



inputs				Outputs				
C2	C1	Product0	counterEquals32	incrementCounter	write	shift	N2	N1
0	0	0	X	X	X	X	1	0
0	0	1	X	X	X	X	0	1
0	1	X	X	X	1	X	1	0
1	0	X	0	1	X	1	0	0
1	0	X	1	1	X	1	1	1

$$N2 = C2' C1' Product0' + C2' C1 + C2 C1' CounterEquals32$$

$$= C2' (C1' Product0' + C1) + C2 C1' CounterEquals32$$

$$N1 = C2' C1' Product0 + C2 C1' CounterEquals32 = C1' (C2' Product0 + C2 CounterEquals32)$$

$$write = C2' C1$$

$$shift = incrementCounter = C2 C1' \rightarrow \text{increment counter is unnecessary.}$$

Adder32bit.v uses 1 h_adder module for the first bit and calculates other bits with full_adder module

Adder Test:

```
add wave -position insertpoint \
sim:/adder32bit_testbench/a \
sim:/adder32bit_testbench/b \
sim:/adder32bit_testbench/sum
VSIM 212> run
# time = 0, a =ffff, b=1, sum=10000
# time = 14, a =fffffff, b=1, sum=10000000
# time = 28, a =abcd6789, b=aaaa, sum=abce1233
# time = 3c, a =abcffff, b=1, sum=abd0000
```

VSIM 213>

Xor32.v, nor32.v, and32.v and or32.v files uses appropriate gate for 32 times.

Test of these modules:

Xor32 test:

```
sim:/xor32_test/sim
VSIM 112> run
# time = 0, a =abcd, b=a987, a-b=1146
# time = 20, a =ff, b=f0, a-b=f
# time = 40, a =a0, b=a, a-b=96
# time = 60, a =ffffff, b=1, a-b=fffffe
VSIM 113> vsim -voptargs+=acc work.xor32_test
# vsim -voptargs+=acc work.xor32_test
# Loading work.xor32_test
# Loading work.xor32
add wave -position insertpoint \
sim:/xor32_test/a \
sim:/xor32_test/b \
sim:/xor32_test/answer
VSIM 115> run
# time = 0, a =ffffff xor b=0 => answer=ffffff
# time = 14, a =f0f0f0f xor b=f0f0f0f0 => answer=ffffff
# time = 28, a =abcd6789 xor b=1111aaaa => answer=badccd23
# time = 3c, a =fff4ffff xor b=1 => answer=fff4ffff
VSIM 116>
```

I. Input: hexadecimal (base 16) ▼
abcd6789

II. Input: hexadecimal (base 16) ▼
1111aaaa

Calculate XOR

III. Output: hexadecimal (base 16) ▼
badccd23

Nor32 gate:

```
add wave -position insertpoint \
sim:/nor32_testv/a \
sim:/nor32_testv/b \
sim:/nor32_testv/answer
VSIM 124> run
# time = 0, a =ffffffff, b=abcd, answer=0
# time = 14, a =f0f0f0f, b=f0f0f0f0, answer=0
# time = 28, a =abcd6789, b=1111aaaa, answer=44221054
# time = 3c, a =fff4ffff, b=1, answer=b0000
```

VSIM 125>

And32 gate:

```
VSIM 6> run
# time = 0, a =ffffffff, b=0, answer=0
# time = 14, a =f0f0f0f, b=f0f0f0f0, answer=0
# time = 28, a =abcd6789, b=1111aaaa, answer=1012288
# time = 3c, a =fff4ffff, b=1, answer=1
```

Or32 gate:

```
add wave -position insertpoint \
sim:/or32_test/a \
sim:/or32_test/b \
sim:/or32_test/answer
VSIM 121> run
# time = 0, a =ffffffff, b=0, answer=ffffffff
# time = 14, a =f0f0f0f, b=f0f0f0f0, answer=ffffffff
# time = 28, a =abcd6789, b=1111aaaa, answer=bbddefab
# time = 3c, a =fff4ffff, b=1, answer=fff4ffff
VSIM 122>
```

Sub32 uses add32 to subtraction. First takes second number's two's complement and calls 32 bit adder.

Sub32 module test:

```
# Loading work.half_adder
# Loading work.full_adder
add wave -position insertpoint \
sim:/sub32_test/a \
sim:/sub32_test/b \
sim:/sub32_test/sub
VSIM 112> run
# time = 0, a =bacd, b=a987, a-b=1146
# time = 20, a =ff, b=f0, a-b=f
# time = 40, a =a0, b=a, a-b=96
# time = 60, a =ffffff, b=1, a-b=ffffffe
VSIM 113>
```

Hexadecimal Calculator

Result

Hex value:
bacd – a987 = **1146**

Decimal value:
47821 – 43399 = **4422**

Mult is not provided.

Less32 module uses subtraction module. Sign bit of answer of subtraction is returned as answer.

Less32 module test:

```
# Loading work_full_alu1
add wave -position insertpoint \
sim:/less32_test/a \
sim:/less32_test/b \
sim:/less32_test/answer
VSIM 179> run
# time = 0, a =160, b=10, A<B=0
# time = 20, a =240, b=255, A<B=1
# time = 40, a =11, b=10, A<B=0
# time = 60, a =1, b=268435455, A<B=1
VSIM 180>
```

Alu32 module:

I designed multiplexer which takes 8 input which have 32 bit. Multiplexer selects one of them according to the alu code given. And operation is used to make answers of other calculations 0. At the and or gate is used to take the answer which is not 0 (of course, if the actual answer is not 0).

```
# time = 0, a =a0, b=a, result= 000000aa aluOp=0
# time = 5, a =140e, b=23a2, result= 000037b0 aluOp=0
# time = 10, a =ffff0000, b=ff00ff00, result= 00ffff00 aluOp=1
# time = 15, a =f0f00f0f, b=ff00ff00, result= 0ff0f00f aluOp=1
# time = 20, a =a0b0, b=a000, result= 000000b0 aluOp=10
# time = 25, a =a0be, b=a03f, result= 0000007f aluOp=10
# time = 30, a =fff0000, b=ffff, result= 00000000 aluOp=100
# time = 35, a =1000, b=ffffff, result= 00000001 aluOp=100
# time = 40, a =ffff0000, b=ff00ff00, result= 000000ff aluOp=101
# time = 45, a =aaaa0000, b=aa00aa00, result= 555555ff aluOp=101
# time = 50, a =ffff0000, b=ff00ff00, result= ff000000 aluOp=110
# time = 55, a =aaaa0000, b=aaaaa, result= 000a0000 aluOp=110
# time = 60, a =ffff0000, b=ff00ff00, result= ffffff00 aluOp=111
# time = 65, a =aaaa0000, b=aa00aa00, result= aaaaaa00 aluOp=111
```

Control module is designed according to boolean expressions found from table.

Control module test:

```

sim:/control_tb/shift \
sim:/control_tb/multiplicand \
sim:/control_tb/counter
VSIM 182> run
# time = 0, PS = xx, NS=x0 write=x,shift=x, counterEquals32 = 0 counter= 0
# time = 5, PS = 00, NS=10 write=0,shift=0, counterEquals32 = 0 counter= 0
# time = 15, PS = 10, NS=00 write=0,shift=1, counterEquals32 = 0 counter= 0
# time = 25, PS = 00, NS=10 write=0,shift=0, counterEquals32 = 0 counter= 0
# time = 35, PS = 10, NS=00 write=0,shift=1, counterEquals32 = 0 counter= 0
# time = 45, PS = 00, NS=10 write=0,shift=0, counterEquals32 = 0 counter= 0
# time = 55, PS = 10, NS=00 write=0,shift=1, counterEquals32 = 0 counter= 0
# ** Note: $finish      : D:/ORG_HW3/control_tb.v(49)
#   Time: 65 ps  Iteration: 0  Instance: /control_tb
# 1
# Break in Module control_tb at D:/ORG_HW3/control_tb.v line 49
VSIM 183>

```

Shift right module is designed to use in datapath but because of datapath is not implemented, it is not used.

Shift right module:

```

add wave -position insertpoint \
sim:/shift_right_tb/a \
sim:/shift_right_tb/answer
VSIM 118> run
# time = 0, a =ffffff, answer=7ffffff
# time = 14, a =f0f0f0f, answer=7878787
# time = 28, a =abcd6789, answer=55e6b3c4
# time = 3c, a =fff4ffff, answer=7ffa7fff
VSIM 119>

```

Datapath is not completed.