Implementation of 8-Bit RISC SPM Processor

Muhammad Burhan Ahmed, Javeria Razzaq, Muhammad Ali Raza

Air University, Islamabad

May 27, 2024

Abstract

This project aims to design an 8-bit RISC SPM processor capable of executing basic instructions. The project involves developing a simplified, yet efficient, 8-bit RISC processor that emphasizes the core concepts of instruction execution, memory hierarchy, and processor design. The processor utilizes a Stored Program Memory architecture, where instructions and data are stored in a single memory space, facilitating streamlined processing and reduced complexity.

Objectives

- Designing simplified RISC architecture.
- Implementation using Verilog hardware language.
- Ensuring correctness and performance through simulation.
- Achieving a single-cycle execution model for all instructions, optimizing for speed.

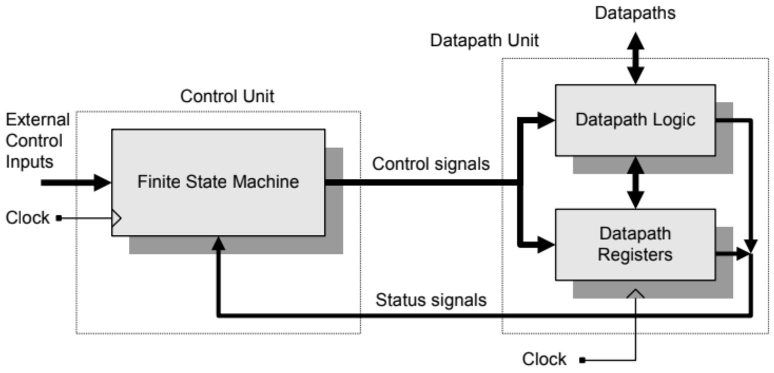
Software Used

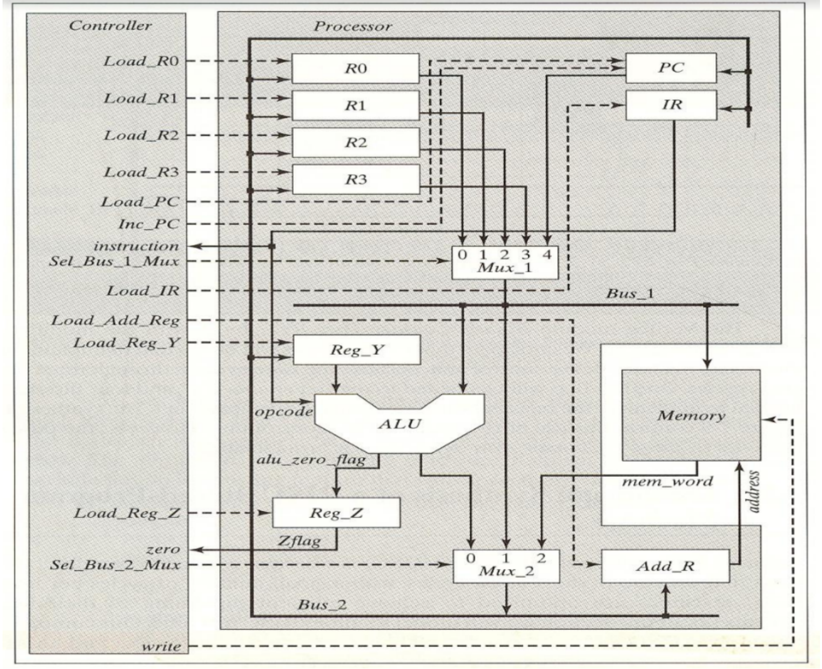
- Xilinx

Introduction

RISC Single Process Multiple Data (SPM) is a processor that executes a single instruction across multiple data elements. The architecture is designed for data-parallel operation commonly found in scientific computing and similar applications. It is used when multiple data need to be processed concurrently.

Architecture





Controller
Load_R0
Load_R1
Load_R2
Load_R3
Load_PC
Inc_PC
instruction
Sel_Bus_1_Mux
Load_IR
Load_Add_Reg
Load_Reg_Y
Load_Reg_Z
zero
Sel_Bus_2_Mux
write
Processor
R0
R1
R2
R3
PC
IR
0 1 2 3 4
Mux_1
Bus_1
Reg_Y
opcode
ALU
alu_zero_flag
Reg_Z
Zflag
0 1 2
Mux_2
Bus_2
Memory
mem_word
address
Add_R

RISC SPM Key Features

- A RISC store-program machine (SPM) consists of three functional units: a processor, a controller, and memory.
- Program instructions and data are stored in memory.
- Instructions are fetched from memory synchronously, decoded, and executed.

Instruction Types

- Short Instruction
- Long Instruction

S_idle	State entered after reset is asserted. No action.
S_fet1	Load the Add_R with the contents of the PC
S_fet2	Load the IR with the word addressed by the Add_R, Increment the PC
S_dec	Decode the IR Assert signals to control datapaths and register transfers.
S_ex1	Execute the *ALU* operation for a single-byte instruction, Conditionally assert the zero flag, Load the destination register

S_rd1	Load Add_R with the second byte of an RD instruction Increment the PC.
S_rd2	Load the destination register with memory[Add_R]
S_wr1	Load Add_R with the second byte of a WR instruction, Increment the PC.
S_wr2	Write memory[Add_R] with the source register
S_br1	Load Add_R with the second byte of a BR instruction Increment the PC.
S_br2	Load the PC with the memory[Add_R]
S_halt	Default state to trap failure to decode a valid instruction

Control Signals

The control signals for the RISC SPM processor include:

- Clock Signal (CLK): Synchronizes the operation of the processor.
- Reset (RESET): Resets the processor to a known state.
- Instruction Fetch (IF): Indicates fetching the next instruction from memory.
- Instruction Decode (ID): Signals that the fetched instruction is being decoded.
- Memory Read (MEM RD): Indicates reading data from memory.
- Memory Write (MEM WR): Signals writing data to memory.
- Register Write (REG WR): Indicates writing data to internal registers.
- ALU Operation (ALU OP): Specifies the operation performed by the ALU.
- Branch (BRANCH): Signals a branch instruction.
- Jump (JUMP): Indicates a jump instruction.
- Load (LOAD): Signals a load instruction.

Blocks of RISC SPM

The RISC SPM datapath is composed of the following components:

- Processing Unit: Address Register, Instruction Register, Program Counter, Multiplexers, ALU, Bus1, Bus2, Register file.
- Control Unit
- Memory Unit (RAM)

Processing Unit Components

- Register File: High-speed memory unit for data storage and manipulation.
- Instruction Register (IR): Holds the current instruction being executed.
- Program Counter (PC): Holds the address of the next instruction to be executed.
- Multiplexers (MUXes): Digital devices used to select and route input signals to a single output.
- ALU (Arithmetic Logic Unit): Performs arithmetic and logical operations on binary data.

Memory Unit

Memory Unit (RAM) is a component where data is stored and retrieved during program execution. It holds both program instructions and data actively being processed by the CPU. It is organized into addressable storage locations, each capable of holding a fixed number of bits or bytes of data.

Control Unit (CU)

The Control Unit coordinates and controls CPU operations, ensuring instructions are executed in the correct sequence and data is processed accurately. It performs functions like instruction decoding, sequencing, operand fetching, execution control, register transfer, and branch handling.

Verilog Module Code

The project involves writing Verilog code for the main datapath, processing unit, register file, instruction register, program counter, multiplexers, ALU, control unit, and memory unit. These modules are interconnected to form the complete RISC SPM processor.

RTL View

- Top Module: Orchestrates the digital system, interconnecting processing, control, and memory units.
- RAM: Handles read and write operations synchronized with the clock signal.
- Processor: Manages data flow, instruction execution, and register operations.
- ALU: Performs arithmetic and logical operations based on control signals.
- Control Unit: Manages control signals and state transitions for the processor.
- Multiplexers: Select and route input signals to appropriate outputs.
- Program Counter: Tracks the address of the next instruction.
- Instruction Register: Holds the instruction currently being executed.
- Address Register: Holds memory addresses for data operations.
- Zero Register: Implements a D flip-flop for control operations.

Test Bench

Thorough testing is essential to validate the correctness of the RISC SPM processor datapath, ensuring accurate instruction execution through various test cases.

Conclusion

Designing an 8-bit RISC SPM processor datapath using Xilinx involves systematic approaches, careful component design, thorough testing, and performance analysis. Adherence to RISC SPM architecture specifications and leveraging Xilinx tools are crucial for successful implementation.

References

- Christine Connolly. “Pros from touchics”. Industrial Robot: An International Journal (2008).
- Michael H Dickinson. “insight into mechanical design”. Proceedings of the National Academy of Sciences 96.25 (1999), pp. 14208–14209.
- Robert Rosen. “revisited”. The machine as metaphor and tool. Springer, 1993, pp. 87–100.
- Rene Romain Roth. “The foundation of nics”. Perspectives in bicine 26.2 (1983), pp. 229–242.