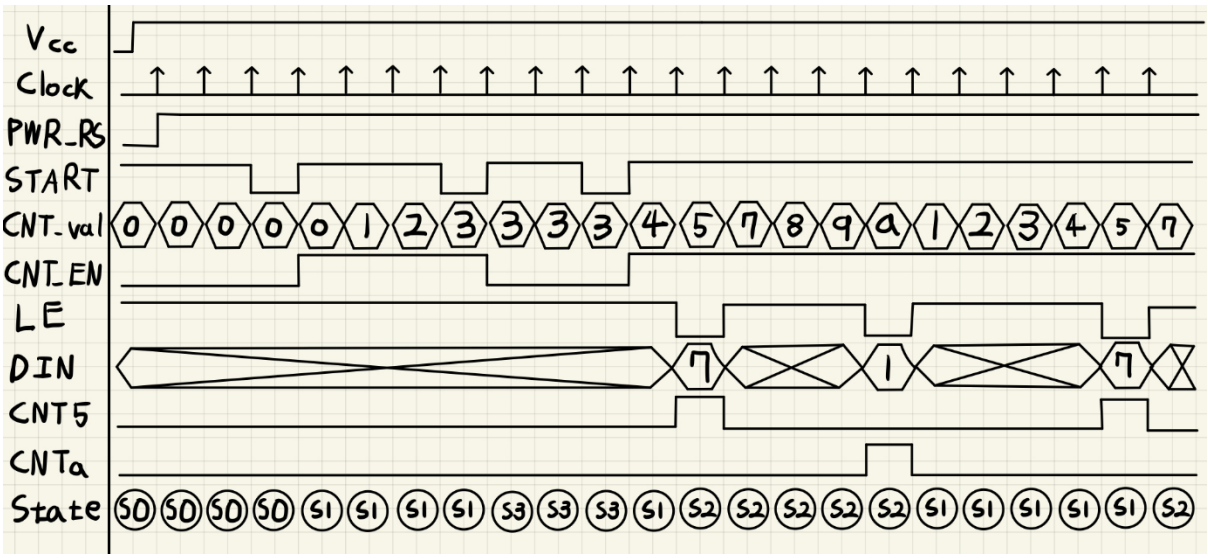
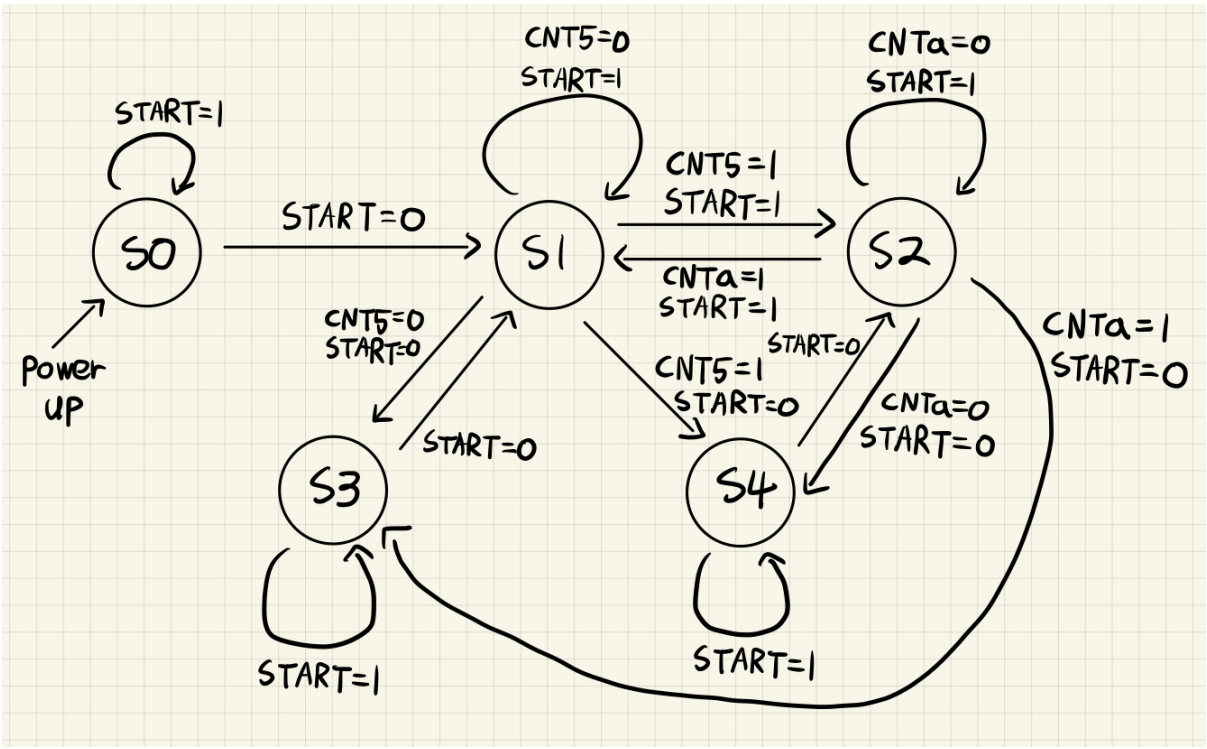
Counter Design

20181536 엄석훈

1. Draw the timing diagram and state diagram for this design





이번 디자인을 위한 timing diagram과 state diagram이다. State는 총 5가지로 처음에 시작되면 S0상태에 있다가 START가 눌리면 S1으로 가서 카운트 업을 시작한다. 그리고 중간에 START가 다

시 눌리면 S3로 가서 카운트업을 멈추었다가 다시 START가 눌리면 S1으로 돌아온다. 그리고 CNT5가 활성화되어 CNT_VAL이 5에서 7로 가야 할 때 START가 안누리면 그대로 state가 S2로 진행되고 만약 START가 눌리면 state가 S4로 가서 다음에 시작될 때 5부터 시작하게 된다. 그리고 S2에서도 유사하게 카운트업을 하다가 CNTa와 START신호에 따라서 state를 S1, S3, S4로 바꾸게 된다.

2. Write the VHDL code for this design

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

entity sequencer is
    Port (
        Clock : in STD_LOGIC;
        PWR_RS : in STD_LOGIC;
        START : in STD_LOGIC
    );
end sequencer;

architecture Behavioral of sequencer is
        type my_state is (a,b,c,d,e);
        signal state : my_state := a;

        signal CNT_VAL : std_logic_vector(3 downto 0) := "0000";
        signal CNT_EN : std_logic := '0';
        signal LE : std_logic := '1';
        signal DIN : std_logic_vector(3 downto 0) := "0111";
        signal CNT5 : std_logic := '0';
        signal CNTa : std_logic := '0';

begin
        process(PWR_RS, Clock)
        begin
                if rising_edge(PWR_RS) then
                        state <= a;
                        CNT_VAL <= "0000";
                elsif rising_edge(Clock) then
```

```
                        case state is
                                when a =>
                                        CNT_VAL <= "0000";
                                        if START = '0' then
                                                state <= b;
                                                CNT_EN <= '1';
                                        else
                                                state <= a;
                                        end if;
                                when b =>
                                        DIN <= "0111";
                                        if CNT_VAL = "0100" then
                                                CNT5 <= '1';
                                                LE <= '0';
                                        end if;
                                        if CNT5 = '0' and START = '1' then
                                                state <= b;
                                                CNT_VAL <=
std_logic_vector(unsigned(CNT_VAL) + 1);
                                        elsif CNT5 = '0' and START = '0' then
                                                state <= d;
                                                CNT_EN <= '0';
                                                CNT_VAL <=
std_logic_vector(unsigned(CNT_VAL) + 1);
                                        elsif CNT5 = '1' and START = '1' then
                                                state <= c;
                                                CNT5 <= '0';
                                                LE <= '1';
                                                CNT_VAL <= DIN;
                                        elsif CNT5 = '1' and START = '0' then
                                                state <= e;
                                                CNT_EN <= '0';
                                                CNT5 <= '0';
                                                LE <= '1';
                                                CNT_VAL <= DIN;
                                        end if;
                                when c =>
                                        DIN <= "0001";
                                        if CNT_VAL = "1001" then
```

```
                                        CNTa <= '1';
                                        LE <= '0';
                                end if;
                                if CNTa = '0' and START = '1' then
                                        state <= c;
                                        CNT_VAL <=
std_logic_vector(unsigned(CNT_VAL) + 1);
                                elsif CNTa = '0' and START = '0' then
                                        state <= e;
                                        CNT_EN <= '0';
                                        CNT_VAL <=
std_logic_vector(unsigned(CNT_VAL) + 1);
                                elsif CNTa = '1' and START = '1' then
                                        state <= b;
                                        CNTa <= '0';
                                        LE <= '1';
                                        CNT_VAL <= DIN;
                                elsif CNTa = '1' and START = '0' then
                                        state <= d;
                                        CNT_EN <= '0';
                                        CNTa <= '0';
                                        LE <= '1';
                                        CNT_VAL <= DIN;
                                end if;
                        when d =>
                                if START = '1' then
                                        state <= d;
                                else
                                        CNT_EN <= '1';
                                        state <= b;
                                end if;
                        when e =>
                                if START = '1' then
                                        state <= e;
                                else
                                        CNT_EN <= '1';
                                        state <= c;
                                end if;
                        when others =>
```

```
                                        CNT_VAL <= "0000";
                                        state <= a;
                        end case;
                end if;
        end process;
end Behavioral;
```

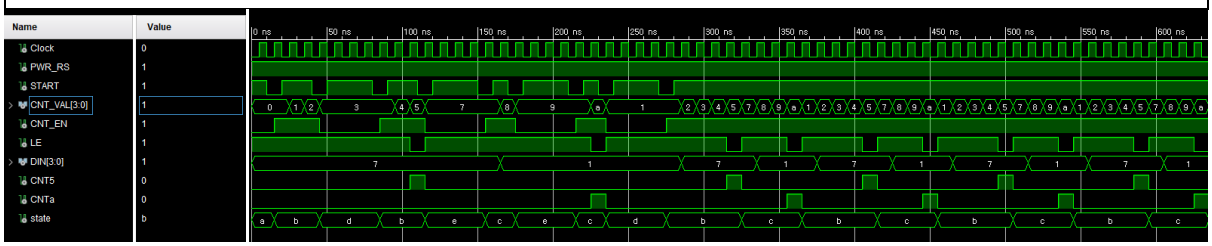
위는 이번 과제를 위한 코드이다. 간단히 설명하면 입력으로는 파워인 PWR_RS과 클럭 신호인 Clock, 마지막으로 제어를 위한 START신호가 있다. 그리고 내부의 signal로는 기본적인 state와 카운트 신호인 CNT_VAL, 카운터의 동작을 제어하는 CNT_EN, 데이터 로드 여부를 제어하는 LE, 로드할 데이터 입력인 DIN, 마지막으로 CNT5와 CNTa가 있다. 그리고 Clock의 라이징 엣지마다 각 상태에서 state diagram에서 그린 것처럼 CNT5, CNTa, START의 값에 따라서 출력과 나머지 signal을 컨트롤 해주었다. 코드는 사실 위에서 설명한 state diagram을 그대로 적은 것이다.

3. Verify your design with the test vector

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

entity sequencer_sim is
--  Port ( );
end sequencer_sim;

architecture Behavioral of sequencer_sim is
        signal Clock, PWR_RS, START : std_logic := '0';
        constant clk_period : time := 10 ns;

        component sequencer is port (
                Clock : in std_logic;
                PWR_RS : in std_logic;
                START : in std_logic
                );
    end component;
begin
        portmap : sequencer port map(
                        Clock => Clock,
                        PWR_RS => PWR_RS,
                        START => START
        );
```

```
Clock_process :process
begin
Clock <= '0';
wait for clk_period/2;
Clock <= '1';
wait for clk_period/2;
end process;

test : process
begin
        PWR_RS <= '1'; START <= '1'; wait for clk_period;

        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        START <= '0'; wait for clk_period; START <= '1';
        wait for clk_period;
        wait for clk_period;
        wait for clk_period;
```

```
                START <= '0'; wait for clk_period; START <= '1';
                wait for clk_period;
                wait;
        end process;
end Behavioral;
```



위의 그림은 내가 설계한 디자인에 대한 시뮬레이션과 그 결과이다. 각 state에서 다른 state로의 변화를 모두 테스트 벡터에 넣어서 확인해 본 결과 정상적으로 동작하는 것을 확인할 수 있다. 카운터는 1,2,3,4,5,7,8,9,a,1의 순서로 모두 동작하며 CNT_EN, LE, CNT5, CNTa모두 위에서 그린 timing diagram과 동일하게 동작하는 것을 확인할 수 있다. 또한 START가 눌릴 때 마다 일시정지와 카운트업을 돌아가면서 정상적으로 수행하는 것을 확인할 수 있다.