library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity pcbranch is

port( rx\_in : in std\_logic\_vector(7 downto 0); -- connects to rx from regbank

opcode : in std\_logic\_vector(3 downto 0); --connects to alu\_opcode not select lines

offset : in std\_logic\_vector(7 downto 0); -- connects to branch from decoder

PC\_jump : in std\_logic\_vector(7 downto 0); --jump address for the PC from the Decoder.

clk : in std\_logic;

pcout : out std\_logic\_vector(7 downto 0) --outputs to instruction mem

);

end entity;

Architecture pcbranch\_arch of pcbranch is

signal pc\_signal : std\_logic\_vector(7 downto 0) := "00000000";

begin

process(rx\_in,opcode,offset,PC\_jump,clk)

begin

----------Jump----------------

if (opcode = "1100") then

if (opcode="1100") then

pc\_signal <= PC\_jump;

elsif (CLK'event and CLK='1') then

--pc\_signal <= pc\_signal + "00000001";

end if;

-----Branch if Zero-----------

elsif (opcode = "1101") then

if (rx\_in = "00000000") then

pc\_signal <= pc\_signal + offset;

elsif (CLK'event and CLK='1') then

pc\_signal <= pc\_signal + "00000001" after 100 ps;

end if;

-----Branch if not Zero0--------

elsif (opcode = "1110") then

if (rx\_in < "00000000" or rx\_in > "00000000") then

pc\_signal <= pc\_signal + offset;

elsif (CLK'event and CLK='1') then

pc\_signal <= pc\_signal + "00000001";

end if;

--increments PC by 1 if no branch instruction-----

elsif (CLK'event and CLK='1') then

pc\_signal <= pc\_signal + "00000001" after 150 ps;

end if;

end process;

pcout <= pc\_signal;

end pcbranch\_arch;