The **Microelectronics Training Center**

The MTC is an initiative within the INVOMEC division

Industrialization & Training in Microelectronics

Lab-exercise

Lab 4:

Design of the memory unit

Cluster: Cluster2
Module: Module3a

Target group: Students

Version: 1.1
Date: 13/12/06
Author: Osman Allam
Modified by: Geert Vanwijnsberghe
History : testbench added, text updated

This material was developed with support of the European Social Fund.
ESF: Prevent and combat unemployment by promoting employability, entrepreneurship, adaptability and equal opportunities between women and men, and by investment in people.
http://www.esf-agentschap.be

Introduction

The main memory is an essential component of almost all computer systems. Programs and data are stored in the main memory.

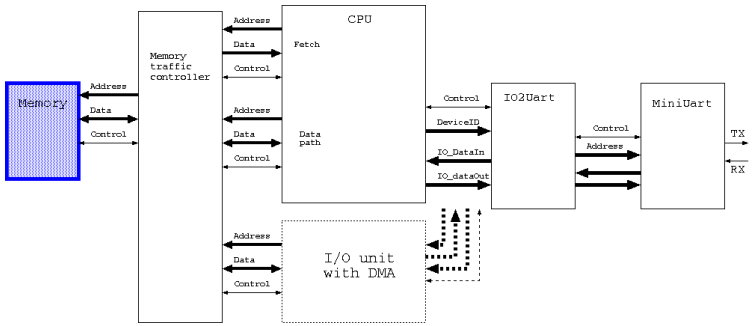

Figure 1: System architecture

Objectives

After completing this module, you should be able to:

- Understand RAM operations;
- Model RAM modules in VHDL.

Knowledge background

- Basic VHDL knowledge
- Basic knowledge about HDL synthesis

Classification

- Level: 2
- Duration: 30 minutes

Input

- VHDL template for the memory block
- Memory contents package: `micro_ram_pk.vhd`
- Testbench to verify you code

The lab

The main memory in Micro6 holds both the program and data. It has two separate ports for reading and writing instead of a single bidirectional port. However, both reading and writing operations use the same address port.

Selection of which operation a synchronous memory performs is accomplished by either one of the following methods:

1. Valid Memory Address (`VMA`) line + Read/Write (`Rd/Wr`) line
   The memory observes the `Rd/Wr` line only when the `VMA` line is active and performs the corresponding operation.
2. Read (`Rd`) line + Write (`Wr`) line
   One of the two lines has to be asserted to perform a memory operation. If both lines are inactive, this resembles the case when `VMA` is inactive in the above method. If both lines are asserted, then the line with the higher priority is observed and its associated operation is performed.

Exercise

Using the template provided in the file `memory.vhd`, design the memory unit, you are going to use as the internal memory of the microprocessor. The word width is the same as the data width of the microprocessor. In this way, an instruction or a data word can be read or written in a single memory cycle.

The addressable space is determined by the width of the address bus; 12 bits in our design. This gives a memory size of 4K words.

The memory contents should be implemented as a variable since this makes the simulation much more efficient and faster.

- Use the data type `memContents_t` for the memory contents.
- Separate read (`rd`) and write (`wr`) lines. When both are asserted simultaneously, the memory performs a write operation.
- The memory asserts the signal (`Ready`) after each successful operation. The (`Ready`) remains asserted until the next operation request.
- The memory contents are initialized by the constant (`RAM_CONTENTS`) declared in the package `micro_ram_pk`. Don't forget to "use" it in your design.

In the next figure you see an example of how data can be written or read.

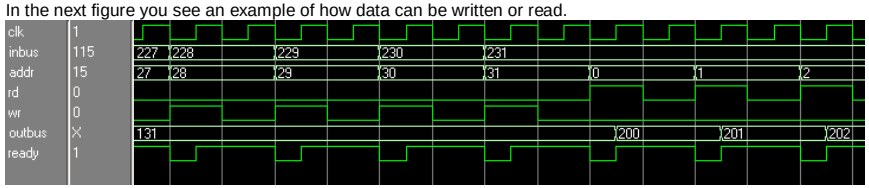

Figure 2

ⓘ Synthesizing memory blocks: When writing VHDL code for any kind of memory block, you have to keep in mind what structures of the target technology meet your specifications. You will probably have to adjust your code to comply with synthesis tools' templates to ensure that you get the required synthesis results.

ⓘ When implementing a RAM on a Xilinx FPGA yu will use the core generator tool of Xilinx . This tool generates automatically the VHDL code for simulation and an EDIF netlist for implementation.

Use the tb_memory.vhd to verify your design.

Memory contents package

The memory contents package `micro_ram_pk` declares one constant, `RAM_CONTENTS`, defining the initial memory contents.