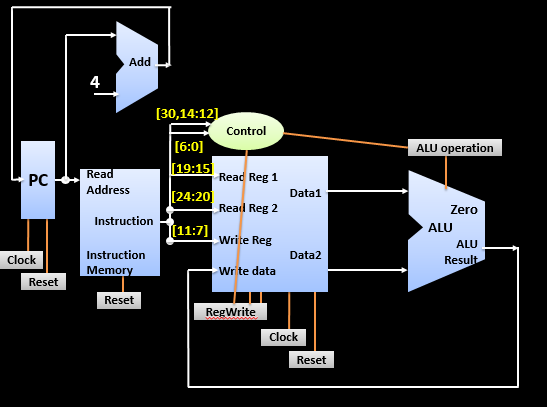
**Experiment No 7: Implementation of a Processor for R-type Instructions**

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Name** | **ID No** |
| **1** | **SURAJ S** | **2019AAPS0317H** |

# Exercise 7.2 Implement partial RISC V processor (Shown below), which executes only R-Type instructions. (Make use of the blocks implemented earlier)

This processor should support the following instructions (Please refer RISC V reference manual for the opcodes, func7 and func3 codes for instructions. Accordingly generate the correct ALU operation bits):

|  |
| --- |
| **add** |
| **sub** |
| **AND** |
| **OR** |
| **sll** |
| **srl** |



1. **Implement the main control unit and copy the image of Verilog code of Main control unit here.**

Answer: 

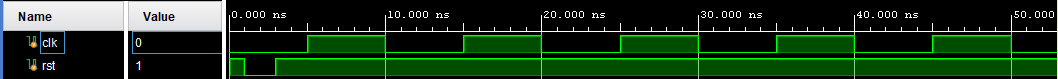
1. **Implement complete processor in Verilog (Instantiate all the datapath blocks and main control unit as modules). Copy the image of Verilog code of the processor here.**

Answer: ****

1. **Test the processor design by initializing the instruction memory with a set of instructions. List below the sequence of instructions you have used to initialize the instruction memory. Verify if the register file is changing according to the instructions. (Register file contains unknowns, you can initialize the register file or you can load values into the register file using li instruction specified earlier).**

Sequence of Instructions Implemented: **add t2, t0, t1   
sub s0, t0, t1   
and s1, t0, t1**

1. **Verify if the register file is getting updated according to your sequence of instructions (mentioned earlier).**

Copy verified **Register file** waveform here (show only the Registers that get updated, CLK, and RESET):   


1. **List the concepts you learnt from this experiment (Conclusions/Observations)**

Answer: In this lab we learnt to implement an R-type processor for the RISC-V 32 I instruction set. We implemented a control unit and a top processor unit that integrates all the different components.