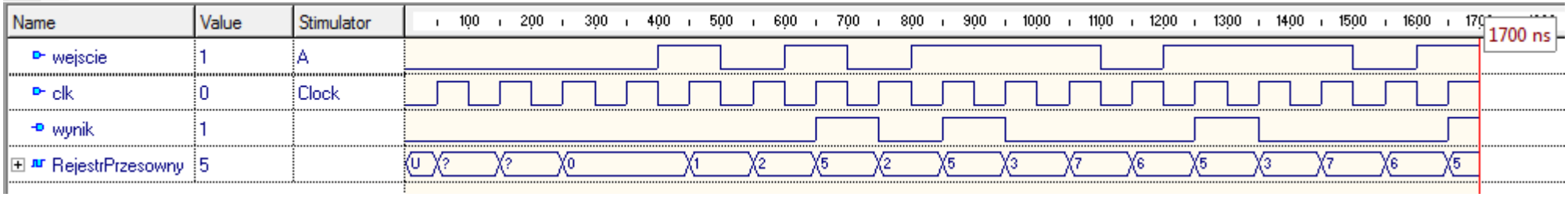
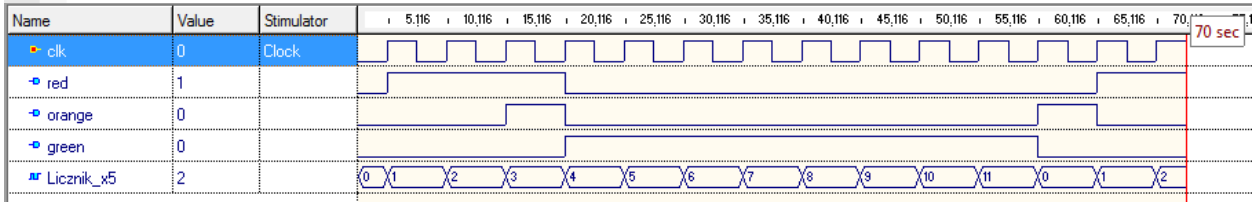
1. Detektor sekwencji „101” zbudowany w oparciu o rejestr przesuwny.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity DetektorSekwencji is
     port(
          wejscie : in STD_LOGIC;
          clk : in STD_LOGIC;
          wynik : out STD_LOGIC
          );
end DetektorSekwencji;


architecture DetektorSekwencji of DetektorSekwencji is
signal RejestrPrzesowny : std_logic_vector (2 downto 0); --3-bitowa pamięć
begin
    process (clk)    --musi być jak chcemy mieć układ taktowany zegarem
    begin
        if rising_edge(clk) then
            RejestrPrzesowny <= RejestrPrzesowny(1 downto 0) & wejscie;
        end if;
    end process;

    wynik <= '1' when RejestrPrzesowny="101"        --wykonuje sie rownolegle z procesem
            else '0';

end DetektorSekwencji;
```



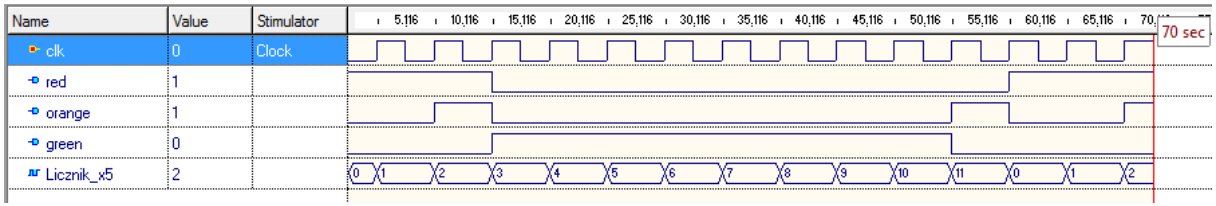
2. Automat sterujący światłami ulicznymi: np. sygnał zegarowy co 5s i stany świateł: Red= 1 dla t= 0-15s, Orange= 1 dla t= 10-15s i t= 55s-60s, Green= 1 dla t= 15-55s. Okres 60s. (użyć 3 czerwone diody).

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

entity skrzyzowanie is
    port(
        clk : in STD_LOGIC;
        red, orange, green : out STD_LOGIC := '0'
        );
end skrzyzowanie;


architecture swiatla of skrzyzowanie is
signal Licznik_x5 : integer range 11 downto 0 := 0; -- szalejemy ;D
begin
    process (clk)
    begin
        if rising_edge(clk) then
            -- jakie światła mamy zapalić??
            case (Licznik_x5) is
                when 0 | 1 =>   Red <= '1';         --<0; 10)
                                Orange <= '0';
                                Green <= '0';
                when 2 =>       Red <= '1';         --<10; 15)
                                Orange <= '1';
                                Green <= '0';
                when 3 to 10 => Red <= '0';         --<15; 55)
                                Orange <= '0';
                                Green <= '1';
                when 11 =>      Red <= '0';         --<55; 60)
                                Orange <= '1';
                                Green <= '0';
                when others =>  Red <= '0';         --żeby nie zrobił zatrzasku
                                Orange <= '0';
                                Green <= '0';
            end case;
            -- i zwiększamy licznik...
            if (Licznik_x5 = 11) then
                Licznik_x5 <=0;
            else
                Licznik_x5 <= Licznik_x5 + 1;
            end if;
        end if;
    end process;
end swiatla;
```



3. Tjw. ale lepiej ;D

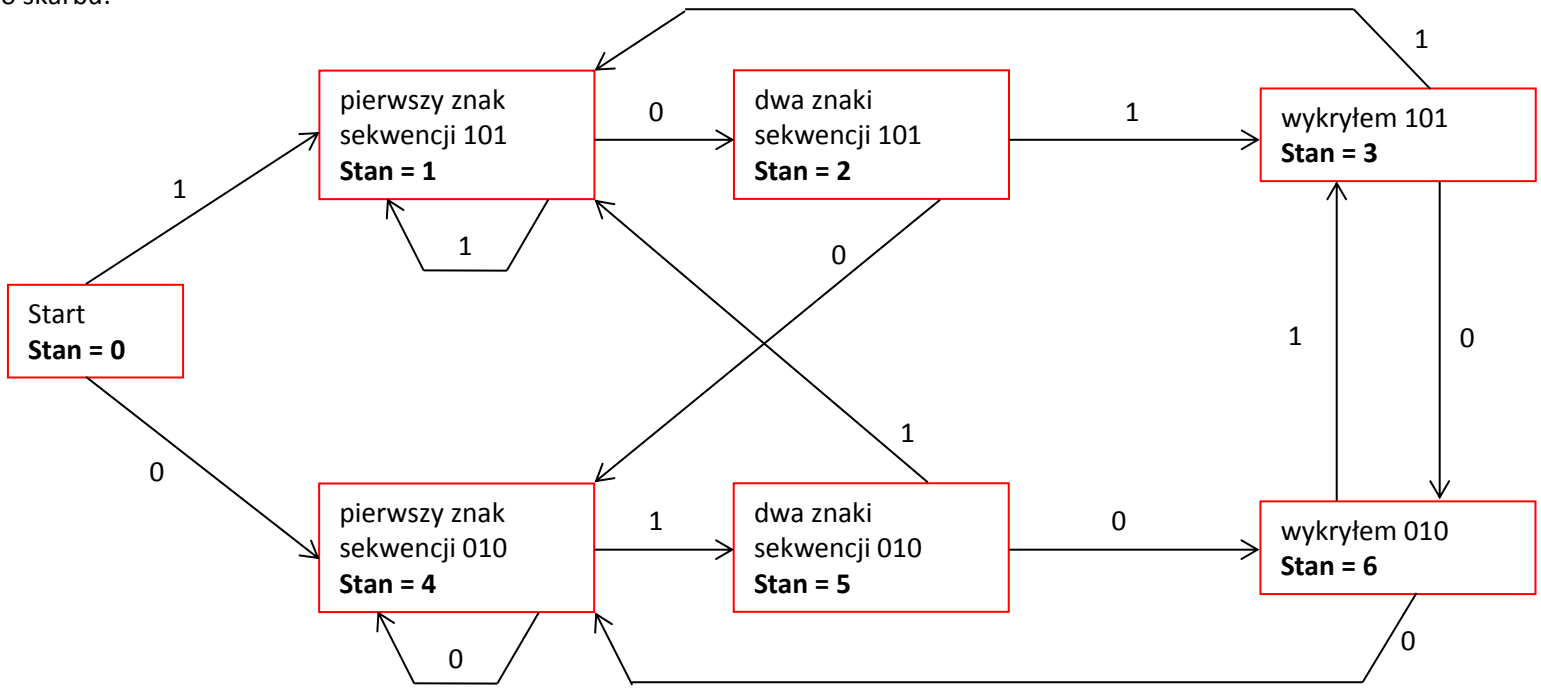
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity skrzyzowanie is
     port(
          clk : in STD_LOGIC;
          red, orange, green : out STD_LOGIC := '0'
          );
end skrzyzowanie;


architecture swiatla2 of skrzyzowanie is
signal Licznik_x5 : integer range 11 downto 0 := 0; -- szalejemy ;D
begin
    process (clk)
    begin
        if rising_edge(clk) then
            if (Licznik_x5 = 11) then
                Licznik_x5 <=0;
            else
                Licznik_x5 <= Licznik_x5 + 1;
            end if;
        end if;
    end process;
    -- warunkowe przypisania współbieżne:
    Red <= '1' when ( (Licznik_x5 <= 2) and (Licznik_x5 >= 0) )
                else '0';

    Orange <= '1' when ( (Licznik_x5 = 2) or (Licznik_x5 = 11) )
                else '0';

    Green <= '1' when ( (Licznik_x5 <= 10) and (Licznik_x5 >= 3) )
                else '0';

end swiatla2;
```

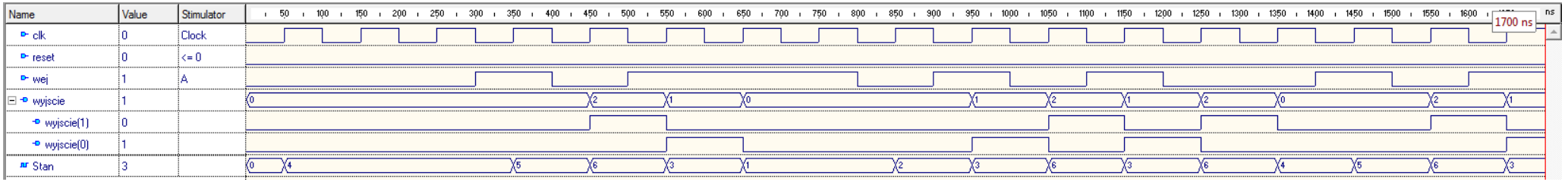


4. Detektor sekwencji 101 oraz 010. Na wyjściu detektora jest 00 jeśli sekwencja nie jest wykryta, 01 jeśli wykryta jest pierwsza sekwencja oraz 10 jeśli zostanie wykryta druga sekwencja.

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

entity DetektorSekwencji is
     port(
          clk : in STD_LOGIC;
          reset : in STD_LOGIC;
          wej : in STD_LOGIC;
          wyjscie : out STD_LOGIC_VECTOR (1 downto 0)
          );
end DetektorSekwencji;


architecture DetektorSekwencji of DetektorSekwencji is
signal Stan : integer range 0 to 6 := 0; --typ całkowity zmieniający się od 0 do 6
begin
    process (clk)    --musi być jak chcemy mieć układ taktowany zegarem
    begin
        if rising_edge(clk) then
            if (reset = '1') then
                Stan <= 0;
            else
                case (Stan) is
                when 0 =>   if (wej = '1') then      -- jak jestes w 'Start'
                                Stan <= 1;
                            else
                                Stan <= 4;
                            end if;
                when 1 =>   if (wej = '1') then      -- jak jestes w 'pierwszy znak sekwencji 101'
                                Stan <= 1;
                            else
                                Stan <= 2;
                            end if;
                when 2 =>   if (wej = '1') then      -- jak jestes w 'dwa znaki sekwencji 101'
                                Stan <= 3;
                            else
                                Stan <= 4;
                            end if;
                when 3 =>   if (wej = '1') then      -- jak jestes w 'wykrylem 101'
                                Stan <= 1;
                            else
                                Stan <= 6;
                            end if;
                when 4 =>   if (wej = '1') then      -- jak jestes w 'pierwszy znak sekwencji 010'
                                Stan <= 5;
                            else
                                Stan <= 4;
                            end if;
                when 5 =>   if (wej = '1') then      -- jak jestes w 'dwa znaki sekwencji 010'
                                Stan <= 1;
                            else
                                Stan <= 6;
                            end if;
                when 6 =>   if (wej = '1') then      -- jak jestes w 'wykrylem 010'
                                Stan <= 3;
                            else
                                Stan <= 4;
                            end if;
                when others => Stan <= 0;
                end case;
            end if;
        end if;
    end process;

    wyjscie <=  "01" when Stan = 3 else      --wykonuje sie rownolegle z procesem
                "10" when Stan = 6 else
                "00";

end DetektorSekwencji;
```

Mapa do skarbu:





W ten sposób można zrealizować maszynę stanów w języku vhdl.
Funkcje przejść i wyjść wygeneruje narzędzie do syntezy…