ASSIGNMENT 4

SATWIK BANCHHOR (2018CS10385)
DIWAKAR PRAJAPATI (2018CS10330)

AIM - Implement the MIPS Processor made in Assignment 3 on FPGA.

DESIGN DETAILS :

In assignment 3 we used 2D array(std_logic_vector) of 32 x 4096 as memory. We initialised the array using a text file. In this assignment, we are initialising the memory using .coe file (coefficient file).

We have a BRAM module (Block Memory Ram) which takes clock, read enable, write enable, data input, data output, address, to read instructions and; read and write data at a particular address.

IMPLEMENTATIONS:

We have made states for fetching instructions: idle, fetch, fetched_lw, fetched_sw, end_fetch.

Initially, the state is at idle and goes to fetch state for fetching the instruction, if the instruction is a ZERO instruction then it goes to end_fetch state, if the opcode is load then it goes to fetched_lw denoting that the loading word from memory is to be executed. Similar is the case with store word.

We have made states for executing instruction: idleex, execute, load, store, end_execute.

Initially, the state is at idleex, and goes to to execute instruction without any condition, since we know that execution starts after the first cycle. In execute state, we are executing various types of instructions. The load state makes sure that we are going to read data in the next cycle using flag value. The store state makes sure that we are going to write data in the next cycle using flag value. It is due to this reason that load and store take two cycles to execute, and all the other instructions takes only one cycle.

If we try to write data to memory at any index between 0 and 999 or any index greater than 4095, then the program terminates.

TEST CASE:

Initially Register 1 contains 8 and Register 2 contains 2.

INPUT	OUPUT
add 2 0 1 sub 3 0 1 sll 4 0 1 srl 5 0 1 sw 5 6 1000 lw 6 6 1000 add 6 6 6 add 7 6 5 add 7 7 5 sub 8 8 7 sub 9 9 8 sll 9 9 1 srl 10 9 1 sw 7 31 1001 lw 11 31 1001 lw 12 31 1000 add 11 12 11	Register 11 contains 20

The machine code for the above input which is to be given in coe file as input is given below. This test case contains various cases of sw->lw, sw->lw->lw:

00000000000000010001000000100000,
00000000000000010001100000100010,
00000000000000000010000001000000,
00000000000000000010100001000010,
10101100110001010000001111101000,
10001100110001100000001111101000,
00000000110001100011000000100000,
00000000110001010011100000100000,
00000000111001010011100000100000,
00000001000001110100000000100010,
00000001001010000100100000100010,
00000000000010010100100001000000,
00000000000010010101000001000010,
10101111111001110000001111101001,
10001111111010110000001111101001,
10001111111011000000001111101000,
00000001100010110101100000100000;

We are displaying the register content (lower 16 bits only) of the register on which the last operation was done. We are also displaying the no of cycles taken to execute the program using seven segment display.

Controlling the display is done using switch button.