ISSN: 2456-043X

International Journal of Engineering Science and Generic Research (IJESAR)
Available Online at www.ijesar.in
Volume 2; Issue 6; November-December; 2016; Page No. 16-20

Designing Architecture for Communication between I2C (inter integrated circuit) and AHB (Advanced High Performance Bus

Rubina Sultana [1], G.ARUNA[2]

[1] PG Scholar, Department of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, India

rubinasultana786@gmail.com

[2] Professor & HOD, Department of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, India

Abstract

The architecture which defines how data are transferred from one protocol to another. It exploits the flexible protocols of I2C to make it compatible with APB protocol. The 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, APB have low frequency .The proposed architecture is a bridge between I2C and AHB bus, this bridge communicate the High performance AHB parallel bus to I2C serial bus.

Keyword: SCL, SDA, APB, AHB.

I. INTRODUCTION

I2C is a 2-wire, SDA and SCL, bus developed with the basic idea of connecting different ICs and application specific modules with processors on a common communication network. I2C is a multi-master bus and the peripheral devices are addressed by unique programmable address. By sampling the SDA above Nyquist rate I2C can communicate with any device. On the other hand APB is a low bandwidth bus with reduced interface complexity. APB has dedicated programmable control registers to access peripherals devices. Like I2C, APB compatible devices are easily incorporated in any design flow.

A. I2C Protocol

I2C bus runs on simple master-slave relationship. All trans-actions begin with detection of START condition and are terminated by encountering STOP condition. As soon as start condition arises bus is considered to be busy and it will re-main in the same state till all requests for the bus have been granted. For the read/write operation, first the slave's address is sent followed by the corresponding data, as shown in figure 1. ACK signal is sent after successful transfer of each data byte. For interrupted transmission NACK signal is raised.

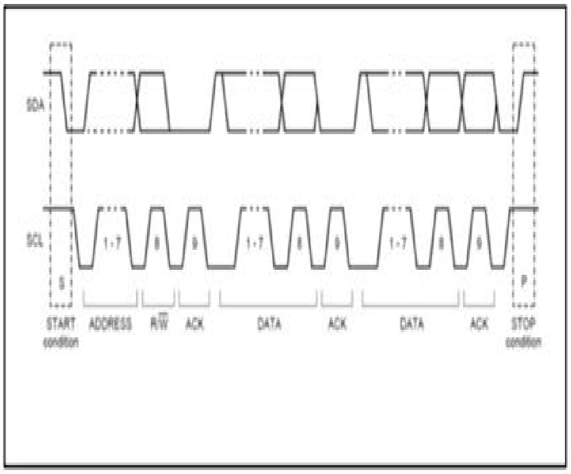


Figure 1: I2C Data Transfer Protocol.

B. APB

Write Protocol Fig. 2 explains the write cycle as- IDLE: The default state. SETUP: When transfer is required the bus moves into the SETUP state, where the select signal, PSELx, is asserted. The bus remains here for one clock cycle and moves to the ENABLE state on the next rising edge of the clock. ENABLE: The enable signal, PENABLE, is asserted. The address, write and select signals have to remain stable during the transition from the SETUP to ENABLE state. If no further transfers are required the bus returns to the IDLE state. Alternatively, if another transfer is to be made then the bus will move to SETUP. Address, write and select signals can glitch during transition.

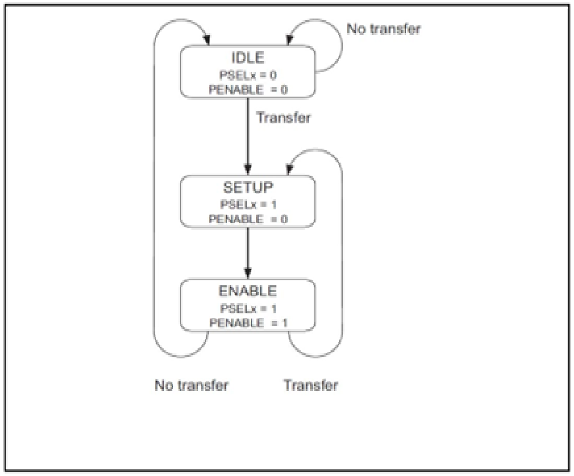


Figure 2: State Diagram for write operation on APB.

II. I2C Master and APB Salve bridge

Architectural block diagram of the implemented communication bridge between I2C and APB is shown in Fig. 3. The Structure contains two main blocks, i.e. I2C Slave and APB Master. I2C Slave takes the data from I2C Master in respective format and provides it to APB Master. This APB Master further sends out this data to APB Slave in APB Protocol. In this way a communication between I2C Master and APB Slave is done.

A. Write Operation

- Whenever I2C Master needs to communicate with APB Slave it would be done via I2C Slave.
- I2C Slave will assert Data Valid and Address Valid signals.
- Seeing these signal high, designed APB Master polls the memory for its availability and starts APB write state machine.
- I2C sends four chunks of 8-bit data serially to be written on APB Memory at four consecutive addresses.
- After transfer of each byte APB Master keeps a check on count whether all four memory locations are updated successfully.
- As soon as the data at APB Master is updated it transfers the same 32-bit data to APB Slave.

B. Read Operation

- Here again when I2C need to read data from the APB Slave, communication will take place via APB Master to I2C Salve to I2C master.

- APB Salve will send a signal to APB Master telling that the data are available to be read.

- APB Salve then transmits the data to APB Master where it is stored in the internal memory to be fetched by I2C Slave at time point of time.

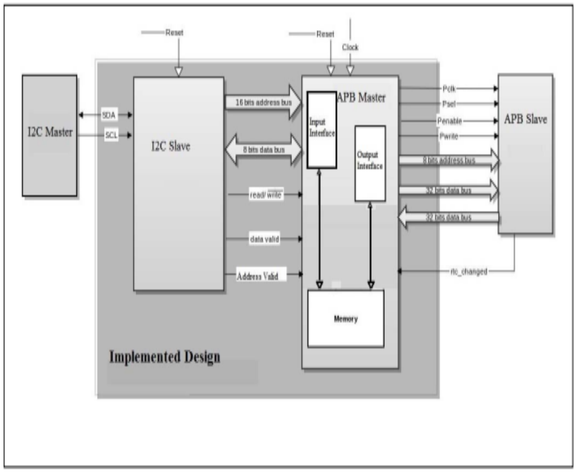


Figure 3: Block Diagram of Proposed Communication Bridge

III. AHB

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation.

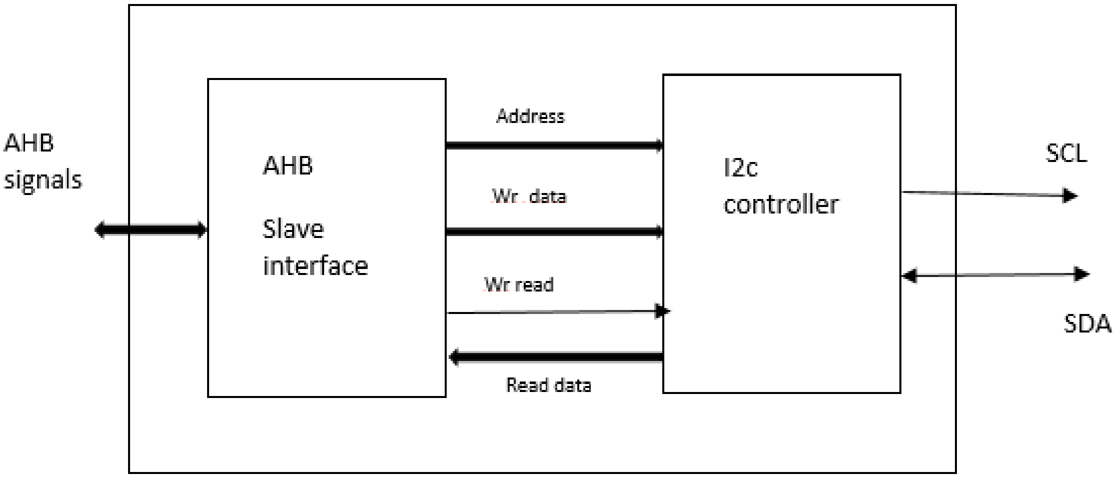
AHB is the high-performance bus in the AMBA family. The protocol defines a 32-bit address bus (HADDR), but this has been extended in some implementations. The read and write data buses (HRDATA and HWDATA) may be defined under the specification as 2n bits wide, from 8-bit to 1024-bit, but the most common implementation has been 32-bit. With additional control signals, and assuming the most common address and data bus width of 32-bit, Up to 16 AHB masters may be connected to a central interconnect which arbitrates between master requests, and multiplexes the winning request and corresponding transfer qualifiers to the slaves. Slave read data is multiplexed back to the masters, it has the following features:

1. single edge clock protocol
2. split transactions
3. several bus masters
4. burst transfers
5. pipelined operations
6. single-cycle bus master handover
7. non-tristate implementation

8. Large bus-widths (64/128 bit).

A simple transaction on the AHB consists of an address phase and a subsequent data phase (without wait states: only two bus-cycles). Access to the target device is controlled through a MUX (non-tristate), thereby admitting bus-access to one bus-master at a time.

IV AHB to I2C Bridge



The proposed design communicate the AHB bus master to I2c device, this bridge receive the data from AHB bus on AHB clock domain and given to I2C interface, I2C controller send the data to I2C slave.

AHB slave inteface module receive the AHB data and it transfer to I2c controller module.

I2C controller module works on two clock domain, it receive the data from AHB slave interface on Hclk, internally it has asynchronous FIFO for synchronize data between AHB clock and I2c Clock.

V. Simulation results

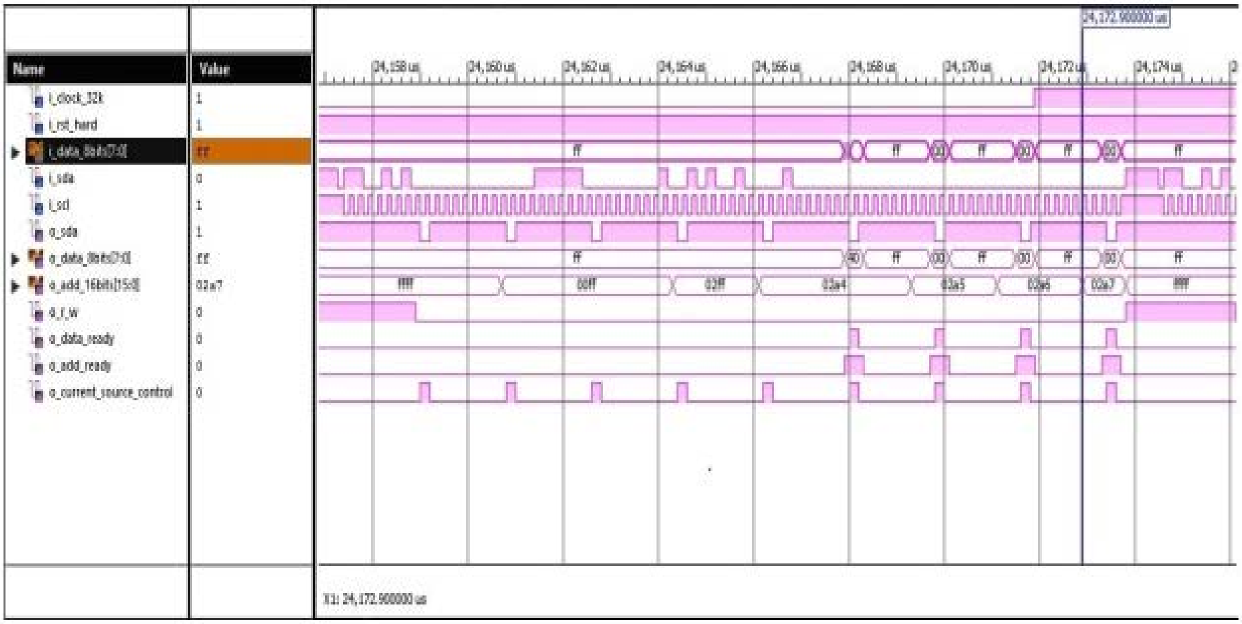


Figure 4: Data transfer format of I2C Bus

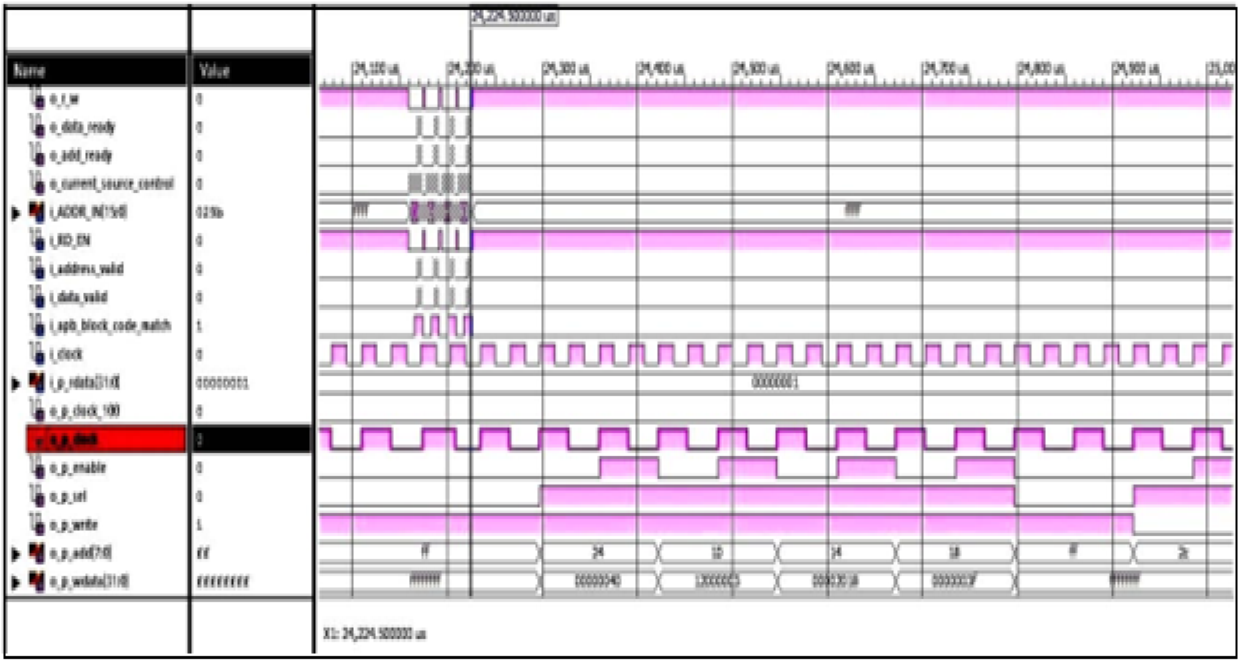


Figure 5: Communication between I2C and designed APB master

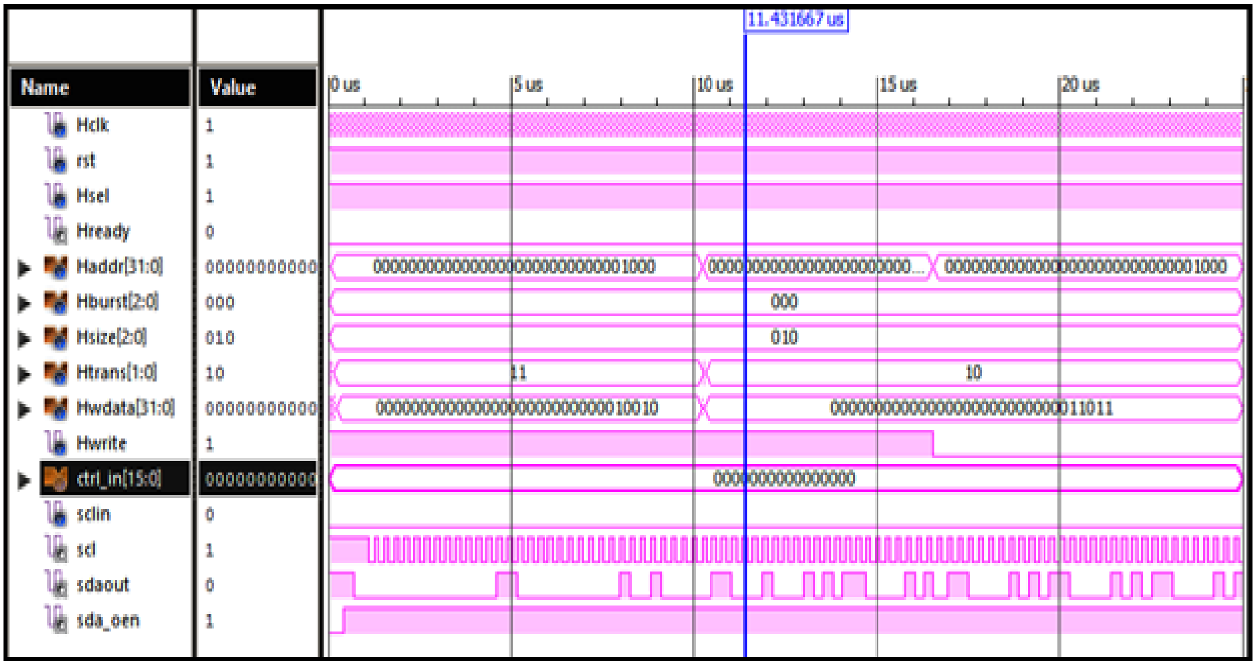


Figure 6: Communication between AHB and I2C

VI.CONCLUSION

I2C Bus was successfully designed according to the standards given by NXP Semiconductors. A working communication model was set up between I2C protocol and APB protocol. Data flow from I2C master to I2C slave to APB master to APB Slave is shown while describing the architecture. APB has low frequency. The proposed **Data** flow from AHB slave to I2C master is described. Because AHB has high frequency. Simulation results are verified and data transfer from I2C master to APB slave, data transfer from AHB slave to I2C master can be clearly seen in provided simulation results. The implemented communication bridges are (I2C and APB), (I2C and AHB) was designed and implemented in Xilinx ISE 13.2, using Verilog HDL.

VII.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. Philips Semiconductors: PCF 8584, I2C bus controller datasheet, http://www.semiconductors.philips.com/acrobat/datasheets/PCF8584_4. Pdf
5. Samir Palnitkar, Verilog HDL, second Edition.
6. UM10204, I2C-bus specification and user manual, Rev. 4 — 13 February 2012, NXP Semiconductors.
7. AMBA® APB Protocol Specification, Version: 2.0, 13 April 2010, issue C, http://www.arm.com.
8. 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.
9. 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.
10. “NXP I2C-bus and SMBus controller PCA9564”, www.nxp.com.
11. Chetan Sharma, Abhisheki Godara, “AHB interface with SPI master by using verilog”, International Journal of Advances in Engineering Research, 2011, Vol. No. 2, Issue No. VI, December.
12. Product brochure, “National Semiconductor I2C Interface”, www.ipextreme.com.
13. “DB-I2C-M-APB-DS-V1.1”, APB Bus I2C Master Controller, Digital Blocks, Inc., September 2013.

Author Details

	Rubina Sultana has completed **B.TECH** in electronics and communication engineering (E.C.E) from City women’s college of engineering and technology Affiliated by **JNTUH** college passed in the year 2014 and completed **M.TECH** in **VLSI System Design** from shadan women’s college of engineering and technology, Affiliated by **JNTUH college**.
	Ms.G.ARUNA as a **professor & HOD, ECE (department**) in shadan women’s college of engineering and technology. She has completed **M.TECH** in Digital System and Computer Electronic (**DSCE**) From **JNTUH and B.TECH** in electronics and communication engineering (**E.C.E**) from Chaitanya Bharathi Instituted of technology (**CBIT**) Affiliated by osmania university (**OU**) college.