

Cuprins

1. Specificația proiectului
2. Schema bloc
3. Componente
4. Proiectare și implementare
5. Justificarea soluției alese
6. Utilizare
7. Posibilități de dezvoltare ulterioară

1. Specificația proiectului

Automat care măsoară temperature ambientă

Cerința proiectului este:

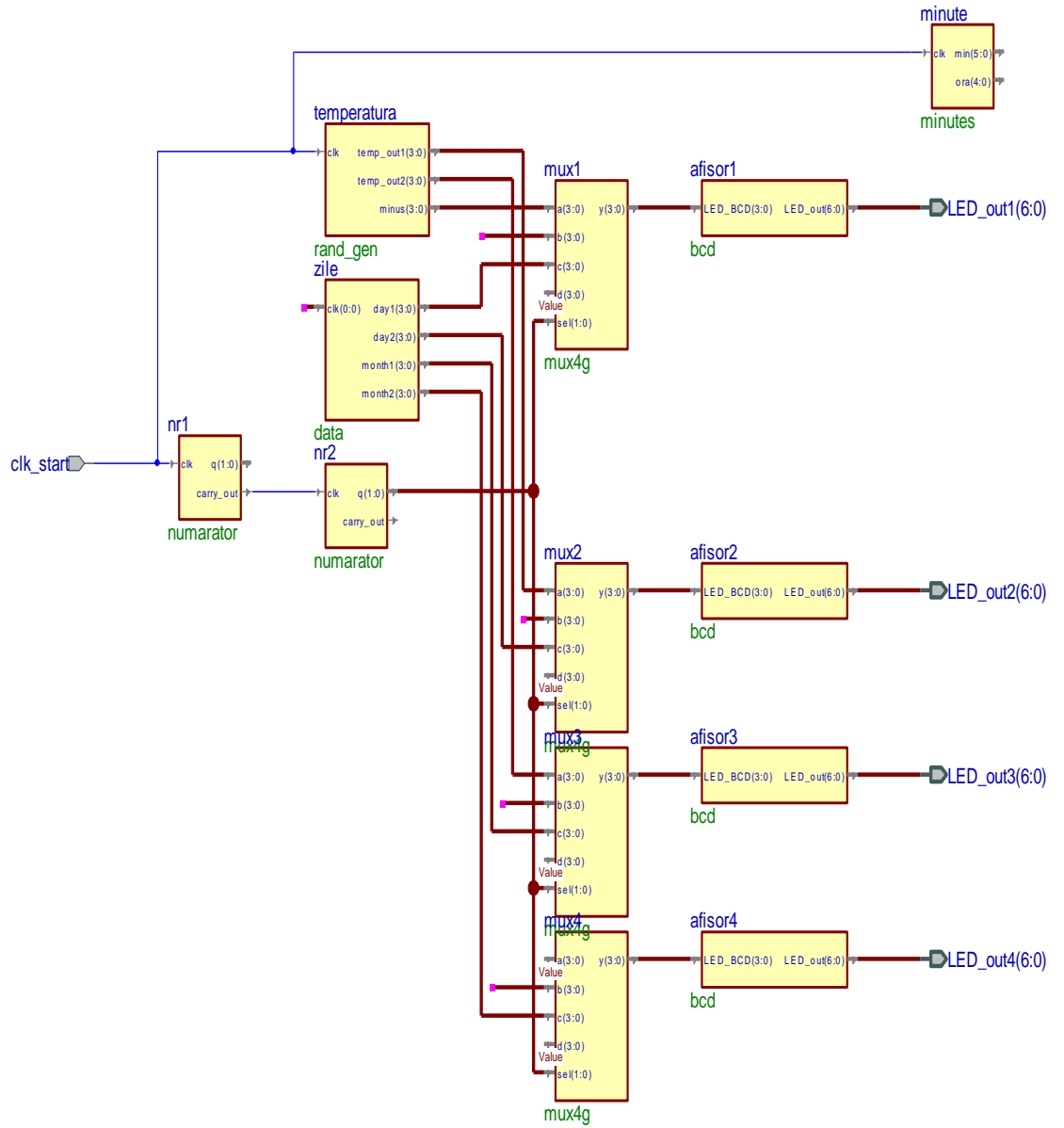
Să se proiecteze un automat care măsoară temperatura ambientă și afișează în mod ciclic temperatura măsurată (pozitivă sau negativă), ora curentă (oră și minut) și data curentă (zi, lună). Afișoarele utilizate vor fi de tipul 7 segmente. Proiectul va fi realizat de 1 student.

2. Schema bloc

Cutia neagră a automatului este prezentată mai jos, având drept intrare un singur buton de start, care pune în funcțiune întreg circuitul. Ieșirile automatului sunt 4 afișoare bcd 7 segmente, pe care se prezintă în mod ciclic, o dată la 3 secunde (o secundă fiind reprezentată de un impuls de tact) temperatura, data în format DD/MM și ora în format MM/HH. Schema logică a automatului este următoarea:

Design Unit Header

```
library ieee;
use ieee.std_logic_1...
use ieee.STD_LOGI...
```



3. Componente

1. Algoritmul de calcul al temperaturii

Dat fiind faptul că nu dispunem de un senzor care să detecteze temperatura, aceasta a fost generată în mod aleator. Se pornește de la o temperatură de 16 grade C care este incrementată sau decrementată în mod treptat la fiecare afișare a sa. Astfel temperature se schimbă o data la 3 secunde.

Leșirea minus din cardul algoritmului de calcul al temperaturii determină dacă numărul afișat este pozitiv sau negativ(pentru numere pozitive minus are valoare 15 iar pentru cele negative minus are valoare 10).

Limita minima a temperaturii este de -30 de grade C iar cea maximă este de 50 grade C.

C este afișat de fiecare data pe ultimul afișor în momentul în care apare pe ecran temperatura.

Pentru a putea să transformăm valoarea temperaturii din integer în std_logic_vector am folosit doua semnale(temp_in1 pentru cifra zecilor și temp_in2 pentru cifra unităților) care sunt transformate în std_logic_vector cu ajutorul funcției conv_std_logic_vector().

2.Calculu orei

Pentru a calcula ora și minutele, am folosit un semnal sec , care crește cu o unitate la un impuls de cesa. Cand sec ajunge la valoarea 59

aceasta se resetează iar minute2 crește cu o unitate (la fel ca la temperatură am folosit cate un semnal pentru cifra zecilor si cifra unităților , minute1 pentru zeci, minute2 pentru unități și ore1 respectiv ore2). În momentul in care se ajunge pe minute2 =9 și minute1= 5 atunci crește valoarea pentru ore2 și se reseteaza minute1 și minute2.

3. Algoritmul de calcul al zilei curente

Acest algoritm pornește de la data 01.01.2020 și pe baza semnalului provenit de carry_ore ziua se crește cu o unitate. În cazul în care se ajunge la ultima zi din lună, valoare zilei se resetează la 1 iar luna crește cu o unitate. Acest algoritm tratează și cazul în care suntem in an bisect astfel luna februarie având 29 de zile.

4. Multiplexor

Are rolul de a selecția informația care urmează să fie afișată. Selecția multiplexorului este reprezentată de valoarea ce provine de la al doilea numărător modulo 3.

Intrările multiplexorului și ieșirea sa sunt pe 4 biți.

5. BCD-ul 7 segment

Componenta BCD 7 segmente are rolul de a afișa informația selectată de către multiplexoare . Pentru temperatură primul bcd afișează un minus dacă avem o temperatură negativă, următorul afișează cifra zecilor, iar al treilea afișează cifra unităților și ultimul numărător

afisează litera C. Pentru dată se afișează în formatul DD/MM iar pentru oră se afișează în formatul MM/HH.

4. Proiectare și implementare

Proiectul simulează un termometru ce are următoarele funcții:

- Afișarea temperaturii
- Afișarea orei în format MM/HH
- Afișarea datei în format DD/MM

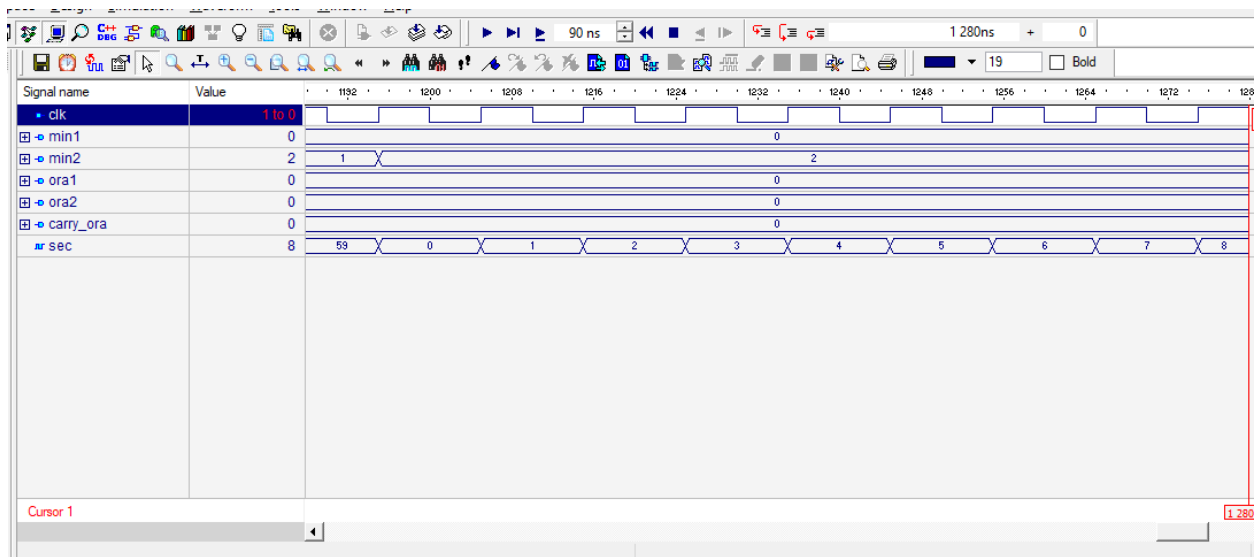
Afișarea se efectuează cu ajutorul unor afișoare de tipul 7 segmente. Aceasta este ciclică, aceasta schimbându-se o dată la 3 secunde.

Avem două numărătoare modulo 3 cascade. Carry out-ul primului numărător va reprezenta clock-ul celui de-al doilea. ieșirea celui de-al doilea numărător va fi selecția celor 4 mux-uri.

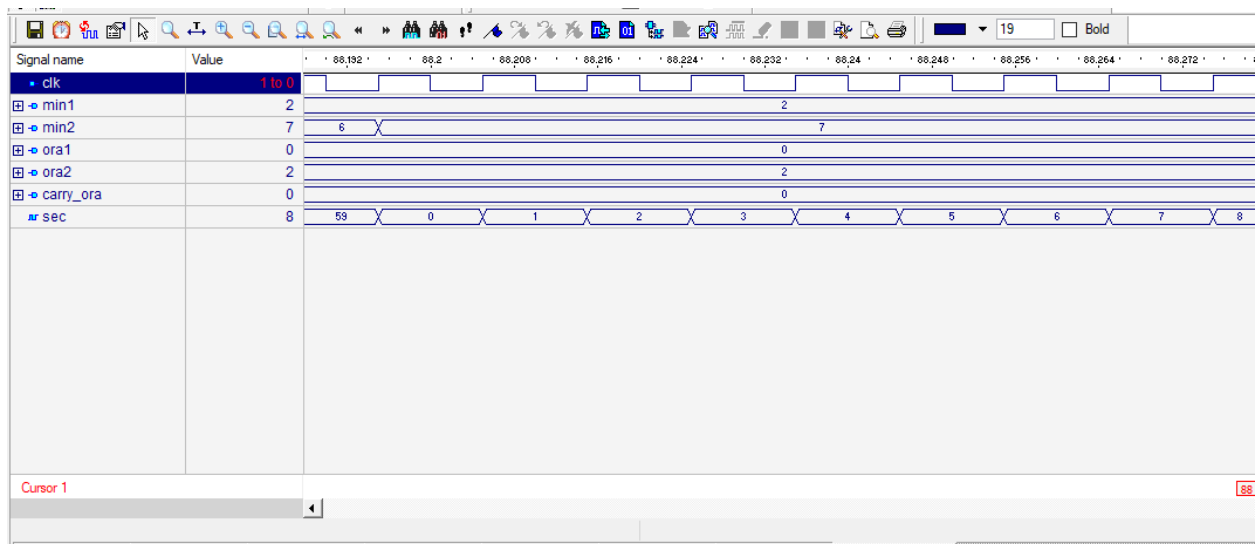
Generarea minutelor și a orelor

Am considerat drept o secundă un impuls de tact. Un exemplu pentru modul de functionare al acestui algoritm este urmatorul:

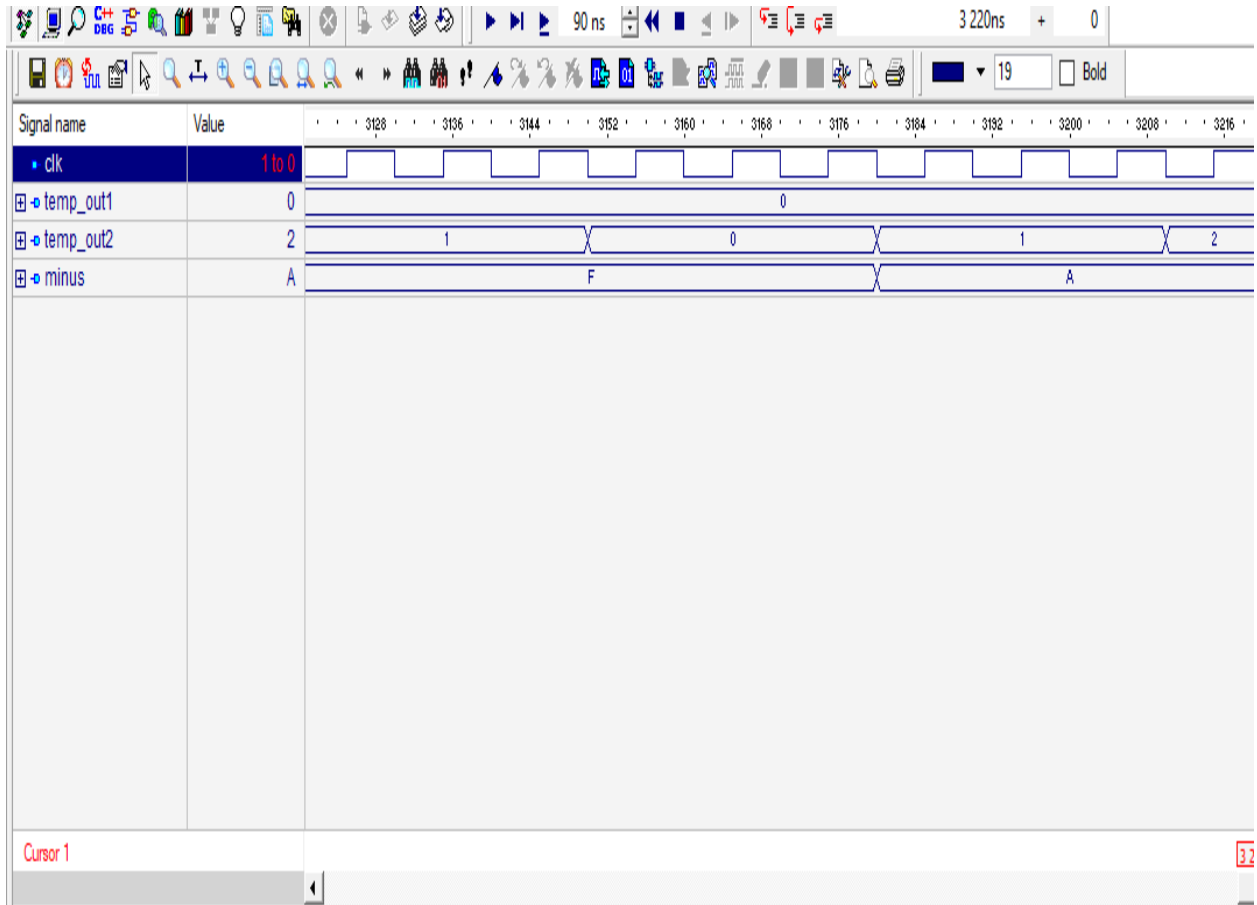
Primul exemplu cazul în care se incrementează valoarea unităților la minute



Al doilea exemplu prezintă și cazul în care s-a incrementat si valoarea unităților la oră.



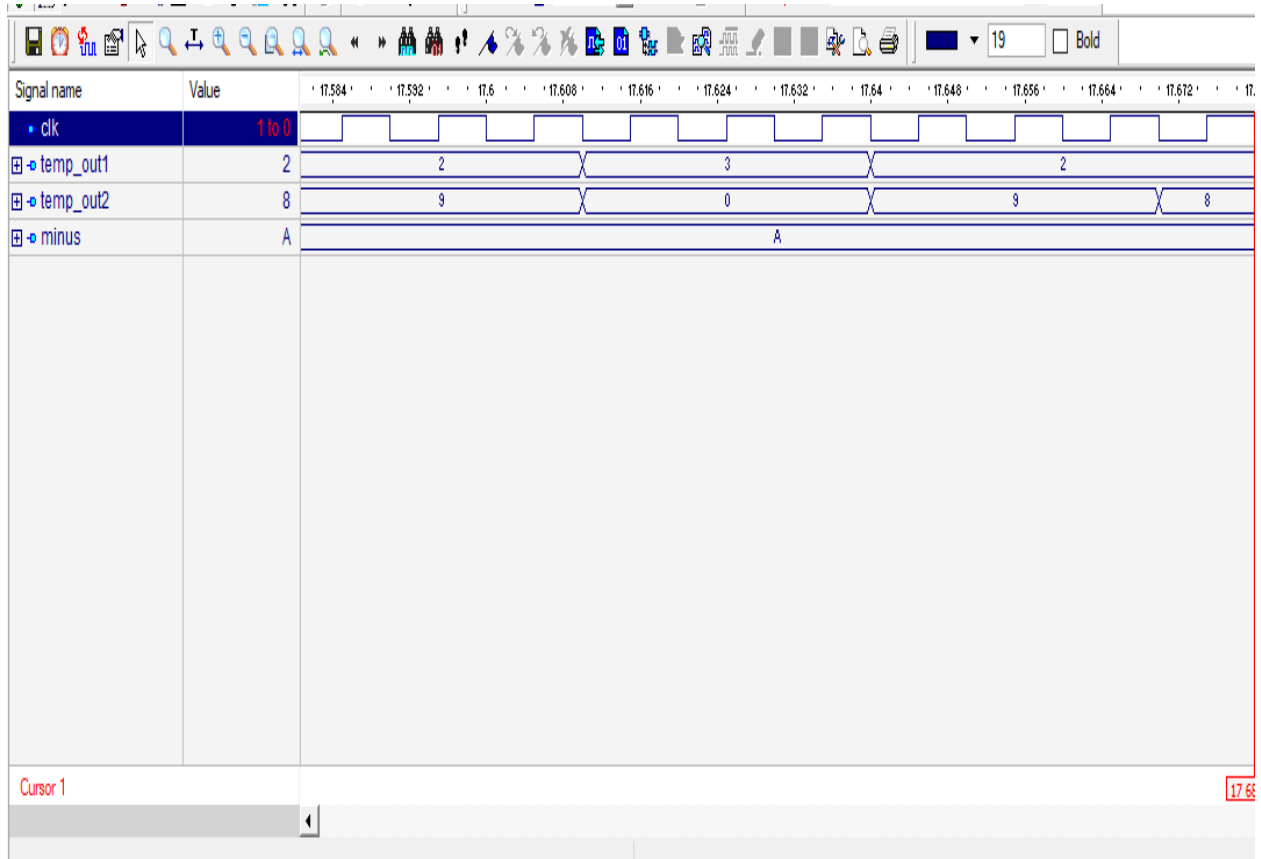
Generarea temperaturii



În imaginea de mai jos se observă trecerea de la un număr pozitiv(reprezentat de valoarea F la minus) într-un număr negative (reprezentat de valoarea A la minus).

În următoarea imagine este prezentată limita inferioară a termometrului(-30 de grade) în momentul în care se ajunge la aceasta valoare, temp_out2 primește

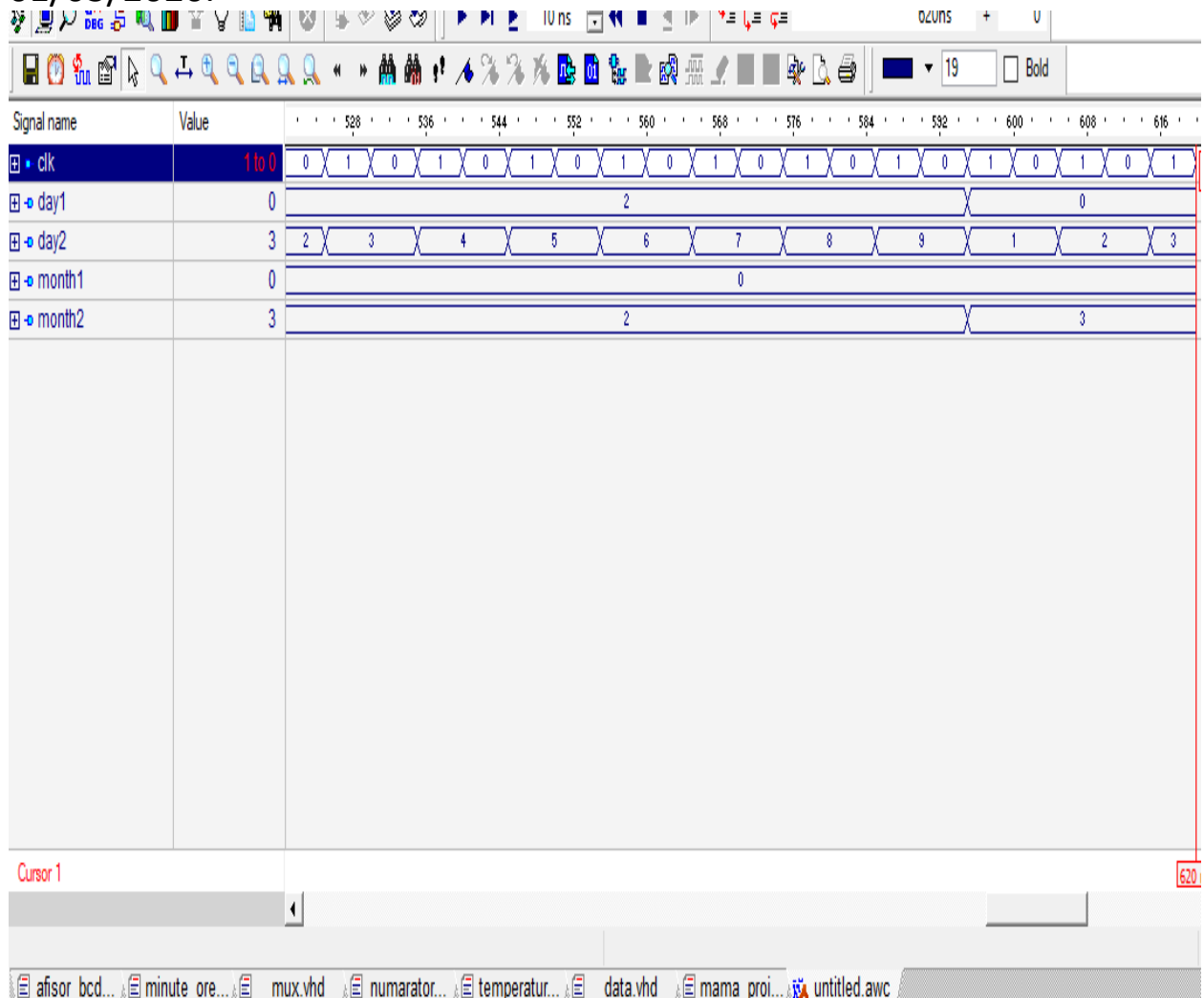
valoarea 9 și temp_out1 valoarea 2.



Algoritm pentru calculu datei

Clock-ul de la această component este reprezentat de carry-ul ce provine de la numararea orelor(carry_ora).

Exemplu pentru trecerea din data de 29/02/2020 in data de 01/03/2020.



Aici avem implementarea afişorului care tratează cazurile speciale pentru a afişa minus şi C în cazul temperaturii:

```
File Design Simulation Tools Window Help
[Icons] 10 ns [Buttons]
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity bcd is
5 port ( LED_BCD: in STD_LOGIC_vector(3 downto 0);
6       LED_out: out STD_LOGIC_vector(6 downto 0));
7 end entity;
8
9 architecture arch_afisaj of bcd is
10
11 begin
12 process(LED_BCD)
13 begin
14
15     case LED_BCD is
16     when "0000" => LED_out <= "0000001"; -- "0"
17     when "0001" => LED_out <= "1001111"; -- "1"
18     when "0010" => LED_out <= "0010010"; -- "2"
19     when "0011" => LED_out <= "0000110"; -- "3"
20     when "0100" => LED_out <= "1001100"; -- "4"
21     when "0101" => LED_out <= "0100100"; -- "5"
22     when "0110" => LED_out <= "0100000"; -- "6"
23     when "0111" => LED_out <= "0001111"; -- "7"
24     when "1000" => LED_out <= "0000000"; -- "8"
25     when "1001" => LED_out <= "0000100"; -- "9"
26     when "1010" => LED_out <= "1111110"; --afiseaza minus
27     when "1110" => LED_out <= "0110001"; --afiseaza C
28     when others => LED_out <= "1111111"; -- nu afiseaza nimic
29
30     end case;
31 end process;
32
33 end architecture;
```

Mai jos este prezentată implementarea multiplexorului 4 la 1 cu calea de intrare pe 4 biți.

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity mux4g is                                --multiplexoru 4 la 1 avand calea de intrare pe 4 biti
5
6      port (
7          a: in STD_LOGIC_VECTOR (3 downto 0);
8          b: in STD_LOGIC_VECTOR (3 downto 0);
9          c: in STD_LOGIC_VECTOR (3 downto 0);
10         d: in STD_LOGIC_VECTOR (3 downto 0);
11         sel: in STD_LOGIC_VECTOR (1 downto 0);
12         y: out STD_LOGIC_VECTOR (3 downto 0)
13     );
14 end mux4g;
15
16 architecture mux4g_arch of mux4g is
17 begin
18     process (sel, a, b, c, d)
19     begin
20         case sel is
21             when "00" => y <= a;
22             when "01" => y <= b;
23             when "10" => y <= c;
24             when others => y <= d; --valoarea d nu o folosim in algoritm sel avand valori doar intre 0 si 2
25         end case;
26     end process;
27 end mux4g_arch;
28
29
30
31

```

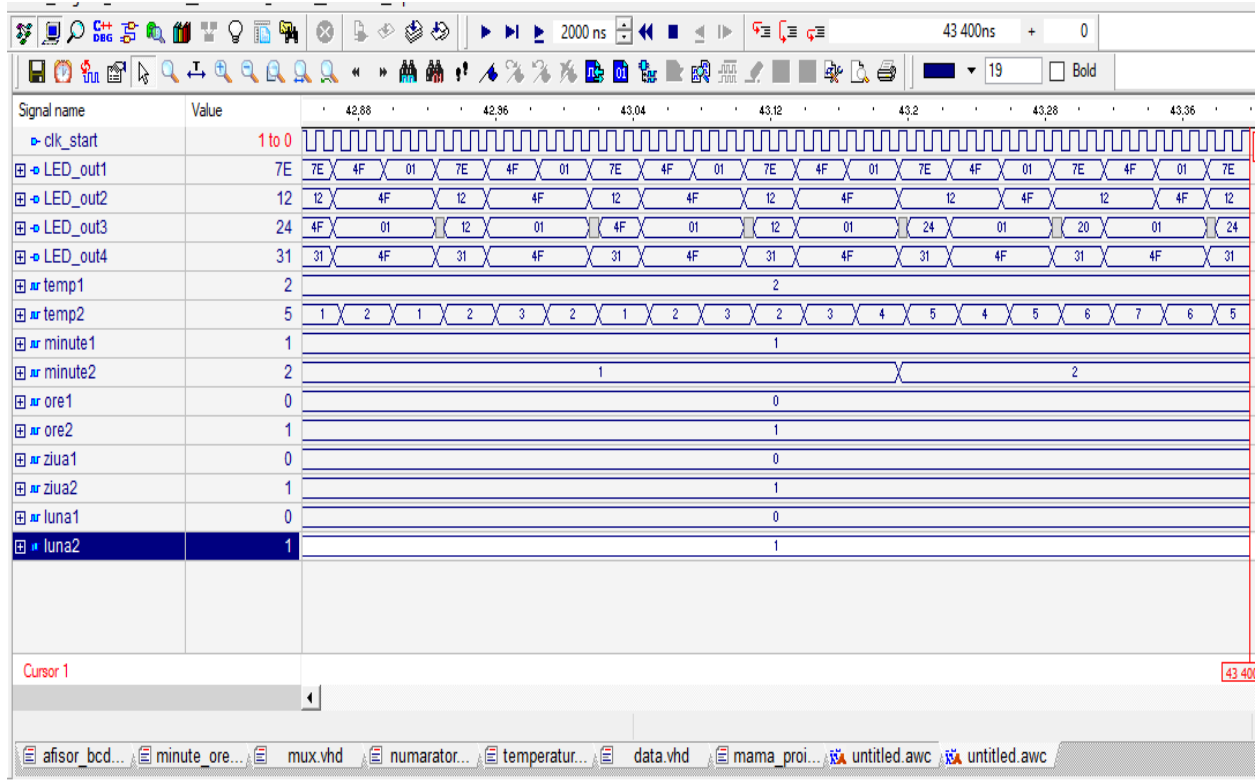
În final o sa prezint legarea tuturor componentelor:

```

75 signal iesire_mux4 : std_logic_vector(3 downto 0);
76 signal temp1 : std_logic_vector(3 downto 0);
77 signal temp2 : std_logic_vector(3 downto 0);
78 signal minuss : std_logic_vector(3 downto 0);
79 signal minutel1: std_logic_vector(3 downto 0);
80 signal minute2: std_logic_vector(3 downto 0);
81 signal ore1: std_logic_vector(3 downto 0);
82 signal ore2: std_logic_vector(3 downto 0);
83 signal carry_ore : std_logic_vector(0 downto 0);
84 signal ziua1: std_logic_vector(3 downto 0);
85 signal ziua2: std_logic_vector(3 downto 0);
86 signal luna1: std_logic_vector(3 downto 0);
87 signal luna2: std_logic_vector(3 downto 0);
88 --aici declar semnale
89 begin
90
91     nrl: numarator port map(clk=> clk_start,carry_out=>carry_nrl );
92     nr2 : numarator port map(clk=> carry_nrl, q=> iesire_nr2);
93     temperatura: rand_gen port map (clk=> clk_start,temp_out1=> temp1,temp_out2 => temp2, minus=>minuss);
94     minute: minutes port map (clk=> clk_start, min1=> minutel1, min2=>minute2, oral=>ore1, ora2=> ore2, carry_ora=> carry_ore );
95     zile: data port map(clk=>carry_ore, day1=>ziua1, day2=>ziua2, month1=>luna1, month2=>luna2 );
96     mux1: mux4g port map(a=> minuss, b=> minutel1, c=> ziua1, sel=> iesire_nr2, d=>"0000", y=>iesire_mux1); --trebuie sa pun minus 1
97     mux2: mux4g port map(a=>temp1, b=> minute2, c=> ziua2, sel=> iesire_nr2, d=> "0000" , y=> iesire_mux2);
98     mux3: mux4g port map(a=>temp2, b=> ore1, c=>luna1, sel=> iesire_nr2, d=> "0000" , y=> iesire_mux3);
99     mux4: mux4g port map(a=>"1110", b=> ore2, c=>luna2, sel=> iesire_nr2, d=> "0000" , y=> iesire_mux4);
100
101     afisor1: bcd port map( LED_bcd => iesire_mux1, LED_out => LED_out1 );
102     afisor2: bcd port map( LED_bcd => iesire_mux2, LED_out => LED_out2);
103     afisor3: bcd port map( LED_bcd => iesire_mux3, LED_out => LED_out3);
104     afisor4: bcd port map( LED_bcd => iesire_mux4, LED_out => LED_out4);
105
106     --port mapuri

```

Simularea finală



Ex: data este 01/01, ora 1 și 21 de minute , temperatura este de -25 grade Celsius.

5. Justificarea soluției alese

Am ales această metodă de implementare pentru că simulează realitatea mai bine decât alte metode considerate anterior. Consider că acest termostat ar putea funcționa ca un prototip pentru unul real întrucât îndeplinește nevoile unui utilizator de rând.

6. Utilizare

Pentru ca programul să funcționeze, trebuie ca utilizatorul să dea un impuls de tact (un clock).

Am considerat ca un clock să fie echivalent cu o secundă. La 60 de clock-uri, contorul pentru secunde revine în 0. În schimb, cel pentru minute va crește. Calculul orei și datei funcționează pe același principiu.

7. Posibilități de dezvoltare ulterioară

Consider că acest automa se poate îmbunătăți prin adăugarea unor butoane care să permită utilizatorului să introducă data și timpul dorit. Pe lângă acestea consider că se poate adăuga și afișarea anului și semnul pentru grade.