CSE331
ASSIGNMENT 3
REPORT

Ahmet Ergani

161044011

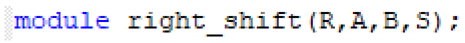
Summarization:

I have implemented a Controller unit and a Register module and modified the right shift module in the ALU in the previous assignment. Then I designed a single-cycle datapath with the Controller, the Register module, the ALU from previous assignment and 2 multiplexers.

Explanation for modules:

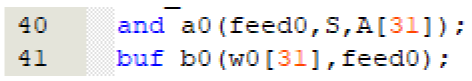
ALU:

I added a new parameter to the right shift module in the ALU.



```
module right_shift(R,A,B,S);
```

I determine whether I should feed the result with a “0” or the most significant bit by using an AND gate

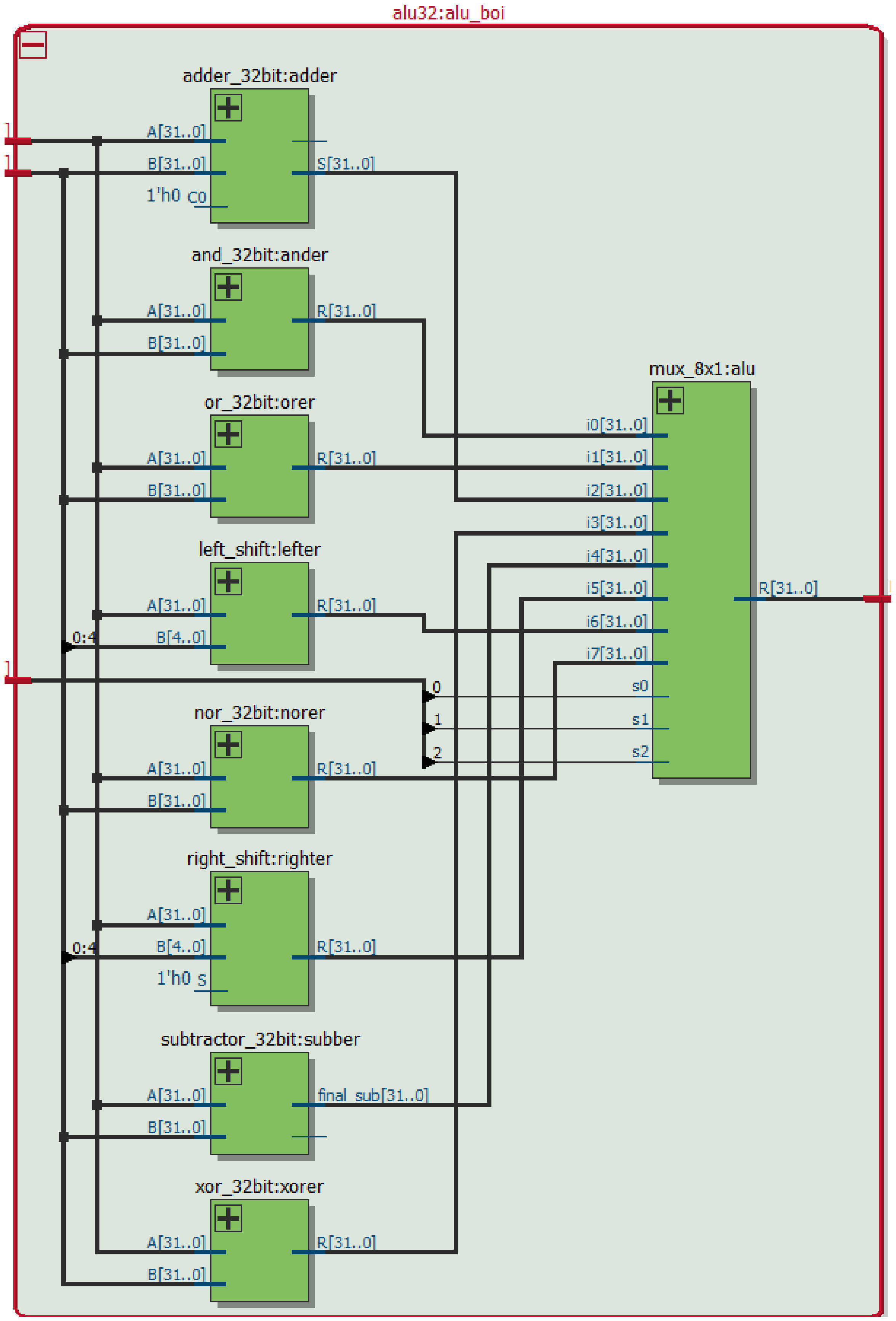


```
and a0(feed0,S,A[31]);
buf b0(w0[31],feed0);
```

(feed0 is what I am gonna feed the result with)

For this assignment the ALU always sends “0” as selection bit to right shift module

Remaining modules and the design of the ALU are the same as the previous assignment


alu32:alu_boi
adder_32bit:adder
A[31..0]
B[31..0]
1'h0 C0
S[31..0]
and_32bit:ander
A[31..0]
B[31..0]
R[31..0]
or_32bit:orer
A[31..0]
B[31..0]
R[31..0]
left_shift:lefter
A[31..0]
0:4
B[4..0]
R[31..0]
nor_32bit:norer
A[31..0]
B[31..0]
R[31..0]
right_shift:righter
A[31..0]
0:4
B[4..0]
1'h0 S
R[31..0]
subtractor_32bit:subber
A[31..0]
B[31..0]
final_sub[31..0]
xor_32bit:xorer
A[31..0]
B[31..0]
R[31..0]
mux_8x1:alu
i0[31..0]
i1[31..0]
i2[31..0]
i3[31..0]
i4[31..0]
i5[31..0]
i6[31..0]
i7[31..0]
s0
s1
s2
0
1
2
R[31..0]

Controller:

5 output bits are expected from Controller unit

-sltu signal

-shift signal

-3 bit ALUop signal

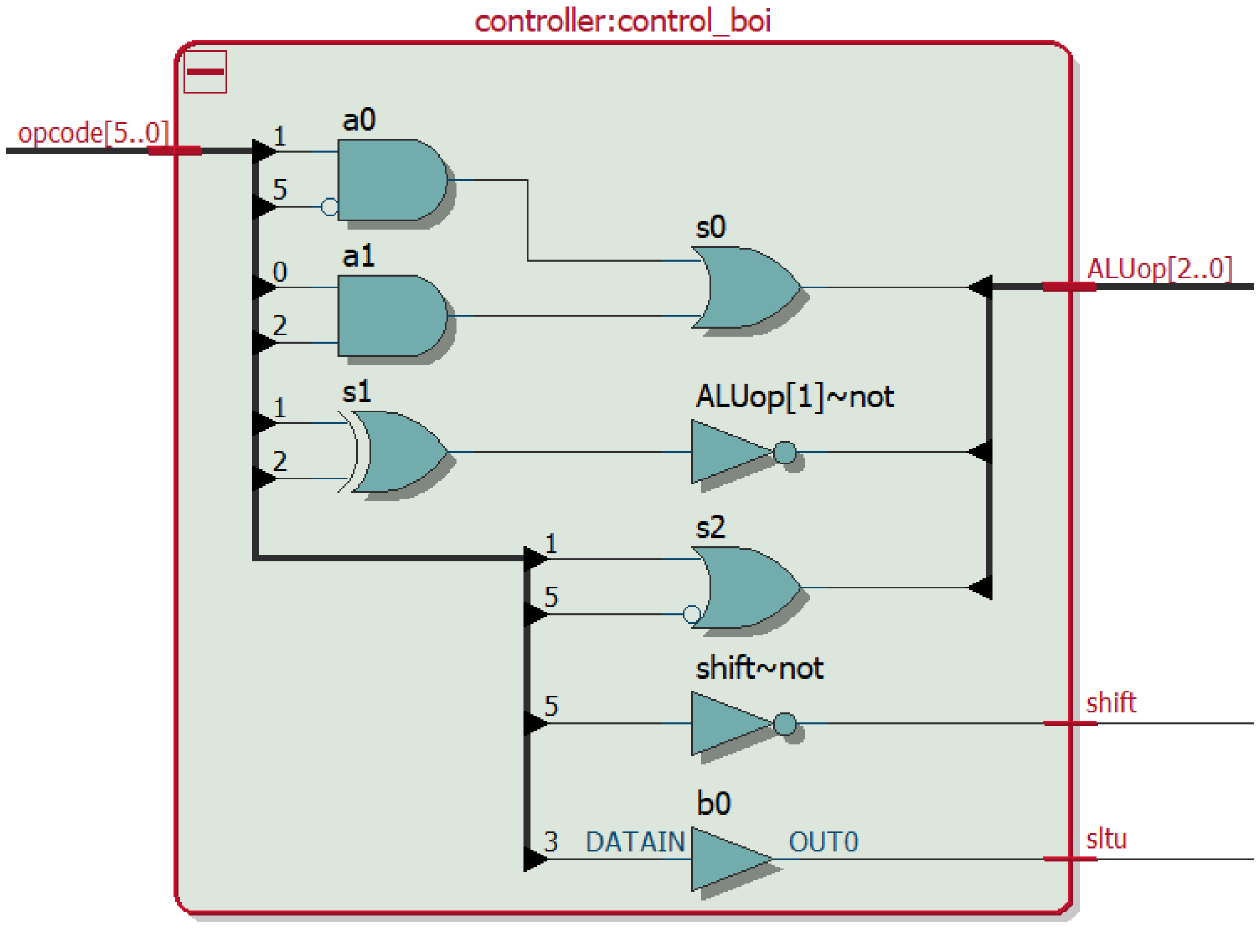
shift = func[5]' because func[5] is only "0" at shift cases

sltu = func[3] because func[3] is only "1" at sltu case

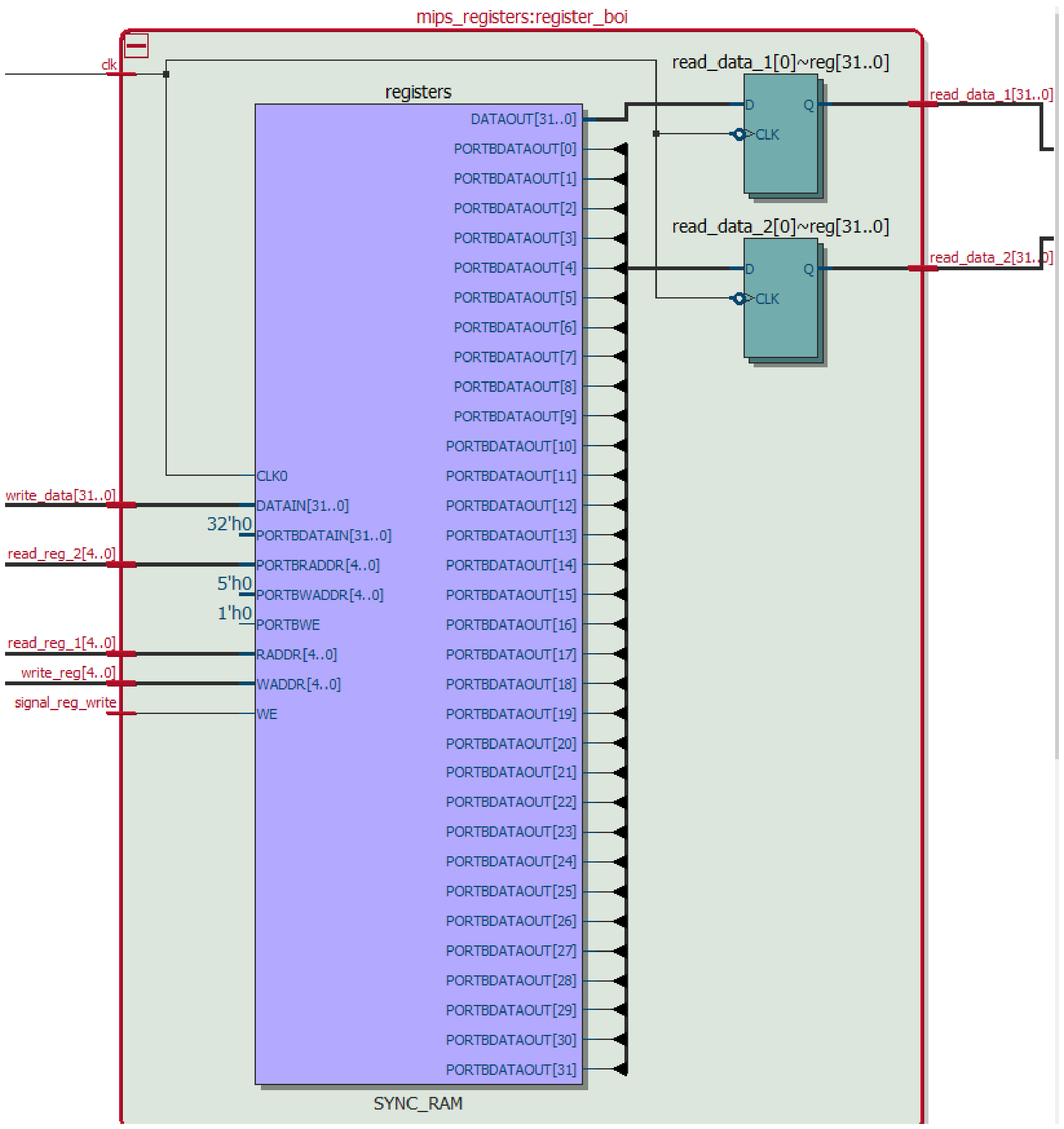
S0 = f5'f1 + f2f0

S1 = f1 XNOR f2

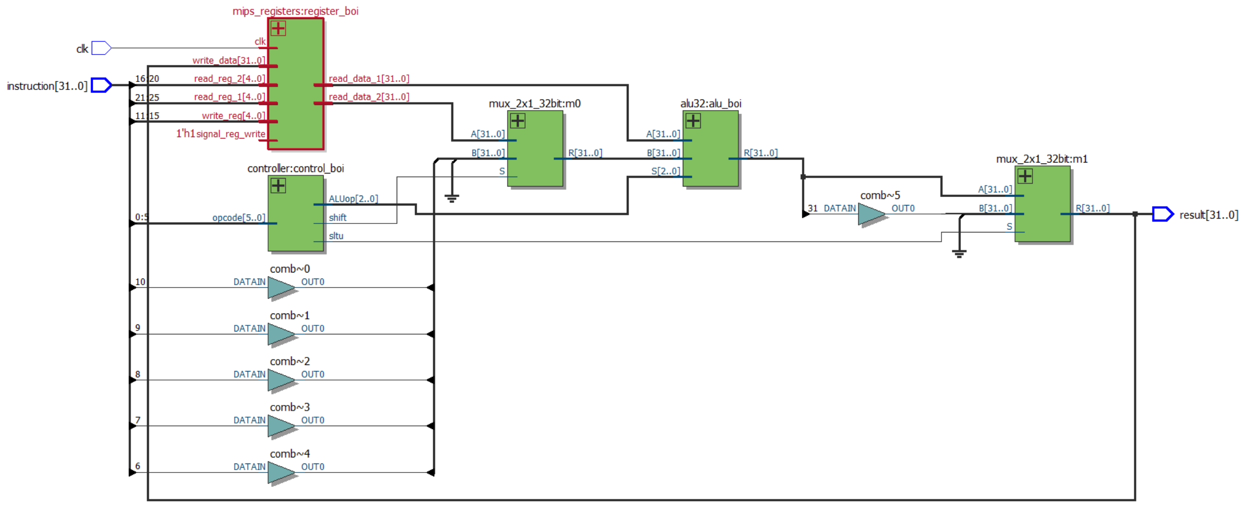
S2 = f5' + f1



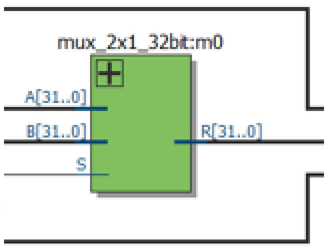
Register:


mips_registers:register_boi
clk
registers
DATAOUT[31..0]
PORTBDATAOUT[0]
PORTBDATAOUT[1]
PORTBDATAOUT[2]
PORTBDATAOUT[3]
PORTBDATAOUT[4]
PORTBDATAOUT[5]
PORTBDATAOUT[6]
PORTBDATAOUT[7]
PORTBDATAOUT[8]
PORTBDATAOUT[9]
PORTBDATAOUT[10]
PORTBDATAOUT[11]
PORTBDATAOUT[12]
PORTBDATAOUT[13]
PORTBDATAOUT[14]
PORTBDATAOUT[15]
PORTBDATAOUT[16]
PORTBDATAOUT[17]
PORTBDATAOUT[18]
PORTBDATAOUT[19]
PORTBDATAOUT[20]
PORTBDATAOUT[21]
PORTBDATAOUT[22]
PORTBDATAOUT[23]
PORTBDATAOUT[24]
PORTBDATAOUT[25]
PORTBDATAOUT[26]
PORTBDATAOUT[27]
PORTBDATAOUT[28]
PORTBDATAOUT[29]
PORTBDATAOUT[30]
PORTBDATAOUT[31]
CLK0
write_data[31..0]
DATAIN[31..0]
32'h0
PORTBDATAIN[31..0]
read_reg_2[4..0]
PORTBRADDR[4..0]
5'h0
PORTBWADDR[4..0]
1'h0
PORTBWE
read_reg_1[4..0]
RADDR[4..0]
write_reg[4..0]
WADDR[4..0]
signal_reg_write
WE
SYNC_RAM
read_data_1[0]~reg[31..0]
D
Q
CLK
read_data_1[31..0]
read_data_2[0]~reg[31..0]
D
Q
CLK
read_data_2[31..0]

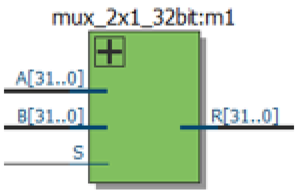
Resulting datapath that uses these modules is:



I divided the instruction as necessary and connected them to the appropriate ports.



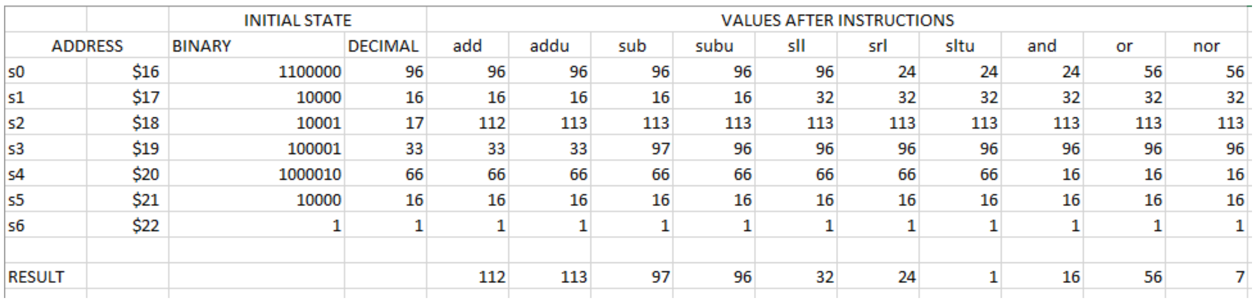
This mux is for shift cases



This mux is for sltu case

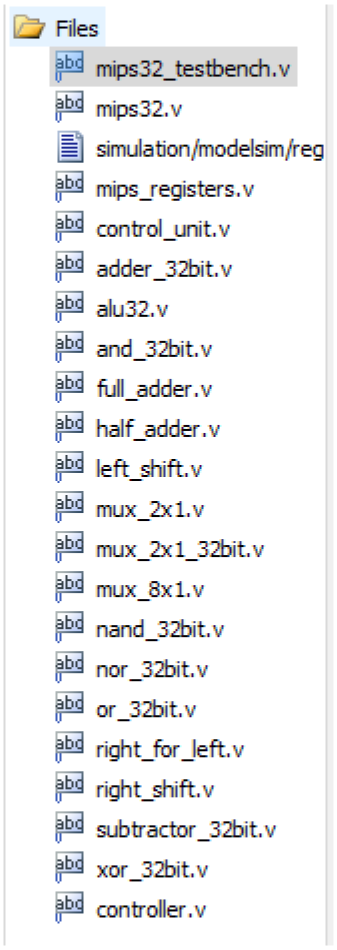
TEST CASE

```
add  s2 s0 s1
addu s2 s2 s6
sub  s3 s2 s1
subu s3 s3 s6
sll s1 s1 1
srl s0 s0 2
sltu s6 s0 s1
and s4 s5 s0
or s0 s0 s1
nor s7 s7 s8
```

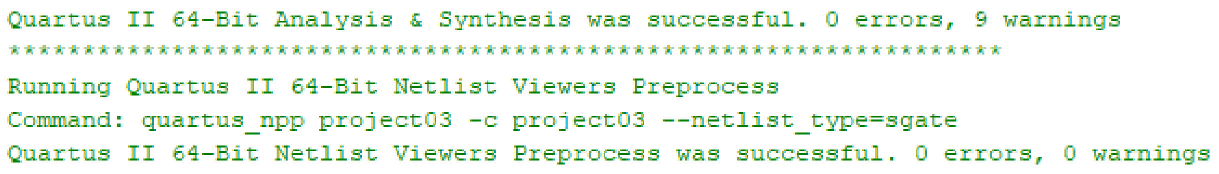


		INITIAL STATE		VALUES AFTER INSTRUCTIONS									
ADDRESS		BINARY	DECIMAL	add	addu	sub	subu	sll	srl	sltu	and	or	nor
s0	$16	1100000	96	96	96	96	96	96	24	24	24	56	56
s1	$17	10000	16	16	16	16	16	32	32	32	32	32	32
s2	$18	10001	17	112	113	113	113	113	113	113	113	113	113
s3	$19	100001	33	33	33	97	96	96	96	96	96	96	96
s4	$20	1000010	66	66	66	66	66	66	66	66	16	16	16
s5	$21	10000	16	16	16	16	16	16	16	16	16	16	16
s6	$22	1	1	1	1	1	1	1	1	1	1	1	1
RESULT				112	113	97	96	32	24	1	16	56	7

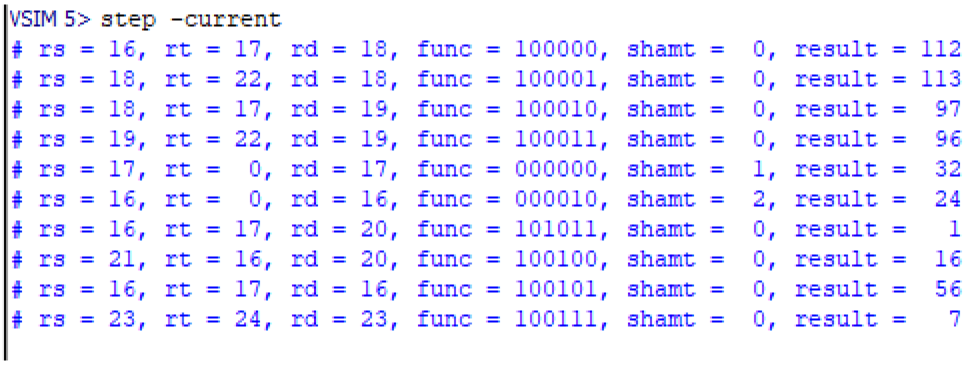
SCREENSHOTS



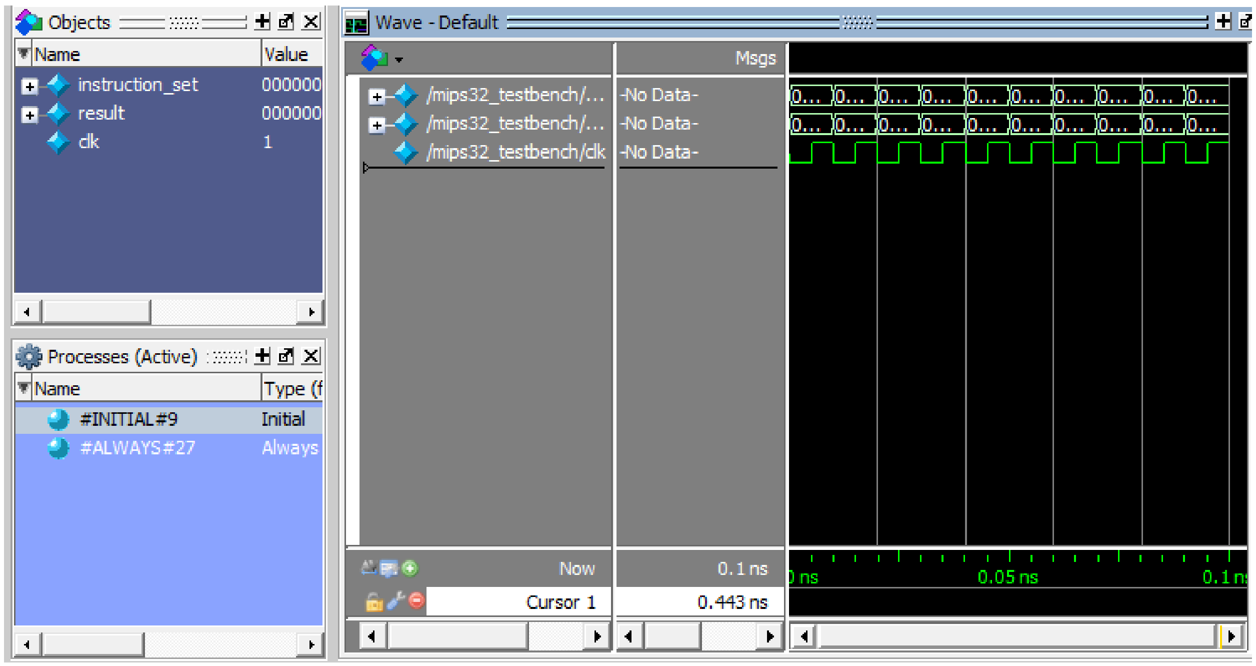
Files
mips32_testbench.v
mips32.v
simulation/modelsim/reg
mips_registers.v
control_unit.v
adder_32bit.v
alu32.v
and_32bit.v
full_adder.v
half_adder.v
left_shift.v
mux_2x1.v
mux_2x1_32bit.v
mux_8x1.v
nand_32bit.v
nor_32bit.v
or_32bit.v
right_for_left.v
right_shift.v
subtractor_32bit.v
xor_32bit.v
controller.v



```
Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 9 warnings
******************************************************************
Running Quartus II 64-Bit Netlist Viewers Preprocess
Command: quartus_npp project03 -c project03 --netlist_type=sgate
Quartus II 64-Bit Netlist Viewers Preprocess was successful. 0 errors, 0 warnings
```



```
VSIM 5> step -current
# rs = 16, rt = 17, rd = 18, func = 100000, shamt =  0, result = 112
# rs = 18, rt = 22, rd = 18, func = 100001, shamt =  0, result = 113
# rs = 18, rt = 17, rd = 19, func = 100010, shamt =  0, result =  97
# rs = 19, rt = 22, rd = 19, func = 100011, shamt =  0, result =  96
# rs = 17, rt =  0, rd = 17, func = 000000, shamt =  1, result =  32
# rs = 16, rt =  0, rd = 16, func = 000010, shamt =  2, result =  24
# rs = 16, rt = 17, rd = 20, func = 101011, shamt =  0, result =   1
# rs = 21, rt = 16, rd = 20, func = 100100, shamt =  0, result =  16
# rs = 16, rt = 17, rd = 16, func = 100101, shamt =  0, result =  56
# rs = 23, rt = 24, rd = 23, func = 100111, shamt =  0, result =   7
```



Objects

Name	Value
instruction_set	000000
result	000000
clk	1

Processes (Active)

Name	Type (f
#INITIAL#9	Initial
#ALWAYS#27	Always

Wave - Default

	Msgs
/mips32_testbench/...	-No Data-
/mips32_testbench/...	-No Data-
/mips32_testbench/clk	-No Data-

Now	0.1 ns
Cursor 1	0.443 ns