A Project Report on

AHB to APB BRIDGE DESIGN

VLSI Design Internship

Batch DI - 45

Submitted By

Rudra Prasad Goswami

rudragoswami5562@gmail.com

1. PROTOCOL

1.1. AMBA based Microcontroller:

An AMBA-based microcontroller generally features a high-performance system backbone bus, such as AMBA AHB or AMBA ASB, which is designed to handle the external memory bandwidth efficiently. The CPU, on-chip memory, and other Direct Memory Access (DMA) devices are connected to this bus, which facilitates a high-bandwidth interface for the majority of data transfers. Additionally, this high-performance bus includes a bridge to the lower bandwidth APB, where most of the system's peripheral devices are connected.

AMBA, developed by Arm in 1996, initially introduced the Advanced System Bus (ASB) and the Advanced Peripheral Bus (APB). With the release of AMBA 2 in 1999, Arm introduced the AMBA High-performance Bus (AHB), a protocol based on a single clock edge. In 2003, Arm released the third generation, AMBA 3, which included the Advanced eXtensible Interface (AXI) for even higher performance interconnects and the Advanced Trace Bus (ATB) as part of the CoreSight on-chip debug and trace solution.

The Advanced Microcontroller Bus Architecture (AMBA) specification outlines a standard for on-chip communication in the design of high-performance embedded microcontrollers. The specification defines three distinct buses:

- Advanced High-performance Bus (AHB)
- Advanced System Bus (ASB)
- Advanced Peripheral Bus (APB).

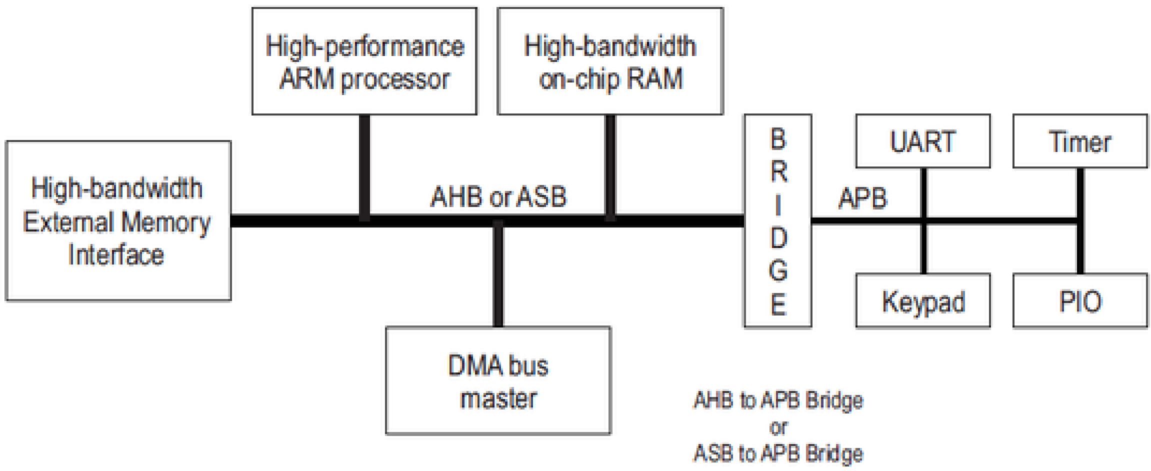


fig 1: A block diagram of Typical AMBA System

AMBA APB provides the basic peripheral macro cell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically:

- have interfaces which are memory-mapped registers
- have no high-bandwidth interfaces
- are accessed under programmed control.

The external memory interface is application-specific and may only have a narrow data path, but may also support a test access mode which allows the internal AMBA AHB, ASB and APB modules to be tested in isolation with system-independent test sets.

1.2 AHB (Advanced High-performance Bus):

The AMBA AHB is for high-performance, high clock frequency system modules.

The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro

cell functions. AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated test techniques.

1.3 APB (Advanced Peripheral Bus):

The AMBA APB is for low-power peripherals.

AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus.

1.4 AHB to APB Bridge:

The AHB-to-APB Bridge functions as an AHB slave, serving as an interface between the high-speed AHB and the low-power APB. It converts read and write transfers on the AHB into corresponding transfers on the APB. Since the APB is not pipelined, wait states are introduced during transfers to and from the APB, causing the AHB to wait when necessary.

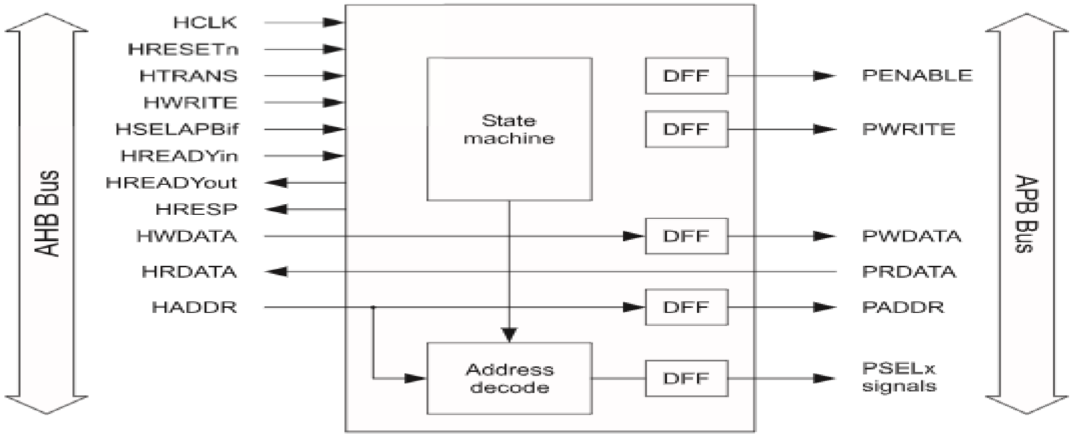
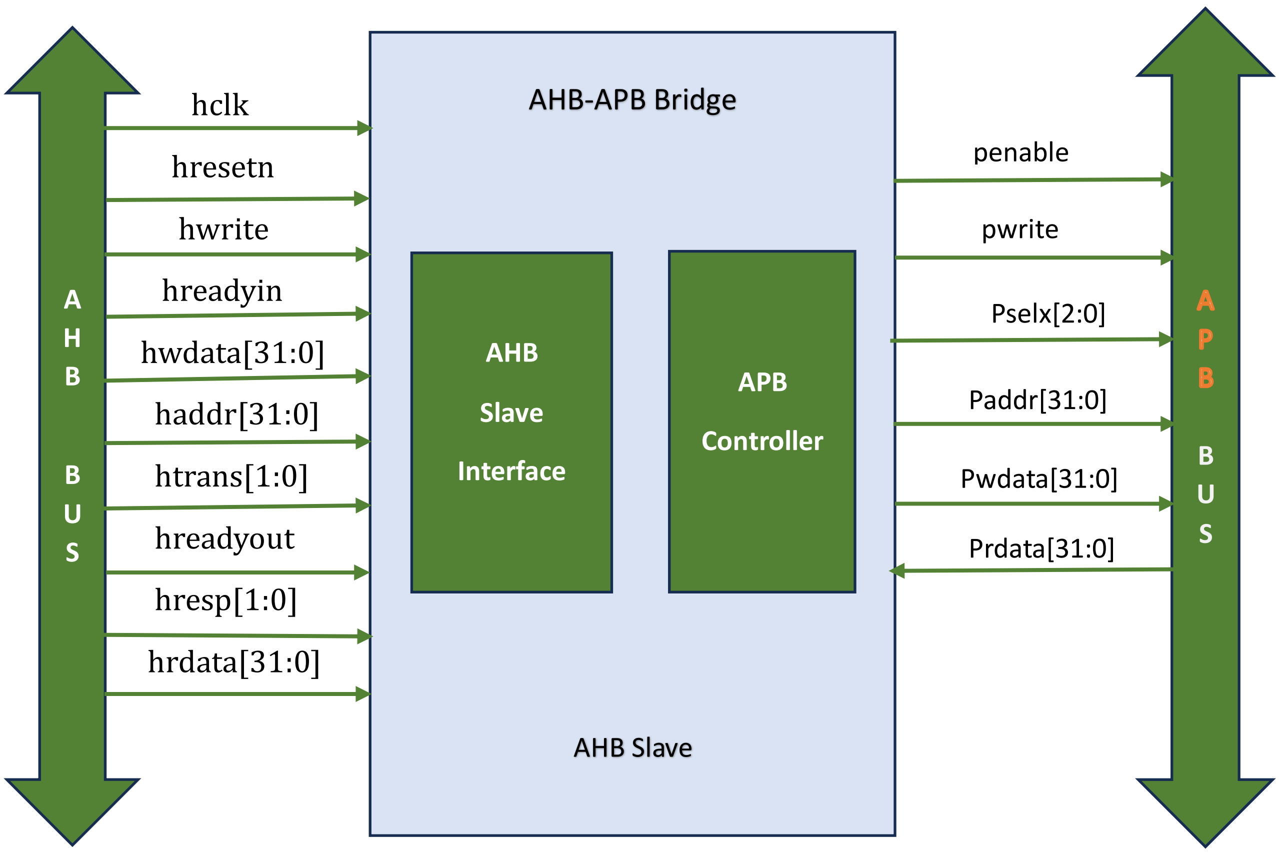
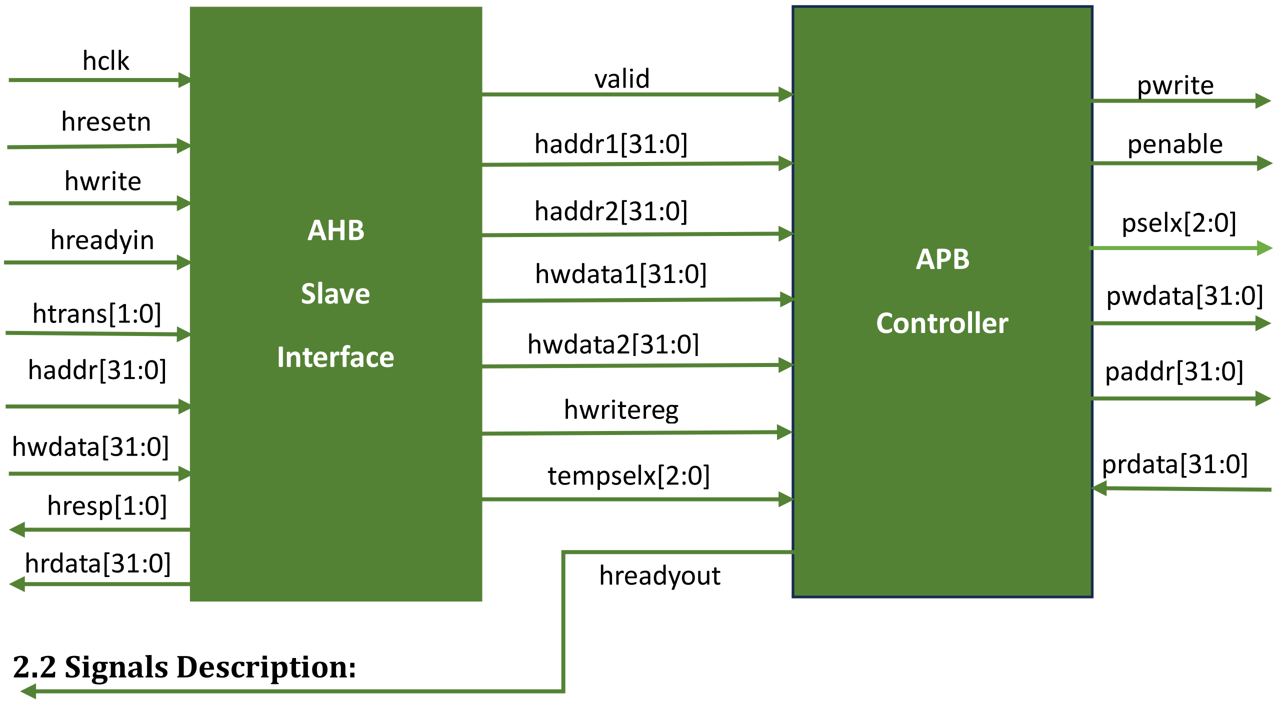


fig 2: Block diagram of Bridge Module

2. BLOCK DIAGRAM & ARCHITECTURE:

2.1 Block diagram of AHB to APB Bridge:





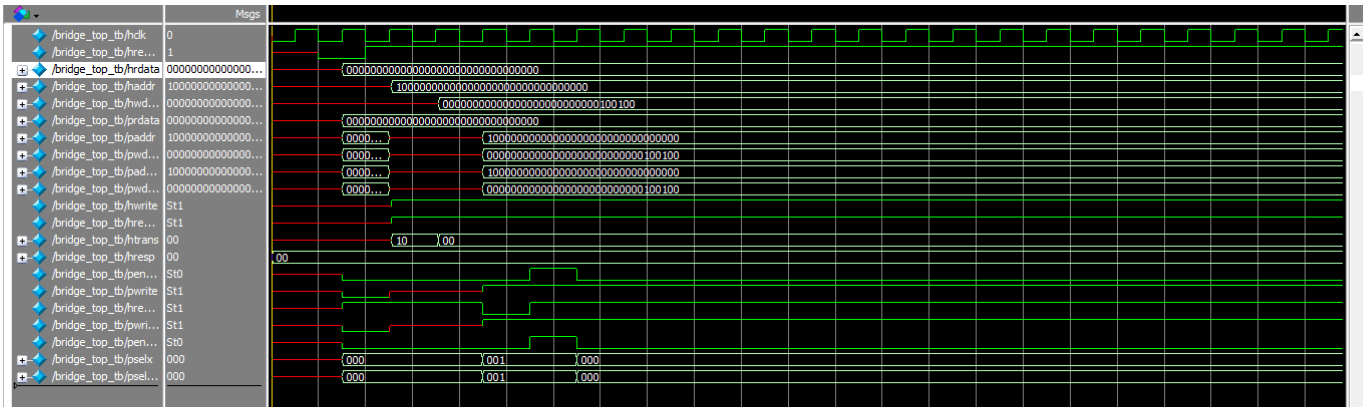
2.2 Signals Description:

Signals	Type	Direction	Description
Hclk	Bus clock	Input	This clock times all bus transfers.
Hresetn	Reset	Input	The bus reset signal is active LOW, and is used to reset the system and the bus.
Hwrite	Transfer Direction	Input	When HIGH this signal indicates a write transfer, and when LOW, a read transfer.
Hreadyin Hreadyout	Transfer Done	Input	When HIGH the HREADY signal indicates that a transfer has finished on the bus. This signal may be driven LOW to extend a transfer.
Htrans[1:0]	Transfer Type	Input	This indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY.
Haddr[31:0]	Address Bus	Input	The 32-bit system address bus.
Hwdata[31:0]	Write Data Bus	Input	The write data bus is used to transfer data from the master to the bus slaves during write operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.
Hresp[1:0]	Transfer response	Input	The transfer response provides additional information on the status

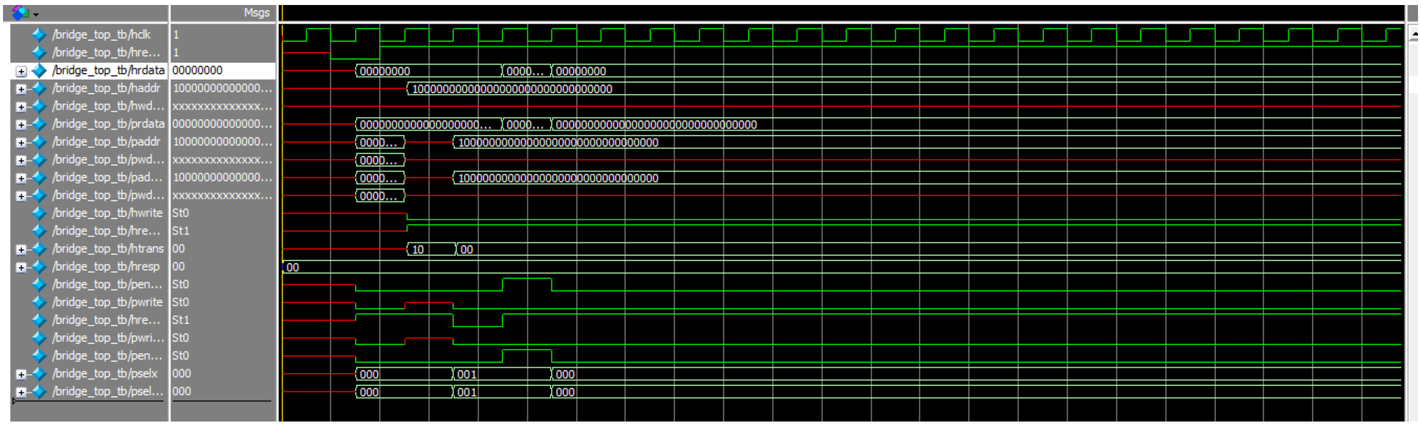
			of a transfer. This module will always generate the OKAY response.
Hrdata[31:0]	Read Data Bus	Input	The read data bus is used to transfer data from bus slaves to the bus master during read operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.
Penable	Peripheral enable	Output	This enable signal is used to time all accesses on the peripheral bus. PENABLE goes HIGH on the second clock rising edge of the transfer, and LOW on the third (last) rising clock edge of the transfer.
Pwrite	Peripheral Transfer Direction	Output	This signal indicates a write to a peripheral when HIGH, and a read from a peripheral when LOW. It has the same timing as the peripheral address bus.
Prdata[31:0]	Peripheral Read Data Bus	Input	The peripheral read data bus is driven by the selected peripheral bus slave during read cycles (when PWRITE is LOW).
Pselx[2:0]	Peripheral Slave Select	Output	There is one of these signals for each APB peripheral present in the system. The signal indicates that the slave device is selected, and that a data transfer is required. It has the same timing as the peripheral address bus. It becomes HIGH at the same time as PADDR, but will be set LOW at the end of the transfer.
Paddr[31:0]	Peripheral Address Bus	Output	This is the APB address bus, which may be up to 32 bits wide and is used by individual peripherals for decoding register accesses to that peripheral. The address becomes valid after the first rising edge of the clock at the start of the transfer. If there is a following APB transfer, then the address will change to the new value, otherwise it will hold its current value until the start of the next APB transfer.
Pwdata[31:0]	Peripheral Write Data Bus	Output	The peripheral write data bus is continuously driven by this module, changing during write cycles (when PWRITE is HIGH).

3. SIMULATION RESULTS: [Tool Used : ModelSim]

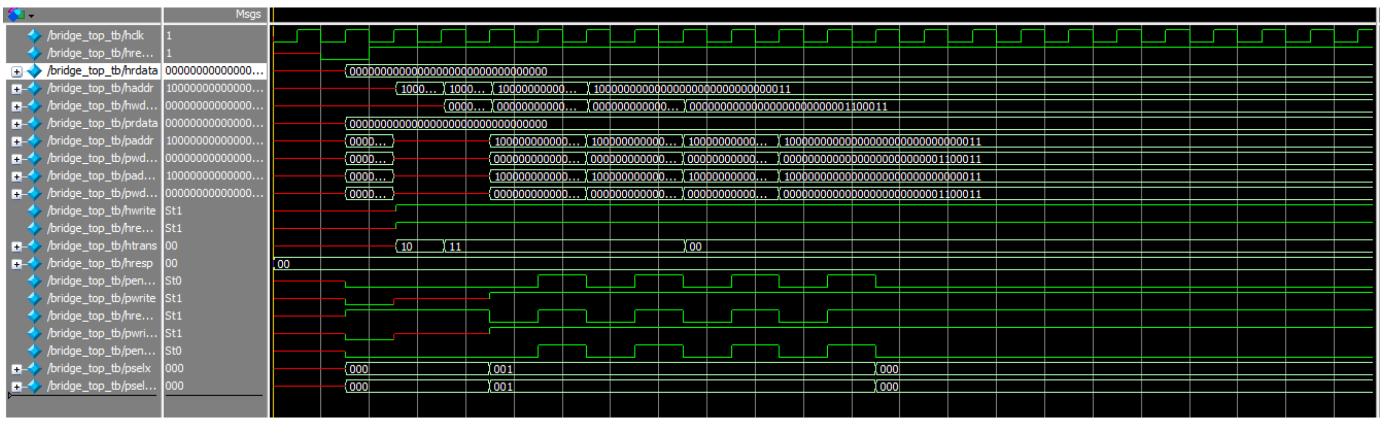
3.1 Single Write:



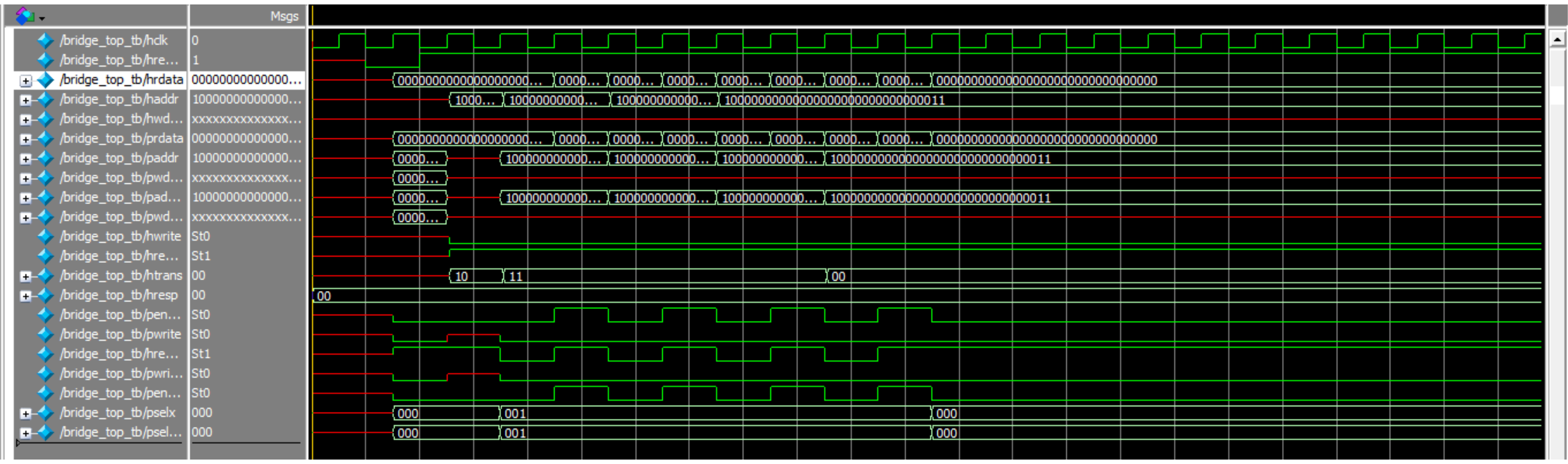
3.2 Single Read:



3.3 Burst Write with Increment 4 :



3.4 Burst Read with Increment 4 :



4. SYNTHESIS RESULTS: [Tool Used : Quartus Prime]

4.1 RTL Schematic: