

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb_Alarme_Automotivo is
end tb_Alarme_Automotivo;

architecture testbench of tb_Alarme_Automotivo is
    component Alarme_Automotivo
        Port (
            Porta    : in  STD_LOGIC;
            Ignicao    : in  STD_LOGIC;
            Farol     : in  STD_LOGIC;
            Alarme    : out STD_LOGIC
        );
    end component;

    signal Porta    : STD_LOGIC := '0';
    signal Ignicao    : STD_LOGIC := '0';
    signal Farol     : STD_LOGIC := '0';
    signal Alarme    : STD_LOGIC;

begin

    UUT: Alarme_Automotivo
        port map (
            Porta => Porta,
            Ignicao => Ignicao,
            Farol => Farol,
            Alarme => Alarme);

    stim_proc: process
    begin
        -- Teste 1: Farol apagado, Ignição desligada, Porta fechada -> Alarme = 0
        Farol <= '0'; Ignicao <= '0'; Porta <= '0';
        wait for 10 ns;

        -- Teste 2: Farol apagado, Ignição desligada, Porta aberta -> Alarme = 0
        Farol <= '0'; Ignicao <= '0'; Porta <= '1';
        wait for 10 ns;

        -- Teste 3: Farol apagado, Ignição ligada, Porta fechada -> Alarme = 0
        Farol <= '0'; Ignicao <= '1'; Porta <= '0';
        wait for 10 ns;

        -- Teste 4: Farol apagado, Ignição ligada, Porta aberta -> Alarme = 1
        Farol <= '0'; Ignicao <= '1'; Porta <= '1';
        wait for 10 ns;

        -- Teste 5: Farol ligado, Ignição desligada, Porta fechada -> Alarme = 1
        Farol <= '1'; Ignicao <= '0'; Porta <= '0';
        wait for 10 ns;
    end process;
end

```

```
-- Teste 6: Farol ligado, Ignição desligada, Porta aberta -> Alarme = 1
    Farol <= '1'; Ignicao <= '0'; Porta <='1';
    wait for 10 ns;

-- Teste 7: Farol ligado, Ignição ligada, Porta fechada -> Alarme = 0
    Farol <= '1'; Ignicao <= '1'; Porta <='0';
    wait for 10 ns;

-- Teste 8: Farol ligado, Ignição ligada, Porta aberta -> Alarme = 1
    Farol <= '1'; Ignicao <= '1'; Porta <='1';
    wait for 10 ns;

    wait;
end process;

end testbench;
```