EE 2000 Logic Circuit Design
Semester B 2023/24

Tutorial 4

1. What are the mistakes for this VHDL?

```
Library ieEe;
use IEEE.std-logic_1164.all;

ENTITY and_gate IS
    port (a&b : in STD_LOGIC;
           S: out STD_LOGIC;);
end;

architecture CKT of anD_Gate IS
begin
          s <= a AND b;
end ckt;
```

```
library ieee;
use ieee.std_logic_1164.all;

ENTITY and_gate IS
  port (a, b : in STD_LOGIC;
        s : out STD_LOGIC);
END;

ARCHITECTURE CKT of and_gate IS
BEGIN
  s <= a and b;
END CKT;
```

```
library ieee;
use IEEE.std_logic_1164.all;

ENTITY mozone is
    port (A1 : in std_logic_vector(7 downto 0);
          A2 : in bit_vector(0 to 4);
       O1 : out std_logic);
End mozone;
```

ry and entity declarations for a logic design entity named
inputs and outputs.
ogic data with the highest index number holding the most

the lowest index number holding the most significant bit
ut

```
ENTITY Prob_3 is
    port (A, B, C : in bit;
          x1, x2, x3 : out bit);
End Prob_3;

architecture behavior of Prob_3 is
begin
      x1 <= (NOT A AND NOT B AND C) OR (A AND NOT (B AND C));
      x2 <= (NOT A AND B OR NOT C) AND NOT (B AND NOT C OR A);
      x3 <= NOT ((A AND NOT (B AND C)) OR (NOT A AND NOT C));
end behavior;
```

with entity and architecture) to implement a
ions. Use concurrent statements and without

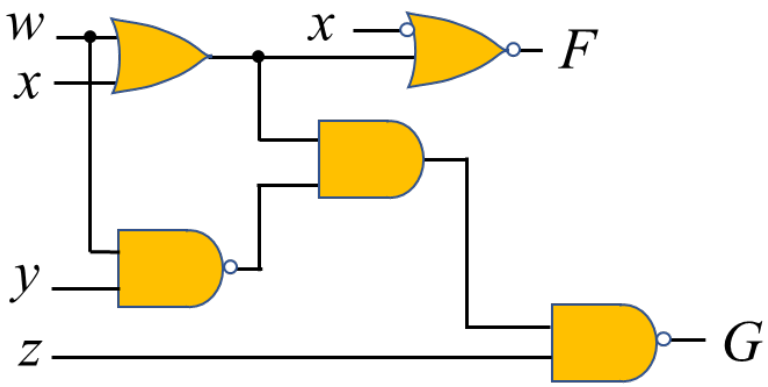
```
library ieee;
use ieee.std_logic_1164.all;

entity myCircuit IS
  port (X, Y, Z : in STD_LOGIC;
        A, B, C : out STD_LOGIC);
END entity;

architecture Behavioral of myCircuit IS
  signal sigW1 : STD_LOGIC;
BEGIN
  -- Define the common logic term
  sigW1 <= XY'Z';

  -- Implement expression A
  A <= (X OR XZ) + (YZ);

  -- Implement expression B
  B <= (sigW1) * (X + Z);

  -- Implement expression C
  C <= YZ';
END architecture;
```

- x3 = (A(BC)' + A'C')

4. Write a complete VHDL design module to implement a circuit with the following Boolean expressions. Assign a signal name sigW1 to represent the common logic term in your design. Use concurrent statements without NAND and NOR operators in your design.

- A = (XY'Z')' + XZ
- B = (XY'Z')'(X + Z)
- C = ((XY'Z')'+X')'

5. Write a complete VHDL design module to implement the combinational circuit shown. Assign signals for intermediate outputs. Use concurrent statements (i) without NAND and NOR operators in your design; and (ii) with NAND and NOR operators.




```
library ieee;
use ieee.std_logic_1164.all;

entity myCircuit IS
  port (W, X, Y : in STD_LOGIC;
        A, B, C, F, G : out STD_LOGIC);
END entity;

architecture Behavioral of myCircuit IS
BEGIN
  -- OR gate for A
  A <= W OR X;

  -- AND gate for B
  B <= W AND Y;

  -- AND gate for C
  C <= A AND B;

  -- NOR gate for F
  F <= NOR(X, A);

  -- NAND gate for G
  G <= NAND(C, Z); -- Assuming 'Z' is an actual input
END architecture;
```