MIPS ISA 32 BIT PROCESSOR

PRESENTED TO DR. HOWAIDA ABDEL LATIF

Project Title: ***multi cycle processor that implements MIPS ISA 32-bit processor***

✓ ***Course:*** ***Hardware Design***

✓ **Instructor: [DR. Howida Abdellatif]**

✓ **Submitted by:**

No	Name	Academic number	Task completed
1	حمدي عماد حمدي	20812022100282	Memory +register file+A&B&ALUout+MDR&PC
2	دنيا عصام فتحي أحمد	20812022200264	ALU+Shift left +s_extended+MUX
3	خلود وليد عطية	20812022200595	Control unit
4	مازن ياسر رفعت	20812022100417	Connection

Table of Contents

1. Assignment Aim

The aim of this project is to design and implement a simple processor in VHDL, capable of executing basic instructions such as load, store, arithmetic operations, branching, and jump. The design includes all major components of a processor and follows the multicycle approach.

2. Problem Solution

The solution involves building a simplified multi-cycle processor architecture, implementing each component using VHDL, and simulating it in Active-HDL to verify functionality.

2.1 Inputs and Outputs

Component	Inputs	Outputs
control_unit	clk, reset, operation (instruction[31:26])	pc_write, pc_write_condition, IorD, memory_read, memory_write, memorytoregister, IR_write, ALUSrcA, RegWrite, RegDst, ALUSrcB, ALU_operation, pc_source
memory	address (MuxToAddress), writeData (RegBOut), memRead, memWrite, clk, enable	memData
register_file	clk, reset, read_reg1, read_reg2, write_reg (MuxToWriteReg), write_data (MuxToWriteData), reg_write	read_data1 (srcA), read_data2 (srcB)
registers (PC)	clk, reset, enable (OrToPC), input (MuxToPC)	output (PC_out)
registers (MDR)	clk, reset, enable = '1', input (memData)	output (MDRToMux)
registers (IR)	clk, reset, enable (IR_write), input (memData)	output (instruction)
registers (Reg_A)	clk, reset, enable = '1', input (srcA)	output (RegAout)

registers (Reg_B)	**clk, reset, enable = '1', input (srcB)**	**output (RegBOut)**
registers (ALU_OUT)	**clk, reset, enable = '1', input (ALUResultTOALUOut)**	**output (ALUOut)**
ALUControl	**funct (instruction[5:0]), ALUop**	**ALUSel (ALUControltoALU)**
ALU	**A (Mux4ToALU), B (Mux5ToALU), ALU_Sel**	**Result (ALUResultTOALUOut), Zero**
SignExtender	**InData (instruction[15:0])**	**OutData (SignExtenderOut)**
shift_left32	**input (SignExtenderOut)**	**output (shift_left32ToMux)**
shift_left26	**input (instruction[25:0])**	**output (jumpAddress[27:0])**
mux1 (2x1)	**A (PC_out), B (ALUOut), Sel (IorD)**	**Y (MuxToAddress)**
mux2 (2x1)	**A (instruction[20:16]), B (instruction[15:11]), Sel (RegDst)**	**Y (MuxToWriteReg)**
mux3 (2x1)	**A (ALUOut), B (MDRToMux), Sel (memorytoregister)**	**Y (MuxToWriteData)**
mux4 (2x1)	**A (PC_out), B (srcA), Sel (ALUSrcA)**	**Y (Mux4ToALU)**
mux5 (4x1)	**A (srcB), B (4), C (SignExtenderOut), D (shift_left32ToMux), Sel (ALUSrcB)**	**Y (Mux5ToALU)**
mux6 (3x1)	**A (ALUResultTOALUOut), B (ALUOut), C (jumpAddress), Sel (pc_source)**	**Y (MuxToPC)**

2.2 Design Description

This processor is designed as a multicycle MIPS-like processor and includes the following components:

- **Memory**: Used for instruction and data storage.
- **Register File A & B**: Hold source operands and temporary data.
- **ALU**: Executes arithmetic and logic operations.
- **MDR (Memory Data Register)**: Temporarily stores data read from or to be written to memory.
- **Shift Left Unit**: Used in branch instruction to shift address.

- **Control Unit**: Generates control signals depending on the current state and opcode.
- **MUX**: Selects between multiple data sources.
- **PC (Program Counter)**: Holds address of next instruction.
- **Sign-Extended**: Extends 16-bit immediates to 32-bit.

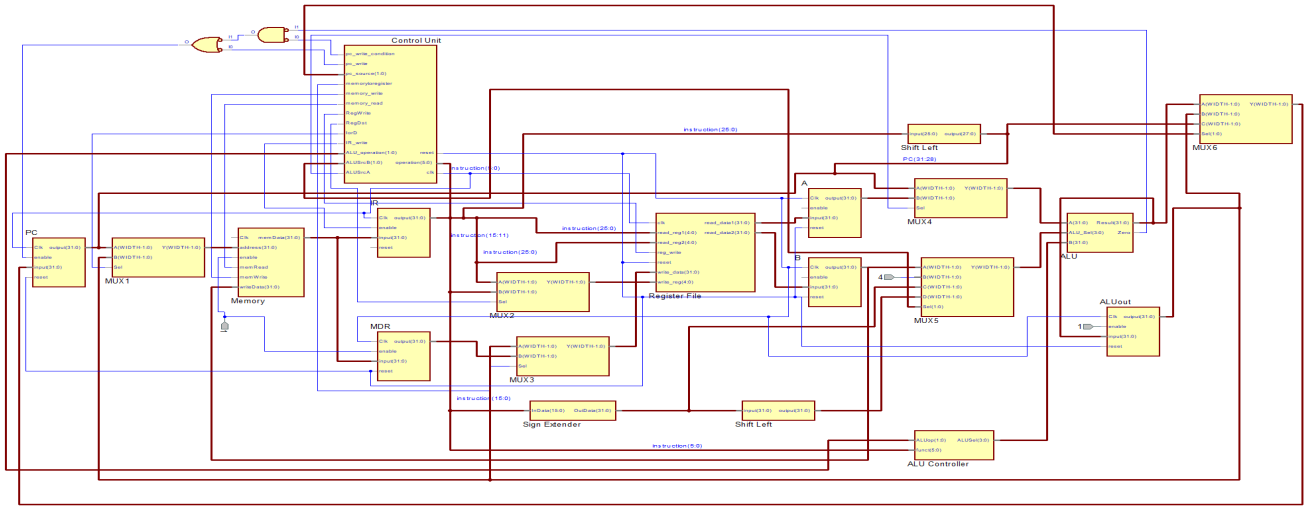
3.Implementation

3.1 Code Overview

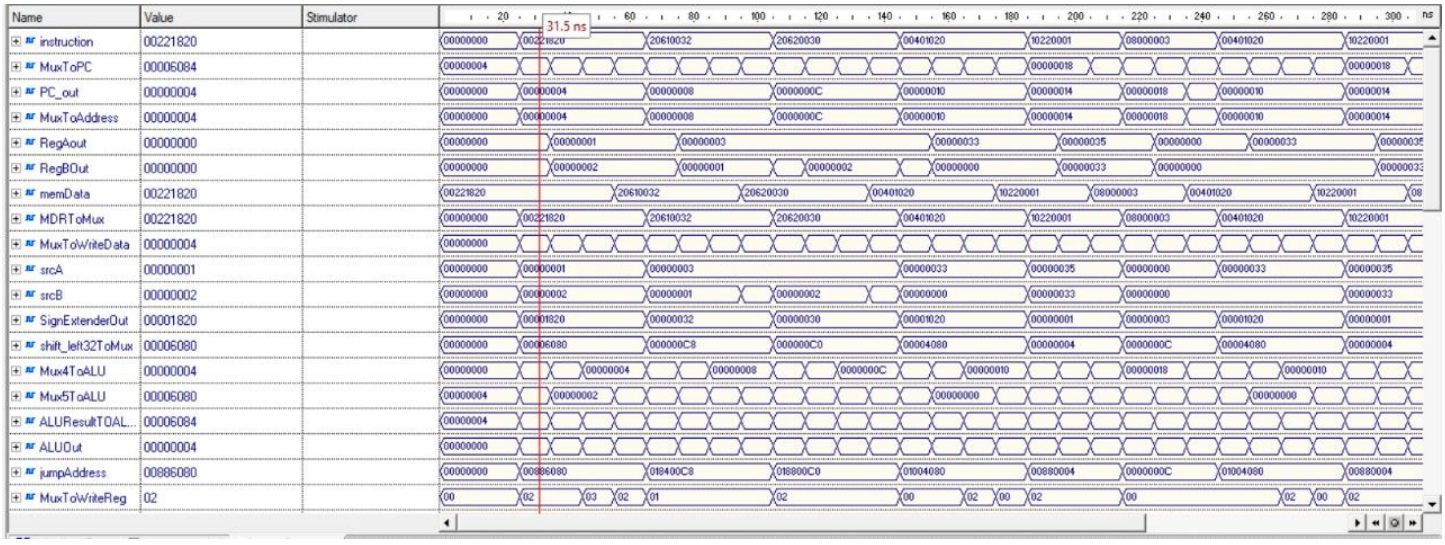
https://github.com/HamdiEmad/MIPS-ISA-32-bit-Microprocessor

https://drive.google.com/drive/folders/17P3WBPT8eUuOjN3-iCnyK-3rRcCPDJEx?usp=sharing

3.2 Schematic Diagram



3.3 Simulation Results



4. References

- *Course Lecture Slides*
- *https://github.com/david-palma/mips-32bit/tree/master/MIPS32_multi_cycle*
- *https://www.slideshare.net/slideshow/mips-implementation-29633476/29633476*
- *https://claude.ai/public/artifacts/e71731d4-93df-4940-8f78-630e97409063*
- *https://claude.ai/public/artifacts/af0bbfa2-103e-45e9-8090-7458443d0440*
- *"Computer Organization and Design" by David A. Patterson and John L. Hennessy*
- *"Digital Design and Computer Architecture" by David Money Harris and Sarah L. Harris*