|  |  |
| --- | --- |
|  |  |
| TERMOSTAT |  |
|  |  |
|  | 29 Mai 2023Proiect PSN |
|  | Student: Fodor HelgaProfesor: Șandru Diana |

1. Specificații

Se proiectează un termostat pentru o centrală termică de apartament. Termostatul este prevăzut cu un termistor pentru măsurarea temperaturii. Se pot programa o valoare minimă şi una maximă de temperatură a apartamentului. Aceste valori pot fi apoi asociate pentru fiecare oră (din cele 24 de ore ale zilei). În funcţie de programare, termostatul trebuie să comande pornirea incalzirii. Adiţional termistorul va primi date de la o unitate de simulare, care va simula încălzirea cu 1° pentru fiecare 3 secunde cu elementul de încălzire pornit. Termostatul trebuie să funcționeze astfel:

* Starea inițială este una de așteptare, iar termostatul este închis.
* Când se apasă butonul de “START” se așteaptă să se introducă o valoare minimă de temperatură și apoi una maximă, pentru a ști de la ce temperatură trebuie să se încălzească centrala și până la ce temperatură trebuie să se încălzească. După care se așteaptă si setarea intervalului de timp în care se încălzește centrala și își menține după încălzire temperatura.
* În momentul în care se ajunge la temperatura dorită se aprinde un led “FINISH\_ÎNCĂLZIRE”.
* Apoi se trece în starea în care se numără intervalul de timp în care se păstrează temperatura maximă. Când se termină timpul(care de regulă este o oră din cele 24 de ore ale zilei) se aprinde un led “FINISH” și revine în starea de așteptare.

1. Proiectare
   1. Schema bloc

FINISH ÎNCĂLZIRE

CLK

TEMPERATURĂ MINIMĂ

START

INTERVAL DE TIMP

TEMPERATURĂ MAXIMĂ

*TERMOSTAT*

FINISH

* **BUTOANE**: START(1 buton)
* **SWITCH-URI**: TEMPERATURĂ MINIMĂ(4 switch-uri), TEMPERATURĂ MAXIMĂ(4 switch-uri), INTERVAL DE TIMP(4 switch-uri)
* **LED-URI**: FINISH(1 led), FINISH ÎNCĂLZIRE(1 led)

Alte intrări auxiliare sunt folosite pentru confirmarea introducerii temperaturilor și intervalor de timp. În acest caz vom avea 3 intrări de acest tip: **OK\_Temp\_**min, **OK\_temp\_max**, **OK\_timp**. Pentru aceste 3 intrări auxiliare vom folosi încă 3 switch-uri.

Pe lângă aceste intrări auxiliare mai avem 2 intrări auxiliare: **Reset**(1 buton) și **Încărcare temperatură**(1 buton).

* 1. Unitatea de Control și Unitatea de Execuție

Cutia neagră a sistemului se descompune mai departe pentru a putea găsi componente implementabile. Vom face o descompunere **top-down** a problemei până când ajungem la circuite cunoscute, iar apoi vom implementa **bottom-up**.

Prima descompunere a oricărui sistem este una în care vom diferenția **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.

* + 1. Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE

TEMPERATURĂ MINIMĂ

**U.E.**

**U.C.**

FINISH

CLK

START

INTERVAL DE TIMP

TEMPERATURĂ MAXIMĂ

FINISH ÎNCĂLZIRE

Putem împărți atât intrările cât și ieșirile în 2 categorii: ***de date*** si ***de control***. Aceasta separare este esențială la început.

**Intrări de date**: *temperatură minimă, temperatură maximă, interval de timp, CLK* , Reset, *En\_temp\_min, En\_temp\_max, En\_ora, En\_cnt\_temp, En\_cnt\_ora, Incarcare temperature,*  *OK\_timp, OK\_temp\_min, OK\_temp\_max* (valori pentru diferite lucruri)

**Intrări de control**: *start, OK\_timp, OK\_temp\_min, OK\_temp\_max, Fin\_temp\_min, Fin\_temp\_max, Fin\_ora, Fin\_cnt\_temp, Fin\_cnt\_ora* (buton de confirmare, buton de selectare a unui program, buton de anulare etc.)

**Ieșiri de date**: *Fin\_temp\_min, Fin\_temp\_max, Fin\_ora, Fin\_cnt\_temp, Fin\_cnt\_ora*(valori de afișat pentru utilizator)

**Ieșiri de control**: *En\_temp\_min, En\_temp\_max, En\_ora, En\_cnt\_temp, En\_cnt\_ora, Incalzire, Finish* (semnale de avertizare sau atentionare a utilizatorului, prin care noi putem sa controlam și indrumam utilizatorul prin funcționarea sistemului)

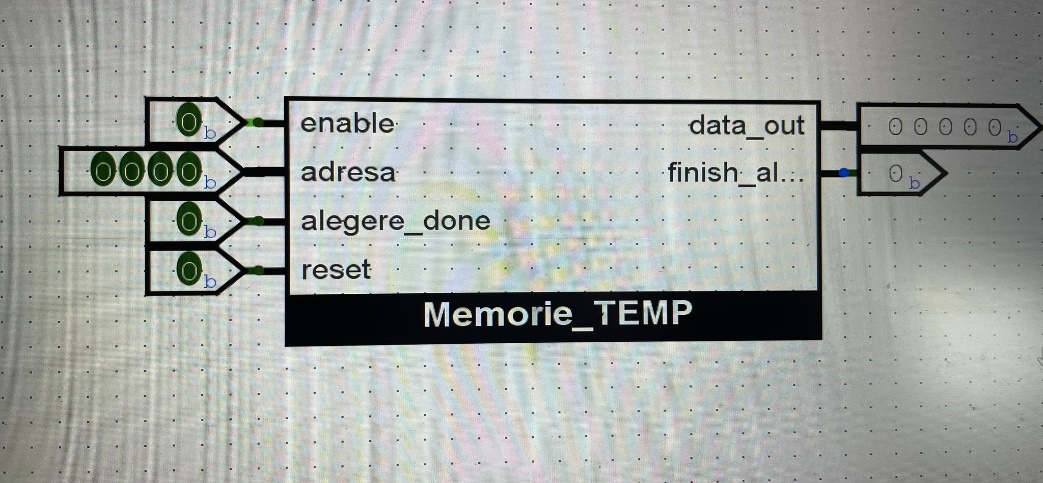
* + 1. Determinarea resurselor (UE)

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.

Orice informație pe baza careia se ia decizii trebuie sa vină de la o resursă care generează acea informație și o transmite mai departe UC.

**RESURSE**:

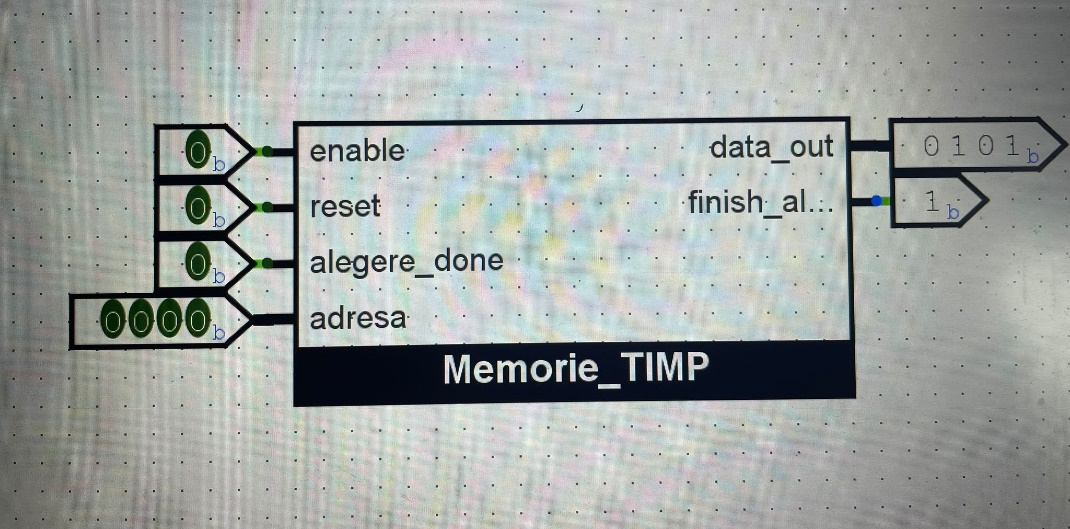
1. Memorie\_TEMP
2. Memorie\_TIMP
3. Numarator\_creste\_TEMP
4. Numărător\_intervalTIMP
5. **Memorie\_TEMP**



Aceasta este o memorie de tip ROM cu capacitate 16x5 în care se păstrează 16 temperaturi din care putem să alegem pentru a seta termostatul.

Mod de funcționare: după ce această memorie este activă(**enable=1**), putem selecta o temperatură codificată pe 4 biți(**adresa**), iar confirmarea temperaturii alese se face prin intrarea **alegere\_done**, după care pe ieșire va apărea temperature aleasă(**data\_out**) și ieșirea **finish\_alegere** se va activa.

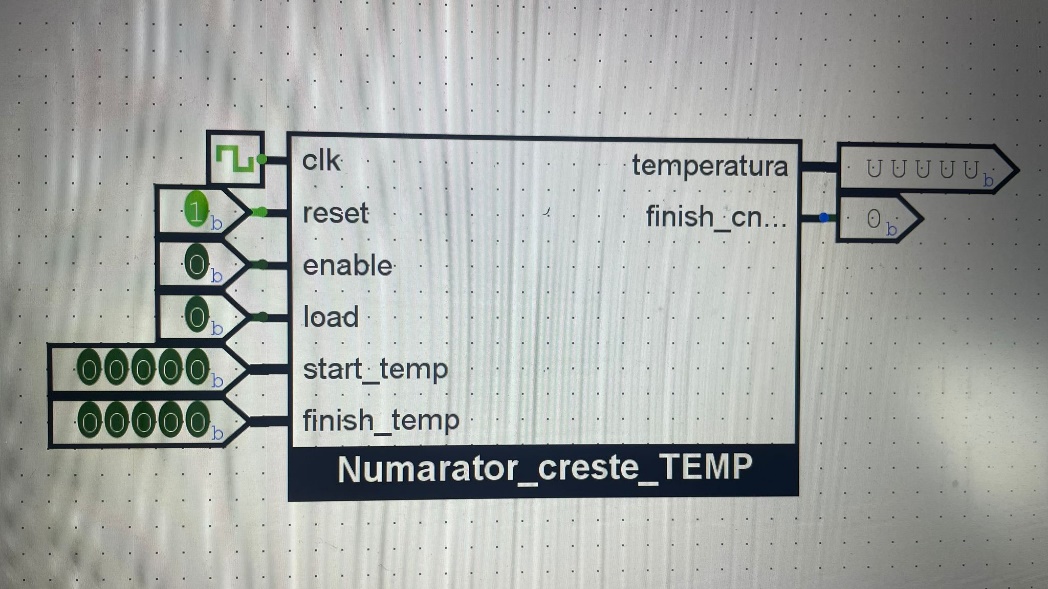
1. **Memorie\_TIMP**

****

Aceasta este o memorie de tip ROM cu capacitate 16x4 în care se păstrează 12 ore din care putem să alegem pentru a seta termostatul.

Mod de funcționare: după ce această memorie este activă(**enable=1**), putem selecta o oră codificată pe 4 biți(**adresa**), iar confirmarea orei alese se face prin intrarea **alegere\_done**, după care pe ieșire va apărea ora aleasă(**data\_out**) și ieșirea **finish\_alegere** se va activa.

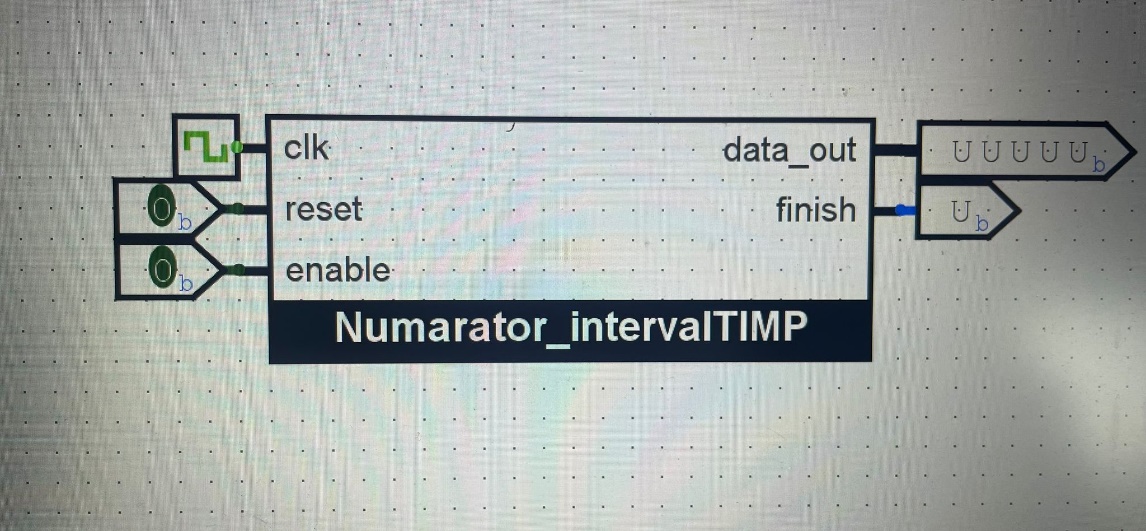
1. **Numarator\_creste\_TEMP**

****

Acesta este un numărător pe 5 biți cu încărcare paralelă(**load**) care numără de la o valoare până la o altă valoare.

Mod de funcționare: după ce numărătorul devine activ(**enable=1**) , încărcăm cele 2 temperaturi alese de memorie(**start\_temp** – temperatura minimă, **finish\_temp** – temperatura maximă), încărcarea se face prin activarea temporară a intrării **load**, după dezactivarea intrării load se începe numărarea până la temperatura maximă. Acest numărător va număra de fiacre data cand au trecut 3 secunde(3 stări). După ce se ajunge la temperature dorită se activează ieșirea **finish\_cnt\_temp**.

1. **Numărător\_intervalTIMP**



Acesta este un numărător pe 5 biți care numără de la 0-20. Prin numărarea a 20 de stări numărăm trecerea celor “60 de minute” de menținere a temperaturii.

Mod de funcționare: după ce numărătorul devine activ(**enable=1**) și se resetează temporar(**reset=1**) se începe numărarea de la 0-20, iar când s-a ajuns la final de buclă se activează ieșirea **finish**.

* + 1. Schema bloc cu legăturile dintre UC și UE

CLK

INTERVAL DE TIMP

TEMPERATURĂ MAXIMĂ

TEMPERATURĂ MINIMĂ

START

FINISH ÎNCĂLZIRE

FINISH

**U.E.**

**U.C.**

Ieșiri de control

Ieșiri de date

**Ieșiri de date**: *Fin\_temp\_min, Fin\_temp\_max, Fin\_ora, Fin\_cnt\_temp, Fin\_cnt\_ora*.

**Ieșiri de control**: *En\_temp\_min, En\_temp\_max, En\_ora, En\_cnt\_temp, En\_cnt\_ora, Incalzire, Finish.*

* + 1. Reprezentarea UC prin diagrama de stări (organigrama)

Diagrama de stări ce reprezintă partea de control, partea decizională din algoritm.

* Stările sunt reprezentate prin:

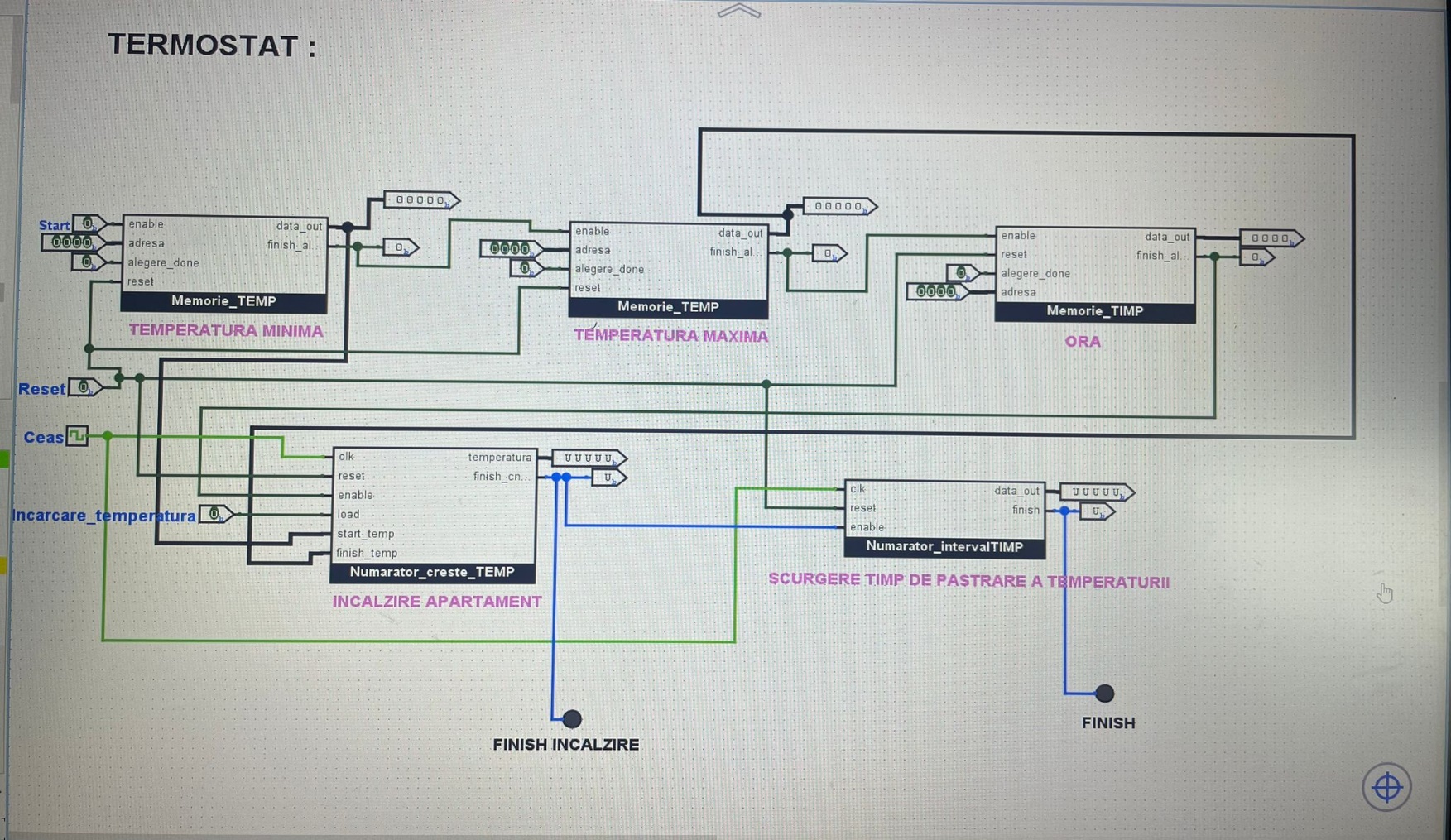
* Deciziile sunt reprezentate prin:
* Ieșirile sunt reprezentate prin:

**Organigrama:**

A picture containing screenshot, text, graphics, design

Description automatically generated

* + 1. Schema detaliată a proiectului



* + 1. Posibilități de dezvoltare
* Integrarea unui senzor de temperatură
* Afișarea temperaturilor și timpului pe 7 segment
* Posibilitatea adăugării mai multor setări de ore însoțite de temperaturi