**Arjun Thiruchchelvarajah and Caleb Joseph**
**COE328 - Digital Systems - F2022**
**Section 09**
**501100599**

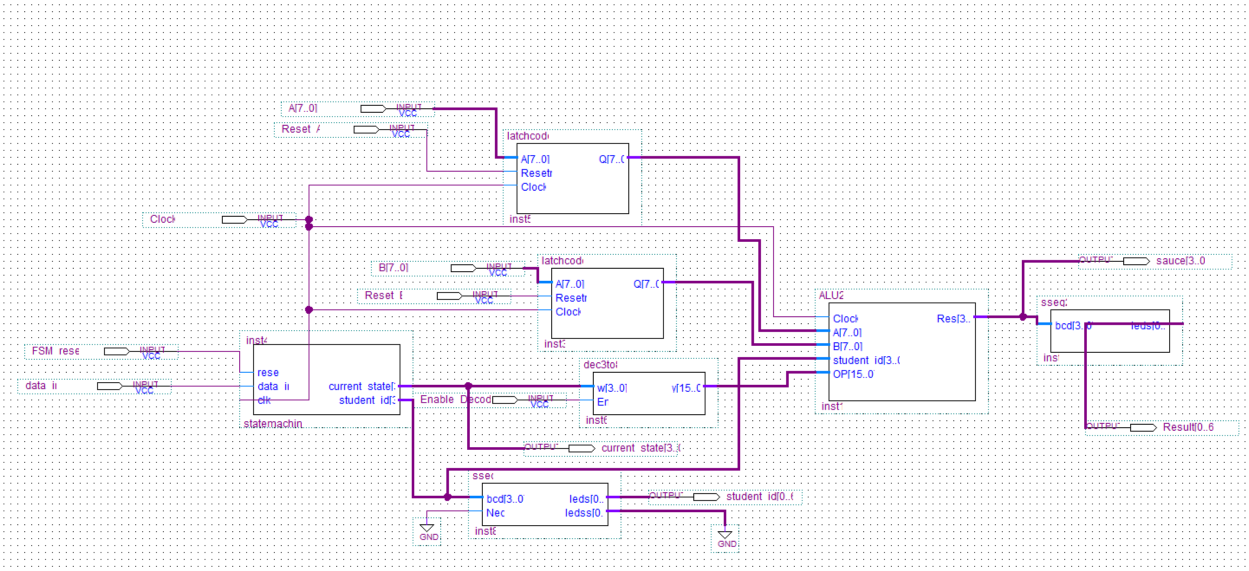
# COE328 LAB 6 - General Purpose ALU Implementation

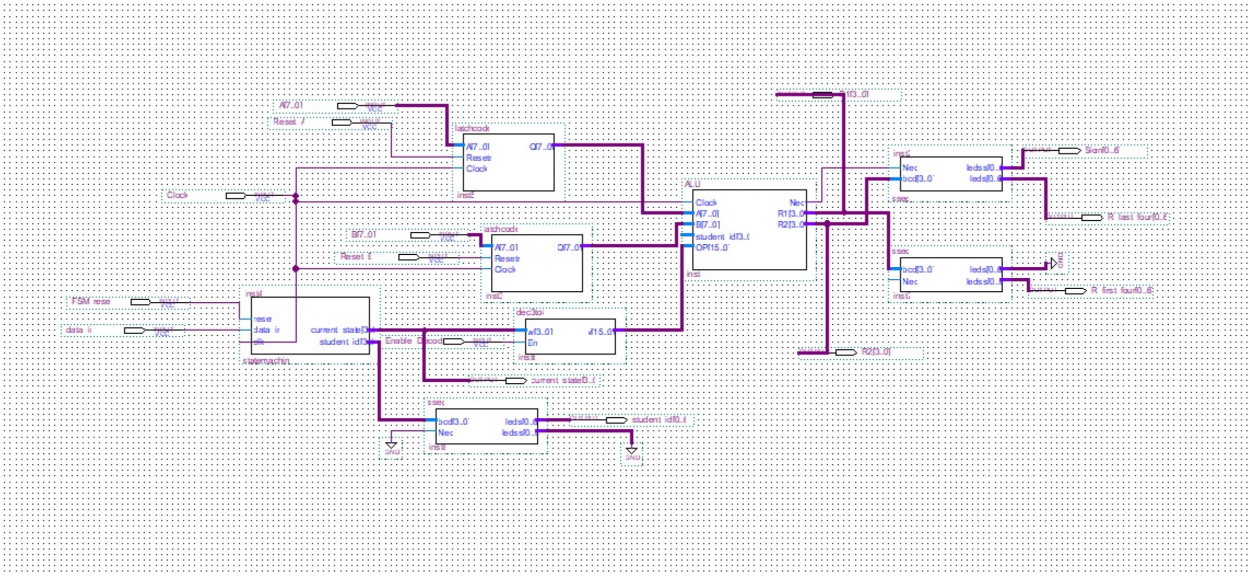
## Objective

The purpose of this lab was to design and construct an Arithmetic and Logic Unit (ALU) in a VHDL environment and implement it on an FPGA board. Specifically, design and build all functions of the ALU, and design, simulate the ALU using VHDL using Quartus Simulator.

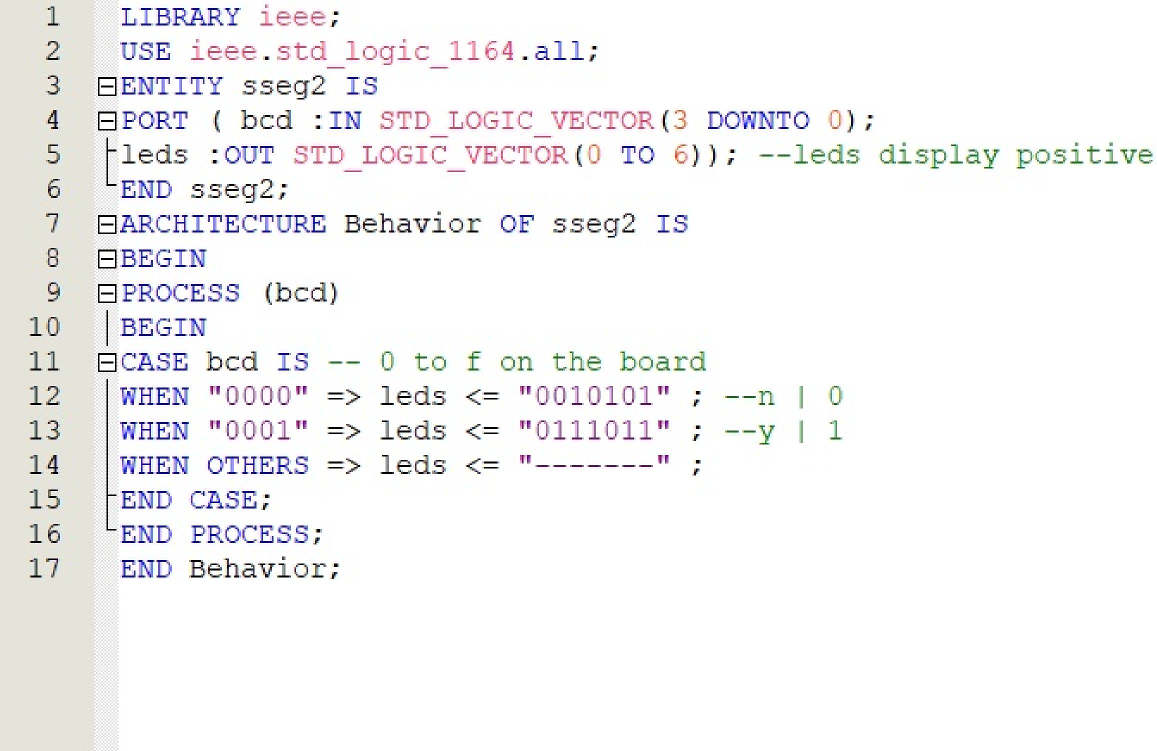
## Theory/Analysis

Block Diagrams:

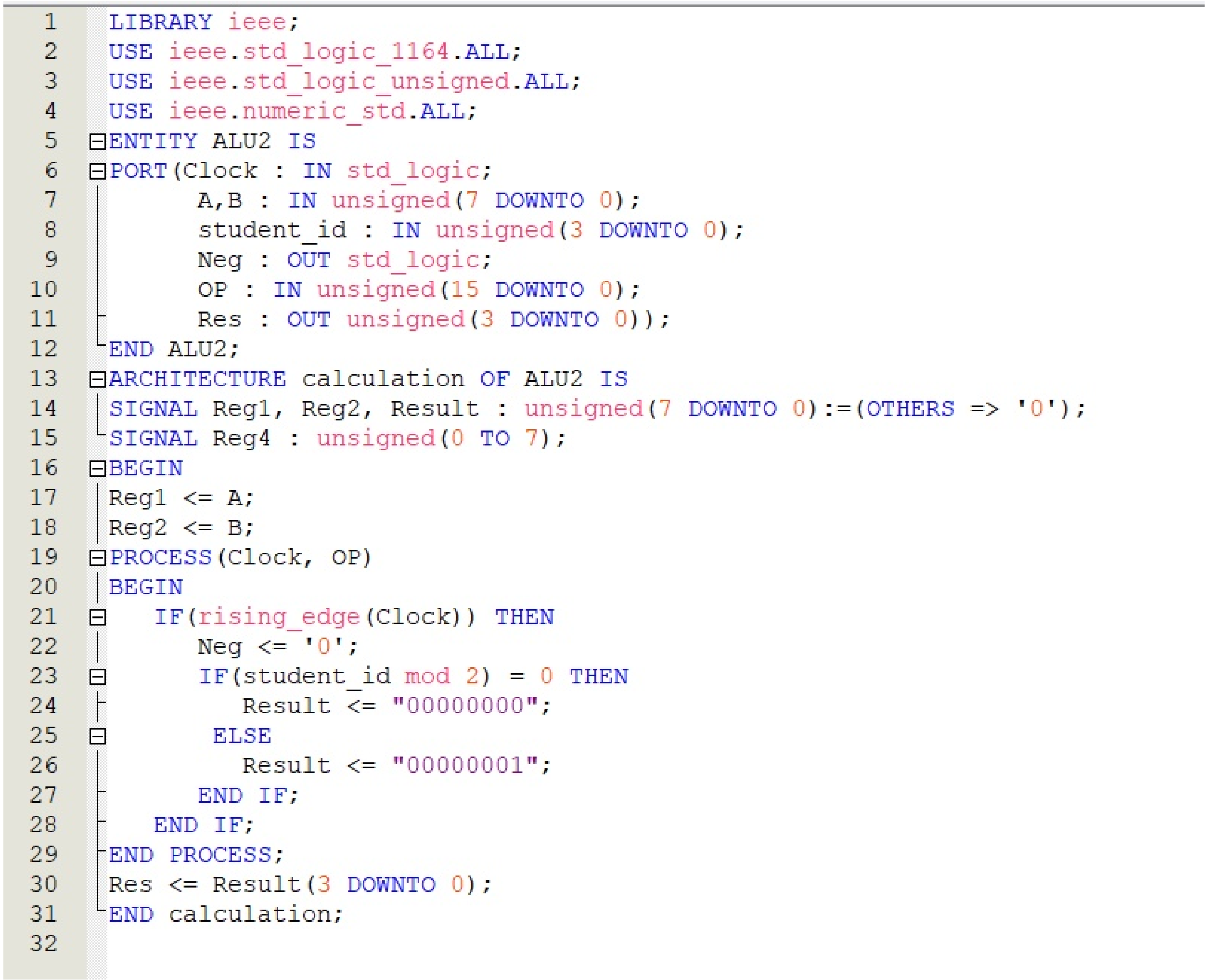



A[7..0]
Reset A
latchcode
A[7..0]
Resetr
Clock
Q[7..0]
inst5
Clock
B[7..0]
Reset B
latchcode
A[7..0]
Resetr
Clock
Q[7..0]
inst2
FSM rese
data in
inst4
reset
data in
clk
current state[3..0]
student id[3..0]
statemachin
Enable Decod
dec3to
w[3..0]
En
y[15..0]
inst8
current state[3..0]
ssec
bcd[3..0]
Nec
leds[0..6]
ledss[0..6]
inst3
student id[0..6]
GND
GND
ALU
Clock
A[7..0]
B[7..0]
student id[3..0]
OP[15..0]
Nec
R1[3..0]
R2[3..0]
inst
R1[3..0]
R2[3..0]
inst6
Nec
bcd[3..0]
ledss[0..6]
leds[0..6]
ssec
Sign[0..6]
R last four[0..6]
ssec
bcd[3..0]
Nec
leds[0..6]
ledss[0..6]
inst7
GND
R first four[0..6]

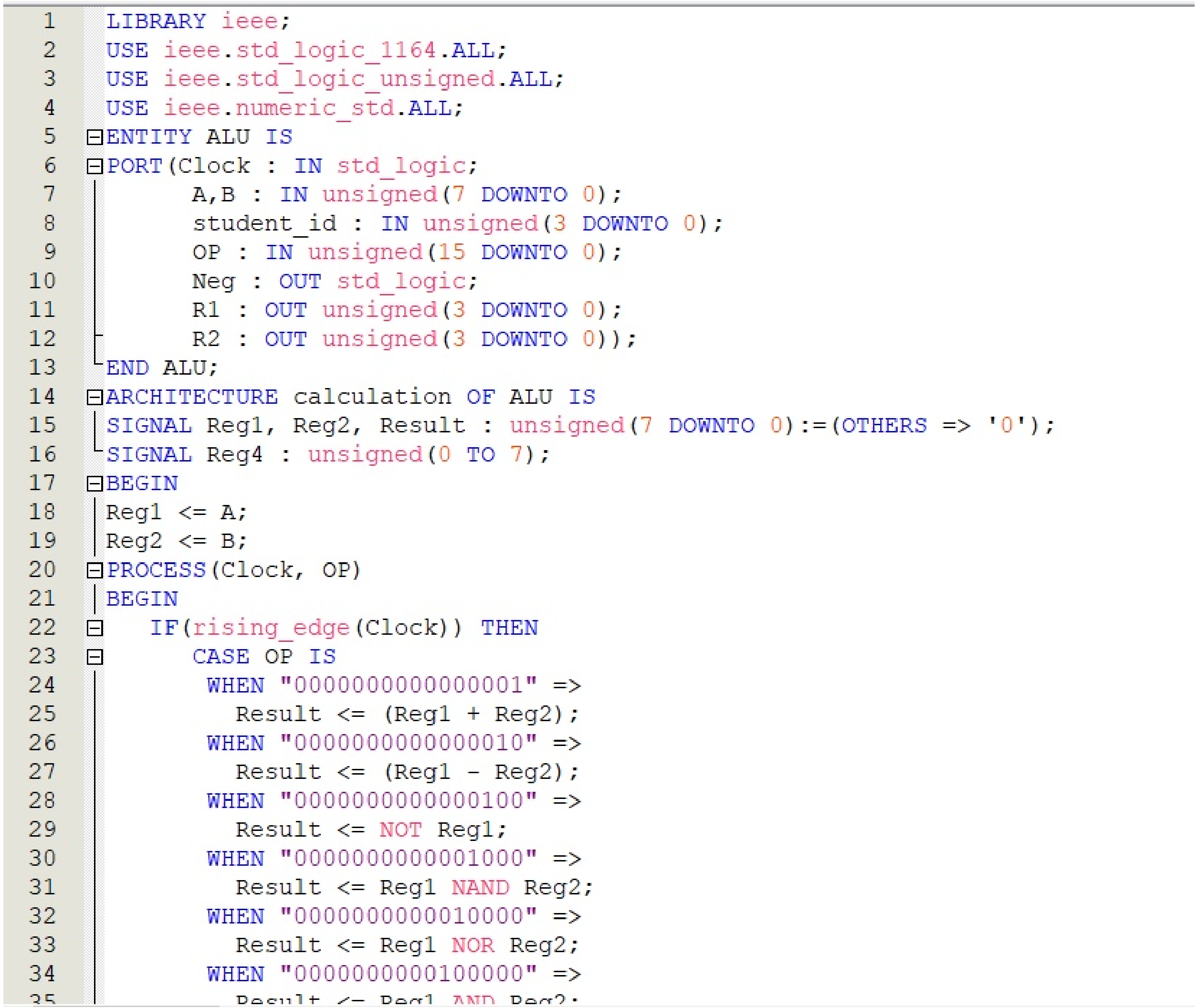
# VHDL CODE:



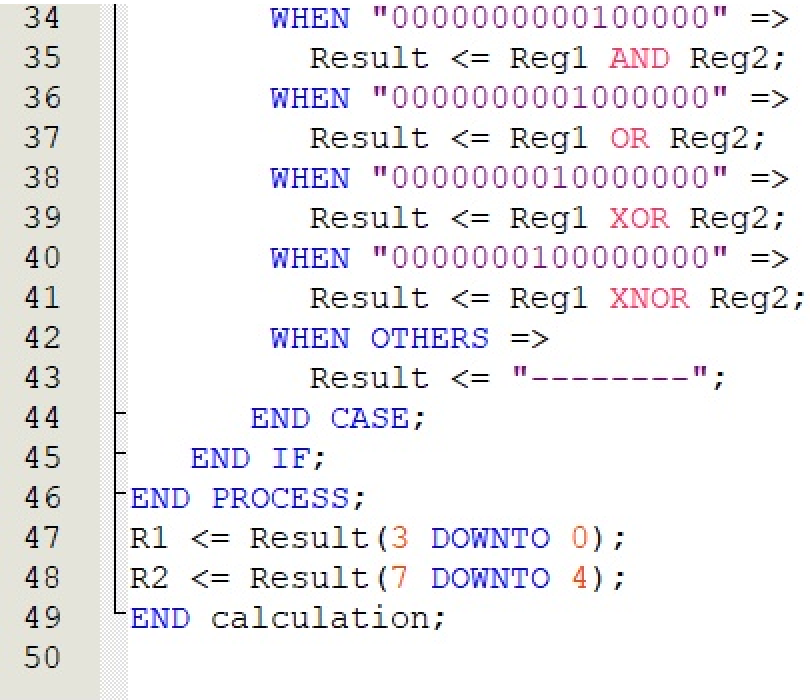
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY sseg2 IS
PORT ( bcd :IN STD_LOGIC_VECTOR(3 DOWNTO 0);
leds :OUT STD_LOGIC_VECTOR(0 TO 6)); --leds display positive
END sseg2;
ARCHITECTURE Behavior OF sseg2 IS
BEGIN
PROCESS (bcd)
BEGIN
CASE bcd IS -- 0 to f on the board
WHEN "0000" => leds <= "0010101" ; --n | 0
WHEN "0001" => leds <= "0111011" ; --y | 1
WHEN OTHERS => leds <= "-------" ;
END CASE;
END PROCESS;
END Behavior;
```



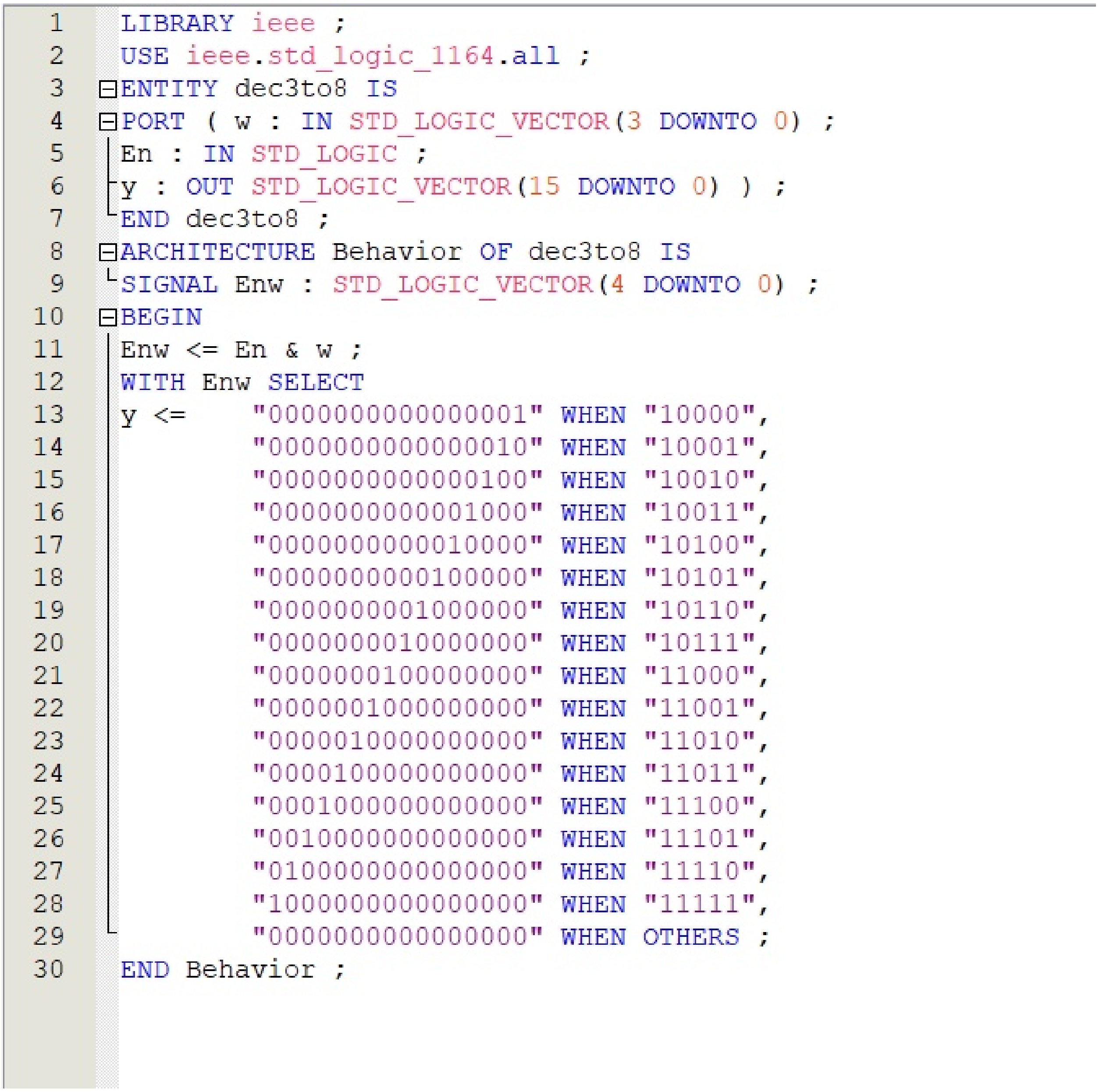
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
USE ieee.numeric_std.ALL;
ENTITY ALU2 IS
PORT(Clock : IN std_logic;
      A,B : IN unsigned(7 DOWNTO 0);
      student_id : IN unsigned(3 DOWNTO 0);
      Neg : OUT std_logic;
      OP : IN unsigned(15 DOWNTO 0);
      Res : OUT unsigned(3 DOWNTO 0));
END ALU2;
ARCHITECTURE calculation OF ALU2 IS
SIGNAL Reg1, Reg2, Result : unsigned(7 DOWNTO 0):=(OTHERS => '0');
SIGNAL Reg4 : unsigned(0 TO 7);
BEGIN
Reg1 <= A;
Reg2 <= B;
PROCESS(Clock, OP)
BEGIN
   IF(rising_edge(Clock)) THEN
      Neg <= '0';
      IF(student_id mod 2) = 0 THEN
         Result <= "00000000";
       ELSE
         Result <= "00000001";
      END IF;
   END IF;
END PROCESS;
Res <= Result(3 DOWNTO 0);
END calculation;
```



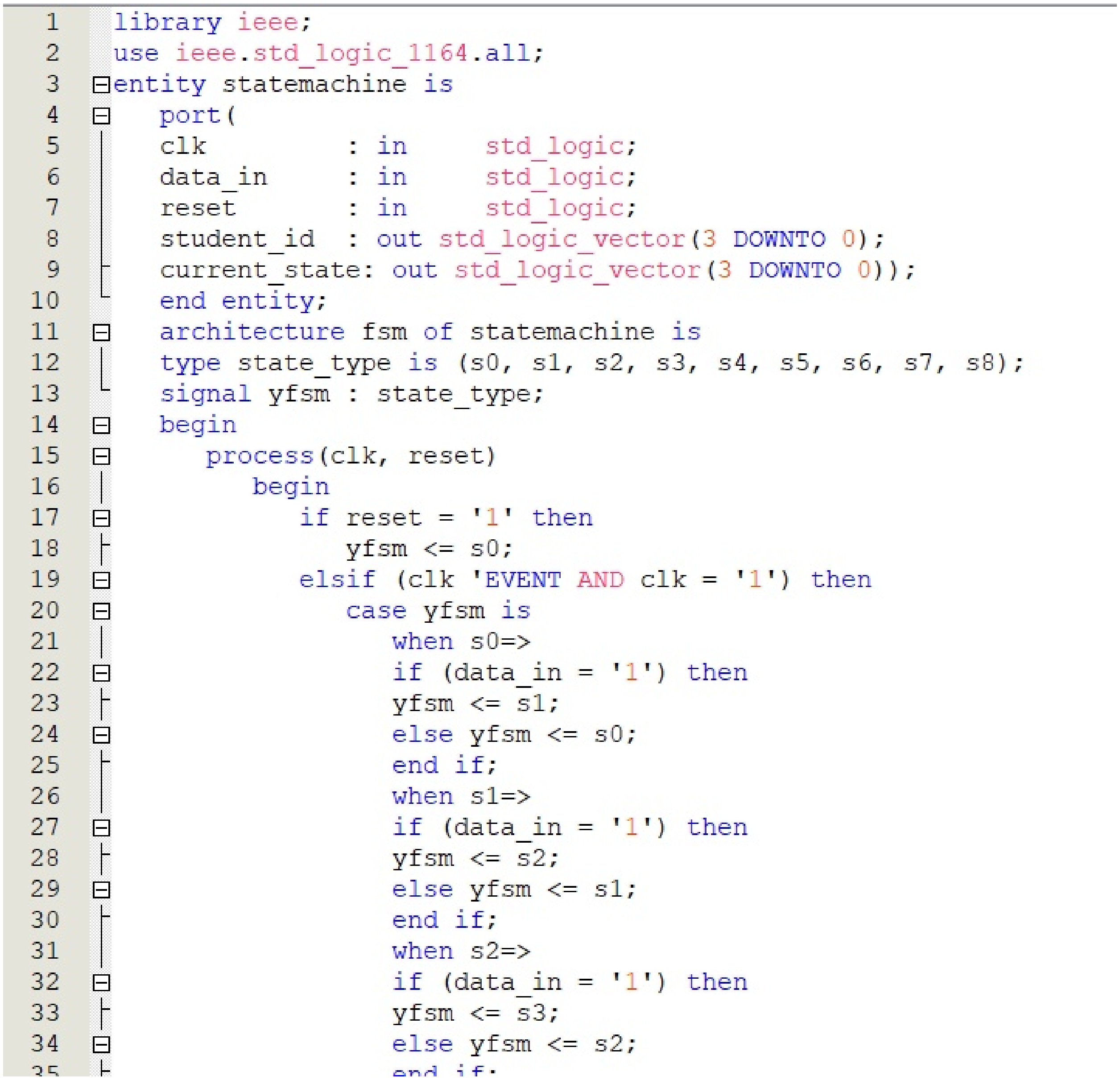
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
USE ieee.numeric_std.ALL;
ENTITY ALU IS
PORT(Clock : IN std_logic;
      A,B : IN unsigned(7 DOWNTO 0);
      student_id : IN unsigned(3 DOWNTO 0);
      OP : IN unsigned(15 DOWNTO 0);
      Neg : OUT std_logic;
      R1 : OUT unsigned(3 DOWNTO 0);
      R2 : OUT unsigned(3 DOWNTO 0));
END ALU;
ARCHITECTURE calculation OF ALU IS
SIGNAL Reg1, Reg2, Result : unsigned(7 DOWNTO 0):=(OTHERS => '0');
SIGNAL Reg4 : unsigned(0 TO 7);
BEGIN
Reg1 <= A;
Reg2 <= B;
PROCESS(Clock, OP)
BEGIN
   IF(rising_edge(Clock)) THEN
      CASE OP IS
       WHEN "0000000000000001" =>
         Result <= (Reg1 + Reg2);
       WHEN "0000000000000010" =>
         Result <= (Reg1 - Reg2);
       WHEN "0000000000000100" =>
         Result <= NOT Reg1;
       WHEN "0000000000001000" =>
         Result <= Reg1 NAND Reg2;
       WHEN "0000000000010000" =>
         Result <= Reg1 NOR Reg2;
       WHEN "0000000000100000" =>
         Result <= Reg1 AND Reg2;
```



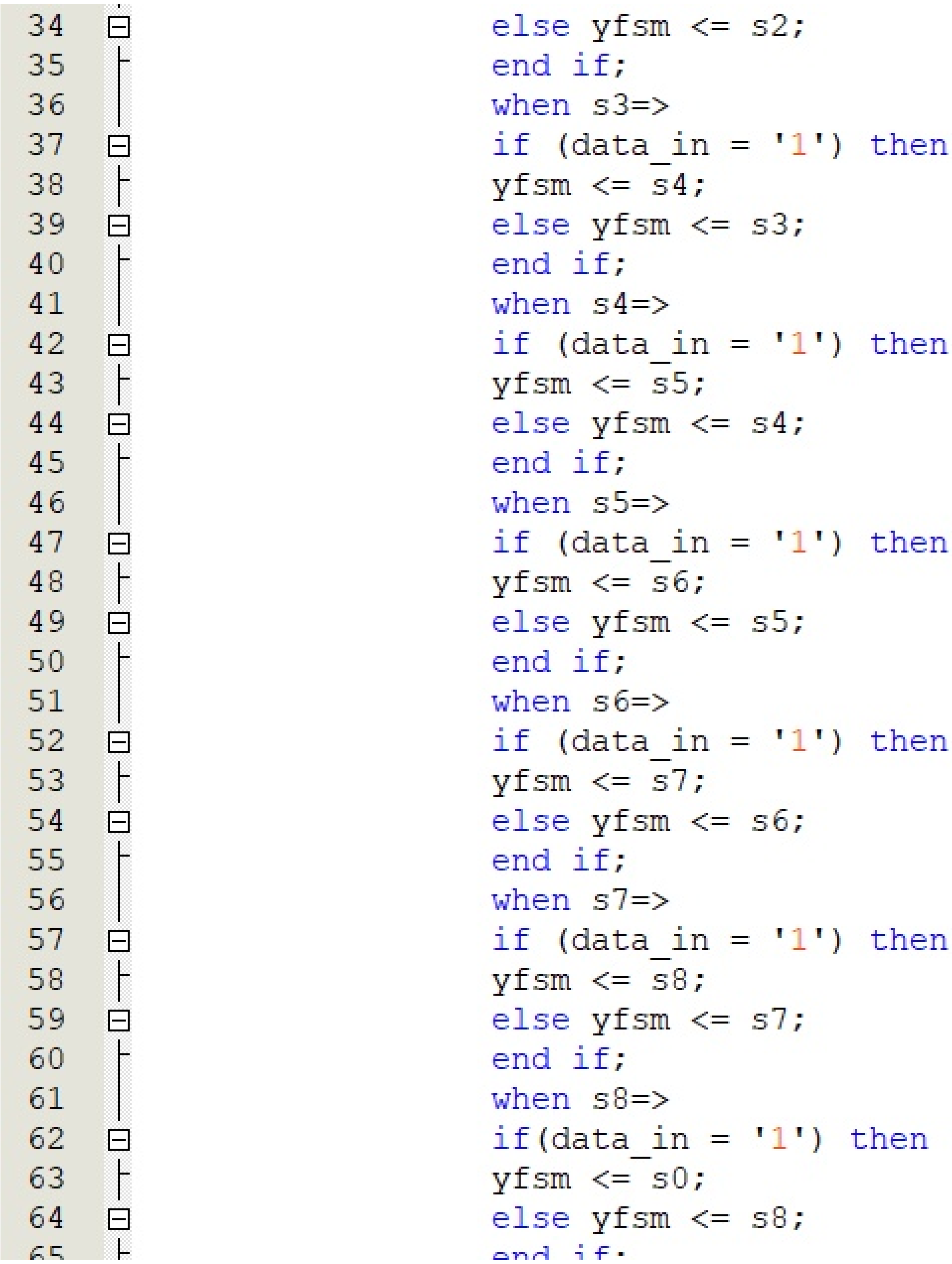
```
       WHEN "0000000000100000" =>
         Result <= Reg1 AND Reg2;
       WHEN "0000000001000000" =>
         Result <= Reg1 OR Reg2;
       WHEN "0000000010000000" =>
         Result <= Reg1 XOR Reg2;
       WHEN "0000000100000000" =>
         Result <= Reg1 XNOR Reg2;
       WHEN OTHERS =>
         Result <= "--------";
      END CASE;
   END IF;
END PROCESS;
R1 <= Result(3 DOWNTO 0);
R2 <= Result(7 DOWNTO 4);
END calculation;

```



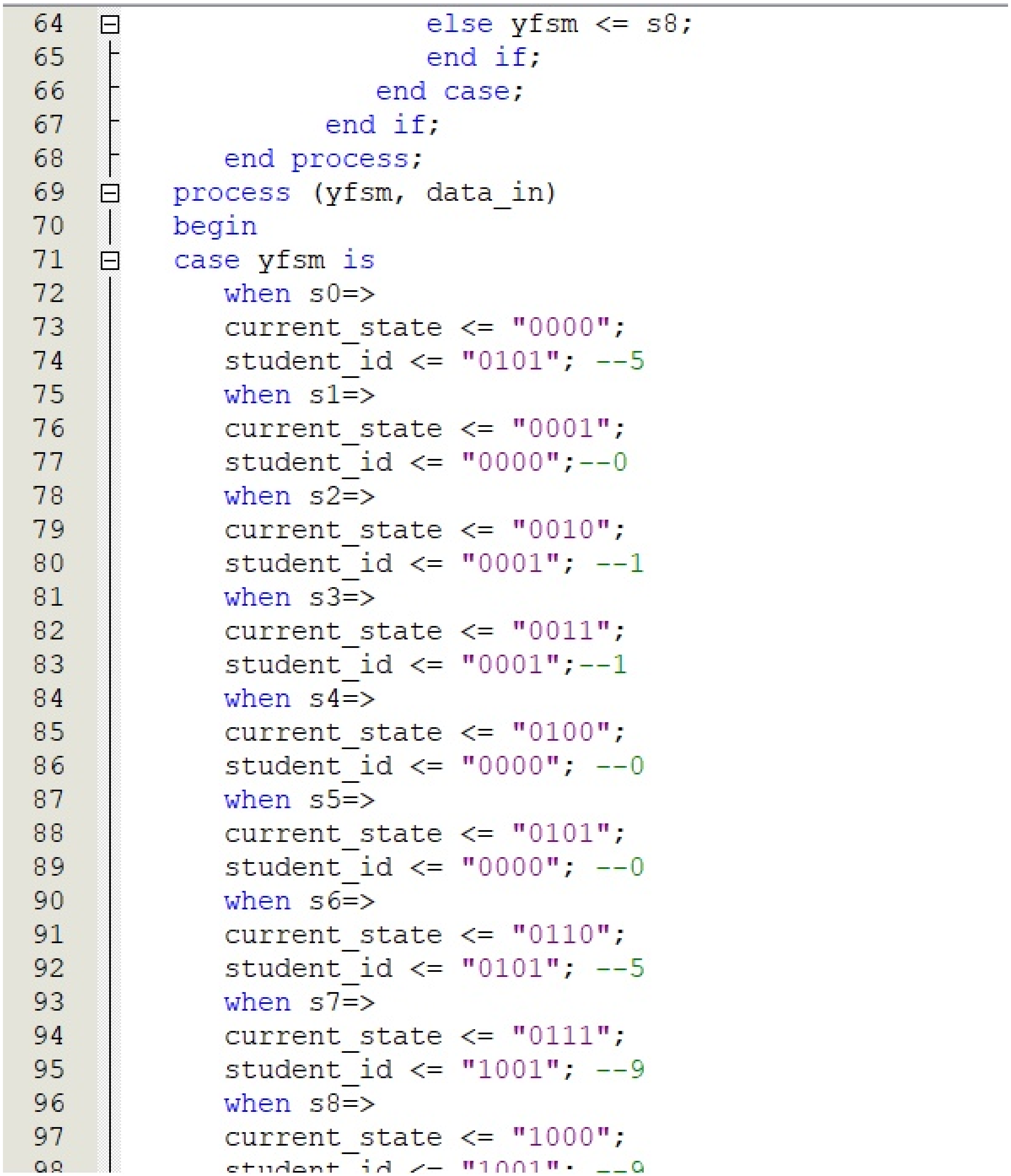
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY dec3to8 IS
PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
En : IN STD_LOGIC ;
y : OUT STD_LOGIC_VECTOR(15 DOWNTO 0) ) ;
END dec3to8 ;
ARCHITECTURE Behavior OF dec3to8 IS
SIGNAL Enw : STD_LOGIC_VECTOR(4 DOWNTO 0) ;
BEGIN
Enw <= En & w ;
WITH Enw SELECT
y <=    "0000000000000001" WHEN "10000",
        "0000000000000010" WHEN "10001",
        "0000000000000100" WHEN "10010",
        "0000000000001000" WHEN "10011",
        "0000000000010000" WHEN "10100",
        "0000000000100000" WHEN "10101",
        "0000000001000000" WHEN "10110",
        "0000000010000000" WHEN "10111",
        "0000000100000000" WHEN "11000",
        "0000001000000000" WHEN "11001",
        "0000010000000000" WHEN "11010",
        "0000100000000000" WHEN "11011",
        "0001000000000000" WHEN "11100",
        "0010000000000000" WHEN "11101",
        "0100000000000000" WHEN "11110",
        "1000000000000000" WHEN "11111",
        "0000000000000000" WHEN OTHERS ;
END Behavior ;
```



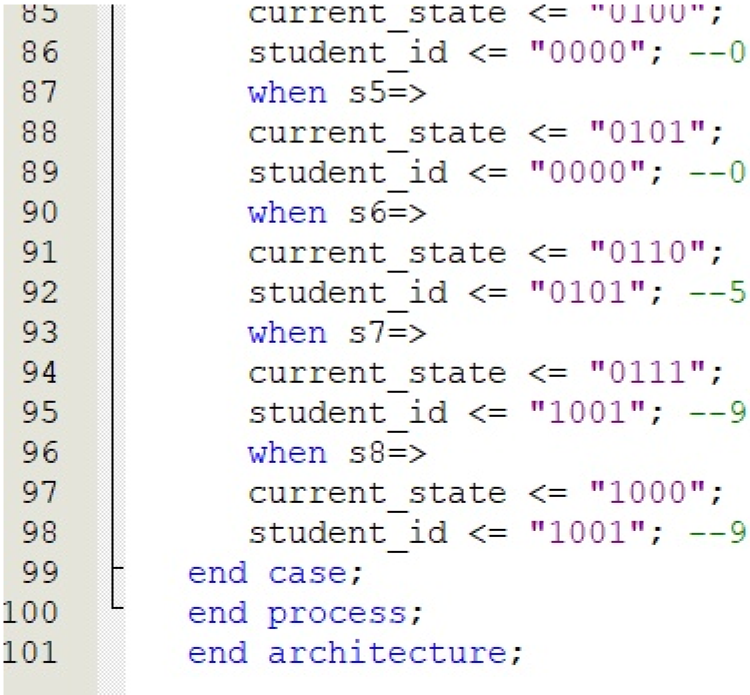
```
library ieee;
use ieee.std_logic_1164.all;
entity statemachine is
   port(
   clk         : in     std_logic;
   data_in     : in     std_logic;
   reset       : in     std_logic;
   student_id  : out std_logic_vector(3 DOWNTO 0);
   current_state: out std_logic_vector(3 DOWNTO 0));
   end entity;
   architecture fsm of statemachine is
   type state_type is (s0, s1, s2, s3, s4, s5, s6, s7, s8);
   signal yfsm : state_type;
   begin
      process(clk, reset)
         begin
            if reset = '1' then
               yfsm <= s0;
            elsif (clk 'EVENT AND clk = '1') then
               case yfsm is
                  when s0=>
                  if (data_in = '1') then
                  yfsm <= s1;
                  else yfsm <= s0;
                  end if;
                  when s1=>
                  if (data_in = '1') then
                  yfsm <= s2;
                  else yfsm <= s1;
                  end if;
                  when s2=>
                  if (data_in = '1') then
                  yfsm <= s3;
                  else yfsm <= s2;
                  end if;
```



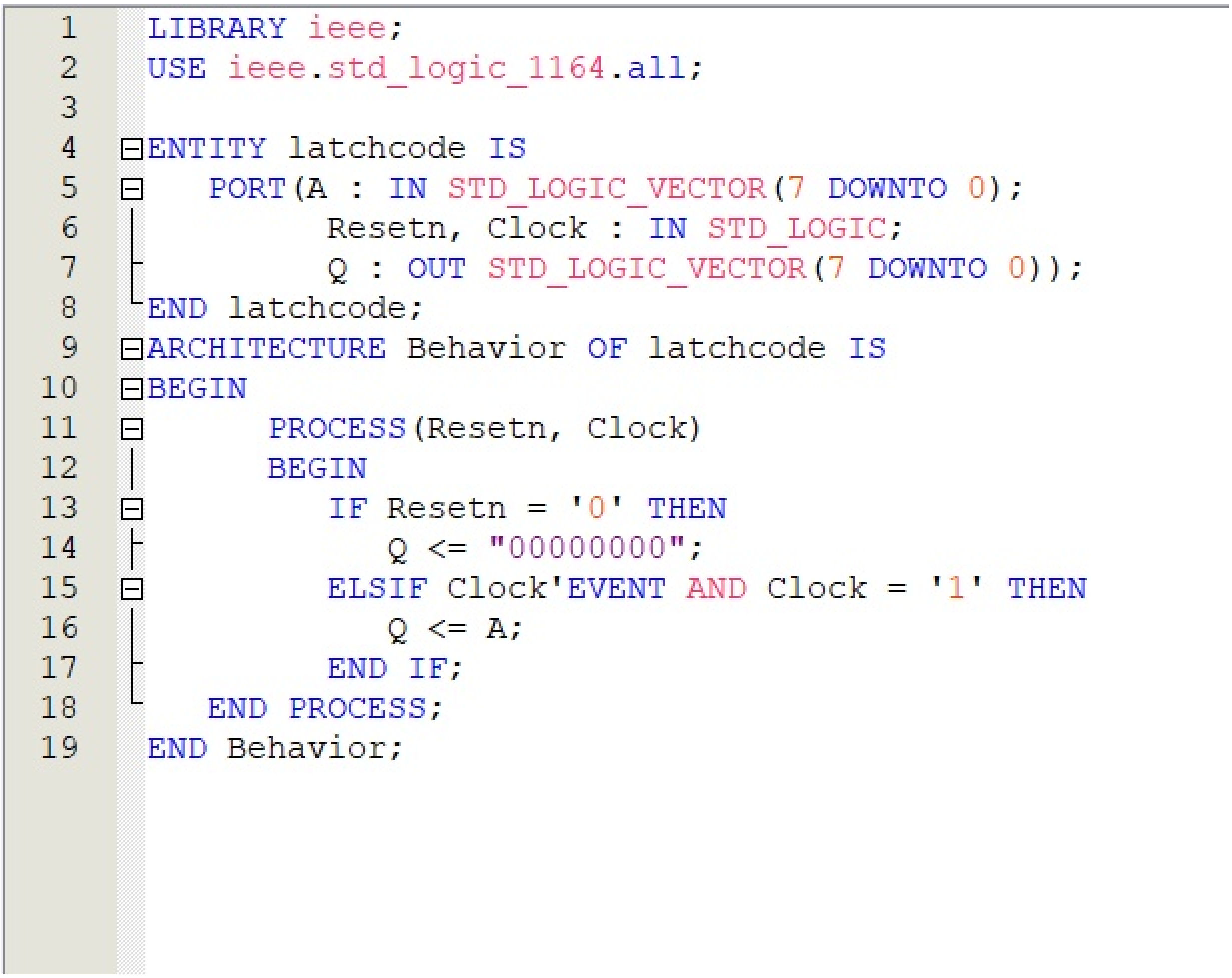
```
else yfsm <= s2;
end if;
when s3=>
if (data_in = '1') then
yfsm <= s4;
else yfsm <= s3;
end if;
when s4=>
if (data_in = '1') then
yfsm <= s5;
else yfsm <= s4;
end if;
when s5=>
if (data_in = '1') then
yfsm <= s6;
else yfsm <= s5;
end if;
when s6=>
if (data_in = '1') then
yfsm <= s7;
else yfsm <= s6;
end if;
when s7=>
if (data_in = '1') then
yfsm <= s8;
else yfsm <= s7;
end if;
when s8=>
if(data_in = '1') then
yfsm <= s0;
else yfsm <= s8;
end if;
```



```
               else yfsm <= s8;
               end if;
            end case;
         end if;
   end process;
process (yfsm, data_in)
begin
case yfsm is
   when s0=>
   current_state <= "0000";
   student_id <= "0101"; --5
   when s1=>
   current_state <= "0001";
   student_id <= "0000";--0
   when s2=>
   current_state <= "0010";
   student_id <= "0001"; --1
   when s3=>
   current_state <= "0011";
   student_id <= "0001";--1
   when s4=>
   current_state <= "0100";
   student_id <= "0000"; --0
   when s5=>
   current_state <= "0101";
   student_id <= "0000"; --0
   when s6=>
   current_state <= "0110";
   student_id <= "0101"; --5
   when s7=>
   current_state <= "0111";
   student_id <= "1001"; --9
   when s8=>
   current_state <= "1000";
   student_id <= "1001"; --9
```



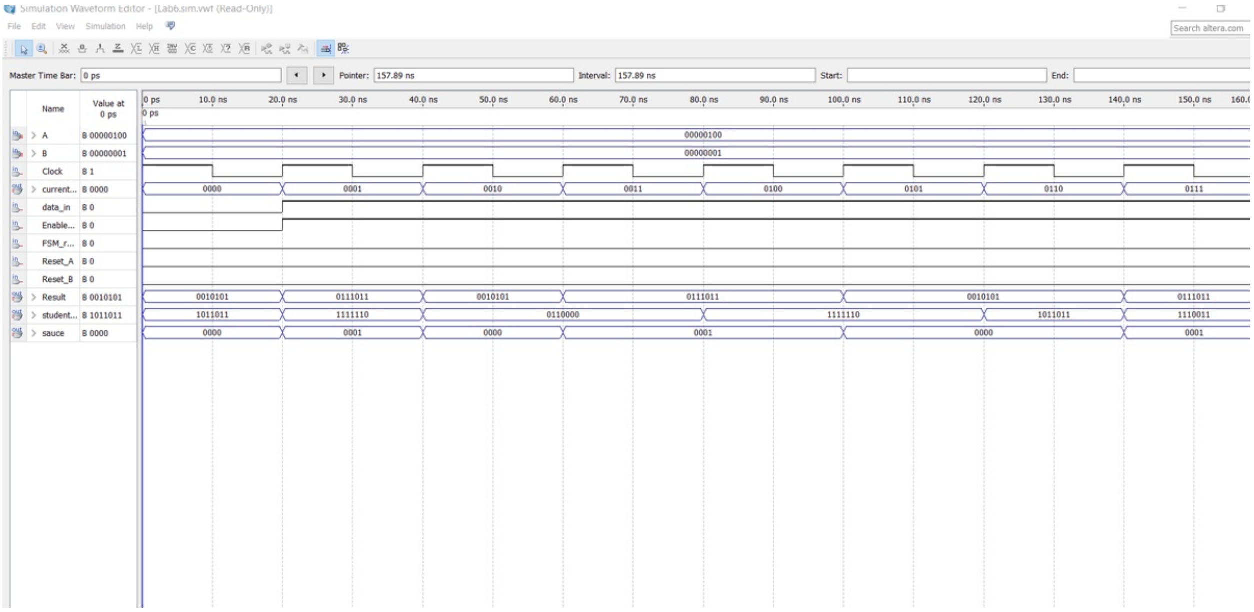
```vhdl
   current_state <= "0100";
   student_id <= "0000"; --0
   when s5=>
   current_state <= "0101";
   student_id <= "0000"; --0
   when s6=>
   current_state <= "0110";
   student_id <= "0101"; --5
   when s7=>
   current_state <= "0111";
   student_id <= "1001"; --9
   when s8=>
   current_state <= "1000";
   student_id <= "1001"; --9
end case;
end process;
end architecture;
```



```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY latchcode IS
   PORT(A : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
         Resetn, Clock : IN STD_LOGIC;
         Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END latchcode;
ARCHITECTURE Behavior OF latchcode IS
BEGIN
      PROCESS(Resetn, Clock)
      BEGIN
         IF Resetn = '0' THEN
            Q <= "00000000";
         ELSIF Clock'EVENT AND Clock = '1' THEN
            Q <= A;
         END IF;
   END PROCESS;
END Behavior;
```

## WAVEFORM

The Successful Compilation of the waveform in the following screenshot:



Simulation Waveform Editor - [Lab6.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

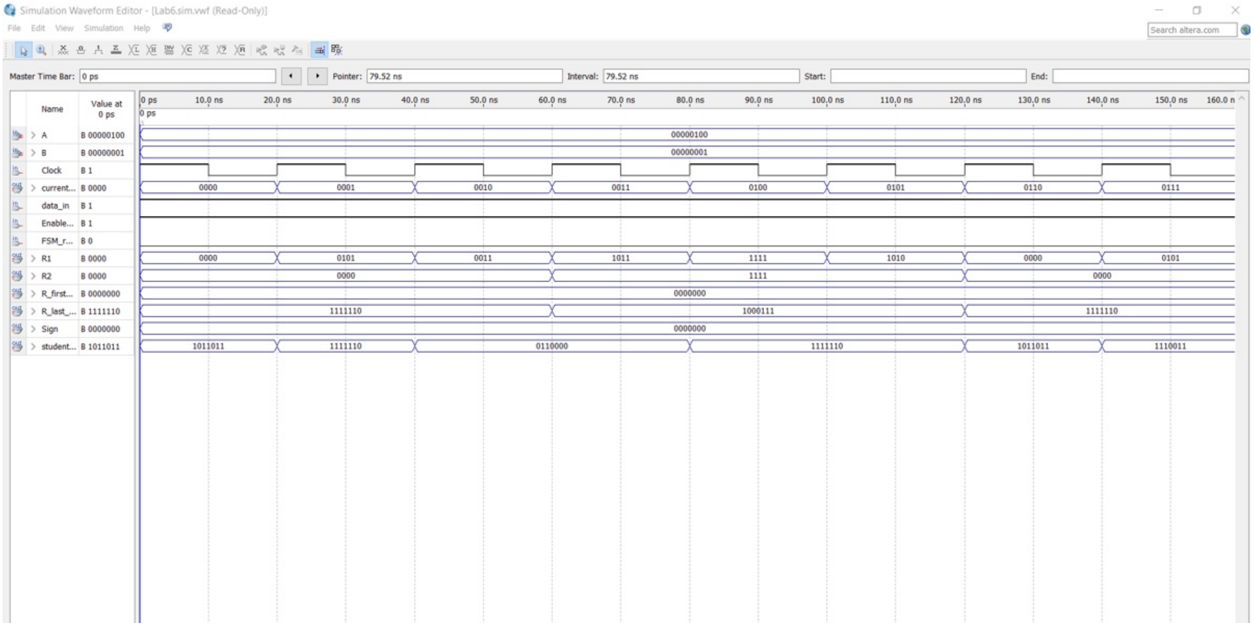
Pointer: 157.89 ns

Interval: 157.89 ns

Start:

End:

| Name | Value at 0 ps |
|---|---|
| A | B 00000100 |
| B | B 00000001 |
| Clock | B 1 |
| current... | B 0000 |
| data_in | B 0 |
| Enable... | B 0 |
| FSM_r... | B 0 |
| Reset_A | B 0 |
| Reset_B | B 0 |
| Result | B 0010101 |
| student... | B 1011011 |
| sauce | B 0000 |



Simulation Waveform Editor - [Lab6.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 79.52 ns

Interval: 79.52 ns

Start:

End:

| Name | Value at 0 ps |
|---|---|
| A | B 00000100 |
| B | B 00000001 |
| Clock | B 1 |
| current... | B 0000 |
| data_in | B 1 |
| Enable... | B 1 |
| FSM_r... | B 0 |
| R1 | B 0000 |
| R2 | B 0000 |
| R_first... | B 0000000 |
| R_last_... | B 1111110 |
| Sign | B 0000000 |
| student... | B 1011011 |

## Conclusion

Conclusively, this lab was a success in regards to the main objectives. The design and construction of an ALU in a VHDL environment and the implementation of it. Specifically, the design and build for all functions of the ALU were successful, and the design+simulation of the ALU was successful using VHDL using Quartus Simulator.