VHDL CODE TEMPLATE FOR 7 LOGIC GATES

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity LogicGate is
    Port (
        A : in STD_LOGIC;
        B : in STD_LOGIC;  (*remove this line only for INVERTER GATE)
        Y : out STD_LOGIC
    );
end LogicGate;

architecture Behavioral of LogicGate is
begin (*anyone from these green lines based on q)
    Y <= A AND B;   -- AND gate
    Y <= NOT (A AND B); -- NAND gate
    Y <= A OR B;  -- OR gate
    Y <= NOT (A OR B); -- NOR gate
    Y <= A XOR B;   -- XOR gate
    Y <= NOT (A XOR B); -- XNOR gate
    Y <= NOT A; -- INVERTER gate
end Behavioral;
```

TESTBENCH CODE TEMPLATE FOR 7 LOGIC GATES

```
library IEEE;
use IEEE.std_logic_1164.ALL;

entity gatebench is
end gatebench;

architecture tb of gatebench is

component LogicGate is
    port (
        a: in std_logic;
        b: in std_logic; (*remove this line only for single-input gates like NOT / INVERTER)
        q: out std_logic
    );
end component;

signal a_in, b_in, q_out : std_logic;

begin
DUT: LogicGate port map(a_in, b_in, q_out);

process
begin
    -- Test case 1: a = 0, b = 0
    a_in <= '0';
    b_in <= '0';
    wait for 100 ns;

    -- Test case 2: a = 0, b = 1
    a_in <= '0';
    b_in <= '1';
    wait for 100 ns;

    -- Test case 3: a = 1, b = 0
    a_in <= '1';
    b_in <= '0';
    wait for 100 ns;

    -- Test case 4: a = 1, b = 1
    a_in <= '1';
    b_in <= '1';
    wait for 100 ns;

    wait;
end process;

end tb;
```

GCD VHDL Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity gcd_4bit is
    Port (
        a : in  STD_LOGIC_VECTOR(3 downto 0);
        b : in  STD_LOGIC_VECTOR(3 downto 0);
        gcd : out STD_LOGIC_VECTOR(3 downto 0)
    );
end gcd_4bit;

architecture Behavioral of gcd_4bit is
begin
    process(a, b)
        variable x, y : STD_LOGIC_VECTOR(3 downto 0);
    begin
        x := a;
        y := b;

        while (x /= y) loop
            if (x < y) then
                y := y - x;
            else
                x := x - y;
            end if;
        end loop;

        gcd <= x;
    end process;
end Behavioral;
```

GCD TestBench Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE. STD_LOGIC_UNSIGNED.ALL;

entity gcd_tb is
end gcd_tb;

architecture tb of gcd_tb is

   component gcd_4bit
      port (
         a : in  STD_LOGIC_VECTOR(3 downto 0);
         b : in  STD_LOGIC_VECTOR(3 downto 0);
         gcd  : out STD_LOGIC_VECTOR(3 downto 0)
      );
   end component;

   signal a : STD_LOGIC_VECTOR(3 downto 0);
   signal b : STD_LOGIC_VECTOR(3 downto 0);
   signal gcd_out  : STD_LOGIC_VECTOR(3 downto 0);

begin

   uut : gcd_4bit
      port map (
         a   => a,
         b   => b,
         gcd => gcd_out
      );

   process
      constant period : time := 100 ns;
   begin
      a <= "1100";
      b <= "1000";
      wait for period;

      a <= "1111";
      b <= "1111";
      wait for period;

      wait;
   end process;

end tb;
```

Largest of the four VHDL Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity max_of_four is
   port (
      a, b, c, d : in  STD_LOGIC_VECTOR(3 downto 0);
      max : out STD_LOGIC_VECTOR(3 downto 0)
   );
end max_of_four;

architecture Behavioral of max_of_four is
   signal temp : STD_LOGIC_VECTOR(3 downto 0);

begin

   process(a, b, c, d)
   begin
      temp := a;
      if b > temp then
         temp := b;
      end if;
      if c > temp then
         temp := c;
      end if;
      if d > temp then
         temp := d;
      end if;
      max <= temp;
   end process;

end Behavioral;
```

Largest of the four Testbench Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity max_tb is
end max_tb;

architecture tb of max_tb is
   component max_of_four
      port (
         a, b, c, d : in  STD_LOGIC_VECTOR(3 downto 0);
         max  : out STD_LOGIC_VECTOR(3 downto 0)
      );
   end component;

   signal a, b, c, d : STD_LOGIC_VECTOR(3 downto 0);
   signal max_out : STD_LOGIC_VECTOR(3 downto 0);

begin

   uut: max_of_four
      port map (
         a   => a,
         b   => b,
         c   => c,
         d   => d,
         max => max_out
      );

   process
      constant period : time := 100 ns;
   begin
      a <= "0100"; b <= "0011"; c <= "1000"; d <= "0110";
      wait for period;

      a <= "0001"; b <= "0000"; c <= "0010"; d <= "0011";
      wait for period;

      wait;
   end process;

end tb;
```