

DESIGN AND VERIFICATION ANALYSIS OF APB3 PROTOCOL WITH COVERAGE

Akhilesh Kumar and Richa Sinha

Department of E&C Engineering, NIT Jamshedpur, Jharkhand, India

ABSTRACT

Today in the era of modern technology micro electronics play a very vital role in every aspects of life of an individual, increasing use for micro electronics equipments increases the demand for manufacturing its components and its availability, reducing its manufacturing time, resulting in increasing the failure rate of the finished product. In order to overcome this problem the Technocrats develop a method called Verification, a process which is a part of manufacturing microelectronics products. So approximately 30% of the effort spent on the average project is consumed by design and 70% in verification. For this reason, methods which improve the efficiency and accuracy of hardware design and verification are immensely valuable. The current VLSI design scenario is characterised by high performance, complex functionality and short time-to-market. A reuse based methodology for SoC design has become essential in order to meet these challenges. The work embodied in this paper presents the design of APB 3 Protocol and the Verification of slave APB 3 Protocol. Coverage analysis is a vital part of the verification process; it gives idea that to what degree the source code of the DUT has been tested. The Functional coverage analysis increases the verification efficiency enabling the verification engineer to isolate the areas of un-tested function. The design and verification IP is built by developing verification components using Verilog and System Verilog respectfully with relevant tools such as Riviera, which provides the suitable building blocks to design the test environment.

KEYWORDS: AMBA (Advanced Microcontroller Bus Architecture), APB(Advanced peripheral Bus), Functional coverage analysis, RTL (Register Transfer Level) design, System Verilog, SOC (System on chip), DUT (Design Under Test), Design intellectual property (DIP), Verification intellectual property (VIP).

I. INTRODUCTION

Intellectual Property (IP) Cores are of first line of choice in the development of Systems-on-chip (SOC). Typically, a SoC is an interconnection of different pre-verified IP blocks which communicate using complex protocols. Approaches adopted to facilitate plug and- play style IP reuse include the development of a few standard on-chip bus architectures such as CoreConnect[11] from IBM, AMBA[9] from ARM among others, and the work of the VSI Alliance[8] and the OCP-IP[10] consortium. Designers are usually provided with voluminous specifications of the protocols used by the IP blocks and the underlying bus architecture. IP Cores are register transfer level (RTL) codes which achieve certain desired functionality. Today the foundation of digital systems design depends on Hardware description languages (HDLs) rather than schematic diagrams. These RTL codes are well tested codes which must be ready for any use in SOC development.

Modern computer systems rely more and more on highly complex on-chip communication protocol to exchange data. The enormous complexity of these protocol results from tackling high-performance requirements. Protocol control can be distributed, and there may be non-atomicity or speculation. The electronics industry has entered the era of multi-million-gate chips, and there is no turning back. This technology promises new levels of integration on a single chip, called the System-on-a- Chip (SOC) design, but also presents significant challenges to the chip designer. Processing cores on a single chip,

may number well into the high tens within the next decade, given the current rate of advancements [1]. The important aspect of a SOC is not only which components or blocks it houses, but also how they are interconnected. The current VLSI design scenario is characterised by high performance, complex functionality and short time-to-market. A reuse based methodology for SOC design has become essential in order to meet these challenges. AMBA is a solution for the blocks to interface with each other.

In the present paper the discussion is made on the Design intellectual property (DIP) of the master and slave of the APB3 protocols and the Verification intellectual property (VIP) slave with coverage analysis.

II. OBJECTIVE OF THE AMBA

The objective of the AMBA specification [1] is to:

1. facilitate right-first-time development of embedded microcontroller products with one or more CPUs, GPUs or signal processors,
2. be technology independent, to allow reuse of IP cores, peripheral and system macrocells across diverse IC processes, encourage modular system design to improve processor independence, and the development of reusable peripheral and system IP libraries
3. Minimize silicon infrastructure while supporting high performance and low power on-chip communication.

2.1 History of AMBA

The AMBA was introduced by ARM in 1996 and is widely used as the on-chip bus in SoC designs. AMBA is a registered trademark of ARM. The first AMBA buses were ASB and APB. In its 2nd version, AMBA 2, ARM [2] added AMBA AHB that is a single clock-edge protocol. In 2003, ARM introduced the 3rd generation, AMBA3 [3], including AXI to reach even higher performance interconnect and the Advanced Trace Bus (ATB) as part of the Core Sight on-chip debug and trace solution. In 2010, ARM introduced the 4th generation, AMBA 4,[1] including AMBA 4 AXI4, AXI4-Lite, and AXI4-Stream Protocol, the AMBA 4.0 protocol defines five buses/interfaces:

- Advanced extensible Interface (AXI)-A high performance ,flexible protocol
- Advanced High-performance Bus (AHB)-retained for compatibility and to ease the transition
- Advanced System Bus (ASB)- no longer actively supported
- Advanced Peripheral Bus (APB) - retained for support of simple, low bandwidth peripherals
- Advanced Trace Bus (ATB)

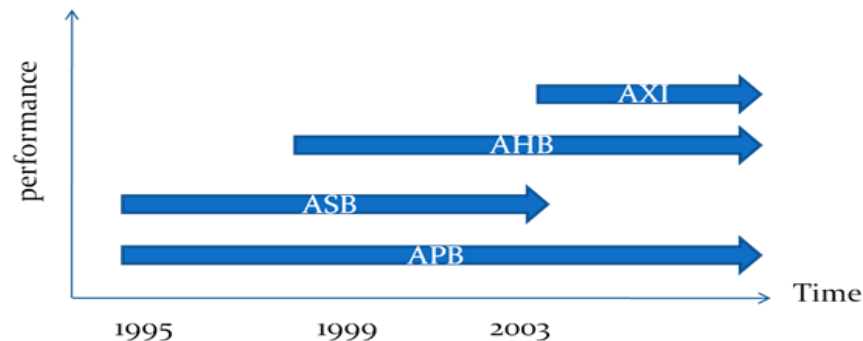


Figure 1. Protocols of AMBA

2.2. AMBA Protocol Family

AHB (Advanced High Performance Bus) is for high performance, high clock frequency system modules with suitable for medium complexity and performance connectivity solutions. It supports multiple masters.

AHB-Lite is the subset of the full AHB specification which intended for use where only a single master is used.

APB (Advanced Peripheral Bus) mainly used as an ancillary or general purpose register based peripherals such as timers, interrupt controllers, UARTs, I/O ports, etc. It is connected to the system bus via a bridge, helps reduce system power consumption. It is also easy to interface to, with little logic involved and few corner-cases to validate.

III. ABOUT APB 3 PROTOCOL

3.1 An AMBA APB 3 Typical System [1][15]

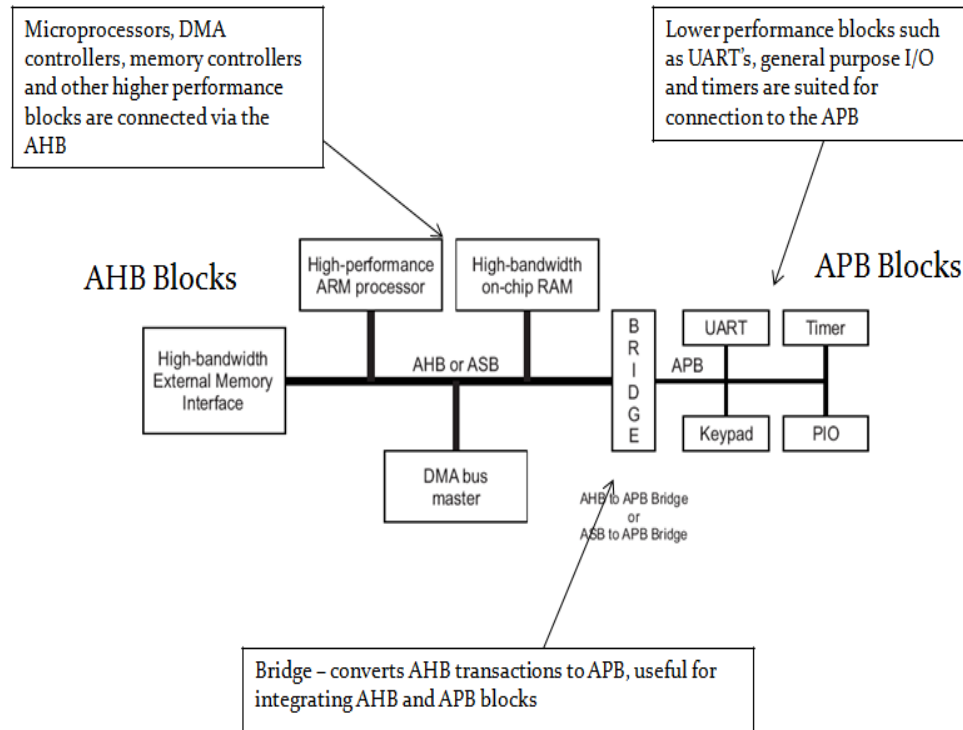


Figure 2. AMBA APB3 Typical System

Figure 2 illustrates a typical AMBA system. Several master or slave devices are connected via AHB which are often used as system bus. The data transfer between each memory module and peripheral devices also can be done by it. The bridge locates between system bus and peripheral bus. While transferring data from processor to peripheral devices like URAT, timer, peripheral I/O and keyboard, the bridge convert the transferred signal from one type to another for satisfying different performance and protocol.

The APB 3 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.

The APB can interface with the AMBA Advanced High-performance Bus Lite (AHB-Lite) and AMBA Advanced Extensible Interface (AXI). You can use it to provide access to the programmable control registers of peripheral devices.

3.2. AHB VS APB [2][16]

Table1. AHB vs APB

	AMBA AHB	AMBA APB
Feature	<ul style="list-style-type: none"> •High performance •Pipelined operation •Multiple bus masters •Burst transfers •Split transactions •Single cycle bus master handover •Single clock edge operation (rising edge) •Wider data bus configuration 	<ul style="list-style-type: none"> •Low power •Latched address and control •Simple interface •Suitable for many peripherals •Single clock edge operation (rising edge)
components	<ul style="list-style-type: none"> •AHB master •AHB slave •AHB arbiter •AHB decoder 	<ul style="list-style-type: none"> •APB bridge (slave on AHB or ASB) •APB slave

3.3. When to use AHB OR APB [17][18]

AHB uses a full duplex parallel communication. It used in external memory interface, with high bandwidth peripheral with FIFO interfaces. It is also used in on chip memory blocks whereas the APB uses massive memory-I/O accesses.

The APB is mainly proposed for connecting to simple peripherals. It can be seen that the APB comes with a low power peripheral. This Bus can also be used in union with either version of the system bus. It group narrow bus peripherals to avoid loading the system bus.

Separate the bus address decoding into two levels make it easier (in most cases) to do timing budget. The address decoding logic will be easier to design as well. Usually, AHB decoder is used to decode larger memory blocks. And then I/O space (small memory blocks) is decoded by APB decoder (inside APB Bus Bridge).

E.g. you might have 4 memory blocks and 20 I/O devices. If you put them all into one level of address decoding, you might end up a big bus multiplexer which operates at lower clock frequency.

By separating I/O devices in APB memory map, you can have a smaller and faster AHB interconnect, and a second level of APB interconnect which might take one or two more extra cycle to access.

IV. APB3 FSM DIAGRAM

Figure 3 shows the Finite State diagram of peripheral bus activity of the APB[14].

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, writes, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state.

Exit from the ACCESS state is controlled by the **PREADY** signal from the slave:

- 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

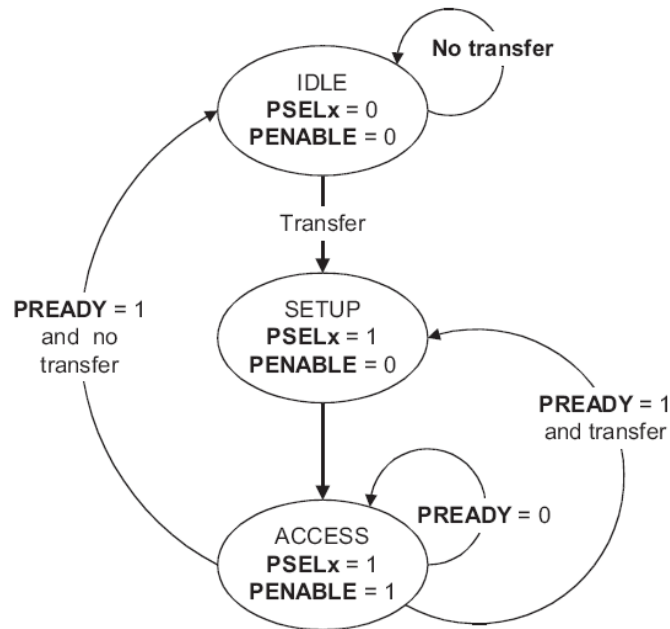


Figure 3. FSM diagram of APB3

V. MICRO ARCHITECTURE OF APB3

Figure 4. shows the micro architecture of APB3 Protocols[1]

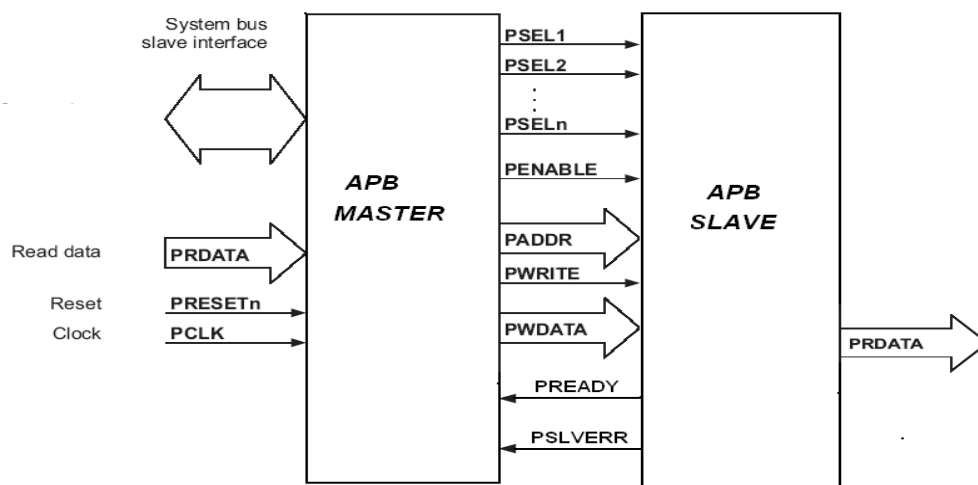


Figure 4. Interfacing of APB Master and Slave

5.1 APB3 Master Description

There is a single bus master on the APB, thus there is no need for an arbiter. The master drives the address and write buses and also performs a combinatorial decode of the address to decide which

VII. SIMULATION RESULT OF VERIFICATION OF APB 3

7.1 SLAVE VERIFICATION

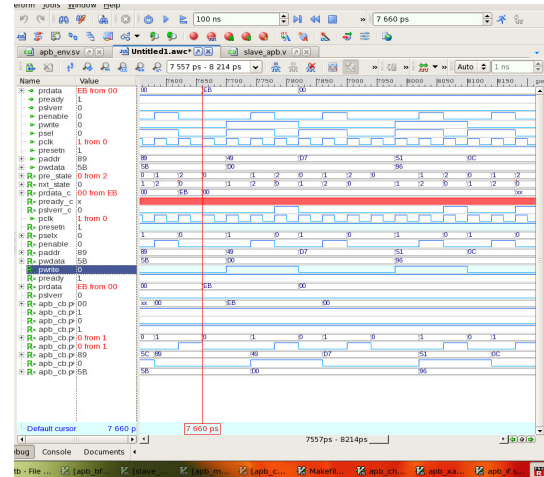


Figure 9. Read Operation

VIII. COVERAGE ANALYSIS

File:///home/student/richa/richa_slave/rundir/fcover_report.html

Most Visited Red Hat Red Hat Magazine Red Hat Network Red Hat Support

Digital Questions fcover_report.html fcover_report.html fcover_report.html

Coverage Report Structure
type apb_xactn.apb_cov

FUNCTIONAL COVERAGE:

Coveragegroup	Hrs	Goal / At Least	Status
TYPE apb_xactn.apb_cov	100.000%	100.000%	Covered
COVERPOINT apb_xactn.apb_cov - addb_cp	100.000%	100.000%	Covered
bin b1	15	1	Covered
bin b2	12	1	Covered
bin b3	9	1	Covered
bin b4	17	1	Covered
bin b5	17	1	Covered
COVERPOINT apb_xactn.apb_cov - data_cp	100.000%	100.000%	Covered
bin b1	16	1	Covered
bin b2	13	1	Covered
bin b3	9	1	Covered
bin b4	24	1	Covered
bin b5	10	1	Covered

TOTAL FUNCTIONAL COVERAGE: 100.000% COVERAGEGROUP TYPES: 1

Done

hemanth - File Browser apb_xactn.sv (~/.hemanth/h... [terminal]

Applications Places System

Vol. 1, Issue 5, pp. 310-317

IX. CONCLUSION

In the paper a general definition for APB3 protocol flexibility and compatibility is shown. We describe study of AMBA3 APB SOC bus protocol and their performance. Here the design and verification of low peripheral processor's data transfer protocol has been discussed. And also how the error has been reduced without loss of data while transferring.

ACKNOWLEDGEMENT

This work was supported by CVC PVT LTD, Bangalore.

REFERENCES

- [1] ARM, "AMBA Specification Overview", available at <http://www.arm.com/>.
- [2] ARM, "AMBA Specification (Rev 2.0)", available at <http://www.arm.com>.
- [3] ARM, "AMBA AXI Protocol Specification", available at <http://www.arm.com>
- [4] Samir Palnitkar "Verilog HDL"
- [5] Chris Spear, SystemVerilog for Verification, New York : Springer, 2006
- [6] <http://www.testbench.co.in>
- [7] http://www.doulos.com/knowhow/sysverilog/ovm/tutorial_0
- [8]. Virtual Socket Interface Alliance. <http://www.vsi.org>.
- [9]. ARM. Advanced micro-controller bus architecture specification. [http://www.arm.com/armtech/AMBA spec](http://www.arm.com/armtech/AMBA_spec), 1999.
- [10]. Open Core Protocol Int'l Partnership Association Inc. Open core protocol specification. <http://www.ocpip.org>, Release 1.0, 2001.
- [11] IBM. 32-bit processor local bus, architecture specifications. <http://www-3.ibm.com/chips/products/coreconnect/>, Version 2.9.
- [12] J.Bergeron, "What is verification?" in Writing Test benches: Functional Verification of HDL Models, 2nd ed. New York: Springer Science, 2003, ch.1, pp. 1-24.
- [13] International Technology Roadmap for Semiconductors [Online]. Available: <http://www.itrs.net/Links/2006Update>
- [14] infocenter.arm.com/help/topic/com.arm.doc.ih0024b/index.html
- [15] nthur.lib.nthu.edu.tw/bitstream/987654321/7242/9/630208.pdf
- [16] http://en.wikipedia.org/wiki/Advanced_Microcontroller_Bus_Architecture
- [17] <http://www.differencebetween.net/technology/difference-between-ahb-and-apb/>
- [18] <http://groups.google.com/group/comp.sys.arm/msg/55e6c80bfd9f99ce?pli=1>

Authors

Akhilesh Kumar received B.Tech degree from Bhagalpur university, Bihar, India in 1986 and M.Tech degree from Ranchi, Bihar, India in 1993. He has been working in teaching and research profession since 1989. He is now working as H.O.D. in Department of Electronics and Communication Engineering at N.I.T. Jamshedpur, Jharkhand, India. His interest of field of research is analog and digital circuit design in VLSI.



Richa Sinha received B. E. Degree from RajaRamBapu Institute of Technology Shivaji University, Kolhapur, Maharashtra, India in 2007. Currently she is pursuing M. Tech project work under the guidance of Prof. Akhilesh Kumar in the Department of Electronics & Communication Engg, N. I. T., Jamshedpur. Her interest of field is ASIC Design & Verification.

