TIMER

DATA: 27.05.2023

Denisa Elena Ghiorghioiu

Îndrumător: ing. DIana Pop

Contents

[1 Specificații 2](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249632)

[2 Proiectare 3](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249633)

[2.1 Schema Bloc 3](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249634)

[2.2 Unitatea de Control și Unitatea de Execuție 3](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249635)

[1.2.1 Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE. 3](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249636)

[1.2.2 Determinarea resurselor (UE) 4](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249637)

[2.2.3 Schema bloc a primei descompuneri 6](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249640)

[2.2.4 Reprezentarea UC prin diagrama de stări (organigrama) 6](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249641)

[2.2.5 Schema de detaliu a proiectului 9](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249642)

[3 Manual de utilizare și întreținere 9](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249643)

[4 Justificarea soluției alese 9](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249644)

[5 Posibilități de dezvoltări ulterioare 9](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249645)

[6 Bibliografie 9](file:///C:\Users\Denisa\Downloads\Exemplu%20proiect%20PSN.docx#_Toc130249646)

TIMER

# Specificații

Să se proiecteze un timer cu următoarea funcţionalitate: dispozitivul are 4 afişaje BCD - 7 segmente. Primele două afişaje sunt pentru minute, următoarele două pentru secunde. Astfel, valoarea maximă care poate fi afişată este de 99 minute şi 59 secunde.

Dispozitivul are 3 butoane: M (de la Minute), S (de la Secunde) şi START / STOP.

Presupunând că iniţial este în starea ZERO, dacă se apasă butonul START / STOP, timerul începe să numere crescător. Dacă se apasă din nou butonul START / STOP, timerul se opreşte la valoarea atinsă în momentul respectiv. Dacă se apasă din nou butonul START / STOP, timerul continuă să numere etc. Dacă ajunge la 99 de minute şi 59 de secunde, urmează din nou ZERO. Dacă se apasă simultan butoanele M (de la Minute) şi S (de la Secunde), timerul se resetează (devine ZERO).

În orice stare, dacă se apasă butonul M, se va incrementa şi afişa valoarea minutelor. În orice stare, dacă se apasă butonul S, se va incrementa şi afişa valoarea secundelor. O dată ce s-a setat o valoare pentru minute şi / sau secunde (prin apăsarea butoanelor M sau S), când se apasă butonul START / STOP, timerul începe să numere descrescător de la valoarea curentă „Minute / Secunde” până la ZERO, iar când se ajunge în starea ZERO se emite un semnal sonor (alarmă) pentru o perioadă de timp care poate fi setată iniţial.

# Proiectare

## Schema Bloc

Secunda01

START/STOP

Secunda 02

M(min)

Minut01

S(sec)

TIMER

Minut02

RST

Semnal sonor/alarma

Figura 1 Cutia neagra a sistemului cu intrările și ieșirile stabilite

|  |  |  |
| --- | --- | --- |
| INTRĂRI | START/STOP | 1 buton, care începe numărătoarea |
| M | 1 buton, care determină minutele |
| S | 1 buton, care determină secundele |
|  | RST | Resetează secundele+minutele |

|  |  |  |
| --- | --- | --- |
| IEȘIRI | Secunda01 | Afișare secunde, 7 biți |
| Secunda02 | Afișare secunde, 7 biți |
| Minut01 | Afișare minute, 7 biți |
| Minut02 | Afișare minute, 7 biți |
| Alarma | Se activează când se ajunge în starea ZERO |

|  |  |  |  |
| --- | --- | --- | --- |
| START/STOP | S | M | Starea timerului |
| 0 | 0 | 0 | Timer oprit |
| 0 | 0 | 1 | Incrementare minute |
| 0 | 1 | 0 | Incrementare secunde |
| 0 | 1 | 1 | Se resetează timer ul |
| 1 | 0 | 0 | Timer numara crescator |
| 1 | 0 | 1 | Timer numara descrescator |
| 1 | 1 | 0 | Timer numara descrescator |
| 1 | 1 | 1 | Timer oprit |

## Unitatea de Control și Unitatea 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 diferenția 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*).

### 1.2.1 Maparea intrărilor și ieșirilor cutiei mari pe cele două componente

A diagram of a block diagram

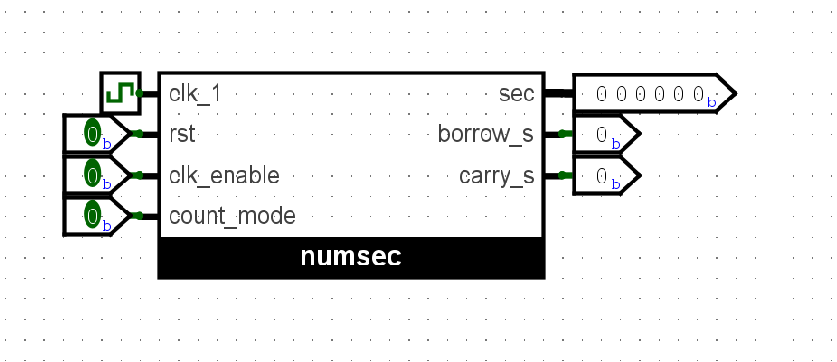
Description automatically generated with medium confidence

Figura 2Maparea intrărilor și ieșisilor cutiei negre pe intrările și ieșirile componentelor din prima descompunere

### 1.2.2 Determinarea resurselor (UE)

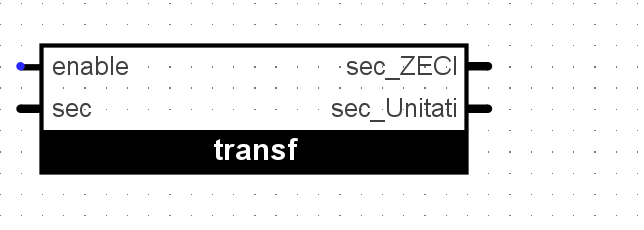
Pentru a stabili mai departe legăturile dintre UC și UE trebuie mai întâi **să identificăm resursele pe baza cărora luăm decizii sau resursele care devin informații pentru utilizator.** Aceste resurse pot sa genereze semnale către unitatea de control și pot fi controlate de UC prin semnale de Enable sau Reset.

**1.Numărător secunde**



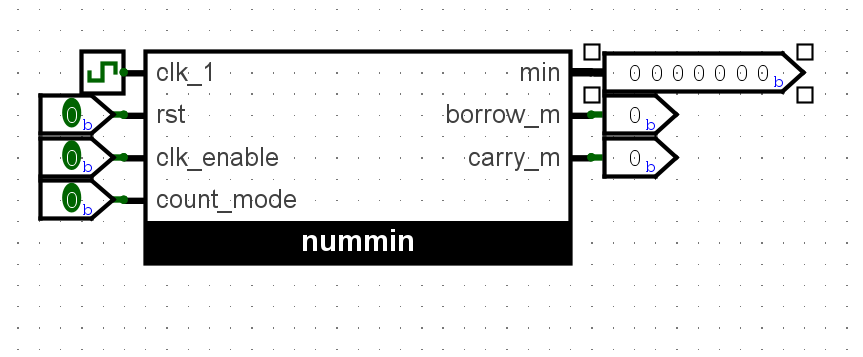
Numărătorul pentru secunde conține: un clk cu frecvența de 1Hz ptr a putea număra secundele în timp real, rst care resetează numărătorul, clk\_enable activ pe 1 adică numărătorul continuă să numere doar dacă clk\_enable are valoarea 1, și count\_mode care reprezintă modul în care numărătorul numără( crescător sau descrescător). Acest numărător numără de la 0-59, iar când ajunge la 59 acesta începe să numere din nou de la 0.

**2.Tranformator secunde din 6 biti in 4 biti pentru unități și 4 biți pentru zeci**



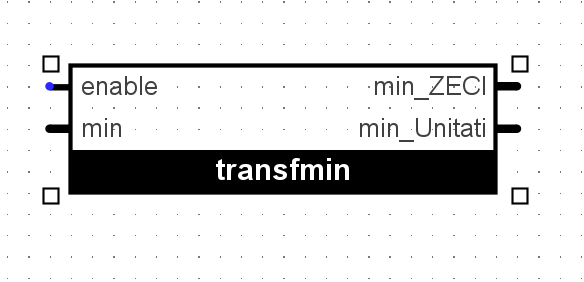
Acesta are ca intrare un număr pe 6 biti(secundele), iar daca enable are valoarea 1, se va transforma valoarea de pe 6 biti în 2 valori pe 4 biti respectiv in sec\_ZECI si sec\_UNITATI.

**3.Numărător minute**



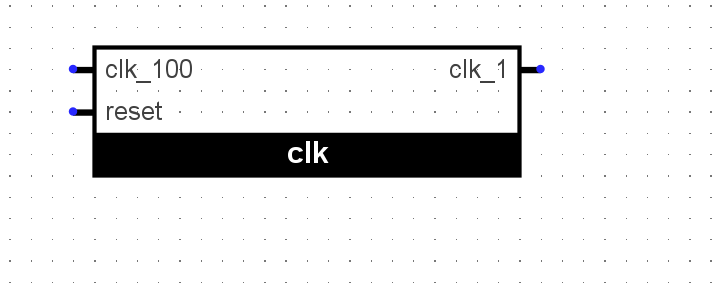
Numărătorul pentru minute conține: un clk cu frecvența de 1Hz ptr a putea număra minutele în timp real, rst care resetează numărătorul, clk\_enable activ pe 1 adică numărătorul continuă să numere doar dacă clk\_enable are valoarea 1, și count\_mode care reprezintă modul în care numărătorul numără( crescător sau descrescător). Acest numărător numără de la 0-99, iar când ajunge la 99 acesta începe să numere din nou de la 0.

**4.Transformator minute din 7 biti in 4 biti pentru unitati si 4 biti pentru secunde**



Acesta are ca intrare un număr pe 7 biti(minutele), iar daca enable are valoarea 1, se va transforma valoarea de pe7 biti în 2 valori pe 4 biti respectiv in min\_ZECI si min\_UNITATI.

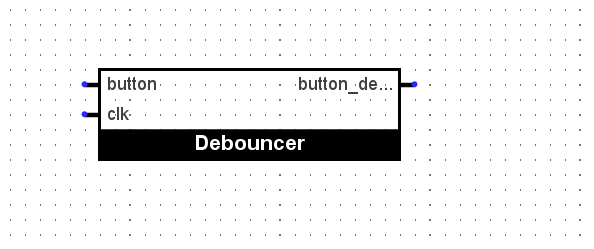
**6.Divizor de frecvență**



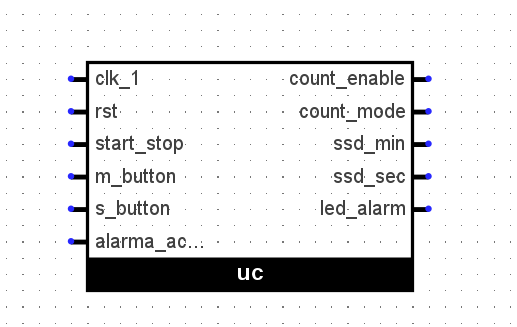
Divizorul de frecvență are o intrare clk\_100 care subdivizează clk ul de 100mHz de pe plăcuță într-un clk de 1Hz(clk\_1).

**7.Debouncer**

Debouncer-ul este conceput pentru a controla „apăsarea” butoanelor de intrare. Prin urmare, vom folosi 3 astfel de circuite, o dată pentru fiecare dintre butoanele noastre: start\_stop, m\_button și s\_button.



**8.State Generator**



Aceasta resursă conține un semnal clk de 1Hz, rst care resetează timer-ul, un buton start\_stop care ne arată că putem începe numărarea, m\_button care daca este apăsat se vor incrementa valoarea minutelor pe afișor, s\_button care dacă este apăsat se vor incrementa valoarea secundelor pe afișor.

**9.Full Display**

A picture containing text, screenshot, diagram, line

Description automatically generated

Este componenta care ne arată ieșirea pe cele 7 segmente. Toate semnalele interfeței SSD (7 semnale comune de catod și 4 semnale distincte de anod) sunt active pe 0. Semnalele de catod controlează ledurile care se aprind de pe acele cifre care au semnalul de anod activ (de exemplu dacă se activează toate 4 anodurile, atunci se va afișa aceeași cifră pe cele 4 poziții). Pentru a afișa 4 cifre diferite pe SSD, este necesară implementarea unui circuit care trimite cifrele pe semnalele de catod ale SSD în concordanță cu diagrama de timp a SSD. Perioada maximă de reîmprospătare (refresh) este astfel calculată încât ochiul uman să nu perceapă aprinderea și stingerea succesivă a fiecărei cifre de pe SSD.

### Reprezentarea UC prin diagrama de stări (organigrama)

Un cerc reprezintă o stare, un dreptunghi reprezintă ieșiri ale UC-uluil, iar romburile reprezintă decizii luate în fiecare stare.

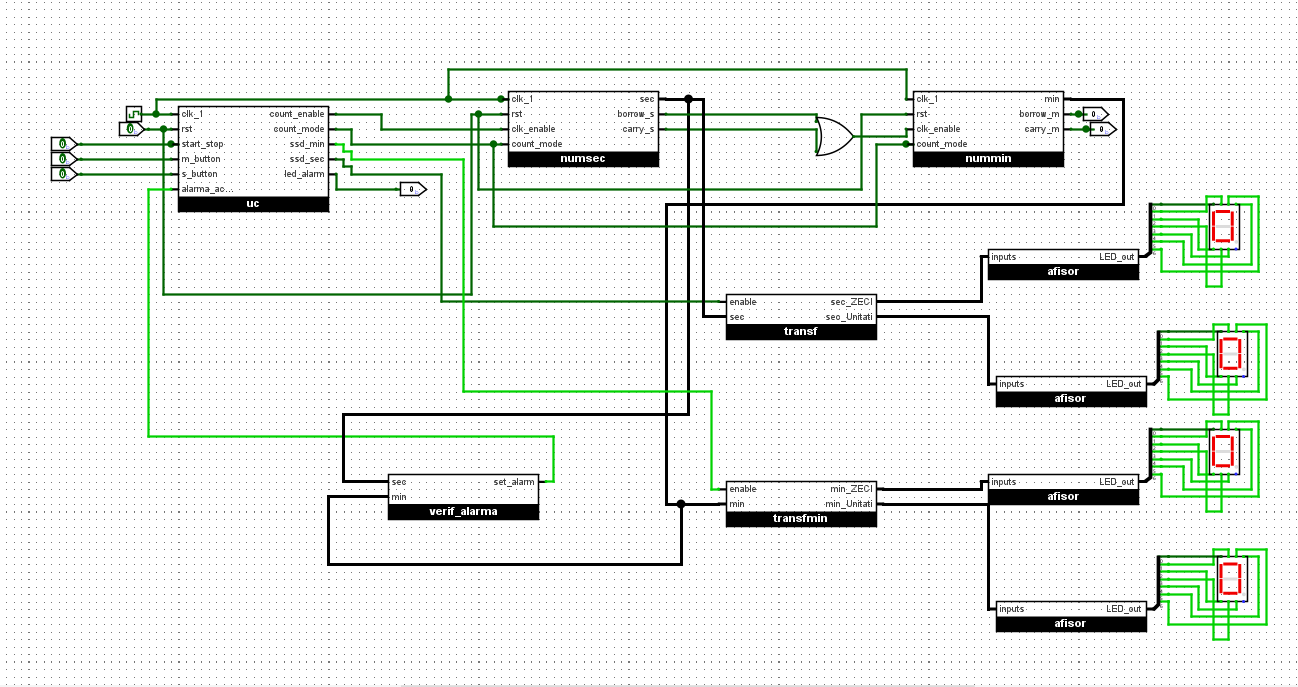
Timer-ul are 7 stări diferite**: Zero**(starea inițială), **Crescător**( numără crescător), **Afis\_sec**( se incrementează valoarea secundelor), **Afis\_min**(se incrementează valoarea minutelor), **Afis\_min\_sec**(se incrementează atât valoarea minutelor, cât și a secundelor), **Descrescător**(numără descrescător**), Alarma**(generează alarma).

Count\_mode-: daca este 1 numără crescător, iar dacă este 0 numără descrescător.(**ZCR**)

Clock\_enable: dacă este activ, timer ul numără (**ZCE**)

A diagram of a flowchart

Description automatically generated with low confidence



### Schema de detaliu a proiectului

Pe această schemă unitatea de control apare ca și o componentă deoarece descrierea este una comportamentală.

# Manual de utilizare și întreținere

Avem 4 butoane: **start\_stop, M,S,RST**. Presupunem că inițial starea noastră este Zero, dacă o să apasăm pe buton de start\_stop, timer-ul va porni. Dacă se va apăsa pe butonul de secunde S, se vor incrementa și se vor afișa secundele la momentul respectiv, iar daca se va apăsa butonul de minute M, se vor incrementa și se vor afișa secundele la momentul respectiv. Dacă după apasarea butoanelor M,S se apasă din nou start\_stop, timer ul va începe să numere descrescător și va funcționa exact la fel. Când se va ajunge la 00:00 se va emite o alarma(led) care va lumina o anumită perioadă de timp.Dacă se apasă butonul RST, timeru-ul va începe din nou de la 00:00.

# Justificarea soluției alese

Am ales această soluție, deoarece mi se pare cea mai ușoară soluție de a implementa un timer. În plus, acest proiect este interesant, deoarece înțelegi cum functionează un timer.

# Posibilități de dezvoltări ulterioare

* Să se incrementeze și numărul de ore, pentru a fi un timer complet.
* Să poți adăuga tu un anumit timp, iar de la acel timp să înceapă să îți numere.
* Să poți pune alarmă la orice moment dorit de utilizator

# Bibliografie

<https://www.doulos.com/knowhow/perl/vhdl-testbench-creation-using-perl/>

https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/350-9%20Oniga.pdf