ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

DESIGN OF I2C-APB PROTOCOL

S. Devakrupa*, Drakshayani Desai

[*] M.Tech, Digital Systems & Computer Electronics Swami Vivekananda Institute of Technology Hyderabad

Assistant Professor, Dept of ECE Swami Vivekananda Institute of Technology Hyderabad

DOI: 10.5281/zenodo.167006

ABSTRACT

I2C is a two wire, bidirectional serial bus that provides effective data communication between two devices. Problem starts when one module follows different protocol as others and each module has its different bit rate or baud rate of data transfer which can be either asynchronous or synchronous. I2C bus supports many devices and each device is recognized by its unique address. The proposed architecture is a bridge between I2C Master and APB Salve. The data travels from a serial bus (I2C) to parallel bus (APB) to serial (I2C) in sync with the respective domain clock. This forms a bidirectional interface between I2C supported module and APB supported module.

KEYWORDS: I2C, APB, Communication, Devices, Interface.

INTRODUCTION

The I2C Interface (I2C) provides full support for the two-wire I2C synchronous serial interface, compatible with the ACCESS with additional support for the SM Bus protocol including Packet Error Checking (PEC). I2C compatibility provides a simple interface to a wide range of low cost memories and I/O devices like EEPROMs, SRAMs, D/A converters, A/D converters and peripheral drivers. I2C is the same I2C Interface IP proven in high-volume devices from National Semiconductor.

The host interface of the I2C Interface complies with the AMBA 2 APB protocol. The Control registers provide CPU control of Serial Clock Line (SCL) frequency start and stop condition generation, PEC byte generation, I2C address assignment, enabling/disabling interrupts & 7-bit or 10-bit addressing. A serial data register shifts the serial I2C data into and out of the I2C Interface during receive and transmit operations. Status registers indicate current operating mode, packet error, and interrupt status.

The I2C serial interface consists of the standard bidirectional I2C signals those are Serial Clock Line (SCL) and Serial Data Line (SDA). I2C IP level has separate unidirectional signals (scl_in_pin/scl_out and sda_in_pin /sda_out). For reducing chip-level pin count, I2C bus interface signals can be shared with other on-chip functions through a General Purpose I/O Controller (GPIO).

I2C protocol uses a two-wire interface for bidirectional communications between the devices connected to the bus. SCL and SDA are connected to a positive supply through pull-up resistors. The devices connected to the bus uses open-collector or open-drain drivers and can only pull the respective bus line low. So both the SCL and SDA are wired-AND type signals.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

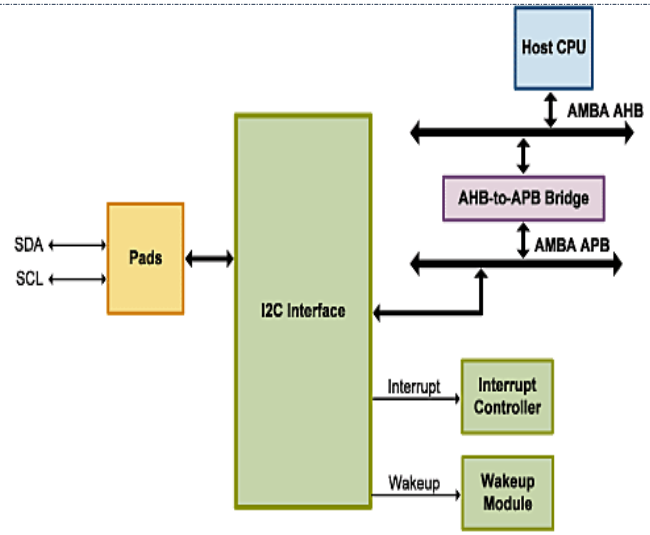

Fig 1: I2C interface

I2C DATA TRANSFERS

Start and Stop conditions in I2C

Each and every data transfer should begin with start and end with stop conditions in I2C. Start condition starts when SDA line goes from high to low and SCL is high. When SCL is high and a low to high transaction stares the stop condition.

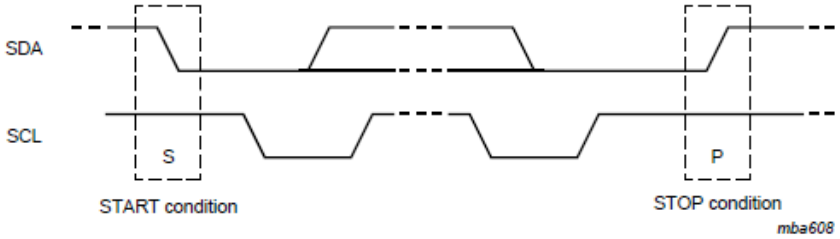

Fig 2: Start and Stop conditions

The master generates the start and stop conditions. After the start condition the bus is said to be busy and will be free after the stop condition. If repeated start conditions are generated the bus stays in busy.

Byte Format

Every byte on the SDA line contains eight bits long. There is no restriction for number of byte transferred. An acknowledge bit is accompanied with each byte and the data will transferred starting from MSB bit.

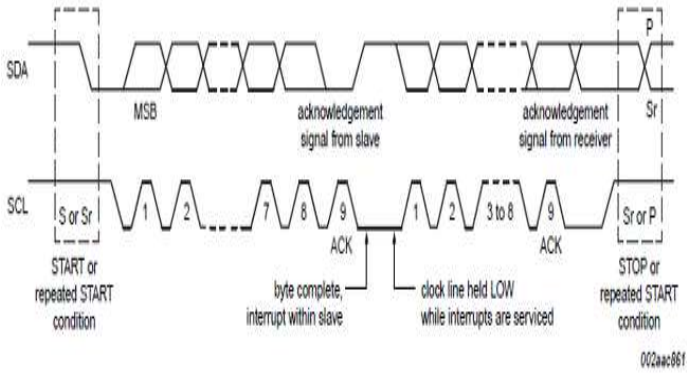

Fig 3: Byte transfer format

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

APB PROTOCOL

The first AMBA buses are advanced system bus(ASB) and advanced peripheral bus(APB). APB is used for low bandwidth control i.e. register interfaces on system peripherals. This APB bus has also an address and data phase similar to AHB with reduced complexity signal list.

Generally APB has two write transfers they are
APB write transfer with no wait states
APB write transfer with wait states.

APB operating states

The state machine operates through the following states:
IDLE: This is the default state of the APB.
SETUP: When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, PSELx, is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock.
ACCESS: The enable signal, PENABLE, is asserted in the ACCESS state. The address, write, and select signals all remain stable during the transition from the SETUP to ACCESS state. The PREADY signal from the slave controls the exit from the ACCESS state:
• If PREADY is held LOW by the slave then the peripheral bus remains in the ACCESS state.
• If PREADY is driven HIGH by the slave then the ACCESS state is exited and the bus returns to the IDLE state if no more transfers are required. Alternatively, the bus moves directly to the SETUP state if another transfer follows.

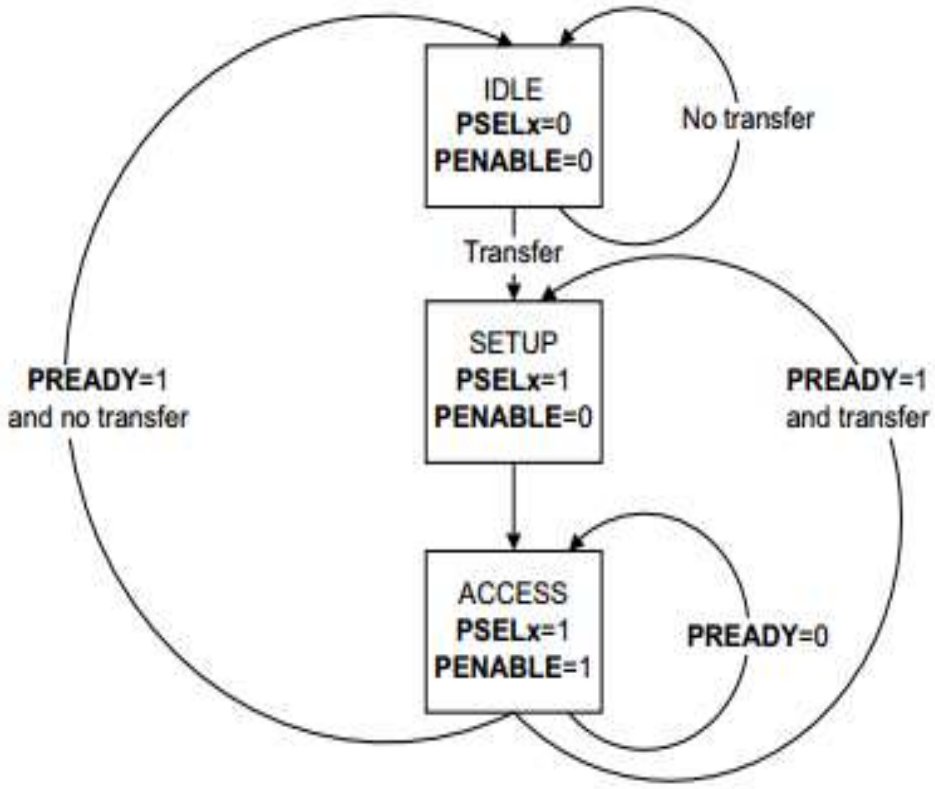

Fig 4: APB operating states

BLOCK DIAGRAM

The communication bridge between I2C and APB is shown in below figure. This architecture contains I2C slave and APB master. As we know that, the master initiates the data transfer, the I2C master sends the data to I2C slave and from slave it is transferred to APB master. From APB master the data sends to APB slave.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

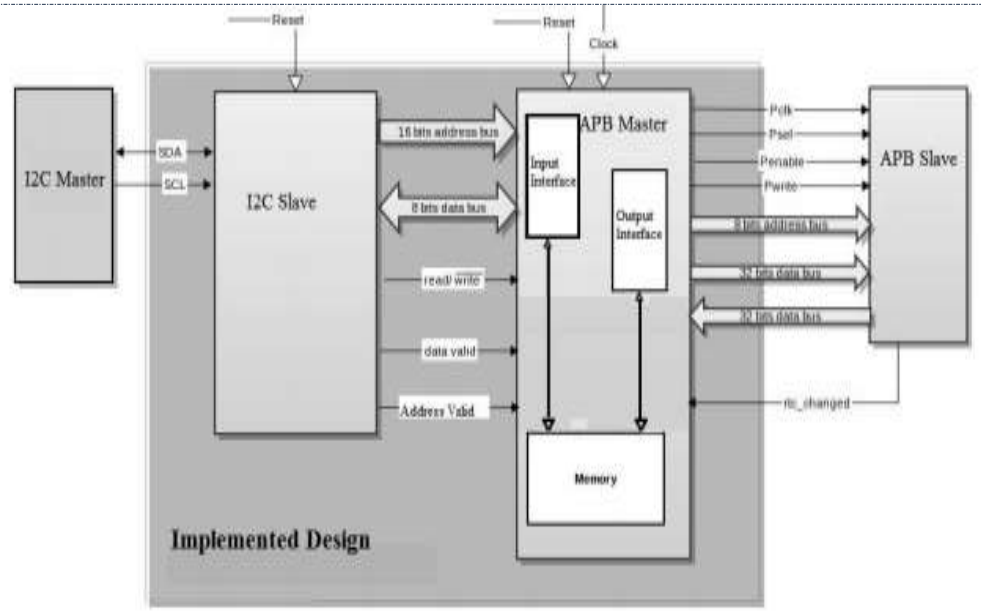

Fig 5: Communication bridge of I2C and APB

The I2C master needs to communicate with APB slave via I2C slave only.
The data transfer is checked out by APB master and it is transferred to APB slave.
The data and address valid signals are given by I2C slave signal.
To read data, communication is done via APB master to I2C slave to I2C master.
An acknowledgement is sent to APB master for APB slave to read the data signals.

RESULTS

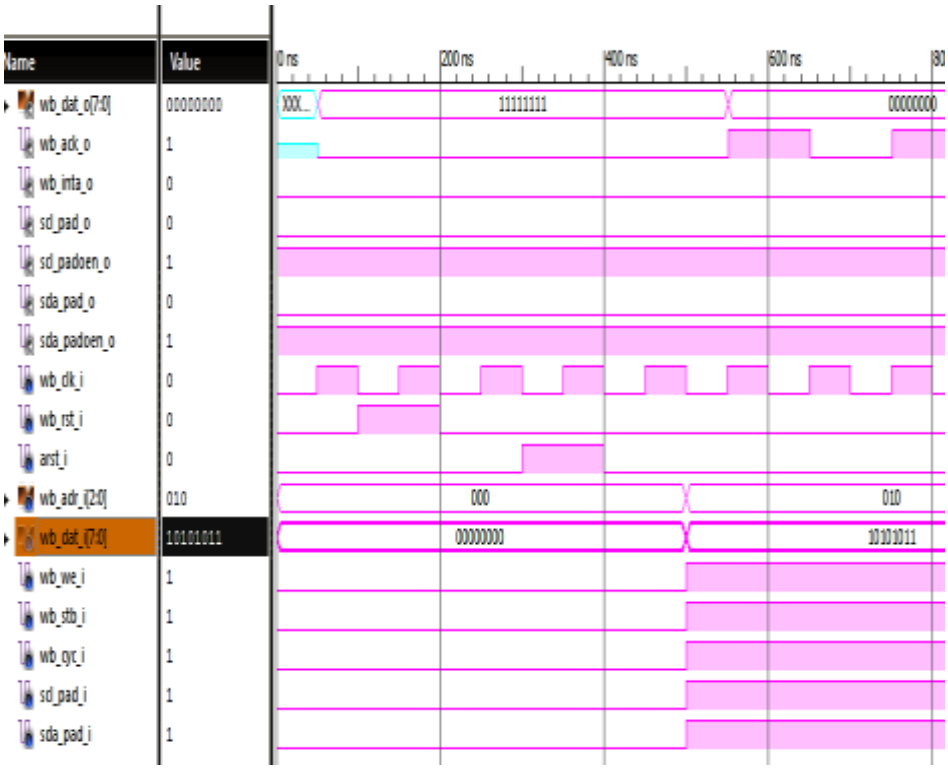

Fig 6: Simulation result for the communication bridge

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

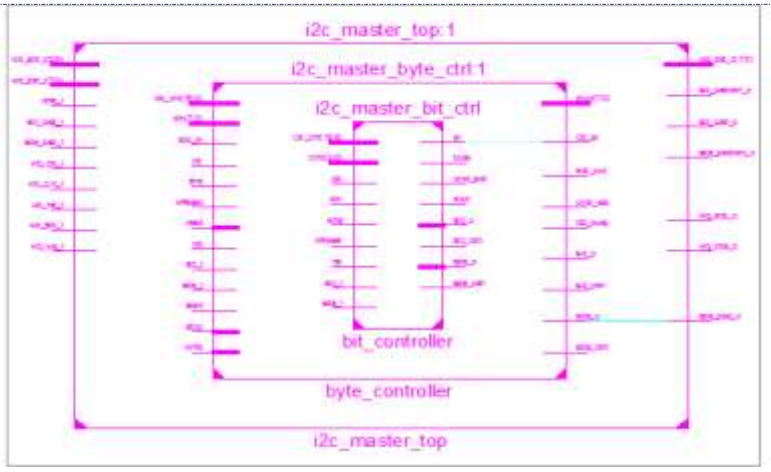

Fig 7: Top view of design

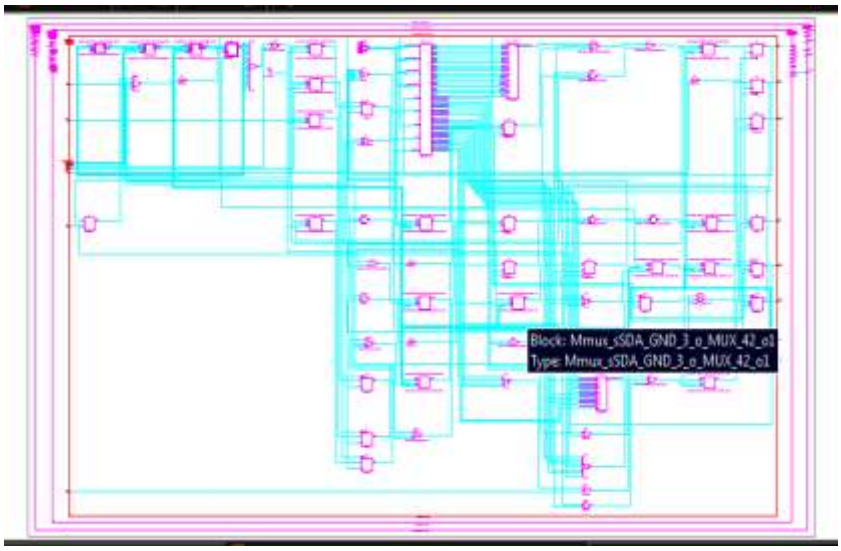

Fig 8: RTL Schematic

i2c_master_top Project Status (07/28/2016 - 15:32:34)			
Project File:	ic2.xise	Parser Errors:	No Errors
Module Name:	i2c_master_top	Implementation State:	Synthesized
Target Device:	xc7vx330t-3ffg1157	• Errors:	No Errors
Product Version:	ISE 13.4	• Warnings:	2 Warnings (2 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	127	408000	0%
Number of Slice LUTs	199	204000	0%
Number of fully used LUT-FF pairs	119	207	57%
Number of bonded IOBs	33	600	5%
Number of BUFG/BUFGCTRLs	1	32	3%

Detailed Reports	[-]

Fig 9 : Synthesied report of the design.

CONCLUSION

The implemented communication bridge between I2C and APB was designed and implemented in Xilinx ISE 13.4, Isim, using Verilog HDL. This project demonstrates how I2C Master Controller (Master) transmits and receives data to and from the (Slave). So that any low speed peripheral devices can be interfaced using I2C bus protocol as master. In future, this can be implemented as real time clock in networks that contains multiple masters and multiple slaves to co-ordinate the entire system by clock synchronization techniques. Simulation results are verified and data transfer from I2C master to APB slave can be clearly seen in provided simulation results.

REFERENCES

[1] Enhanced Universal Serial Communication Interface (eUSCI) – I2C Mode, SLAU425B–August 2012–Revised February 2013, Texas Instruments Incorporated.
[2] M. Alassir, J. Denoulet, O. Romain & P. Garda "A SystemC AMS Model of an I2C Bus Controller", IEEE.
[3] J. J Patel, B.H. Soni, "Design and Implementation of I2C Bus Controller Using Verilog" ISSN: 0975 – 6779| NOV 12 TO OCT 13 | VOLUME – 02, ISSUE – 02
[4] Samir Palnitkar, Verilog HDL, second Edition.
[5] UM10204, I2C-bus specification and user manual, Rev. 4— 13 February 2012, NXP Semiconductors.
[6] AMBA® APB Protocol Specification, Version: 2.0, 13 April 2010, issue C, http://www.arm.com.
[7] D.B.Rane, Ahmed Mustafa M. I. Shaikh, Devanand Mahajan, Mehdi Ali, "Priority Based I2C Bus Controller" (IJCTEE) Volume 3, Special Issue, March-April 2013.
[8] Bacciarelli, L. Lucia, G. ; Saponara, S. ; Fanucci, L. ; Forliti, M., "Design, testing and prototyping of a software programmable I2C/SPI IP on AMBA bus" , IEEE Research in Microelectronics and Electronics 2006, Ph. D.
[9] "NXP I2C-bus and SMBus controller PCA9564", www.nxp.com.
[10] Chetan Sharma, Abhishek Godara, "AHB interface with SPI master by using verilog", International Journal of Advances in Engineering Research, 2011, Vol. No. 2, Issue No. VI, December.
[11] Product brochure, "National Semiconductor I2C Interface", www.ipextreme. com.
[12] "DB-I2C-M-APB-DS-V1.1", APB Bus I2C Master Controller, Digital Blocks, Inc., September 2013.