That means it will be generated after the last data byte has been received.

3 Registers

3.1 STA

The STA register is a read-only register. It control the general enable and reset of the module. It also contains the current value of the finite state machine of the i2c modue (useful for easy debugging).

Bits	Field	Meaning
0	EN	General ENable
1	RST	General ReSeT
2	RD_WRN_INST	Reserved
3	A_RX	
4	A_TX	
8-5	x	Reserved
15-9	i2c_sla_fsm	i2c fsm
31-16	Not used	

3.2 PRE

The PRE register is a write-read register. Right now it is not used.

Bits	Field	Meaning
15-0	PRE	PREscaler value
31-16	-	Not used

3.3 CTR0

The CTR0 register is a write-read register. It controls the indirect addres and holds the I2C address (which in the case of *CONV-TTL-BLO* will be connected to VME64x geographical address pins).

Bits	Field	Meaning
0	EN	general ENable
1	RST	general ReSeT
2	PEN	Prescaler ENable
5-3	x	Reserved
7-6	BIA	Bytes Indirect Addressing
14-8	A[6:0]	I2C address
15	x	Reserved
31-16	-	Not used

3.4 CTR1

The CTR1 register is a write-read register. It shows the fsm of the separate *read* and *write* fsms.

Bits	Field	Meaning
7-0	RDS	fsm status: ReaD Status
15-8	WDS	fsm status: WRite Status
31-16	-	Not used

3.5 DRXA

The DRXA register is a read-only register. It holds the last four received bytes through the I2C. DRX0 has the most recent byte received from the serial interface. DRX3 has the oldest byte received.

Bits	Field	Meaning
7-0	DRX0	Data RX register 0
15-8	DRX1	Data RX register 1
23-16	DRX2	Data RX register 2
31-24	DRX3	Data RX register 3

3.6 DRXB

The DRXB register is a read-only register. It holds the fifth and sixth latest received bytes, respectively.

Bits	Field	Meaning
7-0	DRX4	Data RX register 0
15-8	DRX5	Data RX register 1
31-16	-	Not used

3.7 DTX

The DTX register is a write-read register. It shows the fsm of the separate *read* and *write* fsms.

Bits	Field	Meaning
7-0	RDS	fsm status: ReaD Status
15-8	WDS	fsm status: WRite Status
31-16	-	Not used

4 Internal Memory Mapping

The internal registers map is as follow:

Address	Register	Access
0x0	*STA*	Read-only
0x1	*PRE*	Write-read
0x2	*CTR0*	Write-read
0x3	*CTR1*	Write-read
0x4	*DRXA*	Read-only
0x5	*DRXB*	Read-only
0x6	*DTX*	Write-read

5 How to use it

5.1 Initialization

1. Perform a reset of the module while module is not enabled:
   CTR0: write 0 to EN and 1 to RST.

2. Load the prescaler:
   PRE: set a new value.

3. Set the I2C address of the slave module:
   CTR0[A]: set the I2C address.

4. Set the rest of bits of CTR0, including EN:
   CTR0: set rest of bits.

5.2 Indirect Write from Master to Slave

It is a one-phase transaction: one indirect writing is achieved by signaling only one I2C start condition by the master.

1. The Master I2C device starts an I2C transaction. In the first byte it specifies the type of transaction issued as a write.

2. Then, two bytes are received in the slave. At the end of the reception of the last bit of this second byte (third since the I2C start condition), the finite state machine in *i2c_slave_core.vhd* launches the interrupt *inst_wb_addr*.

   Address prefetching: at this point, the Wishbone Address can be stored. It is found in *DRX0* and *DRX1* registers:

   - *DRX0*: holds the Wishbone Address Lowest Byte
   - *DRX1*: holds the Wishbone Address Highest Byte

3. Following the reception of the two bytes corresponding to the Wishbone Address, four more bytes will be received. They are the data bytes. Once the last bit of this fourth byte is received (seventh byte since the I2C start condition), the finite state machine in *i2c_slave_core.vhd* launches the interrupt *inst_wr*.

   Address and Data fetching: at this point, the *Wishbone Address* and the *Data* to be written in that address can be both fetched through the *DRX* registers:

   - *DRX0*: holds the Data Lowest Byte
   - *DRX1*: holds the Data 2^{nd} Lowest Byte

- *DRX2*: holds the Data 2nd Highest Byte
- *DRX3*: holds the Data Highest Byte
- *DRX4*: holds the Wishbone Address Lowest Byte
- *DRX5*: holds the Wishbone Address Highest Byte

4. The Master I2C device stops the I2C transaction.

5.3 Indirect Read from Master to Slave

It is a two-phases transaction: one indirect read is achieved by signaling only two I2C start conditions by the master.

FIRST PHASE

1. The Master I2C device starts an I2C transaction. In the first byte it specifies the type of transaction issued as a write.

2. Then, two bytes are received in the slave. At the end of the reception of the last bit of this second byte (third since the I2C start condition), the finite state machine in *i2c_slave_core.vhd* launches the interrupt *inst_wb_addr*.

   Address Prefetching: at this point, the Wishbone Address can be stored. It is found in *DRX0* and *DRX1* registers:

   - *DRX0*: holds the Wishbone Address Lowest Byte
   - *DRX1*: holds the Wishbone Address Highest Byte

   Data Prefetching: it is a good practice to do the *data prefetching* of the Wishbone Address (in case this is accessible). The control logic attached to the *i2c_slave_wb* module should perform a wishbone write to the four *DTX[X]* registers:

   - *DTX0*: holds the Data Lowest Byte
   - *DTX1*: holds the Data 2nd Lowest Byte
   - *DTX2*: holds the Data 2nd Highest Byte
   - *DTX3*: holds the Data Highest Byte

   so that the data in the transmission registers is up-to-date, in order to be sent through I2C.

SECOND PHASE

1. The Master I2C device (re)starts an I2C transaction. In the first byte it specifies the type of transaction issued as a read. At the end of the reception of the last bit on the first byte of this *second phase* (third byte sinc the I2C start condition from the *first phase*), the finite state machine in *i2c_slave_core.vhd* launches the interrupt *inst_wb_addr*.

2. The *i2c_slave_wb* module sends the four data bytes in the following order:

   (a) Data Lowest Byte
   (b) Data 2^{nd} Lowest Byte
   (c) Data 2^{nd} Highest Byte
   (d) Data Highest Byte

3. Once the last byte has been already send, the finite state machine in *i2c_slave_core.vhd* launches the interrupt *inst_rd_addr*.