

MINISTERUL EDUCAȚIEI



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Cifru pentru securizarea dulapurilor

PROIECT DSD

Student: Alin Tcaci, Radu Ioan Mariș

Coordonator: Ileana Blaj

2022

Cuprins

1.Specificație proiect	3
2. Schema bloc cu componentele principale	4
2.1 Black Box poti scrie cateva randuri...ca avem atatea intrari is cutare...si atatea iesiri, in dreapta schemei cum ati facut la proiectare si implementare	4
2.2 Schema detaliata.....	4
3.Proiectare și implementare	5
4.Utilizare și rezultate	8
5.Posibilitați de dezvoltare	9
6.Justificarea solutiei alese	9

1.Specificație proiect

Să se implementeze o aplicație care permite utilizatorului adăugarea unui cifru din 3 caractere distincte pentru securizarea unui dulap.

Cerințe funcționale:

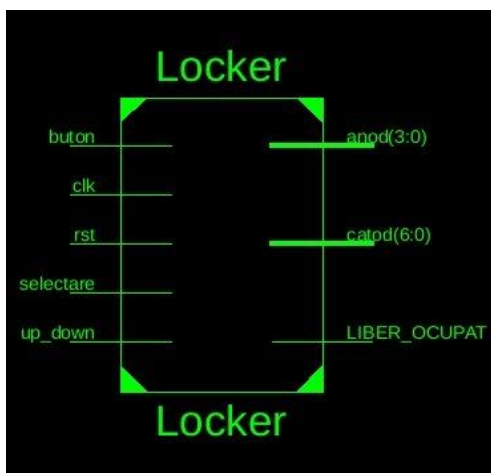
1. Un led **LIBER_OCUPAT** va avea funcția de a semnala faptul ca dulapul este liber (led stins) sau ocupat (led aprins)
2. Utilizatorul va apăsa un buton **SELECTARE** pentru a semnala începerea introducerii codului.
3. Utilizatorul va adăuga pe rând 3 caractere cu ajutorul butonului **BUTON** și switch-ului **UP_DOWN**
4. Caracterele sunt cuprinse în intervalul 0-1...A – F
5. Caracterul curent este afișat pe SSD
6. Pentru trecerea la următorul caracter utilizatorul va apăsa butonul **SELECTARE**
7. Caracterul anterior introdus rămâne afișat
8. Următorul caracter este vizibil pe afișaj pe poziția din dreapta a SSD-ului
9. După introducerea celui de-al treilea caracter, la apăsarea butonului **SELECTARE**, afișajul SSD se va stinge iar cifru va fi în starea blocat prin aprinderea ledului **LIBER_OCUPAT**
10. Existența unui buton **RST** în timpul introducerii cifrului pentru revenire în starea inițială (se va șterge doar conținutul afișajului, nu și parola memorată după aprinderea becului)
11. Utilizatorul va apăsa butonul **SELECTARE** pentru a începe introducerea codului pentru deblocarea cifrului
12. Se vor relua pașii 2-8
13. La introducerea ultimului caracter, la apăsarea butonului **SELECTARE** se va face verificarea, dacă codul introdus corespunde cu codul anterior
14. În cazul de egalitate, ledul **LIBER_OCUPAT** se va stinge, afișajul SSD se golește
15. În cazul de inegalitate, ledul **LIBER_OCUPAT** va rămâne aprins, afișajul SSD se golește și se șterge și parola memorată

Cerințe non-funcționale:

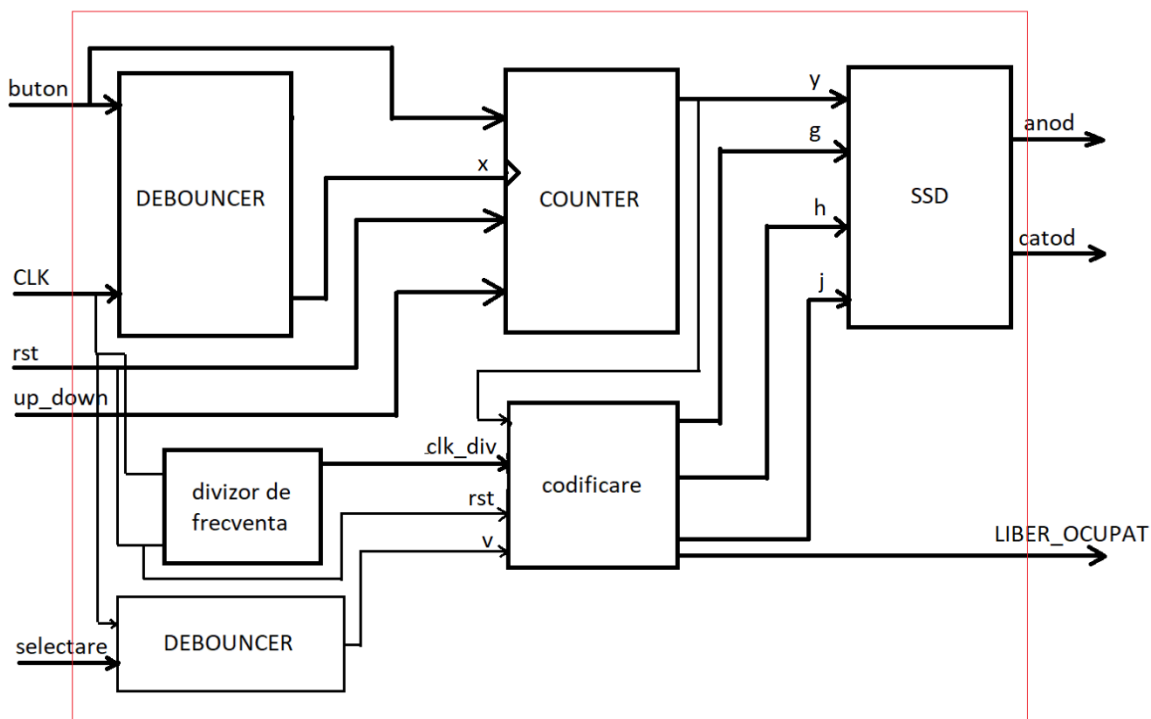
1. Implementare pe plăcuță
2. Utilizare SSD
3. Utilizare switch, led, butoane

2. Schema bloc cu componentele principale

2.1 Black Box -ul conține 5 intrări: buton, selectare, up_down, clock-ul (clk) și resetul (rst). De asemenea acesta conține și 3 ieșiri: anod și catod, folosite pentru reprezentarea pe afișorul cu 7 segmente și un led LIBER_OCUPAT care arată starea în care se află dulapul.



2.2 Schema detaliata



x = semnal folosit pentru a detecta butonul **BUTON**, doar o singură dată la apăsarea acestuia

clk_div = semnal folosit pentru micșorarea clock-ului

v = semnal folosit pentru a detecta butonul **SELECTARE**, doar o singură dată la apăsarea acestuia

y = semnal folosit la stocarea ieșirii de pe COUNTER

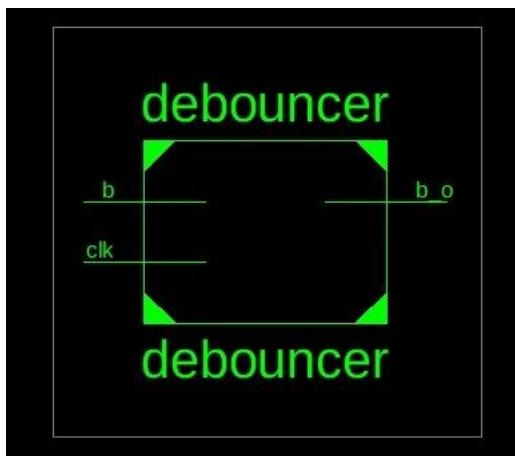
g, h, j = semnale folosite pentru a stoca numerele introduse in COUNTER

3.Proiectare și implementare

Componente utilizate:

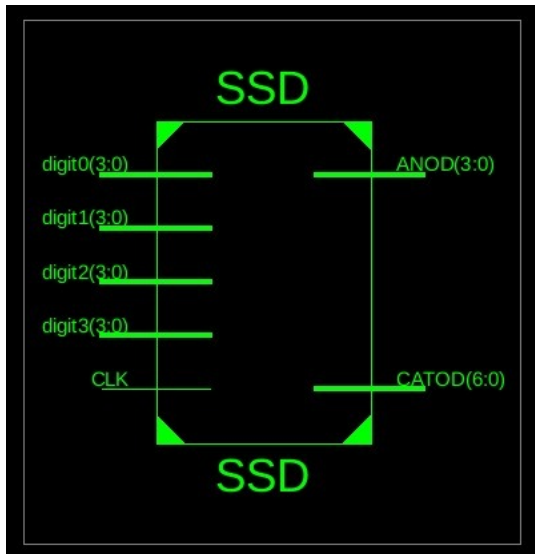
- Debouncer
- Divizor de frecventa
- Counter
- SSD
- Codificare

Debouncer:



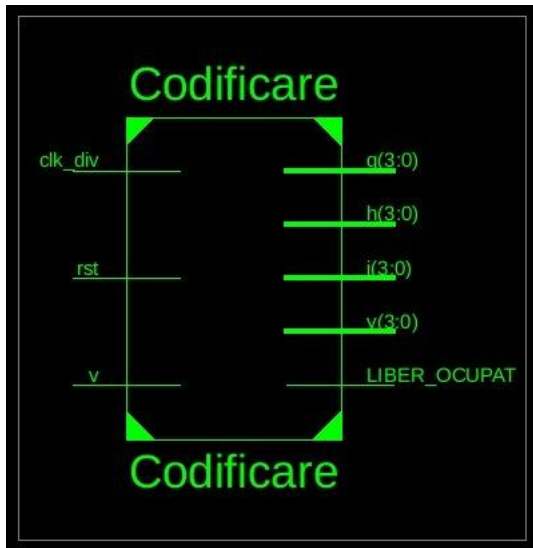
Componenta „Debouncer” face posibila funcționarea corecta a butoanelor **BUTON** si **SELECTARE**. Atunci când unul dintre butoane se apasă, acesta este acționat doar o singură dată (se utilizează pentru butoanele uzate).

SSD:



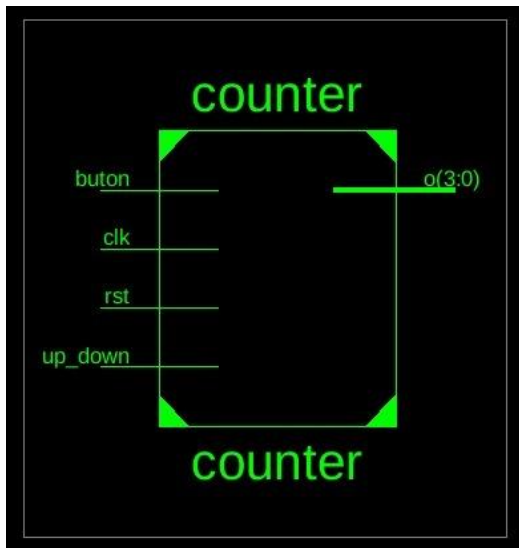
Componenta „SSD” are rolul de a converti semnalele de intrare pe 4 biți in semnale pe 7 biți, pregătite pentru redarea acestora pe afișorul de 7 segmente.

Codificare:



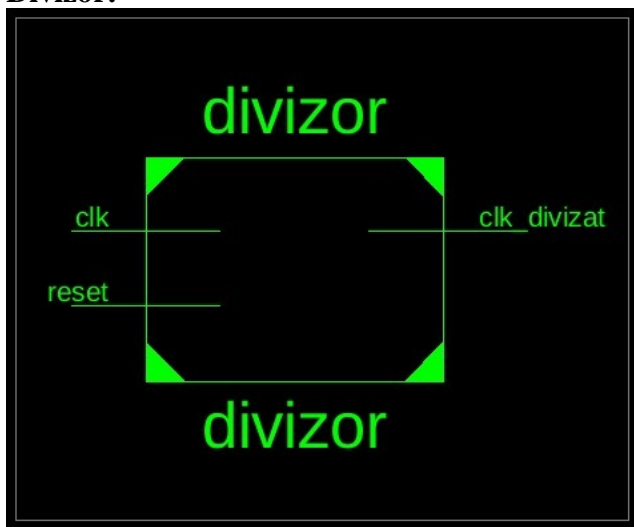
Componenta „Codificare” stochează cifrele de pe numărător, în momentul în care butonul selectare este apăsat, pe variabilele partajate. Când un set de 3 cifre este memorat, becul "LIBER_OCUPAT" se aprinde, iar mai apoi, următorul set de 3 cifre va fi comparat cu cel reținut. Dacă acestea coincid, becul se va stinge, iar toate variabilele vor fi reinițializate. Reset-ul asincron șterge cifrele de pe ultimul set, dar nu șterge setul memorat, tocmai pentru a evita apăsarea acestuia și ștergerea memoriei.

Counter:



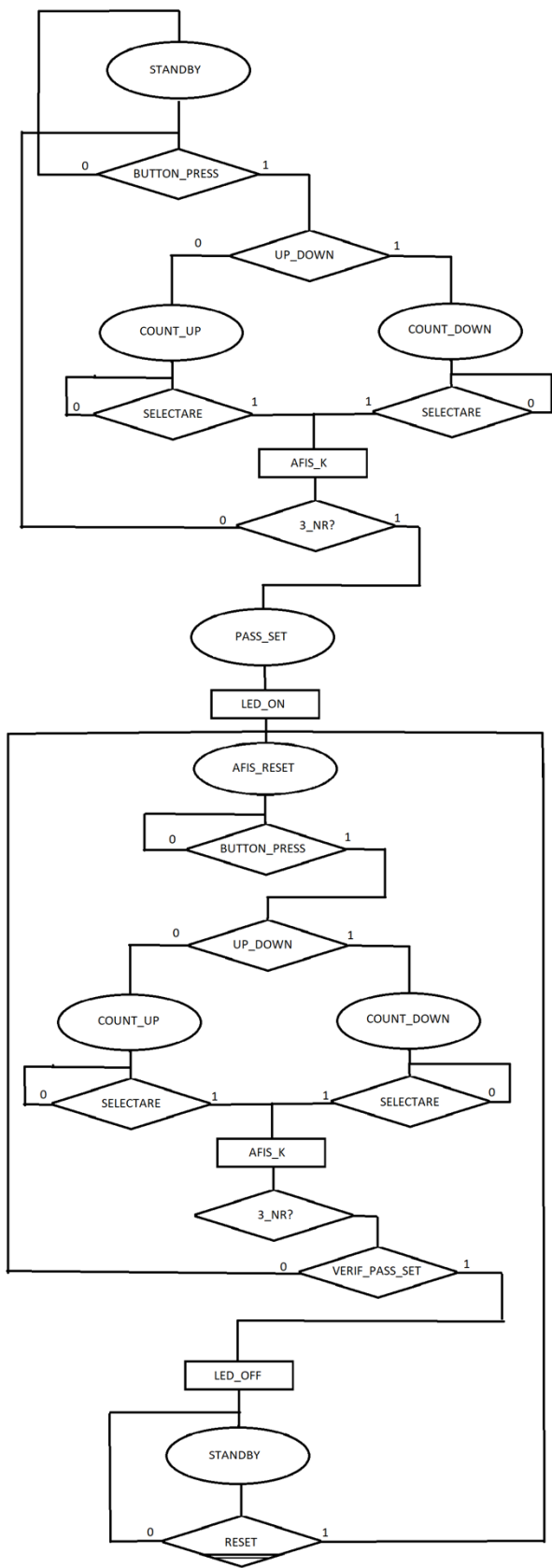
Componenta “Counter” este un numărător hexazecimal reversibil. Totodată incrementează și decrementează output-ul redat pe afișorul de 7 segmente.

Divizor:

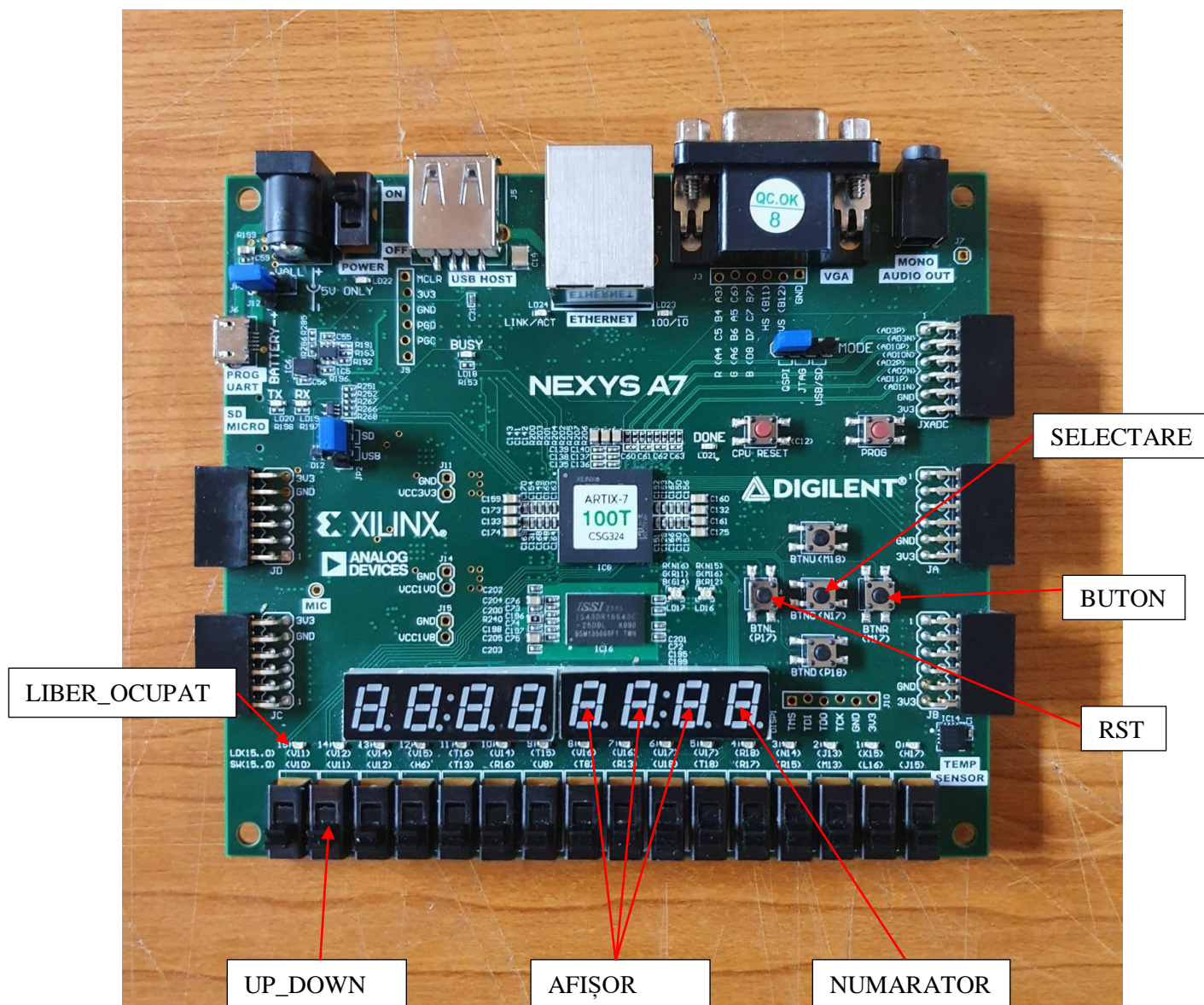


Componenta „Divizor” reduce frecvența clock-ului la 0,8 Hz (Frecvența inițială fiind de 100Mhz).

Flowchart:



4.Utilizare și rezultate



Utilizatorul va folosi butonul **BUTON** și switch-ul **UP_DOWN** pentru a ajunge la unul din cele 3 caractere pe care le conține cifrul. După care va apăsa butonul **SELECTARE**, iar caracterul ales va apărea pe afișor și va trece la alegerea următorului caracter. După ce au fost alese toate cele 3 caractere se va aprinde becul **LIBER_OCUPAT** și utilizatorul va putea introduce încă o dată un cod de 3 caractere. În cazul în care codul este identic cu cel introdus anterior becul se va stinge și alegerea cifrului va fi disponibilă din nou. În cazul opus, becul va rămâne aprins și utilizatorul va introduce un cod nou, până în momentul în care codurile sunt identice. Butonul **RST** golește afișorul, după care se reîncepe introducerea cifrului.

5.Posibilitați de dezvoltare

Ca o posibilă dezvoltare, utilizatorul va putea să își aleagă dulapul pe care dorește să îl folosească (adică unul din cele 16 becuri pe care le avem la dispoziție). Acest lucru va putea fi implementat cu ajutorul unui numărator, ale cărui caractere vor fi afișate pe celălalt afișor. Selectarea dulapului se va face tot cu ajutorul butoanelor **SELECTARE** și **BUTON** (butoanele au aceleași funcții ca în proiectul inițial.

6.Justificarea soluției alese

Soluția aleasă folosește un afișor cu 7 segmente pentru a fi accesibil, oferind utilizatorului, experiența cât mai plăcută. Butonul de reset permite acestuia să reintroducă parola în cazul în care a tasta greșit. Utilizarea unui debouncer pentru butoanele existente evită introducerea eronată a codului din cauze precum apăsarea continuă pe buton sau în cazul în care butonul este deteriorat si creează un contact imperfect.