Final Project – Round Robin Arbiter

1. Round Robin Algorithm

Round robin arbitration is a scheduling scheme which gives each requestor its share of using a common resource for a limited time or data elements. The basic algorithm implies that once a requestor has been served, it would “go around” to the end of the line and be the last to be served again.

2. General Description

In the arbider design there are 4 hosts (masters) requesting access to one slave.
The hosts are : Master Host, Slave Host, Test Host and Extra Host.
They are all identical.

A master starts a transaction by:

- A write or read signal
- The relevant address
- Byte enable signal
- Data. In case of write it holds the data.

The slave answers with the ack signal and returns the data in case of a read

This ends the transaction and free the buses for the next transaction.

3. Signal map description

Name	Description
mwr/swr/twr/ewr	Write signal - indicates a write transaction
mrd/srd/trd/erd	Read signal - indicates a read transaction
maddr/saddr/taddr/eaddr [AW-1:0]	Address bus - the address of the write/read transaction
mbe/sbe/tbe/ebe [BW-1:0]	byte enable. Which bytes are relevant in case of a write transaction. The Be data is transferred to the slave as is
mdwr/sdwr/tdwr/edwr [DW-1:0]	Write data bus - in case of write transaction, the master puts here the data
mdrd/sdrd/tdrd/edrd [DW-1:0]	Read data bus - in case of read transaction, the slave puts here the data

Ack	Acknowledge indication from the slave that ends the transaction The Acknowledge uncludes 2 bits. ack[1] – timeout. No respond was received from the slave ack[0] – the slave's response
mcpu/scup/tcpu/ecpu	Indicates the master CPU. In one of the hosts it has to be 1 and in the others it has to be 0.

4. Write & Read Transaction

Write Trasaction

The host master issues a write signal (mwr/swr/ewr/twr) with the address bus (maddr/saddr/taddr/eaddr) and the write_data bus (mdwr/sdwr/tdwr/edwr).

The host holds the signals until an ack signal is received.

When the host receives the ack it releases the buses and a new transaction can start.

Read Transaction

The host master issues a read signal (mrd/srd/erd/trd) with the address bus (maddr/saddr/taddr/eaddr).

The host holds the signals until an ack signal is received with the read_data bus (mdrd/sdrd/tdrd/edrd).

When the host receives the ack it releases the buses and a new transaction can start.

An example of a write transaction followed by a read transaction can be seen in figure 4.1

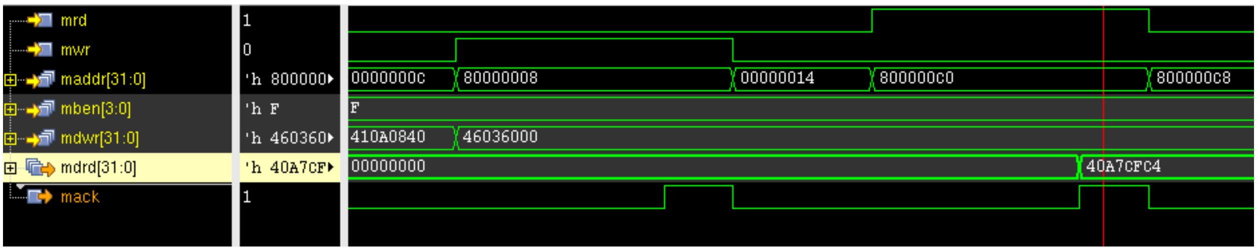

Figure 4-1 Waves Diagram: write and then read transactions

5. Block Diagram

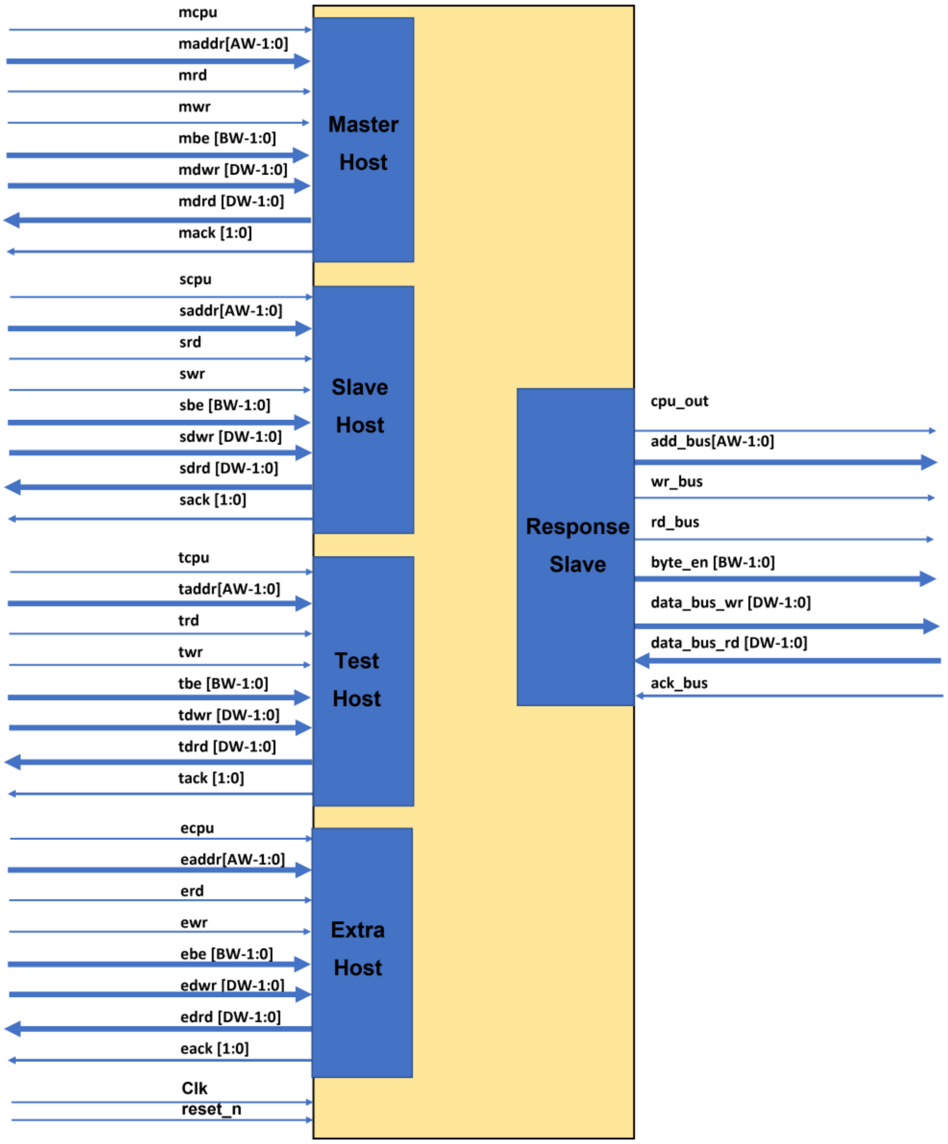

Figure 5-1 Arbiter Block Diagram

6. Parameters Value

timeout_limit = ({TW{1'b1}} - 1);

Parameters	Value
AW	32
DW	32
BW	4
TW	16

7. Timeout Counter

The timeout counter is initialized with timeout_limit = ({TW{1'b1}} - 1) and counts from zero until it reaches the timeout_limit.

When the limit is reached ack[1] is asserted.

8. CPU bus

This signal has to be '1' in one of the hosts and '0' in the other three hosts. The value of this signal is send, as it is, to the slave.