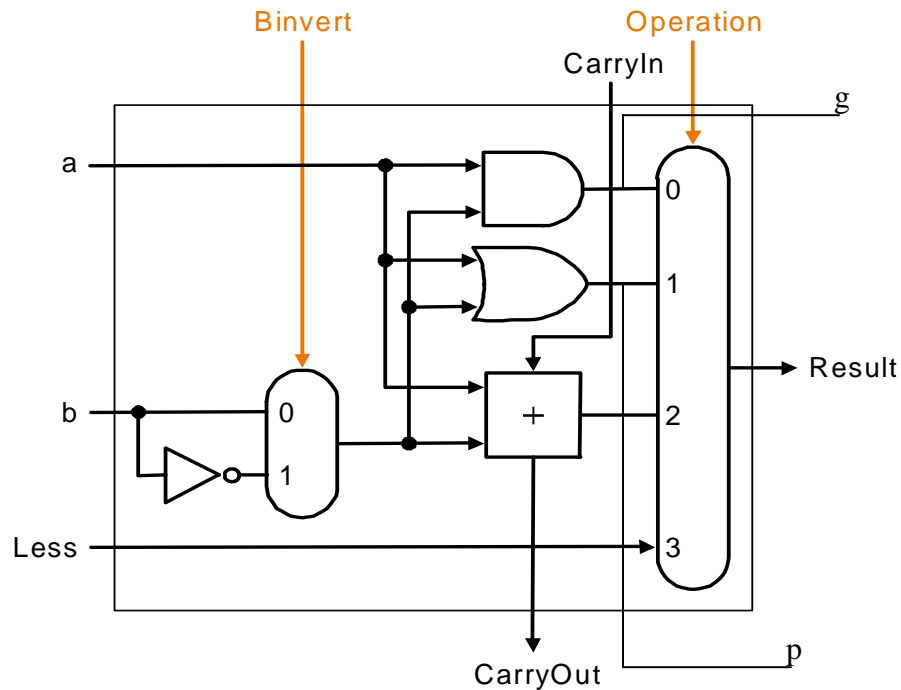


Lab 3  
Due 3:00 p.m. 9/21/09

**Work individually on this lab, prepare a formal lab report, and on the report cover page sign the pledge: “On my honor I have neither given nor received unauthorized aid on this lab.” Highlight and explain all changes and additions you make to the code. In the lab report include a simulation waveform from 0 to 400 ns. You may confer with other students regarding specific questions about VHDL; however, if you need help with the actual assignment ask only the instructor or TA.**

For your lab, you are to complete architecture in the file `alu1.vhd` and correct the assert statements in the test bench `tb_alu1.vhd`.

The book’s 1-bit ALU is shown below. Complete the VHDL entity architecture for a modified version of this 1-bit ALU. For the modified version add the capability to invert not only the `b` input but also the `a` input. Also connect the output of the “and” and “or” gates to the new outputs `p` and `g` where `p` is the output of the “or” gate, and `g` is the output of the “and” gate. Assume all gates have a delay of 1 ns, each mux has a delay of 2 ns, and both outputs of the adder have a delay of 2 ns. Your architecture should model these delays.



Use the provided test bench to verify your design. Run the simulation and notice that the assert statements produce wrong error reports. Fix this problem.

Operation	op
and	00
or	01
+ (also -)	10
Result <= Less	11

You should fully document your code, and in the lab report briefly explain the decisions you made. Include in your lab report simulation waveforms to verify the ALU1 operation, and before 3:00 p.m. on 9/21/09 email to [jhtucker@vcu.edu](mailto:jhtucker@vcu.edu) your source code for the lab. Use "EGRE426 Lab 3-your name" as the subject line. The correct simulation waveform is shown below. Turn in your lab report at the beginning of class on 9/22/09.

```
-- alu_n.vhd Version 1.1
-- Uses  NONE
LIBRARY IEEE;
USE work.all;
USE IEEE.Std_Logic_1164.all;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-----
-- ENTITY: ALU1 - One bit ALU as specified in class.
-----

entity alu1 is
  port(a, b: in  std_logic;
        Ainvert: in  std_logic;
        Binvert: in  std_logic;
        less: in std_logic;
        op:  in  unsigned(1 downto 0);
        cin: In std_logic;
        Result: out std_logic;
        cout: out std_logic;
        p: out std_logic;
        g: out std_logic);
end entity alu1;

architecture arch of ALU1 is

begin

end;
```

