GLADCI - Latin America Group Development Integrated Circuit

Authors: Felipe Fernandes da Costa
Ronal Dario Celaya
Francismar Mariano de oliveira

Contents

Revision	Date	Author	Revised	Comments
0.1	20/01/2014	Felipe F. da Costa	-	Creating a file description IP
0.2	21/01/2014	Felipe F. da Costa	-	Added complete description block
0.3	23/01/2014	Felipe F. da Costa	-	Revising and added a description to figures on I²C Block
0.4	06/02/2014	Felipe F. da Costa	-	Added more description-about module and blcok use more a Frequency work

List of Tables

1 Introduction

1.1 Description

I^2C is a multimaster protocol used to simplify use from uart and defined by philips like a standart module to interface with analogic devices. Here is presented a solution using a interface with APB protocol definided by the ARM. I^2C in general transport 8 bit data through bidiretional ports SDA and use clock SCL to give a pulse and control transport data. But This I2C has propurse is to transport 16 bit data. Figure 1 show a Top block using APB and I^2C and Table 1 show the pinout description used for each block.
I^2C is used in some applications, among them:

1. Reading configuration data from SPD EEPROMs on SDRAM, DDR SDRAM, DDR2 SDRAM memory sticks (DIMM) and other stacked PC boards
2. Supporting systems management for PCI cards, through an SMBus 2.0 connection
3. Accessing NVRAM chips that keep user settings
4. Accessing low speed DACs and ADCs
5. Changing contrast, hue, and color balance settings in monitors
6. Changing sound volume in intelligent speakers
7. Controlling OLED/LCD displays, like in a cellphone
8. Reading hardware monitors and diagnostic sensors, like a CPU thermostat and fan speed
9. Reading real-time clocks
10. Turning on and turning off the power supply of system components
11. A particular strength of I^2C is the capability of a microcontroller to control a network of device chips with just two general purpose I/O pins and software

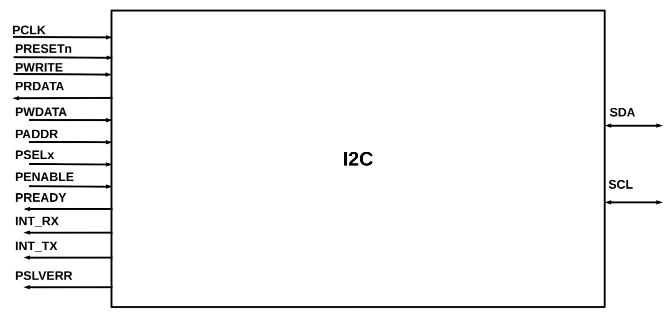

Figure 1: APBI2C top block

Table 1: Pinout description

Signal name	Direction	Size	Description pin
PCLK	Input	1	Clock system
PRESETn	Input	1	Reset is active at LOW
PWRITE	Input	1	When HIGH is write on I^2C, LOW is read operation
PENABLE	Input	1	APB set it notice I^2C data is ready to read or write
PREADY	Output	1	I2C response to APB block data is ready to be read or write
PSELx	Input	1	Pin used to select I^2C
INT_RX	Output	1	Interruption used to notice RX FIFO is EMPTY
INT_TX	Output	1	Interruption used to notice TX FIFO is EMPTY
PSLVERR	Output	1	Used to notice a write or read whitout INT_RX or TX HIGH
PADDR	Input	32	Address used to reference read data or write
PWDATA	Input	32	Input used to write on FIFO TX
PRDATA	Output	32	Output used to read data from FIFO RX
SDA	Inout	1	Bi-diretional data transport
SCL	Inout	1	Bi-diretional Clock

2 Sub-blocks

2.0.1 APB Core

The APB is part of the AMBA 3 protocol family. It provides a low-cost interface that is optimized for minimal power consumption and reduced interface complexity. The APB interfaces to any peripherals that are low-bandwidth and do not require the high performance of a pipelined bus interface. The APB has unpipelined protocol. All signal transitions are only related to the rising edge of the clock to enable the integration of APB peripherals easily into any design flow. Every transfer takes at least two cycles. Figure 2 show protocol to write on APB, figure 3 show protocol to read from APB and figure 4,5 sample a read PSLVERR fail when APB try read data where in the same way is to write protocol.
To access the TX FIFO write is necessary to set the PADDR in 0h and pwrite should be set at a high level so that it is enabled in the mode of writing and performing well in writing PWDATA. To read the RX FIFO is necessary to set PADDR to 4h and pwrite should be low. And lastly writing logger configuration in which PADDR should be set at 8h and pwrite should high level.

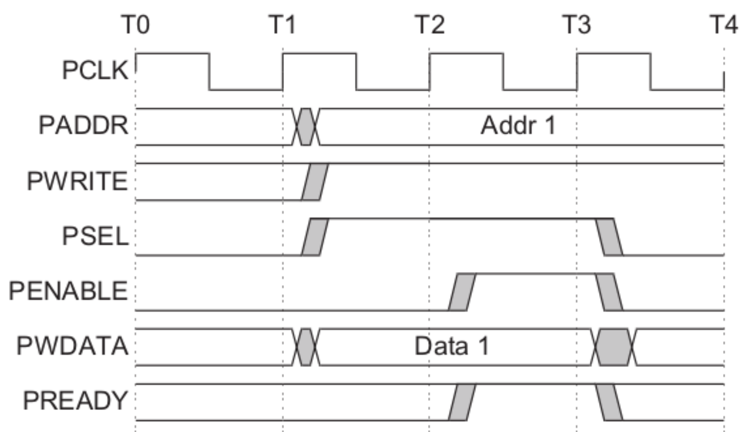

Figure 2: APB Protocol write

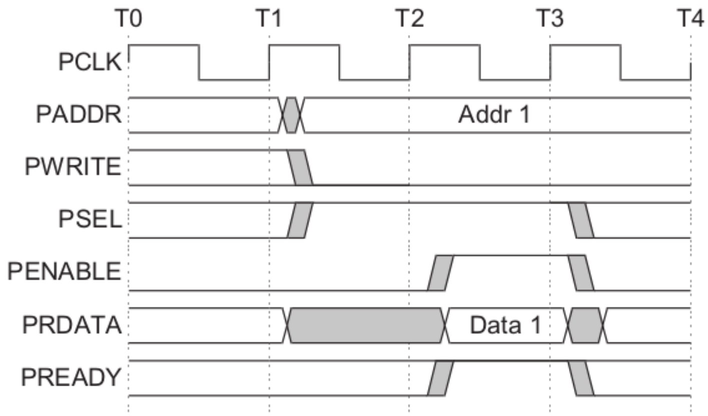

Figure 3: APB Protocol read

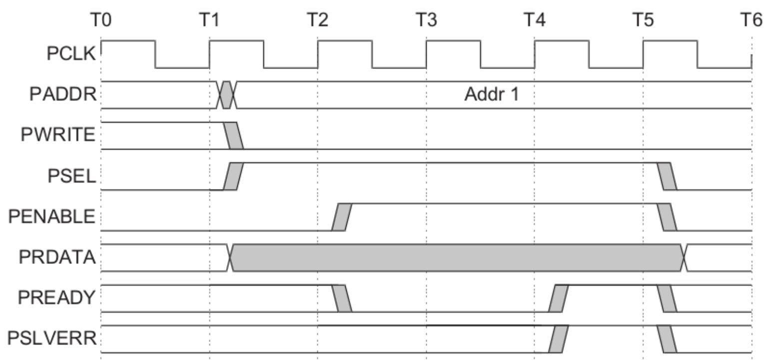

Figure 4: APB Protocol read ERROR

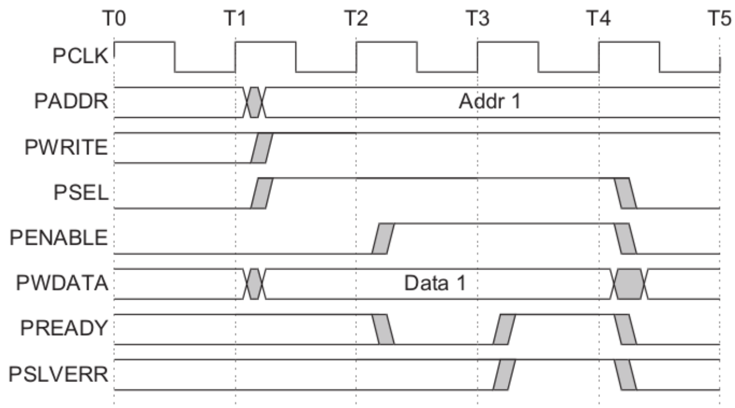

Figure 5: APB Protocol write ERROR

2.0.2 FIFO RX/TX

First In First Out or FIFO was modified to suit the use of the I²C. Being that their signals are simply and in part are used for I²C module to start operating. The FIFO principle to have 16 registers of 32 bits is stored where the pattern of transmission of data.The proposal is work with the FIFO full and FIFO when not completely full.

2.0.3 I²C MODULE

The I2C module to boot operations need the FIFO has any data to be transported and received as well as your your registry properly configured setup. Like other modules present in opencores well as companies in the I²C specification supports the basic operations using basic protocols that will be described later. The principle is used a block of default cominicação with a configuration register that two bits are used to determine the mode of operation and 12 bits to determine the maximum frequency used may not exceed the clock used in the system.
On table 2 we show standard protocol used by many chip designs and your respective means across the wave form signal.

Table 2: Protocol Description

Protocol	Description
A	Start bit used to notice block control we are starting a transmission / receiving
Control	Used to send what peripheral is to be selected
Address	Where is going to be writed
Data	Data to be write
ACK	If all goes right this signal must be LOW for each byte
NACK	If not all goes right this signal must be HIGH for each byte
R	This is a restart condition when we re try send a byte to I^2C
S	Stop bit condition used when we finish a transmission packet

Figure 6 shows a pattern transfer data between blocks of I^2C. For each byte transferred can be seen that there is a ACK and the end of the transfer start and the transfer there is one start bit and one stop bit.

Figure 6: I^2C starndard transmission protocol

Figure 7 shows the attempted transfer / read data. Not necessarily need to be this way implementation. What should be illustrated here is that at any time during data transfer can be a NACK and it will be necessary to retransmit the byte, ie will be remade byte transfer as many times as necessary like showed on figure 8 where we have a restart bit operaion.

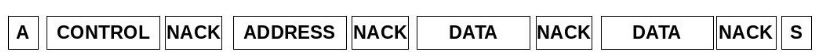

Figure 7: I^2C non starndard transmission fail protocol

Figure 8: I^2C starndard transmission fail protocol

3 I²C Top Block operation

For a successful operation for the I²C module occurs is necessary to obey the following rules:

1. The module in the data write operation after a reset and finally the configuration register with the clock being generated and TX enabled enabler must be written. When the module finishes transmitting all data that is stored in the FIFO then the interrupt is enabled INT_TX warning that the FIFO is ready to receive more data. Since the attempt of written data into the FIFO before the interrupt is generated at high PSLVERR causes what is recognized as an error.

2. The rules for the module is receiving data the same way as described for transmission mode. The only thing that differs is the interrupt used to warn that the data has arrived.

3. For reading and writing of data are used two independent FIFOs which in turn can write one another and can only be read. That is, if you can not read it again since it was already written . The same goes for the logger configuration .

Table 2 shows how is register configuration:

Table 3: Register Configuration Description

Register	13 12 11 10 9 8 7 6 5 4 3 2 1 0
TX - 0	If bit 1 is HIGH TX operation is enable
RX - 1	If bit 1 is HIGH RX operation is enable
CLOCK REGISTER - 2 to13	Counter used to regulate clock used to propagate data,this must be handle with care beacuse this clock can not exceed your global clock

3.0.4 Clock range work

Initially this module should work with frequencies from 100 kHz to 5 MHz But this should be verified in FPGA if it can work in the desired frequencies.

4 Final considerations

The I²C module is still in development and has yet to be verified their functionality. This document need still show the waveforms and a detailed explanation of the waveforms of the transmission and reception of I²C.

5 References