

Name: Richard Raymond J. Canda

C.Y.S.: BSCpE – 3A

The CODE:

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity decod is

Port (A : in STD_LOGIC_VECTOR(2 downto 0); -- 3-bit input

Y : out STD_LOGIC_VECTOR(7 downto 0) -- 8-bit output

);

end decod;

architecture Behavioral of decod is

begin

process(A)

begin

-- Set all outputs to '0'

Y <= "00000000";

-- Activate the output based on the input

case A is

when "000" => Y <= "00000001";

when "001" => Y <= "00000010";

when "010" => Y <= "00000100";

when "011" => Y <= "00001000";

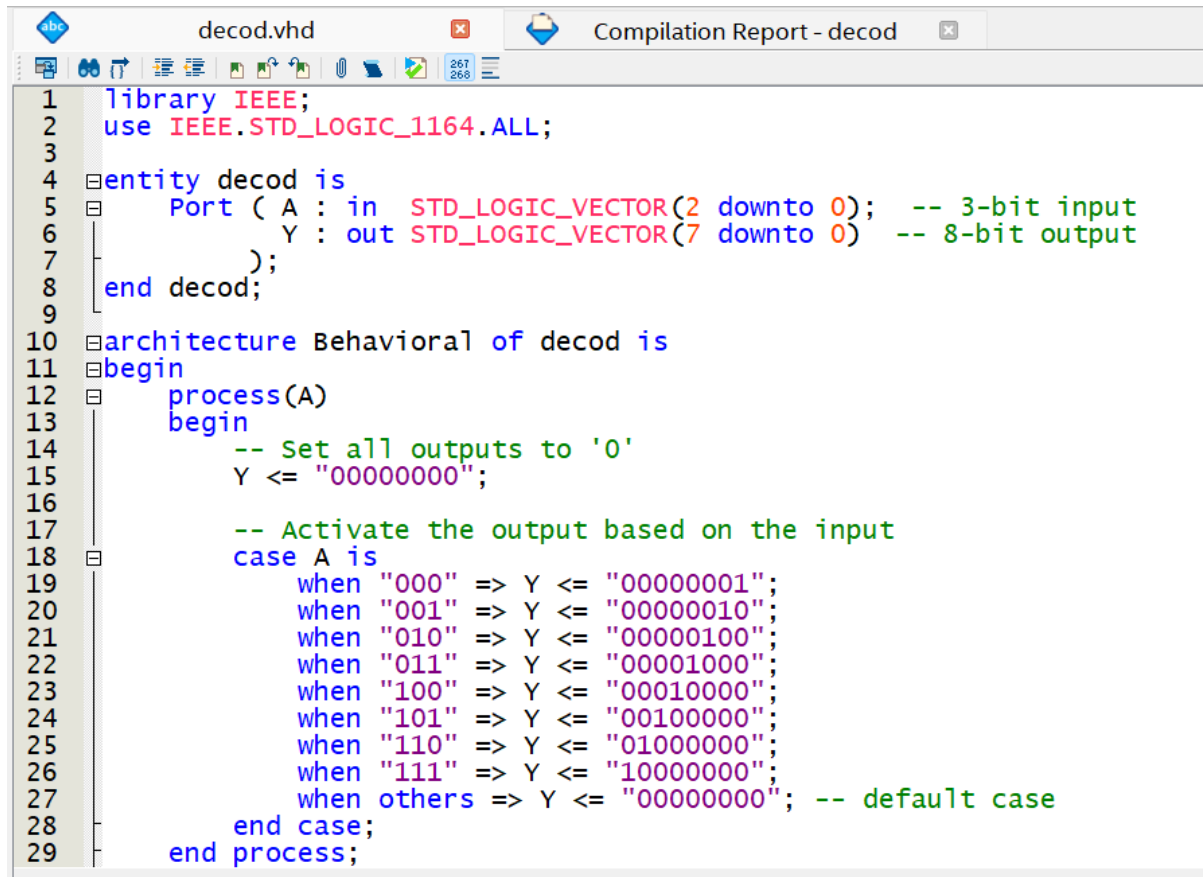
when "100" => Y <= "00010000";

when "101" => Y <= "00100000";

when "110" => Y <= "01000000";

when "111" => Y <= "10000000";

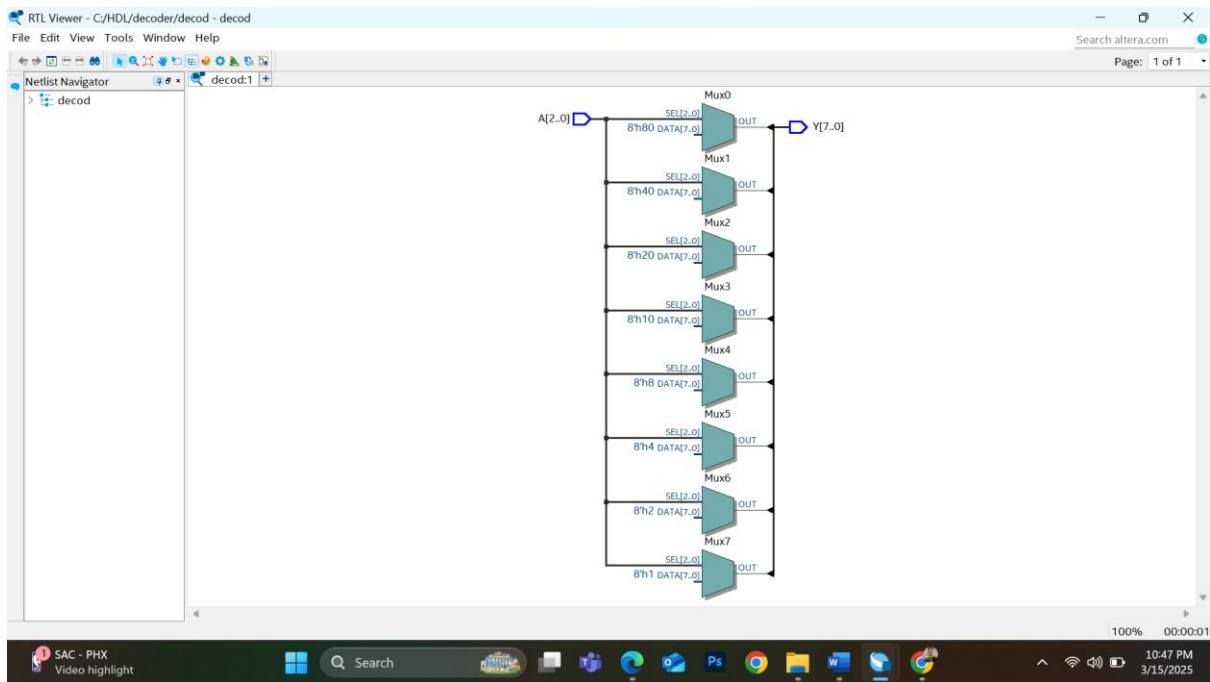
```
        when others => Y <= "00000000"; -- default case
    end case;
end process;
end Behavioral;
```



The screenshot shows a VHDL code editor with a file named 'decod.vhd'. The code implements a 3-to-8 decoder. It starts with a library declaration for IEEE and uses the STD_LOGIC_1164 package. The entity 'decod' has a 3-bit input 'A' and an 8-bit output 'Y'. The architecture 'Behavioral' contains a process for 'A' that initializes 'Y' to '0' and then uses a case statement to set 'Y' based on the value of 'A'. The case statement includes eight 'when' clauses for inputs '000' through '111', each setting 'Y' to a specific 8-bit output vector. A final 'when others' clause sets 'Y' to '00000000' as a default case. The code is numbered from 1 to 29.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity decod is
5  Port ( A : in  STD_LOGIC_VECTOR(2 downto 0); -- 3-bit input
6        Y : out STD_LOGIC_VECTOR(7 downto 0) -- 8-bit output
7        );
8  end decod;
9
10 architecture Behavioral of decod is
11 begin
12     process(A)
13     begin
14         -- Set all outputs to '0'
15         Y <= "00000000";
16
17         -- Activate the output based on the input
18         case A is
19             when "000" => Y <= "00000001";
20             when "001" => Y <= "00000010";
21             when "010" => Y <= "00000100";
22             when "011" => Y <= "00001000";
23             when "100" => Y <= "00010000";
24             when "101" => Y <= "00100000";
25             when "110" => Y <= "01000000";
26             when "111" => Y <= "10000000";
27             when others => Y <= "00000000"; -- default case
28         end case;
29     end process;
```

The LOGIC GATE:



The WAVEFORM:

