AGENDĂ DE BIROU PROGRAMABILĂ

proiect B6

Proiect realizat de:

Irimie Daria-Alexia

CUPRINS

[1. Introducere 3](#_Toc72455955)

[2. Proiectarea agendei 3](#_Toc72455956)

[2.1. Schema Bloc 3](#_Toc72455957)

[2.1.1 Intrări și ieșiri 3](#_Toc72455960)

[2.1.2. Black Box 5](#_Toc72455961)

[2.2. Descompunerea în UC și UE 6](#_Toc72455961)

[2.3 Resurse 7](#_Toc72455962)

[2.3.1 Lista completă a resurselor 7](#_Toc72455963)

[2.3.2  [Descrierea detaliată a resurselor 7](#_Toc72455963)](#_Toc72455964)

[2.4. Organigrama UC 10](#_Toc72455965)

[3. Justificarea soluției alese 11](#_Toc72455966)

[4. Posibilităţi de dezvoltare ulterioare 11](#_Toc72455967)

[5. Surse 11](#_Toc72455968)

1. INTRODUCERE

Să se proiecteze o agendă 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 alarmă sonoră asociată orei și minutului.

2. PROIECTARE

2.1 SCHEMA BLOC

2.1.1 INTRĂRI ȘI IEȘIRI

Un prim pas este stabilirea intrărilor și ieșirilor de care avem nevoie pentru a putea implementa functionalitatea agendei.

INTRĂRI :

• START\_STOP

• AN\_UNIT\_IN

• AN\_ZECI\_IN

• LUNA\_UNIT\_IN

• LUNA\_ZECI\_IN

• ZIUA\_UNIT\_IN

• ZIUA\_ZECI\_IN

Acestea vor fi afișate sub forma : ZZ / LL / AAAA

• ZI\_DIN\_SAPT\_IN

Pentru aceasta vom avea zilele scrise sub forma :

MO, TU, WE, TH, FR, SA, SU

• ORA\_UNIT\_IN

• ORA\_ZECI\_IN

• MINUTE\_UNIT\_IN

• MINUTE\_ZECI\_IN

• ALARMA\_ORA\_UNIT\_IN

• ALARMA\_ORA\_ZECI\_IN

• ALARMA\_MIN\_UNIT\_IN

• ALARMA\_MIN\_ZECI\_IN

• ALARMA\_ON\_OFF\_IN

• CLOCK\_IN

• RESET\_IN

• LOAD\_IN

IEȘIRI:

• AN\_UNIT\_OUT

• AN\_ZECI\_OUT

• LUNA\_UNIT\_OUT

• LUNA\_ZECI\_OUT

• ZIUA\_UNIT\_OUT

• ZIUA\_ZECI\_OUT

• ZI\_LIT\_1\_OUT

• ZI\_LIT\_2\_OUT

• ORA\_UNIT\_OUT

• ORA\_ZECI\_OUT

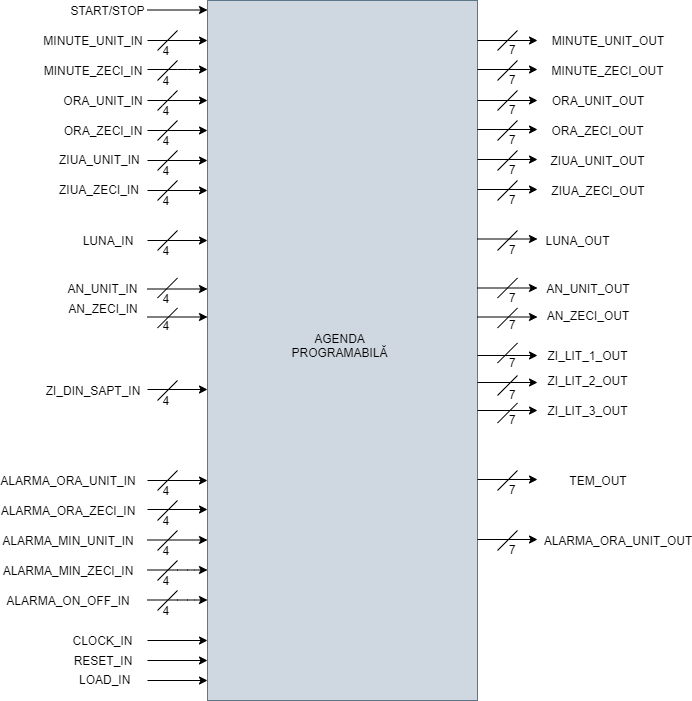
• MINUTE\_UNIT\_OUT

• MINUTE\_ZECI\_OUT

• TEMP\_OUT

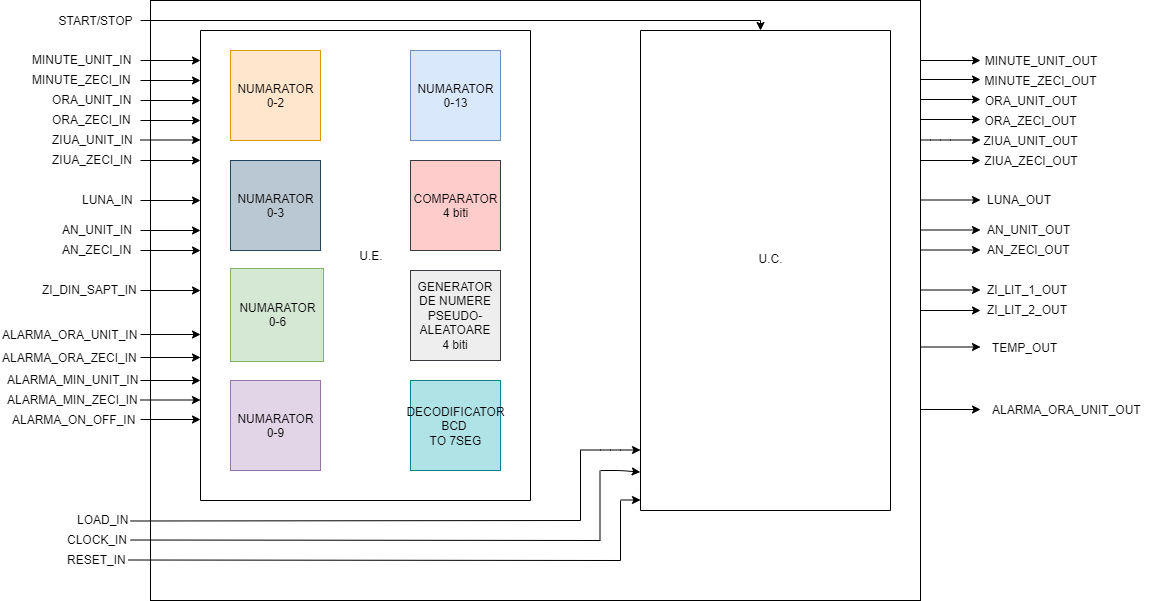
• ALARMA\_ORA\_UNIT\_OUT

2.1.2 BLACK BOX



2.2 UNITATEA DE CONTROL ȘI UNITATEA DE EXECUȚIE

Descompunem cutia neagră în logica de control a sistemului (reprezentată de Unitatea de Control) și resursele sistemului ( reprezentate de Unitatea de Execuție), pentru a putea găsi componente implementabile.



2.3 RESURSE

2.3.1 LISTA COMPLETĂ A RESURSELOR

În vederea realizării proiectului, am folosit următoarele resurse :

• un numărător care numără în bucla 0-2 (de la 0000 la 0010)

• un numărător care numără în bucla 0-3 (de la 0000 la 0011)

• un numărător care numără în bucla 0-6 (de la 0000 la 0110)

• 5 numărătoare care numără în bucla 0-9 (de la 0000 la 1001)

• 9 decodificatoare BCD to 7-segment pentru afisarea ceasului

(se primește un cod pe 4 biți, iar output-ul va fi pe 7 biți corespunzători segmentelor unui afișor)

• 2 decodificatoare BCD to 7-segment pentru afisarea zilei din saptămână cu litere

• un decodificatoar BCD to 7-segment pentru afisarea temperaturii

2.3.2 DESCRIEREA DETALIATĂ A RESURSELOR

* ***NUMARATOARE***

Pentru toate numaratoarele folosite, s-a utilizat o descriere comportamentală. Printr-o condiție care verifică dacă s-a ajuns la valoarea dorită, se numără doar într-o anumită buclă (ex : 0-3, 0-6).

* ***DECODIFICATOR BCD to 7 SEG***

Ieșirile pe 4 biți din numărătoarele avansate sunt cele care intră în decodificatorul nostru, care primește un cod pe 4 biți și generează alt cod pe 7 biți corespunzător pentru fiecare segment care va fi iluminat în funcție de cifra dorită.

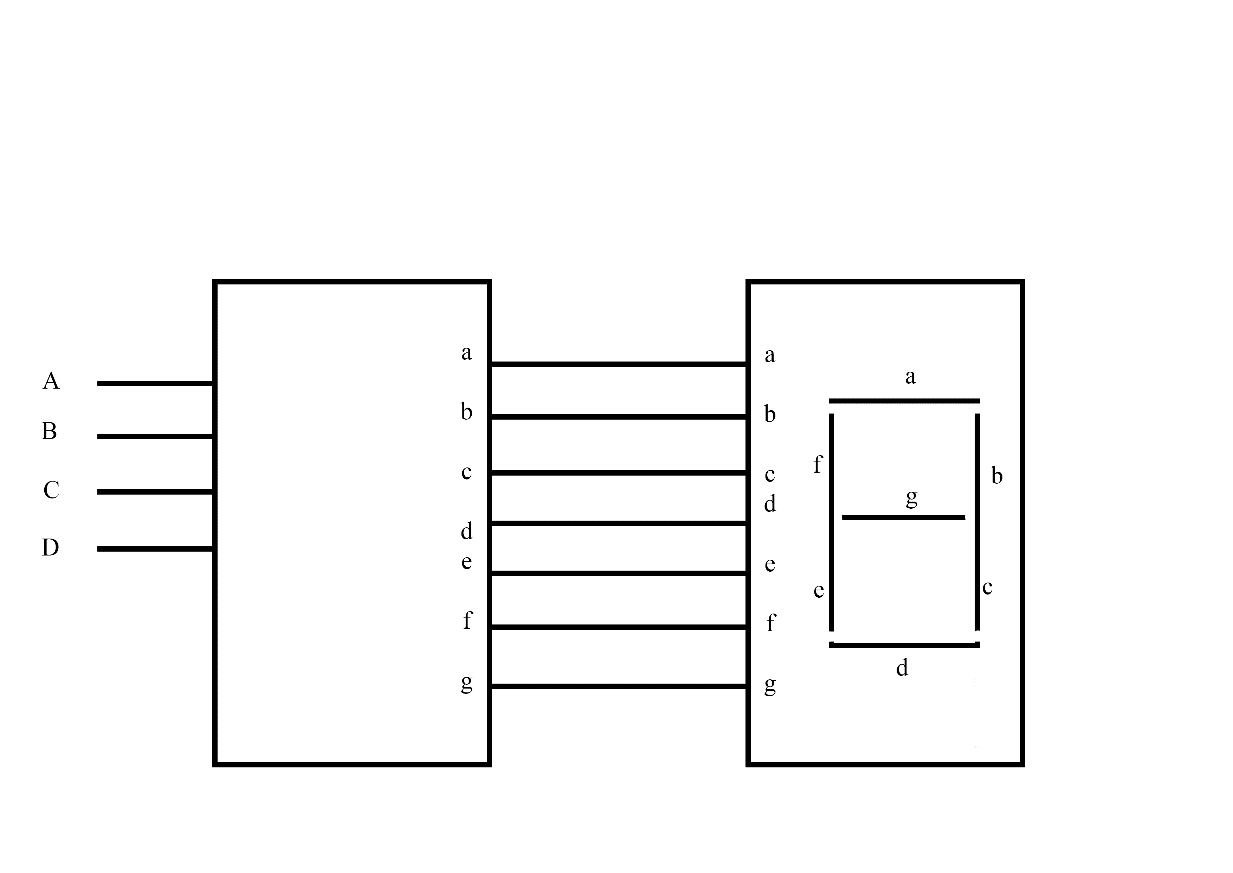
Într-un afișor 7 segment, fiecărui segment îi corespunde o literă de la a la g. Ca un segment să fie activ, vom aplica un semnal cu valoarea logică de ‘1’.

De ce 7 segmente pentru a reprezenta o cifră? Este numărul minim de segmente pentru a fi posibil reprezentarea cifrelor din intervalul 0-9. Evident, există afișoare și mai complexe cu mai multe segmente, unele fiind capabile sa afișeze și litere.

Tabelul de adevăr pentru decodificatorul nostru este:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cifra | A | B | C | D | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Schema logică pentru cum ar funcționa decodificatorul și afișorul 7 bcd ar fi:



* ***COMPARATOR 4 BITI***

Pentru implementarea alarmei am ultilizat 4 comparatoare pe 4 biți.

Acestea verifică dacă s-a ajuns la ora setata pentru alarmă.

Primul comparator – Verifică minutul unități setat pentru alarmă cu minutul unități al ceasului în momentul curent

Al doilea comparator – Verifică minutul zeci setat pentru alarmă cu minutul zeci al ceasului în momentul curent

Al treilea comparator – Verifică ora unități setată pentru alarmă cu ora unități a ceasului în momentul curent

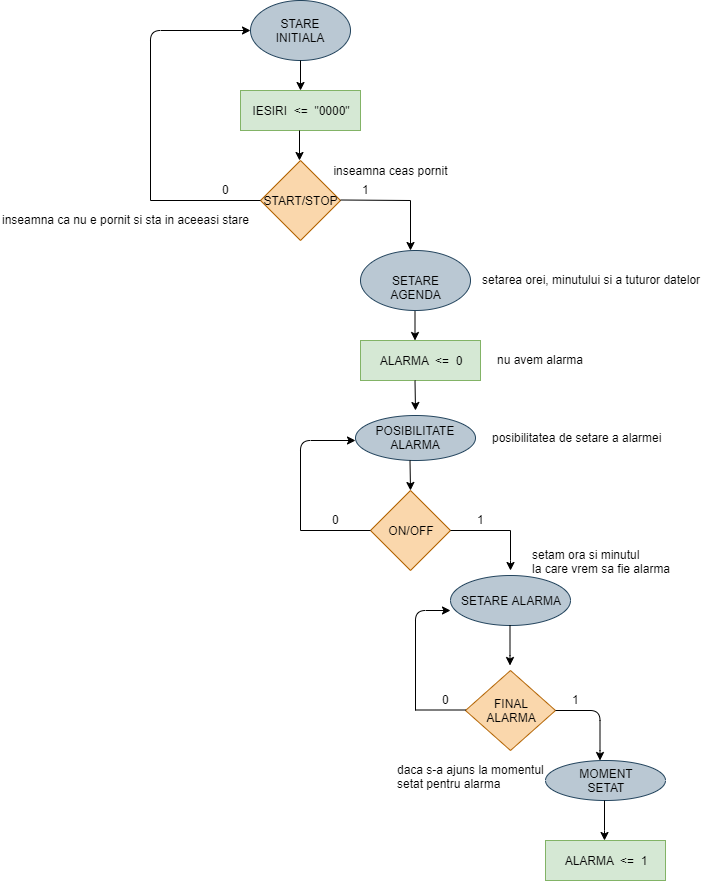
Al patrulea comparator – Verifică ora zeci setată pentru alarmă cu ora zeci a ceasului în momentul curent

* ***GENERATOR DE NUMERE PSEUDOALEATOARE PE 4 BIȚI***

Pentru implementarea temperaturii am utilizat un generator de numere pseudoaleatoare pe 4 biți, care ne generează o temperatură aleatoare.

Aceasta urmează să fie afișată cu ajutorul unui decodificator BCD to 7 SEG.

2.4 ORGANIGRAMA UC



Unde avem:

START/STOP – buton pentru pornire/oprire ceas

ON/OFF – buton pentru pornire/oprire alarmă

FINAL\_ALARMĂ – s-a ajuns la momentul setat pentru alarmă

Stări:

STARE\_INIȚIALĂ – Reset, inițializare

SETARE\_AGENDĂ – setarea orei, minutului și toate celelalte date

POSIBILITATE\_ALARMĂ – posibilitatea de a seta o alarmă

SETARE\_ALARMĂ – setare ora și minut pentru alarmă

MOMENT\_ALARMĂ – dacă s-a ajuns la momentul setat pentru alarmă

3. JUSTIFICAREA SOLUȚIEI ALESE

Am ales o metodă intuitivă de rezolvare a proiectului cu agenda programabilă. Din start am decis să folosesc un numărător pentru fiecare cifră, astfel să îmi fie ușor să incrementez/decrementez fiecare cifră care urma să facă parte din afișare.

Restul componentelor erau esențiale(decodificator, comparator, generator de numere pseudoaleatoare) și nu prea prezentau altă alternativă.

4. POSIBILITĂȚI DE ÎMBUNĂTĂȚIRE

Se pot aduce îmbunătățiri în primul rând funcționalităților implementate, cum ar fi setarea unor alarme doar în anumite zile, cu memorarea acestora. Mai mult decât atât, se pot aduce îmbunătățiri agendei în sine, prin adăugarea unor funcționalități cum ar fi memorarea unor zile, pe care dorim să le evidențiem datorită unor evenimente.

5. SURSE

- <https://allaboutfpga.com/bcd-to-7-segment-decoder-vhdl-code/>

- https://ro.wikipedia.org/wiki/Afi%C8%99aj\_cu\_%C8%99apte\_segmente