CSE 331/503

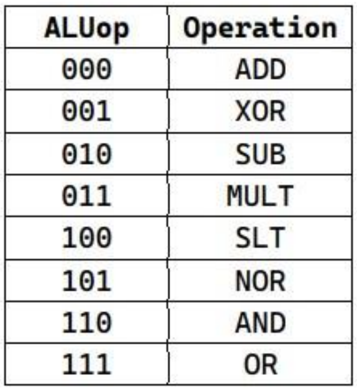
Computer Organization

Homework 3 – ALU with Multiplication Design

1901042705

Zeynep Çiğdem Parlatan

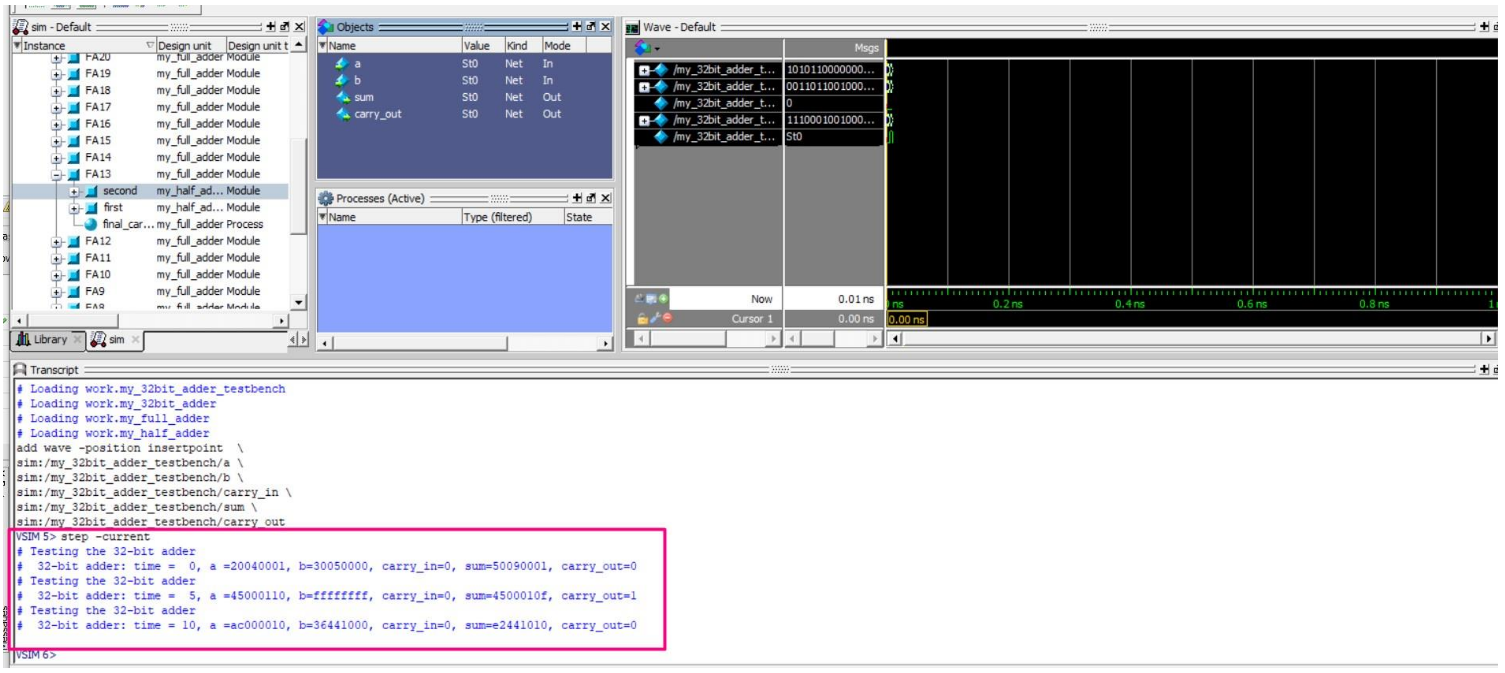
=> alu32.v Module part



ALUop	Operation
000	ADD
001	XOR
010	SUB
011	MULT
100	SLT
101	NOR
110	AND
111	OR

-> I made a separate module for each operation.

✓ **Operation of the 32-bit adder (module my_32bit_adder)**



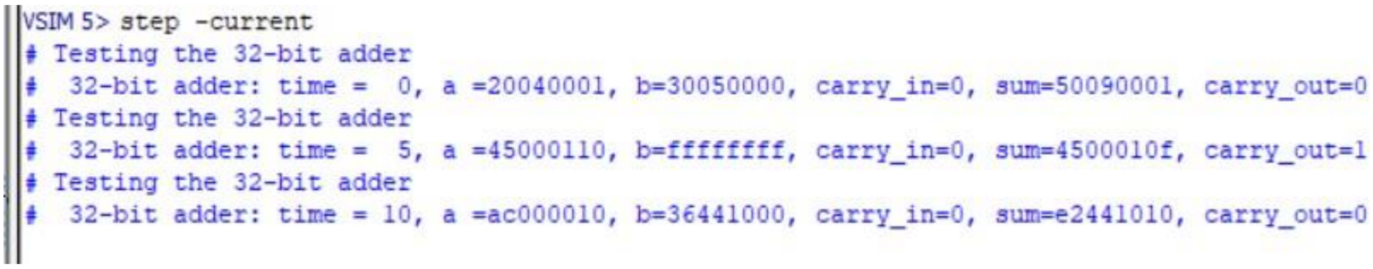
```
sim - Default
Instance        Design unit     Design unit t
  FA20          my_full_adder   Module
  FA19          my_full_adder   Module
  FA18          my_full_adder   Module
  FA17          my_full_adder   Module
  FA16          my_full_adder   Module
  FA15          my_full_adder   Module
  FA14          my_full_adder   Module
  FA13          my_full_adder   Module
    second      my_half_ad...   Module
    first       my_half_ad...   Module
    final_car...my_full_adder   Process
  FA12          my_full_adder   Module
  FA11          my_full_adder   Module
  FA10          my_full_adder   Module
  FA9           my_full_adder   Module
  FA8           my_full_adder   Module
```

```
Objects
Name        Value  Kind  Mode
a           St0    Net   In
b           St0    Net   In
sum         St0    Net   Out
carry_out   St0    Net   Out
```

```
Processes (Active)
Name        Type (filtered)   State
```

```
Wave - Default
                         Msgs
/my_32bit_adder_t...     1010110000000...
/my_32bit_adder_t...     0011011001000...
/my_32bit_adder_t...     0
/my_32bit_adder_t...     1110001001000...
/my_32bit_adder_t...     St0
Now                      0.01 ns
Cursor 1                 0.00 ns
```

```
Transcript
# Loading work.my_32bit_adder_testbench
# Loading work.my_32bit_adder
# Loading work.my_full_adder
# Loading work.my_half_adder
add wave -position insertpoint  \
sim:/my_32bit_adder_testbench/a \
sim:/my_32bit_adder_testbench/b \
sim:/my_32bit_adder_testbench/carry_in \
sim:/my_32bit_adder_testbench/sum \
sim:/my_32bit_adder_testbench/carry_out
VSIM 5> step -current
# Testing the 32-bit adder
#  32-bit adder: time =  0, a =20040001, b=30050000, carry_in=0, sum=50090001, carry_out=0
# Testing the 32-bit adder
#  32-bit adder: time =  5, a =45000110, b=ffffffff, carry_in=0, sum=4500010f, carry_out=1
# Testing the 32-bit adder
#  32-bit adder: time = 10, a =ac000010, b=36441000, carry_in=0, sum=e2441010, carry_out=0

VSIM 6>
```



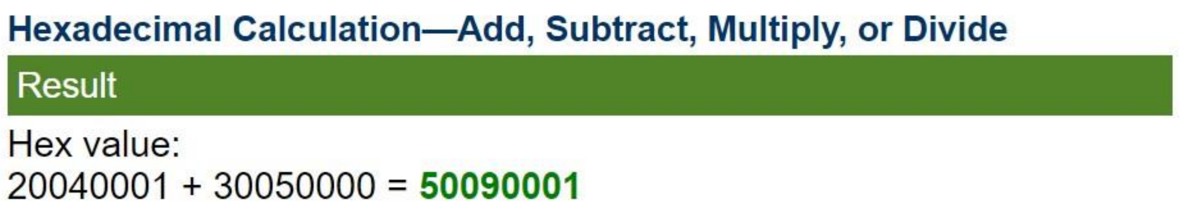
```
VSIM 5> step -current
# Testing the 32-bit adder
#  32-bit adder: time =  0, a =20040001, b=30050000, carry_in=0, sum=50090001, carry_out=0
# Testing the 32-bit adder
#  32-bit adder: time =  5, a =45000110, b=ffffffff, carry_in=0, sum=4500010f, carry_out=1
# Testing the 32-bit adder
#  32-bit adder: time = 10, a =ac000010, b=36441000, carry_in=0, sum=e2441010, carry_out=0
```

-> I used 32 full adders to create 32 bit adders

-> I have writed the 32 bits as hexadecimal for easier understanding.

-> I calculated the accuracy of the results in the online hexadecimal calculator.

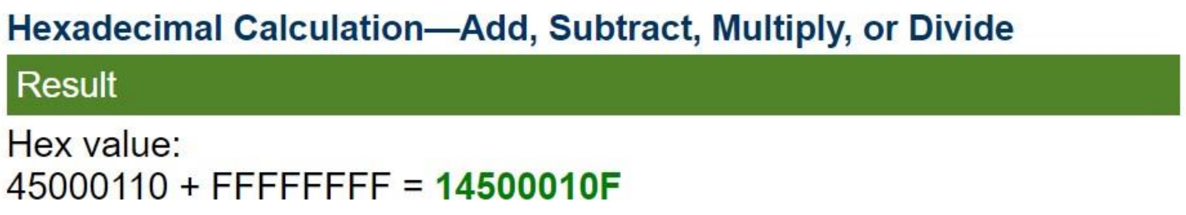
Test1:


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
20040001 + 30050000 = **50090001**

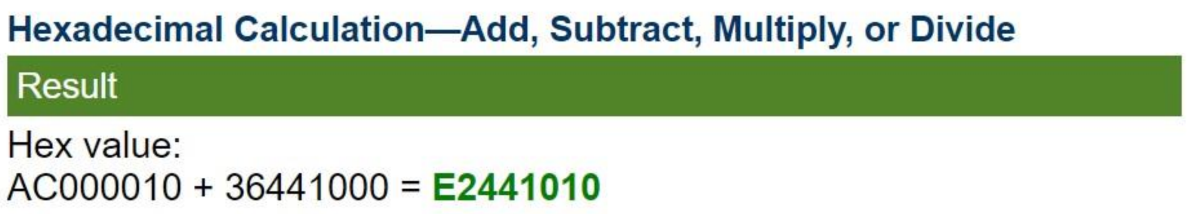
Test2:


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
45000110 + FFFFFFFF = **14500010F**

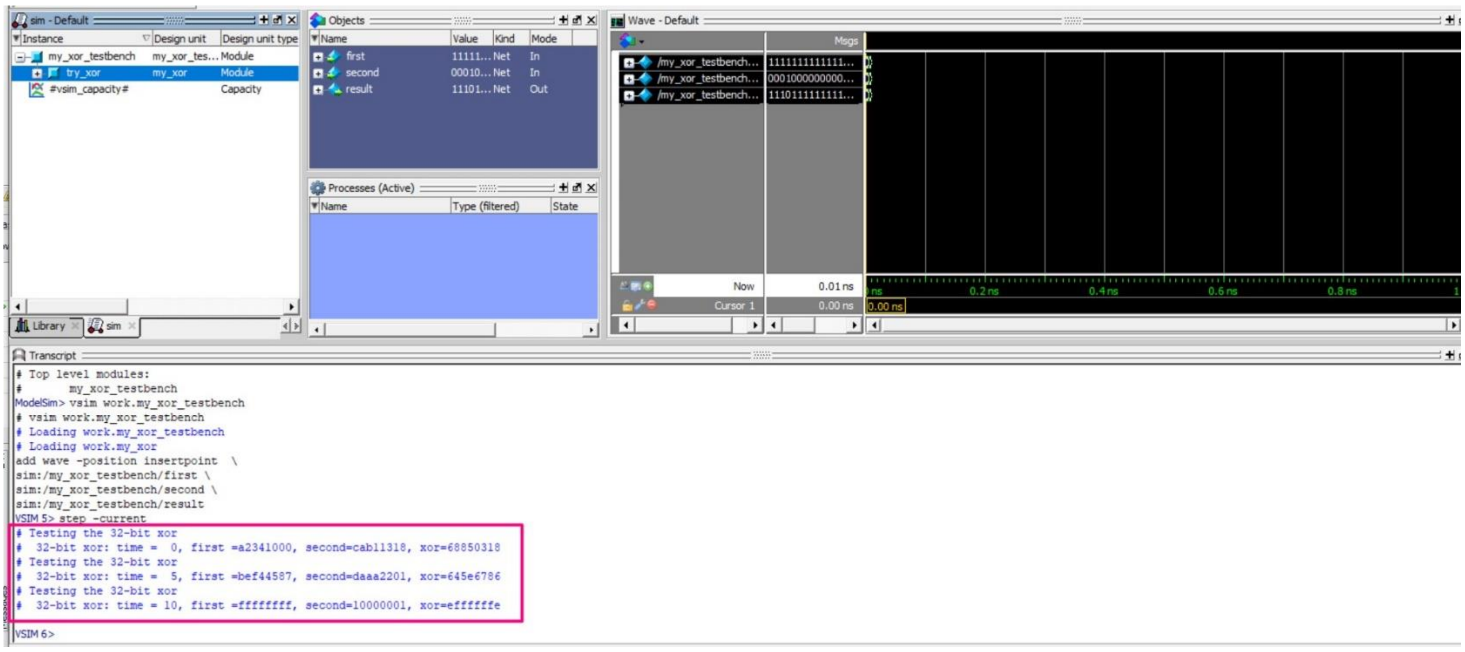
Test3:


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
AC000010 + 36441000 = **E2441010**

✓ **Operation of the 32-bit xor (module my_xor)**

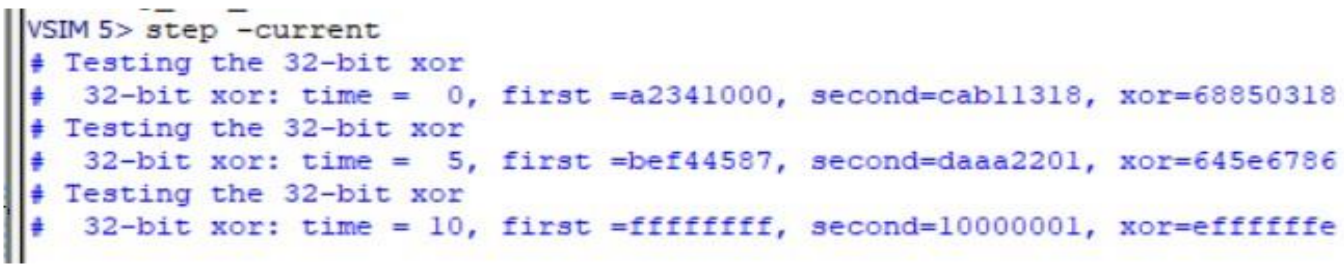


Instance	Design unit	Design unit type
my_xor_testbench	my_xor_tes...	Module
try_xor	my_xor	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
first	11111...	Net	In
second	00010...	Net	In
result	11101...	Net	Out

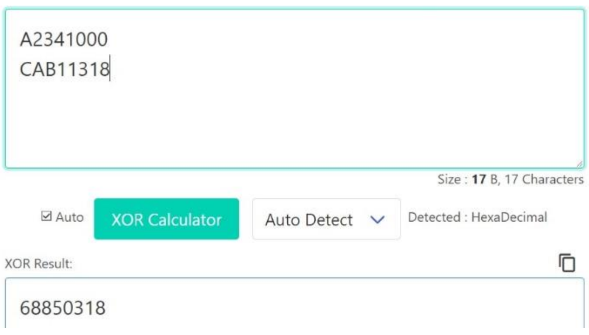
```
/my_xor_testbench...   1111111111111...
/my_xor_testbench...   0001000000000...
/my_xor_testbench...   1110111111111...
```

```
# Top level modules:
#       my_xor_testbench
ModelSim> vsim work.my_xor_testbench
# vsim work.my_xor_testbench
# Loading work.my_xor_testbench
# Loading work.my_xor
add wave -position insertpoint  \
sim:/my_xor_testbench/first \
sim:/my_xor_testbench/second \
sim:/my_xor_testbench/result
VSIM 5> step -current
# Testing the 32-bit xor
#  32-bit xor: time =  0, first =a2341000, second=cab11318, xor=68850318
# Testing the 32-bit xor
#  32-bit xor: time =  5, first =bef44587, second=daaa2201, xor=645e6786
# Testing the 32-bit xor
#  32-bit xor: time = 10, first =ffffffff, second=10000001, xor=effffffe
VSIM 6>
```



```
VSIM 5> step -current
# Testing the 32-bit xor
#  32-bit xor: time =  0, first =a2341000, second=cab11318, xor=68850318
# Testing the 32-bit xor
#  32-bit xor: time =  5, first =bef44587, second=daaa2201, xor=645e6786
# Testing the 32-bit xor
#  32-bit xor: time = 10, first =ffffffff, second=10000001, xor=effffffe
```

Test1:



```
A2341000
CAB11318
```

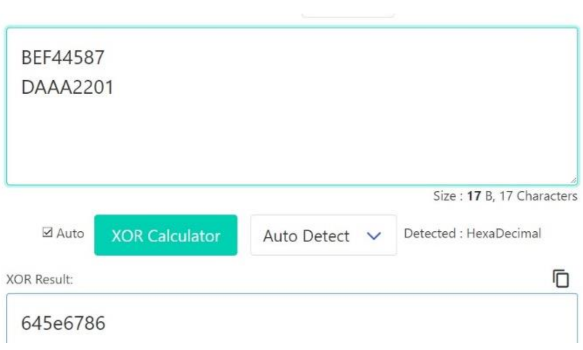
Size : **17** B, 17 Characters

☑ Auto | XOR Calculator | Auto Detect | Detected : HexaDecimal

XOR Result:

```
68850318
```

Test2:



```
BEF44587
DAAA2201
```

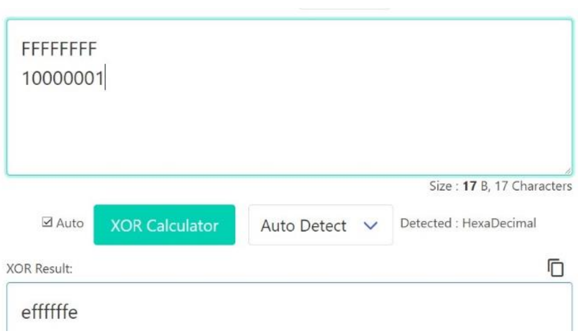
Size : **17** B, 17 Characters

☑ Auto | XOR Calculator | Auto Detect | Detected : HexaDecimal

XOR Result:

```
645e6786
```

Test3:



```
FFFFFFFF
10000001
```

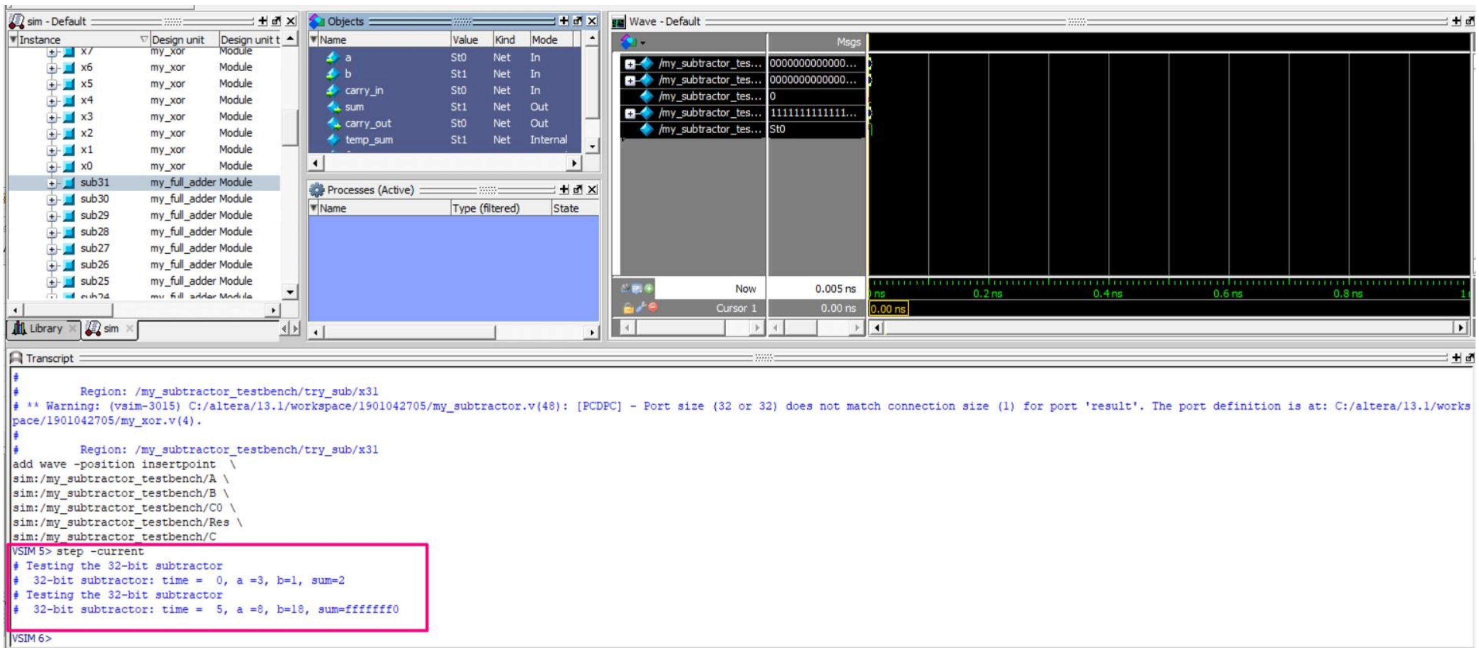
Size : **17** B, 17 Characters

☑ Auto | XOR Calculator | Auto Detect | Detected : HexaDecimal

XOR Result:

```
effffffe
```

✓ Operation of the 32-bit subtractor (module my_subtractor)



sim - Default

Instance	Design unit	Design unit t
x7	my_xor	Module
x6	my_xor	Module
x5	my_xor	Module
x4	my_xor	Module
x3	my_xor	Module
x2	my_xor	Module
x1	my_xor	Module
x0	my_xor	Module
sub31	my_full_adder	Module
sub30	my_full_adder	Module
sub29	my_full_adder	Module
sub28	my_full_adder	Module
sub27	my_full_adder	Module
sub26	my_full_adder	Module
sub25	my_full_adder	Module
sub24	my_full_adder	Module

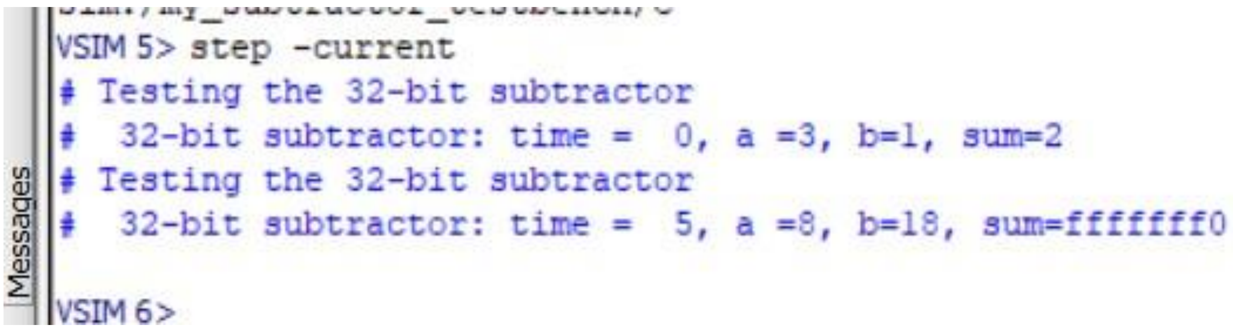
Objects

Name	Value	Kind	Mode
a	St0	Net	In
b	St1	Net	In
carry_in	St0	Net	In
sum	St1	Net	Out
carry_out	St0	Net	Out
temp_sum	St1	Net	Internal

Wave - Default

	Msgs
/my_subtractor_tes...	0000000000000...
/my_subtractor_tes...	0000000000000...
/my_subtractor_tes...	0
/my_subtractor_tes...	1111111111111...
/my_subtractor_tes...	St0
Now	0.005 ns
Cursor 1	0.00 ns

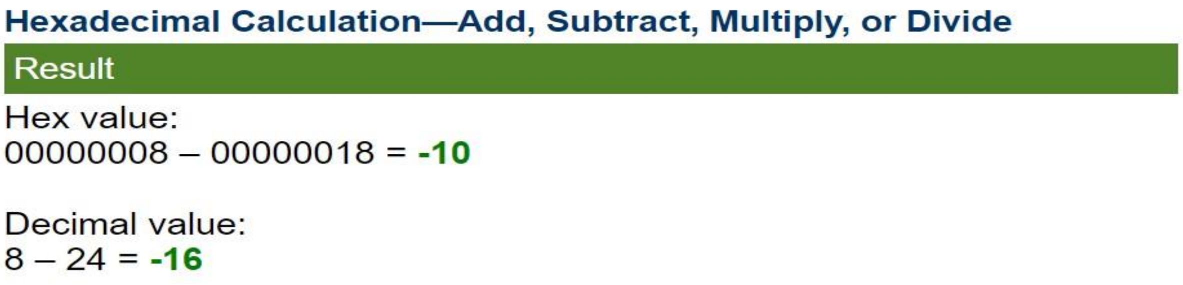
```
#
#          Region: /my_subtractor_testbench/try_sub/x31
# ** Warning: (vsim-3015) C:/altera/13.1/workspace/1901042705/my_subtractor.v(48): [PCDPC] - Port size (32 or 32) does not match connection size (1) for port 'result'. The port definition is at: C:/altera/13.1/workspace/1901042705/my_xor.v(4).
#
#          Region: /my_subtractor_testbench/try_sub/x31
add wave -position insertpoint  \
sim:/my_subtractor_testbench/A \
sim:/my_subtractor_testbench/B \
sim:/my_subtractor_testbench/C0 \
sim:/my_subtractor_testbench/Res \
sim:/my_subtractor_testbench/C
VSIM 5> step -current
# Testing the 32-bit subtractor
#   32-bit subtractor: time =  0, a =3, b=1, sum=2
# Testing the 32-bit subtractor
#   32-bit subtractor: time =  5, a =8, b=18, sum=fffffff0

VSIM 6>
```



```
VSIM 5> step -current
# Testing the 32-bit subtractor
#   32-bit subtractor: time =  0, a =3, b=1, sum=2
# Testing the 32-bit subtractor
#   32-bit subtractor: time =  5, a =8, b=18, sum=fffffff0

VSIM 6>
```

-> It gives two's complement for negative results in subtraction. I tried, converting it gives the correct answer.

Test:


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

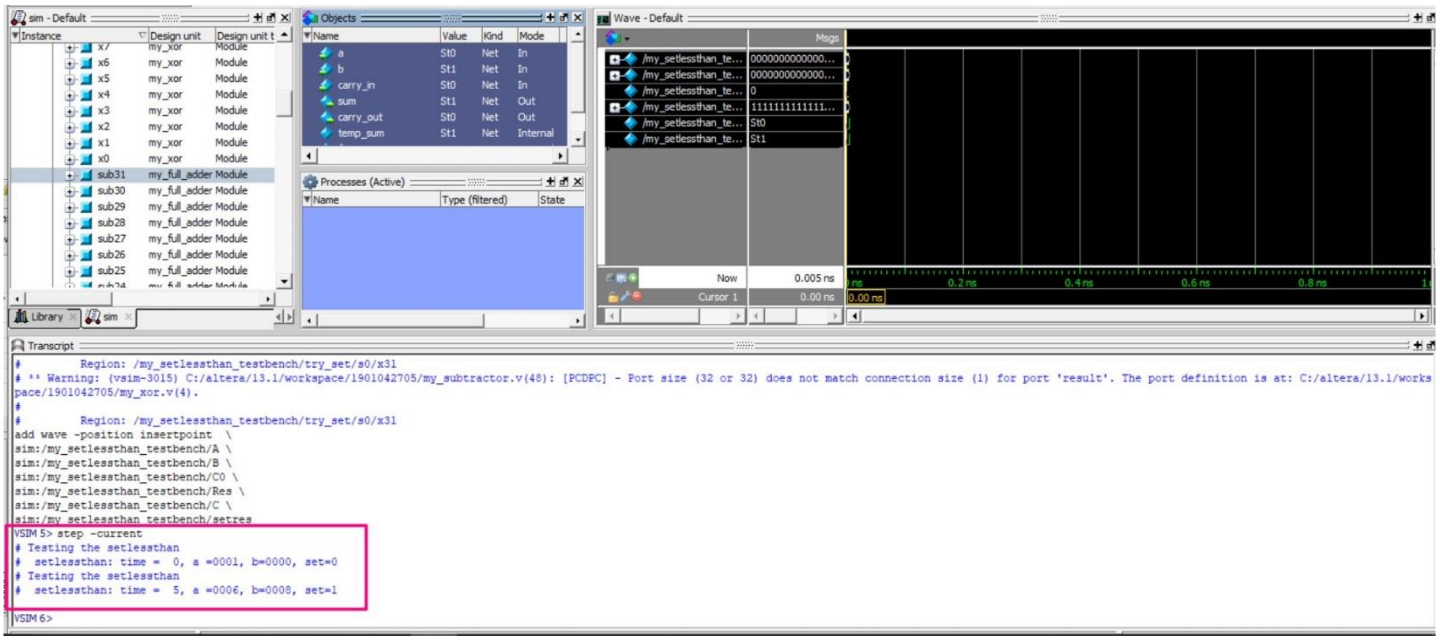
Result

Hex value:
00000008 – 00000018 = **-10**

Decimal value:
8 – 24 = **-16**

✓ **Operation of the sett less than (module my_setlessthan)**

-> I used the subtraction module while doing the Settlessthan operation. I subtracted the second number from the first number. Then I looked at the most significant bit. If the most significant bit is one, the result is negative. I adjusted the Setlessthan operation accordingly.

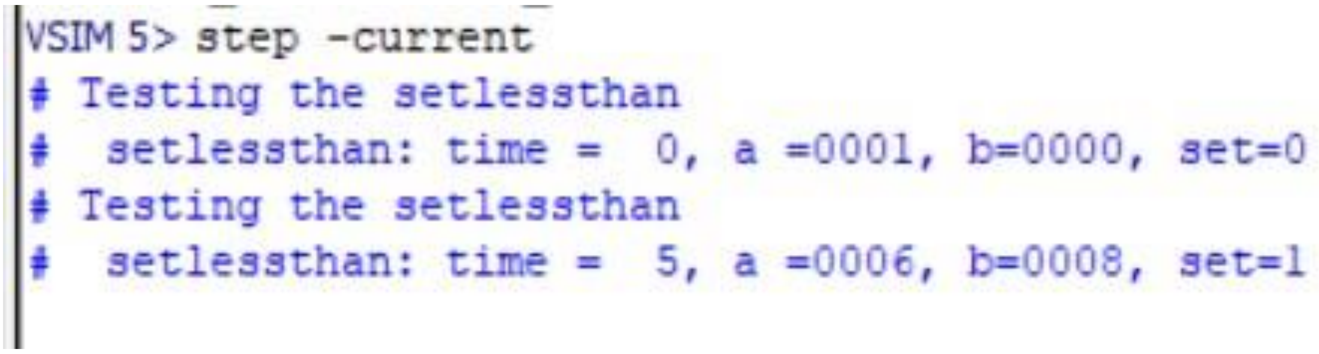


Instance	Design unit	Design unit t
x7	my_xor	Module
x6	my_xor	Module
x5	my_xor	Module
x4	my_xor	Module
x3	my_xor	Module
x2	my_xor	Module
x1	my_xor	Module
x0	my_xor	Module
sub31	my_full_adder	Module
sub30	my_full_adder	Module
sub29	my_full_adder	Module
sub28	my_full_adder	Module
sub27	my_full_adder	Module
sub26	my_full_adder	Module
sub25	my_full_adder	Module
sub24	my_full_adder	Module

Name	Value	Kind	Mode
a	St0	Net	In
b	St1	Net	In
carry_in	St0	Net	In
sum	St1	Net	Out
carry_out	St0	Net	Out
temp_sum	St1	Net	Internal

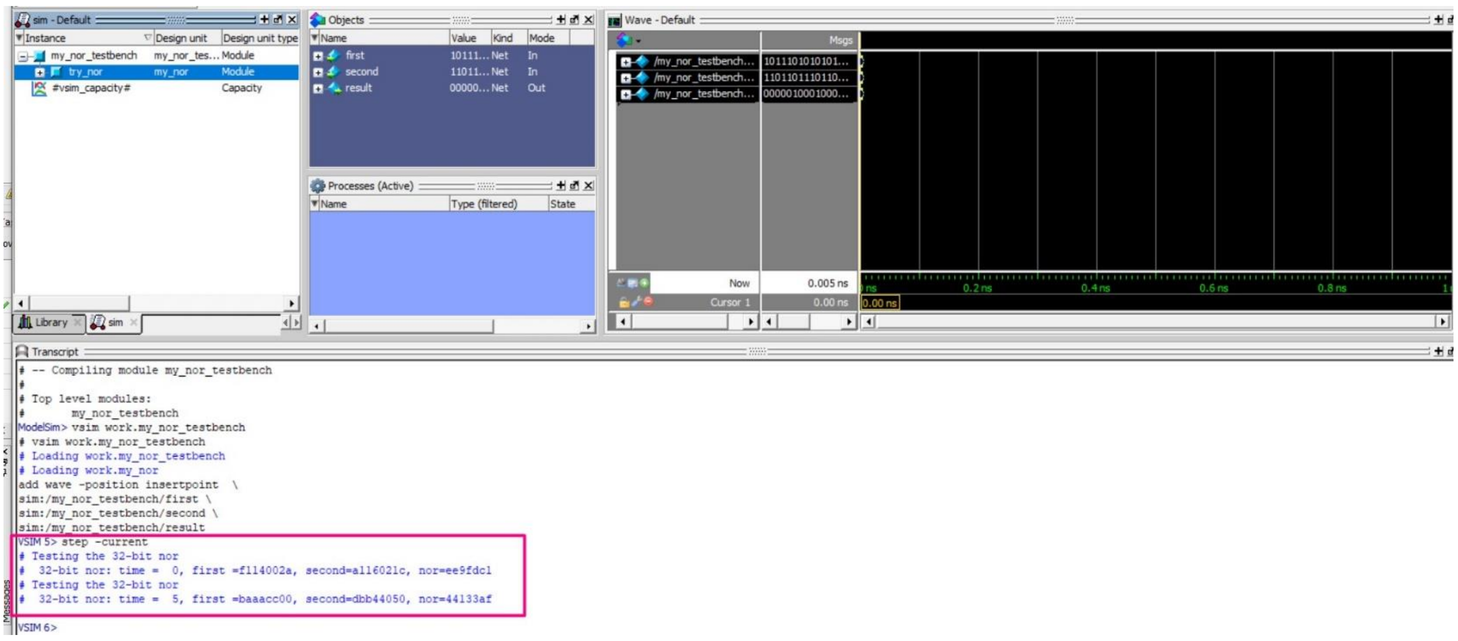
	Msgs
/my_setlessthan_te...	0000000000000...
/my_setlessthan_te...	0000000000000...
/my_setlessthan_te...	0
/my_setlessthan_te...	1111111111111...
/my_setlessthan_te...	St0
/my_setlessthan_te...	St1

```
#         Region: /my_setlessthan_testbench/try_set/s0/x31
# ** Warning: (vsim-3015) C:/altera/13.1/workspace/1901042705/my_subtractor.v(48): [PCDPC] - Port size (32 or 32) does not match connection size (1) for port 'result'. The port definition is at: C:/altera/13.1/workspace/1901042705/my_xor.v(4).
#
#         Region: /my_setlessthan_testbench/try_set/s0/x31
add wave -position insertpoint  \
sim:/my_setlessthan_testbench/A \
sim:/my_setlessthan_testbench/B \
sim:/my_setlessthan_testbench/C0 \
sim:/my_setlessthan_testbench/Res \
sim:/my_setlessthan_testbench/C \
sim:/my_setlessthan_testbench/setres
VSIM 5> step -current
# Testing the setlessthan
#  setlessthan: time =  0, a =0001, b=0000, set=0
# Testing the setlessthan
#  setlessthan: time =  5, a =0006, b=0008, set=1

VSIM 6>
```



```
VSIM 5> step -current
# Testing the setlessthan
#  setlessthan: time =  0, a =0001, b=0000, set=0
# Testing the setlessthan
#  setlessthan: time =  5, a =0006, b=0008, set=1
```

✓ Operation of the 32-bit nor (module my_nor)

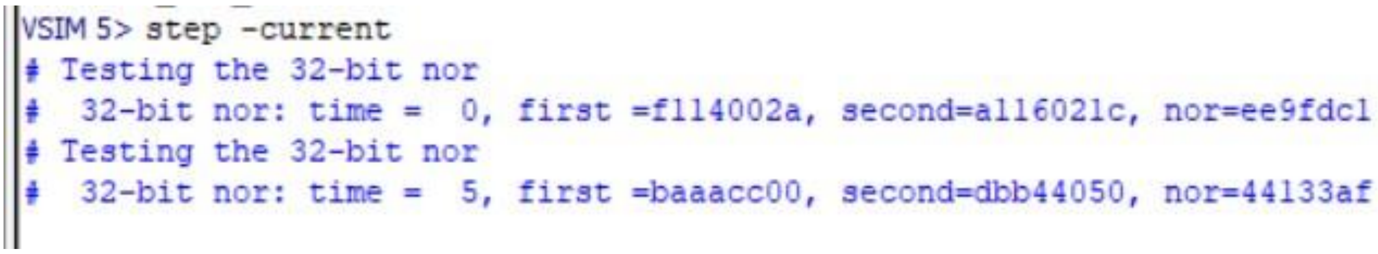


Instance	Design unit	Design unit type
my_nor_testbench	my_nor_tes...	Module
try_nor	my_nor	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
first	10111...	Net	In
second	11011...	Net	In
result	00000...	Net	Out

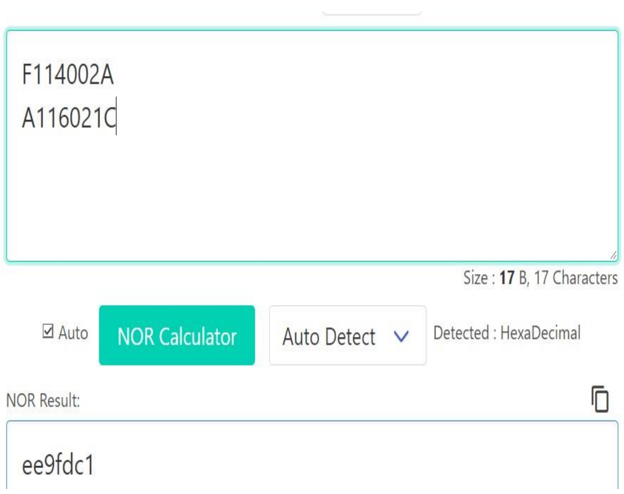
```
/my_nor_testbench...   1011101010101...
/my_nor_testbench...   1101101110110...
/my_nor_testbench...   0000010001000...
Now       0.005 ns
Cursor 1  0.00 ns     0.00 ns
```

```
# -- Compiling module my_nor_testbench
#
# Top level modules:
#       my_nor_testbench
ModelSim> vsim work.my_nor_testbench
# vsim work.my_nor_testbench
# Loading work.my_nor_testbench
# Loading work.my_nor
add wave -position insertpoint  \
sim:/my_nor_testbench/first \
sim:/my_nor_testbench/second \
sim:/my_nor_testbench/result
VSIM 5> step -current
# Testing the 32-bit nor
#  32-bit nor: time =  0, first =f114002a, second=a116021c, nor=ee9fdc1
# Testing the 32-bit nor
#  32-bit nor: time =  5, first =baaacc00, second=dbb44050, nor=44133af

VSIM 6>
```



```
VSIM 5> step -current
# Testing the 32-bit nor
#  32-bit nor: time =  0, first =f114002a, second=a116021c, nor=ee9fdc1
# Testing the 32-bit nor
#  32-bit nor: time =  5, first =baaacc00, second=dbb44050, nor=44133af
```

Test1:



F114002A
A116021C

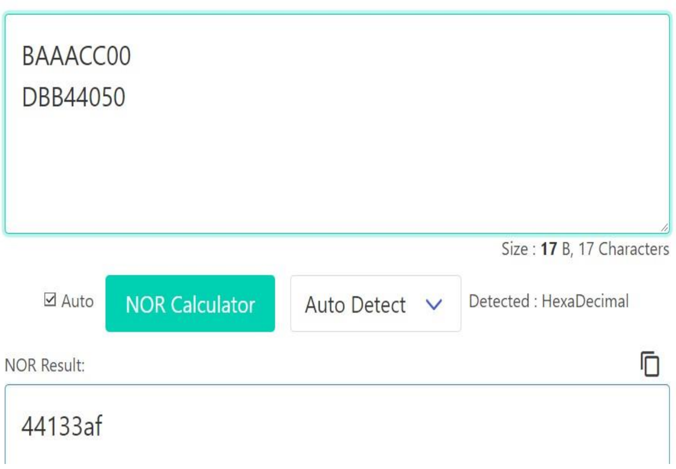
Size : **17** B, 17 Characters

Detected : HexaDecimal

NOR Result:

ee9fdc1

Test2:



BAAACC00
DBB44050

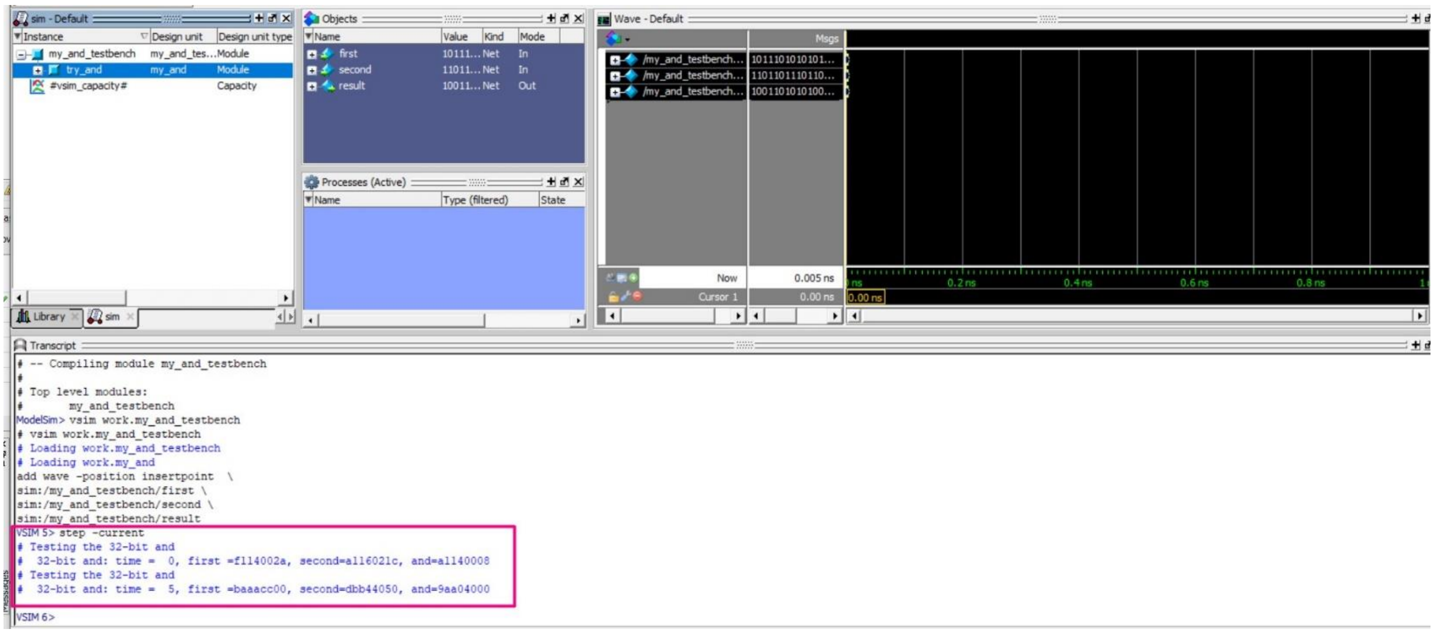
Size : **17** B, 17 Characters

Detected : HexaDecimal

NOR Result:

44133af

✓ Operation of the 32-bit and (module my_and)

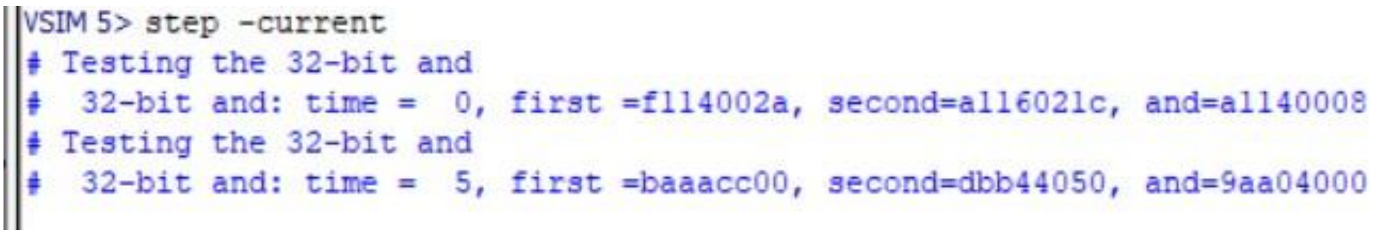


Instance	Design unit	Design unit type
my_and_testbench	my_and_tes...	Module
try_and	my_and	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
first	10111...	Net	In
second	11011...	Net	In
result	10011...	Net	Out

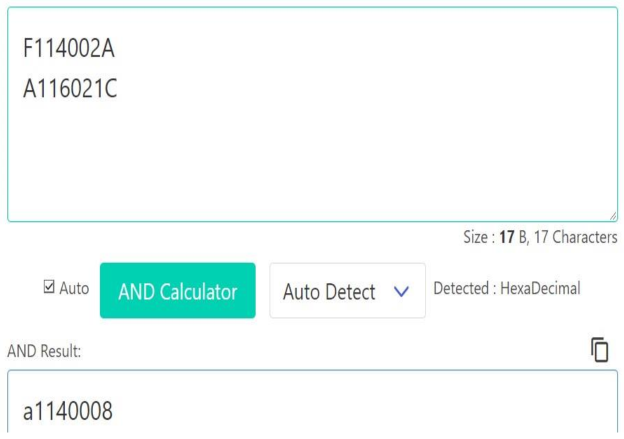
	Msgs
/my_and_testbench...	101110101010 1...
/my_and_testbench...	1101101110110...
/my_and_testbench...	1001101010100...
Now	0.005 ns
Cursor 1	0.00 ns

```
# -- Compiling module my_and_testbench
#
# Top level modules:
#       my_and_testbench
ModelSim> vsim work.my_and_testbench
# vsim work.my_and_testbench
# Loading work.my_and_testbench
# Loading work.my_and
add wave -position insertpoint  \
sim:/my_and_testbench/first \
sim:/my_and_testbench/second \
sim:/my_and_testbench/result
VSIM 5> step -current
# Testing the 32-bit and
#  32-bit and: time =  0, first =f114002a, second=a116021c, and=a1140008
# Testing the 32-bit and
#  32-bit and: time =  5, first =baaacc00, second=dbb44050, and=9aa04000
VSIM 6>
```



```
VSIM 5> step -current
# Testing the 32-bit and
#  32-bit and: time =  0, first =f114002a, second=a116021c, and=a1140008
# Testing the 32-bit and
#  32-bit and: time =  5, first =baaacc00, second=dbb44050, and=9aa04000
```

Test1:



F114002A
A116021C

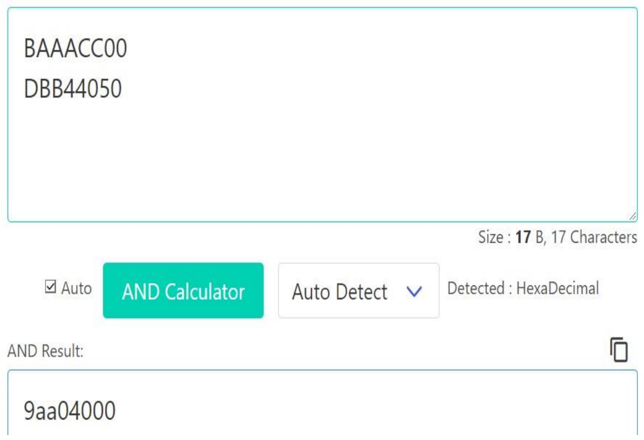
Size : **17** B, 17 Characters

☑ Auto | AND Calculator | Auto Detect | Detected : HexaDecimal

AND Result:

a1140008

Test2:



BAAACC00
DBB44050

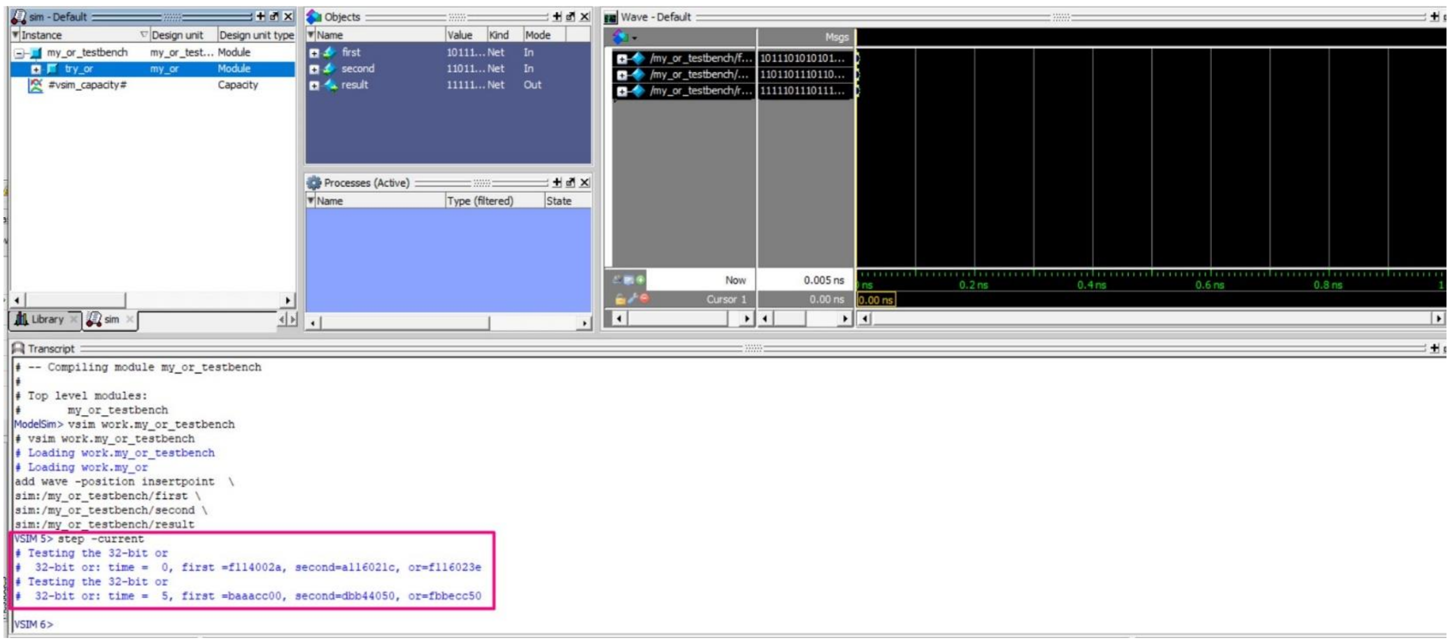
Size : **17** B, 17 Characters

☑ Auto | AND Calculator | Auto Detect | Detected : HexaDecimal

AND Result:

9aa04000

✓ Operation of the 32-bit or (module my_or)



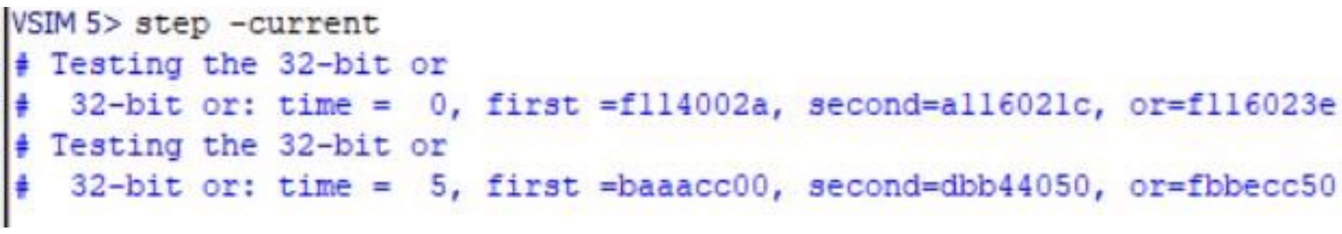
Instance	Design unit	Design unit type
my_or_testbench	my_or_test...	Module
try_or	my_or	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
first	10111...	Net	In
second	11011...	Net	In
result	11111...	Net	Out

Name	Type (filtered)	State

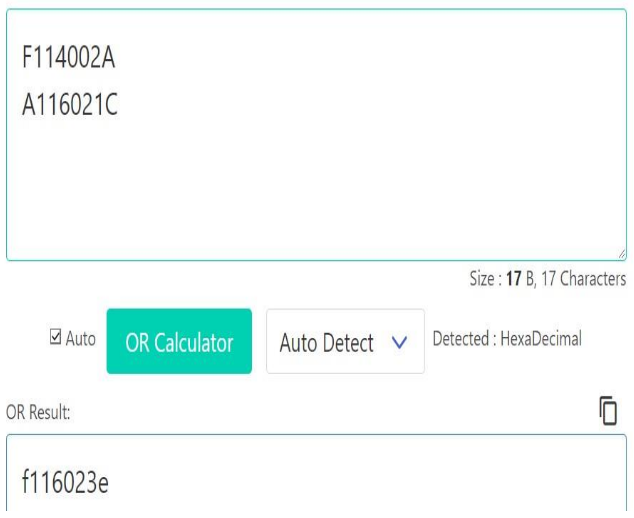
	Msgs
/my_or_testbench/f...	1011101010101...
/my_or_testbench/...	1101101110110...
/my_or_testbench/r...	1111101110111...
Now	0.005 ns
Cursor 1	0.00 ns

```
# -- Compiling module my_or_testbench
#
# Top level modules:
#       my_or_testbench
ModelSim> vsim work.my_or_testbench
# vsim work.my_or_testbench
# Loading work.my_or_testbench
# Loading work.my_or
add wave -position insertpoint  \
sim:/my_or_testbench/first \
sim:/my_or_testbench/second \
sim:/my_or_testbench/result
VSIM 5> step -current
# Testing the 32-bit or
#  32-bit or: time =  0, first =f114002a, second=a116021c, or=f116023e
# Testing the 32-bit or
#  32-bit or: time =  5, first =baaacc00, second=dbb44050, or=fbbecc50

VSIM 6>
```



```
VSIM 5> step -current
# Testing the 32-bit or
#  32-bit or: time =  0, first =f114002a, second=a116021c, or=f116023e
# Testing the 32-bit or
#  32-bit or: time =  5, first =baaacc00, second=dbb44050, or=fbbecc50
```

Test1:



F114002A
A116021C

Size : **17** B, 17 Characters

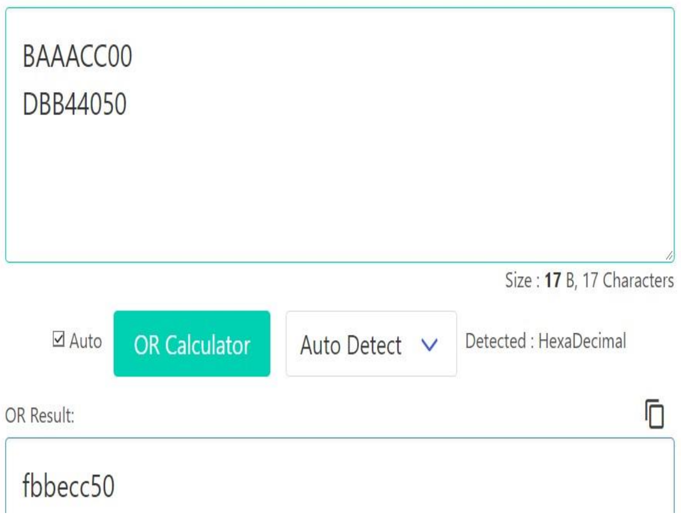
☑ Auto

Detected : HexaDecimal

OR Result:

f116023e

Test2:



BAAACC00
DBB44050

Size : **17** B, 17 Characters

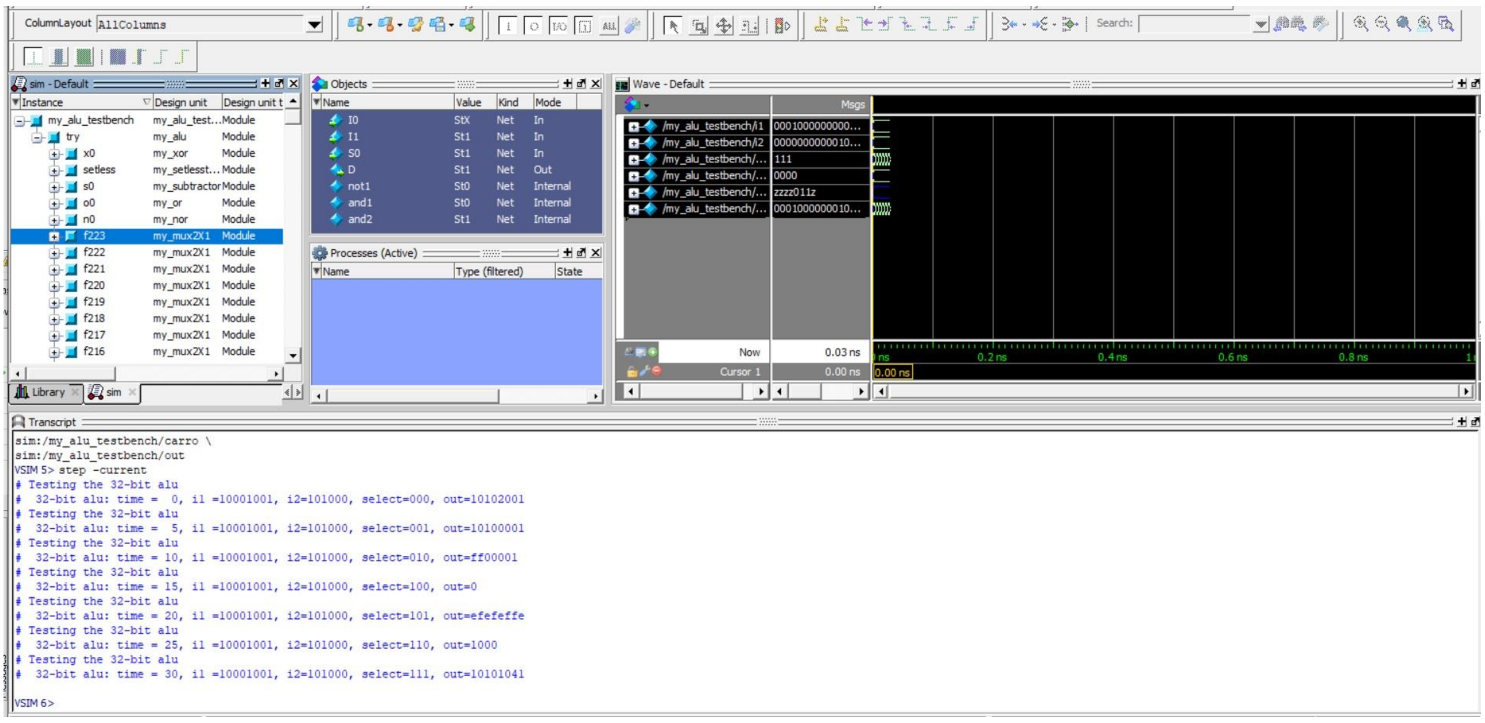
☑ Auto

Detected : HexaDecimal

OR Result:

fbbecc50

✓ 32-bit alu (module my_alu)

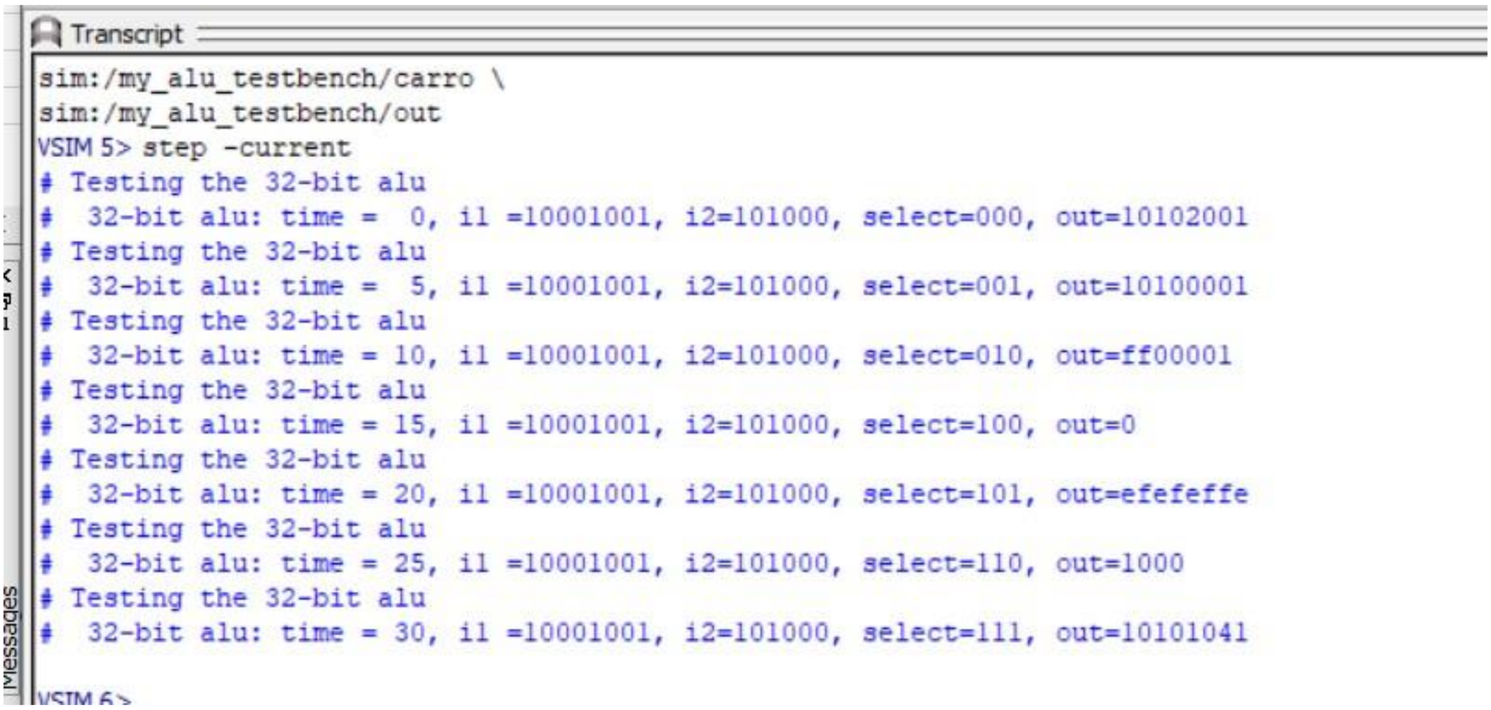


Instance	Design unit	Design unit t
my_alu_testbench	my_alu_test...	Module
try	my_alu	Module
x0	my_xor	Module
setless	my_setlesst...	Module
s0	my_subtractor	Module
o0	my_or	Module
n0	my_nor	Module
f223	my_mux2X1	Module
f222	my_mux2X1	Module
f221	my_mux2X1	Module
f220	my_mux2X1	Module
f219	my_mux2X1	Module
f218	my_mux2X1	Module
f217	my_mux2X1	Module
f216	my_mux2X1	Module

Name	Value	Kind	Mode
I0	StX	Net	In
I1	St1	Net	In
S0	St1	Net	In
D	St1	Net	Out
not1	St0	Net	Internal
and1	St0	Net	Internal
and2	St1	Net	Internal

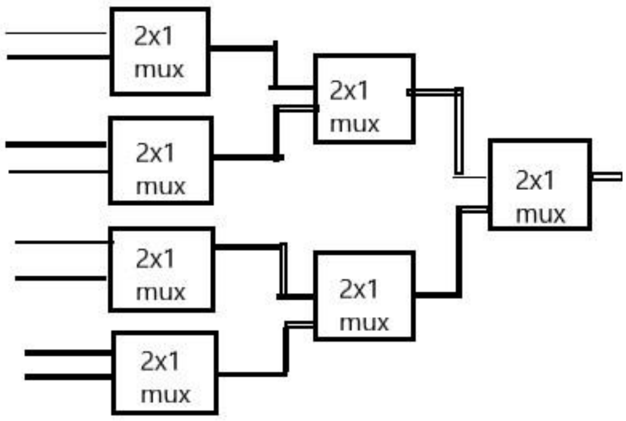
	Msgs
/my_alu_testbench/i1	0001000000000...
/my_alu_testbench/i2	000000000010...
/my_alu_testbench/...	111
/my_alu_testbench/...	0000
/my_alu_testbench/...	zzzz011z
/my_alu_testbench/...	000100000010...
Now	0.03 ns
Cursor 1	0.00 ns

```
sim:/my_alu_testbench/carro \
sim:/my_alu_testbench/out
VSIM 5> step -current
# Testing the 32-bit alu
#  32-bit alu: time =  0, i1 =10001001, i2=101000, select=000, out=10102001
# Testing the 32-bit alu
#  32-bit alu: time =  5, i1 =10001001, i2=101000, select=001, out=10100001
# Testing the 32-bit alu
#  32-bit alu: time = 10, i1 =10001001, i2=101000, select=010, out=ff00001
# Testing the 32-bit alu
#  32-bit alu: time = 15, i1 =10001001, i2=101000, select=100, out=0
# Testing the 32-bit alu
#  32-bit alu: time = 20, i1 =10001001, i2=101000, select=101, out=efefeffe
# Testing the 32-bit alu
#  32-bit alu: time = 25, i1 =10001001, i2=101000, select=110, out=1000
# Testing the 32-bit alu
#  32-bit alu: time = 30, i1 =10001001, i2=101000, select=111, out=10101041

VSIM 6>
```



```
sim:/my_alu_testbench/carro \
sim:/my_alu_testbench/out
VSIM 5> step -current
# Testing the 32-bit alu
#  32-bit alu: time =  0, i1 =10001001, i2=101000, select=000, out=10102001
# Testing the 32-bit alu
#  32-bit alu: time =  5, i1 =10001001, i2=101000, select=001, out=10100001
# Testing the 32-bit alu
#  32-bit alu: time = 10, i1 =10001001, i2=101000, select=010, out=ff00001
# Testing the 32-bit alu
#  32-bit alu: time = 15, i1 =10001001, i2=101000, select=100, out=0
# Testing the 32-bit alu
#  32-bit alu: time = 20, i1 =10001001, i2=101000, select=101, out=efefeffe
# Testing the 32-bit alu
#  32-bit alu: time = 25, i1 =10001001, i2=101000, select=110, out=1000
# Testing the 32-bit alu
#  32-bit alu: time = 30, i1 =10001001, i2=101000, select=111, out=10101041

VSIM 6>
```

-> This is how I thought of the 8x1 mux.


2x1
mux
2x1
mux
2x1
mux
2x1
mux
2x1
mux
2x1
mux
2x1
mux