

Lab 2: 16/32-bit ALU

Synopsis

You are required to write a **16-bit ALU**, a **32-bit ALU**, and **testbenches for both**. The ALUs should take two Two's-complement numbers as input (A and B), a select input, and perform the following operations:

Operation	Value of select lines
Addition (Output = A + B)	0
Subtraction (Output = A - B)	1
Negate A (Output = -A)	2
AND (Output = A AND B)	3
OR (Output = A OR B)	4
XOR (Output = A XOR B)	5
Pass Through A (Output = A)	6
Pass Through B (Output = B)	7

Each input (A and B) to the 16-bit ALU should be 16-bits wide, and the output should also be 16-bits wide. Similarly, the inputs and outputs for the 32-bit ALU should be 32-bits wide. The select lines should be 3-bits wide.

In addition to the normal output, there should be two additional output signals: overflow and underflow. Overflow should go high when the result of an addition exceeds the output's capability. Underflow should go high when the result of a subtraction is smaller than the output can handle.

Overflow occurs if:

-- we add two positive numbers and get a negative number

$$127 + 127 =$$

01111111 + 01111111 = 11111110 which is not correct, so we need an "overflow"

or

---we add two negative numbers and get a positive number.

$$-128 + (-128) =$$

10000000 + 10000000 = 100000000 which is clearly an overflow since there is one additional bit to the very left.

Underflow occurs if

--- a positive number subtracts a negative and results in a negative

$$1 - (-128) =$$

00000001 - 10000000 = 10000001, which is not correct, so we need an “underflow”

or

--- a negative subtracts a positive and results in a positive

$$(-128) - 1 =$$

10000000 - 00000001 = 01111111, which is not correct because we’ve got a positive number! Therefore, we need an underflow signal.

What to turn in

Turn in the VHDL code of the 16-bit ALU, 32-bit ALU, and testbenches (**all together 4**) only. Be sure to include your name on all files. For this assignment you can turn in one submission per partnership, but it is in your best interest to be sure that everyone knows how the code works and how to create it as you will be using VHDL for more assignments and may be tested on it.

Important notes:

1. turn in one time only
2. turn in only the VHDL files
3. Include your teammate(s)’s name and group member as part of the comment of your file.
4. testbench should cover all the possible case and functions.