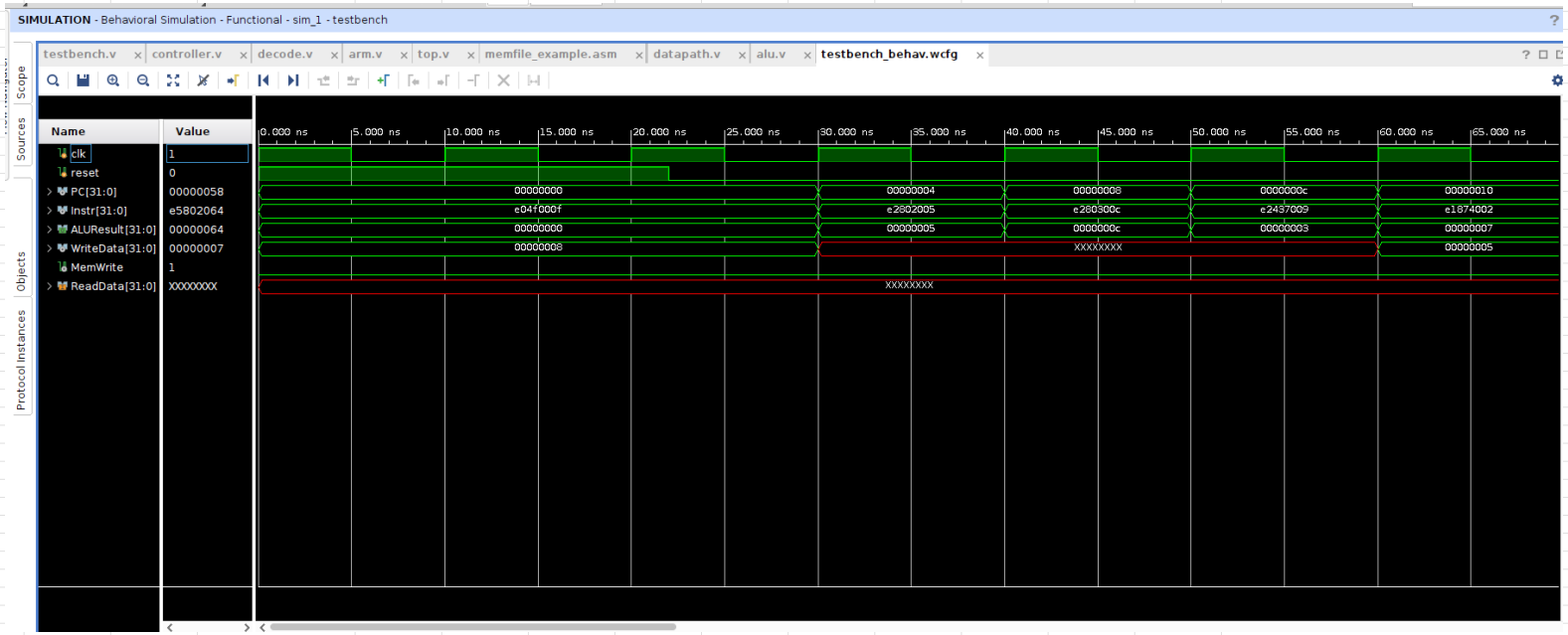
cycle	reset	pc	instr	srcA	srcB	branch	aluResult	Flags(NZCV)	condEx	writeData	memWrite	readData
1	1	0	SUB R0, R15, R15	8	8	0	0	?	1	8	0	x
2	0	4	ADD R2, R0, #5	0	5	0	5	?	1	x	0	x
3	0	8	ADD R3, R0, #12	0	c	0	c	?	1	x	0	x
4	0	c	SUB R7, R3, #9	c	9	0	3	?	1	x	0	x
5	0	10	ORR R4, R7, R2	3	5	0	7	?	1	5	0	x
6	0	14	AND R5, R3, R4	c	7	0	4	?	1	7	0	x
7	0	18	ADD R5, R5, R4	4	7	0	b	?	1	7	0	x
8	0	1c	SUBS R8, R5, R7	b	3	0	8	0010	1	3	0	x
9	0	20	BEQ END	28	30	1	58	0010	0	x	0	x
10	0	24	SUBS R8, R3, R4	c	7	0	5	0010	1	7	0	x
11	0	28	BGE AROUND	30	0	1	30	0010	1	0	0	x
12	0	30	AROUND SUBS R8, R7, R2	3	5	0	FFFFFFFE	1000	1	5	0	x
13	0	34	ADDLT R7, R5, #1	b	1	0	c	1000	1	x	0	x
14	0	38	SUB R7, R7, R2	c	5	0	7	1000	1	5	0	x
15	0	3c	STR R7, [R3, #84]	c	54	0	60	1000	1	7	1	x
16	0	40	LDR R2, [R0, #96]	0	60	0	60	1000	1	0	0	1
17	0	44	ADD R15, R15, R0	4c	0	0	4c	1000	1	0	0	x
18	0	4c	B END	54	4	1	58	1000	1	x	0	x
19	0	58	END STR R2, [R0, #100]	0	64	0	64	1000	1	7	1	x

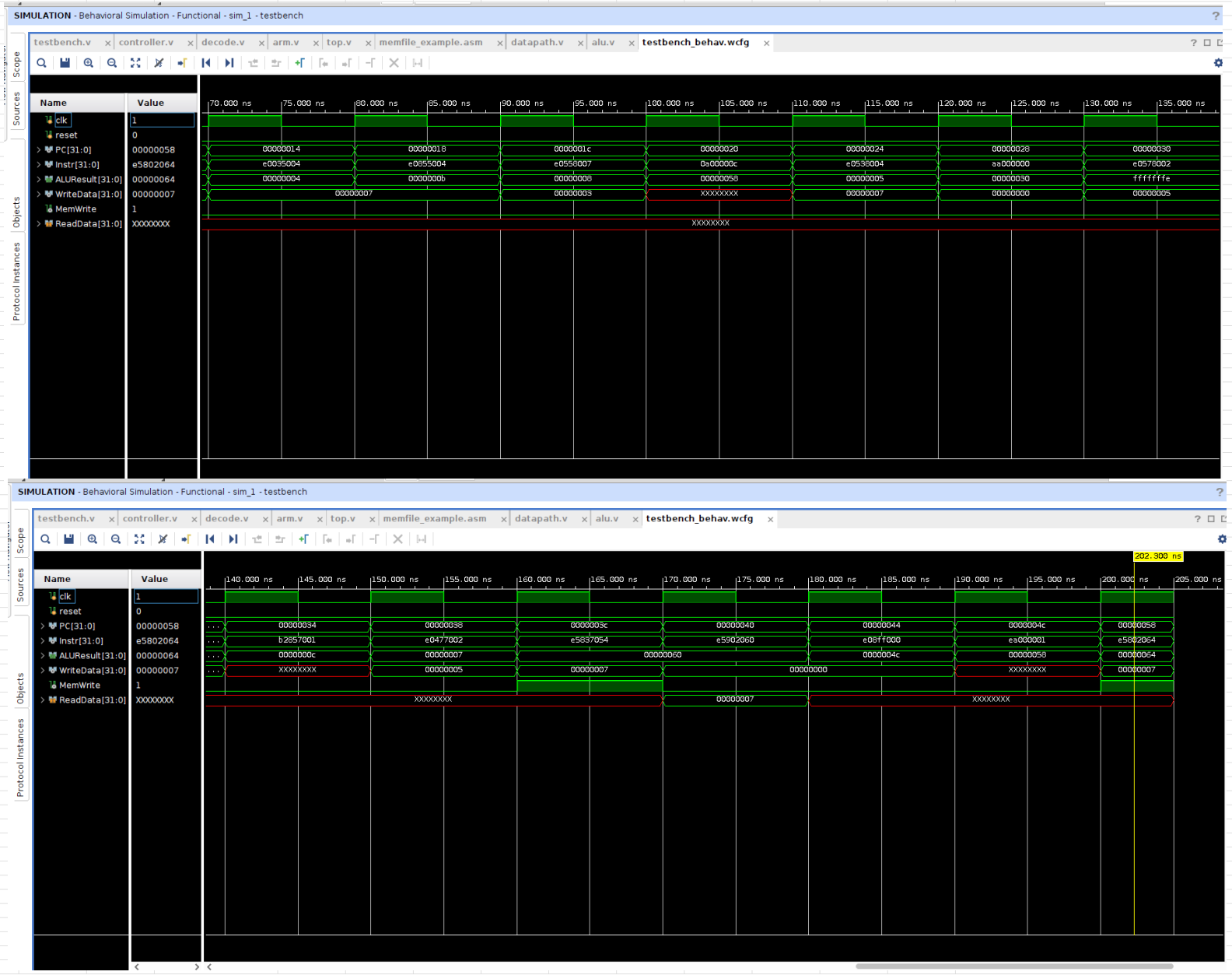
Nombres
Joaquin Mauricio Salinas Salas
Jose Guillermo Galvez Pacori
Lando Fabrizio Veramendi Hilario

What address will the final STR instruction write to and what value will it write?
address: 0x64 and value: 7

An image of the simulation waveforms showing correct operation of the processor. Does it write the correct value to address 100?
yes, in the last image we can see that we come to the adrres 100 with the right value



sí


SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
testbench.v
controller.v
decode.v
arm.v
top.v
memfile_example.asm
datapath.v
alu.v
testbench_behav.wcfg
Scope
Sources
Objects
Protocol Instances
Name
Value
clk
reset
PC[31:0]
Instr[31:0]
ALUResult[31:0]
WriteData[31:0]
MemWrite
ReadData[31:0]
70.000 ns
75.000 ns
80.000 ns
85.000 ns
90.000 ns
95.000 ns
100.000 ns
105.000 ns
110.000 ns
115.000 ns
120.000 ns
125.000 ns
130.000 ns
135.000 ns
00000058
e5802064
00000064
00000007
1
XXXXXXXX
00000014
00000018
0000001c
00000020
00000024
00000028
00000030
e0035004
e0855004
e0558007
0a00000c
e0536004
aa000000
e0578002
00000004
0000000b
00000008
00000058
00000005
00000030
fffffffe
00000007
00000003
XXXXXXXX
00000007
00000000
00000005
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
202.300 ns
140.000 ns
145.000 ns
150.000 ns
155.000 ns
160.000 ns
165.000 ns
170.000 ns
175.000 ns
180.000 ns
185.000 ns
190.000 ns
195.000 ns
200.000 ns
205.000 ns
00000034
00000038
0000003c
00000040
00000044
0000004c
00000058
b2857001
e0477002
e5837054
e5902060
e08ff000
ea000001
e5802064
0000000c
00000007
00000060
0000004c
00000058
00000064
XXXXXXXX
00000005
00000007
00000000
XXXXXXXX
00000007
XXXXXXXX
00000007
XXXXXXXX