RISC-V Single Cycle Processor – Project Documentation

1. Introduction

This project implements a single-cycle RISC-V processor in Verilog HDL. The design is modular and follows the standard microarchitecture for executing one instruction per clock cycle. The processor supports a subset of the RV32I instruction set, including arithmetic, logical, memory, and branch operations.

The system was simulated and synthesized using Xilinx Vivado, with hardware mapping designed for FPGA targets.

2. Microarchitecture Overview

The processor executes each instruction in a single cycle, which means:
- Instruction fetch, decode, execute, memory access, and write-back occur within one clock cycle.
- This reduces latency but increases combinational logic complexity.

Key Components:
- Program Counter (PC): Holds current instruction address.
- Instruction Memory: Stores instructions.
- Register File: 32 × 32-bit general-purpose registers.
- Sign Extension Unit: Extends immediate values.
- ALU: Performs arithmetic and logical operations.
- Data Memory: For load and store instructions.
- Control Unit (Main Decoder + ALU Decoder).
- Multiplexers (MUXes).

3. Instruction Formats Supported

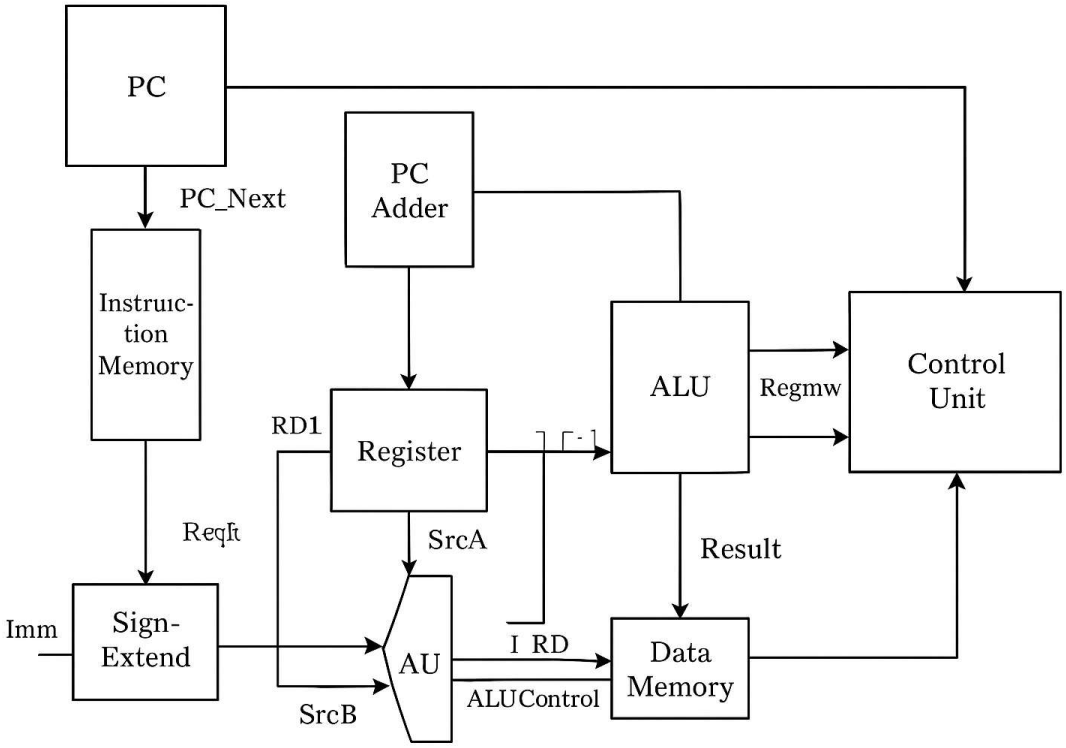
The CPU supports the following RISC-V instruction formats:
- R-type (Register–Register): add, sub, and, or, slt
- I-type (Immediate, Load): lw, addi, etc.
- S-type (Store): sw
- B-type (Branch): beq, bne, etc.

4. Datapath Design

4.1 Load Word (I-type): ALU computes address, memory fetches data, result written to register.
4.2 Store Word (S-type): ALU computes address, register value written to memory.
4.3 R-Type: ALU performs operation on two registers, result written back.
4.4 Branch: ALU compares registers, PC updated if condition met.

5. Block Diagram

The architecture diagram illustrates the datapath of the RISC-V single cycle processor. Refer to the included block diagram image in the repository.



6. Control Path

The Control Unit generates control signals:
- RegWrite: Enables register write.
- ALUSrc: Selects ALU input (register or immediate).
- ResultSrc: Selects ALU result or memory value.
- MemWrite: Enables data memory write.
- ImmSrc: Selects immediate type.
- ALUControl: Determines ALU operation.

7. Verilog Implementation

Design hierarchy:

```
Single_Cycle_Top.v
 ├── PC.v
 ├── Instruction_memory.v
 ├── PC_Adder.v
 ├── Register.v
 ├── ALU.v
 ├── Data_Memory.v
 ├── Control_Unit.v
 │    ├── Main_Decoder.v
 │    └── ALU_decoder.v
 ├── Mux.v
 └── Sign_Extend.v
```

8. Simulation & Synthesis

Simulation:

- Verifies PC, ALU, memory, and register updates.

Synthesis (Vivado):

```
read_verilog src/*.v
synth_design -top Single_Cycle_Top
opt_design
place_design
route_design
```

9. Advantages

- Educational clarity.
- Modular design.
- FPGA-friendly.

10. Limitations

- Single cycle → long critical path.
- No U-type or J-type instructions.
- Limited instruction subset.

11. Future Work

- Implement pipelining.
- Hazard detection & forwarding.

- Support U-type and J-type.
- Exception handling & interrupts.

12. Author & Contact

Suraj – NIT Patna
Email: surajsinha094@gmail.com
Phone: 9430283483