Design and Implmentation of a MIPS Single-Cylce Processor with ADC Integration

Project Overview

This project involves the design and implementation of a 32-bit single-cycle MIPS processor with an integrated Analog-to-Digital Converter (ADC) using Xilinx Vivado and SystemVerilog. Implemented on the Digilent Basys3 FPGA, the design features two clock domains: the processor operates at 60 MHz, while the ADC runs at 50 MHz. The processor follows the RISC architecture and supports R-type, I-type, and J-type instructions, executed in five stages: Instruction Fetch, Decode, Execute, Memory, and Write Back. This modular design ensures simplicity and scalability, providing a foundation for future enhancements like pipelining.

Features

- **Comprehensive Instruction Set:** Supports R-type, I-type and J-tpye instructions, enabling a wide range of operations and functionalities.
- **Five-Stage Datapath:** Instructions are executed in five distinct stages based on their types. the stages include: *Instcution Fetch(IF) Instcution Decode(ID) Instcution Execute(EXE) Instcution Memory(MEM) Instcution WriteBack(WB)*
- **ALU (Arithmetic and Logic Unit):** Incorporates a a comprehensive arithmetic and Logic unit with a support for a total of 6 arithmetic and logic operations namely: *ADD SUB AND OR SRL SLT*.
- **Control Unit:** Incorporates a comprehensive Control Unit which manages the flow of operations in the various stages of instruction execution.
- **Thermocoder:** This gives a visual of the sampled data from the analog to digital converter using the 16 onboard LED on the Basys 3 board.
- **4-Digit seven segment display:** The Basys3 board features a 4-digit seven segment display which is used in this project to output sampled voltage values from 0V to 1V.
- **Analog to Digital Converter (XADC):** The Basys 3 board has an onboard 16-bit analog to digital converter with the 12 most significant providing accurate and reliable output. This is integrated through into the processor using **memory map IO**.
- **ADC Wrapper:** This is an interface between the XADC and the MIPS processor which handles the synchronization of signals moving between clock domains.i.e. from processor clock domain(60MHz) to ADC clock domain(50MHz).

Prerequisites

- Xilinx Design Suite (Vivado).
- FPGA Board(preferably Digilent Basys3 for full compatilibity with the project)
- Understanding of hardware description language and FPGA programming.
- Basic understanding of the Vivado Software.

Installation

1. Clone the repository to your local machine.
2. Open the project in Vivado.
3. Configure your FPGA board settings as required. The constraint file in this project is specific to the the Basys3 board.
4. Generate the IP clock module with two output clocks one for the processor running at 60MHz and the other for the XADC running at 50MHz.
    - Name the output of the processor clock **clk** and name the output of the XADC clock **clk_adc** .
5. Generating the IP XADC module:
    - Set the frequency of the XADC module to 50MHz.
    - Select the mode of operation to be in **Event Mode**.
    - select channel **VAUX_15**
    - Under the alarm, uncheck all boxes.
6. Run the synthesis and implemenation of the design, then generate the bitstream.
7. Upload the bitstream to your FPGA board.

Usage

- **Project files:** The project follows a modular design approach with separate files implementing various components of the processor and XADC, ensuring better organization, scalability, and maintainability.
- **Memory Initialization:** The processor follows the Harvard architecture, with separate 64×32-bit block memories for instruction and data, each providing 256 bytes of storage. The instruction memory is preloaded with a program that functions as a voltmeter, displaying real-time voltage changes on the 16 LEDs and the 4-digit seven-segment display of the Basys 3 board.
- **Modifications and Enhancements:** Feel free to modify or enhance the CPU design. You could also initialize the instruction memory with any program you wish to run using the following available instruction set:
    - **Arithmetic and logic:** *and, or, add, sub, slt, addi, srl*
    - **Data transfer:** *lw, sw*
    - **Branch and jump:** *beq, j*

Architecture

The project is is structured into several Verilog modules that together form the MIPS single-cycle processor (CPU) with a integrated analog to digital converter XADC. below is a small description to the main modules that implement the architecture.

Testbench (file mytestbench.sv)

This module used for instantiating the clock module that generates the different clocks for both the CPU and XADC as well as the instantiating the CPU and XADC found inside module **top**.

Top module (file top.sv)

Instantiates the MIPS processor, Instruction memory as well as the memory mapping containing the *XADC, thermocoder* and the *Data memory*.

Module MIPS(files mips.sv)

This module interfaces the instantiation of the control Unit and the datapath.

Module controller (files controller.sv)

This is the control unit of the MIPS single-cycle processor, which is made up of the two instruction decoders namely: The main decoder and the alu decoder.

Datapath (file datapath.sv)

The datapath module is the core of the architecture, responsible for linking and coordinating all major components. It facilitates data flow between the processor's registers, ALU, memory, and control unit, ensuring proper execution of instructions.

Module memory map(file memory_map.sv)

The memory_map module manages the organization and access of memory within the system. It defines address mapping for instruction and data memory, ensuring efficient communication between the processor and memory components. This module instantiates the XADC wrapper for communication with the XADC, the thermocoder, as well as the Data memory.

Operation flow

- **Fetching:** instruction memory reads the instruction from the address supplied by the current **PC** address.
- **Decoding:** Extract instruction field and the control unit generates the appropriate signals required for instruction execution.

- **Execution:** The Arithmetic and Logic Unit **ALU** performs the various computations while the other components rout data as based on the generated control signals from the Control Unit.
- **Memory Operations:** The memory map module handles the **load and store** instructions for both the **data Memory** as well as the **memory-mapped IOs**.
- **Write Back:** Results from either the **ALU** or the memory map module is written back into register file(module regfile.sv).

Scalability and Enhancement

The design is optimized for single-cycle operation but can be enhanced with pipelining. Introducing pipeline registers between stages would transform it into a pipelined processor, boosting throughput and performance.

Authors

Tadouanla Guetchuin Billy

Sources

Book: Harris Digital Desgin and Computer architecture.

License