AHB TO APB BRIDGE DESIGN

PROJECT REPORT BY:

SAURAV RAJ

[21BEC2244]

MAVEN
SILICON
Centre of Excellence in VLSI

VIT – Design Internship

(VIT – DI – Bridge Design)

MAVEN COPORATE TRAINING

Under the guidance of:

Prof.Manjunath N L

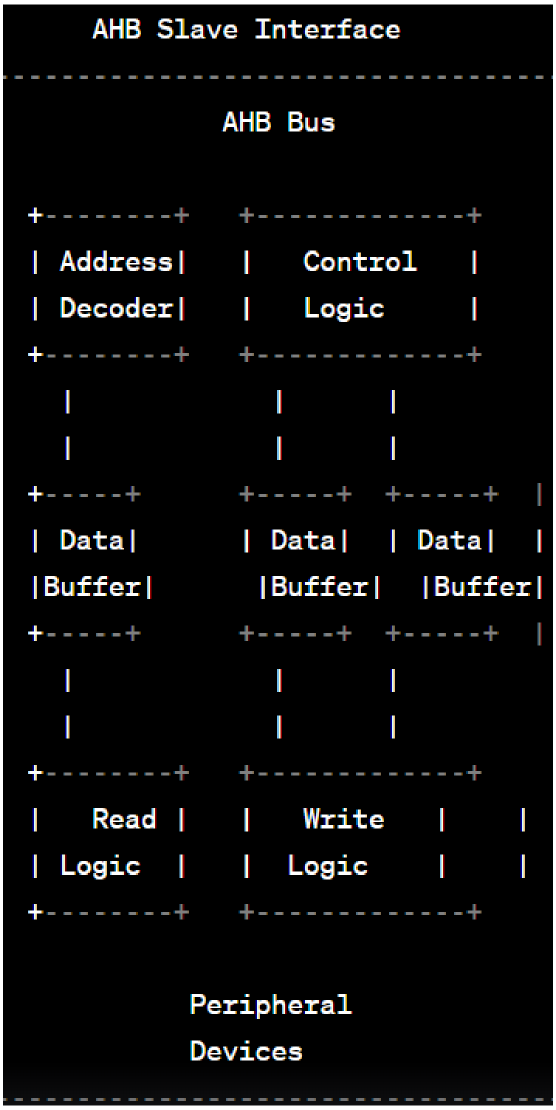
ABSTRACT

The AHB to APB bridge is a critical component in modern System-on-Chip (SoC) designs, enabling communication between the high-performance AHB bus and the low-power, peripheral-oriented APB bus. This bridge serves as an interface to connect various peripherals, including sensors, communication modules, and memory components, to the central processing unit while ensuring efficient data transfer and synchronization.

This abstract outlines the fundamental aspects of designing an AHB to APB bridge, focusing on key elements such as bus protocols, arbitration, data transfer, and clock domain synchronization.

AHB SLAVE INTERFACE

BLOCK DIAGRAM:



```
     AHB Slave Interface
---------------------------------
            AHB Bus


+--------+   +------------+
| Address|   |   Control   |
| Decoder|   |   Logic     |
+--------+   +------------+
   |            |      |
   |            |      |
+-----+      +-----+  +-----+  |
| Data|      | Data|  | Data|  |
|Buffer|      |Buffer|  |Buffer|
+-----+      +-----+  +-----+  |
   |            |      |
   |            |      |
+--------+   +------------+
|   Read |   |   Write   |     |
| Logic  |   |  Logic    |     |
+--------+   +------------+


          Peripheral
          Devices
---------------------------------
```

An AHB (Advanced High-Performance Bus) slave interface is a crucial component in a system-on-chip (SoC) or microcontroller design. It is responsible for connecting various peripheral devices or memory components to the AHB bus, enabling them to communicate with the CPU and other system components. Here is a block diagram of an AHB slave interface with an explanation of its key components:

Explanation of Components:

AHB Bus: The AHB bus is the central communication channel within the system, connecting the CPU, memory, and various peripherals. It consists of address lines, data lines, and control signals.

Address Decoder: The address decoder is responsible for determining which peripheral device or memory location is being accessed on the AHB bus. It compares the address provided by the CPU with predefined address ranges for different peripherals.

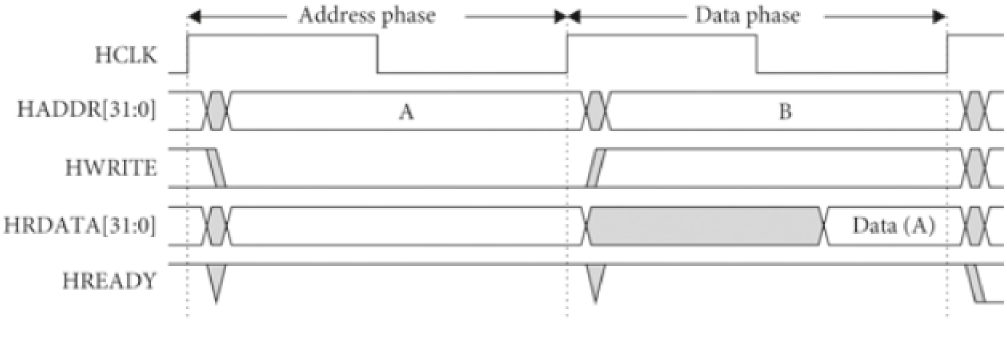
Control Logic: The control logic interprets the control signals on the AHB bus, including signals like HREADY (indicating the readiness of the slave), HSEL (selection of the slave), HTRANS (transaction type), and others. It manages the state of the interface and ensures proper communication between the CPU and the selected peripheral.

Data Buffers: Data buffers are used for temporarily storing data being read from or written to the peripheral device. They ensure data integrity and enable synchronous data transfer between the AHB bus and the peripheral. These buffers are often organized into separate read and write buffers.

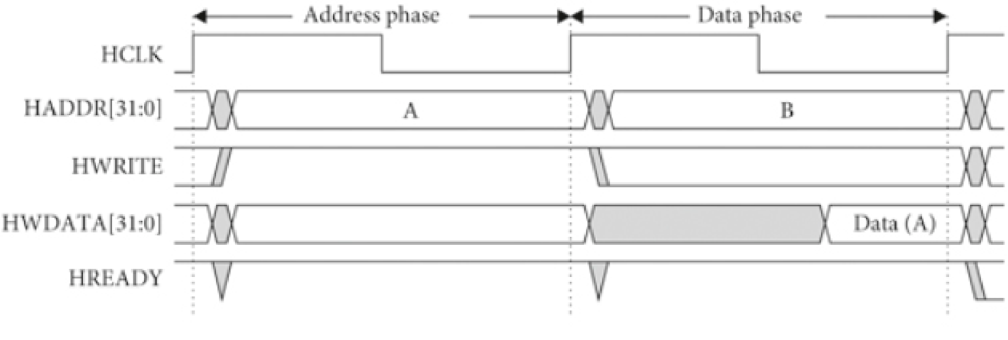
Read and Write Logic: The read logic and write logic components handle read and write transactions on the AHB bus. They manage the data transfer between the AHB bus and the peripheral device, ensuring that data is read from or written to the correct location within the device.

Peripheral Devices: These are the external devices or memory components connected to the AHB slave interface. Examples of peripheral devices include memory modules, communication interfaces, GPIO (General Purpose Input/Output) ports, and various other hardware components.

The AHB slave interface acts as an intermediary between the CPU and the peripheral devices, allowing the CPU to access and control these devices through a standardized bus protocol. It plays a crucial role in ensuring the smooth and efficient operation of an SoC or microcontroller system by managing data transfer, addressing, and control signals between the CPU and the connected peripherals.



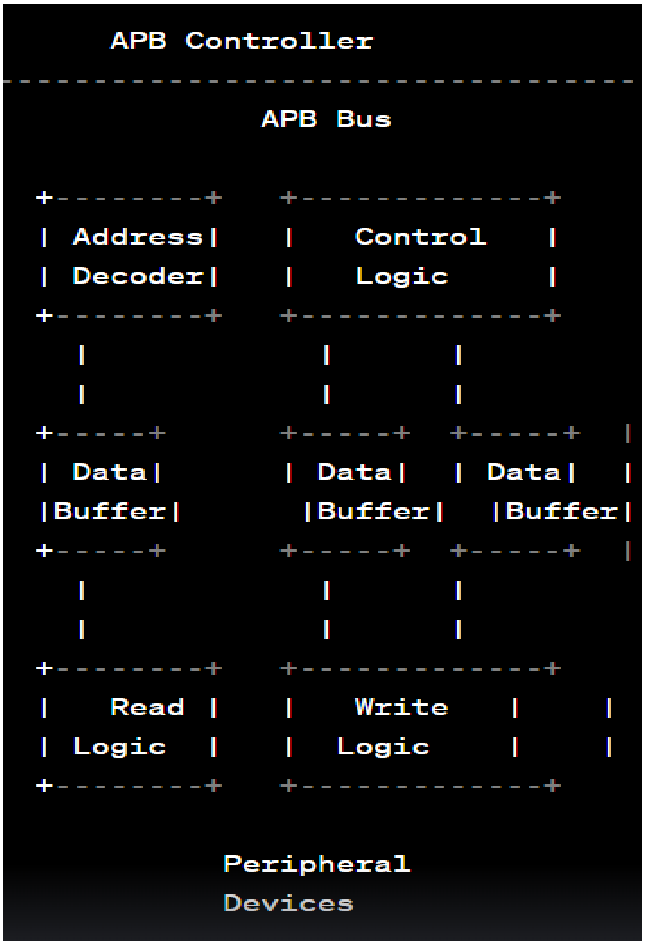
Read transfer



Write transfer

APB CONTROLLER

BLOCK DIAGRAM:



```
     APB Controller
----------------------------------
              APB Bus

 +--------+   +-------------+
 | Address|   |    Control   |
 | Decoder|   |    Logic     |
 +--------+   +-------------+
   |             |      |
   |             |      |
 +-----+      +-----+  +-----+   |
 | Data|      | Data|  | Data|   |
 |Buffer|      |Buffer|  |Buffer|
 +-----+      +-----+  +-----+   |
   |             |      |
   |             |      |
 +--------+   +-------------+
 |   Read |   |    Write    |    |
 | Logic  |   |  Logic      |    |
 +--------+   +-------------+

           Peripheral
           Devices
```

An APB (Advanced Peripheral Bus) controller is a key component in a microcontroller or system-on-chip (SoC) design. It serves as an interface between the CPU and various peripheral devices connected to the APB bus. Below is a block diagram of an APB controller with an explanation of its main components:

Explanation of Components:

APB Bus: The APB bus is the central communication channel within the system, linking the CPU with various peripherals. It includes address lines, data lines, and control signals, similar to the AHB bus but generally designed for lower-speed peripherals.

Address Decoder: The address decoder is responsible for determining which peripheral device or register is being accessed on the APB bus. It compares the address provided by the CPU with predefined address ranges for different peripherals and generates a select signal for the target peripheral.

Control Logic: The control logic interprets control signals on the APB bus, including signals like PREADY (indicating the readiness of the peripheral), PSEL (selection of the peripheral), PTRANS (transaction type), and others. It manages the state of the interface and ensures proper communication between the CPU and the selected peripheral.

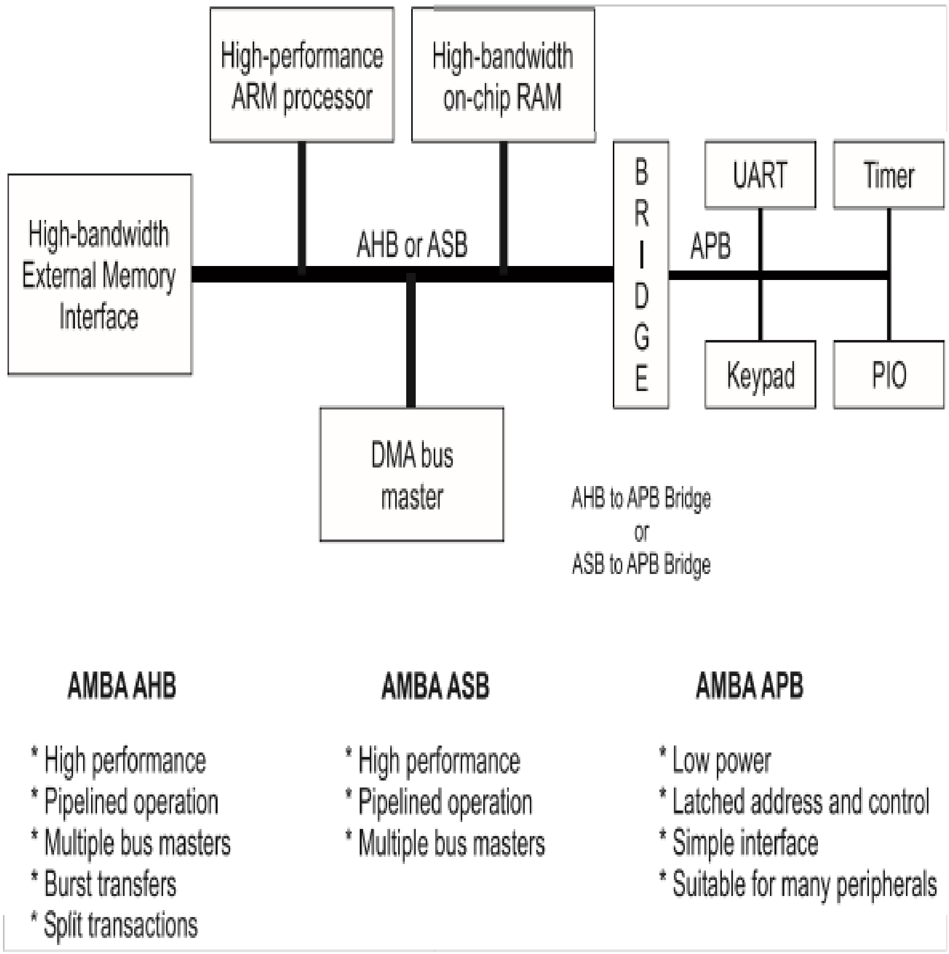
Data Buffers: Data buffers temporarily store data being read from or written to the peripheral device. They ensure data integrity and facilitate synchronous data transfer between the APB bus and the peripheral. These buffers are often organized into separate read and write buffers.

Read and Write Logic: The read logic and write logic components handle read and write transactions on the APB bus. They manage the data transfer between the APB bus and the peripheral device, ensuring that data is read from or written to the correct location within the peripheral.

Peripheral Devices: These are the external devices or registers connected to the APB controller. Peripheral devices may include GPIO ports, UART (Universal Asynchronous Receiver-Transmitter) modules, timers, and other hardware components. These devices are controlled and accessed by the CPU through the APB controller.

The APB controller acts as an intermediary between the CPU and peripheral devices, enabling the CPU to communicate with and control various hardware components using a standardized bus protocol. It plays a crucial role in maintaining efficient and orderly data transfer, addressing, and control signal management within the system. The use of an APB controller allows for a modular and scalable design, making it easier to add or remove peripheral devices from the system without major architectural changes.

Bridge Top(AHB slave or Bridge)



The "Bridge Top (AHB Slave or Bridge)" is a component that often serves as an interface between an AHB (Advanced High-Performance Bus) slave and an APB (Advanced Peripheral Bus) controller in a microcontroller or SoC design. This interface allows for communication between high-performance peripherals connected to the AHB bus and lower-speed peripherals on the APB bus. Here's a block diagram of this setup with an explanation of its key components:

Explanation of Components:

Bridge Top (AHB Slave or Bridge): This component serves as the bridge or interface between the AHB bus and the APB bus. It is responsible for routing and translating transactions between the two buses.

AHB Bus: The AHB bus is the high-performance bus in the system and is used for connecting high-speed peripherals, such as memory modules, advanced communication interfaces, or other high-bandwidth devices.

AHB Slave Interface: This is an AHB slave interface as described in a previous response. It includes an address decoder, control logic, data buffers, read logic, and write logic to manage transactions on the AHB bus.

APB Controller: The APB controller is responsible for controlling and managing transactions on the APB bus, which is typically used for lower-speed peripherals. It includes an address decoder, control logic, data buffers, read logic, and write logic to interface with APB peripherals.

Data Buffers: Data buffers are used to temporarily store data being transferred between the AHB and APB buses. They ensure proper synchronization and data integrity during the transfer.

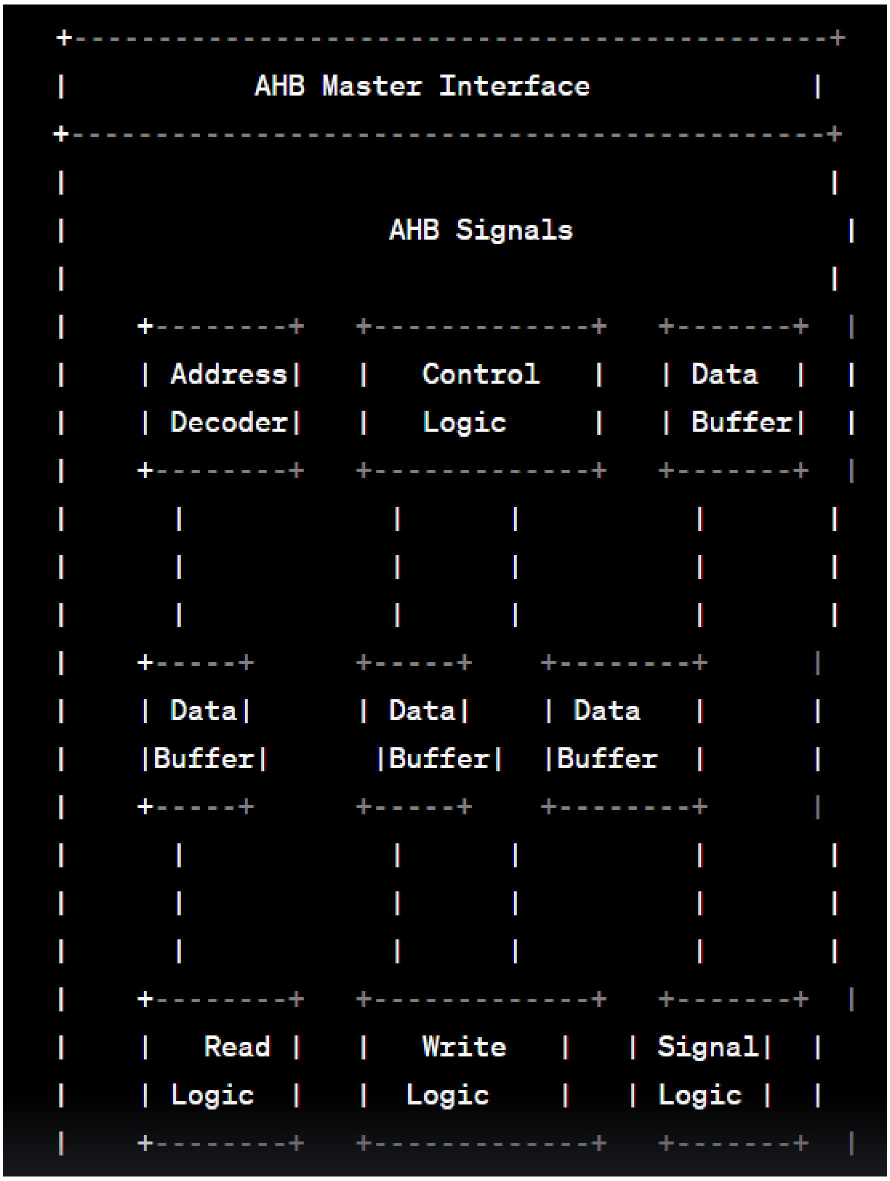
Read and Write Logic: The read logic and write logic components manage read and write transactions on both the AHB and APB buses. They facilitate data transfer and ensure data is correctly read from or written to peripheral devices connected to both buses.

APB Bus: The APB bus is a lower-speed bus typically used for connecting less critical or lower-bandwidth peripherals, such as GPIO ports, timers, or UART modules.

This block diagram shows how the Bridge Top (AHB Slave or Bridge) facilitates communication between high-performance peripherals on the AHB bus and lower-speed peripherals on the APB bus, allowing for a diverse range of peripherals to be connected to a microcontroller or SoC while maintaining efficient data transfer and control.

AHB Master interface

BLOCK DIAGRAM:



```
+-----------------------------------------+
|           AHB Master Interface          |
+-----------------------------------------+
|                                         |
|                 AHB Signals               |
|                                         |
|    +--------+  +------------+  +-------+  |
|    | Address|  |   Control   |  | Data  |  |
|    | Decoder|  |   Logic     |  | Buffer|  |
|    +--------+  +------------+  +-------+  |
|      |          |     |          |      |
|      |          |     |          |      |
|      |          |     |          |      |
|    +-----+     +-----+    +--------+    |
|    | Data|     | Data|    | Data   |    |
|    |Buffer|     |Buffer|  |Buffer  |    |
|    +-----+     +-----+    +--------+    |
|      |          |     |          |      |
|      |          |     |          |      |
|      |          |     |          |      |
|    +--------+  +------------+  +-------+  |
|    |   Read |  |   Write    |  | Signal|  |
|    | Logic  |  |  Logic     |  | Logic |  |
|    +--------+  +------------+  +-------+  |
```

The AHB (Advanced High-Performance Bus) master interface is a critical component in a system-on-chip (SoC) or microcontroller design that allows the CPU or other master devices to communicate with the AHB bus and access various peripherals and memory components. AHB signals are a set of control and data lines used to manage the communication between the AHB master interface and the AHB bus. The tasks of signal write and signal read are fundamental operations in this interface. Below is a block diagram of the AHB Master Interface with a focus on the signal write and signal read tasks, along with an explanation of these components:

Explanation of Components:

AHB Master Interface: The AHB Master Interface is responsible for managing the communication between the CPU or master device and the AHB bus. It includes components for addressing, control, and data transfer.

AHB Signals: These are the set of control and data signals that make up the AHB bus, including address lines, data lines, and various control signals like HREADY (indicating readiness of the slave), HSEL (selecting the slave), HTRANS (indicating transaction type), and others.

Address Decoder: The address decoder is responsible for interpreting the address provided by the CPU and determining which peripheral device or memory location is being accessed on the AHB bus.

Control Logic: The control logic component interprets control signals on the AHB bus, including those related to read and write operations. It manages the state of the interface and ensures proper communication between the CPU and the selected peripheral.

Data Buffer: Data buffers are used to temporarily store data during read and write operations. They ensure data integrity and facilitate synchronous data transfer between the CPU and the AHB bus.

Read Logic: The read logic component manages read transactions on the AHB bus, ensuring that data is correctly read from the selected peripheral device.

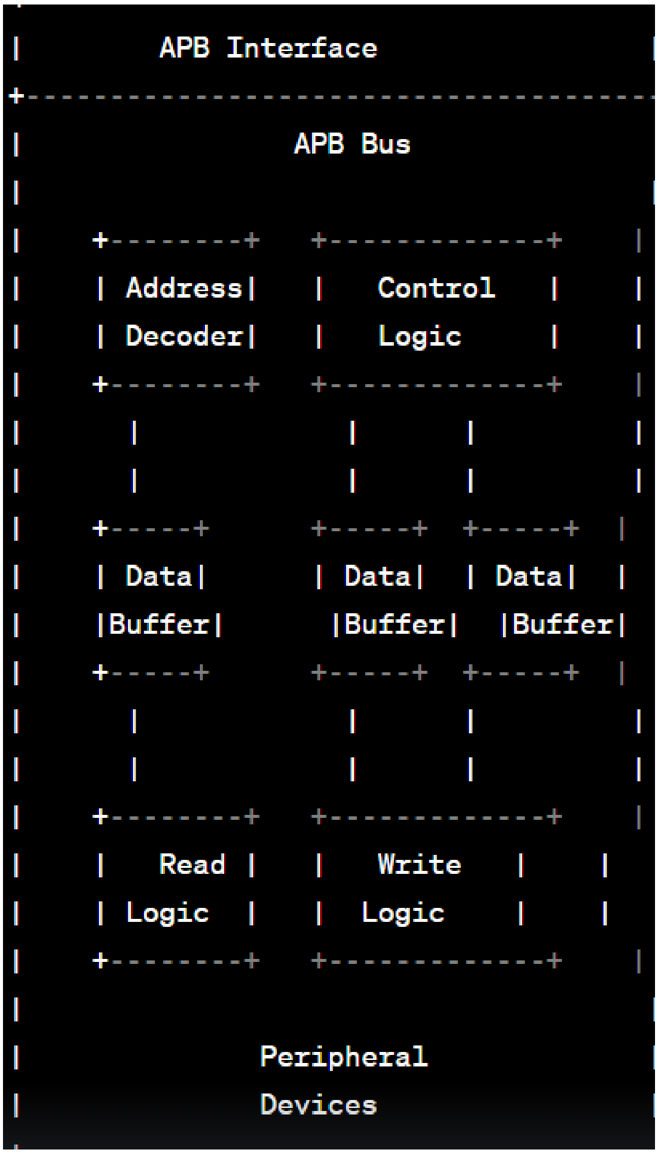
Write Logic: The write logic component handles write transactions, ensuring that data is correctly written to the selected peripheral device.

Signal Logic (Signal Read and Signal Write): These components manage the specific tasks of signaling the initiation of read and write operations to the AHB bus. Signal Read Logic initiates read operations, while Signal Write Logic initiates write operations.

The AHB Master Interface is crucial for ensuring the CPU or other master devices can communicate with and control peripheral devices and memory components on the AHB bus. It manages address decoding, control signals, and data transfer, while the Signal Read and Signal Write Logic components initiate the read and write operations, respectively, ensuring that data is exchanged efficiently and accurately between the master and the AHB bus.

APB INTERFACE

BLOCK DIAGRAM:



```
|         APB Interface
+--------------------------------------
|                APB Bus
|
|     +--------+   +-------------+     |
|     | Address|   |   Control   |     |
|     | Decoder|   |   Logic     |     |
|     +--------+   +-------------+     |
|       |            |      |          |
|       |            |      |          |
|     +-----+      +-----+  +-----+   |
|     | Data|      | Data|  | Data|   |
|     |Buffer|      |Buffer|  |Buffer|
|     +-----+      +-----+  +-----+   |
|       |            |      |          |
|       |            |      |          |
|     +--------+   +-------------+     |
|     |   Read |   |   Write   |    |
|     | Logic  |   |  Logic    |    |
|     +--------+   +-------------+     |
|
|            Peripheral
|            Devices
```

An APB (Advanced Peripheral Bus) interface is a key component in a microcontroller or system-on-chip (SoC) design. It serves as an interface between a CPU or a master device and various peripheral devices connected to the APB bus, facilitating communication and data transfer. Here's a block diagram of an APB interface with an explanation of its key components:

Explanation of Components:

APB Bus: The APB bus is the central communication channel in the system, connecting the CPU or master device with various peripheral devices. It includes address lines, data lines, and control signals.

Address Decoder: The address decoder is responsible for determining which peripheral device or register is being accessed on the APB bus. It compares the address provided by the CPU or master device with predefined address ranges for different peripherals.

Control Logic: The control logic interprets control signals on the APB bus, including signals like PREADY (indicating the readiness of the peripheral), PSEL (selection of the peripheral), PTRANS (transaction type), and others. It manages the state of the interface and ensures proper communication between the CPU or master device and the selected peripheral.

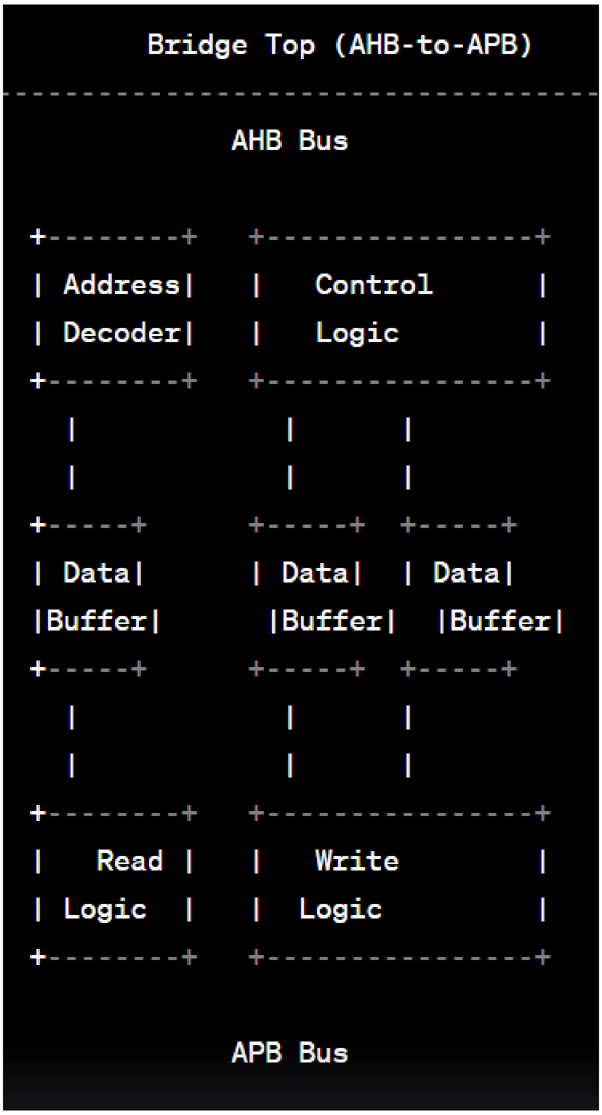
Data Buffers: Data buffers are used for temporarily storing data being read from or written to the peripheral device. They ensure data integrity and enable synchronous data transfer between the APB bus and the peripheral. These buffers are often organized into separate read and write buffers.

Read and Write Logic: The read logic and write logic components handle read and write transactions on the APB bus. They manage the data transfer between the APB bus and the peripheral device, ensuring that data is read from or written to the correct location within the device.

Peripheral Devices: These are the external devices or registers connected to the APB interface. Peripheral devices may include GPIO ports, UART modules, timers, and various other hardware components. These devices are controlled and accessed by the CPU or master device through the APB interface.

The APB interface acts as an intermediary between the CPU or master device and peripheral devices, allowing the CPU or master device to access and control these devices using a standardized bus protocol. It plays a crucial role in ensuring smooth and efficient operation by managing data transfer, addressing, and control signals between the CPU or master device and the connected peripherals. This modular design allows for easy integration of various peripheral devices into the system without major architectural changes.

BRIDGE TOP:



```
        Bridge Top (AHB-to-APB)
------------------------------------
             AHB Bus

 +--------+   +----------------+
 | Address|   |   Control      |
 | Decoder|   |   Logic        |
 +--------+   +----------------+
    |           |      |
    |           |      |
 +-----+      +-----+ +-----+
 | Data|      | Data| | Data|
 |Buffer|      |Buffer|  |Buffer|
 +-----+      +-----+ +-----+
    |           |      |
    |           |      |
 +--------+   +----------------+
 |   Read |   |   Write        |
 | Logic  |   |  Logic         |
 +--------+   +----------------+

             APB Bus
```

A "Bridge Top" typically refers to a component or interface that connects two different bus architectures or protocols, allowing data and control signals to pass between them. One common example is a bridge that connects an AHB (Advanced High-Performance Bus) to an APB (Advanced Peripheral Bus) in a microcontroller or system-on-chip (SoC) design. Below is a block diagram of a Bridge Top with an explanation of its key components:

Explanation of Components:

Bridge Top (AHB-to-APB Bridge): This is the bridge component responsible for connecting the AHB bus to the APB bus, allowing data and control signals to pass between them. It serves as an intermediary between the two buses.

AHB Bus: The AHB bus is the high-performance bus that connects the CPU and high-speed peripherals or memory components. It includes address lines, data lines, and control signals.

Address Decoder: The address decoder interprets the address signals from the AHB bus, determining which peripheral device or register is being accessed.

Control Logic: The control logic interprets control signals on the AHB bus, ensuring proper communication between the CPU and the peripheral devices.

Data Buffers: Data buffers temporarily store data being read from or written to the peripheral devices, facilitating synchronous data transfer between the AHB bus and the peripherals.

Read and Write Logic: These components manage read and write transactions on the AHB bus, ensuring data is correctly read from or written to the peripheral devices.

APB Bus: The APB bus is the lower-speed bus used for connecting lower-speed peripherals. It has its own set of address lines, data lines, and control signals.

The Bridge Top acts as an interface between the AHB bus and the APB bus, allowing the CPU to access and control peripherals on both buses. It manages address decoding, control signals, and data transfer between the two buses, enabling efficient data exchange and control. This modular design allows for easy integration of different peripherals with varying speeds into the system without significant architectural changes.

Bridge Functional Description:

APB output signal generation : The generation of all APB output signals is based on the status of the transfer state machine:

• PWDATA is a registered version of the HWDATA input, which is only enabled during a write transfer. As the bridge is the only bus master on the APB, then it can drive PWDATA continuously.

• PENABLE is only set HIGH during one of three enable states, in the last cycle of an APB transfer. A register is used to generate this output from the next state of the transfer state machine.

- PSELx outputs are decoded from the current transfer address. They are only valid during the read, write and enable states, and are all driven LOW at all other times so that no peripherals are selected when no transfers are being performed.

- PADDR is a registered version of the currently selected address input (HADDR or the address register) and only changes when the read and write states are entered at the start of the APB transfer.

- PWRITE is set HIGH during a write transfer, and only changes when a new APB transfer is started. A register is used to generate this output from the next state of the transfer state machine.

- The APBen signal is used as an enable on the PSEL, PWRITE and PADDR output registers, ensuring that these signals only change when a new APB transfer is started, when the next state is ST_READ, ST_WRITE, or ST_WRITEP.

AHB output signal generation :

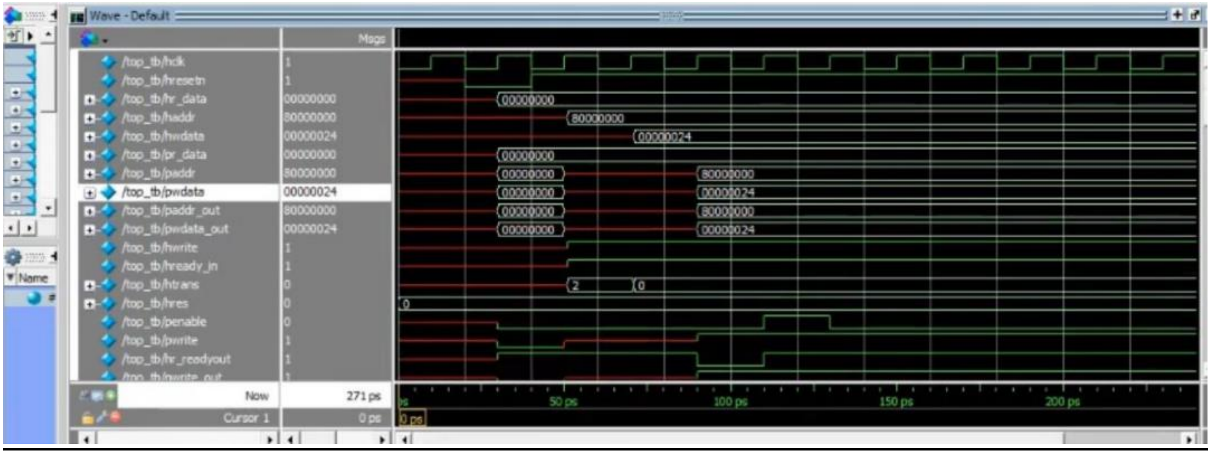
HRDATA is directly driven with the current value of PRDATA. APB slaves only drive read data during the enable phase of the APB transfer, with PRDATA set LOW at all other times, so bus clash is avoided on HRDATA (assuming OR bus connections for both the AHB and APB read data buses).

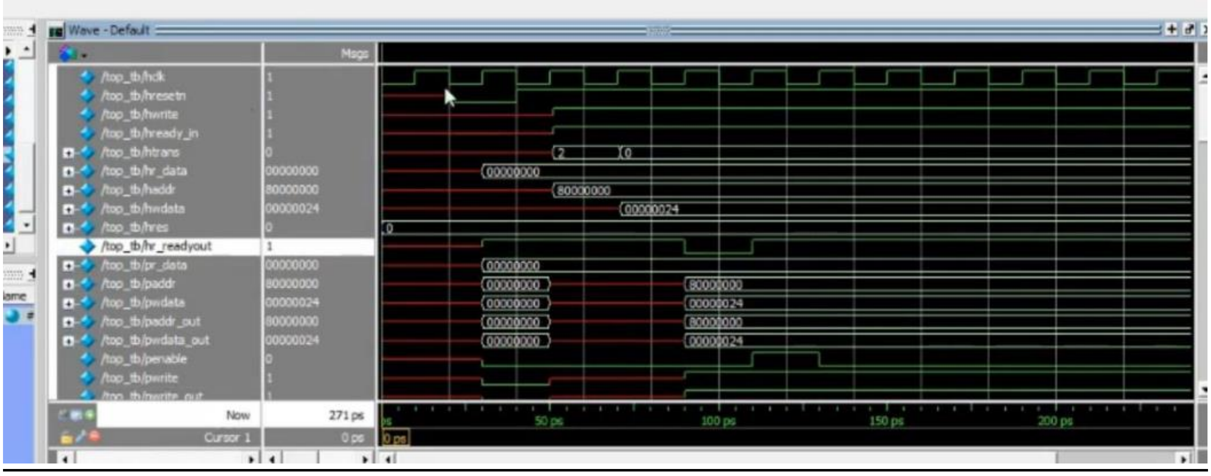
- HREADYout is driven with a registered signal to improve the output timing. Wait states are inserted by the APB bridge during the ST_READ and ST_WRITEP states, and during the ST_WENABLEP state when the next transfer to be performed is a read.

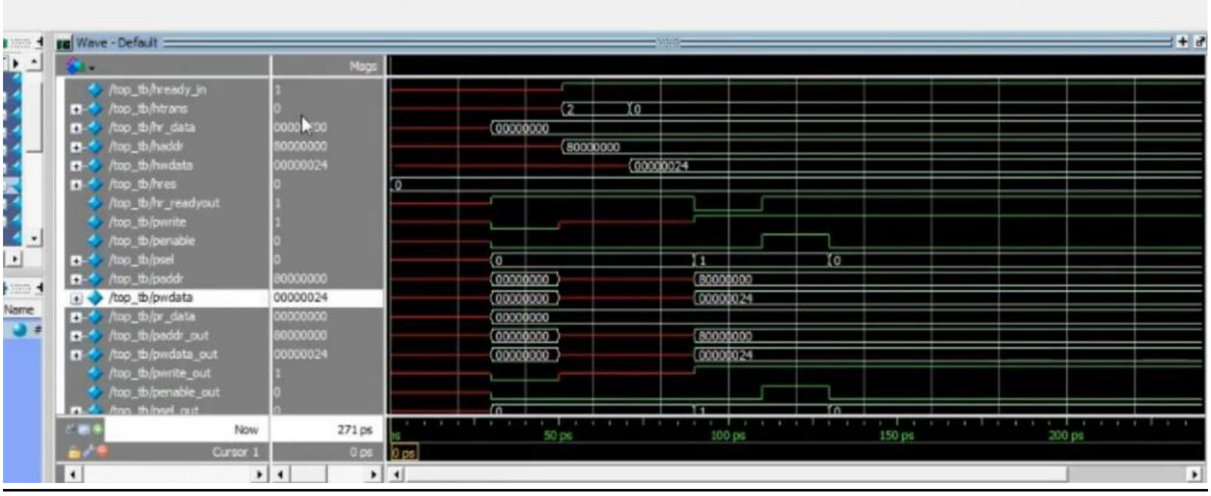
- HRESP is continuously held LOW, as the APB bridge does not generate SPLIT, RETRY or ERROR responses.

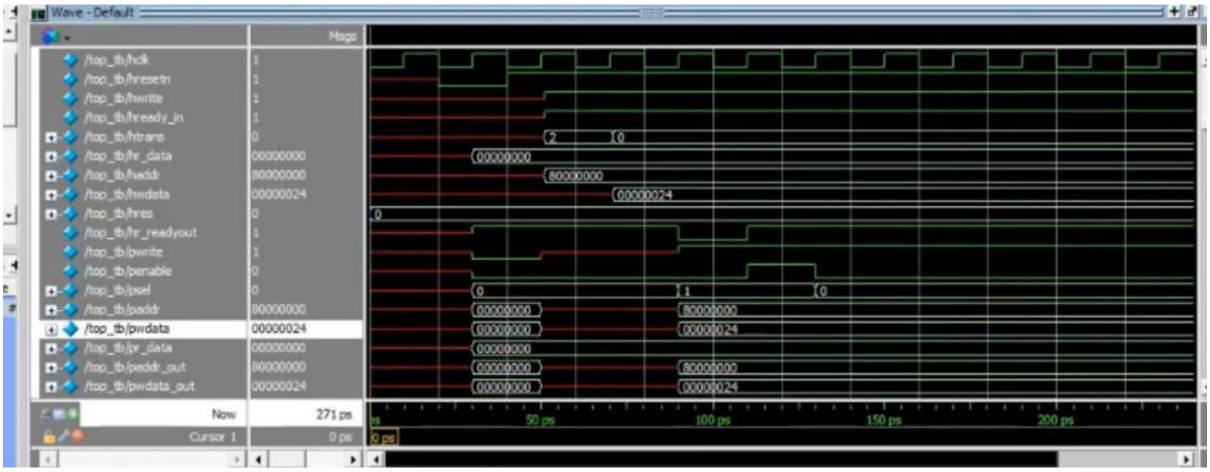
SIMULATION:

APB BRIDGE SIGNAL:

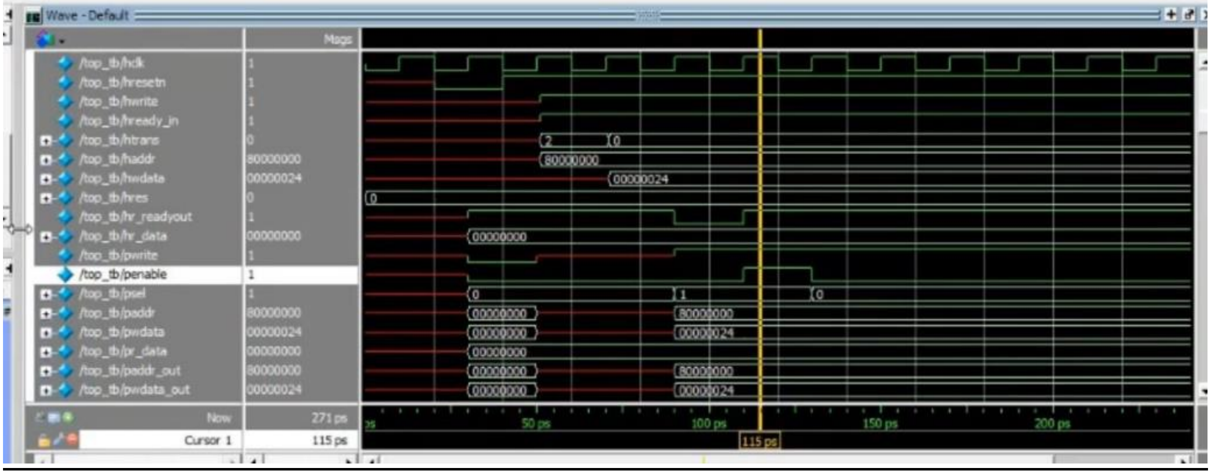






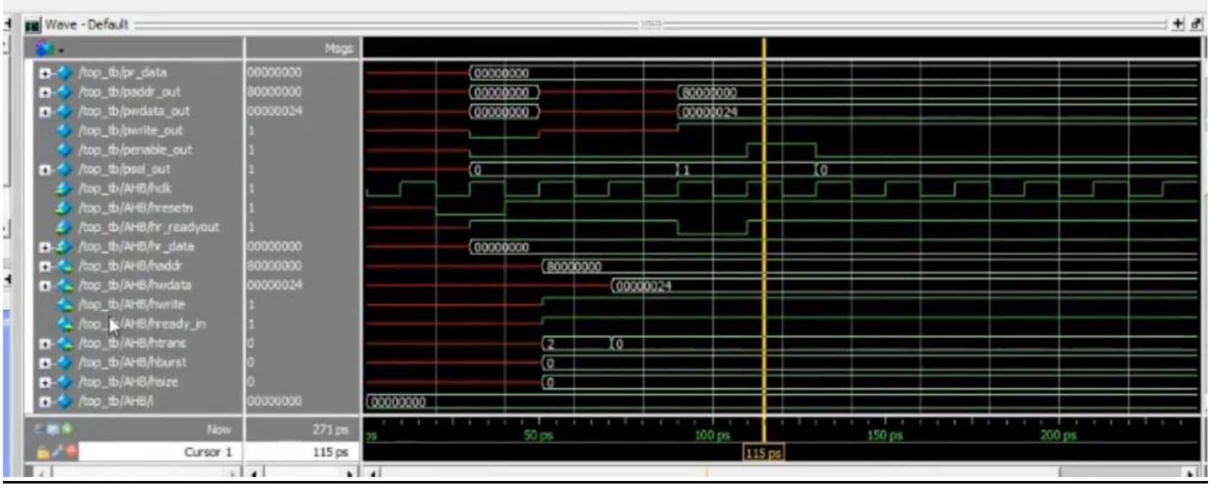


```
Wave - Default
                              Msgs
/top_tb/hclk                  1
/top_tb/hresetn               1
/top_tb/hwrite                1
/top_tb/hready_in             1
/top_tb/htrans                0
/top_tb/hr_data               00000000
/top_tb/haddr                 80000000
/top_tb/hwdata                00000024
/top_tb/hres                  0
/top_tb/hr_readyout           1
/top_tb/pwrite                1
/top_tb/penable               0
/top_tb/psel                  0
/top_tb/paddr                 80000000
/top_tb/pwdata                00000024
/top_tb/pr_data               00000000
/top_tb/paddr_out             80000000
/top_tb/pwdata_out            00000024
Now                           271 ps
Cursor 1                      0 ps
```

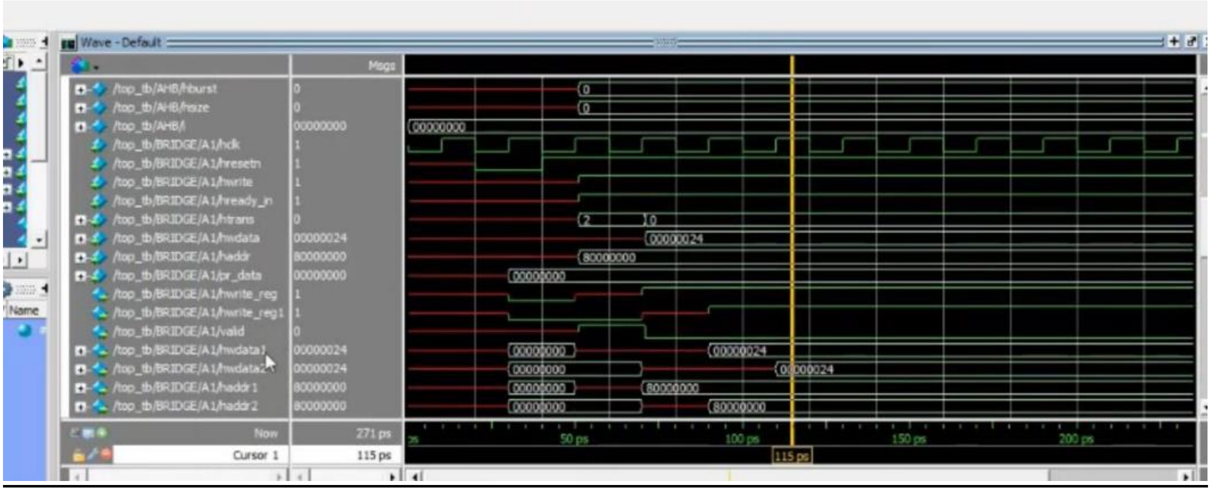


```
Wave - Default
                              Msgs
/top_tb/hclk                  1
/top_tb/hresetn               1
/top_tb/hwrite                1
/top_tb/hready_in             1
/top_tb/htrans                0
/top_tb/haddr                 80000000
/top_tb/hwdata                00000024
/top_tb/hres                  0
/top_tb/hr_readyout           1
/top_tb/hr_data               00000000
/top_tb/pwrite                1
/top_tb/penable               1
/top_tb/psel                  1
/top_tb/paddr                 80000000
/top_tb/pwdata                00000024
/top_tb/pr_data               00000000
/top_tb/paddr_out             80000000
/top_tb/pwdata_out            00000024
Now                           271 ps
Cursor 1                      115 ps
```

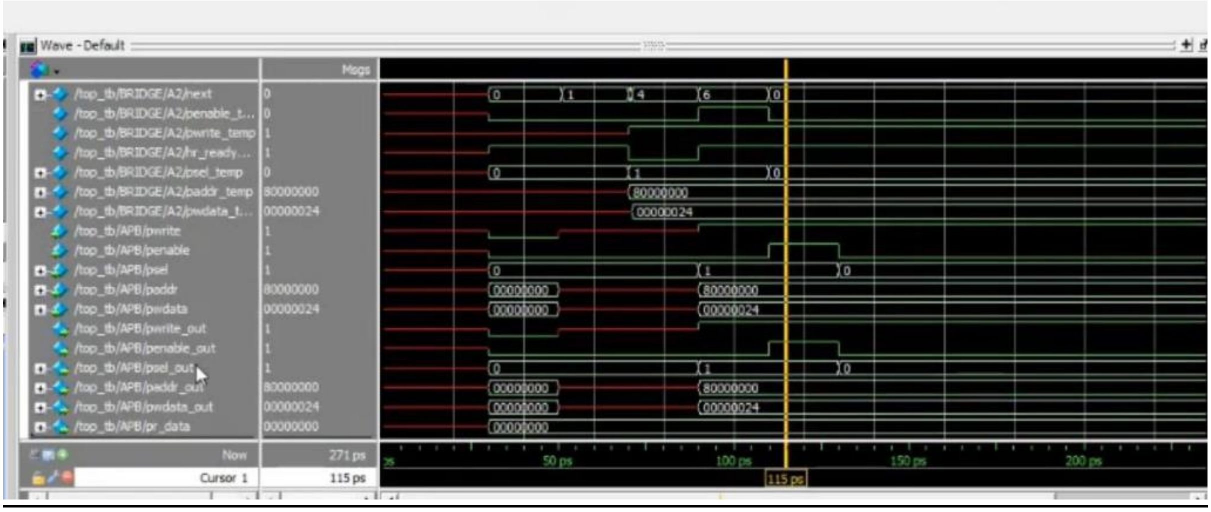
AHB BRIDGE:



```
Wave - Default
                              Msgs
/top_tb/pr_data               00000000
/top_tb/paddr_out             80000000
/top_tb/pwdata_out            00000024
/top_tb/pwrite_out            1
/top_tb/penable_out           1
/top_tb/psel_out              1
/top_tb/AHB/hclk              1
/top_tb/AHB/hresetn           1
/top_tb/AHB/hr_readyout       1
/top_tb/AHB/hr_data           00000000
/top_tb/AHB/haddr             80000000
/top_tb/AHB/hwdata            00000024
/top_tb/AHB/hwrite            1
/top_tb/AHB/hready_in         1
/top_tb/AHB/htrans            0
/top_tb/AHB/hburst            0
/top_tb/AHB/hsize             0
/top_tb/AHB/i                 00000000
Now                           271 ps
Cursor 1                      115 ps
```



	Msgs
/top_tb/AHB/hburst	0
/top_tb/AHB/hsize	0
/top_tb/AHB/i	00000000
/top_tb/BRIDGE/A1/hclk	1
/top_tb/BRIDGE/A1/hresetn	1
/top_tb/BRIDGE/A1/hwrite	1
/top_tb/BRIDGE/A1/hready_in	1
/top_tb/BRIDGE/A1/htrans	0
/top_tb/BRIDGE/A1/hwdata	00000024
/top_tb/BRIDGE/A1/haddr	80000000
/top_tb/BRIDGE/A1/pr_data	00000000
/top_tb/BRIDGE/A1/hwrite_reg	1
/top_tb/BRIDGE/A1/hwrite_reg1	1
/top_tb/BRIDGE/A1/valid	0
/top_tb/BRIDGE/A1/hwdata1	00000024
/top_tb/BRIDGE/A1/hwdata2	00000024
/top_tb/BRIDGE/A1/haddr1	80000000
/top_tb/BRIDGE/A1/haddr2	80000000
Now	271 ps
Cursor 1	115 ps



	Msgs
/top_tb/BRIDGE/A2/next	0
/top_tb/BRIDGE/A2/penable_t...	0
/top_tb/BRIDGE/A2/pwrite_temp	1
/top_tb/BRIDGE/A2/hr_ready...	1
/top_tb/BRIDGE/A2/psel_temp	0
/top_tb/BRIDGE/A2/paddr_temp	80000000
/top_tb/BRIDGE/A2/pwdata_t...	00000024
/top_tb/APB/pwrite	1
/top_tb/APB/penable	1
/top_tb/APB/psel	1
/top_tb/APB/paddr	80000000
/top_tb/APB/pwdata	00000024
/top_tb/APB/pwrite_out	1
/top_tb/APB/penable_out	1
/top_tb/APB/psel_out	1
/top_tb/APB/paddr_out	80000000
/top_tb/APB/pwdata_out	00000024
/top_tb/APB/pr_data	00000000
Now	271 ps
Cursor 1	115 ps

CONCLUSION:

Designing an AHB to APB bridge is a complex and crucial endeavor in modern System-on-Chip (SoC) design. This bridge serves as the linchpin between the high-performance AHB bus and the low-power APB bus, facilitating communication between the central processing unit and peripheral devices. This conclusion summarizes the key takeaways from the design process and highlights the significance of a well-designed AHB to APB bridge.

In conclusion, the successful design of an AHB to APB bridge is critical for the overall performance, power efficiency, and reliability of an SoC. It acts as the bridge between high-performance and low-power components, allowing them to coexist harmoniously within the same system. The efficient translation of bus protocols, intelligent arbitration, and robust error handling are key elements that must be addressed. A well-designed bridge empowers modern SoCs to deliver optimal performance and functionality while conserving power, making it an indispensable component in complex embedded systems.