

PROIECT REALIZAT DE STUDENȚII:

- SIMOTA MIHNEA – DRAGOȘ
- MUREȘAN RAUL – CĂLIN

COORDONATOR:

- DR. ING. POP DIANA IRENA

Contents

Contents.....	2
1 Specificații.....	3
2 Proiectare.....	3
2.1 Schema Bloc	3
2.2 Unitatea de Control și Unitatea de Execuție.....	4
2.2.1 Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE.....	4
.....	4
2.2.2 Determinarea resurselor (UE)	4
2.2.3 Schema bloc a primei descompuneri	7
2.2.4 Reprezentarea UC prin diagrama de stări (organigrama)	8
2.2.4 Schema de detaliu a proiectului	10
.....	10
3 Manualul de utilizare și întreținere	10
4 Justificarea soluției alese	11
5 Posibilități de dezvoltări ulterioare	12
6 Bibliografie.....	12

APLICAȚIE COD CIFRU PENTRU SECURIZAREA DULAPURILOR

1 Specificații

Acest proiect prezintă arhitectura internă a unei aplicații pentru securizarea dulapurilor. Funcționarea acesteia se poate descrie astfel: utilizatorul trebuie să introducă pe rând 3 caractere, reprezentând cifrele de la 0 la 9 la care se adaugă literele A – F, pentru a securiza dulapul. După ce se securizează dulapul. Pentru a selecta cifrele, utilizatorul se folosește de butonul AD_CIFRU, și de butoanele UP și DOWN. Butonul RESET are scopul de a șterge ce este pe afișorul SSD, iar în cazul în care este blocat, după apăsarea butonului dulapul rămâne tot blocat. Butonul MASTER_RESET resetează toată aplicația la starea inițială.

Pentru a deschide dulapul, utilizatorul trebuie să introducă în mod corect codul.
Codul introdus apare pe afișorul BCD 7 segmente.

2 Proiectare

2.1 Schema Bloc

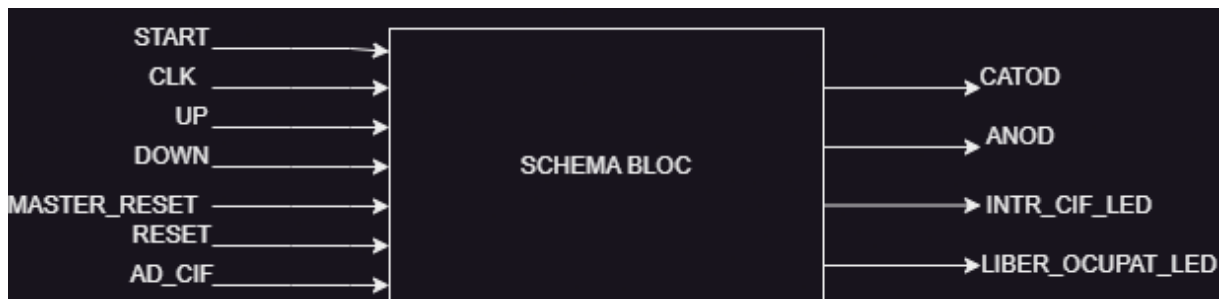


Figura 1 Cutia neagra a sistemului cu intrările și ieșirile stabilite

2.2 Unitatea de Control și Unitatea de Execuție

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

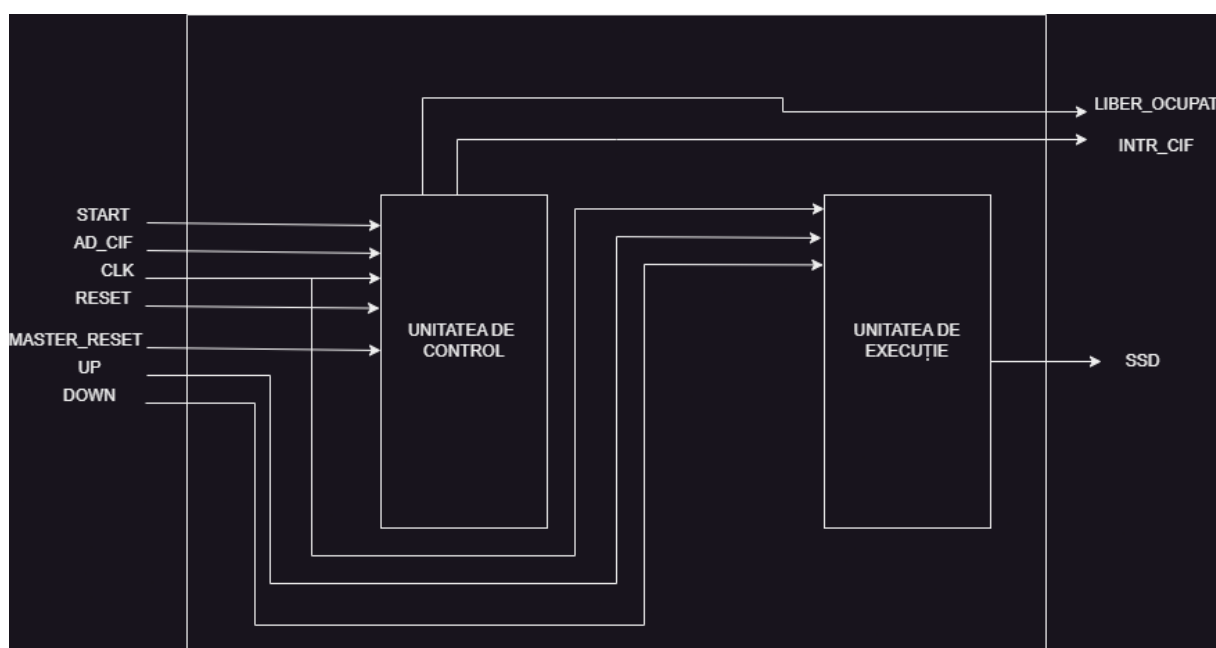


Figura 2 Descompunere în UC și UE

Intrările de control: butoanele “START”, “AD_CIF”, “RESET”, “UP”, “DOWN” semnalul sincron “CLK”.

Ieșirile de control: led-urile “LIBER_OCUPAT”, “INTR_CIF”

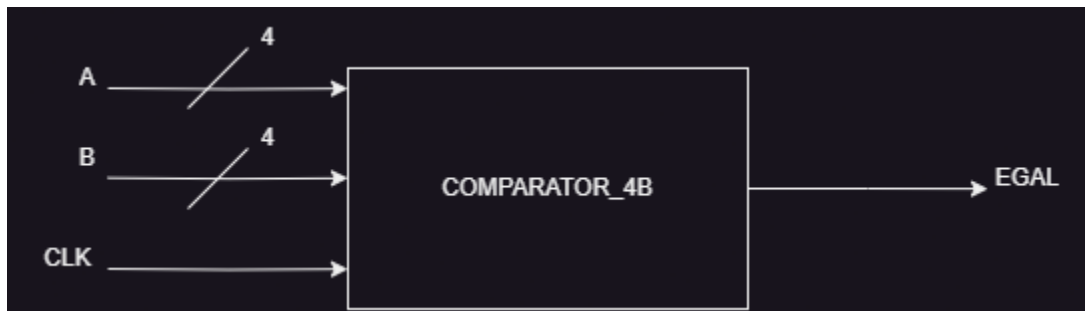
Intrările de date: si toate led-urile prezente pe cele 3 “SSD”.

2.2.2 Determinarea resurselor (UE)

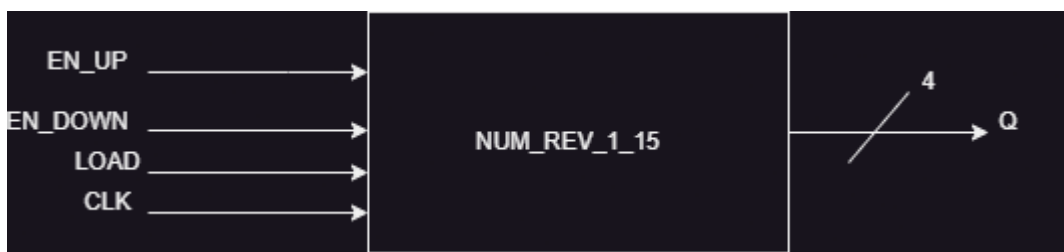
Pentru a evidenția legăturile dintre unitatea de control și unitatea de execuție, este necesară identificarea resurselor. Există resurse pe baza cărora se vor lua decizii, dar și resurse care servesc ca informații pentru utilizator. Aceste resurse pot să genereze semnale către unitatea de control, și pot fi controlate prin intermediul unor intrări globale, precum “RESET”, “MASTER_RESET” sau semnalul sincron “CLK”.

Resursele:

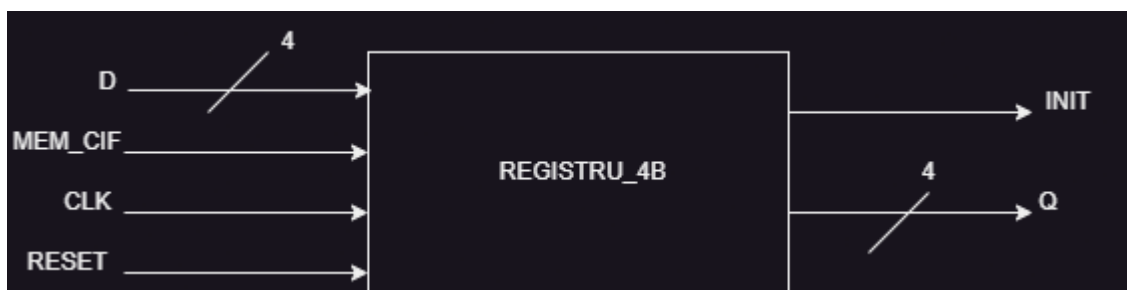
1. Comparator de două numere pe 4 biți



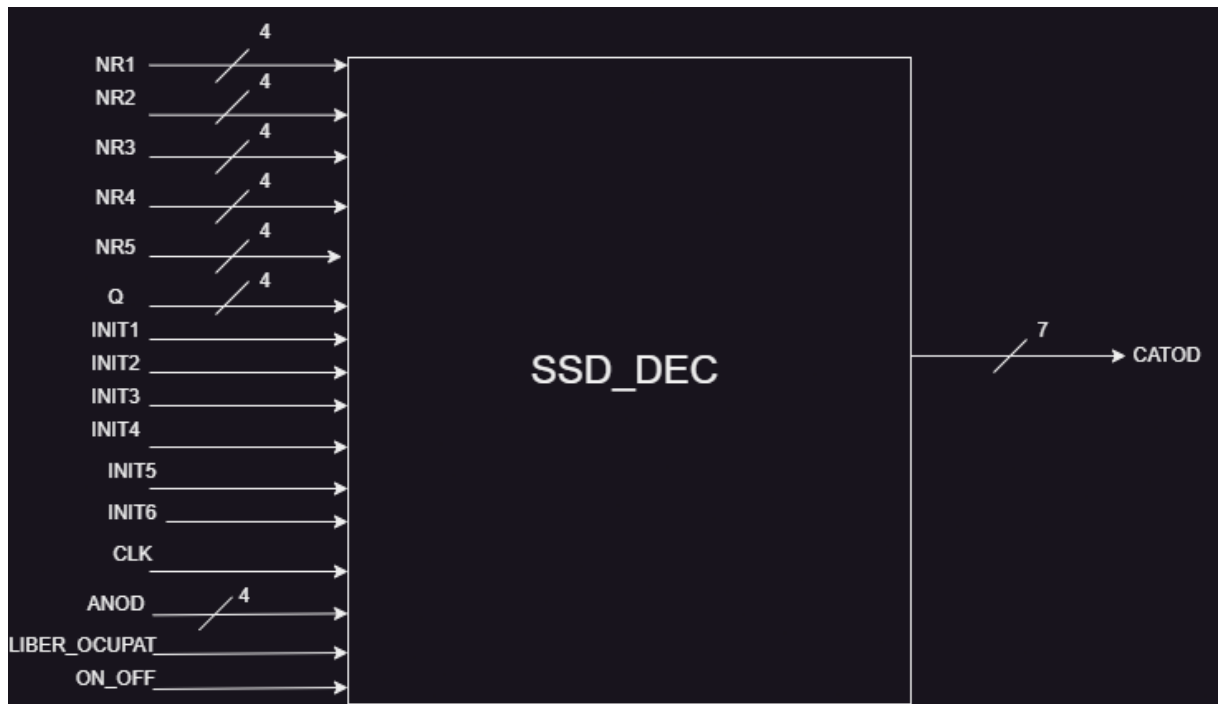
2. Numărător reversibil 1-15



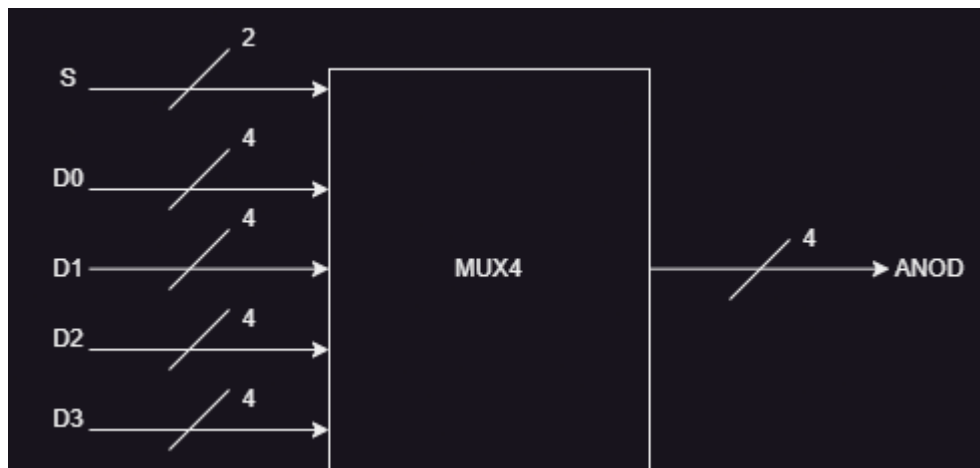
3. Registru cu două intrări pe 4 biți



4. Decodificator pe 7 segmente (SSD)



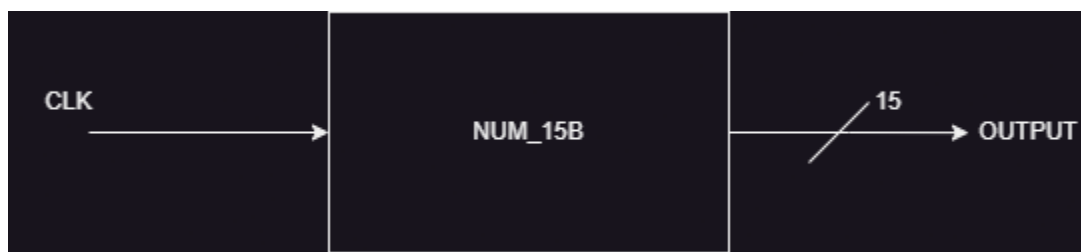
5. Multiplexor 4:1 cu intrări pe 4 biți



6. Multiplexor 4:1 cu intrări pe 7 biți



7. Numărător 15 biți



2.2.3 Schema bloc a primei descompuneri

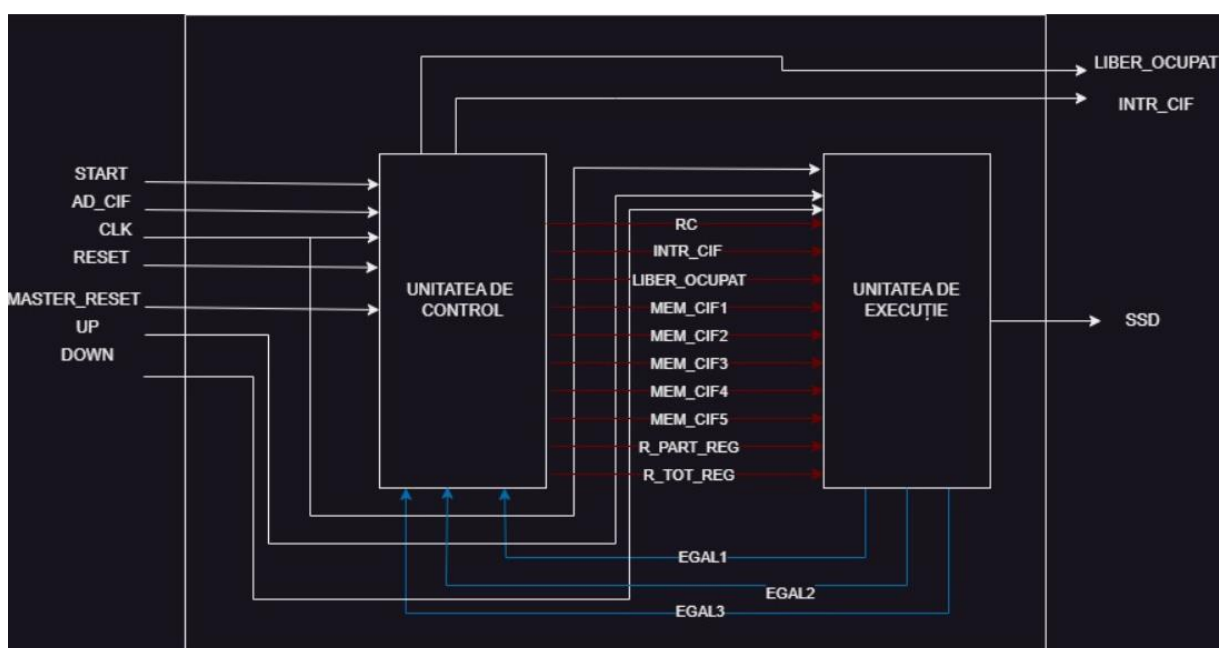
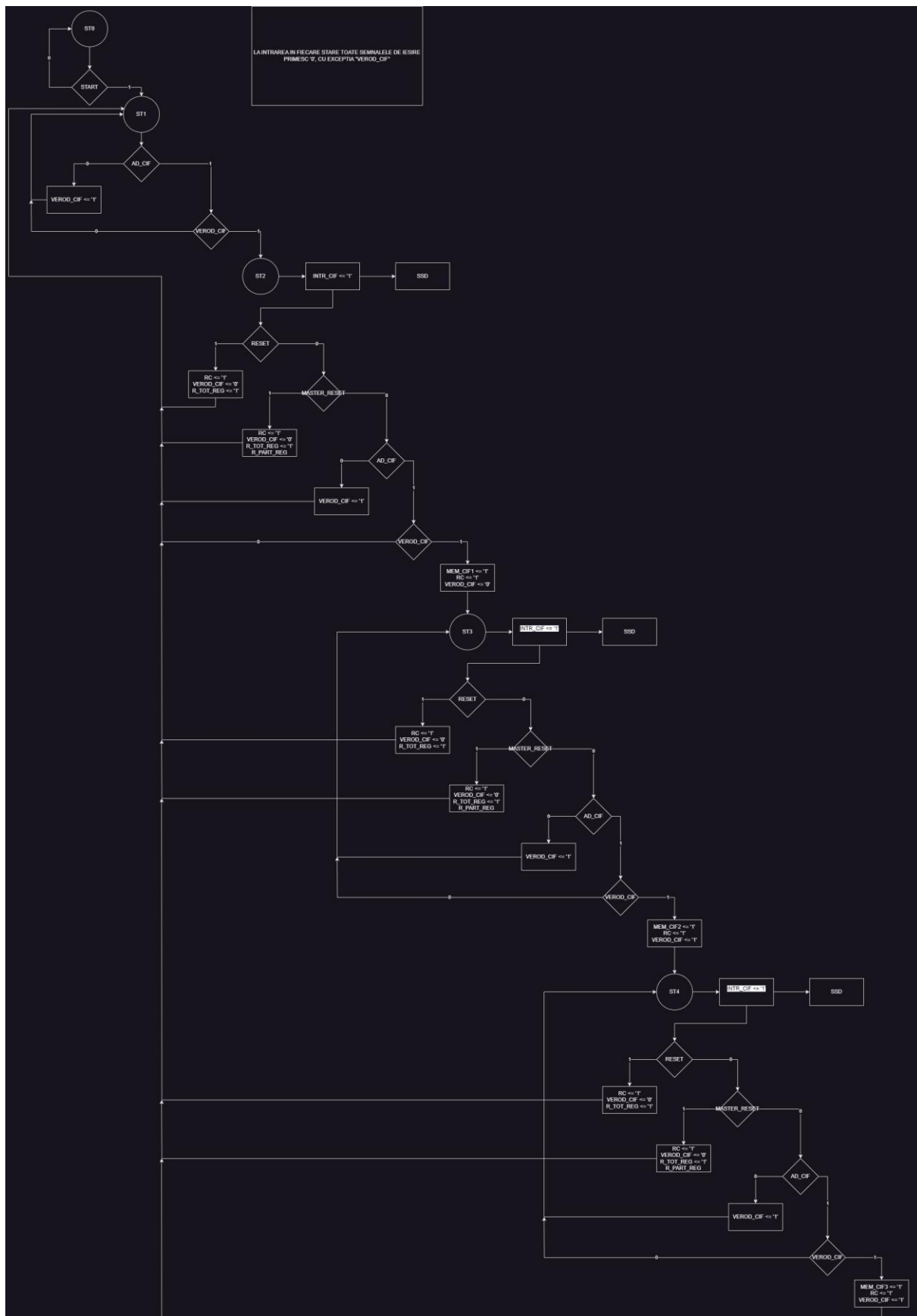
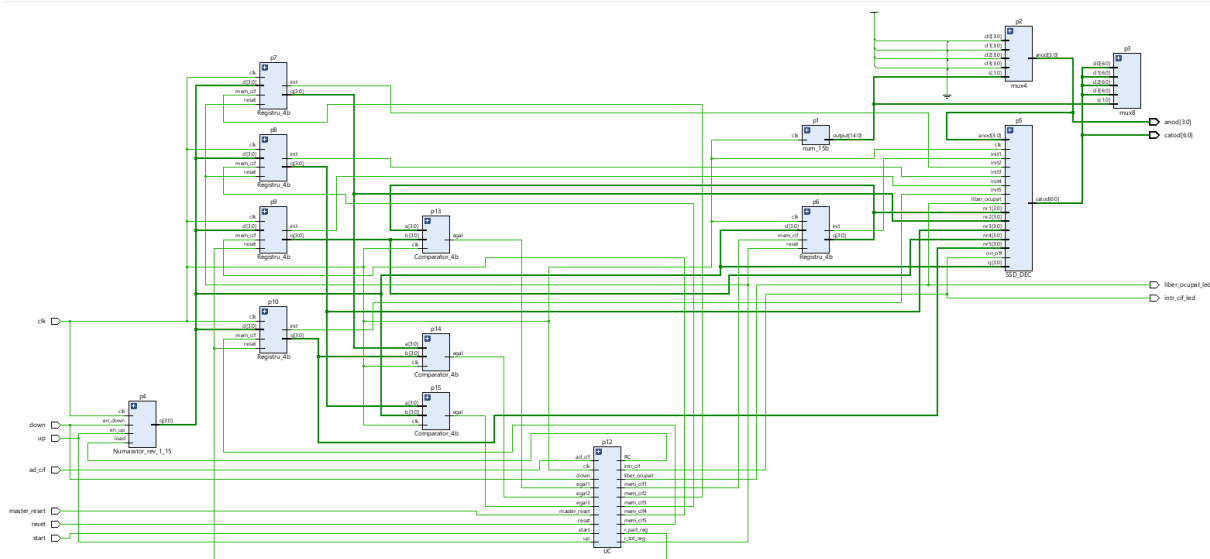


Figura 3 Schema bloc a primei descompuneri

2.2.4 Reprezentarea UC prin diagrama de stări (organigrama)



2.2.4 Schema de detaliu a proiectului



3 Manualul de utilizare și întreținere

Descriere: Sa se implementeze o aplicatie care permite utilizatorului adaugarea unui cifru din 3 caractere distincte pentru securizarea unui dulap (asemanator dulapurilor folosite la vestiarele de sport, mall, etc)

Cerinte functionale:

1. Un led LIBER_OCUPAT va avea functia de a semnala faptul ca dulapul este liber(led stins) sau ocupat(led aprins)
2. Utilizatorul va apasa un buton ADAUGA_CIFRU pentru a semnala inceperea introducerii codului. Un led INTRODUC_CARACTERE se va aprinde pentru a marca starea
3. Utilizatorul va adauga pe rand 3 caractere cu ajutorul butoanelor UP si DOWN
4. Caracterele sunt curpinse in intervalul 0-1-...-8-9-A-B-...-F
5. Caracterul curent este afisat pe SSD
6. Pentru trecerea la urmatorul caracter utilizatorul va apasa butonul ADAUGA_CIFRU
7. Caracterul anterior introdus ramane afisat
8. Urmatorul caracter este vizibil pe afisaj pe pozitia urmatoare
9. Dupa introducerea celui de al treilea caracter, la apasarea butonului ADAUGA_CIFRU, afisajul SSD se va stinge iar cifru va fi in starea blocat prin aprinderea ledului LIBER_OCUPAT.
10. Ledul INTRODUC_CARACTERE se va stinge
11. Existenta unui buton/switch RESET in timpul introducerii cifrului pentru revenire in starea initiala(ledul LIBER_OCUPAT se va stinge, afisajul SSD este gol, ledul INTRODUC_CARACTERE se va stinge)

12. Utilizatorul va apasa butonul/switch ADAUGA_CIFRU pentru a incepe introducerea codului pentru deblocarea cifrului
13. Se vor relua pasii 2-8
14. La introducerea ultimului caracter, la apasarea butonului ADAUGA_CIFRU se va face verificarea, daca codul introdus corespunde cu codul anterior.
15. In cazul de egalitate, ledul LIBER_OCUPAT se va stringe, ledul INTRODUC_CARACTERE se va stinge, afisajul SSD se goleste
16. In cazul de inegalitate, ledul LIBER_OCUPAT va ramane aprins, ledul INTRODUC_CARACTERE se va stinge, afisajul SSD se goleste.

Cerinte non-functionale:

- Implementare pe placuta
- Utilizare SSD
- Utilizare switch, led, butoane

Exemplu use case: Utilizatorul alege un dulap cu ledul Liber_ocupat stins. Apasa pe butonul ADAUGA_CIFRU pentru a introduce caracterele. Caracterul "0" este vizibil pe SSD. Introduce primul caracter "2" prin apasarea de 2 ori a butonului DOWN. Pe SSD se modifica afisajul o data cu apasarea butonului si anume: 0->1->2. Utilizatorul apasa din nou pe ADAUGA_CIFRU pentru a introduce al doilea caracter "1". Utilizatorul apasa din nou pe ADAUGA_CIFRU pentru a introduce al doilea caracter "3". Utilizatorul apasa din nou ADAUGA_CIFRU, codul este salvat, continutul ssd este gol, ledul LIBER_OCUPAT este aprins, ledul INTRODUC_CARACTERE se va stinge.

4 Justificarea soluției alese

Soluția aleasă de noi este una eficientă și îndeplinește toate cerințele. Am reușit acest lucru prin utilizarea componente:

- Comparator de două numere pe 4 biți – această componentă are ca intrări două numere pe 4 biți și un clock, iar rezultatul returnat este 0, dacă cele două numere nu sunt egale, sau 1, dacă cele două sunt egale. Avem nevoie de ea pentru a realiza verificarea de la final dintre două perechi de registre, respectiv un registru si valoarea din numărător.
- Numărător reversibil 1-15 – utilizatorul modifică valoarea acestuia prin intermediul butoanelor UP și DOWN, iar această valoare urmează să fie memorată într-unul din cele cinci registre. Numărătorul este sincron. Dacă input-ul LOAD are valoarea 1, acesta acționează ca un reset și îi dă acestuia valoarea 0, altfel nu se resetează.
- Registru cu două intrări pe 4 biți – este o componentă esențială deoarece memorează valoarea returnată de numărător, iar valorile din aceștia vor fi comparate la final pentru a decide dacă dulapul se deblochează sau nu. Registrul este sincron, intrarea D este ceea ce urmează a fi memorat, MEM_CIF are rol de enable, dacă este 1 valoarea se memorează, altfel nu, iar RESET resetează registrul la 0. Totodată, are ca ieșiri Q

pe același număr de biți ca D (4 biți), și INIT care ia valoarea 1 dacă în registru a fost memorată o valoare, altfel 0.

- Decodificator pe 7 segmente (SSD) – pentru această componentă vom folosi mai multe subcomponente, printre care: decodificatorul pe 7 segmente, două multiplexoare de 4:1, unul cu intrări pe 4 biți, pentru a decide care anod se activează și unul cu intrări pe 7 biți pentru a stabili catodii. Selecțiile pentru cele două multiplexoare vor fi primii 2 cei mai semnificativi biți ai rezultatului unui numărator pe 15 biți. Astfel toate acestea împreună generează un vector de anodi și unul de catodi. Intrările decodificatorului vor fi cele 5 numere din registre (NR1, ... ,NR5), ieșirea din numărator reversibil pe 4 biți (Q), semnalele de INIT din registre (INIT1, ... , INIT5) care verifică dacă în registru s-a memorat o valoare, clock-ul (CLK), vectorul de anodi (ANOD), valoarea semnalului LIBER_OCUPAT, pentru a stabili dacă ne folosim de regiștrii pentru blocare sau cei de deblocare și ON_OFF, care ia valoarea lui INTR_CIF pentru a stabili dacă se aprinde afișorul sau nu. Rolul acestuia este de a verifica dacă se aprinde afișorul și de a stabili cei 7 catodi.
- Multiplexor 4:1 cu intrări pe 4 biți – are ca intrări cele patru combinații posibile pentru anodi și selecția pe doi biți (primii doi biți semnificativi ai numărului pe 15 biți). Rolul este de a forma vectorul anod, prin care se decide aprinderea unui afișor specific.
- Multiplexor 4:1 cu intrări pe 7 biți – are ca intrări cele patru combinații posibile pentru catodi și selecția pe doi biți (primii doi biți semnificativi ai numărului pe 15 biți). Este important pentru a forma vectorul de catodi, ce ulterior aprinde segmentele.
- Numărător 15 biți – are ca intrare doar clock-ul. Va număra pe 15 biți de la 0 la valoarea maximă și va scoate ca output numărul respectiv.

5 Posibilități de dezvoltări ulterioare

- Extinderea posibilităților de alegere a caracterelor pentru cifru, astfel codul ar putea fi mult mai dificil de ghicit de altcineva. În loc de caractere de la 0 la F , putem avea de la 0 la Z.
- Extinderea numărului de cifre memorate, astfel în loc de 3 ar putea fi memorate mai multe caractere. Utilizatorul chiar ar putea alege câte cifre dorește să aibă cifrul lui.
- Pentru a mări nivelul securității dulapului, utilizatorul are posibilitatea de a utiliza amprenta sau recunoașterea facială. Totodată se poate folosi un master key, care se comportă ca și master reset-ul din sistem.

6 Bibliografie

- Model documentație: https://teams.microsoft.com/_?culture=en-us&country=us#/apps/d7958adf-f419-46fa-941b-1b946497ef84/sections/MyNotebook