AHP AMBA PROTOCOL

design of Master block

MOHAMED HAYSAM AFIFY

I. INTRODUCTION

FPGA technology has been mostly utilized for many on-chip applications to obtain higher throughput. ARM has devised Advanced Microcontroller Bus Architecture (AMBA) to support efficient on-chip communication. AMBA is hierarchically organized into two bus segments, system and peripheral bus. These are mutually connected via bridge that buffers data and operations between them. The figure 1 shows a schematic overview of a typical microprocessor design [1][2]. In this design, communication is provided by using system bus, i.e. Advanced System Bus (ASB), or Advanced High-performance Bus (AHB). All high performance components are connected to the system bus. Low speed devices are connected to the Advanced Peripheral Bus (APB). AMBA does not determine method of arbitration.

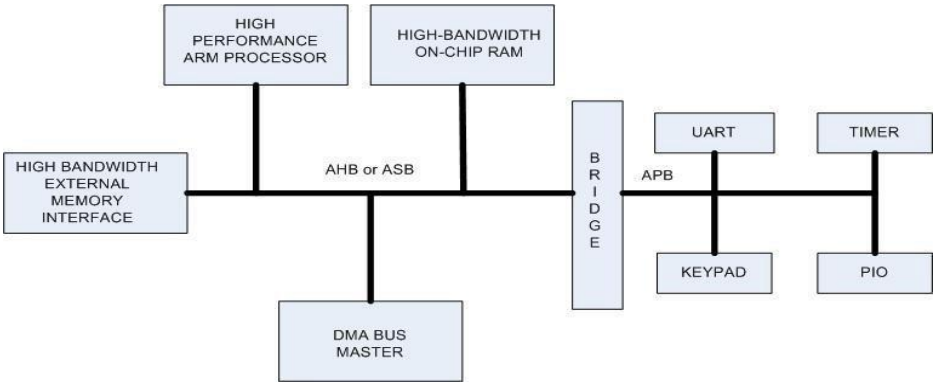
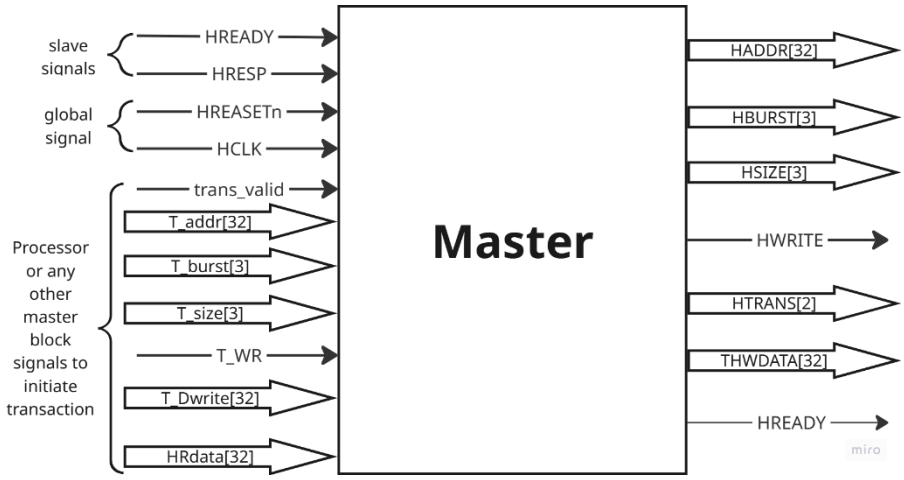


Fig.1 Schematic overview of a typical microprocessor design

Overview

This module implements a simplified AHB Master capable of handling two types of burst transfers: SINGLE and INCR (undefined length). The master generates the necessary address and control signals for each AHB transfer and handles basic write and read operations.



AHB Master Output Signals Description

Signal	Destination	Description
HADDR[31:0]	**Slave and decoder**	The 32-bit system address bus. Carries the address for each transfer.
HBURST[2:0]	**Slave**	Indicates the burst type: SINGLE, INCR, WRAP (not all supported). Used to determine burst length and behavior.
HSIZE[2:0]	**Slave**	Specifies the size of each transfer (Byte, Halfword, Word, etc.). Sizes > 4 Bytes are downgraded and reported via a warning.
HTRANS[1:0]	**Slave**	Indicates the transfer type: IDLE, BUSY, NONSEQ, or SEQ. Crucial for burst progression and cancellation.
HWDATA[31:0]	**Slave**	Carries data from the master to the slave during write operations. Driven during the data phase.
HWRITE	**Slave**	Specifies transfer direction. HIGH = write, LOW = read. Must remain stable throughout a burst.
HREADY	**Slave**	Master-driven signal used to indicate readiness to start or finish a transfer. Toggled appropriately in each FSM state.

Burset operation

It supports incrementing an single bursts:

Incrementing bursts access sequential locations and the address of each transfer in the burst is an increment of the previous address.

HBURST[2:0]	Type	Description
3'b000	**SINGLE**	A single beat transaction. HTRANS is set to NONSEQ.
3'b001	**INCR**	**Incrementing burst of undefined length** First beat: HTRANS = NONSEQ Subsequent beats: HTRANS = SEQ
3'b010–3'b111	**Reserved/Unsupported**	**Not handled in current design, defaulted to SINGLE with warning**

Transfer size

HSIZE[2:0] indicates the size of a data transfer.

HSIZE[2]	HSIZE[1]	HSIZE[0]	Size (bits)	Description
0	0	0	8	Byte
0	0	1	16	Halfword
0	1	0	32	Word

Any unsupported HSIZE value (i.e., greater than 3'b010) will be treated as a Word (4 bytes) transfer. A warning message will be displayed during simulation.

Transfer types

Transfers can be classified into one of four types, as controlled by **HTRANS[1:0]**.

HTRANS[1:0]	Type	Description
2'b00	IDLE	Indicates that no data transfer is required. Used to terminate locked transfers. Slaves must respond with zero wait state OKAY and ignore the transfer.
2'b01	BUSY	Enables idle cycles within bursts. Used when next transfer in a burst cannot proceed. Slaves must respond with zero wait state OKAY and ignore the transfer.
2'b10	NONSEQ	Indicates a single transfer or first transfer of a burst. Address and control unrelated to previous transfer.
2'b11	SEQ	Indicates remaining transfers in a burst. Address is derived from previous transfer plus transfer size.

Slave response

After a master has started a transfer, the slave controls how the transfer progresses. Amaster cannot cancel a transfer after it has commenced.A slave must provide a response that indicates the status of the transfer when it is accessed. The transfer status is provided by the HRESP signal

HRESP	Response	Description
0	**OKAY**	**Transfer completed successfully or is ongoing. HREADY indicates finality.**
1	**ERROR**	**Transfer failed. Slave must respond over two cycles. Master takes recovery action in error_state1 and error_state2.**

Error Response Handling

According to the AMBA AHB-Lite specification, a slave can issue an ERROR response to indicate an exceptional condition—commonly protection violations such as writing to a read-only region. The protocol defines a two-cycle response mechanism for ERROR to account for the pipelined nature of the bus:

- Cycle 1 (error pending):
    - HRESP = 1 (indicating ERROR) ○ HREADY = 0 (extend the data phase by one cycle)
- Cycle 2 (error completed):
    - HRESP = 1
    - HREADY = 1 (completes the transfer)
    - During this cycle, the master is expected to drive HTRANS = 2'b00 (IDLE) to cancel or prepare for the next transfer.

The AHB-Lite spec also allows the master either to cancel the remaining transfers in a burst or to continue the burst after the error, depending on system design.

AHB-Lite Transactions **Master Side:**

1. Basic Read
2. Basic Write
3. Burst Read –inc.burst lengths -1,4,8,16
4. Burst Write –inc.burst lengths-1,4,8,16
5. Communication is initiated by master

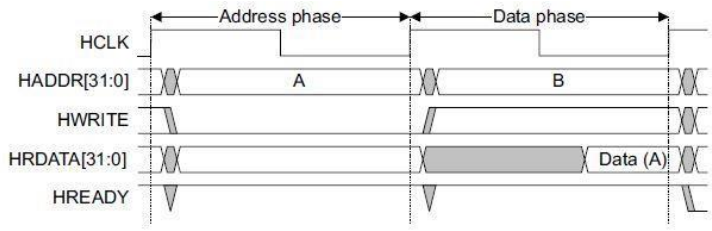


Figure 3-1 Read transfer

Slave Side:

1. Can make the master wait
2. Can give an Error Response
3. But, slave can't terminate transaction; can ask master to insert wait state

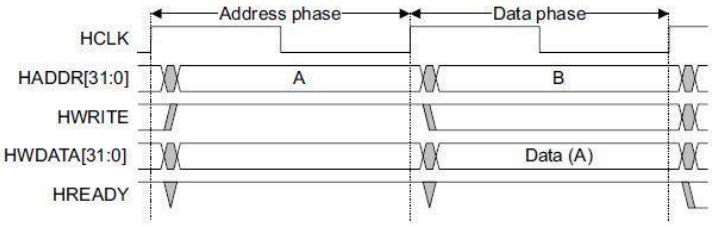


Figure 3-2 Write transfer

•All transactions are pipelined

Types of transfer that I have supported in my design

Undefined length bursts, INCR

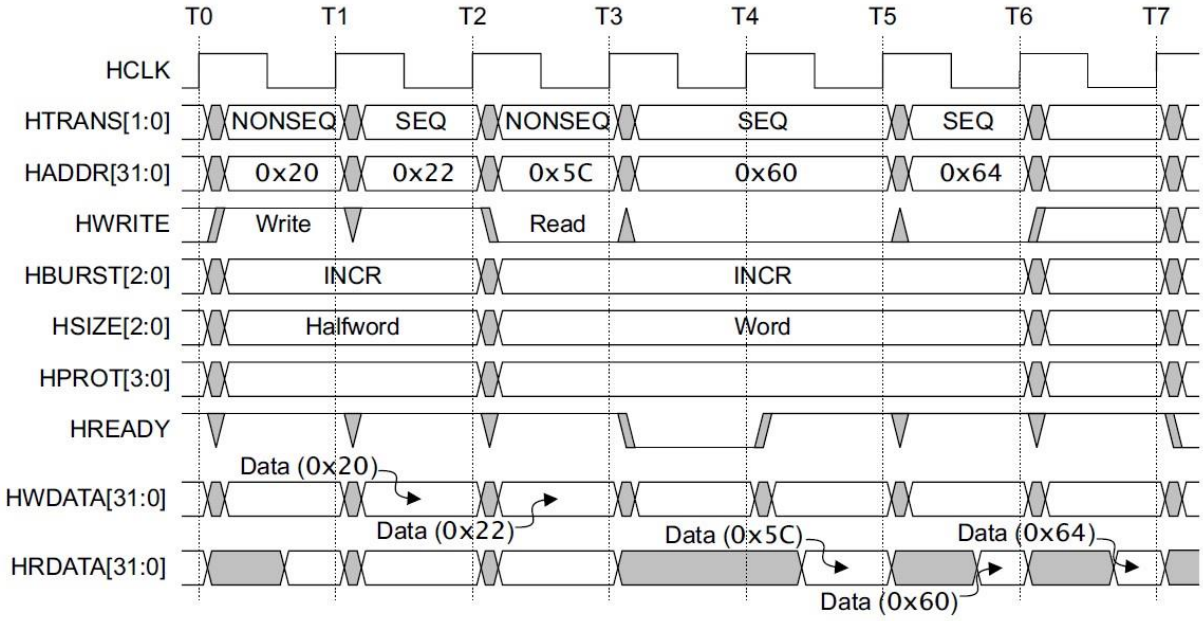


Figure 3-12 Undefined length bursts

Waited transfers

Slaves use HREADY to insert wait states if they require more time to provide or sample the data. During a waited transfer, the master is restricted to what changes it can make to the transfer type and address.

IDLE transfer

During a waited transfer, the master is permitted to change the transfer type from IDLE to NONSEQ. When the HTRANS transfer type changes to NONSEQ the master must keep HTRANS constant, until HREADY is HIGH.

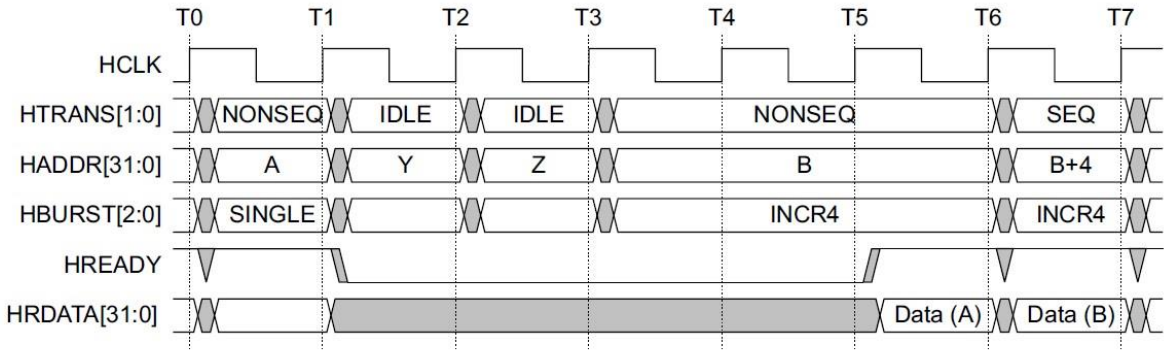


Figure 3-13 Waited transfer, IDLE to NONSEQ

BUSY transfer, undefined length burst

During a waited transfer for an undefined length burst, INCR, the master is permitted to change from BUSY to any other transfer type, when HREADY is LOW. The burst continues if a SEQ transfer is performed but terminates if an IDLE or NONSEQ transfer is performed.

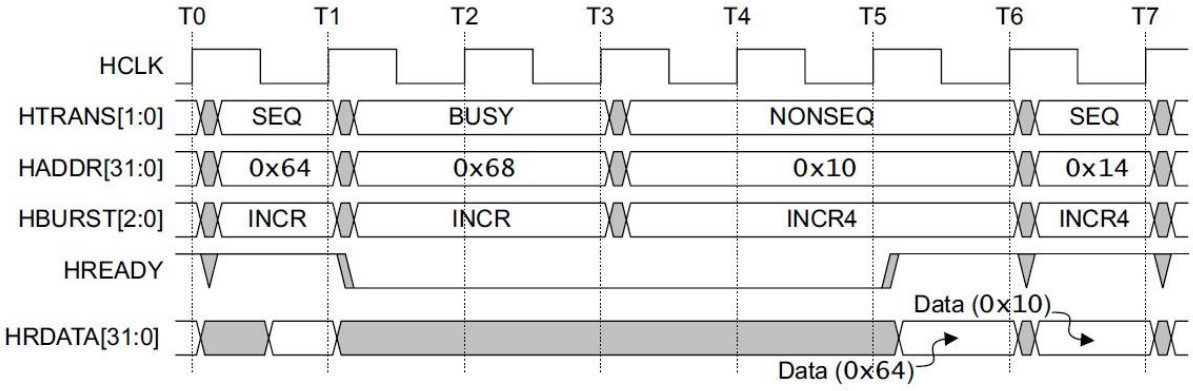


Figure 3-15 Waited transfer, BUSY to NONSEQ for an undefined length burst

After an ERROR response

During a waited transfer, if the slave responds with an ERROR response then the master is permitted to change the address when HREADY is LOW.

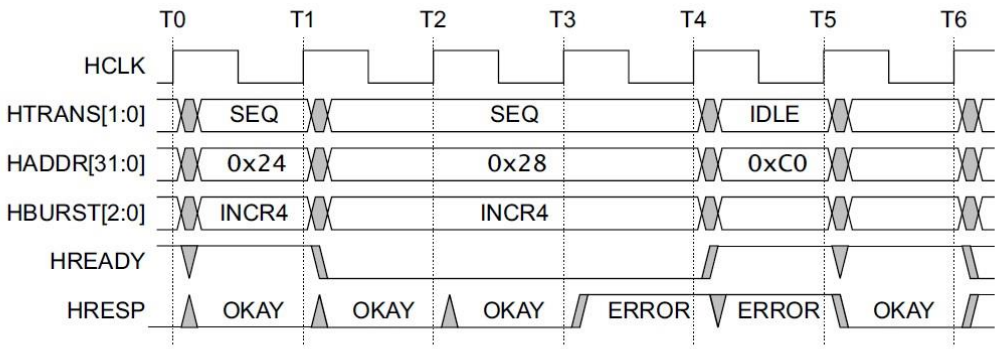
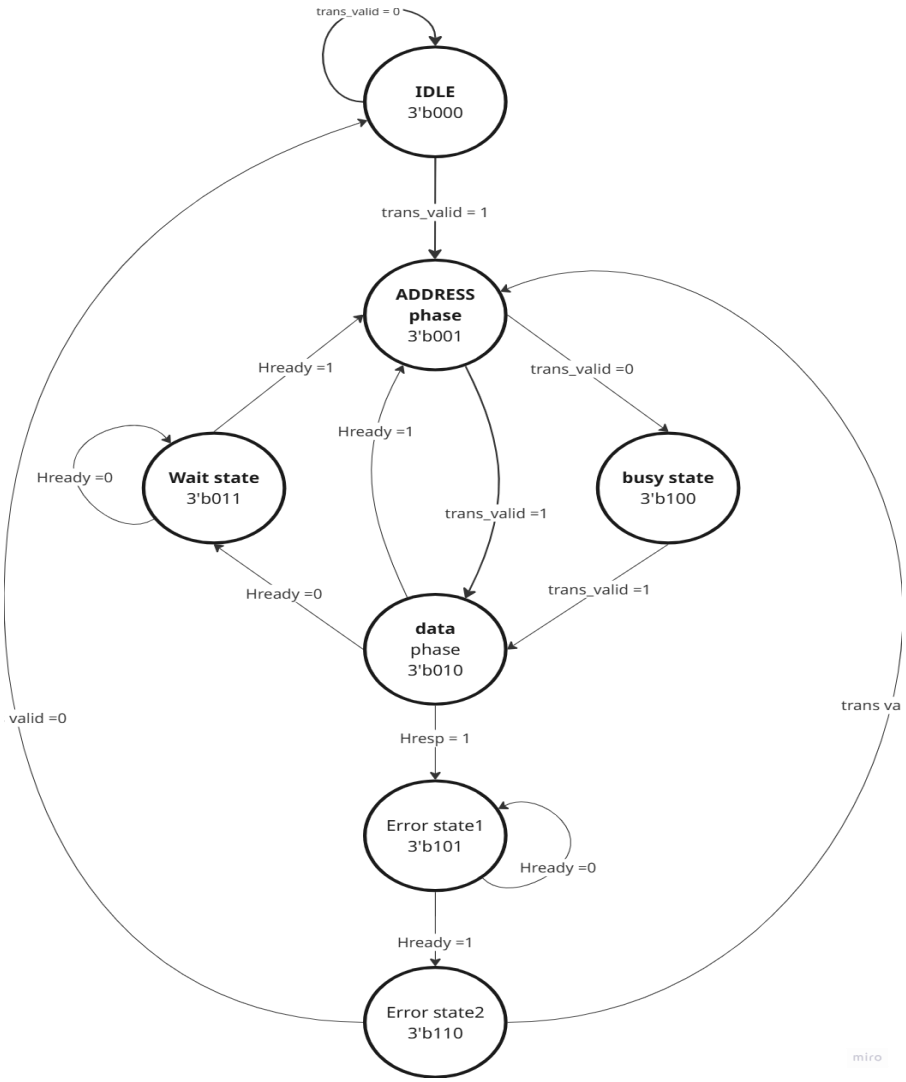


Figure 3-17 Address changes during a waited transfer, after an ERROR

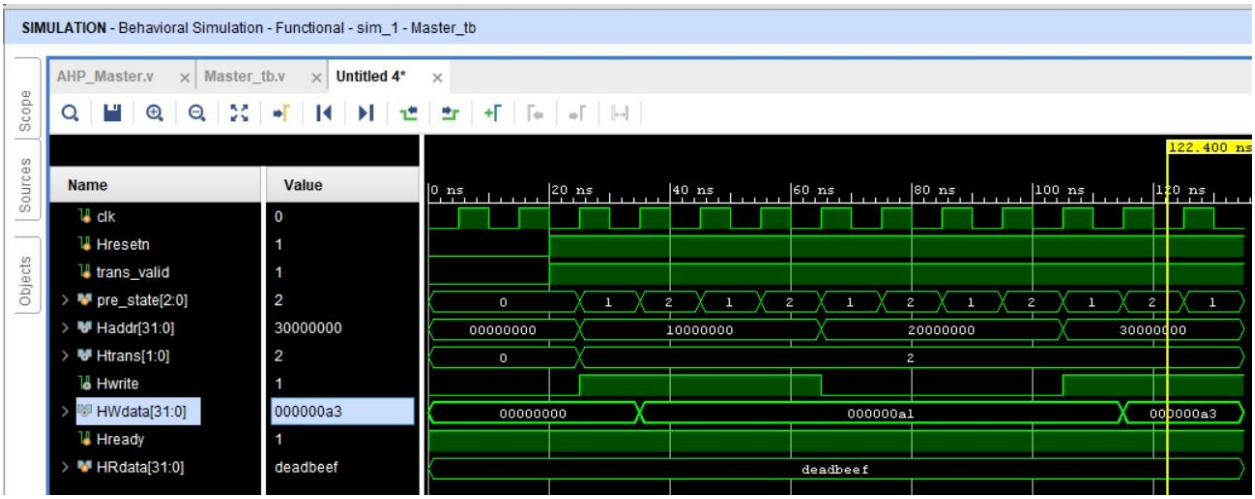
Master FSM



State	Description and Purpose
IDLE	This is the default and initial state. The master waits for a valid transaction (trans_valid = 1). All outputs are cleared. Transitions to address_phase when a transaction is valid. Outputs: HTRANS = IDLE, all other signals are cleared, HREADY = 0.
address_phase	The master drives the address (HADDR), burst (HBURST), size (HSIZE), and transfer type (HTRANS) depending on burst type. It handles address sequencing for burst transfers. Transitions to data_phase or busy_state (if INCR burst and trans_valid = 0). Outputs: Valid address and control signals, HREADY = 1.
data_phase	Drives write data (HWDATA) or displays read data (HRDATA). Checks for slave response. Transitions: to error_state1 if HRESP = 1, to wait_state if HREADY = 0, else to address_phase to continue burst. Outputs: HREADY = 0, valid write data for write operations.
wait_state	Holds all outputs constant while waiting for HREADY = 1 from the slave. Transitions: to address_phase once HREADY = 1. Outputs: All signals are held; HREADY = 1.
busy_state	Used for INCR burst if trans_valid = 0. The master waits until trans_valid returns to 1. Transitions: to data_phase when trans_valid = 1. Outputs: HREADY = 0, other outputs are held.
error_state1	First cycle of error response: HRESP = 1, HREADY = 0. The slave signals an error; master holds. Transitions: to error_state2 when HREADY = 1. Outputs: HREADY = 0. No changes to other signals.
error_state2	Second cycle of error: HRESP = 1, HREADY = 1. Master clears outputs and chooses whether to continue or cancel the burst. Transitions: to IDLE if trans_valid = 0, else to address_phase to resume burst. Outputs: HTRANS = IDLE, all other outputs cleared, first_transfer = 0.

Simulation of results:

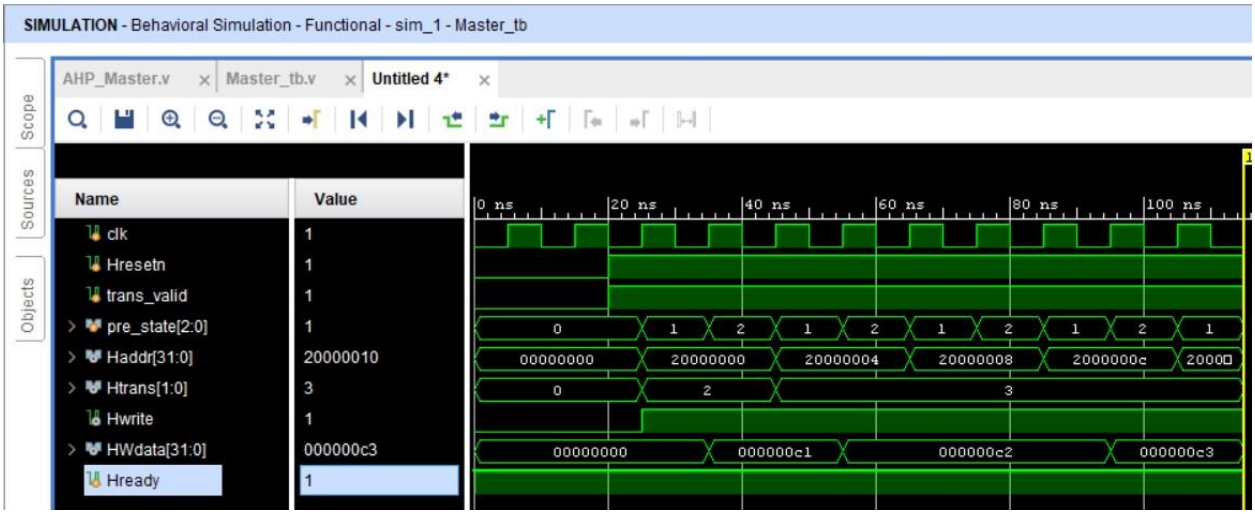
1. Single burst with sizes 1, 2, 4 bytes



First burst and the final one are with write operation the middle one is with read operation

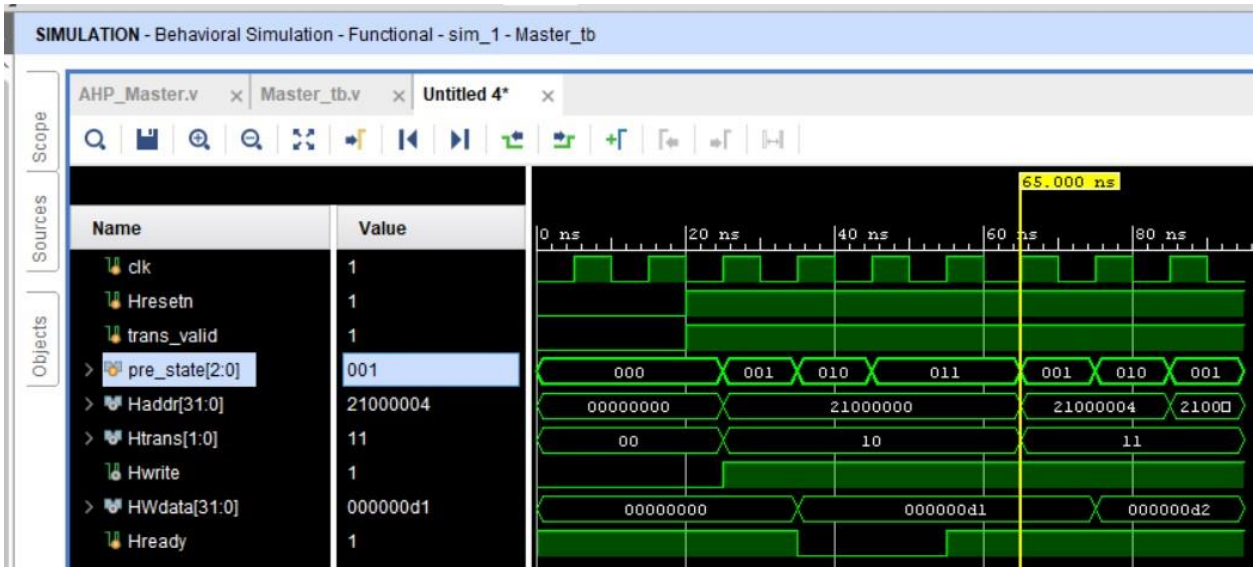
Htrans = 2'b10 ➒ 2 (non seq) in all bursts

2.INCR burst with 4-byte size, no wait



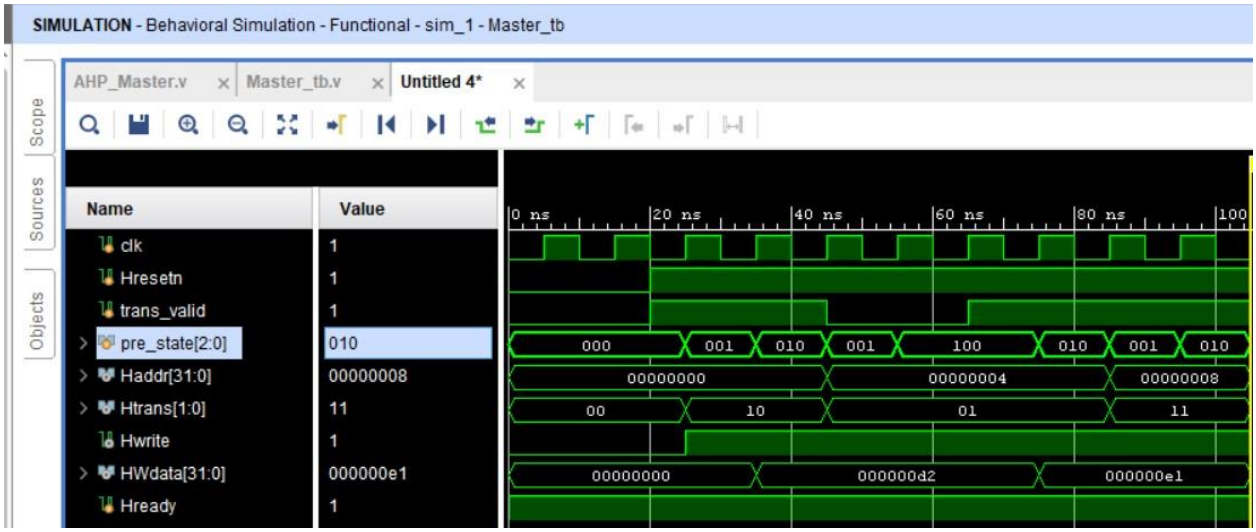
Htrans = 2'b10 ➒ 2 (non seq) and with second beat Htrans 2'b11 ➒ 3 (seq)

3.INCR burst with wait state



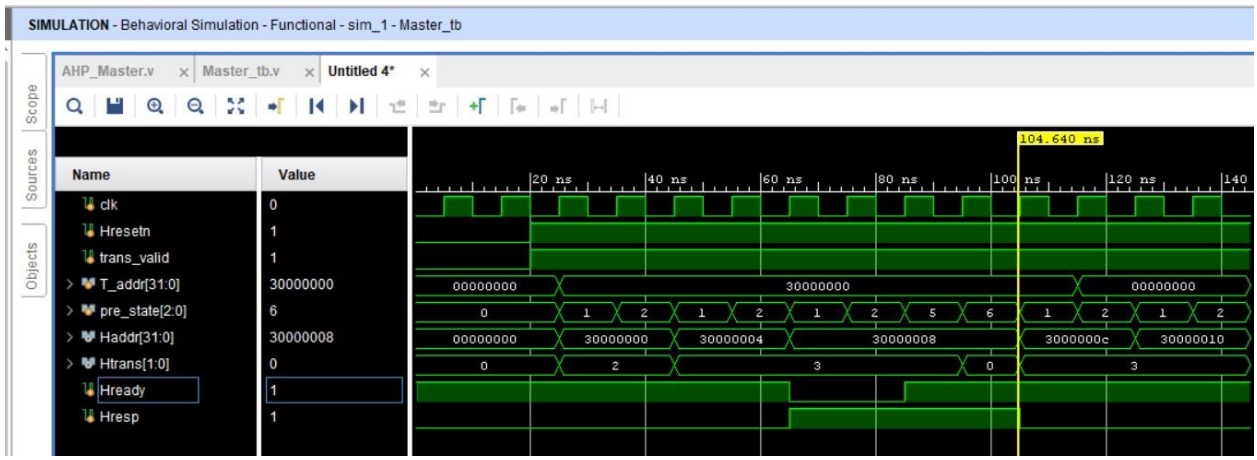
Pre_state = 3’b011 (wait state) when Hready = 0

4.INCR burst with BUSY master



When trans_valid = 0 the master transfer to busy state so pre_state = 3’b100 (busy state) and Htrans = 2’b01 (busy type) notice data still constant until master transfer to data_phase

5.ERROR and master CONTINUES burst



When there's an error first cycle Hready =0 ,Hresp =1 after that master changes the transfer type to IDLE Htrans = 0 finally when cycle 2 of error end Hready =1 ,Hresp =1 master continue the remaining transfers in the burst

6. ERROR and master CANCELS burst

After the cycle 2 of error master can cancel the remaining transfers in the burst and begin new one