

# AGENDĂ DE BIROU PROGRAMABILĂ

STUDENT: TRIF GABRIELA- IOANA

# Cuprins

Capitolul I. Proiectare .....	3
Specificația proiectului. Tema .....	3
Schema bloc .....	3
Unitatea de control și de execuție .....	4
Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE .....	5
Determinarea resurselor (UE) .....	5
Schema bloc a primei descompuneri .....	8
Capitolul II. Justificarea soluției alese .....	8
Capitolul III. Manual de întreținere și utilizare .....	9
Capitolul IV. Posibilități de dezvoltări ulterioare .....	10
Capitolul V. Bibliografie .....	11

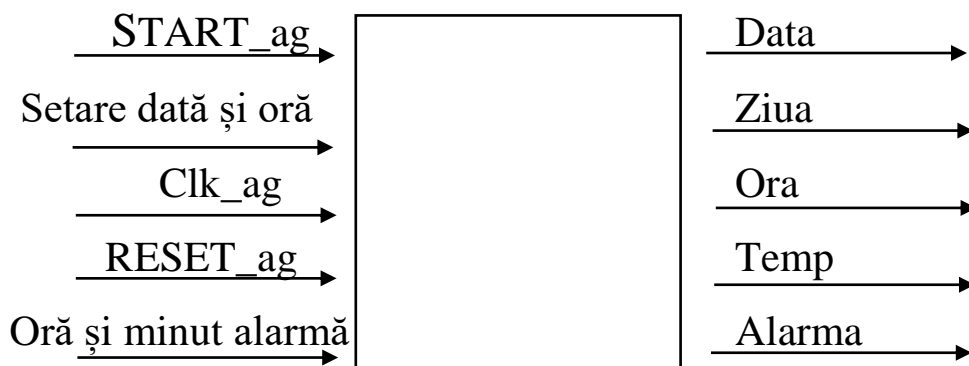
# Capitolul I. Proiectare

## Specificația proiectului. Tema

Să se proiecteze o agenda de birou programabilă care să afișeze:

- Anul, luna, ziua: cu cifre;
- Ziua: cu litere;
- Ora și minutul: cu cifre;
- Temperatura ambiantă în grade Celsius;
- Agenda va fi prevăzută și cu o alarmă sonoră asociată orei și minutului.

## Schema bloc



### Intrări:

- **START\_ag** – intrarea se activează atunci când am terminat de introdus data și ora.
- Setare dată și oră cuprinde intrările
  - **Min\_initial**- reprezintă minutul de început
  - **Ora\_initial**- reprezintă ora de început
  - **Ziua\_initial**- reprezintă ziua de început
  - **Luna\_initial**- reprezintă luna de început
  - **An\_initial**- reprezintă anul de început

Toate aceste intrări se introduc în baza 10.

- **Clk\_ag**- intrarea pentru semnalul de tact.

- **RESET\_ag**- intrare pentru resetarea întregului circuit.
- Oră și minut alarmă cuprinde intrările
  - **Ora\_alarma**
  - **Min\_alarma**

Aceste intrări setează timpul pentru alarmă. Se introduc în baza 10.

## Ieșiri:

- Data cuprinde iesirile:
  - **Ziua\_out**
  - **Luna\_out**
  - **An\_out**

Care sunt reprezentate în baza 10.

- **Ziua** reprezintă ziua din săptămână care corespunde datei curente. Iesirea este reprezentă dintr-un string.
- Ora cuprinde iesirile:
  - **Min\_out**
  - **Ora\_out**

Care sunt reprezentare în baza 10.

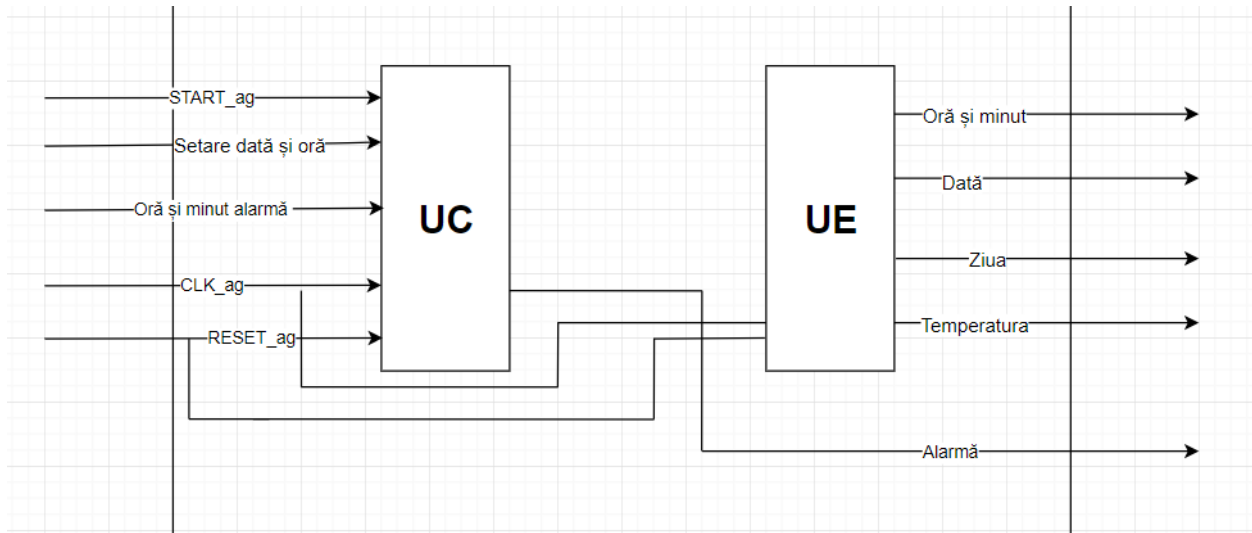
- **Temp** reprezintă temperatura ambiantă în grade Celsius.
- **Alarma** se va activa în momentul în care ora\_out va fi egală cu ora\_alarma , respectiv min\_out va fi egal min\_alarma.

## Unitatea de control și de execuție

Cutia neagră a sistemului trebuie descompusă mai departe pentru a putea găsi componente implementabile. Vom face o descompunere top-down a problemei pana cand ajungem la circuite cunoscute, iar apoi vom implementa bottom-up.

Prima descompunere a oricărui sistem este una în care vom diferența între logica de control din sistem și resursele sistemului. Logica de control este reprezentată de Unitatea de Control iar resursele sunt reprezentate de Unitatea de Execuție. Orice algoritm poate fi descompus în acest fel (reprezentarea abstractă a unui algoritm se face printr-un flow-chart).

# Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE



## Determinarea resurselor (UE)

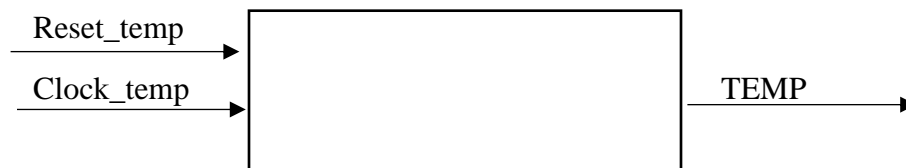
### a. Numărător pe 6 biți

Acesta este un numărător pe 6 biți modulo 59, folosit pentru contorizarea secundelor și a minutelor. Va avea 2 ieșiri, prima va fi un semnal de carry folosit pentru marcarea trecerii celor 60 de secunde/minute, iar a doua ieșire va fi chiar numărul secundelor/minutelor, reprezentat printr-un integer.



### b. Generator de numere pseudo-aleatoare

Această resursă este folosită pentru determinarea temperaturii. Are ca intrări un semnal de clock și un semnal de reset. Va determina temperatura ambientală cuprinsă între 0 grade Celsius și 31 de grade Celsius.



c. *Numărător pe 5 biți*

Acesta este un numărător pe 5 biți modulo 23, folosit pentru contorizarea orelor. Va avea 2 ieșiri, prima va fi un semnal de carry folosit pentru marcarea trecerii celor 23 de ore, iar a doua ieșire va fi chiar numărul orelor, reprezentat printr-un integer.



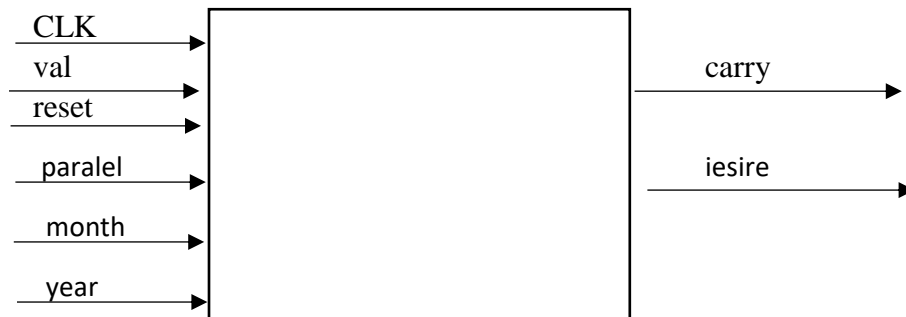
d. *Algoritm pentru determinarea numărului de zile dintr-o lună*

Acest algoritm are ca intrări month\_in și c\_year, ambele fiind numere întregi care reprezintă luna și anul curent. Singura ieșire a acestei resurse este days\_nr(integer).



e. *Numărător modulo p*

Acest numărător este folosit pentru contorizarea zilelor dintr-o lună. Este pe 5 biți și poate număra până la 28, 29, 30 sau 31 în funcție de semnalul days\_nr determinat de resursa de mai sus.



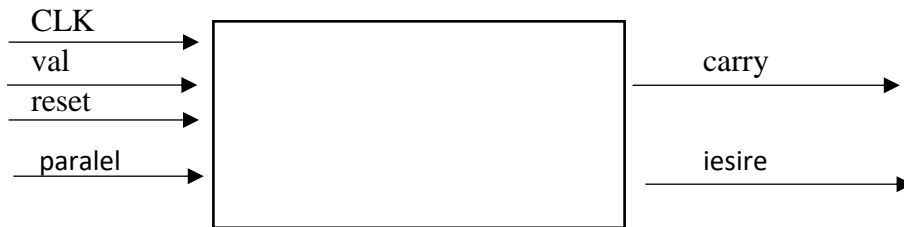
*f. Numărător pe 4 biți*

Acesta este un numărător pe 4 biți modulo 11, folosit pentru contorizarea lunilor. Va avea 2 ieșiri, prima va fi un semnal de carry folosit pentru marcarea trecerii celor 11 de ore, iar a doua ieșire va fi chiar numărul lunilor, reprezentat printr-un integer.



*g. Numărător pe 12 biți*

Acesta este un numărător pe 12 biți, folosit pentru contorizarea anilor. Va avea 2 ieșiri, prima va fi un semnal de carry folosit pentru marcarea trecerii unui an, iar a doua ieșire va fi chiar anul curent, reprezentat printr-un integer.



*h. Comparator*

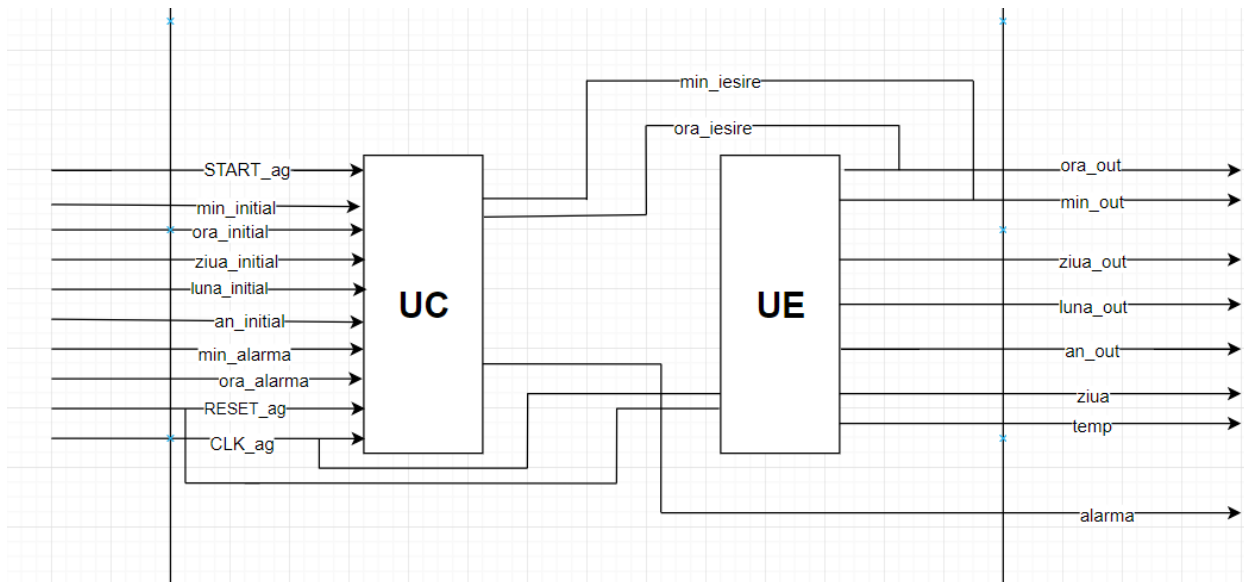
Această resursă are ca intrări ora și minutul la care este setată alarma și ora împreună cu minutul curente. Are ca ieșire un semnal care se activează atunci când condiția este adevărată.



*i. Algoritmul lui Sakamoto*

Acest algoritm are ca intrări data ziua, luna și anul, iar ca ieșire are un string în care se va memora ziua din săptămână corespunzătoare intrărilor.

## Schema bloc a primei descompuneri



## Capitolul II. Justificarea soluției alese

În vederea realizării proiectului, am utilizat o metodă mai simplă din punctul meu de vedere, adică împărțirea proiectului în componente. Am implementat fiecare componentă, apoi am creat un alt fișier în care am unit toate componentele folosind o arhitectură structurală.


Implementarea acestui proiect este ușor de explicat chiar și unui utilizator care nu are suficiente cunoștințe despre limbajul de programare VHDL.


Pe de altă parte, soluția aleasă de mine pentru realizarea proiectului este ușor de urmărit în cazul unei probleme tehnice sau a unei erori deoarece se poate determina rapid sursa problemei.




# Capitolul III. Manual de întreținere și utilizare

1. Se deschide programul Active-HDL 6.3

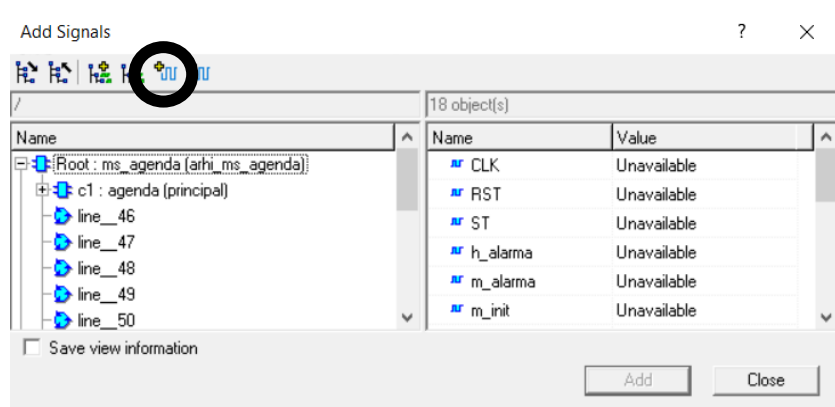
2. Se compilează cu ajutorul comenzii “Compile” 

3. . Se simulează cu “New waveform”  :

a. Se face click dreapta pe fereastra nou deschisă.

b. Se selectează “Add signals...”  **Ctrl+I**

c. Adăugam din listă toate intrările, ieșirile si semnalele intermediare dând click pe butonul “SELECT ALL SIGNALS”;



d.După ce toate acestea au fost alese se apasă butonul „F5” de la tastatură.

e.Se urmăresc rezultatele obținute.

# Capitolul IV. Posibilități de dezvoltări ulterioare

Cum oricărui dispozitiv i se pot aduce îmbunătățiri, la fel se poate face și în cazul agendei de birou programabile. Spre exemplu, ar fi util să se poate programa mai multe alarme care să poată fi setate și după dată, nu doar după oră și să li se asocieze fiecareia un sunet individual. O altă îmbunătățire ar putea fi adusă modului de afișare a temperaturii astfel încât să se poată alege unitatea de măsură în care să fie afișată temperatura ambiantă.

Ca ultimă îmbunătățire adusă proiectului ar putea fi preluarea automată a datei și a orei curente, fără a fi nevoie utilizatorul să le introducă.

# Capitolul V. Bibliografie

- Materiale de curs „Proiectarea sistemelor numerice”;
- Îndrumător de laborator „Proiectarea sistemelor numerice”;
- [https://en.wikipedia.org/wiki/Determination\\_of\\_the\\_day\\_of\\_the\\_week](https://en.wikipedia.org/wiki/Determination_of_the_day_of_the_week)
- <https://semiwiki.com/fpga/6129-pseudo-random-generator-tutorial-in-vhdl-part-1-3/>
- <https://stackoverflow.com/questions/18935639/make-calendar-which-shows-month-number-and-days-of-month-in-vhdl>