WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Comparative Analysis of Communication Protocols in VLSI Design: I2C, SPI, UART, AMBA, and CAN

Laharika Salivoju[a],Balaga Bhagavati Rao[b],Bapuji Banothu[c],Dr Bommidi Sridhar[d],Prahas Surabhi[e]

[a]UG Student,Ibrahimpatnam,RangaReddy and 501506,India
[b]Associate Professor,Ibrahimpatnam,RangaReddy and 501506,India
[c]PG Student,Ibrahimpatnam,RangaReddy and 501506,India
[d]Professor and HOD ,Ibrahimpatnam,RangaReddy and 501506,India
[e]PG Student,Ibrahimpatnam,RangaReddy and 501506,India

ABSTRACT :

Communication protocols play a vital role in very-large-scale integration (VLSI) design, enabling seamless interaction between various integrated components on a chip. This paper explores the significance of protocols such as Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver/Transmitter (UART), and Advanced Microcontroller Bus Architecture (AMBA) in facilitating efficient on-chip data transfer and communication. The I2C protocol, developed by Philips Semiconductors in the early 1980s, is widely used for low-speed, short-distance communication between integrated circuits. However, it has limitations in terms of speed, bus length, and noise susceptibility. To address these drawbacks, protocols like SPI, UART, and Controller Area Network (CAN) have been developed, offering improvements in specific areas. The AMBA protocol, consisting of the Advanced High-Performance Bus (AHB) and the Advanced Peripheral Bus (APB), is extensively used in complex VLSI systems to ensure seamless communication between different modules and peripherals. The APB protocol, designed for low-power and low-latency peripherals, operates on a single clock edge and follows a two-phase transaction model. On the other hand, AHB is designed for high-performance, high-bandwidth communication within SoCs, supporting features like burst transfers, split transactions, and multiple bus masters. The judicious selection and implementation of these protocols based on the specific requirements of the VLSI system are crucial for achieving optimal performance, power efficiency, and reliability in modern integrated circuits.

Keywords: VLSI, Communication protocols, Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver/Transmitter (UART), Advanced Microcontroller Bus Architecture (AMBA), Advanced High-Performance Bus (AHB), Advanced Peripheral Bus (APB), System-on-Chip (SoC)

Introduction :

In VLSI (Very Large Scale Integration), one of the earliest and most fundamental protocols for communication is the I2C (Inter-Integrated Circuit) protocol. Developed by Philips Semiconductors (now NXP Semiconductors) in the early 1980s, I2C is widely used for low-speed, short-distance communication between integrated circuits on a single board. It is a simple, bidirectional, two-wire serial bus that consists of a data line (SDA) and clock line (SCL). The I2C protocol is widely used and has many advantages, but it also has several disadvantages, such as speed limitations, limited bus length, bus contentions, additional limitations, slave complexity, power consumption, and noise susceptibility.

To address the disadvantages of I²C, several other communication protocols have been developed and widely adopted in VLSI systems. Each of these protocols offers improvements in specific areas where I²C has limitations: SPI (Serial Peripheral Interface): Advantages: Higher speeds than I²C (up to tens of MHz), full-duplex communication (simultaneous bidirectional data transfer), and simpler implementation for multiple devices using separate chip select lines. Disadvantages: Requires more pins for communication (MOSI, MISO, SCLK, and SS), which can be a limitation in pin-constrained systems. UART (Universal Asynchronous Receiver-Transmitter): Advantages: Simple and widely supported; suitable for longer distances than I2C; and asynchronous operation eliminates the need for a clock line. Disadvantages: Typically slower than SPI and more complex in terms of error detection and correction mechanisms. CAN (Controller Area Network): Advantages: robust and reliable, particularly in noisy environments; supports multiple masters and nodes on the same bus; has error detection and handling mechanisms. Disadvantages: more complex protocol and higher implementation cost compared to I²C.

AMBA Protocol :

The advanced microcontroller bus architecture (AMBA) protocol is a commonly used communication protocol in VLSI. AMBA is a widely used on-chip bus standard for the design of high-performance embedded systems. It defines a set of protocols for communication and data transfer between different

components in a system-on-chip (SoC). The AMBA protocol consists of different interconnect specifications, such as the AHB Advanced High-Performance Bus (AHB) and an Advanced Peripheral Bus (APB), which facilitate efficient communication between various components within an SoC. AMBA protocols are extensively used in the design of complex VLSI systems to ensure seamless communication between different modules and their peripherals.

The following figure shows the block diagram of the AMBA bus.

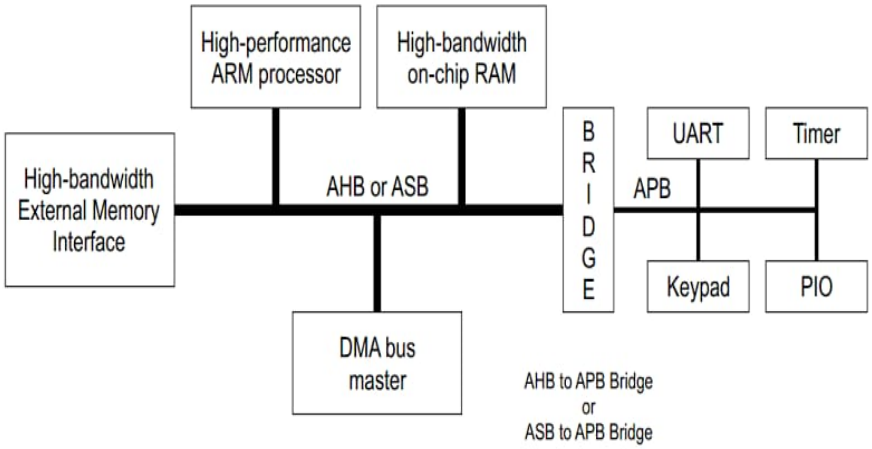

Fig-1: Block diagram of the AMBA Bus

The AMBA (Advanced Microcontroller Bus Architecture) protocol suite developed by ARM includes several protocols designed for different purposes within a system-on-chip (SoC). One of these is the Advanced Peripheral Bus (APB) protocol. The APB is a simple, low-power, low-latency protocol intended for peripheral devices that do not require the high performance of other

The following fig shows the block diagram of the APB Master and APB Slave.

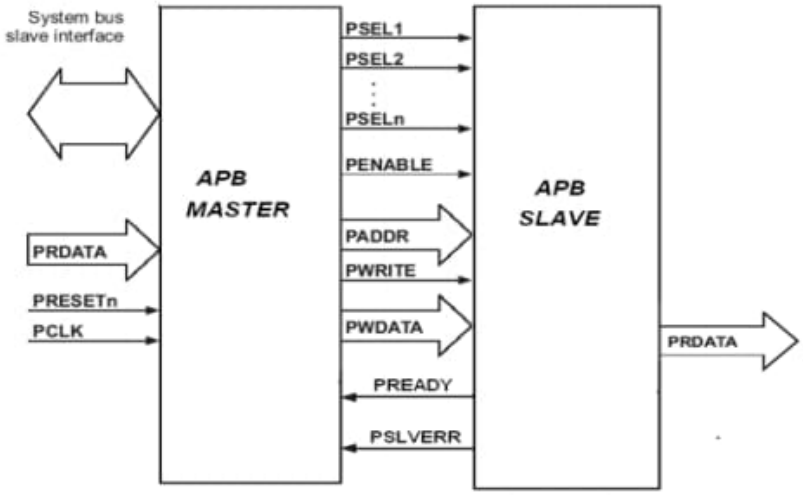

Fig-2 APB master and APB Slave.

2.1 . Key Features of APB Protocol

Simple Design: The APB is designed to be simple and easy to implement, making it suitable for peripherals such as timers, UARTs, and GPIOs.

Low power consumption: Owing to its simplicity, the APB is power-efficient, which is critical for battery-powered and low-power applications.

Single Clock Edge Operation: The APB operates on a single clock edge, which simplifies timing analysis and integration into the SoC.

Address and Data Phases: APB transactions consist of an address phase followed by a data phase. This two-phase operation allows straightforward control and data transfer.

Non-pipelined: APB transactions are non-pipelined, meaning that each transaction must be completed before the next transaction begins. This reduces the complexity but limits the throughput compared to pipelined buses.

Signal Lines: Key signal lines in APB include PADDR: Address bus PSEL: Select a signal for the peripheral PENABLE: Indicates the start of the access phase. PWRITE: Indicates if the operation is a write (1) or read (0). PWDATA: Write data bus PRDATA: read data bus PREADY: Indicates that the peripheral is ready for the next transfer. PSLVERR: indicates a transfer error. Operation
Setup Phase: The PADDR, PSEL, and PWRITE signals were set by the master. The PENABLE value was low during this phase.
Access Phase: The PENABLE signal is asserted high to indicate that the transfer is active. Data transfer occurs on the PWDATA or PRDATA lines depending on whether it is a write or read operation, which indicates when the transfer is complete. If PREADY is high, then the transfer is completed in the next clock cycle.
Completion: Upon completion, PENABLE is deasserted, and the bus returns to the idle state or proceeds with the next transaction.
Advantages: APB is typically used for connecting low-bandwidth peripheral devices within an SoC, such as UARTs. TimersGPIO ports Watchdog timers Low-speed ADCs/DACs The APB protocol's simplicity and low power consumption make it an ideal choice for peripheral communication in systems where high performance is not essential but efficiency and ease of implementation are crucial.
The Advanced High-performance Bus (AHB) is a part of the AMBA (Advanced Microcontroller Bus Architecture) protocol suite developed by ARM. AHB is designed for high-performance, high-bandwidth communication within system-on-chip (SoC) architectures. It is widely used for connecting high-speed components, such as processors, memory, and other high-performance peripherals.

2.2 Key Features of the AHB Protocol

High Bandwidth: AHB supports high-speed data transfers, making it suitable for performance-critical components.
Pipelined Operation: AHB uses a pipelined structure to increase the data throughput. This allows multiple operations to overlap, thereby enhancing overall performance.
Burst Transfers: AHB supports burst transfers, which enable efficient data movement by allowing multiple data transfers in a single bus transaction.
Single Clock Edge Operation: Similar to APB, AHB operates on a single clock edge, simplifying the timing analysis.
Multi-master Support: AHB allows multiple bus masters, enabling various components to initiate transactions and share the bus.
Address and Data Phases: AHB transactions consist of address and data phases, but unlike APB, these can be overlapped by pipelining.
Split and Retry Responses: AHB supports split and retry responses, allowing peripherals to signal when they are not ready to complete a transaction, which improves bus utilization.

2.3 Key Signal Lines:

HADDR: Address the bus for the transaction.
HTRANS: Transaction type (IDLE, BUSY, NONSEQ, SEQ)
HWRITE: Indicates whether the operation is a write (1) or read (0).
Size: Indicates the size of the transfer (byte, halfword, or word).
HBURST: Indicates if the transfer is a single transfer or part of a burst.
HPROT: Protection signals indicating the privilege level, security, and cacheability
HWDATA: Write data bus HRDATA: Read data bus HREADY: Indicates that the current transfer can be completed.
HRESP: Response from the slave (OKAY, ERROR, RETRY, SPLIT)
HSEL: Select a signal for the slave.
Operation: Address Phase: The master sets the address, controls, and writes data signals. The HTRANS signal is used to indicate the type of transfer.
Data Phase: The HREADY signal is used to determine whether the slave is ready to complete data transfer. If the HREADY is high, data transfer occurs in the next clock cycle.
Pipelining: While the current data transfer is in progress, the next address phase can begin, allowing continuous data flow and high throughput.
Burst Transfers: AHB supports incrementing and wrapping bursts, which allow multiple data transfers to occur sequentially, reducing the overhead.
advantages. AHB is typically used for high-speed communication within an SoC, such as
Connecting the CPU to memory controllers
Interface with high-speed peripherals like DMA controllers and high-speed I/O ports.
Connecting high-bandwidth components such as graphics controllers and network interfaces
The high performance of the AHB protocol, pipelined operation, and support for burst transfers make it ideal for applications that require efficient and fast data communication within an SoC.

HDL IMPLEMENTATION

The AMBA Protocol is designed using the Verilog HDL Language, and it is verified on the XILINX VIVADO Platform.

RESULTS

The following fig shows the results obtained for the AMBA Protocol using the XILINX VIVADO Tool.

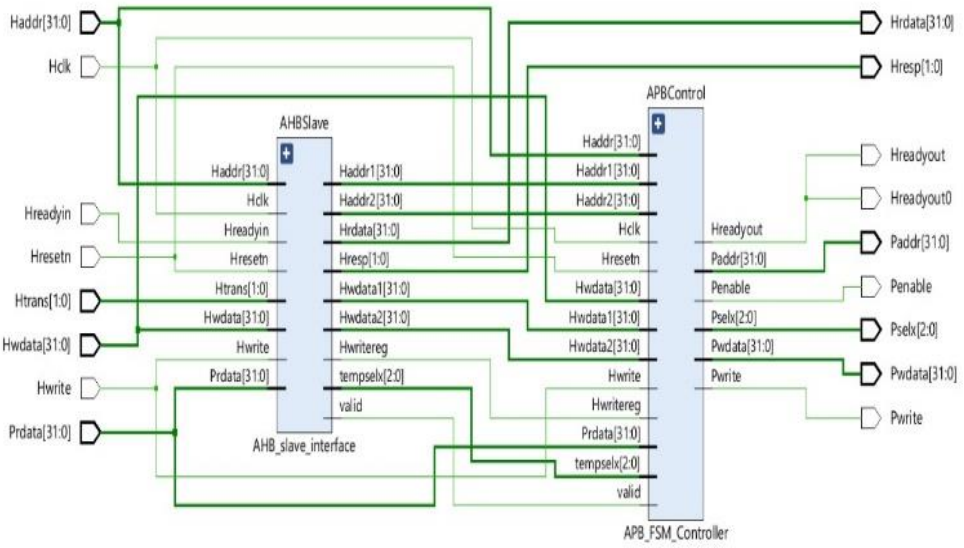

Fig-3 Schematic Diagram of the AMBA BUS

The above fig shows the AMBA Bus schematic diagram, which is verified in the XILINX VIVADO.

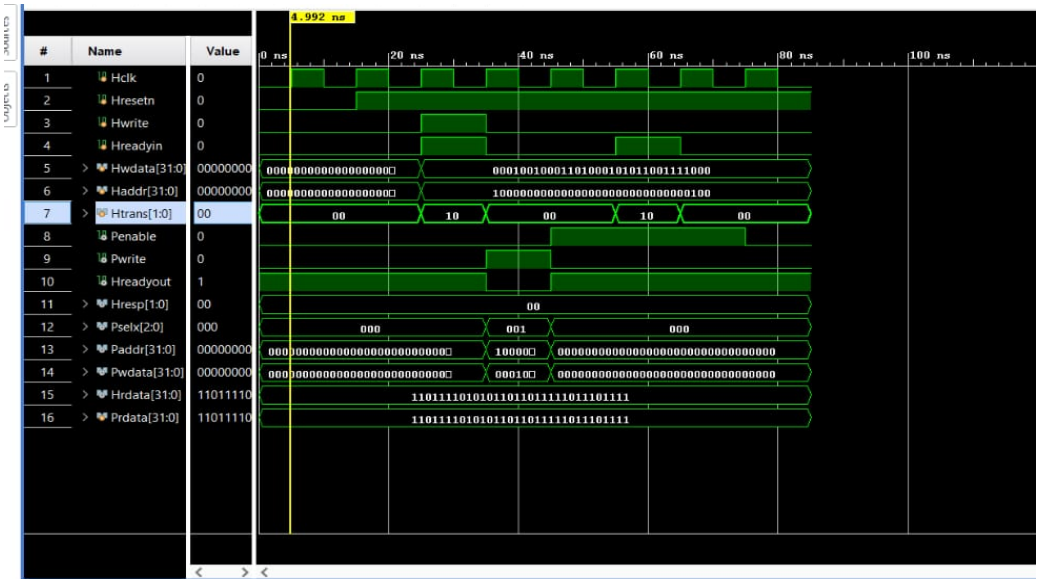

Fig-4 simulation waveform of the AMBA BUS

The above fig shows the simulation waveform of the AMBA bus, which is verified using the XILINX VIVADO Tool.

Conclusion :

Communication protocols like I2C, SPI, UART, and AMBA play a crucial role in VLSI design, enabling efficient data transfer between integrated components on a chip. I2C, developed by Philips Semiconductors, is widely used for low-speed, short-distance communication but has limitations in speed, bus length, and noise susceptibility. To address these drawbacks, protocols like SPI, UART, and CAN have been developed. The AMBA protocol, consisting of AHB and APB, is extensively used in complex VLSI systems. APB is designed for low-power, low-latency peripherals and operates on a single clock edge with a two-phase transaction model. AHB, on the other hand, is designed for high-performance, high-bandwidth communication, with supporting features like burst transfers, split transactions, and multiple bus masters. Selecting the appropriate protocol based on the specific requirements of the VLSI system is crucial for optimal performance, power efficiency, and reliability.

REFERENCES :

1. N.Venkateswara Rao, PV Chandrika, Abhishek Kumar and Sowmya Reddy,“ Design of AMBA based AHB2APB protocol for efficient utilization of AHB and APB”, International Research Journal of Engineering and Technology (IRJET), Volume 07, Issue 03, pp. 2395- 0072, 2020.
2. Vaishnavi R.K, Bindu.S and Sheik Chandbasha, “Design and Verification of APB Protocol by using System Verilog and Universal Verification Methodology”, International Research Journal of Engineering and Technology (IRJET), Volume 06, Issue 06, pp. 2395- 0072, 2019.
3. Kommirisetti Bheema Raju and Bala Krishna Konda, “Design and Verification of AMBA APB Protocol”, Int. Journal of Engineering Research and Application, Volume 7, Issue 1, pp.87-90, 2017.
4. M. Kiran Kumar, Amrita Sajja and Dr. Fazal Noorbasha, “Design and FPGA Implementation of AMBA APB Bridge with Clock Skew Minimization Technique”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Volume 7, Issue 3, pp. 42-45, 2017.
5. Kiran Rawat, Kanika Sahni and Sujata Pandey, “RTL Implementation for AMBA ASB APB Protocol at System on Chip level”, IEEE International Conference on Signal Processing and Integrated Networks (SPIN), 2015.