

Sterownik akwarium.

Zasada działania:

Na wejście sterownika jest podawana temperatura wody. Sterownik powinien włączać światło w akwarium co 12 godzin, na 8 godzin. Gdy temperatura wody spadnie poniżej temp. 10 stopni powinna zostać włączona grzałka.

Układ za pomocą zewnętrznego zegara wytwarza swój własny zegarek 24-godzinny. Dzięki niemu, o odpowiednich porach dnia włącza i wyłącza światło. Ponadto, na bazie odczytu temperatury, włącza i wyłącza grzałkę tak, aby temperatura wody nie spadła poniżej 10 stopni Celsjusza.

Układ ten podzielony jest na kilka modułów:

Zegarek – w takt podanego na wejściu (zbrocza narastającego) zegara *zeg* (1 Hz) zlicza on i zwraca czas wyrażony 3 liczbami: godzin *zeg_g*, minut *zeg_m*, sekund *zeg_s*. Czas przedstawiony jest w formacie 24-godzinny. Sekundę po godzinie 23:59:59 zegarek „zawraca” do 00:00:00.

```
20  -- Zegarek -----
21  procedure zegarek
22      (signal zeg      : in std_logic; -- zegar zewnętrzny 1 Hz
23       signal zeg_g    : inout std_logic_vector (4 downto 0); -- liczba godzin
24       signal zeg_m    : inout std_logic_vector (5 downto 0); -- liczba minut
25       signal zeg_s    : inout std_logic_vector (5 downto 0)) is -- liczba sekund
26  begin
27      if (zeg'event and zeg = '1') then
28          if (zeg_s < 59) then
29              zeg_s <= zeg_s + 1;
30          else
31              zeg_s <= "000000";
32              if (zeg_m < 59) then
33                  zeg_m <= zeg_m + 1;
34              else
35                  zeg_m <= "000000";
36                  if (zeg_g < 23) then
37                      zeg_g <= zeg_g + 1;
38                  else
39                      zeg_g <= "000000";
40                  end if;
41              end if;
42          end if;
43      end if;
44  end zegarek;
45  -----
```

Grzałka – moduł ten sprawdza czy temperatura wody *temp* jest niższa niż 10 stopni Celsjusza. Jeśli tak - włącza grzałkę (*state* = 1); jeśli nie – wyłącza (*state* = 0). Temperatura wyrażona jest w kelwinach (a więc liczbowo jest większa o 273 od wyrażonej w stopniach Celsjusza – przyjmuje wartości od 0 wzwyż).

```
47  -- Grzałka -----
48  procedure grzalka
49      (signal temp      : in std_logic_vector (8 downto 0); -- temperatura (0 - 511 K)
50       signal stan      : out std_logic) is -- stan grzałki (1 - wł, 0 - wł)
51  begin
52      if (temp < 283) then -- jeśli temperatura poniżej 10°C
53          stan <= '1';
54      else
55          stan <= '0';
56      end if;
57  end grzalka;
58  -----
```

Światło – moduł ten włącza co 12 godzin światło, które świeci przez 8 godzin. Jego wejścia to aktualne: godzina *clk_h* i minuta *clk_m*, oraz godzina *wl_h* i minuta *wl_m* włączenia światła - reprezentowane wektorami 4- i 5-bitowymi. Port dwukierunkowy *state* reprezentuje stan światła (1 – włączone, 0 – wyłączone). Zmienne *tmp*, *tmp_on_1*, *tmp_off_1*, *tmp_on_2*, *tmp_off_2* reprezentują kolejno: aktualną godzinę, oraz godziny włączenia i wyłączenia światła - wyrażone w minutach od godziny 0:00 (np. dla godziny 11:20 -> $tmp = 11 \cdot 60 + 20 = 680$). Dla każdego z 4 ogólnych przypadków godziny włączenia alarmu *wl_h* (I – przed godziną 4; II – od 12, przed 16; III – od 4, przed 12; IV – od 16) są opisane oddzielne algorytmy wyznaczania wartości w/w godzin włączania i wyłączania oraz stanu światła w zależności od aktualnej godziny.

```

60  -- Światło -----
61  procedure swiatlo
62      (signal clk_h : in std_logic_vector (4 downto 0); -- aktualna godzina
63       signal clk_m : in std_logic_vector (5 downto 0); -- aktualna minuta
64       signal wl_h  : in std_logic_vector (4 downto 0); -- godzina włączenia światła
65       signal wl_m  : in std_logic_vector (5 downto 0); -- minuta włączenia światła
66       signal state : inout std_logic) is -- stan światła (1 - wł, 0 - wł)
67       variable tmp : integer; -- zmienna pomocniczna na aktualny czas w minutach
68       variable tmp_on_1, tmp_off_1, tmp_on_2, tmp_off_2 : integer; -- zmienne pomocnicze
69                               -- na czasy włączania i wyłączania w minutach
70  begin
71      tmp := 60*to_integer(unsigned(clk_h)) + to_integer(unsigned(clk_m));
72
73      if (wl_h < 4) then
74          -- ustalenie punktów zmiany stanu światła na 24h osi czasu --
75          tmp_on_1 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
76          tmp_off_1 := tmp_on_1 + 60*8; -- 8h po tmp_on_1 wyłącza się
77          tmp_on_2 := tmp_on_1 + 60*12; -- włącza się 12h później niż tmp_on_1
78          tmp_off_2 := tmp_on_2 + 60*8; -- 8h po tmp_on_2 wyłącza się
79          -- zachowanie światła w zależności, w zależności od aktualnej godziny --
80          if ((tmp >= tmp_on_1 and tmp < tmp_off_1) or (tmp >= tmp_on_2 and tmp < tmp_off_2)) then
81              state <= '1';
82          else
83              state <= '0';
84          end if;
85
86      elsif (wl_h >= 12 and wl_h < 16) then
87          tmp_on_2 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
88          tmp_off_2 := tmp_on_2 + 60*8; -- 8h po tmp_on_2 wyłącza się
89          tmp_on_1 := tmp_on_2 - 60*12; -- włącza się 12h wcześniej niż tmp_on_2
90          tmp_off_1 := tmp_on_1 + 60*8; -- 8h po tmp_on_1 wyłącza się
91          if ((tmp >= tmp_on_1 and tmp < tmp_off_1) or (tmp >= tmp_on_2 and tmp < tmp_off_2)) then
92              state <= '1';
93          else
94              state <= '0';
95          end if;
96
97      elsif (wl_h >= 4 and wl_h < 12) then
98          tmp_on_1 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
99          tmp_off_1 := tmp_on_1 - 60*4; -- 4h przed tmp_on_1 wyłącza się
100         tmp_on_2 := tmp_on_1 + 60*12; -- włącza się 12h później niż tmp_on_1
101         tmp_off_2 := tmp_on_2 - 60*4; -- 4h przed tmp_on_2 wyłącza się
102         if ((tmp < tmp_off_1) or (tmp >= tmp_on_1 and tmp < tmp_off_2) or (tmp >= tmp_on_2)) then
103             state <= '1';
104         else
105             state <= '0';
106         end if;
107
108      elsif (wl_h >= 16) then
109          tmp_on_2 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
110          tmp_off_2 := tmp_on_2 - 60*4; -- 4h przed tmp_on_2 wyłącza się
111          tmp_on_1 := tmp_on_2 - 60*12; -- włącza się 12h wcześniej niż tmp_on_2
112          tmp_off_1 := tmp_on_1 - 60*4; -- 4h przed tmp_on_1 wyłącza się
113          if ((tmp < tmp_off_1) or (tmp >= tmp_on_1 and tmp < tmp_off_2) or (tmp >= tmp_on_2)) then
114              state <= '1';
115          else
116              state <= '0';
117          end if;
118      end if;
119  end swiatlo;
120 -----
121

```

Opis całego układu w języku VHDL:

Układ ma wejście zegarowe 1-bitowe *Clk* na zegar zewnętrzny (zastosowano 1 Hz). Ponadto ma porty dwukierunkowe: *time_h*, *time_m*, *time_s* reprezentujące aktualny czas – odpowiednio godzinę, minutę, sekundę (zapisane w postaci wektorów 5- i 6-bitowych, w zależności czy potrzeba zapisać 24 czy 60 różnych stanów, mając do dyspozycji $2^5=32$ lub $2^6=64$ bity). Domyślnie porty te są wyzerowane – zegarek startuje od 00:00:00. Kolejny port dwukierunkowy to 9-bitowy *temperature* podający pomiar temperatury wody (wyrażonej w Kelwinach) – z uwagi na 9-bitową reprezentację temperatura może wynosić od 0 do 511 K (odpowiednio -273°C do 238°C). Domyślnie ustawiony jest na 282K. Port dwukierunkowy *state_grzałka* reprezentuje stan grzałki. Kolejne porty dwukierunkowe to: *light_h* i *light_m* będące wektorami (5- i 6-bitowym) reprezentującymi godzinę i minutę włączenia światła (układ akceptuje zarówno godziny przed jak i po południu, tzn. zachowuje się tak samo, jeśli każemy włączać światło o 1:30 albo 13:30). Domyślnie godzina włączania światła to 13:30. Port dwukierunkowy *state_swiatlo* reprezentuje stan tego światła.

Układ działa sekwencyjnie, jego działanie wyzwalane jest zewnętrznym zegarem *Clk*. W procesie kolejno wywoływane są procedury: *zegarek* (reprezentująca zegar 24-godzinny), *grzałka* (sterująca grzałką), *swiatlo* (sterująca światłem).

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4  USE IEEE.NUMERIC_STD.ALL;
5
6  entity Zad2 is
7      port(Clk          : in std_logic; -- zegar zewnętrzny 1 Hz
8           time_h       : inout std_logic_vector (4 downto 0) := "00000"; -- aktualna godzina
9           time_m       : inout std_logic_vector (5 downto 0) := "000000"; -- aktualna minuta
10          time_s       : inout std_logic_vector (5 downto 0) := "000000"; -- aktualna sekunda
11          temperature  : inout std_logic_vector (8 downto 0) := "100011010"; -- temperatura wody (w Kelwinach)
12          state_grzałka : inout std_logic; -- stan grzałki (1 - wł, 0 - wł)
13          light_h      : inout std_logic_vector (4 downto 0) := "01101"; -- godzina włączenia światła
14          light_m      : inout std_logic_vector (5 downto 0) := "011110"; -- minuta włączenia światła
15          state_swiatlo : inout std_logic; -- stan światła (1 - wł, 0 - wł)
16  end Zad2;
17
18  architecture Behavioral of Zad2 is
19
20      -- Zegarek -----
21      procedure zegarek
22          (signal zeg : in std_logic; -- zegar zewnętrzny 1 Hz
23           signal zeg_g : inout std_logic_vector (4 downto 0); -- liczba godzin
24           signal zeg_m : inout std_logic_vector (5 downto 0); -- liczba minut
25           signal zeg_s : inout std_logic_vector (5 downto 0)) is -- liczba sekund
26      begin
27          if (zeg'event and zeg = '1') then
28              if (zeg_s < 59) then
29                  zeg_s <= zeg_s + 1;
30              else
31                  zeg_s <= "000000";
32                  if (zeg_m < 59) then
33                      zeg_m <= zeg_m + 1;
34                  else
35                      zeg_m <= "000000";
36                      if (zeg_g < 23) then
37                          zeg_g <= zeg_g + 1;
38                      else
39                          zeg_g <= "000000";
40                      end if;
41                  end if;
42              end if;
43          end if;
44      end zegarek;
45      -----
```

```

46
47 -- Grzałka -----
48 procedure grzałka
49     (signal temp : in std_logic_vector (8 downto 0); -- temperatura (0 - 511 K)
50      signal stan : out std_logic) is -- stan grzałki (1 - wł, 0 - wył)
51 begin
52     if (temp < 283) then -- jeśli temperatura poniżej 10°C
53         stan <= '1';
54     else
55         stan <= '0';
56     end if;
57 end grzałka;
58 -----
59
60 -- Światło -----
61 procedure swiatlo
62     (signal clk_h : in std_logic_vector (4 downto 0); -- aktualna godzina
63      signal clk_m : in std_logic_vector (5 downto 0); -- aktualna minuta
64      signal wl_h : in std_logic_vector (4 downto 0); -- godzina włączenia światła
65      signal wl_m : in std_logic_vector (5 downto 0); -- minuta włączenia światła
66      signal state : inout std_logic) is -- stan światła (1 - wł, 0 - wył)
67     variable tmp : integer; -- zmienna pomocniczna na aktualny czas w minutach
68     variable tmp_on_1, tmp_off_1, tmp_on_2, tmp_off_2 : integer; -- zmienne pomocnicze
69                                     -- na czasy włączenia i wyłączenia w minutach
70 begin
71     tmp := 60*to_integer(unsigned(clk_h)) + to_integer(unsigned(clk_m));
72
73     if (wl_h < 4) then
74         -- ustalenie punktów zmiany stanu światła na 24h osi czasu --
75         tmp_on_1 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
76         tmp_off_1 := tmp_on_1 + 60*8; -- 8h po tmp_on_1 wyłącza się
77         tmp_on_2 := tmp_on_1 + 60*12; -- włącza się 12h później niż tmp_on_1
78         tmp_off_2 := tmp_on_2 + 60*8; -- 8h po tmp_on_2 wyłącza się
79         -- zachowanie światła w zależności, w zależności od aktualnej godziny --
80         if ((tmp >= tmp_on_1 and tmp < tmp_off_1) or (tmp >= tmp_on_2 and tmp < tmp_off_2)) then
81             state <= '1';
82         else
83             state <= '0';
84         end if;
85
86     elsif (wl_h >= 12 and wl_h < 16) then
87         tmp_on_2 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
88         tmp_off_2 := tmp_on_2 + 60*8; -- 8h po tmp_on_2 wyłącza się
89         tmp_on_1 := tmp_on_2 - 60*12; -- włącza się 12h wcześniej niż tmp_on_1
90         tmp_off_1 := tmp_on_1 + 60*8; -- 8h po tmp_on_1 wyłącza się
91         if ((tmp >= tmp_on_1 and tmp < tmp_off_1) or (tmp >= tmp_on_2 and tmp < tmp_off_2)) then
92             state <= '1';
93         else
94             state <= '0';
95         end if;
96
97     elsif (wl_h >= 4 and wl_h < 12) then
98         tmp_on_1 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
99         tmp_off_1 := tmp_on_1 - 60*4; -- 4h przed tmp_on_1 wyłącza się
100        tmp_on_2 := tmp_on_1 + 60*12; -- włącza się 12h później niż tmp_on_1
101        tmp_off_2 := tmp_on_2 - 60*4; -- 4h przed tmp_on_2 wyłącza się
102        if ((tmp < tmp_off_1) or (tmp >= tmp_on_1 and tmp < tmp_off_2) or (tmp >= tmp_on_2)) then
103            state <= '1';
104        else
105            state <= '0';
106        end if;

```

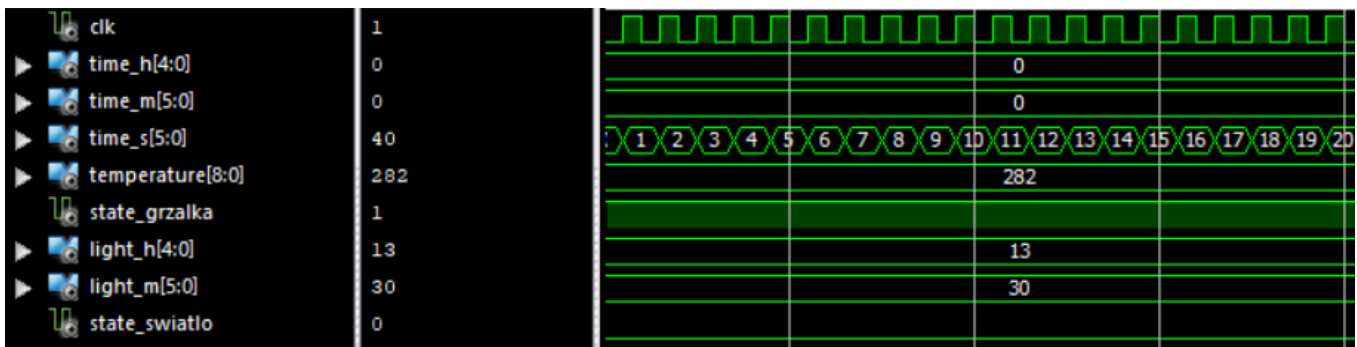
```

107
108     elsif (wl_h >= 16) then
109         tmp_on_2 := 60*to_integer(unsigned(wl_h)) + to_integer(unsigned(wl_m));
110         tmp_off_2 := tmp_on_2 - 60*4; -- 4h przed tmp_on_2 wyłącza się
111         tmp_on_1 := tmp_on_2 - 60*12; -- włącza się 12h wcześniej niż tmp_on_2
112         tmp_off_1 := tmp_on_1 - 60*4; -- 4h przed tmp_on_1 wyłącza się
113         if ((tmp < tmp_off_1) or (tmp >= tmp_on_1 and tmp < tmp_off_2) or (tmp >= tmp_on_2)) then
114             state <= '1';
115         else
116             state <= '0';
117         end if;
118
119     end if;
120 end swiatlo;
121 -----
122
123 -- Układ -----
124 begin
125     process (Clk)
126     begin
127         zegarek (Clk, time_h, time_m, time_s);
128         grzalka (temperature, state_grzalka);
129         swiatlo (time_h, time_m, light_h, light_m, state_swiatlo);
130     end process;
131 end Behavioral;
132 -----

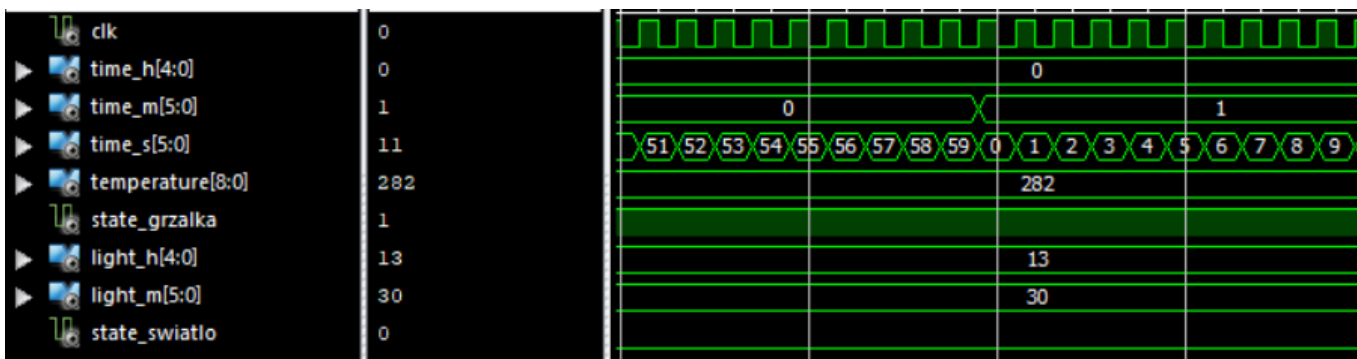
```

Przebiegi symulacyjne:

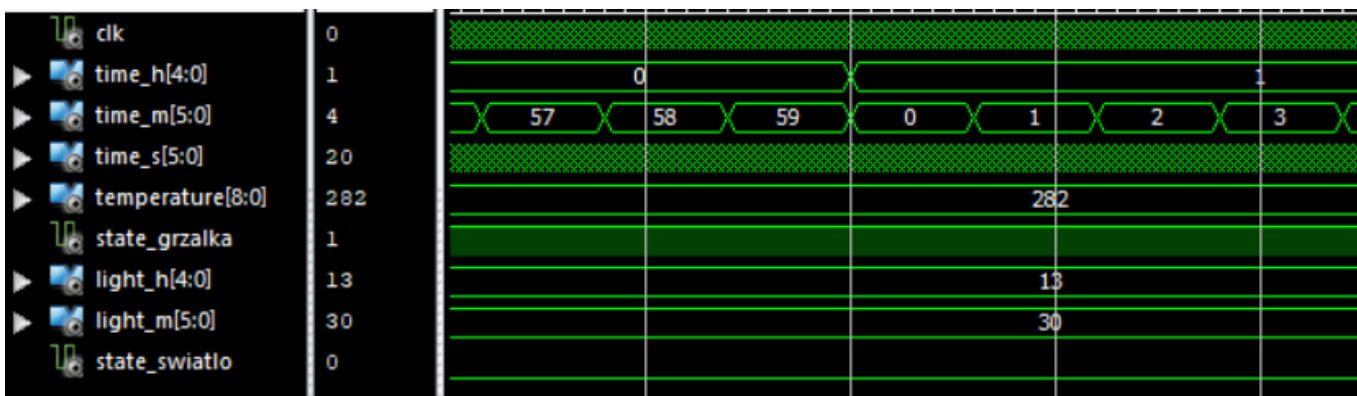
Pierwotnie ustalona godzina to 00:00:00. W takt (zbrocza narastającego) zegara wejściowego *clk* inkrementuje się *time_s*. Ponieważ *temperature* wynosi 282, a więc jest niższa od 283, *state_grzałka* wynosi 1 – grzałka pracuje. Godzina włączenia światła jest ustalona na 13:30. Oznacza to, że będzie ono pracować od 13:30:00 do 21:29:59 i od 01:30:00 do 09:29:59. Ponieważ jest północ, *state_swiatlo* = 0 – światło jest wyłączone.



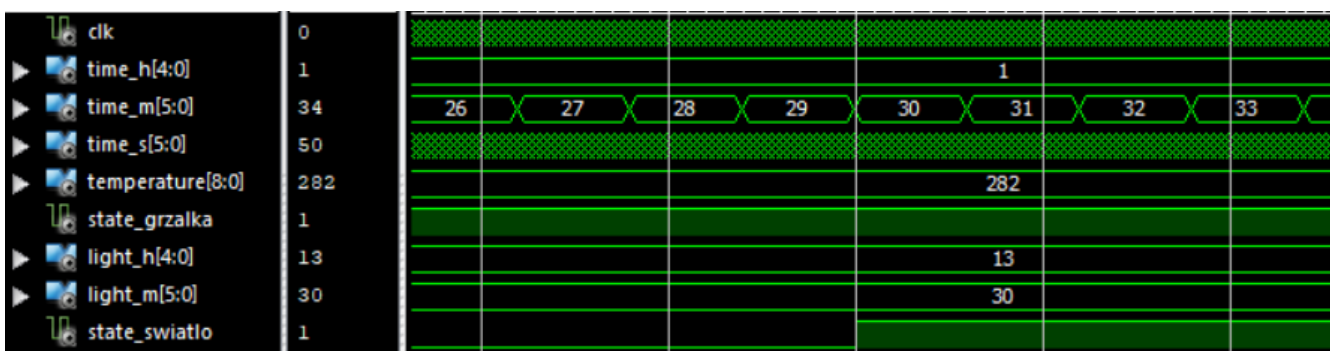
Gdy jest godzina 00:00:59 i nadchodzi kolejne zbocze narastające zegara zewnętrznego *Clk*, *time_s* „zawraca” do 00, a *time_m* inkrementuje się. Następnie *time_s* dalej inkrementuje się w takt zbocza narastającego *Clk*.



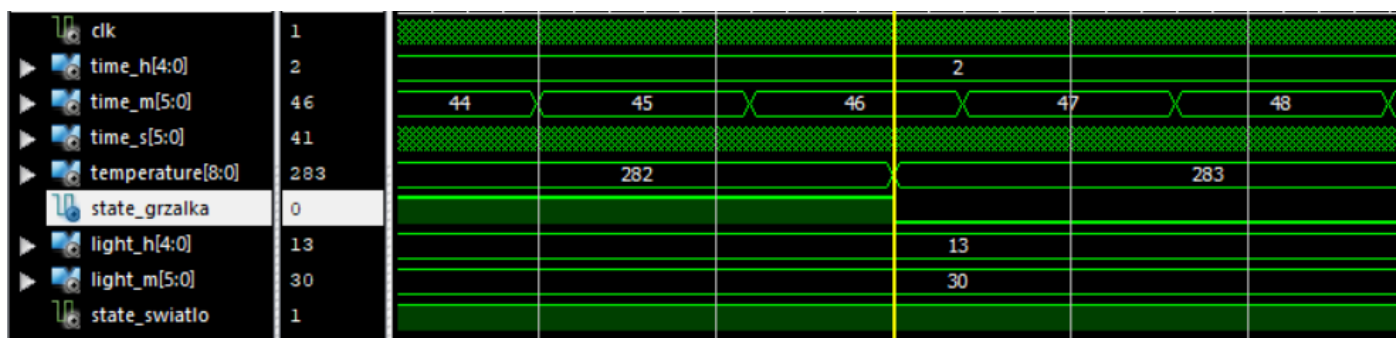
Gdy jest godzina 00:59:59 i nadchodzi kolejne zbocze narastające zegara zewnętrznego *Clk*, *time_m* „zawraca” do 00, a *time_h* inkrementuje się. Następnie *time_m* dalej inkrementuje się, gdy *time_s* „zawraca” do 00 – czyli co minutę.



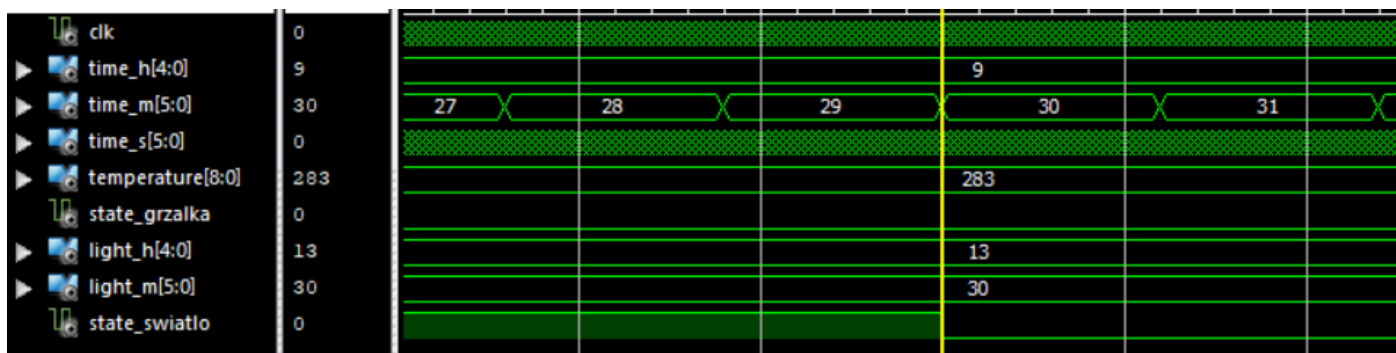
O godzinie 01:30:00, zgodnie z harmonogramem, *state_swiatlo* = 1 – światło zaczyna pracować.



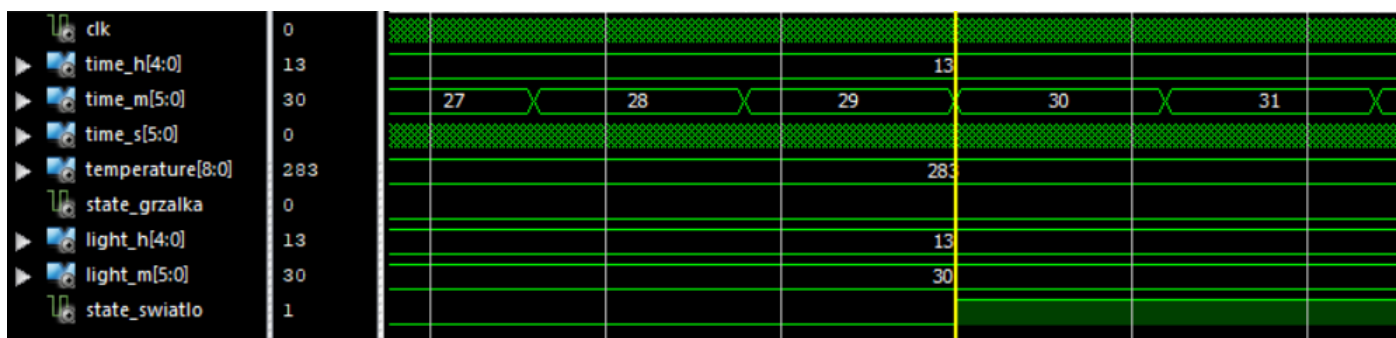
O godzinie 2:46:41, $temperature = 283$. Temperatura nie jest już niższa od 283K, więc $state_grzalka = 0$ – grzałka przestaje pracować.



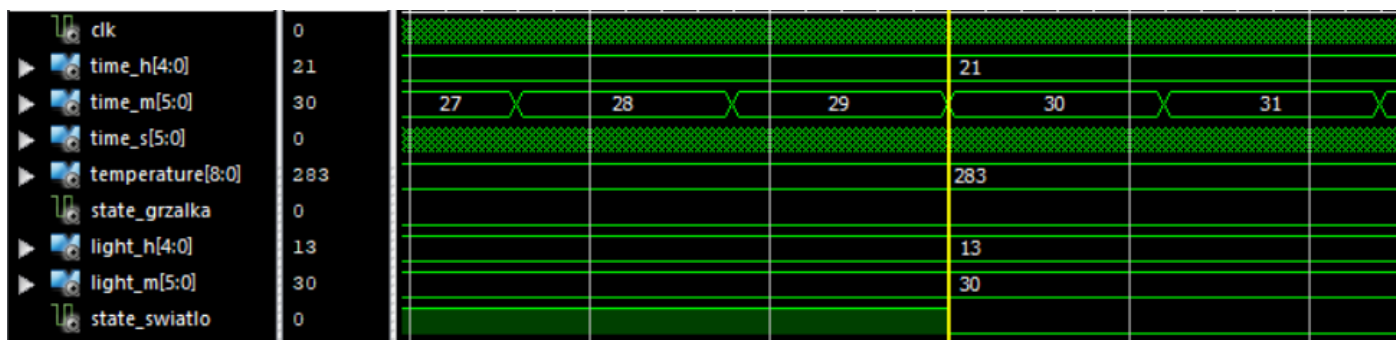
O godzinie 9:30, zgodnie z harmonogramem, $state_swiatlo = 0$ – światło przestaje pracować.



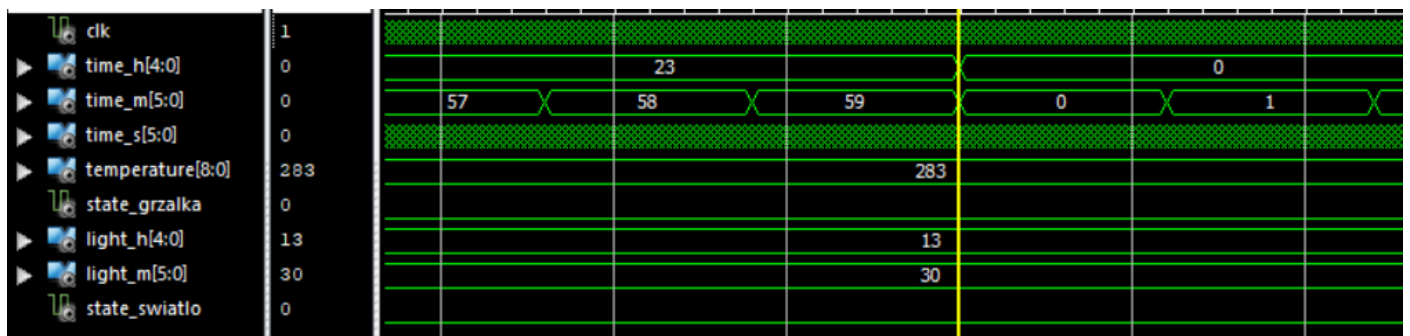
O godzinie 13:30:00, zgodnie z harmonogramem, $state_swiatlo = 1$ – światło zaczyna pracować.



O godzinie 21:30, zgodnie z harmonogramem, $state_swiatlo = 0$ – światło przestaje pracować.



Zegarek „zawraca” z 23:59:59 na 00:00:00.



Cały cykl pracy układu powtarza się dzień po dniu.

