Design and Implementation of Efficient FSM for AHB Master

Contents

Abstract:

Nowadays industries are more focused on the development of System On-Chip (SOC) which integrates all the electronic components on a single IC with reusable IP cores. As per De Moore's law the integration of transistors goes on increasing, hence it is required to provide efficient lossless on chip communication for these reusable IP cores. The AMBA protocol from ARM Ltd. is designed to overcome this issue. The AMBA AHB protocol provides the high bandwidth data bus for the IP cores on SOC devices. In this document we design FSM for AHB Master interface in AMBA AHB protocol using verilog .

Key words: Advanced High-Performance Bus, AHB Master, System on Chip.

Introduction:

Advanced Microcontroller Bus Architecture (AMBA)is a protocol that is used as an open standard, on chip interconnect specification for the connection and management of functional blocks in a system-on-chip (SoC).

1. AMBA assists the progress of right-first-time development of multiprocessor designs with a large number of controllers & peripherals.

2. The Advanced Microcontroller Bus Architecture (AMBA) has the ability to re-use designs and here it means it has the ability to re-use IP. IP re-use in today's technology is an important factor in reducing the development costs and timescales for System-on-chip (SoC).

3.AMBA is a standard interface specification that makes sure of the compatibility between IP components provided by different design teams or vendors.

The worldwide reception of AMBA specifications all over the semiconductor industry has driven a comprehensive market in third party IP products and tools to support the development of AMBA based systems.

Overview of AMBA Buses:

There are three different buses defined within the AMBA specification.

Advanced High Performance Bus (AHB),

Advanced System Bus (ASB), and

Advanced Peripheral Bus (APB).

This document mainly deals with AMBA AHB and particularly AHB master .

Advanced High Performance Bus (AHB)

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.

Advanced System Bus (ASB)

The AMBA ASB is for high-performance system modules. AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required. ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions.

Advanced Peripheral Bus (APB)

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.

A typical AMBA-based microcontroller

An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access(DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high performance bus is a bridge to the lower bandwidth APB, where most of the peripheral devices in the system are located.

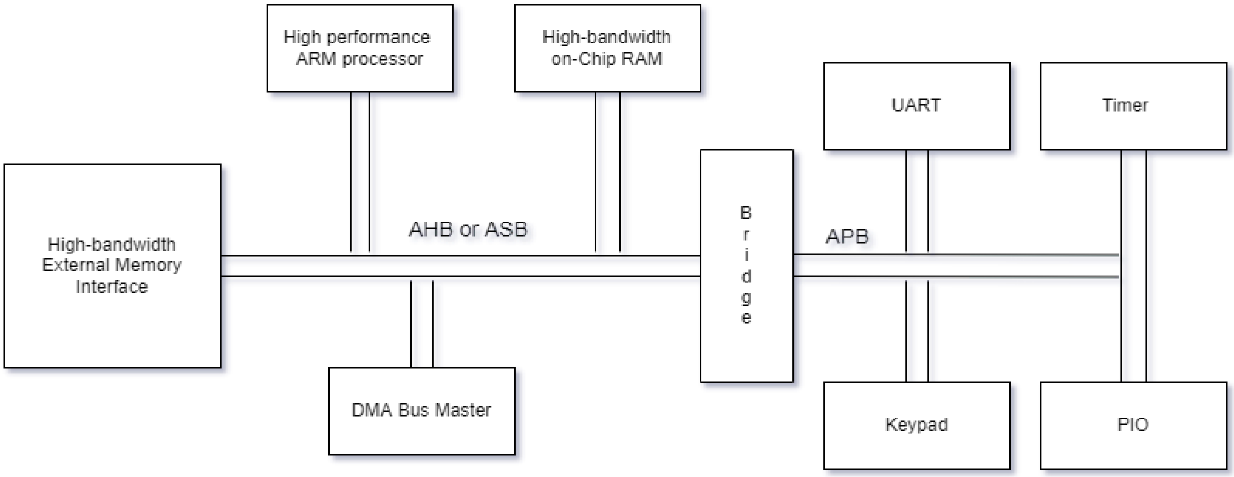
High performance ARM processor
High-bandwidth on-Chip RAM
UART
Timer
High-bandwidth External Memory Interface
AHB or ASB
B r i d g e
APB
DMA Bus Master
Keypad
PIO

Figure 1 . Typical AMBA System.

AHB INTERCONNECTION :

AHB is a new generation of AMBA bus which is intended to address the requirements of high performance synthesizable designs. It is a high performance system bus that supports multiple bus masters and provides high bandwidth operation. AMBA AHB implements the features required for high-performance, high clock frequency systems including:

- Burst transfers
- Single-cycle bus master handover
- Wider data bus configurations (64/128, up to1024 bits).
- SEQ, NON-SEQ, BUSY IDLE transfer types
- Address decoding

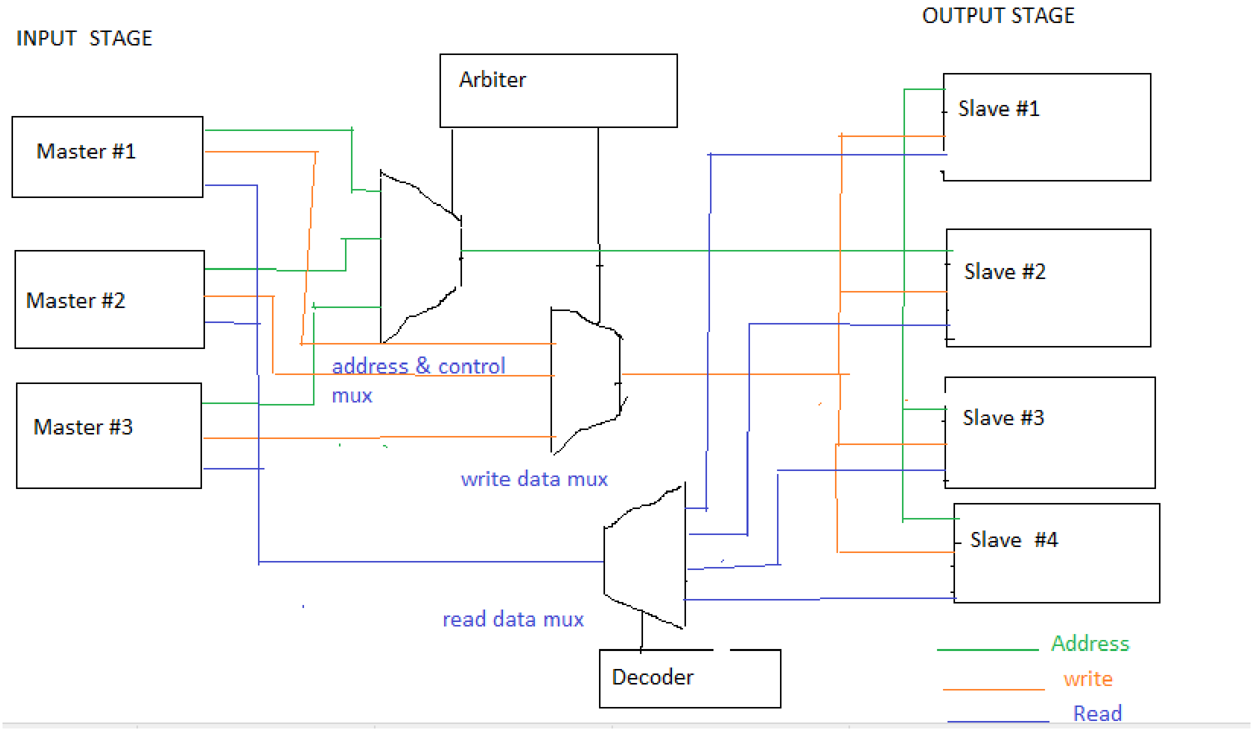

Figure 2. AHB Interconnection diagram.

An AMBA AHB design is having following components:

AMBA AHB MASTER: An AMBA AHB bus master is able to initiate read and write operations by making use of an address and control information. Only one bus master at a time is allowed to actively use the bus.

AMBA AHB SLAVE: An AMBA AHB bus slave responds to read and write operation initialized by master within a given address space range. The bus slave signals back to active master about the success, failure or waiting of the data transfer.

AMBA AHB ARBITER: An AMBA AHB bus arbiter gives an assurance that only one bus master at a time is allowed to initiate the data transfers.Even though the arbitration scheme is fixed, any arbitration scheme can be used like Round Robin, Fair Chance etc. depending on the application requirement.

AMBA AHB DECODER: The AMBA AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.

Figure shows the working principle of AHB Bus. Before starting the AMBA AHB transfer, the bus master must have to be granted access to the bus. In this process first of all the master asserts a request signal to an arbiter. Now the arbiter will indicate when the master will get the grant of the bus. This decision of granting the access to the bus is achieved using some arbitration mechanism like priority based or round robin mechanism etc. A granted bus master then starts the AHB transfer by first driving an address and control signals. These address and control signals provide information about an address, direction and width of the transfer, burst transfer information if the transfer forms the part of the burst.

AHB MASTER INTERFACE: AHB master takes the appropriate action according to the Transfer response signals, and sends the required address, control and data signals to the slave and AHB master is responsible for sending the address and control signal to the slave device weather the operation is read or write. The AHB master output depends upon the hready and hresp [1:0] provided by the slave device.The hrdata carries the read data signal from the slave device and hwdata carries the data to be written to the slave device by the master device. hwrite signal tells whether the read operation is being carried or the write operation is taking place. The data width and the address width are 32 bit in size. The hsize [2:0] tells the size of the transfer, and the type of data transfer is given by the htrans [1:0] signal. The hburst [2:0] signal gives the information about the burst transfer, i.e. single type, increment by 4, increment by 8, increment by 16.

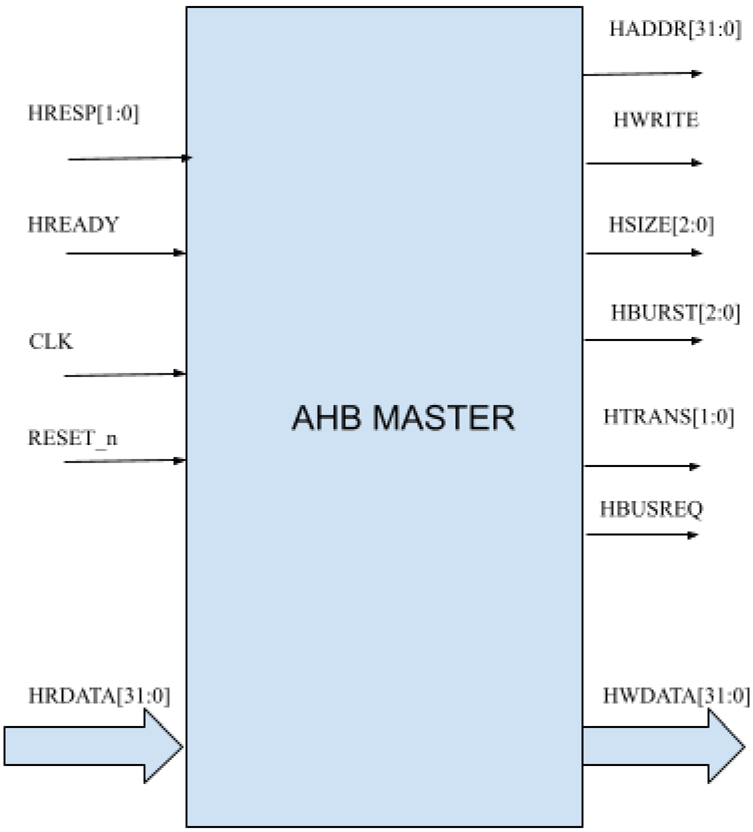

Figure 3. AHB Master interface diagram

Operation :

The master starts a transfer by driving the address and control signals. These signals provide information about the address, direction, width of the transfer, and indicate if the transfer forms part of a burst.

Transfers can be:

- single
- incrementing bursts that do not wrap at address boundaries
- wrapping bursts that wrap at particular address boundaries.

The write data bus moves data from the master to a slave, and the read data bus moves data from a slave to the master.

Every transfer consists of:

Address phase one address and control cycle.

Data phase one or more cycles for the data.

A slave cannot request that the address phase is extended and therefore all slaves must be capable of sampling the address during this time. However, a slave can request that the master extends the data phase by using HREADY. This signal, when LOW, causes wait states to be inserted into the transfer and enables the slave to have extra time to provide or sample data. The slave uses HRESP to indicate the success or failure of a transfer.

AHB Master Signals:

All signals are prefixed with the letter H.

- **HBUSREQ:** this signal contains a width of 1-bit and it is driven by an arbiter. This signal is used to initiate data transfer.
- **HCLK**: This signal contains a width of 1-bit and it is driven by an external clock source. The data can be transferred at each rising edge of HCLK.
- **HRESET_n :** the bus reset signal is active low and resets the system and the bus . This is the only active low AHB signal.
- **HADDR [32:0]** : Width of this signal is 32-bit and driven by the master to assign address.
- **HTRANS**: Width of this signal is 2-bit and driven by master. It indicates the type of the current transfer happening.

HTRANS [1:0]	Type	Description
b’00	IDLE	indicates that no data transfer is required. the master uses IDLE transfer when it does not want to perform a data transfer slave must always provide a zero wait state OKAY response to idle transfer. the transfer must be ignored by slave
b’01	BUSY	The busy transfer type enables the master to insert idle cycles in the middle of a burst . only undefined length burst can have a busy transfer as the last cycle of a burst slave must always provide a zero wait state OKAY response to idle transfer.
b’10	NON-SEQ	Indicates a single transfer or the first transfer of a burst. The address and control signals are unrelated to the previous transfer.
b’11	SEQ	The remaining transfers in a burst are SEQUENTIAL and the address is related to the previous transfer. The control information is identical to the previous transfer.

- **HWRITE**: : Width of this signal is 1-bit it is also driven by the master.
  HWRITE controls the direction of data transfer to or from the master. Therefore, when:
  HWRITE is HIGH, it indicates a write transfer and the master broadcasts data on the write data bus, HWDATA[31:0]
  HWRITE is LOW, a read transfer is performed and the slave must generate the data on the read data bus, HRDATA[31:0].

- **HSIZE** : Width of this signal is 3-bit and driven by master . It indicates the size of the transfer.

HSIZE[2:0]	Size (bits)	Description
3’b000	8	byte
3’b001	16	half-word
3’b010	32	word
3’b011	64	double-word
3’b100	128	4-word line
3’b101	256	8-word line
3’b110	512	-
3’b111	1024	-

- **HBURST** : Width of this signal is 3-bit, and driven by master. It indicates if the transfer forms part of a burst.

HBURST [2:0]	Type	Description
b’000	SINGLE	single burst
b’001	INCR	increment burst of undefined length
b’010	WRAP4	4-beat wrapping burst
b’011	INC4	4 beat incrementing burst
b’100	WRAP8	8-beat wrapping burst
b’101	INC8	8-bit incrementing burst

b’110	WRAP16	16-beat wrapping burst
b’111	WRAP16	16-beat wrapping burst

- **HWDATA [31:0]** : Width of this signal is 32-bit and driven by the master. It is used to transfer data from the master to the bus slaves during write operations.
- **HRDATA [31:0]** : Width of this signal is 32-bit and driven by slave. It is used to transfer data from slaves to the bus master during read operations.
- **HREADY** : Width of this signal is 1-bit and driven by slave. When high the HREADY signal indicates that transfer has finished on the bus. This signal may be driven low to extend a transfer.
- **HRESP**:- Width of this signal is 2-bit and driven by slave. It provides additional information on the status of a transfer.

HRESP	Response	Description
0	OKAY	The transfer has completed successfully. the hready signal indicates whether the transfer is pending or complete
1	ERROR	An error has occurred during the transfer. the error signal condition must be signaled to master so that it is aware of the transfer has been unsuccessfully A two cycle response is required for an error condition with hready being asserted in the second cycle

Finite state machine for AHB Master

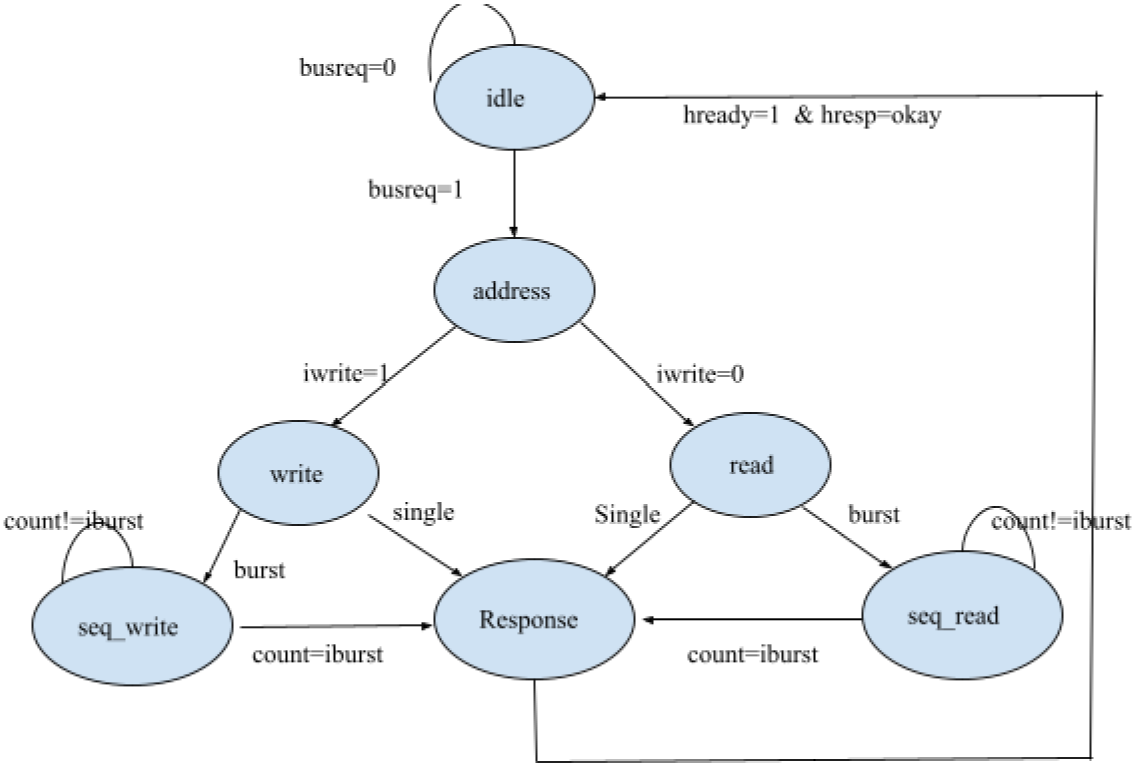

Figure 5. FSM for AHB Master

IDLE: Initially the Master will be in the IDLE states during this state no data transfer takes place. Buses are free and masters are not accessing any buses during this state. The Master goes to ADDRESS state from the IDLE state when it wants to perform the data transfer. The IDLE state can be changed to ADDRESS state by asserting the hbusreq signal.

ADDRESS : In this state the master waits for the iwrite signal to be obtained; it then jumps to WRITE or READ states according to the input write given through the test bench. If the write signal is 1 then write mode is activated and master will generate the output hwrite as 1. At this state the address and control signals are also given to the master through the test bench. The address signal is iaddr, and the control signals being isize, itrans, iburst, .The hready signal and hresp signal from the slave is also checked in this state.

WRITE: In this state the master will check the control signals trans, size, burst. If the burst of the transfer is 00 which indicates single, the transfer is of single burst and the state will change to RESPONSE state. The itrans signal value for the start of each transfer is 10 which indicates non sequential. If the burst signal is incr4, incr8, incr16 the next state will be SEQ_WRITE. In general we can say that at this state the master interface will check whether it needs to transfer a single transfer or a sequential incrementing transfer.

SEQ_WRITE: This state is the sequential write state. The master comes to this state from the WRITE state if the burst is an incrementing burst. The master will be in this state until it finishes the number of bursts to be transferred. For achieving this a counter is generated inside the state which counts the number of bursts sent by the master when the count reaches values the master changes to RESPONSE state.

RESPONSE: Mater will be in this state, when the transfer is single or it is the last transfer of the burst. The final data is written when the hready is high and hresp from the slave is OKAY signal

READ: In this state the master will check the control signals trans, size, burst. If the burst of the transfer is 00 which indicates single, the transfer is of single burst and the state will change to RESPONSE state. The itrans signal value for the start of each transfer is 10 which indicates non sequential. If the burst signal is incr4, incr8, incr16 the next state will be SEQ_READ. In general we can say that at this state the master interface will check whether it needs to transfer a single transfer or a sequential incrementing transfer.

SEQ_READ: This state is the sequential write state. The master comes to this state from the WRITE state if the burst is an incrementing burst. The master will be in this state until it finishes the number of bursts to be transferred. For achieving this a counter is generated inside the state which counts the number of bursts sent by the master when the count reaches values the master changes to RESPONSE state.

Verification goals :

- Minimum 32-bit transfer size is recommended. It supports only 32 bit size .
- During burst transfer iwrite must be constant .
- When write=1 , it should perform a write operation. if write=0 it performs a read operation .
- If hready is high indicates that read and write transfer is completed and if it's low extend the transfer (insert cycles).
- It supports idle , non-sequential , sequential transfer.

Simulation Result:

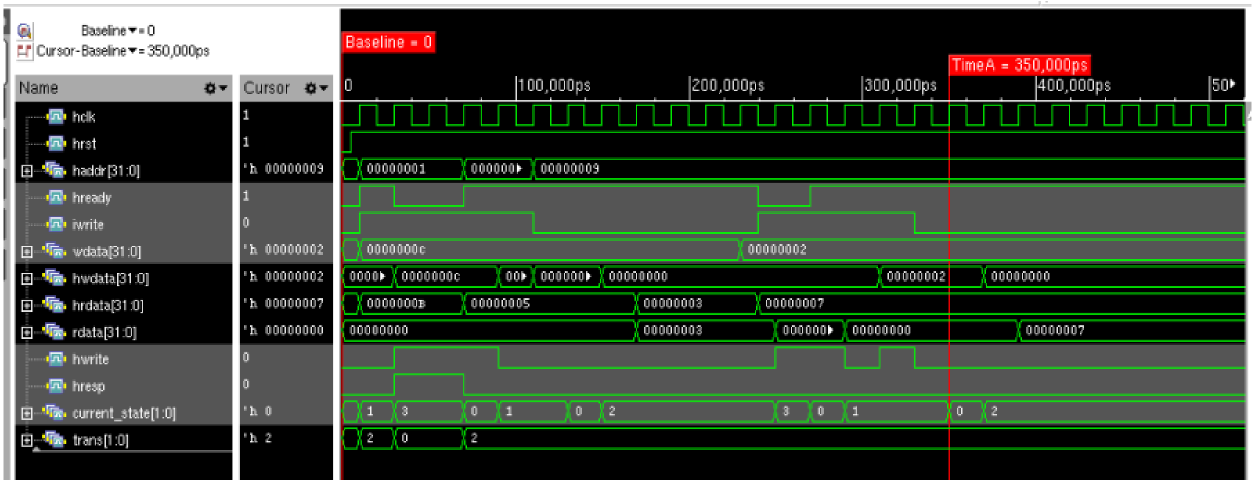

Figure 6. Simulation Waveform of AHB Master for Read and Write Operations

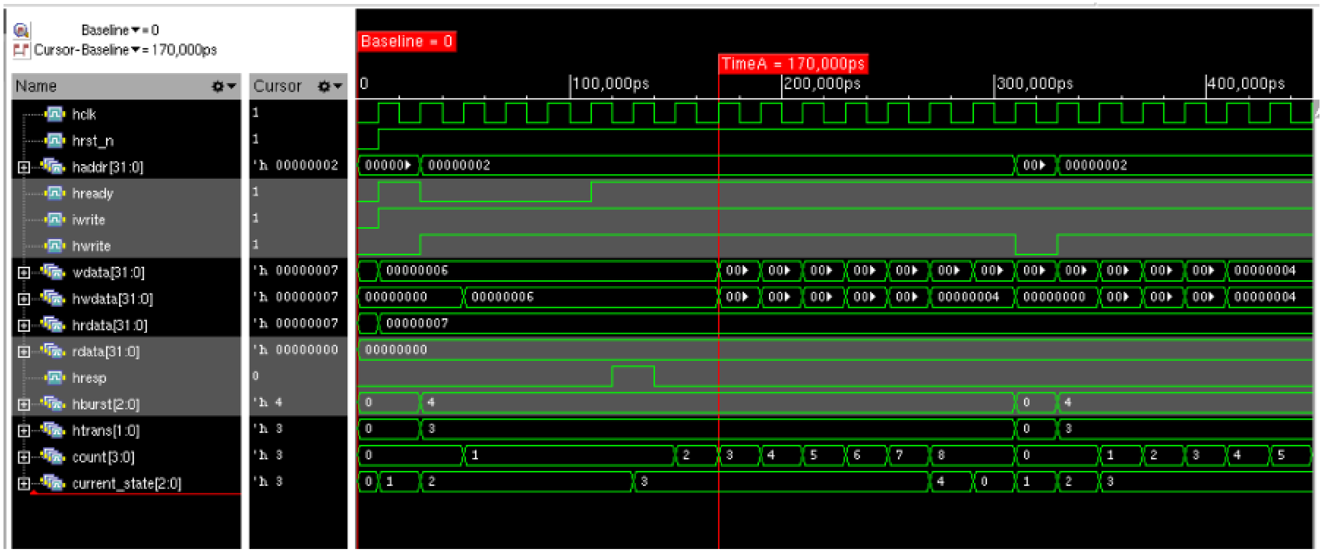

Figure 7. Simulation waveform for burst transfer.