3/20/2023

Aplicatie cod cifru pentru securizarea dulapurilor

Proiect PSN

NUME PRENUME: Dascalu Mircea, Ungureanu George-Octavian

Grupa : 30214

Îndrumător: ing. DIana Pop

Contents

[1 Specificații 2](#_Toc130249632)

[2 Proiectare 3](#_Toc130249633)

[2.1 Schema Bloc 3](#_Toc130249634)

[2.2 Unitatea de Control și Unitatea de Execuție 3](#_Toc130249635)

[1.2.1 Maparea intrărilor și ieșirilor cutiei mari pe cele două componente UC și UE. 3](#_Toc130249636)

[1.2.2 Determinarea resurselor (UE) 4](#_Toc130249637)

[2.2.3 Schema bloc a primei descompuneri 8](#_Toc130249640)

[2.2.4 Reprezentarea UC prin diagrama de stări (organigrama) 8](#_Toc130249641)

[2.2.5 Schema de detaliu a proiectului 12](#_Toc130249642)

[3 Manual de utilizare și întreținere 13](#_Toc130249643)

[4 Justificarea soluției alese 14](#_Toc130249644)

[5 Posibilități de dezvoltări ulterioare 15](#_Toc130249645)

[6 Bibliografie 16](#_Toc130249646)

Aplicatie cod cifru pentru securizarea dulapurilor

Specificații

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)

1. Un led LOCKED va avea functia de a semnala faptul ca dulapul este liber(led stins) sau ocupat(led aprins)

2. Utilizatorul va adauga pe rand 3 caractere cu ajutorul butoanelor a 3 switch- uri

3. Caracterele sunt curprinse in intervalul 0-1-..-8-9-A-B-..-F

4. Caracterul curent este afisat pe SSD

5. Pentru trecerea la urmatorul caracter utilizatorul va opri switch-ul curent si va porni swich-ul pentru urmatoarea cifra.

6. Caracterul anterior introdus ramane afisat

7. Urmatorul caracter este vizibil pe afisaj pe pozitia urmatoare

8. Dupa introducerea celui de al treilea caracter, la apasarea butonului CONFIRM, afisajul SSD se va stinge iar cifru va fi in starea blocat prin aprinderea ledului LOCKED.

9. Utilizatorul va apasa butonul/switch CONFIRM pentru a incepe introducerea codului pentru deblocarea cifrului

13. Se vor relua pasii 2-7

14. La introducerea ultimului caracter, la apasarea butonului CONFIRM se va face verificarea, daca codul introdus corespunde cu codul anterior.

15. In cazul de egalitate, ledul LOCKED se va stringe, afisajul SSD se goleste

16. In cazul de inegalitate, ledul LOCKED va ramane aprins, afisajul SSD se goleste.

# Proiectare

## Schema Bloc

LOCKED

Input\_1

Input\_2

ADMIN\_ON

Input\_3

Confirm

Admin

Figura 1 Cutia neagra a sistemului cu intrările și ieșirile stabilite

## 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*).

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

SSD

Locked

Admin\_ON

Input\_1

Admin

Confirm

Input\_3

Input\_2

U.E.

U.C.

Figura 2Maparea intrărilor și ieșisilor 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**: valori pentru diferite lucruri(cost bilet, distanța de parcurs,nr pin card, număr de bilete etc).
* **Intrari de control:** buton de confirmare, buton de selectare a unui program, buton de anulare etc.
* **Ieșiri de date:** valori de afișat pentru utilizator (timpul rămas dintr-un program, restul datorat, costul calculat al unui bilet etc.)
* **Iesiri de control:** semnale de avertizare sau atentionare a utilizatorului, prin care noi putem sa controlam și indrumam utilizatorul prin funcționarea sistemului.(leduri, semnal sonor)

### 2.2.2 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.

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.

Resursele pot fi **circuite simple,** care pot fi implementate direct (numarator, registru etc) sau **resurse complexe** (algoritm de dat rest, algoritm de inmultire, etc). Aceste resurse complexe pot apărea în prima descompunere cu cutii negre cărora trebuie sa le stabilim intrari si iesiri, dar ulterior trebuie descompuse mai departe (de obicei tot în UC și UE) pana cand ajungem la circuite cunoscute.

**RESURSE:**

1. Divizor de frecventa clk\_div - Ungureanu George-Octavian

Acesta este un divizor de frecventA care returneazA un ceas cu frecventa de 1Hz si cu factorul de umplere de 50%. Acesta ne va trebui la SSD.

clk clk\_out

2. Debouncer 20ns DEB\_3 - Dascalu Mircea

Aceasta resursa verifica daca un buton a fost apasat pentru o durata de 20 de nanosecunde, raspunsul fiind transmis prin parametrul Y.

CLK

Y

B

3. Registru de 1 bit REG - Dascalu Mircea

Acest registru retine valoarea de pe intrarea de 1 bit doar daca aceasta este 1 si are un reset pentru a face iesirea 0.

V\_IN

RST V\_OUT

CLK

4. Afisor SSD pentru 4 biti SSD16 - Ungureanu George Octavian

Afiseaza pe SSD valoare unui sir de 16 biti la o frecventa de 100MHz.

Clock\_100MHz

RESET Anode\_active

DATA LED\_out

5. Sistem de concatenare a 3 registrii CONCANATOR - Ungureanu George Octavian

Acest sistem concateneaza 3 siruri de 4 biti cu un sir “0000” pentru a crea un sir de lungime 16 biti pentru afisor.

REG1

REG2 OUTPUT\_REG

REG3

CLK

6. Numarator de la 0 pana la F NUMARATOR4 – Dascalu Mircea

Acest numarator afiseaza codul introdus de utilizator, crescandu-i-se valoarea la fiecare secunda.

RST Q

EN

CLK

7. Comparator de doua siruri de 16 biti COMPARATOR – Dascalu Mircea

Compara doua siruri de 16 biti si afiseaza 1 daca sunt egale

IN1

IN2 EG

EN

CLK

8. Registru de 16 biti REG\_MARE – Dascalu Mircea

Retine valoare introdusa si o introduce intr-un sir de 16 biti.

V\_IN

V\_OUT

EN

CLK

9. T-Flip-Flop T\_FF – Ungureanu George Octavian

Este un T-Flip-Flop ce face schimbarea intre introducerea parolei pentru blocare si apoi pentru deblocare.

Este asincron.

T

Q

RST

10. Dmux 1:2 DMUX\_2 – Ungureanu George Octavian

Face schimbarea intre doi registrii. Mai intai introduce intr-un registru parola introdusa pentru blocare , apoi in alt registru parola introdusa pentru deblocare.

intr out0

sel out1

11. Registru pentru palora de administrator REG\_ADMIN – Dascalu Mircea

Retine parola de administrator

V\_OUT

12. D-Flip-Flop BIS\_D – Dascalu Mircea

Reseteaza numaratoarele cand selectia e 1.

CLK Q

D

13. U.C.-Dascalu Mircea

Controleaza inputurile utilizatorului.

ADMIN

CIF1 SEL

CIF2 EN1

CIF3 EN2

Confirm EN3

CLK



### Schema bloc a primei descompuneri

**EN1**

U.E.

U.C.

**EN2**

**EN3**

**SEL**

Figura 3 Schema Bloc cu legăturile dintre UC și UE mapate

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

Diagrama de stări **nu este un flow-chart**, ci reprezintă partea de control, partea decizională din orice algoritm, și ea poate fi apoi implementată direct în VHDL dacă e facută corect.

* **Stările** sunt reprezentate prin . O stare reprezinta un moment de timp (o perioada).





* **Deciziile** luate în fiecare stare sunt reprezentate prin romb.
* **Ieșirile** generate în fiecare stare sunt reprezentate prin . În interiorul dreptunghiului se enumera ieșirile care sunt adevărate în acel moment.

O imagine care conține diagramă, model

Descriere generată automat

Figura 4 Organigrama unității de control

O imagine care conține diagramă, model

Descriere generată automat

O imagine care conține diagramă, origami, model

Descriere generată automat

### Schema de detaliu a proiectuluiO imagine care conține text, diagramă, Plan, Paralel Descriere generată automat

Figura 5 Schema de detalii a proiectului cpnform implementării din Logisim

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

# Manual de utilizare și întreținere

1. Pentru a introduce cifre utilizatorul va folosi pe rand primele 3 switch-uri de la stanga la dreapta:

( NU este admis codul ”000” )  
- Pentru a alege prima cifra se activeaza primul switch si se asteapta pana pe afisor apare cifra dorita apoi se inchide switch-ul.

* Pentru a pentru a alege a doua cifra se activeaza al doilea switch switch si se asteapta pana pe afisor apare cifra dorita apoi se inchide switch-ul.
* Pentru a pentru a alege a treia cifra se activeaza al treilea switch si se asteapta pana pe afisor apare cifra dorita apoi se inchide switch-ul.

1. Pentru a incuia dulapul se apasa pe butonul CONFIRM (cel din stanga)

* se va aprinde automat Led-ul de LOCKED pentru a sti ca dulapul este incuiat

1. Pentru a descuia dulapul utilizatorul va introduce din nou cifrul folosit pentru a-l incuia conform nr. 1. si se va apasa CONFIRM. Daca codul a fost bun dulapul se va descuia si Led-ul de LOCKED se va stinge.

Daca cifrul nu a fost corect dulapul va ramane incuiat si se asteapta introducerea altui cod.

1. In caz ca utilizatorul a uitat codul poate solicita asistenta. Cifrul poate fi pus in modul ADMIN\_ON prin apasarea butonului ADMIN. Astfel dulapul poate fi deschis cu un cod predefinit.

* Modul ADMIN\_ON este semnalat pin aprinderea unui nou led si pentru a parasi acest mod ori se apasa iarasi pe butonul ADMIN ori se deblocheaza dulapul.

# Justificarea soluției alese

# Posibilități de dezvoltări ulterioare

**Extinderea cifrului la 4 cifre**:

Acest lucru se poate realiza prin adaugarea unui numarator de acelasi tip cu cele folosite in acest proiect, si legarea acestuia la Concanatorul deja existent deoarece momentan se folosesc doar 12 biti din 16.

**Introducera cifrului folosind un keypad**:

Aceasta poate fi foarte utila mai ales daca am avea mai multe cifre.

Acest lucru pate fi implementat prin legarea la placa a unui keypad si prin decodificarea informatiei primite de la acesta, apoi introducerea informatiei direct in registre.

# Bibliografie

<https://www.fpga4student.com>