5/9/2023

DIANA DINCA & LUCA-ANDREI CODOREAN

Grupa 30213

Îndrumător: ing. SZABLOCS-ANDRAS CSILLAG

Cod cifru pentru securizarea dulapurilor

Proiect pentru Proiectarea Sistemelor Numerice

Cuprins

[1 Specificații 2](#_Toc134543717)

[2 Proiectare 3](#_Toc134543718)

[2.1 Schema Bloc 3](#_Toc134543719)

[2.2 Unitatea de Control și Unitatea de Execuție 3](#_Toc134543720)

[2.2.1 Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE. 4](#_Toc134543721)

[2.2.2 Determinarea resurselor (UE) 5](#_Toc134543722)

[2.2.3 Schema bloc a primei descompuneri 7](#_Toc134543723)

[2.2.1 Schema de detaliu a proiectului 8](#_Toc134543724)

[3 Manual de utilizare și întreținere 10](#_Toc134543725)

[4 Justificarea soluției alese 11](#_Toc134543726)

[5 Posibilități de dezvoltări ulterioare 11](#_Toc134543727)

[6 Bullet-list 12](#_Toc134543728)

[7 Bibliografie 12](#_Toc134543729)

COD CIFRU PENTRU SECURIZAREA DULAPURILOR

# Specificații

Se dorește implementarea funcționalității unui cifru capabil să securizeze un dulap, compact și ieftin, conceput în special pentru dulapurile aflate în incinta căminelor studențești. Cifrul funcționează după cum urmează:

* Starea inițială este una de așteptare, care semnalează faptul că dulapul este nefolosit (deschis), printr-un LED de culoare verde.
* Se începe introducerea celor trei caractere care formează codul, pe rând, astfel încât după introducerea celor trei caractere, LED-ul devine roșu.
* Se așteaptă un nou input, asemănător celui inițial, dar care validează codul inițial. Utilizatorul are trei încercări de introducere a codului, după care dulapul intră într-un mod de blocare.
* Modul de deblocare poate să fie resetat manual, prin acționarea unui buton, de către administrator;
* Deblocarea este marcată de resetarea codului introdus inițial, respectiv colorarea LED-ului de stare din roșu în verde.

Considerăm mecanismul fizic de blocare deja existent.

# Proiectare

## Schema Bloc

O imagine care conține grafic

Descriere generată automat

Figura 1. Cutia neagra a sistemului cu intrările și ieșirile stabilite

Schema bloc a cutiei negre este reprezentată de trei intrări și cinci ieșiri, astfel: “Character” marchează locul în care sunt introduse pe rând cele trei caractere ale cifrului, “Reset” marchează revenirea cifrului in stare inițială, pentru ca parola să se poată reseta , “Clock” marchează semnalul de tact care după fiecare apăsare permite reținerea următorului caracter din cod. Ieșirile “First SSD”, “Second SSD” și “Third SSD” permit afișarea caracterelor ce compun codul cifrului, “Status” marchează starea în care se găsește dulapul, cu ajutorul unui led luminat verde respectiv roșu, iar “Blocked” marchează dacă dulapul este ocupat sau liber. Acest circuit este acceptat de placa *Nexys4DDR 16 switch-uri, 5 butoane, 16 led-uri, 8 SSD*.

Oricând pot sa mai fie adăugate semnale de intrare si ieșire pe parcurs.

## 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 când 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*).

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

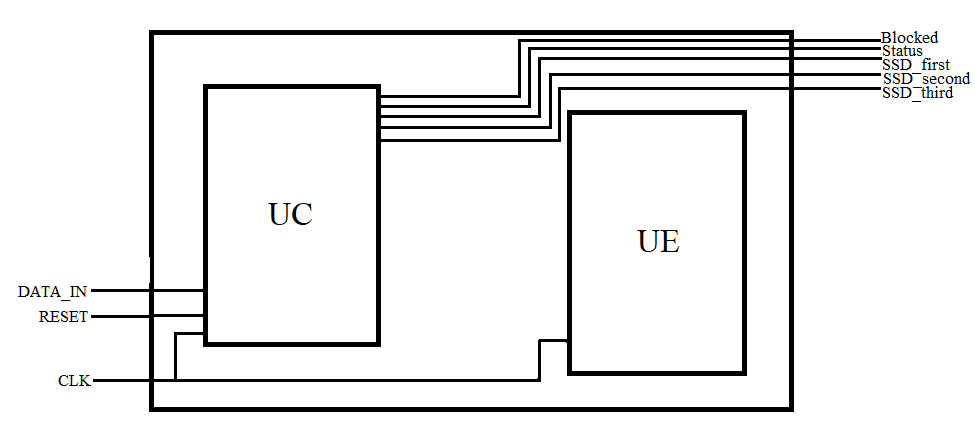


Figura 2. Maparea intrărilor și ieșirilor cutiei negre pe intrările și ieșirile componentelor din prima descompunere

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

* **Intrări de date**:
  + - secvențe binare pe 4 biți reprezentate de DATA\_IN;
* **Intrări de control:**
  + - butonul de resetare a stării, RESET;
    - ceasul, care emite semnale ce ceas folosite în circuit;
* **Ieșiri de date:** 
  + - cele trei Seven Segment Display, SSD\_first, SSD\_second și SSD\_third;
* **Ieșiri de control:**
  + - semnale de avertizare precum cele două LED-uri indicatoare pentru starea BLOCKED, respectiv STATUS.

### 2.2.2 Determinarea resurselor (UE)

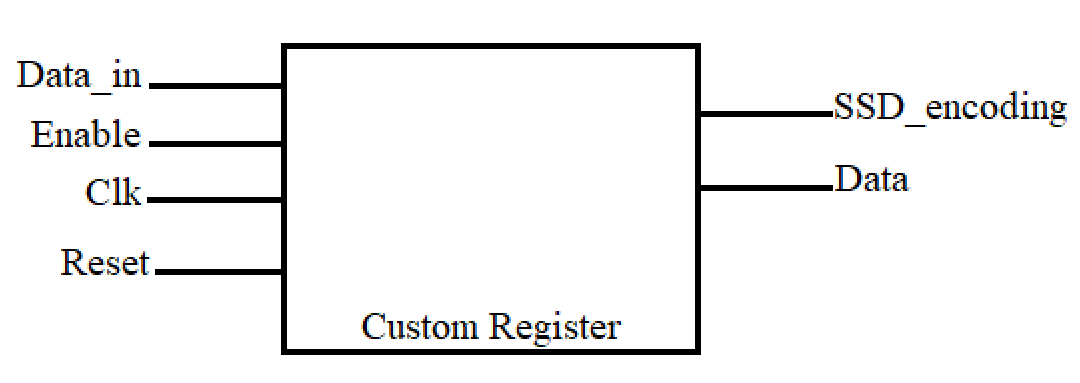
Pentru a stabili legăturile dintre UC și UE trebuie mai întâi **să identificăm resursele, pe baza cărora luăm decizii sau resursele care devin informații pentru utilizator.** *Un exemplu concret semnalul de resetare, controlat de unitatea de control.*

Orice informație pe baza căreia se iau decizii trebuie sa vină de la o resursă care generează acea informație și o transmite mai departe. *În această categorie este inclusă secvența binară de 4 biți transmisă prin DATA\_IN.*

**RESURSE:**

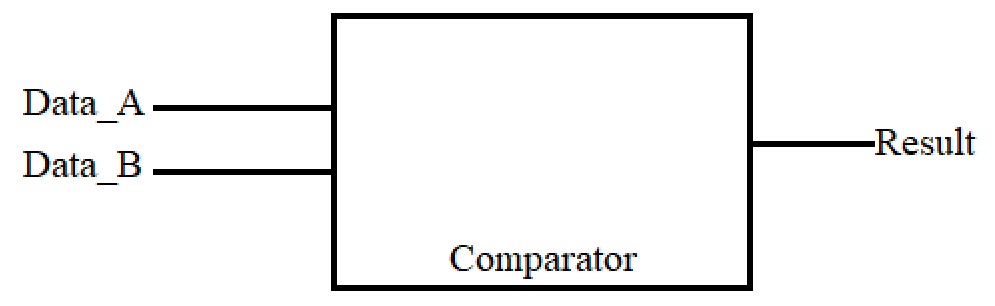
1. Registru

Acesta este un registru care primește datele cifrului pe o intrare de 4 biți și transmite la ieșire codul în format 7 segmente pe calea “SSD\_encoding”, respectiv reține datele de intrare pe calea “Data”. De asemenea, se dorește să se pornească sau să se oprească registrul la momentul potrivit, deci este nevoie de o intrare de “Enable”.



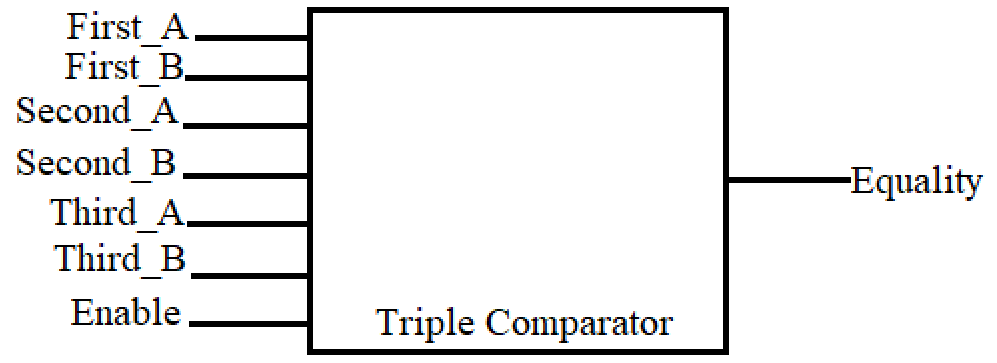
1. Comparator

Acesta este un comparator pe 4 biți, care va avea o ieșire “Result” cu valoarea adevărată sau falsă. Ieșirea va returna 1 atunci când au fost introduse 2 șiruri identice de biți în Data\_A și Data\_B, altfel valoarea va fi 0.



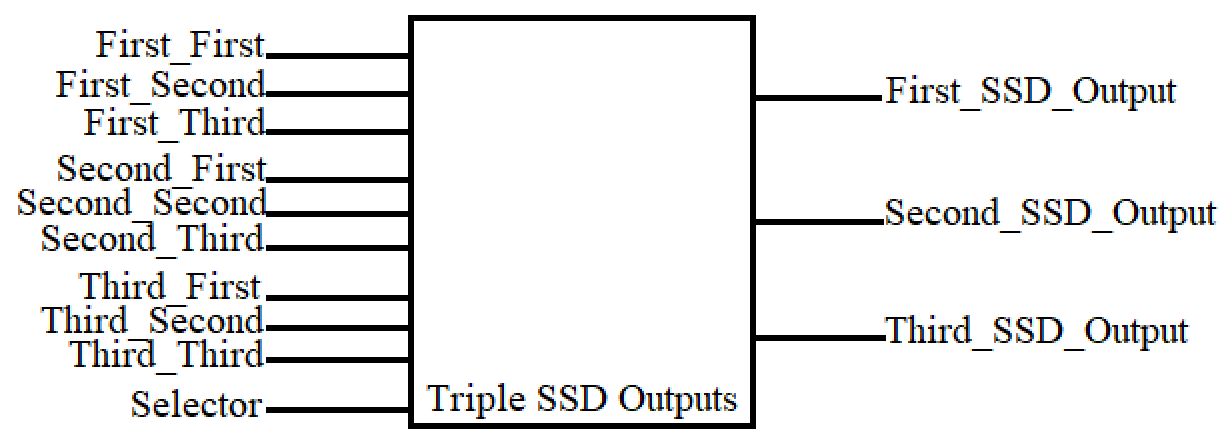
1. Comparator triplu

Acesta este un comparator pe 4 biți, format din trei comparatoare prezentate anterior, care va avea o ieșire “Equality” cu valoarea adevărată sau falsă. Comparatorul primește pe rând valorile inițiale ale cifrului și le compara cu valorile introduse după închiderea dulapului. Ieșirea va returna 1 atunci când numerele sunt identice, altfel valoarea va fi 0. De asemenea, este nevoie de o intrare “Enable” pentru a opri si a porni comparatorul la momentul introducerii unei noi încercări de deschidere a cifrului.



1. Selector triplu

Acesta este un selector triplu, care primește pe intrări datele fiecărei cifre a codului în format 7 segmente. De asemenea este necesară o intrare “Selector” pentru a printa pe rând ieșirile pe display, folosind “First\_SSD\_Output”, “Second\_SSD\_Output”, “Third\_SSD\_Output”.



### 2.2.3 Schema bloc a primei descompuneri

O imagine care conține diagramă, linie, Desen tehnic, Plan

Descriere generată automat

Figura 3 Schema Bloc cu legăturile dintre UC și UE mapate

2.2.4 Reprezentarea UC prin diagrama de stări (organigrama)

Diagrama de stări reprezintă partea de control, partea decizională din orice algoritm, care poate fi apoi implementată direct în VHDL.

* **Stările** reprezintă un moment de timp (o perioada) și sunt reprezentate prin



* **Deciziile** luate în fiecare stare sunt reprezentate prin romb. 
* **Ieșirile** generate în fiecare stare sunt reprezentate dreptunghiuri, în care se enumeră ieșirile care sunt adevărate în acel moment. 

Pentru implementarea organigramei avem mai multe metode la dispoziție:

* Descrierea comportamentala a organigramei folosind 3 procese
* Descrierea structurală a organigramei. Acest lucru presupune sinteza organigramei după metodele învățate în primul semestru. Aceasta metoda rezultă în continuarea descompunerii UC în componente.

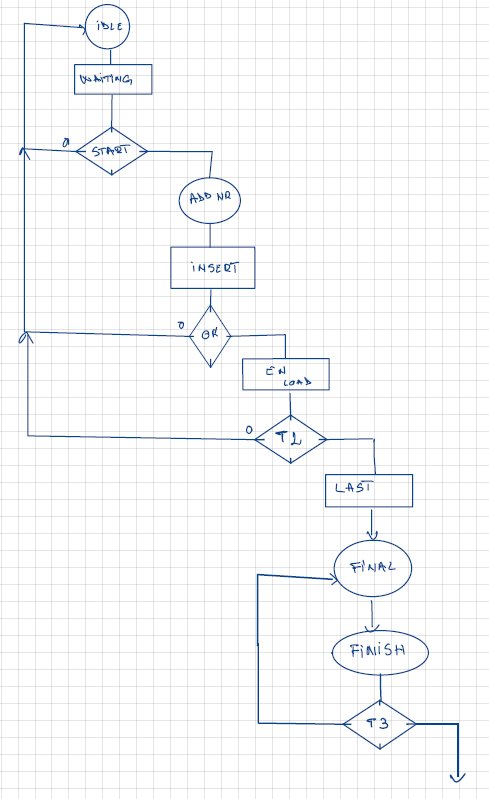


Figura 4 Organigrama unității de control

### Schema de detaliu a proiectului

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

Figura 5. Schema de detalii a proiectului conform implementării din Logisim Evolution

Pe această schemă unitatea de control apare ca și o component deoarece descrierea este una comportamentală. Prin intermediul aplicației Logisim Evolution, putem să scoatem în evidență mai bine relația dintre unitatea de execuție și unitatea de comandă.

Astfel, se poate observa cum elementele unității de comandă comunică cu unitatea de execuție și invers. Unitatea de execuție trimite LED-urile de output spre unitatea de comandă conform diagramelor anterior prezentate.

# Manual de utilizare și întreținere

Sistemul este conceput să funcționeze cât mai simplu, astfel încât utilizatorul să nu întâmpine probleme în utilizarea produsului. Prin urmare, pentru a utiliza in mod corect produsul, este necesar ca utilizatorul să introducă și să rețină un set de trei caractere cuprinse în intervalele 0-9 și A-F.

Introducerea codului se realizează caracter cu caracter. Stadiul de introducere a codului (numit de acum înainte cifru) se încheie odată ce al treilea caracter al cifrului a fost introdus și validat.

În continuare, sistemul așteaptă un set de caractere care coincid cu cifrul. Se introduc trei caractere care ulterior sunt verificate cu cifrul, la nivelul fiecărui caracter.

Exemplu de validare eronată:

Cifru:13A

Cod introdus pentru deblocare: 13B

|  |  |  |
| --- | --- | --- |
| Char. Cifru | Char. Cod | Status |
| 1 | 1 | **Egale** |
| 3 | 3 | **Egale** |
| A | B | **Diferite** |

Pentru ca cifrul să fie validat cu succes de către codul introdus, cele trei operații trebuie să fie „**Egale**”, drept urmare, codul introdus este considerat greșit.

Exemplu de validare corectă:

Cifru: 13A

Cod introdus pentru deblocare: 13A

|  |  |  |
| --- | --- | --- |
| Char. Cifru | Char. Cod | Status |
| 1 | 1 | **Egale** |
| 3 | 3 | **Egale** |
| A | A | **Egale** |

În urma verificării, toate cele trei rezultate sunt „**Egale**”, deci cifrul este validat.

În cazul validării corecte, se realizează o resetare a întregului sistem. Suprafața protejată de cifru se deblochează, iar indicatorul LED care afișează statusul, va lumina în culoarea verde, care indică faptul că sistemul este pregătit pentru o nouă folosire.

În cazul în care validarea este eronată, sistemul revine în stadiul în care așteaptă un cod pentru deblocare. Sistemul oferă utilizatorului 3 șanse de validare a cifrului, ulterior se blochează. Statusul de blocare este indicat de pornirea ledului de blocare, care va lumina în culoarea roșie.

În cazul în care sistemul se blochează, personalul calificat poate să realizeze o resetare totală a sistemului, prin apăsarea unui buton specific, resetând astfel cifru și deblocând suprafața protejată de sistem.

# Justificarea soluției alese

În conceperea soluției au fost folosite elemente destul de elementare din universul proiectării logice, cu scopul ca soluția finală să fie cât mai simplă. Există totuși și elemente compuse din circuite logice secvențiale și CLC-uri.

Elementele formate din CLS-uri și CLC-uri au ca rol simplificarea unității de execuție, ele fiind capabilă să producă un rezultat mai concis.

Exemplu: *Componenta custom\_register*

Componenta custom\_register este formată dintr-un registru și un decodificator din *cod binar* pe 4 biți, într-un semnal pe șapte biți, care este conceput pentru *transpunerea* codului binar pe un *Seven Segment Display*.

Al doilea output al componentei este chiar codul binar pe 4 biți, salvat în registru, care este mai departe folosit în diferite comparări cu alte secvențe binare de 4 biți, care au ca obiectiv final contribuirea la procesul de validare.

# Posibilități de dezvoltări ulterioare

O posibilă dezvoltare ulterioară este oferirea oportunității de a folosi un cifru biometric precum amprenta unui deget. Cât despre cifrul introdus fizic, s-ar putea realiza extinderea caracterelor de care sistemul dispune pentru introducerea cifrului.

Folosirea tastaturii pentru a introduce caracterele cifrului.

# Bullet-list

În realizarea proiectului au fost parcurși diferiți pași, în echipă. Am început printr-un brainstormig în care am împărțit diferite assignment-uri.

În ceea ce privește proiectul:

Diana:

1. MUX 4:1
2. D\_FLIP\_FLOP
3. Validarea numărului de încercări
4. Selector triplu

Luca:

1. hex\_to\_7\_seg
2. comparator\_triple
3. custom\_register
4. Unirea componentelor în unitatea de execuție
5. Fila main, care unește UE cu UC

Pentru documentație:

Diana:

1. Schema bloc
2. Capitolul legat de resurse
3. Justificarea soluției

Luca:

1. Manualul de utilizare
2. Specificații
3. Posibilități de dezvoltare

# Bibliografie

În realizarea proiectului au fost folosite diferite informații găsite pe canalul de Teams al cursului, inclusiv videoclipuri și fragmente de cod prelucrate în cadrul orelor de laborator.