AMBA AXI/AHB/APB Bus and System Design and Verification

June 20-22, 2017

Ando Ki, Ph.D.
(adki@future-ds.com)

Copyright Notice

-- Contact information --
Ando Ki, Ph.D.

adki@future-ds.com
www.Futue-DS.com

Abstract

SoC (System-on-Chip, 시스템반도체)를 위한 IP (Intellectual Property, 반도체설계자산)는 프로세서의 제어를 받아 동작하고, 프로세서는 업계 표준 버스를 통해 IP를 제어한다. 따라서 업계 표준 버스에 잘 인터페이스 되는지를 기준으로 IP를 설계하고 검증하는 것이 필요하고, 이 강좌에서는 업계 표준 버스(예, AMBA APB, AHB, AXI 등)와 BFM(Bus Functional Model)을 활용하여 IP를 설계하고 검증하는 환경과 기법을 다룬다.

IP of SoC works under the control of processor that controls IP through de facto standard bus. Therefore, IP should be designed and verified in the context of bus. This lecture addresses de facto standard buses (AMBA APB, AHB, AXI) and BFM-based IP design/verification environments and techniques.

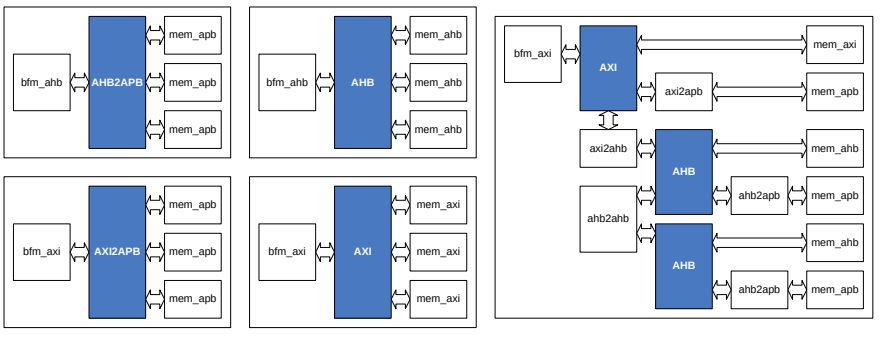
Target audience and prerequisites

- Target audience
  - This lecture is prepared for engineers and students who are interested in developing IP based on AMBA AXI, AHB and APB.
- Prerequisites
  - Basic knowledge of digital logic design
  - Basic knowledge of the Verilog HDL language
  - Basic knowledge of the C language
  - Experience with industry standard HDL simulator such as ModelSim
  - Experience with FPGA such as Xilinx and its development environment, i.e., ISE
  - Experience with industry standard C compilation tool-chain such as Visual Studio and GNU GCC
  - Experience with industry standard Operating Systems and text editor such as Windows/Linux and Vi/Vim.

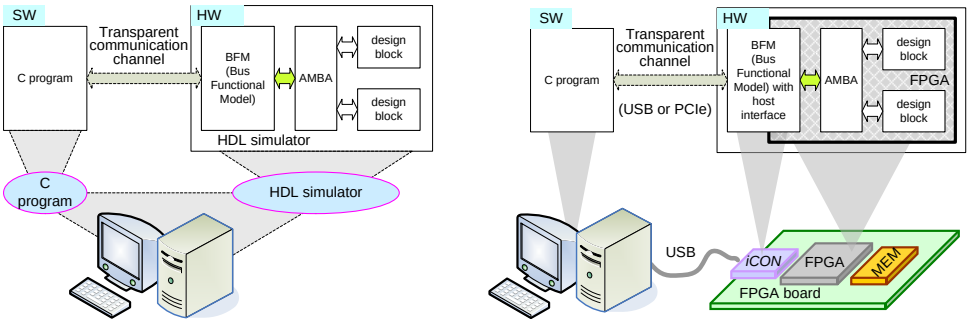
Goals and objectives

- Goals
  - Understanding of IP design and verification in the context of de facto standard buses
  - Acquiring the working knowledge of well-known on-chip buses including AMBA AXI, AHB and APB
  - Practicing design and verification of bus-based IP
- Objectives
  - Understanding of AMBA AXI/AHB/APB as on-chip bus
  - Acquiring skills to use AMBA AXI/AHB/APB BFM to develop IP
  - Understanding of task/file/BFM based design and verification

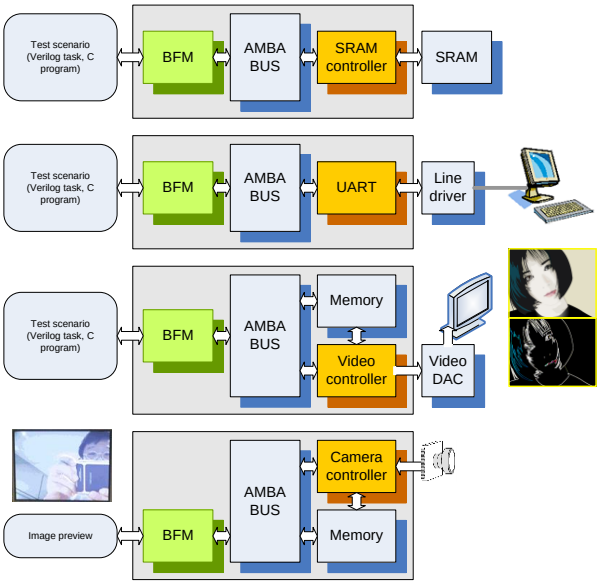
AMBA



BFM with HW/SW Co-Simulation



Bus-based example designs



Lecture schedule

	10:00	11:00	12:00	1:00	2:00	3:00	4:00
1st	0/1	2/3		4	5	6	7
2nd	8/9	10		11	12	13	14/15
3rd	16/17	18/19		20	21/22	23/24	25

- 0: Lecture overview
- 1: Bus introduction
- 2: OCB introduction
- 3: BFM verification
- 4: Arbiter project

- 5: AMBA APB
- 6: AMBA APB BFM task
- 7: UART with APB

- 8: AMBA AHB
- 9: AMBA AHB 5
- 10: AMBA AHB BFM task
- 11: AMBA AHB design

- 12: AMBA AHB to APB bus bridge
- 13: AHBA AHB to AHB bus bridge
- 14: DMA introduction
- 15: AMBA AHB DMA

- 16: AMBA AXI introduction
- 17: AMBA AXI details
- 18: AMBA AXI BFM task
- 19: AMBA AXI design

- 20: AMBA AXI stream

- 21: AMBA AXI to APB
- 22: UART verification with AMB AXI-to-APB (demo)
- 23: AMBA AXI to AXI
- 24: AMBA AXI DMA

- 25: Summary

Environment

- General environment
  - Cygwin on Windows
  - gvim (vim or vi) for text editor
  - GCC C compiler tool-chan
- HDL simulation
  - HDL simulator: ModelSim
  - VCD waveform viewer: gtkwave
- FPGA targeting
  - Xilinx FPGA development: ISE
    - Logic synthesis: XST
    - PnR: map, par, …
- FPGA board
  - FPGA board: Dynalith iNCITE or Core-A BD
  - FPGA board IDE: iNSPIRE
  - Host interface: iVORY
  - C API: iVORY
- USB-to-serial
- Text-terminal emulator: teraterm or hyperterminal

Coding guidelines

- Module name and file name should be the same
- Each directory should have directory clean-up script: Clean.bat, Clean.sh, Makefile
- Each HW IP would contain the following sub-directories

directory		remarks
bench	Test-bench	
	c/verilog/vhdl/systemc	Test-bench written in the specific language
beh	behavioral model if applicable	
	c/verilog/vhdl/systemc	behavioral model written in the specific language
doc	manual and other helpful document	
drv	device driver if applicable and would contains the following sub-directory	
	c	
rtl	RTL model if applicable and would contains sub-directory like 'beh'	
	verilog/vhdl/systemc/c	RTL model written in the specific language
sim	simulation related if applicable	
	modelsim/vcs/ncsim	Sub-directories for HDL simulator
syn	synthesis related if applicable	
	xst/synp/dc/fc	Sub-directories for logic synthesizer

Reference

- 기안도, 시스템 집적 반도체 설계검증 환경과 기법, 홍릉과학출판사, 2008.
- Chong-Min Kyung, Ando Ki et.al., '*Chapter: SoC Prototyping and Verification' in 'Essential Issues in System-on-a-Chip Design*', edited by Youn-Long Lin, Springer-Verlag, 2006.
- D.L. Perry and H.D. Foster, *Applied Formal Verification,* McGraw-Hill, 2005.
- D. Del Corso et.al., Microcomputer buses and links, Academic press, 1986.
- D. Cohen, On holy wars and a plea for peace, IEEE Computer Vol.14, No.10, Oct. 1981, p.48-54.
- 기안도 외, 고중첩 버스 : HiPi-Bus (Highly Pipelined Bus : Hipi-Bus), 대한전자공학회 학술발표회 논문집 (반도체/재료부품/CAD/VLSI) 제10권 1호, 1992.1, 31-37.
- Ando Ki et.al., Highly Pipelined Bus: HiPi-Bus, JTC-CSCC : Joint Technical Conference on Circuits Systems, Computers and Communications, 1991.
- Seongwoon Kim and Ando Ki et.al., RACE on a physically distributed and logically shared memory system.
- AMBA Specification, Rev 2.0, ARM Limited. (AMBA 2.0 APB)
- AMBA™ 3 APB Protocol v1.0, IHI 0024B, ARM, 2004. (AMBA 3.0 APB 1.0)
- AMBA® APB Protocol Version: 2.0, IHI 0024C (ID041610), ARM, 2010. (AMBA 4.0 APB 2.0)
  AMBA® AXI Protocol v1.0 Specification, IHI 0022B, ARM Limited, 2004.
- AMBA AXI Protocol Specification, v1.0, ARM IHI 0022B, 2003-2004.
- AMBA AXI and ACE Protocol Specification, ARM IHI 0022E, 2003-2013.
- AMBA 4 AXI4-Stream Protocol Specification, v1.0, ARM IHI 0051A, 2010.
- 기안도, Core-A 프로세서를 활용한 응용 설계, 홍릉과학출판사, 2011.12.
- 기안도, Core-A 프로세서를 활용한 플랫폼 설계, 홍릉과학출판사, 2010.3.
- S. Pasricha and N. Dutt, On-Chip Communication Architectures System on Chip Interconnect, Morgan Kaufmann Pub. 2011.