FSM & Handshaking Based AHB to APB Bridge for High Speed Systems

Prof. Ravi Mohan Sairam[1] Prof. Sumit Sharma[2] Miss. Geeta Pal[3]

[1]Head of the Department (M.Tech)
Shri Ram Institute of Technology, Jabalpur 482002 (M.P) India

[2]Head of the Department (Electronics & Communication)
Shri Ram Institute of Technology, Jabalpur 482002 (M.P) India

[3]Master of Engineering IV Semester (VLSI)
Shri Ram Institute of Technology, Jabalpur 482002 (M.P) India

Abstract

Microprocessor performance has improved rapidly these years. In contrast memory latencies and bandwidths have improved little by using advanced microcontroller bus architecture with its advanced high performance bus. The Advanced Microcontroller Bus Architecture (AMBA) is a widely used interconnection standard for System on Chip (SoC) design. In order to support high-speed pipelined data transfers, AMBA supports a rich set of bus signals, making the analysis of AMBA-based embedded systems a challenging proposition. The goal of this work is to synthesize and simulate complex interface bridge between Advanced High performance Bus (AHB) and Advanced Peripheral Bus (APB) known as AHB2APB Bridge. To achieve high performance proposed architecture is FSM based pipelined APB-to-AHP Bridge and Vice-versa. This also involves the Back notation for Synthesized of Bridge module and to perform Functional and Timing Simulation using Xilinx ISE.

1 Introduction

Integrated circuits have entered the era of System-on-a-Chip (SoC), which refers to integrating all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions – all on a single chip substrate. With the increasing design size, IP is an inevitable choice for SoC design. And the widespread use of all kinds of IPs has changed the nature of the design flow, making On-Chip Buses (OCB) essential to the design. Of all OCBs existing in the market, the AMBA bus system is widely used as the de facto standard SoC bus. ARM announced availability of the AMBA 4.0 specifications. As the de facto standard SoC bus, AMBA bus is widely used in the high-performance SoC designs. The AMBA specification defines an on-chip
Communication standard for designing high-performance embedded microcontrollers. The AMBA 4.0 specification defines five buses/interfaces.

- Advanced extensible Interface (AXI)
- Advanced High-performance Bus (AHB)
- Advanced System Bus (ASB)
- Advanced Peripheral Bus (APB)
- Advanced Trace Bus (ATB)

AXI, the next generation of AMBA interface defined in the AMBA 4.0 specification, is targeted at high performance; high clock frequency system designs and includes features which make it very suitable for high speed sub-micrometer interconnection.

Advanced high-performance bus (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. AMBA AHB [6] implements the features required for high-performance, high clock frequency systems including:

1. High performance
2. Pipelined operation
3. Multiple bus masters
4. Burst transfers
5. Single-cycle bus master handover
6. Non-tri state implementation
7. Wider data bus configurations (64/128bits).

Bridging between this higher level of bus and the current ASB/APB can be done efficiently to ensure that any existing designs can be easily integrated. An AMBA AHB design may contain one or more bus masters typically a system would contain at least the processor and test interface. However, it would also be common for a Direct Memory Access (DMA) or Digital Signal Processor (DSP) to be included as bus masters. The external memory interface, APB Bridge and any internal memory are the most common AHB slaves. Any other peripheral in the system could also be included as an AHB slave. However, low-bandwidth peripherals typically reside on the APB.

B. Advanced System Bus (ASB)

The AMBA ASB is for high-performance modules. It is an alternative system bus suitable for use where high-performances features of AHB are not required.ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions.

1. Features of ASB:
2. Burst transfers
3. Pipelined transfer operation
4. Multiple bus masters.
5. Advanced peripheral bus (APB)

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) hierarchy [7] 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. The latest revision of the APB ensures that all signal transitions are only related to the rising edge of the clock.

This improvement means the APB peripherals can be integrated easily into any design flow.

Features of APB:

1. Low power
2. Latched address and control
3. Simple interface
4. Suitable for many peripherals

These changes to the APB also make it simpler to interface it to the new Advanced High-performance Bus (AHB).

IV. OPERATION OF AHB2APB BRIDGE

The AHB2APB interfaces AHB and APB. It buffers address, controls and data from the AHB, drives the APB peripherals and return data along with response signal to the AHB. The AHB2APB interface is designed to operate when AHB and APB clocks have the any combination of frequency and phase [8].TheAHB2APB performs transfer of data from AHB to APB for write cycle and APB to AHB for Read cycle

A. Features of AHB2APB Bridge

Interface between AMBA high performance bus (AHB) and AMBA peripheral bus (APB) [2], provides latching of address, controls and data signals for APB peripherals. Supports for the following

APB compliant slaves and peripherals.

Peripherals which require additional wait states.

Figure 2. Pin details of AHB2APB Bridge

B. AHB Response

The sub-module AHB Response sequences the way that the AHB2APB responds to AHB requests. Valid commands are forwarded to control transfer for action. Invalid commands are not forwarded and an error message is generated. It operates on AHB CLOCK and RESET. The control Transfers block in Fig. 3 transfers AHB control signal to the APB access with appropriate delays inserted to map the pipelined AHB protocol to the two cycle APB protocol. It ensures that only one request is presented to the APB access while it is processing a request. It operates on AHB CLOCK and RESET.

APB Bus

The APB access generates the control signals on the APB for read and writes cycles. It operates on APB CLOCK and RESET. The APB Bridge is the only bus master on the AMBA APB. In addition, the APB Bridge is also a slave on the higher-level system bus. The bridge unit converts system bus transfers into APB transfers and performs the following functions:

Latches the address and holds it valid throughout the transfer.

Decodes the address and generates a peripheral select (PSEL). Only one select signal can be active during a transfer.

Drives the data onto the APB for a write transfer.

Drives the APB data onto the system bus for a read transfer.

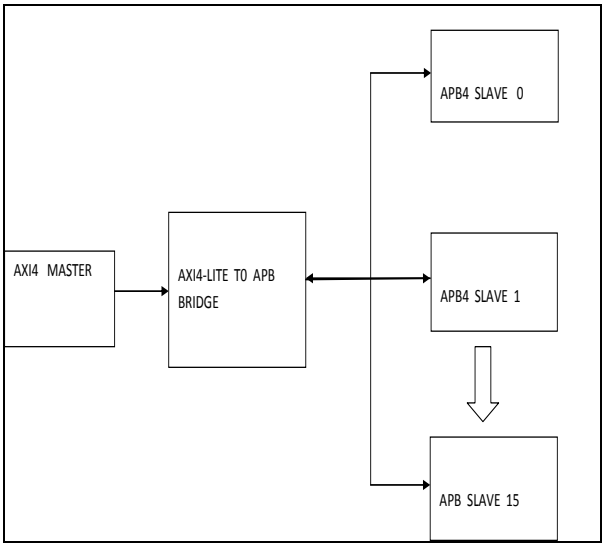
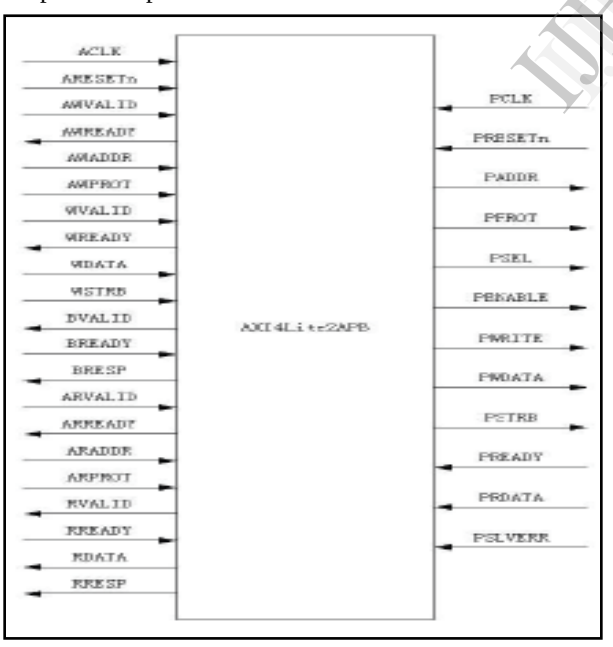


Figure1-Block Diagram

1.1 SIGNAL CONNECTION

The bridge uses:

- AMBA AXI-Lite signals as described in the AMBA.
- AXI-Lite 4.0 protocol specification.
- AMBA APB signals as described in the AMBA APB.
- 4.0 protocol specification.



Generates a timing strobe, PENABLE, for the transfer.

Figure 2 - Signal Connection

1.2 AXI HANDSHAKE MECHANISM

In AXI 4.0 specification, each channel hREADY signals for handshaking. The sour when the control information or data destination asserts READY when it can a information or data. Transfer occurs only VALID and READY is asserted. Cases of VALID/READ handshaking. Note asserts VALID, the corresponding control in must also be available at the same time. Indicate when the transfer occurs. A transfer the positive edge of clock. Therefore, the register input to sample the READY signal.

The APB bridge buffer addres,,control and data from AXI4-lite ,and drives the APB peripherals and returns data and response signal to the AXI4-lite.it decodes the address using an internal address map to select the peripherals.the bridge is designed to oprerate when the APB and AXI4-lite have independent clock frequency and phase.for every AXI channel, invalid commonds are not forwarded and an error response generated.thay is once and peripheral acess does not exists ,the APB bridge will generate DECERR as response through the response channel.(read or write).and if the target peripheral exists,but asserts PSLVERR, it will give a SLVERR response.

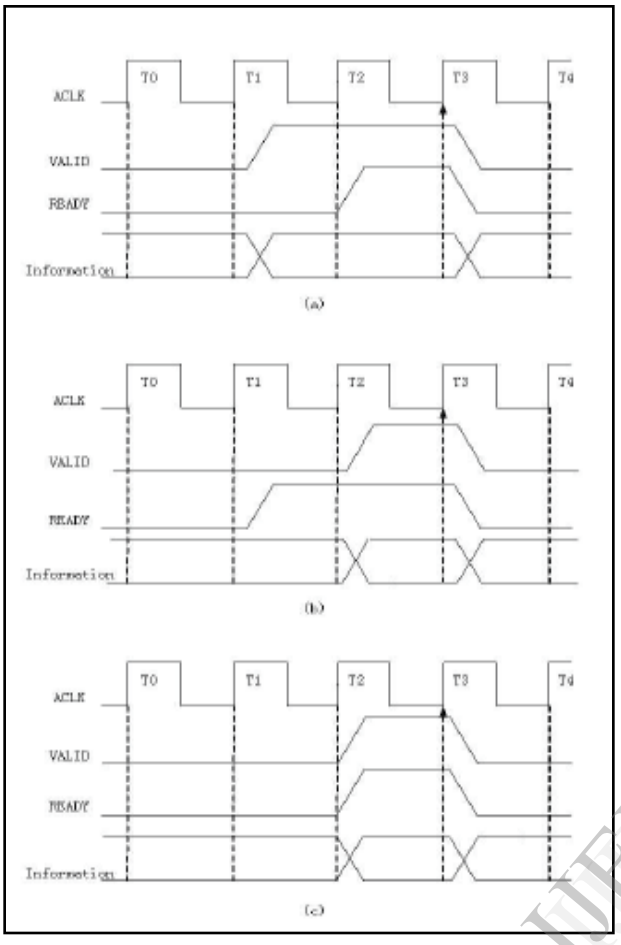


Figure 3-Handshake Mechanism

2 .CLOCK DOMAIN CROSSING

A clock domain crossing (Cis) when a signal crosses from one clock to another. If a signal does not assert long may appear asynchronous on Metastability happens when setup/hold time window. Sync into a higher clocked domain registering the signal through a source domain, thus holding detected by the higher Synchronizing a signal travers in more cumbersome. This typical clock domain with a form of domain to the source domain, detected.

2.1 METASTABILITY

Metastability cannot be another metastable signal enabled sign. The metastability ousting the mean time between mtbf Where C1 and C2 are constant used to build the flip-flop metastable output, and clock and synchronous clock and the asynchronous respectively.

$$\text{MTBF} = \frac{e^{c2 + \text{thBT}}}{c1 \times \text{fclk} \times \text{fdata}}$$

2.2SYNCRONISER

Designers can use special metastable hardened flops for increasing the MTBF. Synchronizer flop is used following the signal DB. So, instead of the metastable signal DB being used in the function downstream. the stable signal DB2 is used in the logic downstream[8]. In the AXI4-Lite to APB bridge, we use synchronizer block designs for communicate between the AXI and APB clock domain.

3. FINITE STATE MACHINE

A finite state machine is a mathematical abstraction sometimes used to design digital logic or computer programs.It is a behavior model composed of a finite number of states, transitions between those states, and actions, similar to a flow graph in which one can inspect the way logic runs when certain conditions are met. The state transition diagram is a picture of our state machine model. Figure. 5 is the state transition diagram of our FSM.

The state machine operates through the following states:

- IDLE. This is the default state of the FSM. SETUP. When a write transfer request is asserted, the FSM moves into the SETUP state.
- SETUP. When a read transfer request is asserted, the FSM moves into the SETUP state.
- ENABLE. The enable signal, PENABLE, is asserted in the ENABLE state. READ_ACCESS. The enable signal, PENABLE, is asserted in the ENABLE state.
- HRESP. When the AXI read data channel is not ready for receiving signal RRESP, then stay in HRESP state. States HRESP and ENABLE are added, because the APB is not pipelined, wait states are added during transfers between the APB and AXI interface states HRESP and ENABLE are added, because the APB is not pipelined, wait states are added during transfers between the APB and AXI interface. System Verilog source code sample of the enumerated type encoded FSM is given below.

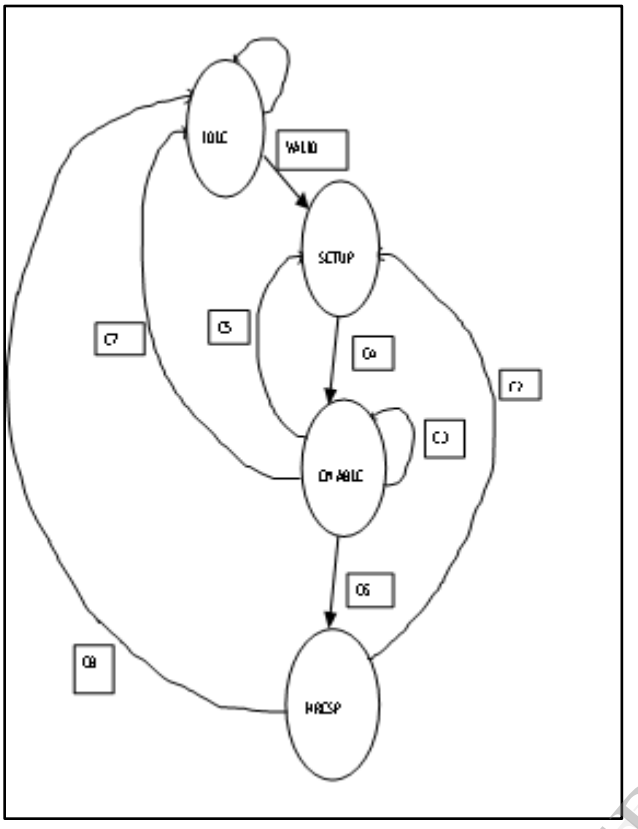


Figure 5-FSM State Diagram

According AXI specification, the read the read address channel, write address channel and write data channel are completely independent. Each channel has a set of forward signals and a feedback signal for handshaking. A read and a write requests may be issued simultaneously AWVALID/WVALID and ARVALID are asserted high simultaneously) from AXI4-Lite, the AXI4-Lite to APB bridge will give more priority to the read request than to the write request. That is, when both write and read requests are valid, the write request is initiated on APB after the read is requested on APB

4 SIMULATIONS AND IMPLEMENTATION

The timing diagram shown in Figure below. illustrates theAXI4-Lite to APB bridge operation for various read and write transfers It shows that when both read and write requests are active, read is given more priority.

Before Static Timing Analysis (STA), it is necessary to inform the EDA tools that ACLK and PCLK are two asynchronous clock domains:

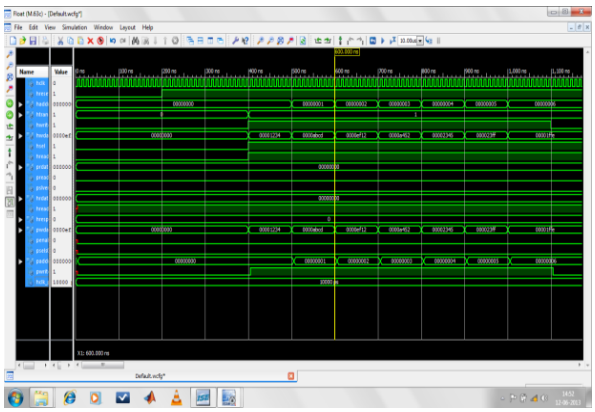
Figure 6 Typical Read and Write Transfer

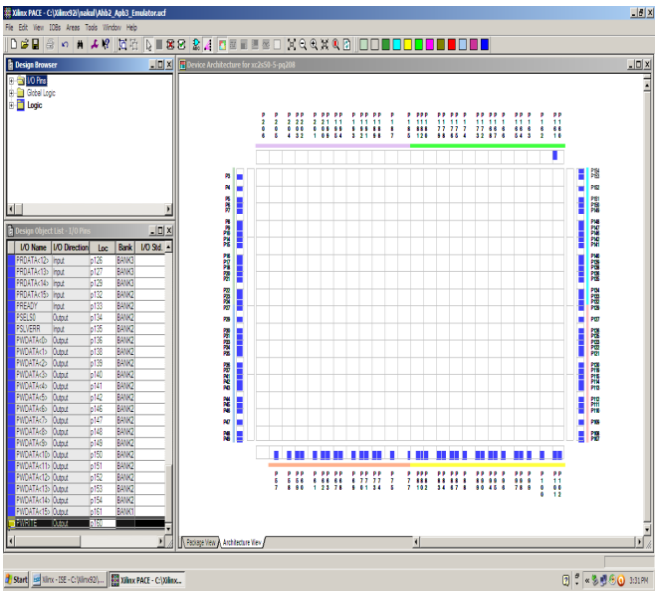
Area Report

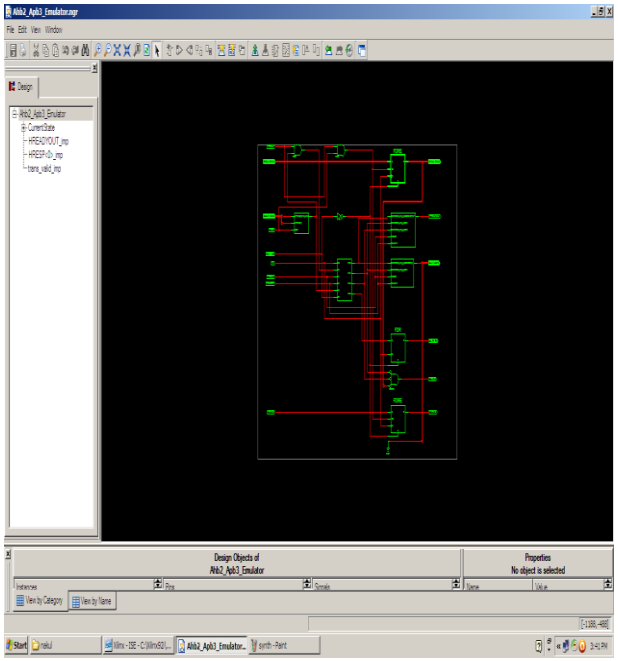
Number of Slices	5	out of	768	0%
Number of Slice Flip Flops	3	out of	1536	0%
Number of 4 input LUTs	10	out of	1536	0%
Number of IOs	111			
Number of bonded IOBs	110	out of	140	78%
IOB Flip Flops	18			
Number of GCLKs	1	out of	4	25%

Timing report

Minimum period	5.928ns (Maximum Frequency: 168.691MHz)
Minimum input arrival time before clock	6.943ns
Maximum output required time after clock	10.702ns
Maximum combinational path delay	10.134ns







5. CONCLUSION

In this study, we provide an implementation of AXI4-Lite to APB bridge which has the following features:

1. 32-bit AXI slave and APB master interfaces.
2. PCLK clock domain completely independent of ACLK clock domain.
3. Support up to 16 APB peripherals.
4. Support the PREADY signal which translates to wait states on AXI.
5. An error on any transfer results in SLVERR as the AXI read/write response.

6. REFERENCES

[1] ARM,"AMBA Protocol Specification 4.0", www.arm.com, 2010.

[2]Ying-Ze Liao, "System Design and Implementation of AXI Bus", National Chiao Tung University, October 2007.

[3] Clifford E. Cummings, "Coding And Scripting Techniques For FSM Designs With Synthesis-Optimized, Glitch-Free Outputs," SNUG (Synopsys Users Group Boston, MA 2000) Proceedings, September 2000.

[4] Clifford E. Cummings, “Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs,” SNUG 2001

[5] Chris Spear, "SystemVerilog for Verification, 2nd Edition"Springer,www.springeronline.com, 2008.

[6] Lahir, K., Raghunathan A., Lakshminarayana G., “LOTTERYBUS: anew high-performance communication architecture for system-on-chip deisgns,” in Proceedings of Design Automation Conference, 2001.

[7] Sanghun Lee, Chanho Lee, Hyuk-Jae Lee, “A new multi-channel onchip-bus architecture for system-on-chips,” in Proceedings of IEEE international SOC Conference, September 2004.

[8] Martino Ruggiero, Rederico Angiolini, Francesco Poletti, Davide Bertozzi, Luca 86

[9] Benini, Roberto Zafalon, “Scalability Analysis of Evolving SoCInterconnect Protocols,” Int. Symposium on System-on-Chip, 2004. Lukai Cai, Daniel Gajski, “Transaction level modeling: an overview,” in Proceedings of the 1st IEEE/ACM/IFIP international conference on
Hardware/software codesign and system synthesis, October 2003.

[10] Min-Chi Tsai, “Smart Memory Controller Design for Video Applications,” Master thesis: National Chiao Tung University, July 2006.