Report

COL216 Assignment-4

CHIRAG MOHAPATRA

2018CS50403

VISHAL BINDAL

2018CS50425

- In this assignment, we designed a block memory RAM and input instructions into it using a .coe file. Then, we read the instructions one by one and executed them.
- Our program takes one clock cycle for add, sub, sll and srl instructions and two clock cycles for lw and sw instructions. We have defined different states to achieve this in synchronous process .
- On the seven segment display , we print either the clock cycle count or the lower 16 bits of the last instruction to be executed . They can be toggled using a switch .
- For our main process , we have used a 1 kHz clock and for the memory , we have provided the 100 Mhz clock . This was done after extensive testing and observing that the memory takes a bit more time. This ensures that dina and douta are updated when addra is changed, before the next clock cycle of the main process.

Our test cases are:

Test case no.	Instruction (assembly/words)	Instruction(machine)	Final Output
1	store $0 in 1024($1) add $0,$0,$0 sub $1 , $0, $1 load $2, 1022($1)	10101100001000000000010000000000, 00000000000000000000000000100000, 00000000000000010000100000100010, 10001100001000100000001111111110;	2
2	store $0 in 1024($1)	10101100001000000000010000000000,	2

	sll $0,1 sub $1 , $0, $1 load $2, 1022($1)	00000000000000000000000001000000, 00000000000000010000100000100010, 10001100001000100000001111111110;	
3	store $0 in 1024($1) sll $0,2 srl $0,1 sub $1 , $0, $1 load $2, 1022($1)	10101100001000000000010000000000, 00000000000000000000000010000000, 00000000000000000000000001000010, 00000000000000010000100000100010, 10001100001000100000001111111110;	2

4	store $0 in 1000($0)
	store $1 in 1001($0)
	add $2, $1, $0
	sub $3, $1, $0
	store $2 in 1003($1)
	store $3 in 1004($1)
	load 1000($0) in $4
	load 1001($0) in $5
	load 1002($0) in $6
	load 1003($0) in $7
	add $6, $6, $7
	add $5, $5, $6
	add $4, $5, $4
		10101100000000000000001111101000,
		10101100000000010000001111101001,
		00000000001000000001000000100000,
		00000000001000000001100000100010,
		10101100001000100000001111101011,
		10101100001000110000001111101100,
		10001100000001000000001111101000,
		10001100000001010000001111101001,
		10001100000001100000001111101010,
		10001100000001110000001111101011,
		00000000110001110011000000100000,
		00000000101001100010100000100000,
		00000000101001000010000000100000;
			5

5	sll $0, $0, 1 sll $2, $0, 2 sll $2, $2, 4	00000000000000000000000001000000, 00000000000000000001000010000000, 00000000000001000010001000000000;	256
6	sll $0, $0, 10 srl $2, $0, 5	00000000000000000000001010000000, 00000000000000000001000101000010;	64
7	sll $0, $0, 3 add $0, $0, $1 srl $0, $0, 1	00000000000000000000000011000000, 00000000000000010000000000100000, 00000000000000000000000001000010;	8
8	sub $0, $1, $0 sll $0, $0, 5	00000000001000000000000000100010, 00000000000000000000000101000000;	32

9	sll $2, $0, 14
	sll $1, $0, 13
	sll $0, $0, 5
	add $1, $0, $1
	add $2, $2, $1
		00000000000000000001001110000000,
		00000000000000000001011101000000,
		00000000000000000000000101000000,
		00000000000000010000100000100000,
		00000000010000010001000000100000;
			2^15 +
			2^14 + 2^6