Implementation of Advanced Peripheral Bus Based Devices for SOC Application

Roopa .M
Department of E&C
DSCE,
Bangalore, India

Aswatha A.R
Professor & Head
Department of Telecommunication
DSCE,
Bangalore, India

Vani .R.M
Reader & Head, University Science Instrumentation Center, Gulbarga University
Gulbarga, India

***Abstract*— Today's system-on--chip (SOC) is designed with reusable intellectual property cores to meet short time to market requirements. Embedded systems design focuses on low Power dissipation and system-on-chip. A reliable on-chip communication standard is a must in any SOC. The AMBA 2.0 APB is a peripheral bus standard for low bandwidth peripheral. Here, we present the design of APB controller which handles the transactions between the master and peripheral devices. The final design which integrates the peripheral devices with the APB controller is implemented on the FPGA device.**

***Keywords*- System –on- chip (SOC), Advanced Peripheral Bus (APB), advanced microcontroller bus architecture (AMBA)**

I INTRODUCTION

The increasing complexity of Systems-on-Chip (SOC) has led to the critical "design productivity gap" problem. On-chip communication architectures have a significant impact on system performance, power dissipation and time-to-market. System designers, as well as the research community have focused on the issue of exploring, evaluating, and designing SOC communication architectures to meet the targeted design goals [1].

II ADVANCED PHERIPHERL BUS

The APB is part of the AMBA hierarchy of buses and is optimized for minimal power consumption and reduced interface complexity. The AMBA APB should be used to interface to any peripherals which are low bandwidth and do not require the high performance of a pipelined bus interface [2].

A. APB Specifications

The APB specification is described under three Sections as shown in headings State diagram, Write transfer and Read transfer.

- **State Diagram:**

The state diagram, shown in Figure 2.1, can be Used to represent the activity of the peripheral bus.

Operation of the state machine is through the three states described below:

Idle: The default state for the peripheral bus.

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 will always move To the ENABLE state on the next rising edge of the clock.

Enable: In the ENABLE state the enable signal, **PENABLE** is asserted. The address, write and select signals all remain stable during the transition from the SETUP to ENABLE state. The ENABLE state also only lasts for a single clock cycle and after this state the bus will return to the IDLE state if no further transfers are required. Alternatively, if another transfer is to follow then the bus will move directly to the SETUP state. It is acceptable for the address, write and select signals to glitch during a transition from the ENABLE to SETUP states

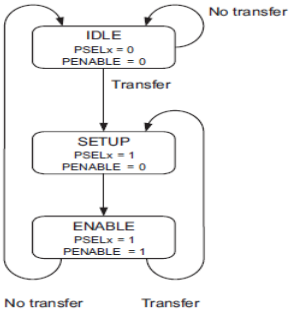

Figure 2.1: APB Controller State Diagram [2]

- **Write Transfer:**

The basic write transfer of an APB is shown in Figure 2.2.

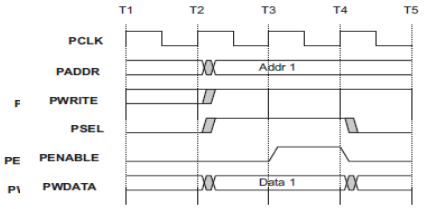

Figure 2.2: Write Cycle of APB Controller [2].

The write transfer starts with the address, write Data, write signal and select signal all changing after the rising edge of the clock. The first clock cycle of the transfer is called the SETUP cycle. After the following clock edge the enable signal **PENABLE** is asserted and this indicates that the ENABLE cycle is taking place. The address, data and control signals all remain valid throughout the ENABLE cycle. The transfer completes at the end of this cycle [3].

Identify applicable sponsor/s here. *(sponsors)*

- **Read Transfer:**

Figure 2.3 shows a read transfer depicting the clock and control signal.

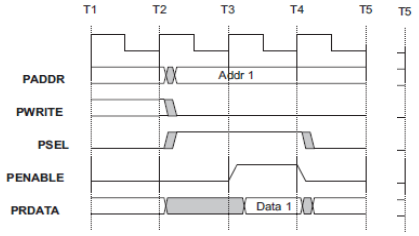

Figure 2.3: Read cycle of APB Controller [2].

The timing of the address, write, select and strobe signals are all the same as for the write transfer. In the case of a read, the slave must provide the data during the ENABLE cycle. The data is sampled on the rising edge of clock at the end of the ENABLE cycle.

III ARCHITECTURE OVERVIEW

The architecture is developed based on the standard specifications [2] of the protocol. The architecture explains the function modules designed.

A. Design Module

The design module is a result of integration of low Bandwidth peripheral devices with the APB controller. The operation of the slave devices are initiated and controlled by APB controller. The Top module is presented first and the sub modules are discussed in further sections.

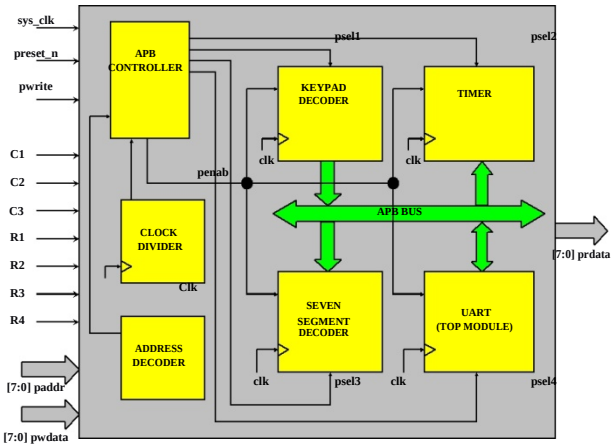

Figure 3.1: APB Controller Design Block Diagram

The block diagram is depicted in figure 3.1. Major control signals between the APB controller and the other modules are also indicated in the figure. APB Controller is the unit responsible for the flow of data and control signals between the Slaves and the Master. The different peripherals modules are UART, Timer, Seven Segment Decoder and Keypad Decoder. Input for the APB controller is available from the AHB-to- APB Bridge which acts as the master for the APB. The APB switches data between the slaves and the master using the control signals.

The Hierarchy of designs is as shown below:

➔ T O P **MODULE** – APB controller integrated with peripherals (Slaves)

- APB Controller.
- UART Top Module UART
  - Transmitter UART
  - Receiver
  - Baud Rate Generator
- Seven Segment Decoder
- Timer
- Keypad Decoder

B. Top Module – APB Integrated with Peripheral Devices

Block diagram presented here does not indicate the Internal units of the design. The detailed discussion of the architectures of the different modules is done in future sections.

Signal List:

Table I: Signal list of top module design block

I/O NAMES	I/O DESCRIPTION	I/O DESCRIPTION
sys_clk	input	System clock.
preset_n	input	System reset signal.
pwrite	input	Read or Write signal.
C1	input	Column 1 input.
C2	input	Column 2 input.
C3	input	Column 3 input.
R1	input	Row 1 input.
R2	input	Row 2 input.
R3	input	Row 3 input.
R4	input	Row 4 input.
paddr [7:0]	input	8 bit address bus.
pwdata [7:0]	input	8 bit Write Data Bus.
prdata [7:0]	output	8 bit Read Data Bus.

Description:

Figure 3.1 depicts the block diagram of APB Top Module. The signals responsible for functioning of the module are listed in table I. Top Module integrates all the peripheral devices with the APB Controller. The “sys_clk” is the system clock signal and the “preset_n” is the system reset signal for the unit. The “paddr” is 8 bit address available from the master i.e. AHB-to-APB Bridge. The “pwdata” signal is the 8 bit write data bus.The “pwrite” is the Read/Write signal. If pwrite = 1 it is a write operation else read operation. C1, C2, C3 and R1, R2, R3, R4 are Column and Row inputs from the 4X3 keypad matrix. The “prdata” is the 8 bit read bus.

When an address is placed, the APB controller selects the slave device which needs a transaction. The operation is enabled by the controller and read or writes transaction takes place.

C. APB Controller

The APB controller acts as the intermediate unit between master i.e. AHB-to-APB Bridge, and the APB slaves. The slave devices are selected using the select lines from the APB controller and its operation is enabled using the enable signal. The block diagram of the APB controller is as shown in figure 3.2

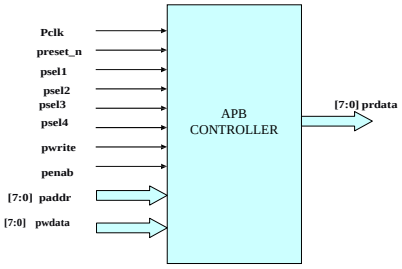

Figure 3.2: Block Diagram of APB Controller.

Description:

APB Controller controls the flow of data and control signals between the master and the slave. It works according to state machine as shown in the figure 2.1.

The Controller works with clock "pclk" and has "preset_n" as the system reset signal. The data read slave places its data on the "prdata" bus and data is written to slave device through "pwdata" bus. The state machine has 3 states IDLE, SETUP and ENABLE. Initially the controller stays in the IDLE state. If any transfer is then the controller moves to the SETUP state. The SETUP state lasts for only one clock cycle and the controller moves to the ENABLE state. The operation of the peripheral device is enabled in the ENABLE state by issuing the enable signal in next signal.

D. UART Top Module

The UART is a serial transmission and reception Unit where the transmitter and receiver works asynchronously i.e. With different clock signals. The block diagram of the UART top module is depicted in the figure 3.3.

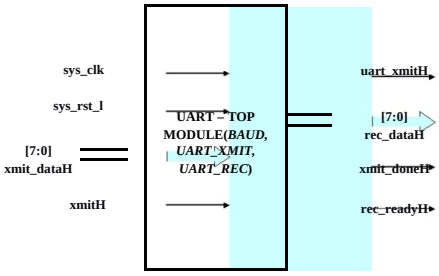

Figure 3.3: Block Diagram of UART Top Module**.**

Description:

The Top module UART design includes three subs-Modules i.e. UART Transmitter, UART Receiver and Baud Rate Generator. The top module takes "sys_clk" as input and the Transmitter and Receiver works with the "uart_clk" which is generated by the baud rate generator. The UART Transmitter transmits the loaded data (8 Bit) serially through "uart_xmitH" line when the"xmitH" signal is asserted. After the transmission is complete, "xmit_doneH" signal is issued.The UART Top module is configured in such a way that the transmitter line of UART Transmitter is connected to the receiver line of the UART Receiver module. Input data at the Receiver is synchronized and sampled by the internal units. The serial data is converted to parallel and stored in a register"rec_dataH". After the reception operation is complete the "rec_readyH" signal goes high. The individual blocks and state diagrams of the transmitter and receiver are in further sections.

- **UART Transmitter**

The UART transmitter as an individual unit starts its operation when xmitH signal goes high. The transmitter module has internal units such as Parallel to Serial converter, counters, start, data and stop bit selection mux and a control unit (state machine). The Parallel to serial converter or Serializer unit shifts the loaded 8 bit data one bit at a time and transmits for a particular baud rate. The counters keep track of the number of bits of data that are transmitted. The state machine controls the data flow and the operation among the different units in the transmitter. The state machine issues the control signals among the units to achieve a proper transmission.

- **Baud Rate Generator**

The baud rate generator is the important unit of any UART module. The receiver and transmitter works with the uart clock which is provided by this unit. The baud rate is fed to this unit for which the receiver and transmitter works. The baud rate generator internally consists of a counter which is loaded with the baud rate data. The uart clock is the output of the unit which switches its value for every reset of the counter. This results in a clock signal which is fed to transmitter and the receiver.

- **UART Receiver**

The UART receiver design involves lot of complexity. It has a dual rank synchronizer, bit cell counter, received bit counter, de-serialize (serial to parallel converter) and a state machine as its units. The state machine is the control unit which controls all the operations in the receiver unit by issuing the control signals. The receiver takes uart clock as the reference clock signal. The data line is the input for this unit and output in the parallel register which stores the received data.

E. Seven Segment Decoder

The Seven Segment Decoder is a simple decoder module which takes ASCII code for corresponding number and characters (# and *) in hexadecimal form and gives the seven segment code for corresponding number and character.

F. Timer

The timer module presented is a countdown timer This gives a time out signal after a particular amount of time. The time duration at which the time out signal is to be issued is loaded in the form of input data to the timer. This data is loaded onto the counter register and the countdown operation starts on the control signal.

The Timer unit is an 8 Bit countdown timer. The Timer module consists of a data path and a control unit. The datapath unit is the one where the data are processed and the control unit controls the data process by using the control signals. The data path unit contains of two counters: "fast_counter" and "counter". The "counter" determines the time out interval. The initial data (8 bit) that has to be loaded to the timer is fed through "SW" input. On "start" signal the countdown operation starts. The fast_counter is a continuous counter. The loaded data will be moved to the "counter" register. For one complete count of "fast_counter" i.e.256 counts, the "counter" will be decremented by one. This count continues till the "counter" register will become zero. After the "counter" becomes zero a "time_out" signal is issues which is the output of the timer unit.

G. Keypad Decoder

The Keypad Decoder is an input device which reads values or data from the user. The user or programmer can enter the input through a keypad matrix (4X3 matrix is used in the present design). The Keypad Decoder unit takes combination the Rows and Columns i.e. R1, R2, R3, R4, C1, C2, and C3 as input from the keypad 4X3 matrix and gives the ASCII code (8 -bit) for corresponding number or character (# and *) in hexadecimal form.

IV. IMPLEMENTATION OF APB MODEL

The Implementation of the design is achieved by dividing the design module into two sub-modules:

- ➔ Module 1: FPGA-1 Implementation
- ➔ Module 2: FPGA-2 Implementation

The top module constitutes of more number of inputs and outputs. The design which is to be implemented on the SPARTAN 3 board becomes complicated while configuring the pins for the FPGA. So the modules of the design are split into two so that the design is implemented using two SPARTAN 3 boards. During this process, care is taken that the functionality is maintained as with the design module.

A MODULE 1: FPGA-1 Implementation

The first module consists of two peripheral units: Keypad Decoder and Seven Segment Decoder. The block diagram of the FPGA-1 Implementation module is shown in the figure 4.1. The inputs, outputs and signal used in the module are listed in the table II

Block Diagram:

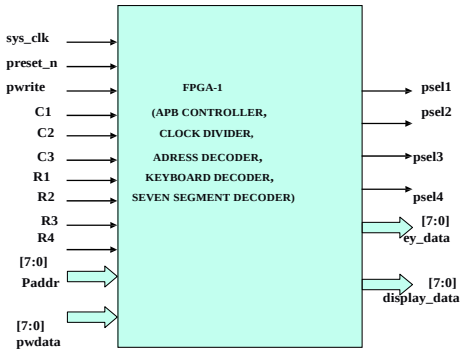

Figure 4.1: FPGA-1 Implementation block diagram

Signal List:

Table II: FPGA-1 Implementation module signal list**.**

I/O NAME	I/O DIRECTION	DESCRIPTION
sys_clk	input	System clock
preset_n	input	
C1	input	Column 1 input.
C2	input	Column 2 input.
C3	input	Column 3 input.
R1	input	Row 1 input.
R2	input	Row 2 input.
R3	input	Row 3 input.
R4	input	Row 4 input.
paddr[7:0]	input	8 bit Address bus
pwdata[7:0]	input	8 bit Write Data Bus.
psel1	output	Select line for Keypad Decoder.
psel2	output	Select line for Timer.
psel3	output	Select line for Seven Segment Decoder.
psel4	output	Select line for UART.
key_data[7:0]	output	8 bit Keypad data (ASCII code in Hex form).
disp_data[7:0]	output	8 bit display data (Seven Segment code).

Description:

The FPGA-1 Implementation block includes the following modules:

- APB Controller.
- Clock Divider.
- AddressDecoder.
- Keypad Decoder.
- Seven Segment Decoder.

The FPGA-1 Implementation block works with the Clock of the SPARTAN 3 board i.e. 50MHz. The Clock Divider module gives two different clock signals which are used by the internal peripheral modules. The APB controller works with the "pclk" clock signal provided by the clock divider module. The Address Decoder module decodes the address and gives the select signals i.e. pselx, which is used

by the APB controller to select the slaves for read or write operation APB controller issues control signals to enable the peripheral operation.

The Keypad Decoder takes row and column inputs and gives ASCII codes for respective combination of rows and columns in HEX form. This data is displayed using the LED's on the SPARTAN 3 board.

The Seven Segment Decoder is a write device. The Seven Segment Decoder takes ASCII codes in HEX form of different numbers from 0 to 9 and two characters i.e. * and #, and gives the seven segment code for respective number or character. This output data is displayed on the seven segment display on the SPARTAN 3 board when Seven Segment Decoder slave is selected.

The select signals for the other two slave devices i.e. UART and Timer are output to the FPGA-2 implementation block where the two slave devices are implemented.

B MODULE 2: FPGA-2 Implementation

The FPGA-2 implementation includes the Timer and UART designs, shown in figure 4.2. These modules are controlled by the select signals from the FPGA-1 module.

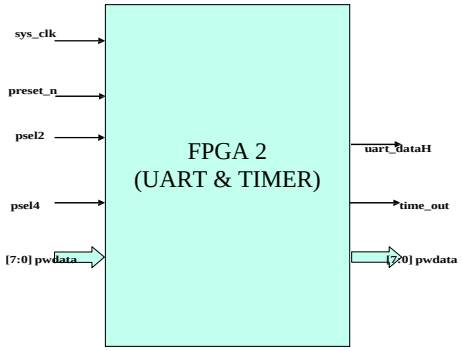

Figure 4.2 FPGA-2 Implementation block diagram

Signal List:Table III:Pin configuration of FPGA-2 Implementation Block.

I/O NAME	I/O DIRECTION	DESCRIPTION
sys_clk	input	System clock
preset_n	input	System reset
psel2	input	Slave select signal for Timer unit.
psel4	input	Slave select signal for UART unit.
pwdata[7:0]	input	8 bit write data bus.
uart_dataH	output	UART data transmission line.
time_out	output	Time out signal from timer unit.
rec_dataH[7:0]	output	8 bit Receive data register of UART unit.

Description:

The FPGA-2 Implementation block constitutes of Two peripheral modules:

- Timer Module.
- UART Top Module.

The modules in this block works with the clock of the SPARTAN 3 board i.e. 50MHz. The inputs, outputs and signals for this block are described in the table III. This implementation module takes control inputs from the FPGA-1 implementation module.

The Timer unit in this block is a countdown timer which decrements the loaded value in the counter register and gives a time out signal. This time out signal can be used as an interrupt signal for other devices in a SoC environment.

The UART unit is the top module consists UART Transmitter, UART Receiver and Baud Rate Generator. The UART Transmitter and UART Receiver works with the clock signal provided by the Baud Rate Generator.

V. SIMULATION RESULTS

The simulation results of the APB design model and the APB implementation model is presented.

A Design Module

- **Top Module:**

The figure 5.1 below shows the simulation result of the Top Module which integrates all the design with the APB controller.

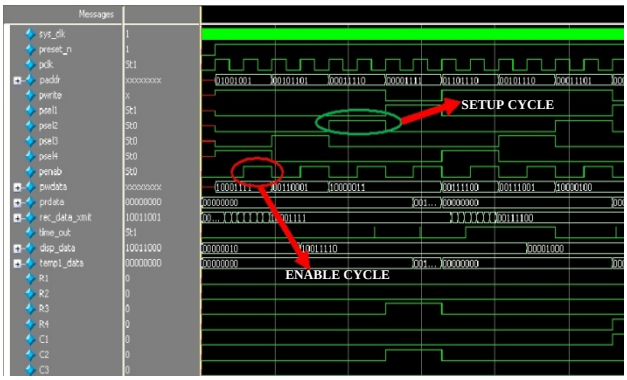

FIGURE 5.1: SIMULATION RESULT OF TOP MODULE

- **Memory Contents:**

The simulation result in figure 5.2 shows the contents of the Memory where the data is read and written onto the respective memory location.

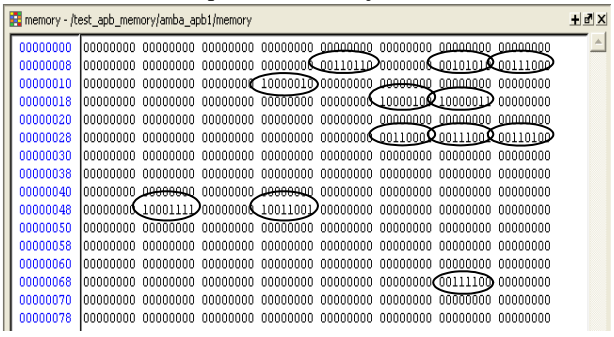

FIGURE 5.2: MEMORY CONTENTS OF THE DESIGN.

- **Seven Segment Decoder Module:**

The figure 5.3 shows the simulation result of seven Segment Decoder module

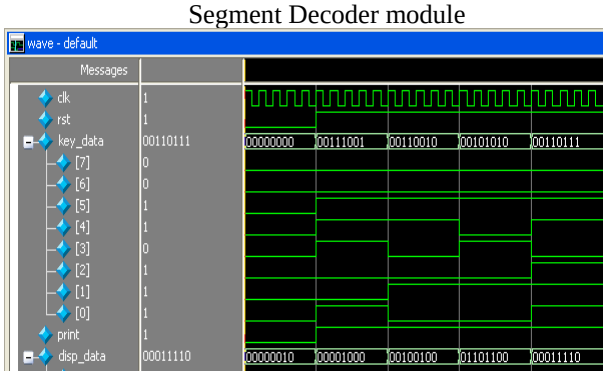

Figure 5.3: Simulation result of Seven Segment Decoder module

- **Keypad Decoder Module**

The figure 5.4 shows the simulation relation of the Keypad Decoder module. The acknowledge signal and the send signal are indicated in the waveform.

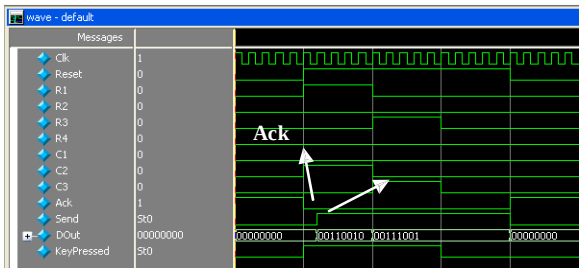

Figure 5.4: Simulation result of Keypad Decoder module

B Implementation Module:

This section presents the simulation result of the FPGA-1 and FPGA-2 implementation blocks.

- **Simulation results of FPGA-1 Implementation Block**

The figure 5.5 below shows the simulation results of the FPGA-1 Implementation block.

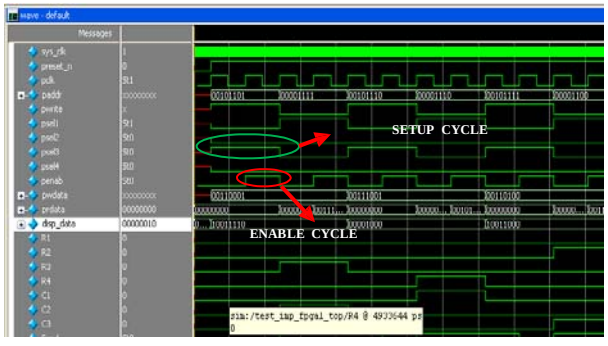

Figure 5.5: Simulation Result of FPGA1 Implementation

- **Simulation results of FPGA-2 Implementation Block**

The figure 5.6 below shows the FPGA-2 Implementation Block which includes the UART and Timer units. The control signal and the select lines are available from the FPGA-1 implementation block

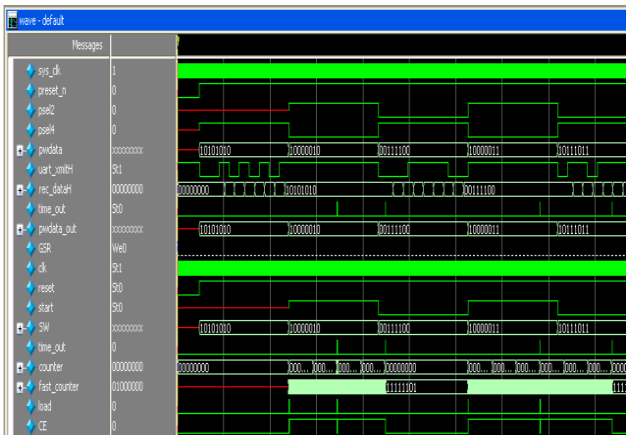

Figure 5.6: Simulation Result of FPGA2 Implementation block

CONCLUSION

AMBA APB Protocol has been implemented with low Bandwidth Peripherals on FPGA. The Read and Write Operation are performed on peripherals and controlled by APB Controller. The read or write transaction is completed in two bus cycles when a particular slave is selected by select line and the operation is enabled at the latch of enable signal.

REFERENCES

[1] Nikil Dutt, Kaustav Banerjee, Luca Benini, Kanishka Lahiri, Sudeep Pasricha,"Tutorial 5: SoC Communication Architectures: Technology, Current Practice, Research, and Trends," vlsid, pp.8, 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07), 2007

[2] AMBA Specification (Rev 2.0).

[3] Flynn,D. Adv.RISC Machines Ltd., Cambridge, "AMBA:enabling reusableon-chip designs", IEEE Micro, Publication Date: Jul/Aug 1997.

[4] "Using formal techniques to Debug the AMBA System-on-Chip Bus Protocol" Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2003.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] Xilinx ISE Synthesis and Verification Design Guide.

AUTHORS PROFILE

Roopa. M. received B.E, Degree from U.V.C.E Bangalore University, Bangalore In 1990, M.E, Degree from U.V.C.E, Bangalore University, Bangalore, India in 1994, pursuing Ph.D in the Department of Applied Electronics, Gulbarga University, Gulbarga, INDIA, currently she is working as Assistant Professor in the Department of Electronics and communication, Dayananda sagar college of engineering, Bangalore, INDIA, her research interests include Design of Digital circuits, VLSI Design and Embedded systems. Email: surajroopa@gmail.com.

Dr.A.R.Aswath, received B.E Degree from Mysore University in 1991, M.Tech Degree from M.I.T Manipal in 1996, M.S. Degree from B.I.T.S. Pilani in 2002, Ph.D degree in 2010 from Dr. M.G.R University. Currently he is working as Professor & Head in Telecommunication Department, Dayanand Sagar College of Engineering, Bangalore, India. His main research Interests include Analysis and design of Low Power VLSI Circuits, Embedded System Design and Image Processing. Email: aswath.ar@gmail.com.

Dr.VANI. R.M. received her B.E. in Electrical and Electronics from the B.I.ET., Davanagere, Karnataka, and M.Tech in Industrial Electronics from S.J.C.E., Mysore. She has received her Ph.D in Applied Electronics from Gulbarga University, Gulbarga, India, in year 2005. She is working as Reader & Head, University Science Instrumentation Center, Gulbarga, since 1995.She has more than 85 research publications in national and international reputed journals and Conference proceedings. She presented many research papers in India & abroad. She has conducted several courses, workshops for the benefit of faculties and field engineers. Her areas of interest are microwave antennas, PC based Instrumentation, Embedded controllers and wireless communication. She has one UGC major research project to her credit. Email:vanirm12@rediffmail.com