

CSE331
HOMEWORK
#3 REPORT

BİLAL YALÇINKAYA

Half_adder: XOR inputs for sum and AND for carry out.

Full_adder: Calls first half adder then uses its carry out for next adder.

ADD32: Adds 1 bit numbers 32 times according to previous adders cout.

AND32: AND 1 bit numbers 32 times, assign it to result.

OR32: OR 1 bit numbers 32 times, assign it to result.

XOR32: XOR 1 bit numbers 32 times, assign it to result.

Complement2s: Adds 1 to not(!) of input.

SUB32: Finds 2s complement of second number and adds it to first number since these are signed numbers.

SLT32: First uses SUB32 to a-b for result then Checks for overflow situation with couts and xor it with results last bit, according to that last bit of result will be 0 or 1.

NOT32: NOT 1 bit numbers 32 times, assign it to result;

NOR32: NOR 1 bit numbers 32 times, assign it to result.

MUX2: Choices between 2 input according to choice input. Mux32 and mux64 uses this module 32 or 64 times also.

MUX3: (3bit)8x1 mux choses proper result.

ALU32: First finds all possible results, assigns them to 8bit 32 wires to use it after with mux then calls mux3 32 times to assign proper result according to choice input.

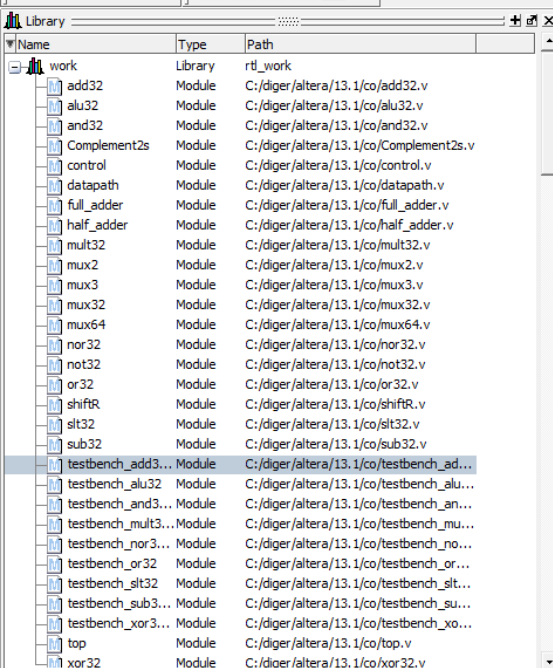
ShiftR: Shifts input to right 1 bit.

Control: According to reset,clock,p0,done and currentState it will generates nextState according to state table.

Datapath: First if write is 0 then adds product 0, else adds multiplicand to product. Shifts product and if shift is 1 shifts 1 bit, else shifts 0 bit which will decided by mux then adds counter 1 if shift input is 1.

Mult32: After initializing variables it goes to control module to get what we gonna do and with them it goes to datapath to get next step of multiplication. Adds multiplicand step by step to result which we learned at course. If counter reaches 31, input done will be 1 and product will be assigned to result so alu32 can use it.

My mult32 can runs only 1 time even after i change the inputs it will give previous result. So i need to compile again for another input. I couldn't solve that problem.



```


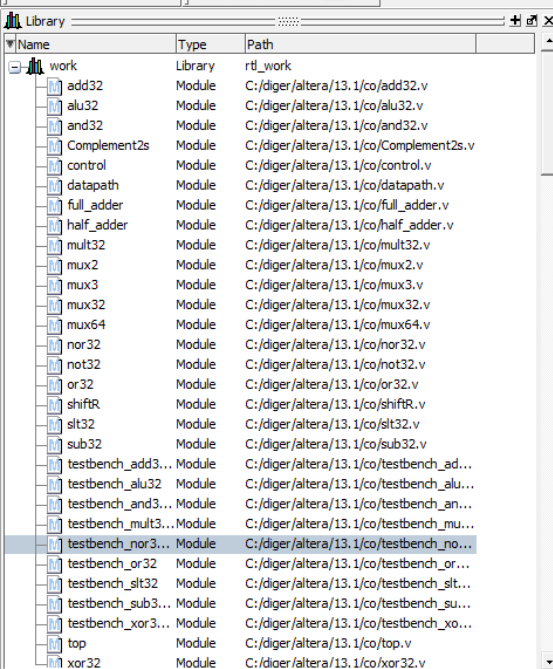
# Transcript
# Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov 2 2012
# -- Compiling module testbench_xor32
#
# Top level modules:
#     testbench_xor32
ModelSim> vsim work.testbench_add32
# vsim work.testbench_add32
# Loading work.testbench_add32
# Loading work.add32
# Loading work.full_adder
# Loading work.half_adder
VSIM 12> run
# time = 0, First Number = 00000000000000000000000000000000 | 1111101 | 125, Second Number = 00000000000000000000000000000000 | 11111 | 63, RESULT = 00000000000000000000000000000000 | 10111100 | 188
# time = 20, First Number = 11111111111111111111111110000011 | -125, Second Number = 00000000000000000000000000000000 | 11111 | 63, RESULT = 111111111111111111111111111000010 | -62
# time = 40, First Number = 11111111111111111111111110000011 | -125, Second Number = 1111111111111111111111111000001 | -63, RESULT = 111111111111111111111111101000100 | -188
# time = 60, First Number = 00000000000000000000000000000000 | 1111101 | 125, Second Number = 111111111111111111111111111000001 | -63, RESULT = 00000000000000000000000000000000 | 111110 | 62
VSIM 13>
Now: 100 ps Delta: 0 work

```


Name	Value	Kind	Mode
FN	0000000000000000000000000000000011111101	Pack...	Internal
SN	1111111111111111111111111111111110000001	Pack...	Internal
result	0000000000000000000000000000000010000001	Net	Internal

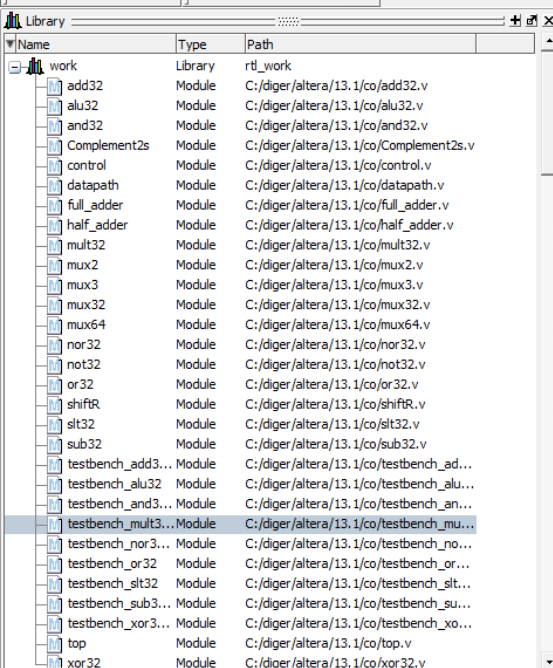
The screenshot shows the 'Wave - Default' window. The top panel is a large gray area. The bottom panel is a black area with a yellow vertical line. The bottom status bar shows 'Now', 'Cursor 1', and '0 ps'.

Now: 100 ps Delta: 0	Module
----------------------	--------



```
Transcript
# Loading work.and32
VSIM 16> run
# time = 0, First Number = 000000000000000000000000000000000000 |      125, Second Number = 000000000000000000000000000000000000 |      63, RESULT = 000000000000000000000000000000000000 |      61
# time = 20, First Number = 11111111111111111111111110000001 |    -125, Second Number = 000000000000000000000000000000000000 |      63, RESULT = 000000000000000000000000000000000001 |       3
# time = 40, First Number = 11111111111111111111111110000001 |    -125, Second Number = 11111111111111111111111110000001 |   -127, RESULT = 1111111111111111111111111111111110000001 |   -127
# time = 60, First Number = 000000000000000000000000000000000000 |      125, Second Number = 11111111111111111111111110000001 |   -63, RESULT = 00000000000000000000000000000000010000001 |     65
VSIM 17> vsim work.testbench_nor32
# vsim work.testbench_nor32
# Loading work.testbench_nor32
# Loading work.nor32
# Loading work.not32
VSIM 18> run
# time = 0, First Number = 000000000000000000000000000000000000 |      125, Second Number = 000000000000000000000000000000000000 |      63, RESULT = 1111111111111111111111111111111110000000 |   -128
# time = 20, First Number = 11111111111111111111111110000001 |    -125, Second Number = 000000000000000000000000000000000000 |      63, RESULT = 0000000000000000000000000000000000010000000 |     64
# time = 40, First Number = 11111111111111111111111110000001 |    -125, Second Number = 11111111111111111111111110000001 |   -63, RESULT = 0000000000000000000000000000000000000111100 |     60
# time = 60, First Number = 000000000000000000000000000000000000 |      125, Second Number = 111111111111111111111111111110000001 |   -63, RESULT = 000000000000000000000000000000000000000010 |      2
VSIM 19>
```

Now: 100 ps Delta: 0 sim:/testbench_nor32 0 ps to 1 ns

[illegible]

The screenshot shows the 'sim - Default' window. The 'Instance' pane on the left lists the components of the testbench. The 'testbench_mult32' instance is selected, showing its sub-components: 'c' and '#vsim_capacity#'. The 'testbench_mult32' instance is highlighted in blue.

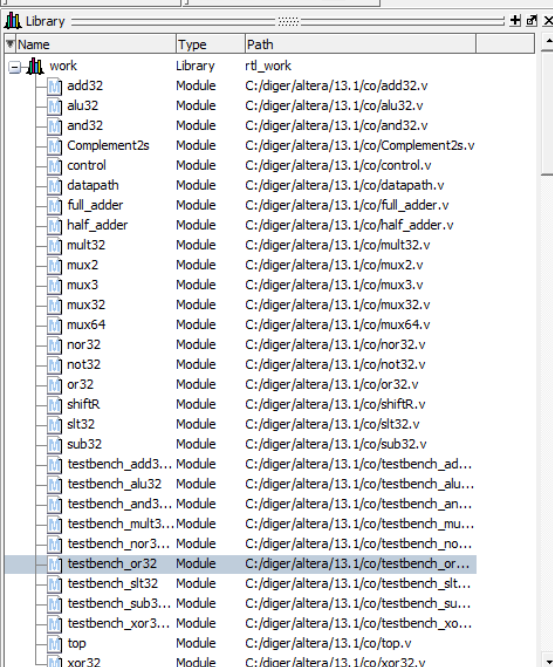
Wave - Default

Msgs

Now 1100 ps
Cursor 1 0 ps

500 ps 1000 ps

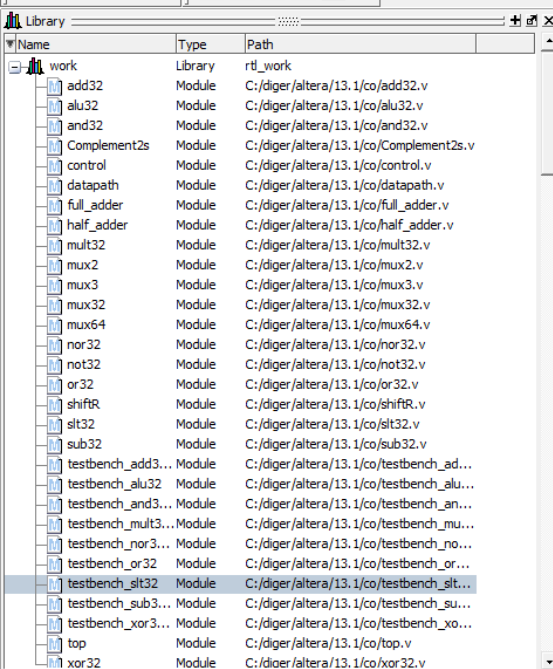
[illegible]



```
run
run
run
run
run
run
# time = 1000, First Number = 00000000000000000000000000000000 |      125, Second Number = 00000000000000000000000000000000 |      63, RESULT = 00000000000000000000000000000000 |      7875
VSIM 21> vsim work.testbench_or32
VSIM 21> work.testbench_or32
# Loading work.testbench_or32
# Loading work.or32
VSIM 22> run
# time = 0, First Number = 00000000000000000000000000000000 |      125, Second Number = 00000000000000000000000000000000 |      63, RESULT = 00000000000000000000000000000000 |      127
# time = 20, First Number = 11111111111111111111111111111111 | -125, Second Number = 00000000000000000000000000000000 |      63, RESULT = 11111111111111111111111111111111 |     -65
# time = 40, First Number = 11111111111111111111111111111111 | -125, Second Number = 11111111111111111111111111111111 |    -63, RESULT = 11111111111111111111111111111111 |     -61
# time = 60, First Number = 00000000000000000000000000000000 |      125, Second Number = 11111111111111111111111111111111 |    -63, RESULT = 11111111111111111111111111111111 |     -3
VSIM 23>
```

Now: 100 ns. Delta: 0

sim:/testbench_or32	0 ps to 1 ns
---------------------	--------------

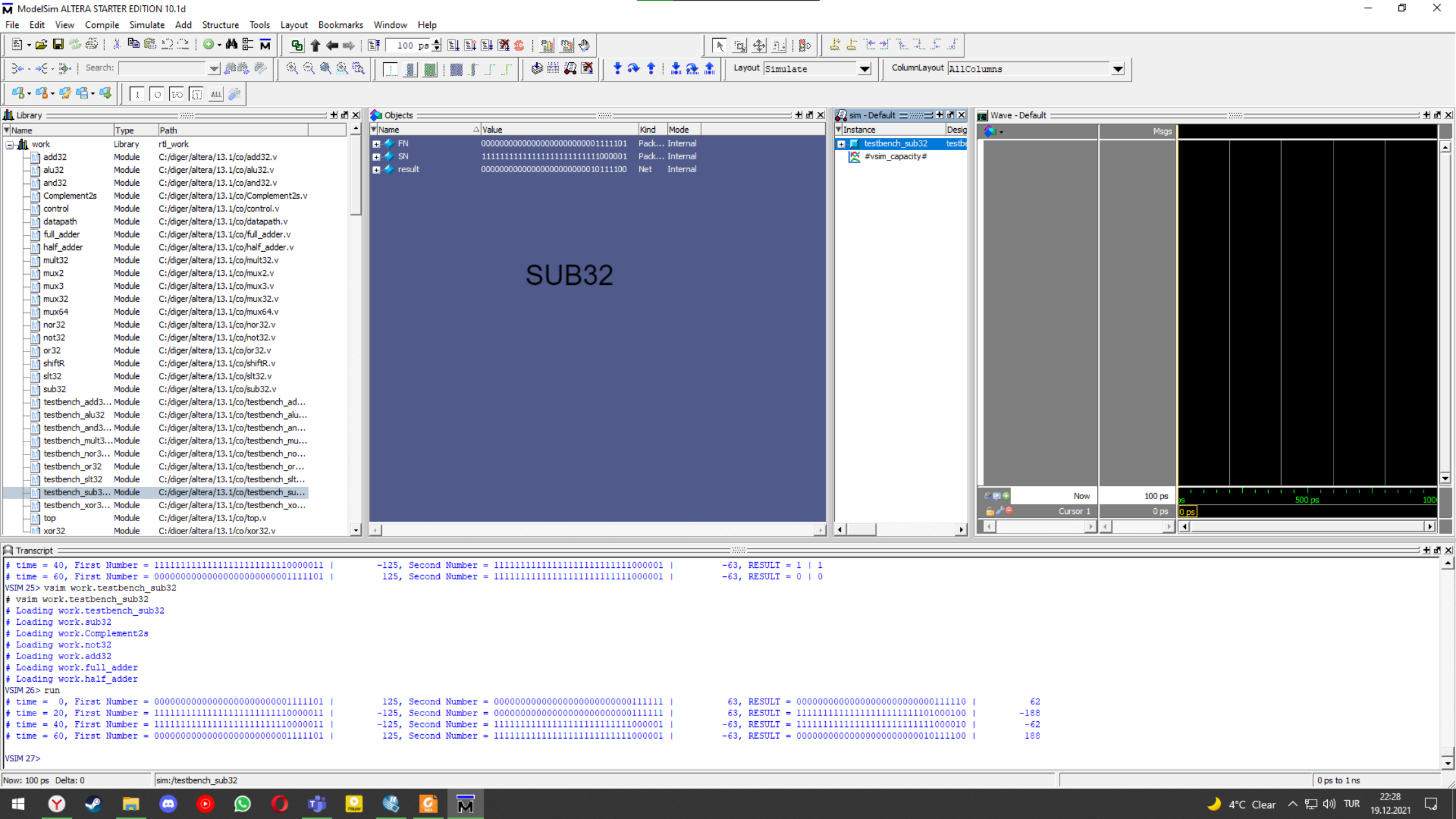



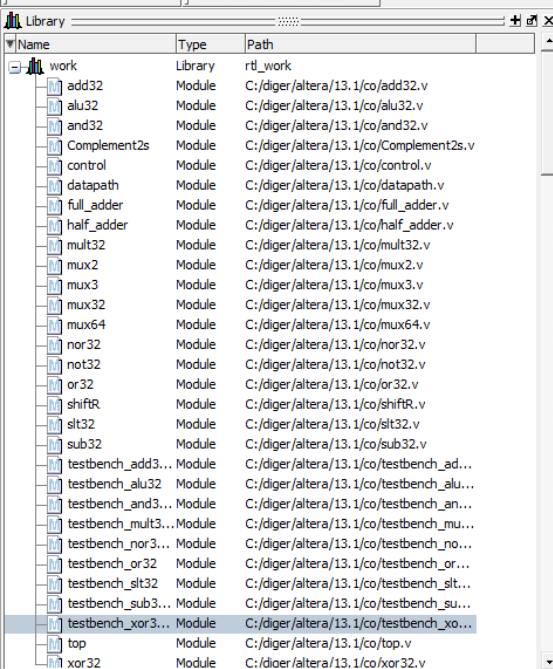
The screenshot shows the 'Wave - Default' window. The top bar contains standard window controls and a toolbar. The main area is divided into three vertical panels. The leftmost panel is a large gray rectangle. The middle panel is labeled 'Msgs' and is also gray. The rightmost panel is a large black rectangle, likely for displaying waveforms. At the bottom, there is a time scale with markers at 'Now', '100 ps', '500 ps', and '1000 ps'. A cursor is positioned at '0 ps'. The bottom status bar shows navigation controls and a time value of '0 ps'.

```

Transcript
# time = 40, First Number = 111111111111111111110000011 | -125, Second Number = 11111111111111111111000001 | -63, RESULT = 11111111111111111111000011 | -61
# time = 60, First Number = 000000000000000000000000111101 | 125, Second Number = 11111111111111111111000001 | -63, RESULT = 111111111111111111111101 | -3
VSI3M 23> vsim work.testbench_sl32
# vsim work.testbench_sl32
# Loading work.testbench_sl32
# Loading work.sl32
# Loading work.Complement2s
# Loading work.not32
# Loading work.add32
# Loading work.full_adder
# Loading work.half_adder
VSI3M 24> run
# time = 0, First Number = 000000000000000000000000111101 | 125, Second Number = 0000000000000000000000001111 | 63, RESULT = 0 | 0
# time = 20, First Number = 111111111111111111110000011 | -125, Second Number = 0000000000000000000000001111 | 63, RESULT = 1 | 1
# time = 40, First Number = 111111111111111111110000011 | -125, Second Number = 11111111111111111111000001 | -63, RESULT = 1 | 1
# time = 60, First Number = 000000000000000000000000111101 | 125, Second Number = 11111111111111111111000001 | -63, RESULT = 0 | 0
VSI3M 25>
Now: 100 ns. Delta: 0
sim:/testbench_sl32

```





```
sim - Default
Instance
testbench_xor32
xor32_test
#vsim_capacity#
```

The screenshot shows the 'Wave - Default' window. The top bar includes standard window controls and a toolbar. The main area is divided into three vertical panels. The leftmost panel is a large gray rectangle. The middle panel is labeled 'Msgs' and is also gray. The rightmost panel is a large black rectangle. At the bottom, there is a time scale with green tick marks. Labels on the scale include 'Now', '100 ps', '500 ps', and '1000 ps'. A yellow cursor is positioned at '0 ps'. The bottom status bar contains navigation arrows and a time value of '0 ps'.

[illegible]