

هدف آزمایش: ایجاد یک Sequence Detector با زبان VHDL

شرح آزمایش: می‌خواهیم با استفاده از یک ماشین Moore یک Sequence Detector برای دنباله زیر طراحی کنیم :

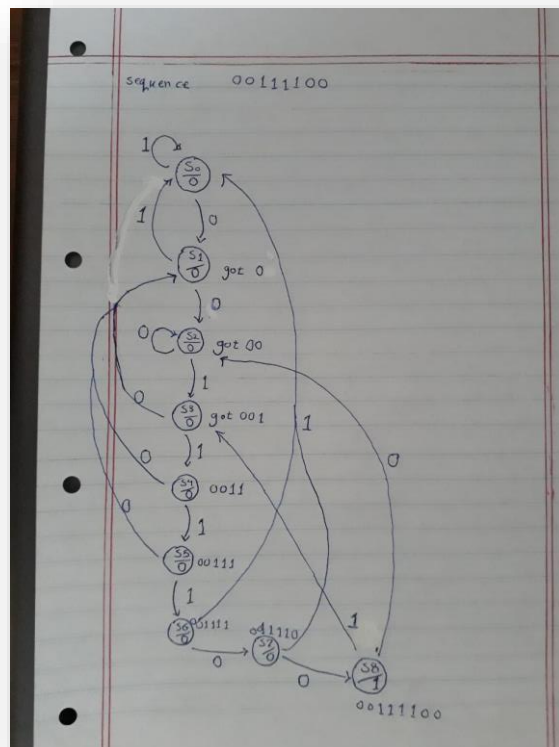
00111100

نکاتی که بایستی در طراحی لحاظ کنیم:

- ماشین Moore باشد
- ریست آسنکرون باشد
- همپوشانی داشته باشد

روش انجام آزمایش:

ابتدا بر روی کاغذ State Diagram ماشین مورد نظر را طراحی میکنیم:



در طراحی بالا به همپوشانی توجه شده است.

سپس با استفاده از دیاگرام طراحی شده اقدام به نوشتن کد VHDL میکنیم:

ابتدا State ها را تعریف میکنیم:

```
architecture Behavioral of Sequence_Detector is
    type states is (S0, S1, S2, S3, S4, S5, S6, S7, S8);
    signal ps_state, nxt_state : states;
```

سپس ریست آسنکرون را تعریف میکنیم:

```
process(clk,rst)
begin
    if(rst='0') then
        ps_state <= S0;
    elsif(rising_edge(clk)) then
        ps_state <= nxt_state;
    end if;
end process;
```

در مرحله بعد با استفاده از State Diagram طراحی شده، Case ها مختلف را پیاده سازی میکنیم:

```
process(ps_state ,input)
begin
  case(ps_state) is
    when S0 =>
      if(input='1') then
        nxt_state <= S0;
      else
        nxt_state <= S1;
      end if;
    when S1 =>
      if(input='1') then
        nxt_state <= S0;
      else
        nxt_state <= S2;
      end if;
    when S2 =>
      if(input='1') then
        nxt_state <= S3;
      else
        nxt_state <= S2;
      end if;
    when S3 =>
      if(input='1') then
        nxt_state <= S4;
      else
        nxt_state <= S1;
      end if;
    when S4 =>
      if(input='1') then
        nxt_state <= S5;
      else
        nxt_state <= S1;
      end if;
    when S5 =>
      if(input='1') then
        nxt_state <= S6;
      else
        nxt_state <= S1;
      end if;
    when S6 =>
      if(input='1') then
        nxt_state <= S0;
      else
        nxt_state <= S7;
      end if;
    when S7 =>
      if(input='1') then
        nxt_state <= S0;
      else
        nxt_state <= S8;
      end if;
    when S8 =>
      if(input='1') then
        nxt_state <= S3;
      else
        nxt_state <= S2;
      end if;
  end case;
end process;
```

و در نهایت با توجه به آنکه ماشین، ماشین Moore است و در نتیجه خروجی تنها وابسته به استیت است، خواهیم داشت:

```
-- evaluate output
process(ps_state)
begin
  if(ps_state=s8) then
    output <= '1';
  else
    output <= '0';
  end if;
end process;
```

بعد از طراحی ماژول اصلی نوبت میرسد به کدنویسی Test Bench:

ابتدا سیگنال ها را تعریف کرده، سپس آنها را به ماژول اصلی مپ میکنیم:

```
--Inputs
signal clk : std_logic := '0';
signal rst : std_logic := '0';
signal input : std_logic := '0';

--Output
signal output : std_logic;

-- period of clock
constant clk_period : time := 10 ns;

BEGIN

    -- Instantiate Unit Under Test(uut)
    uut: entity work.Sequence_Detector PORT
    MAP (
        clk => clk,
        rst => rst,
        input => input,
        output => output
    );
```

سپس یک Clock Generator میسازیم:

```
-- generate clock
clk_process :process
begin
    clk <= '0';
    wait for clk_period/2;
    clk <= '1';
    wait for clk_period/2;
end process;
```

و در نهایت با یک Process کد اصلی تست را مینویسیم:

```
Test Bench VHDL

tb_process: process
begin
    -- hold reset state for 5 ns(half of
    clock period).
    rst <= '0';
    wait for clk_period/2;
    rst <= '1';
    input <= '1';

    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

    wait for clk_period;
    input <= '1';

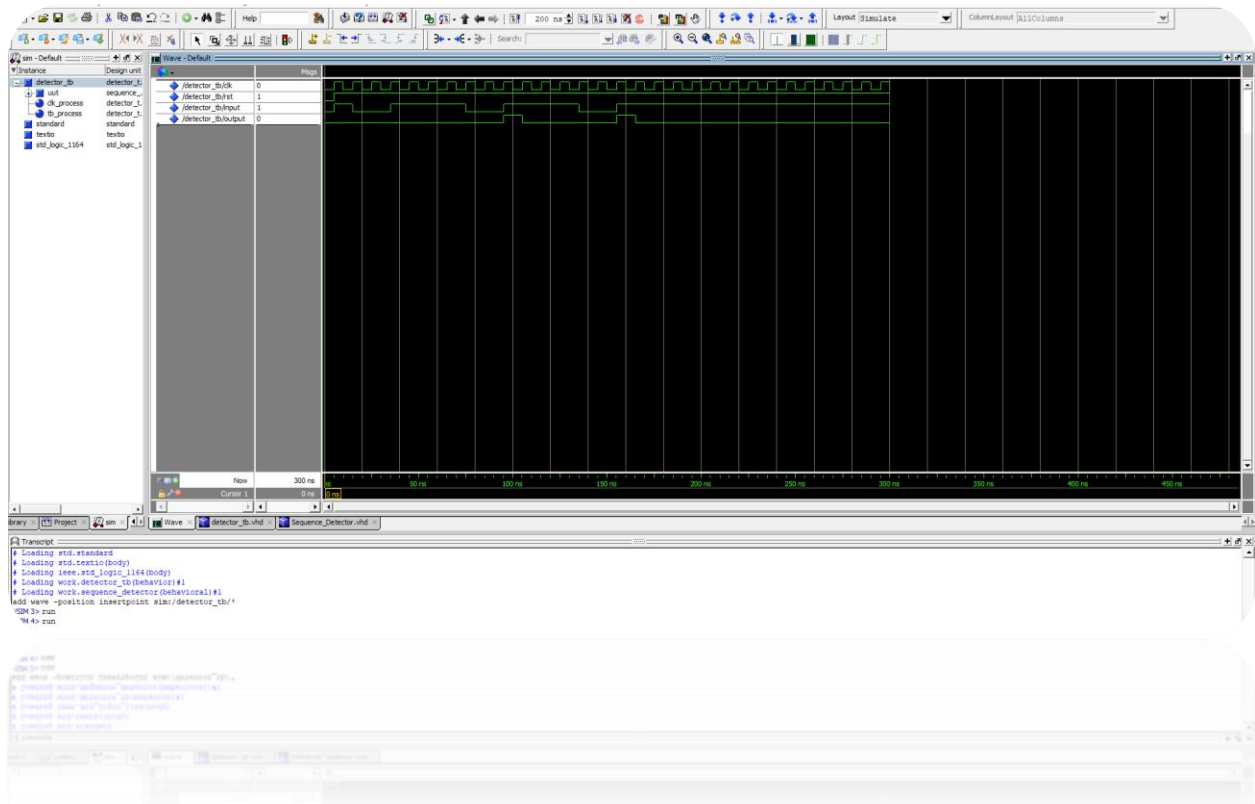
    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '0';

    wait for clk_period;
    input <= '1';

    wait;
end process;
```

سپس تست بنچ را اجرا میکنیم و نتیجه را ملاحظه میکنیم:



موفق باشید