HyperRAM interface

This document is a guide to understand the code implementation and decisions made as the development of the interface is in process.

Note that various signals and I/O ports have been modified in order to be able to debug as the code is being implemented. The final code will have the same ports as in [this link](https://infrequent-cormorant-741.notion.site/HyperRAM-interface-a996bf71213f4ebea602a78541cd65c9).

The information for the configuration of the devices was taken from Winbond’s product datasheet W958D8NBYA5I.pdf.

**Files**

Three vhd files are presented.

clk\_div\_2.vhd is a simple clock divider which is presented in a separate file for more clarity.

hyperram.vhd is the file where the behavior of the interface is being implemented.

hyperram\_tb.vhd is the testbench for the project.

**Implementation**

In order to get the requested behavior, the program starts requesting the devices ID’s. This is done via a read request considering some particular requirements that the datasheet informs. Note that the bidirectional signals have been separated in order to be able to perform the simulations since ModelSim’s tool does not allow to edit bidirectional signals. All the debug signals and ports are identified with a \_dbg suffix to be removed and replaced by the definitive signals and ports.

The transaction then starts by lowering CS with the RAM’s clock in 0 as the datasheet requests.

Six bytes have to be sent to request each one of the identification registers. The devices respond with two bytes which are repeated, so only the first two are considered for each register. The latency at this point should be 7 since it has not yet been configured but the code just waits for a raising edge in RWDS so that is not considered.

The command addresses that are sent are “E0 00 00 00 00 00” for the first register and “E0 00 00 00 00 01” for the second and the data that is expected to be read is “0E86” and “0001”.

If everything is correct, then the configuration of the devices is performed. If not, an error flag is set and the interface will be halted until reset.

For the configuration, no latency is needed, so the command and data are sent consecutively.

The addressing is “60 00 01 00 00 00” for command register 0 and “60 00 01 00 00 01” for command register 1. The data that will be sent is “DFEC” and “FF81” respectively. This configures the devices as requested (burst length 128, Fixed latency, minimum latency 3,27 Ohm drive strength). Then the interface is put in idle state waiting for read/write requests from the user with ready = ‘1’.

The read and write requests are yet to be implemented

**Simulations**

To verify the project’s performance, timing simulations have been done with Active-HDL. All the initial setup.

The time simulations are found in ..\hyperramtest\hyperramtest.adf

The behavior of the RAM devices were set manually.

Protected Verilog files of the devices were found but for the time being couldn’t be used successfully yet.

The folder Docs contains the datasheet, protected Verilog files, original requirements and this document.