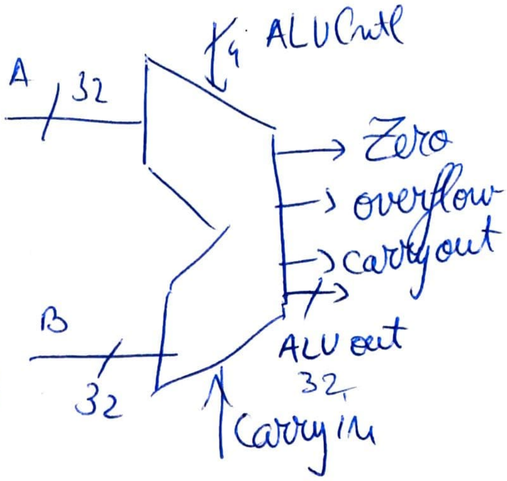
15:47

30413

Problema 2

ALU cnt	Function
1000	And
0001	Or
0011	Xnor
0110	Add
1110	Substract
1100	Multiply by 2 (ALUout=2A)



```
library IEEE;
 use IEEE. std-logic-1164. all;
 use IEEE. std-logic-arith. all;
 use IEEE. std-logic-unsigned. all;
```

~~entity ALU is~~

~~port(A, B: in std_logic_vector(31 down to 0)~~;

~~ALUout: out std_logic_vector (31 downto 0~~

~~(31 downto 0~~));

~~end ALU~~;

~~architecture arh~~ of ALU is

~~begin~~

```
entity ALU is
   port( A, B: in std_logic_vector(31 downto 0);
         ALUcntl: in std_logic_vector( 3 downto 0);
         Carryin: in std_logic;
         zero, overflow, carryout: out std_logic;
         ALUout: out std_logic_vector(31 downto 0));
end ALU;
```

```
architecture arch of ALU is
signal extendedA', B', carryEx : std_logic_vector(32 downto 0);
begin
    process (A, B, ALUcntl)
    begin
        Case ALUcntl is
            when "1000" =>                    -- and
                ~~ALUout <= A-B;~~
                ALUout <= A and B;
                zero <= '0';
                overflow <= '0';
                carryout <= '0';
            when "0001" =>                    -- or
                ALU out <= A or B;
                zero <= '0';
                overflow <= '0';
                carryout <= '0';
            when "0011" =>                    -- xnor
                ALUout <= A xnor B;
                zero <= '0';
                overflow <= '0';
                carryout <= '0';
            when "1100" =>                  -- multiply-by-2
              ALUout <= A(30 downto 0) & '0';  -- shift left
              if A < "100....0" then
                      (31 bits)
                  overflow <= '0';
              else
                  overflow <= '1';
              end if;
              if A = "00.....0" then
                      (32 bits)
                  zero <= '1';
              else
                  zero <= '0';
              endif;
    carryout <= '0';
  ~~when "0110" =>          -- add~~
    ~~A1 <= A '0' & A(31 downto 0);~~
    ~~B1 <= '0' & B(31 downto 0);~~
    ~~extended <= A1 + B1;~~
    ~~ALUout <= extended(31 downto 0);~~
    ~~carryout <= extended(32);~~
    ~~if extended(31 downto 0) = "000...0" then~~
                                 32 bits
          ~~zero <= '1';~~
    ~~else~~
          ~~zero <= '0';~~
    ~~endif;~~
    ~~overflow <= extended(32);~~  ~~-- In this situation I~~
                                 ~~-- considered overflow and~~
                                 ~~-- carryout to be the same~~

  when "1110" =>     -- substract A-B
     if A < B then
         carryout <= '1';
    else
       ~~A~~ carryout <= '0';
    end if;
    overflow <= '0';
    if A=B then
         zero <= '1';
    else
         zero <= '0';
    end if;
    A1 <= '0' & A(31 downto 0);
    B1 <= '0' & B(31 downto 0);
    extended <= A1 - B1;
    ALU out <= extended (31 downto 0);
~~when others =>~~
    ~~ALUout~~
```

```
when "0110" =>   -- add
    A1 <= '0' & A(31 downto 0);
    B1 <= '0' & B(31 downto 0);
    carryEx <= "00....0" & carryin;
                 32 bits
    extended <= A1+B1+ carryEx;
    ALUout <= extended (31 downto 0);
    Carryout <= extended(32);
    overflow <= extended(32);
    if extended (31 downto 0) = "00...0" then
                                 32 bits
          zero <= '1';
    else
          zero <= '0';           -- I considered OF and carry
                                 -- to represent the same
    end if;
 when others =>
    zero <= '0';
    overflow <= '0';
    Carryout <= '0';
    ALUout <= "00...0";
               32 bits
  end case;
 end process;
end arh;
```