CS223- Hardware Laboratory

CS223-MINIPROJECT 3

MINI PROJECT - 3

PRESENTED BY: MRITUNJAY KUMAR

1401CS27

Testing Code

	Machine	Address	Assembly	
main:	20020005	0	addi $2, $0, 5	
	2003000C	4	addi $3, $0, 12	
	2067FFF7	8	addi $7, $3, -9	
	00E22025	c	or $4, $7, $2	
	00642824	10	and $5, $3, $4	
	00A42820	14	add $5, $5, $4	
	10A7000A	18	beq $5, $7, end	
	0064202A	1c	slt $4, $3, $4	
	10800001	20	beq $4, $0, around	
	20050000	24	addi $5, $0, 0	
around:	00E2202A	28	slt $4, $7,$2	
	00853820	2c	add $7, $4,$5	
	00E23822	30	sub $7, $7, $2	
	00E23827	34	nor $7, $7, $2	(Added new)
	00E23826	38	xor $7, $7, $2	(Added new)
	00E53827	3C	nor $7, $7, $5	(Added new)
	20E70005	40	addi $7, $7, 5	(Added new)
	30E70005	44	andi $7, $7, 5	(Added new)
	00E2381A	48	div $7, $7, $2	(Added new)
	AC670044	4C	sw $7, 68($2)	
	8C020050	50	lw $2, 80($0)	
	08000017	54	j end	(Updated)
	20020001	58	addi $2, $0, 1	
end:	AC020054	5C	sw $2, 84($0)	

Instructions added to extend the single cycle CPU architecture:

- XOR
- DIV
- NOR
- ANDI

XOR:

R-Type Instruction.
Opcode: 100110

Changes:

In aludec (input [5:0] funct, input   [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b100110: alucontrol <= 4'b0101; // XOR *

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b101: result <= a ^ b; //xor *

DIV:

R-Type Instruction.
Opcode: 011010

Changes:

In aludec (input [5:0] funct, input   [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b011010: alucontrol <= 4'b0100; // DIV *

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b100: result <= a / b; //div *

NOR:

R-Type Instruction.
Opcode: 100111

Changes:

In aludec (input [5:0] funct, input   [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b100111: alucontrol <= 4'b0110; // NOR *

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b110: result <= ~OR; //nor *

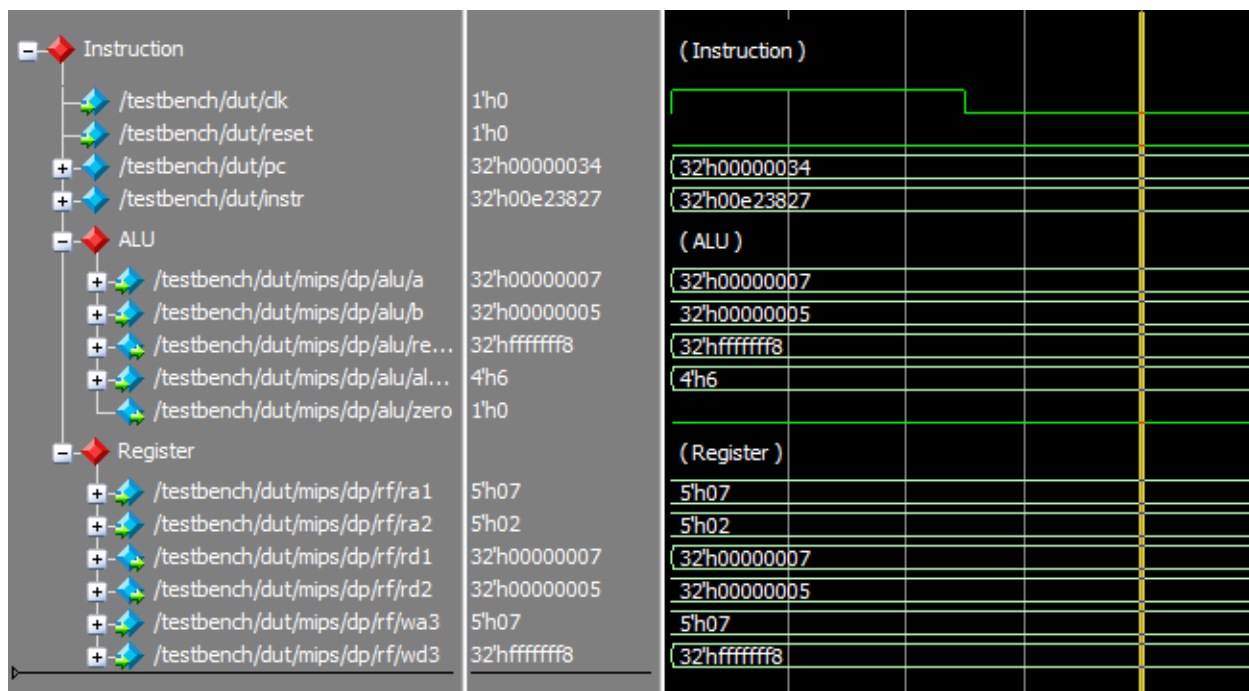***ANDI:***

I-Type Instruction.
Opcode: 100110

Changes:

In aludec (input [5:0] funct, input   [1:0] aluop, output reg [3:0] alucontrol);

Utilized spare 2'b11 aluopcode :: 2'b11: alucontrol <= 4'b0000; // andi *

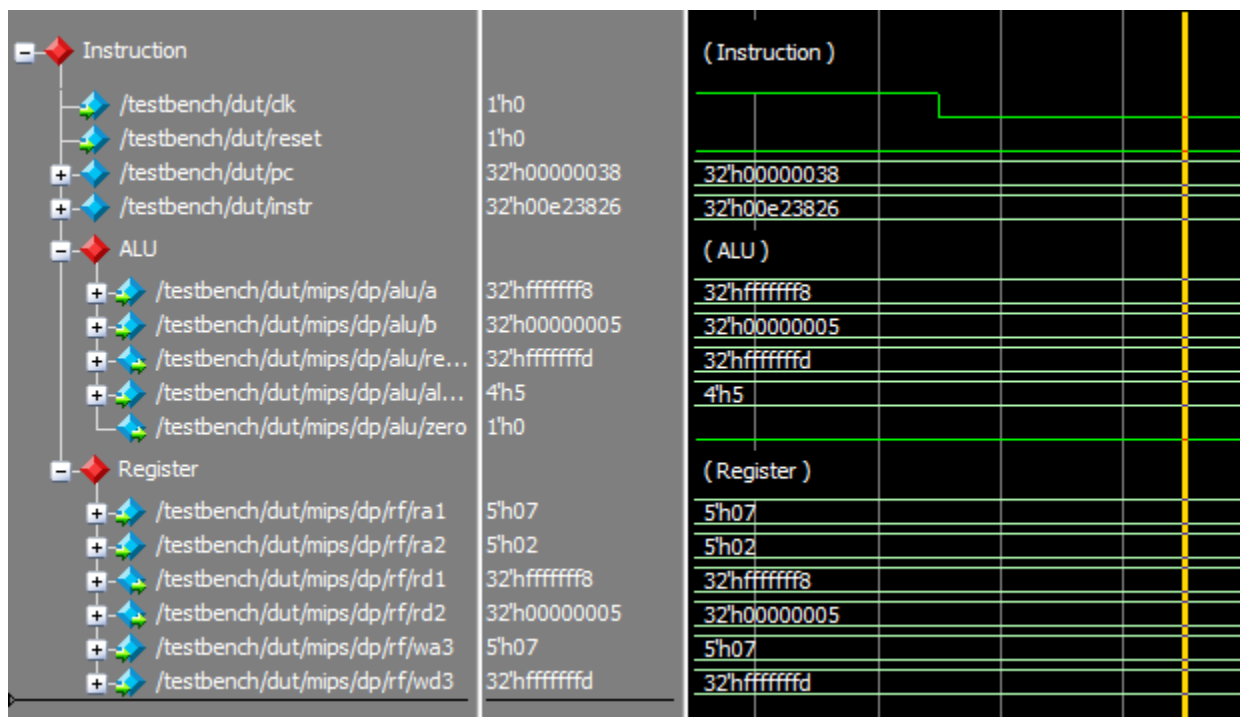# Wave of New Updated Instructions used by MIPS single-cycle.

1.

00E23827   34   nor $7, $7, $2   (Added new)
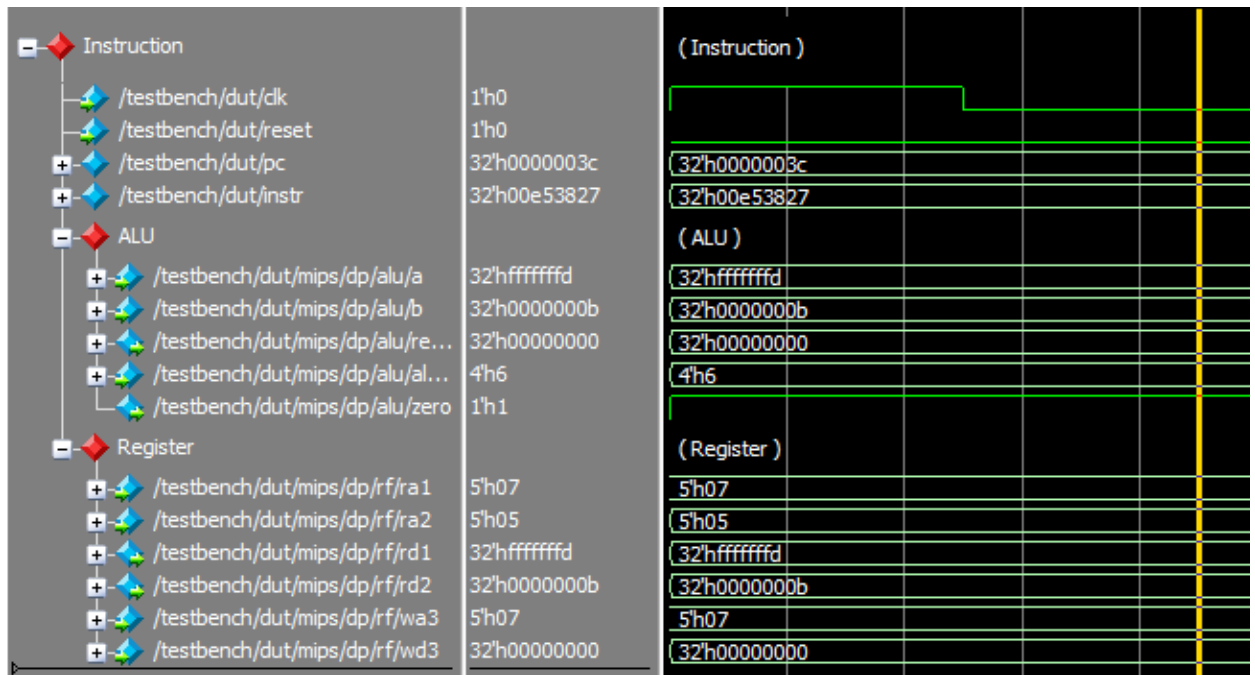

2.

00E23826   38   xor $7, $7, $2   (Added new)

3.

00E53827   3C   nor $7, $7, $5   (Added new)
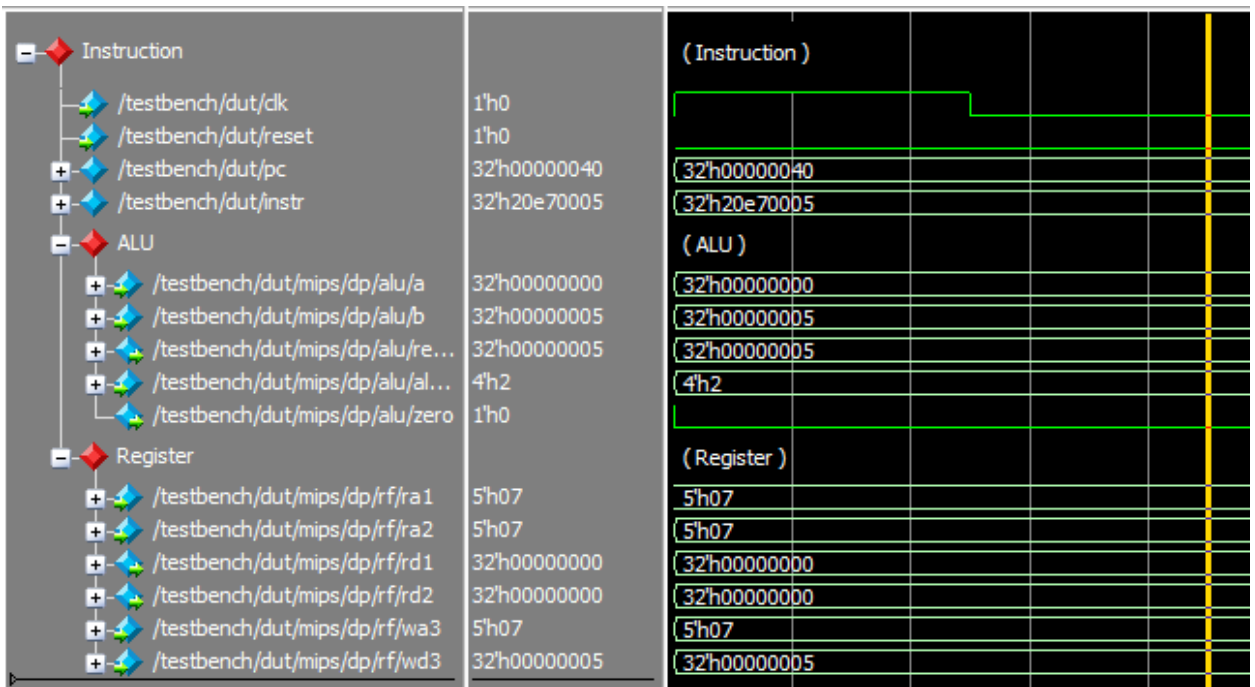

4.

20E70005   40   addi $7, $7, 5   (Added new)
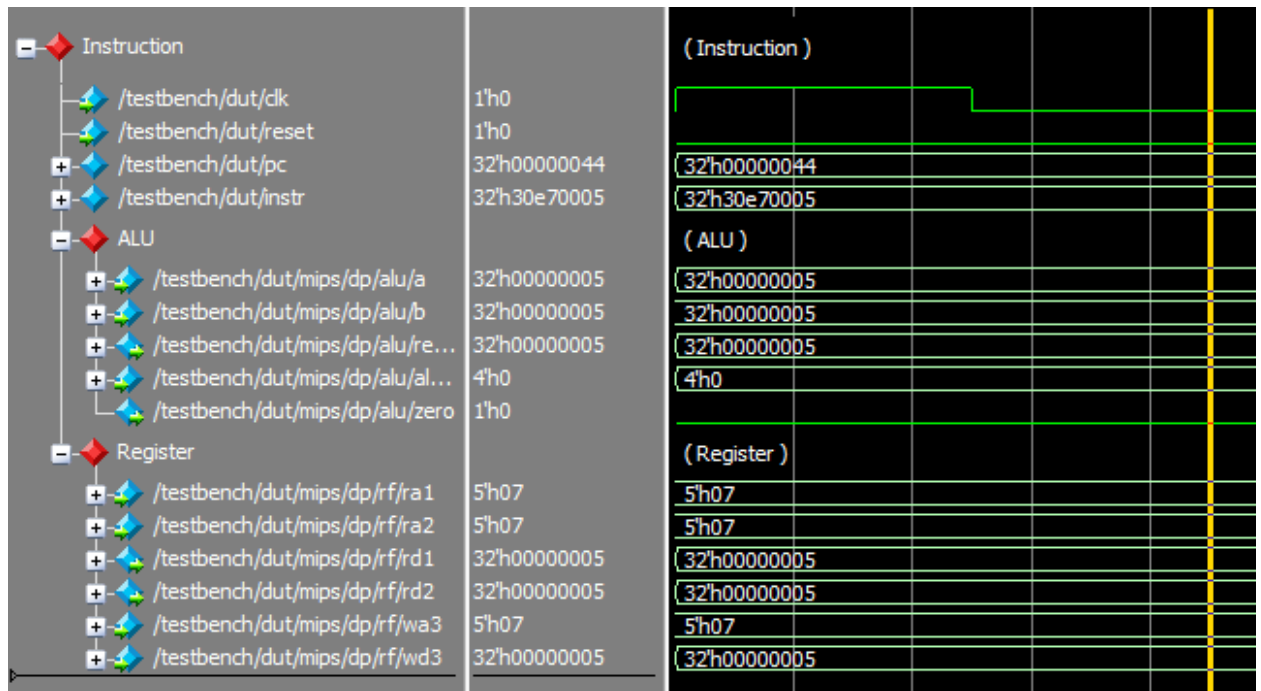
5.

        30E70005   44   andi $7, $7, 5   (Added new)
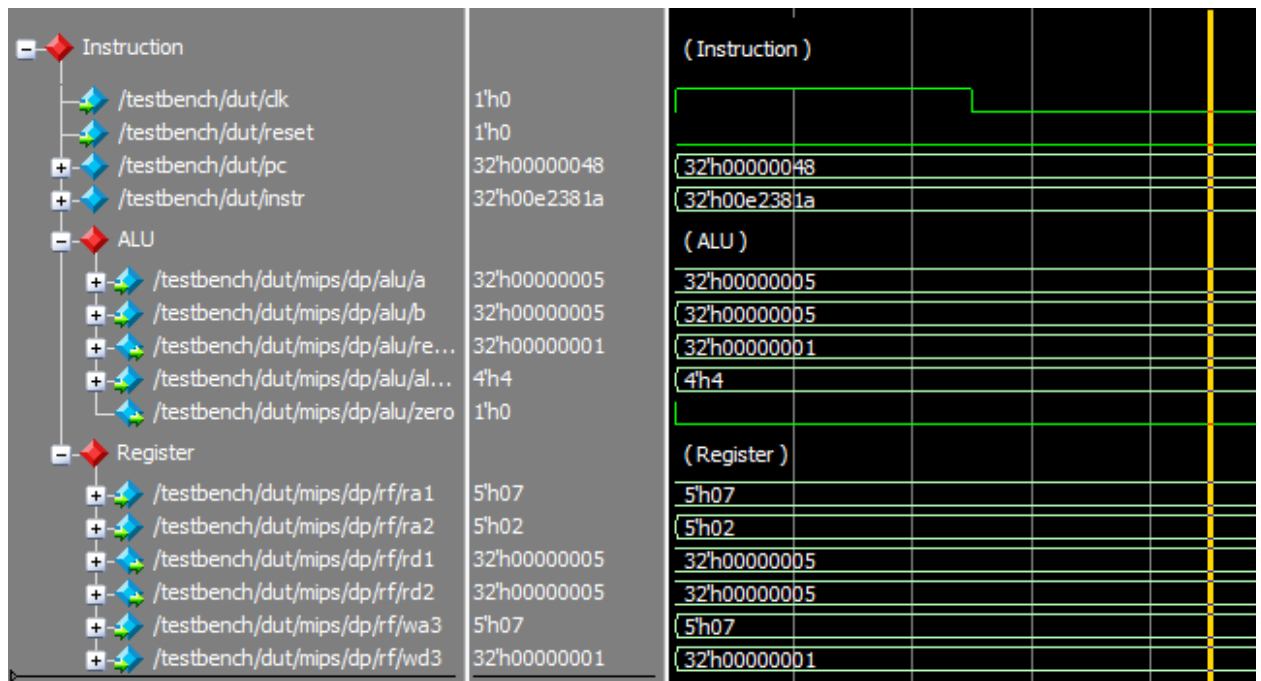

6.

        00E2381A   48   div $7, $7, $2   (Added new)
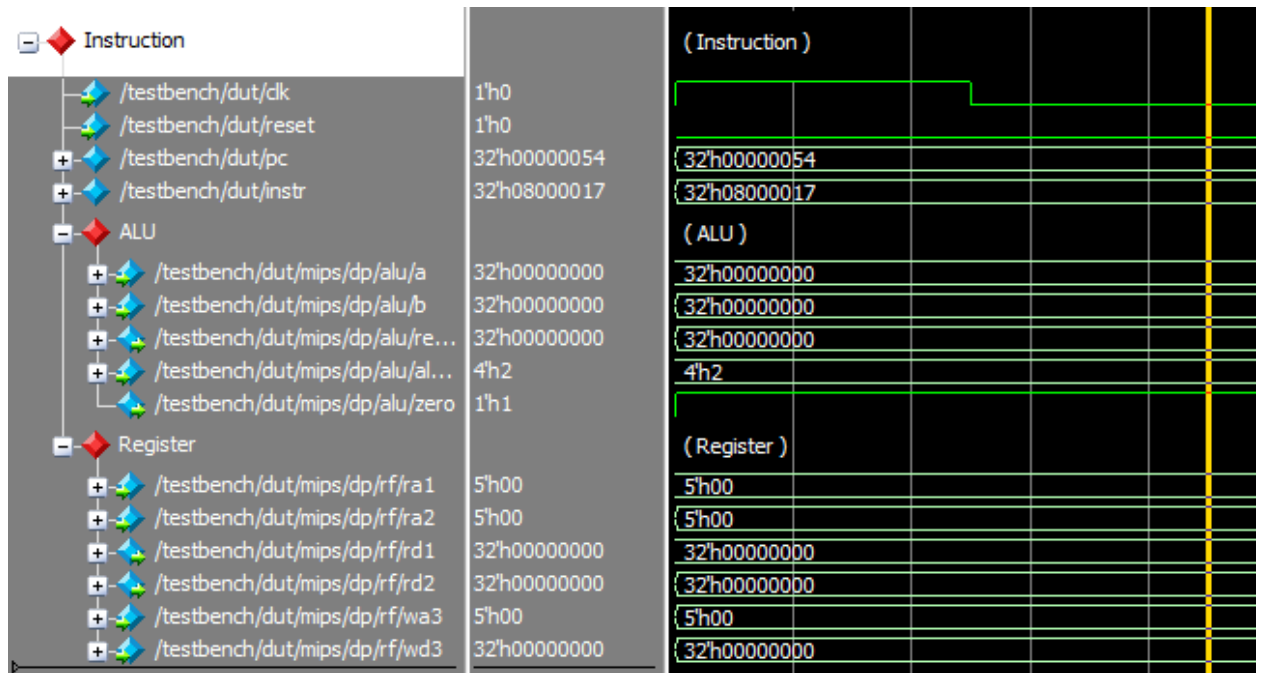
7.

08000017   54   j end   (Updated)
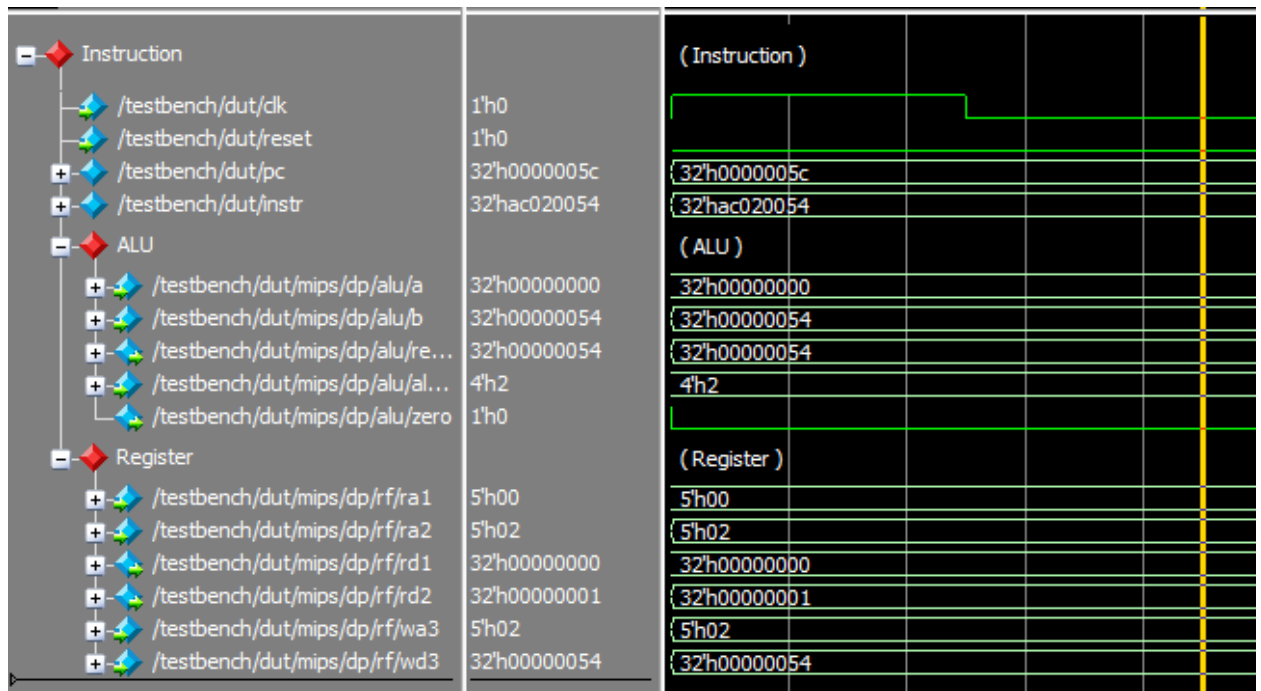

8.

end: *AC020054   5C   sw $2, 84($0)*

THANK YOU.