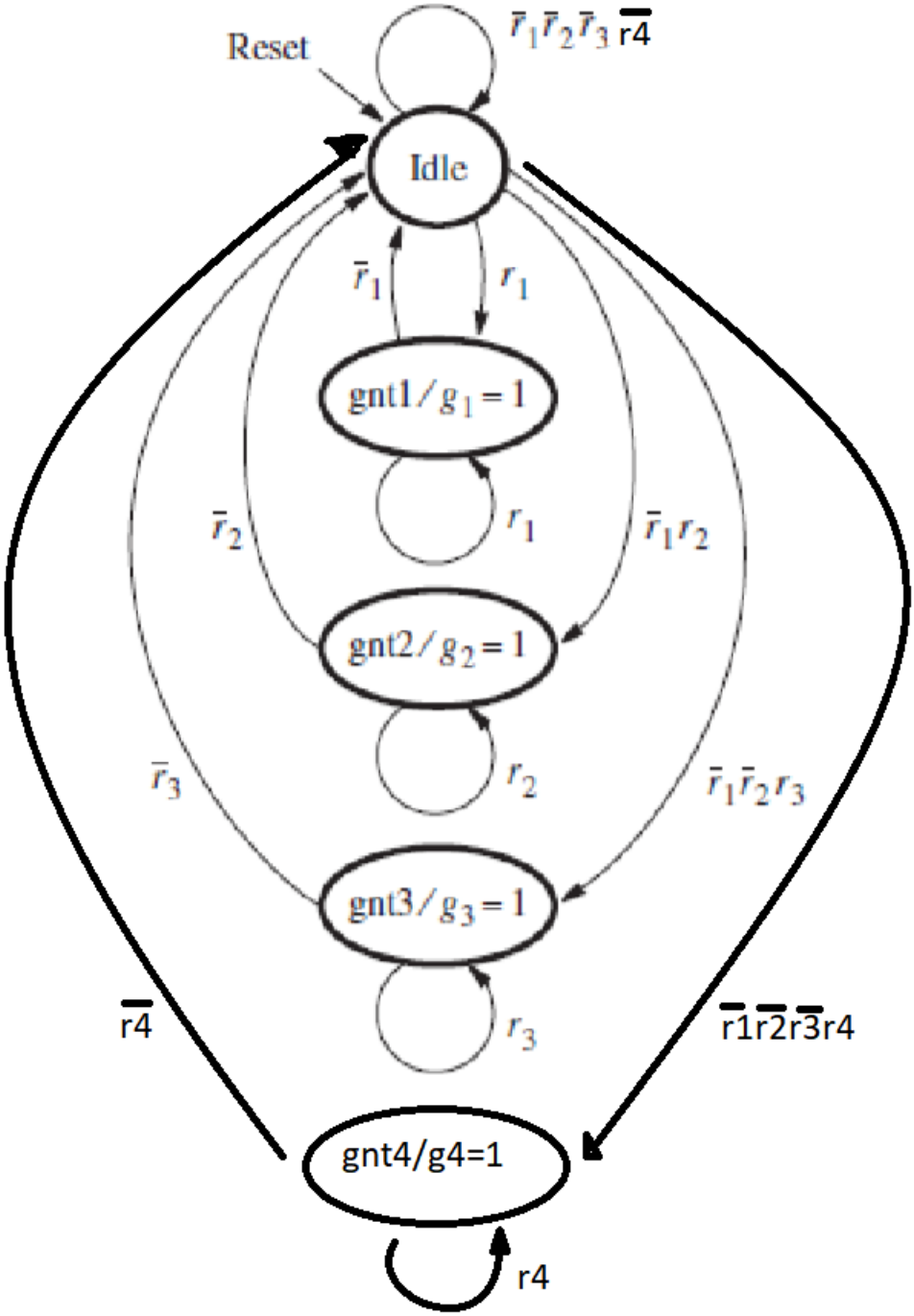
BRIAN SEBASTIAN CACERES PINZON 1803245

MAQUINA DE ESTADOS



IMPLEMENTACION

```
1
2   library IEEE;
3   use IEEE.STD_LOGIC_1164.ALL;
4
5
6   entity FsmTaller1 is
7       Port ( r : in  STD_LOGIC_VECTOR (1 to 4);
8              g : out  STD_LOGIC_VECTOR (1 to 4);
9              Clock : in  STD_LOGIC;
10             Resetn : in  STD_LOGIC);
11  end FsmTaller1;
12
13  architecture Behavioral of FsmTaller1 is
14     TYPE State_type IS (Idle, gnt1, gnt2, gnt3, gnt4);
15     SIGNAL y : State_type;
16  begin
17     PROCESS(Resetn, Clock )
18     BEGIN
19        IF Resetn = '1' THEN y <= Idle;
20        ELSIF(Clock'EVENT AND Clock = '1')THEN
21        CASE y IS
22           WHEN Idle=>
23              IF r(1)='1' THEN y <= gnt1;
24              ELSIF r(2)='1' THEN y<= gnt2;
25              ELSIF r(3)='1' THEN y<= gnt3;
26              ELSIF r(4)='1' THEN y<= gnt4;
27              ELSE y <= Idle;
28              END IF;
29           WHEN gnt1=>
30              IF r(1)='1' THEN y <= gnt1;
31              ELSE y <= Idle;
32              END IF;
33           WHEN gnt2=>
34              IF r(2)='1' THEN y<= gnt2;
35             ELSE y <= Idle;
36             END IF;
37           WHEN gnt3=>
38              IF r(3)='1' THEN y<= gnt3;
39              ELSE y <= Idle;
40              END IF;
41           WHEN gnt4=>
42              IF r(4)='1' THEN y<= gnt4;
43              ELSE y <= Idle;
44              END IF;
45        END CASE;        .
46     END IF;
47  END PROCESS;
48  g(1) <='1' WHEN y = gnt1 ELSE '0';
49  g(2) <='1' WHEN y = gnt2 ELSE '0';
50  g(3) <='1' WHEN y = gnt3 ELSE '0';
51  g(4) <='1' WHEN y = gnt4 ELSE '0';
52
53  PROCESS(y)
54  BEGIN
55     g(1)<='0';
56     g(2)<='0';
57     g(3)<='0';
58     g(4)<='0';
59     IF y = gnt1 THEN g(1) <='1';
60     ELSIF y = gnt2 THEN g(2) <='1';
61     ELSIF y = gnt3 THEN g(3) <='1';
62     ELSIF y = gnt4 THEN g(4) <='1';
63     END IF;
64     END PROCESS;
65  END Behavioral;
```

SIMULACION

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY simulacion IS
END simulacion;
ARCHITECTURE behavior OF simulacion IS
    COMPONENT FsmTaller1
    PORT(
         r : IN  std_logic_vector(1 to 4);
         g : OUT  std_logic_vector(1 to 4);
         Clock : IN  std_logic;
         Resetn : IN  std_logic
        );
    END COMPONENT;
   --Inputs
   signal r : std_logic_vector(1 to 4) := (others => '0');
   signal Clock : std_logic := '0';
   signal Resetn : std_logic := '0';
   --Outputs
   signal g : std_logic_vector(1 to 4);
   -- Clock period definitions
   constant Clock_period : time := 10 ns;
BEGIN
   -- Instantiate the Unit Under Test (UUT)
   uut: FsmTaller1 PORT MAP (
          r => r,
          g => g,
          Clock => Clock,
          Resetn => Resetn
        );
   -- Clock process definitions
   Clock_process :process
   begin
      Clock <= '0';
      wait for 50 ns;
      Clock <= '1';
      wait for 50 ns;
   end process;
   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;
      wait for 10 ns ;
      -- insert stimulus here
r(1) <= '0' ; r(2) <= '0' ; r(3) <='0' ; r(4) <='0'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '0' ; r(3) <='0' ; r(4) <='1'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '0' ; r(3) <='1' ; r(4) <='0'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '0' ; r(3) <='1' ; r(4) <='1'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '1' ; r(3) <='0' ; r(4) <='0'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '1' ; r(3) <='0' ; r(4) <='1'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '1' ; r(3) <='1' ; r(4) <='0'; wait for 10 ns ;
r(1) <= '0' ; r(2) <= '1' ; r(3) <='1' ; r(4) <='1'; wait for 10 ns ;
r(1) <= '1' ; r(2) <= '0' ; r(3) <='0' ; r(4) <='0'; wait for 10 ns ;
wait for 10 ns ;
end process;
END;
```

DIAGRAMA DE TIEMPOS