A RISC-V computer system design

Xiong Juncheng

December 11, 2023

Abstract

In this project, I designed a simple RISC-V computer system on FPGA. The FPAG board is Digilent Nexys A7-100T. The compter system is composed of: a CPU, a data memory, an instruction memory, a vga screen, a keyboard, and a PWM audio output. The CPU is a 5-stage pipeline processor, which can execute 32-bit RISC-V instructions. The system interface is a terminal, which can execute basic commands and run two software - Snake and Piano.

Keywords: RISC-V, FPGA, 5-stage pipeline, computer system

Contents

1 Introduction

1.1 Overall introduction

In the realm of digital systems design, the creation of a fully functional computer system on a Field-Programmable Gate Array (FPGA) stands as a testament to the designer's knowledge of digital systems. This project centers around the development of a simple yet robust RISC-V computer system, implemented on the Digilent Nexys A7-100T FPGA board. At the heart of this project lies the CPU, which is a 5-stage pipeline processor that can execute 32-bit RISC-V instructions. The computer system designed for this project includes essential components such as a CPU, a data memory, an instruction memory, a VGA screen, a keyboard, and a PWM audio output. The user interacts with the system through a terminal, which supports several commands and can run two software - Snake and Piano.

This report is divided into two parts. The first segment delves into the hardware design, concentrating on the intricacies of the 5-stage pipeline CPU, memory management, and all other devices. The second part shifts focus to the software design, encompassing the implementation of the terminal, the engaging Snake game, and the musical pursuit in the form of the Piano game.

1.2 FPGA

The FPGA board for this project is Digilent Nexys A7-100T, a circuit design and implementation platform for classroom use. For more information, please refer to `https://digilent.com/reference/programmable-logic/nexys-a7/start`.

1.3 RISC-V

RISC-V is an open-source instruction set architecture (ISA) based on reduced instruction set computer (RISC) principles. It is a standard ISA designed to be simple, extensible, and easy to implement. In this project, I implemented a 32-bit RISC-V CPU, which can execute all 37 base instructions. For more information about RISC-V itself, please refer to `https://riscv.org/`.

2 Hardware

2.1 5-stage pipeline CPU

2.2 Memory management

The CPU uses byte addressing, and access all other devices through memory-mapped I/O. The memory address is 32 bits wide, and the first 12 bits are used to select the device. The address map is shown in Table **??**.

Table 1: Address Map

Address range	Device
0x00000000 - 0x000FFFFF	Instruction memory
0x00100000 - 0x001FFFFF	Data memory
0x00200000 - 0x002FFFFF	VGA screen
0x00300000, 0x00300004	Keyboard
0x00400000, 0x00400004	Timer

2.3 Instruction and data memory

The instruction and data memory are both implemented using block RAM (BRAM). The instruction memory is read-only, and the data memory is read-write. Both of them are 1MB in size. The instruction address must be aligned to 4 bytes, while the data address can be any byte address.

2.4 VGA screen

The screen is read-write. The resolution is 640x480, and the color depth is 12 bits. Because the system is totally based on a terminal and the games are using characters as the basic unit, the screen is divided into 80x30 characters of 8x16 pixels. This allows us only store 80x30 characters' ascii code. To accelerate the screen access speed, the screen is implemented with an one-dimensional array instead of a two-dimensional array. Considering the line and column size, 11-7 bits of the address are used to represent the line number, and 6-0 bits are used to represent the column number($2^5 = 32, 2^7 = 128$).

2.5 Keyboard

2.6 PWM audio output

2.7 Timer

3 Software

3.1 Terminal

3.2 Snake

3.3 Piano