Alexandria University
Faculty of Engineering
Computer and Systems Engineering Department
CS 401 - Graduation Project

MIPS Home Computer System

System Design Reference

Student	Supervisor
Mostafa Abd El-Aziz	Prof. Dr. Layla Abou Hadid

Table of Contents

1 Introduction

The project is implemented using Nexys-2 FPGA board. The board contains a Spartan-3e FPGA chip, RAM chip, ROM chip, 8 input switches, 4 push buttons, 4 7-segment LEDs, VGA port, RS-232 port, PS/2 port, 32 General Purpose I/O pins, 50MHz crystal oscillator, JTAG, USB (for interfacing with a PC), and other relevant components.

We can divide the design of the system into two main parts:

- **Solid part**: the solid components on Nexys-2 board that are used as functional part of the computer system. The components are:
  - On-board ROM chip: stores computer firmware.
  - On-board RAM chip: mainly used to store software that is being executed by CPU (except firmware, which is executed directly from ROM) and temporary data.
  - VGA port: a VGA-compatible CRT screen is connected to the VGA port to display information.
  - PS/2 port: a PS/2 keyboard is connected to PS/2 port to feed the machine with data.
  - 50MHz crystal: responsible for synchronization of the system.

- **Volatile part**: implemented in VHDL and compiled using Xilinx SDK. The compiled design is uploaded to Spartan-3e FPGA chip through USB and JTAG. The logic uploaded to FPGA communicate with the outer world (components on Nexys II boards) through FPGA chip pins that are connected to this components through wires. This part consists of the following HDL modules:
  - Central Processing Unit: implements MIPS I instruction set.
  - Memory Interface Unit: interfaces on-board RAM and ROM.
  - VGA Controller: interface on-board VGA port.
  - PS/2 Controller: interface on-board PS/2 port (not implemented).
  - Bus controller: logic of communication between CPU and other components.

When the computer boots up, CPU starts executing firmware instructions that are stored on the ROM chip on Nexys-II board. Software can access I/O devices (like VGA controller) using memory instructions (load and store). The CPU doesn't differentiate between I/O accesses and memory accesses (there is no isolated I/O space). Consequently, I/O devices share the same bus with RAM.

Memory locations that are reserved for communication with I/O devices are called memory-mapped I/O registers. The following table shows the memory layout and memory-mapped I/O locations:

Address Region	Region Size	Device
0x00000000-0x0000FFFF	64KB	Lowest 64KB of on-board ROM
0x00010000-0x00017FFF	32KB	Lowest 32KB of on-board RAM
0x00018000-0x0001FFFF	32KB	VGA frame buffer
0xFFF00000-0xFFFFFFFF	1MB	Keyboard I/O

Figure 1: Memory Address Space layout

When CPU needs to fetch an instruction or a data word from memory, the location of this instruction/data is put onto the address bus. Logic on the FPGA decides which component shall reply with data based on the table shown above. This logic is part of top-level design of the machine. Sheet 1 (appended to this documents) shows the high-level components of the system.

2 Central Processing Unit

The CPU module implements MIPS I instruction set architecture along with an instruction cache and a data cache. It is planned to implement a TLB in the future. Logically, the module consists of two submodules: the core pipeline and the cache system (as shown in sheet 2).

2.1 The Core Pipeline

This module contains the instruction path and data path which are pipelined into 5 stages: IF, ID, EX, MEM, and WB. IF is responsible for interfacing instruction cache, while ID contains

register file, hazard detection unit, and instruction decoding logic. EX contains ALU, MEM interface data cache, and finally, WB interfaces register file for write.

The pipeline is an extended version of the simple 5-stage pipeline of MIPS subset. Our proposed design completely implements MIPS I ISA as we said before. The pipeline is written in VHDL as a single module that is made to interface the cache directly. Sheet 3 shows the design of the pipeline.

2.2 Cache System

The cache module consists of an instruction cache, a data cache, a buffer, and a sequential control unit that detects cache misses and fills in the buffer with memory orders. As long as the buffer is not empty, the pipeline is stalled and memory cycles are issued to execute the entries of the buffer.

Both instruction and data caches are write-through caches, so any store instruction will eventually stall the pipeline until memory write cycle is done (and the buffer becomes empty). The cache array is implemented using block memory on the FPGA chip.

Cache module is a multiplexing unit. Instruction cache hit and data cache hit can be processed in the same time, but if instruction and data cache misses occur in the same time, instruction cache miss will be processed first, then data cache miss. This limitation is due to the single-bus nature of Von Neumann architecture (Von Neumann Bottleneck).

3 Memory Interface

The memory interface module is responsible for controlling the bus between FPGA chip and on-board ROM and RAM chips. Each chip has 16MB memory, arranged in 8M 16-bit words. This results in inconsistency between word size of both ICs and the word size of CPU data bus.

To deal with this problem, the memory interface module translate 32-bit read/write cycles into two 16-bit read/write orders that are processed in sequence. 8- and 16-bit cycles don't need to be divided into two cycles, so they can finish up faster.

4 VGA Controller

The VGA module is designed to have the same software interface of IBM PC VGA in text-mode. In PC, VGA provides a buffer (starting at 0xB8000) that can be accessed directly by software to draw characters on screen (text mode). In mode 03h, screen has 25 rows and 80 columns, resulting in 2000 characters (4000 bytes).

Our implementation provides this interface by implementing several VGA RAM modules that are mapped to specific memory locations (0x18000-0x1FFFF). The VGA RAM is interfaced to system bus through graphics controller. VGA contains logic (the sequencer and CRT controller) that continuously scans VGA RAM and generates RGB signals to refresh CRT screen view. Nexys II provides an On-chip DAC (a set of resistors) to convert RGB signals into analog form.