COL216 Assignment-2 Part 5

by Harshit Mawandia 2020CS10348

Report:

I have made 9 files + 1 testbench.

1. ALU.vhd
2. register.vhd
3. Memory.vhd
4. programCounter.vhd
5. flags.vhd
6. condChecker.vhd
7. processor.vhd
8. FSM.vhd
9. Shifter.vhd

ALU, Register, Memory, PC, flags, Shifter and ConditionChecker are components, while FSM and Processor are used for binding logic and gluing the components together.

In addition to the previous report, this submission contains the component called Shifter. It performs all the necessary shifting operations as and when necessary. Implementing all the 4 types of shift, LSL, LSR, ASL, ROR.

This required some changes in the Processor.vhd and FSM.vhd file as well. Processor now has some more signals and UUT, while there has been a significant change made to the FSM file. Our State Machine now has 5 more states to handle all the necessary shifting conditions as necessary.

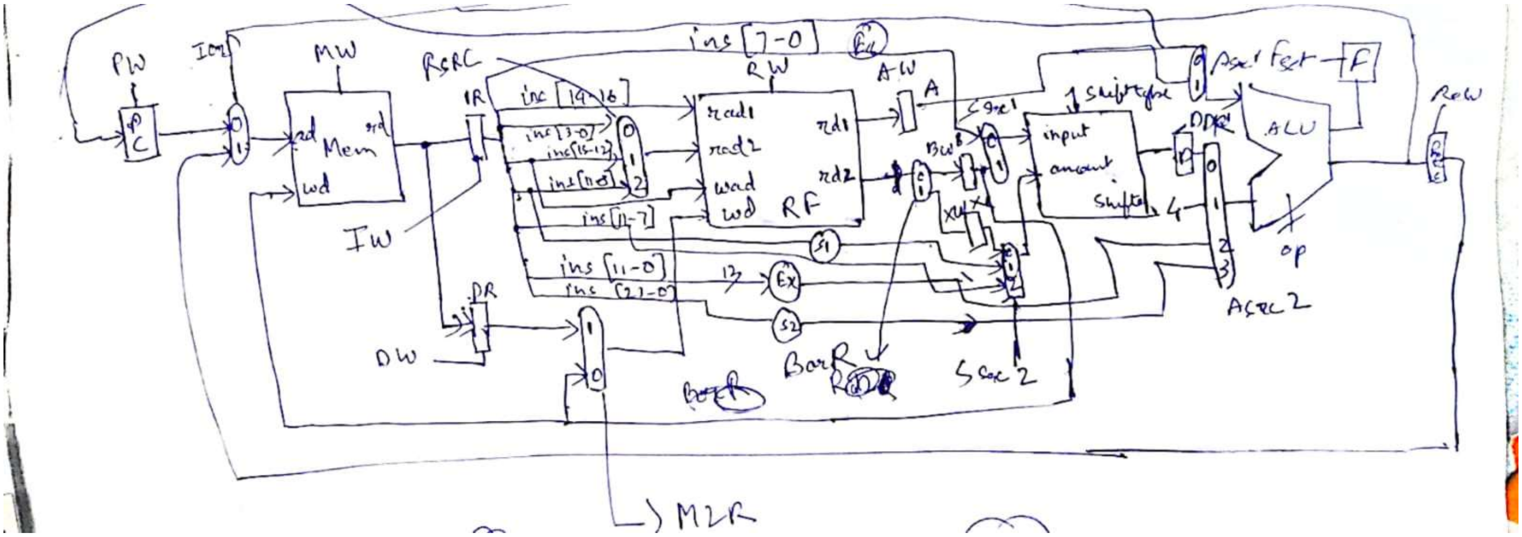
I have used the following libraries:

library IEEE; use ieee.NUMERIC_STD.all; use

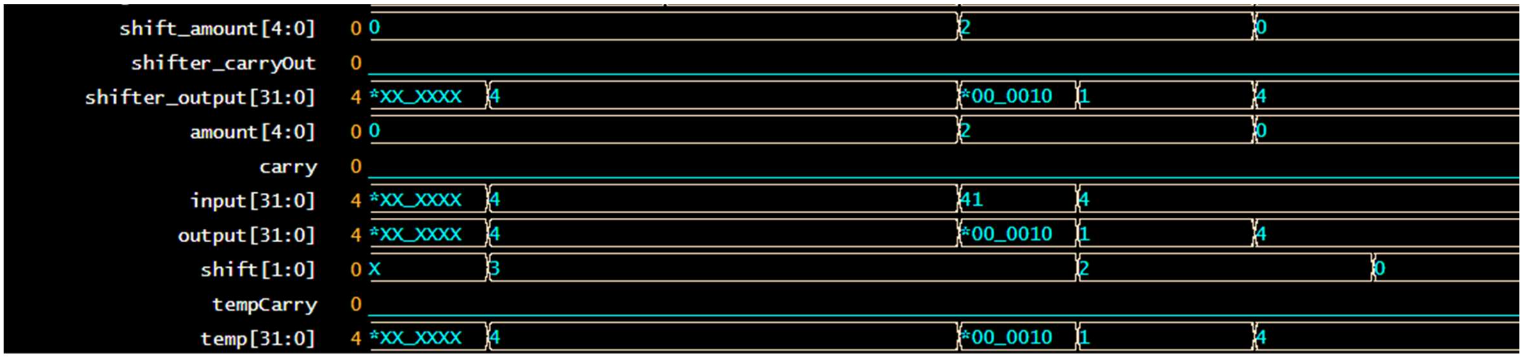
IEEE.STD_LOGIC_1164.ALL; use

IEEE.STD_LOGIC_UNSIGNED.A

LL;



This is the new Multi Cycle DataPath that I have used in my code


shift_amount[4:0] 0 0 2 0
shifter_carryOut 0
shifter_output[31:0] 4 *XX_XXXX 4 *00_0010 1 4
amount[4:0] 0 0 2 0
carry 0
input[31:0] 4 *XX_XXXX 4 41 4
output[31:0] 4 *XX_XXXX 4 *00_0010 1 4
shift[1:0] 0 X 3 2 0
tempCarry 0
temp[31:0] 4 *XX_XXXX 4 *00_0010 1 4

The shifter gives the correct results
As you can see Shifter implements ROR shift on 4 to give the answer 1 which is correct

The program ends when the instruction (IR) becomes 0, we can also use a reset state to set all signals to 0 after this point.