Design Verification of Incrementor Logic using AMBA-APB Protocol

Ayush Gupta
Department of Electronics & Communication Engineering
SRM Institute of Science and Technology, Kattankulathur
Chennai, India
ag5445@srmist.edu.in

***Abstract*—Featured in this paper is the design verification of an incrementor logic using the state operations undergone by the Advanced Peripheral Bus (APB) which is a part of the Advanced Microcontroller Bus Architecture (AMBA) protocol family. This paper gives an outline of the AMBA-APB bus architecture and explain the APB bus state operations in detail. The RTL design is done using the Verilog HDL according to the state diagram specification and underlying logic taken into reference and is verified using the Vyoma's UpTickPro Verification Tool.**

***Keywords*—AMBA, APB, Bus Architecture, State Machine**

I. Description

Advanced microcontroller bus architecture (AMBA) is an architecture that is widely used in system-on-chip designs, which are found on chip buses. The AMBA specification standard is used for designing high-level embedded microcontrollers. Five interfaces are defined within the AMBA specification out of which we will looking at the Advanced peripheral bus (APB) interface. Unlike AHB, it is a Non-Pipelined protocol, used to connect low-bandwidth peripherals. Mostly, used to connect the external peripheral to the SOC. In APB, every transfer takes at least two clock cycles (SETUP Cycle and ACCESS Cycle) to complete. It can also interface with AHB and AXI protocols using the bridges in between.

II. State Diagram

All the operating states of the APB protocol are shown in the state diagram in figure 1. The states are described as IDLE, SETUP and ACCESS. IDLE is the default state of the APB Interface. 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. The enable signal, PENABLE, is asserted in the ACCESS state. The address, write, 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 such that: if PREADY is held LOW by the slave then the peripheral bus remains in the ACCESS state if it 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.

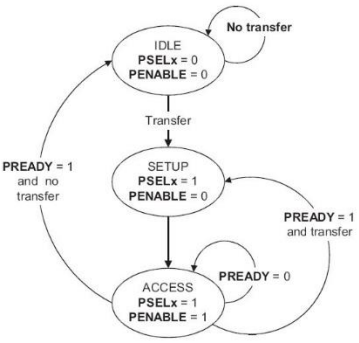

Fig. 1 APB Operational Activity State Diagram

III. Read/Write Operation

The Design here focuses on how to realize an incrementor logic where the source verilog RTL code serves as our APB master and the testbench for the same serves as our APB Slave. PRDATA is given as stimulus in the APB Slave [testbench] along with PREADY signal and the PWDATA holds the value that is read from the PADDR and stored in PRDATA during read operation. In Fig 3. an incrementor signal is also taken labelled as add_i which determines when the read and write operation would be performed for the same.

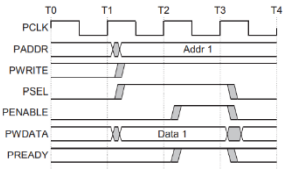

Fig. 2[a] Write Operation

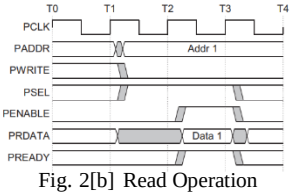

Fig. 2[b] Read Operation

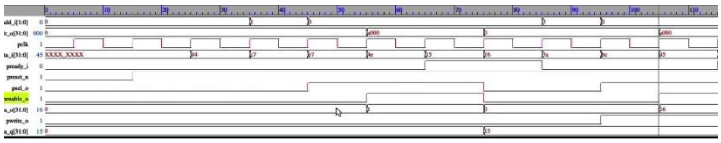

Fig. 3

References

[1] Meghana Jain H K1, Dr. Punith Kumar M B2, Verification of Advanced Peripheral Bus Protocol (APB), IRJET, June 2021
[2] Shankar, Dipti Girdhar, Neeraj Kr. Shukla, Ph.D, Design and Verification of AMBA APB Protocol, IJCA (0975 – 8887), Volume 95– No.21, June 2014
[3] https://github.com/guptayush2112/DV_AMBA_APB