3/20/2023

Lift P+12 etaje

Proiect PSN

MATASĂ ALEXANDRA-VIORICA, MERCAȘ RALUCA-IOANA

Grupa: 30211

Îndrumător: DR. ing. DragoȘ FLORIN LIȘMAN

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 6](#_Toc130249640)

[2.2.4 Reprezentarea UC prin diagrama de stări (organigrama) 6](#_Toc130249641)

[2.2.5 Schema de detaliu a proiectului 9](#_Toc130249642)

[3 Manual de utilizare și întreținere 9](#_Toc130249643)

[4 Justificarea soluției alese 9](#_Toc130249644)

[5 Posibilități de dezvoltări ulterioare 9](#_Toc130249645)

[6 Bibliografie 9](#_Toc130249646)

Lift P+12 etaje

# Specificații

Să se proiecteze un automat care comandă un lift într-un hotel cu P+12 etaje. Liftul trebuie să răspundă solicitărilor persoanelor aflate în interior şi cererilor exterioare (sus, jos) care apar pe parcurs de la uşile aflate la fiecare nivel. Ordinea de onorare a cererilor ţine cont de sensul de mers (urcare sau coborâre). Se onorează cererile în ordinea etajelor, indiferent de unde provin ele (lift sau exterior). Liftul are o intrare care sesizează depăşirea greutăţii maxime admise şi nu porneşte în acest caz. Plecarea nu are loc dacă uşile nu sunt închise. Uşile trebuie să stea deschise un interval de timp programabil. Uşile nu se închid dacă există vreo persoană în uşă. Viteza liftului va fi selectabilă între două valori: 1 sau 3 secunde / etaj. Se consideră că în momentul iniţial liftul se găseşte la parter, cu uşile deschise.

Liftul va funcționa după cum urmează:

* Liftul se află inițial într-o stare de staționare, atât timp cât ușile sunt deschise.
* Starea liftului se schimbă în momentul în care apare o cerere. O persoană aflată în lift, selectează etajul la care dorește să meargă, și se stabilește sensul de mers (0 pentru coborâre și 1 pentru urcare), ușile se închid și liftul pornește spre etajul respectiv.
* Daca sensul de mers stabilit este de coborâre, liftul se deplasează spre etajele inferioare. În cazul în care primește o cerere din exterior care corespunde cu sensul de mers, onorează cererea. În momentul în care ajunge la destinație, ușile se deschid, iar liftul trece din nou în starea 1, de staționare.
* Daca sensul de mers stabilit este de urcare, liftul se deplasează spre etajele superioare. În cazul în care primește o cerere din exterior care corespunde cu sensul de mers, onorează cererea. În momentul în care ajunge la destinație, ușile se deschid, iar liftul trece din nou în starea 1, de staționare.

Functionare:

Liftul initial se afla in starea 1, la parter (A) cu usile deschise. In acest moment, orice cerere exterioara ar primi, o accepta (se scrie cererea in memoria RAM). Acesta este momentul in care se stabileste directia de deplasare a liftului. Cererea exterioara consta in specificarea unei directii (up sau down) si a unui etaj (B). Cererea exterioara se scrie in memorie. Persoana intra in lift, iar cererea exterioara se transforma in una interioara. la fiecare etaj la care se opreste intre etajul A si etajul B, se verifica:

1. Exista vreo cerere exterioara noua? Daca da, se scrie in memorie cererea (directie+etaj). Daca nu, nu se face nimic.

2. Daca in drumul de la A la B exista in memorie o cerere pentru un etaj > A si < B, iar directia coincide cu cea a liftului, liftul face o oprire la etajul respectiv. Daca nu exista, isi vede de drum.

Cand liftul a onorat cererea curenta, persoana iese din lift si cererea este stearsa din memorie.

Cand liftul a urcat pana la etajul 12, directia de deplasare se schimba si liftul coboara. cauta in memorie etajul la care trebuie sa mearga. verifica in memorie daca trebuie sa faca opriri pe drum, conform punctului 2. Si cand se opreste, verifica conform punctului 1.

Memorarea cererilor se face intr-o memorie RAM, sub forma unei liste. (lista poate contine intre 5 si 10 cereri

# Proiectare

## Schema Bloc

ETAJ

USA\_D

PANOU

OK

GREUTATE\_D

GREUTATE

AFISOR

PERS\_IN\_USA

RESET

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*).

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

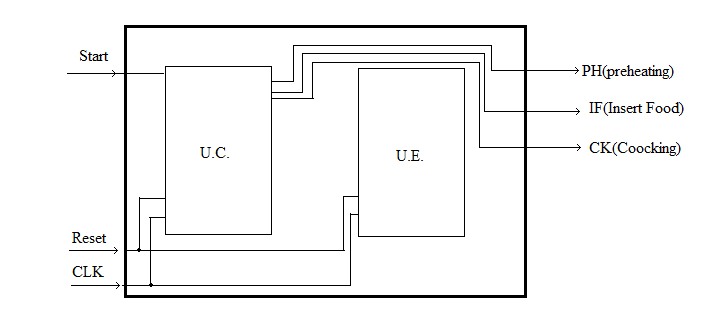


Figura 2: Maparea intrărilor și ieșisilor cutiei negre pe intrările și ieșirile componentelor din prima descompunere

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

* **intrări de date**:

ETAJ (4 switch-uri) – intrare pe 4 biti prin care se introduce in binar etajul la care a aparut o cerere. Daca cererea apare din interior, ETAJ reprezinta numarul etajului la care liftul trebuie sa se deplaseze pentru a lasa persoana. Daca cererea apare din exterior, ETAJ reprezinta numarul etajului la care exista o persoana care a apasat pe butonul de chemat liftul, iar liftul trebuie sa se deplaseze intr-acolo pentru ca persoana sa se poata urca.

PANOU (switch) – intrare pe 1 biti prin care se stabileste de la ce panou s-a primit cererea: 0 - panou interior, 0- panou exterior

* **Intrari de control:**

OK (buton) – intrare activa cand cererea a fost introdusa

GREUTATE (buton) – senzor activ cand a fost depasita greutatea maxim admisa

PERS\_IN\_USA (buton) – señzor activ cand este o persoana in usa

RESET (buton) – intrare pentru initializarea/ resetarea liftului

* **Ieșiri de date:**

AFISOR (afisor 7 – segmente) – iesire pe 7 biti care tine evidenta etajului curent al liftului

* **Iesiri de control:**

USA\_D (led) – senzor care se aprinde atunci cand usile liftului un se pot inchide din cauza unei persoane care sta in usa

GREUTATE\_D (led) – senzor care se aprinde atunci cand greutatea permisa este depasita

### 1.2.2 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.

R**ESURSE:**

**Senzor de greutate**:

Aceasta resursa are intrarea GREUTATE si genereaza iesirea GREUTATE\_D. Daca GREUTATE\_D = 1 inseamna ca s-a depasit greutatea maxima si se aprinde ledul corespunzator.

GREUTATE GREUTATE\_D

Senzor de greutate

**Senzor de persoana in usa**:

Aceasta resursa are intrarea PERS\_IN\_USA si genereaza iesirea USA\_D. Daca USA\_D = 1 inseamna ca exista o persoana in usa si se aprinde ledul corespunzator.

PERS\_IN\_USA USA\_D

Senzor de persoana in usa

**Decodificator BCD\_7 segmente**:

Acest circuit va avea o intrare hex\_in, pe 4 biti, care este numarul care trebuie afisat pe un afisor, reprezentat in binar, si semnalul de tact clk. De asemenea, avem iesirea abcdefg, pe 7 biti. Avem nevoie de un singur afisor care sa afiseze cifrele 0-9 si literele A, B, C, pentru etajele 10, 11, respectiv 12 (exprimate in hex).

hex\_in

Decodificator

abcdefg

clk

**Afisor 7 segmente:**

Aceasta componenta se va folosi la afisarea etajele. Afișoarele au cele 7 leduri reprezentând segmentele. Fiecare afisor are un anod activ pe '0' logic. Toate cele 4 afisoare sunt legate la un anod comun, ceea ce inseamna ca pe fiecare afisor se va afisa acelasi lucru. Pentru a afisa informatii diferite pe afisoare trebuie sa activam pe rand cate un anod, in timp ce ceilalti se dezactiveaza.

**Numarator reversibil:**

In entitatea numaratorului am declarat intrarile: semnalul de tact CLK, RESET, ENABLE si Y\_REZ (vector pe 13 biti care contine toate cererile aparute pana in acel moment).

Daca ENABLE = 1, numaratorul este activ pentru numarare, iar daca ENABLE = 0 numaratorul intra in functia de HOLD a numarului curent (folosit atunci cand liftul stationeaza la un etaj).

Daca RESET este activ, se va afisa valoarea “0”.

Pentru a tine evidenta etajului curent la care se afla liftul, ne va trebui o variabila ajutatoare etaj\_curent, care se initializeaza la inceput cu 0, deoarece liftul se afla initial la parter. Pentru a retine sensul de deplasare al liftului, avem nevoie de o variabila ajutatoare, numita SENS, care initial este 0 deoarece liftul trebuie sa urce de la parter spre etajele superioare. (SENS = 0 => urcare; SENS = 1 => coborare).

Daca ENABLE este activ (‘1’) se vor verifica urmatoarele:

* Daca se numara crescator (SENS = 0), se numara in continuare de la “etaj\_curent” pana la prima cerere din vectorul “Y\_REZ” aflata la un etaj superior. Daca nu mai avem cereri la etaje superioare, incepe sa se numere descrescator (SENS = 1).
* Daca se numara descrescator (SENS = 1), se numara in continuare de la “etaj\_curent” pana la prima cerere din vectorul “Y\_REZ” aflata la un etaj inferior. Daca nu mai avem cereri la etaje inferioare, incepe sa se numere crescator (SENS = 0).

Fie X- etajul la care am avut cererea la care trebuie sa ne deplasam. Initial liftul se afla in starea de hold (ENABLE = 0). In aceasta stare se cauta etajul X in vectorul Y\_REZ. Cat timp se numara de la “etaj\_curent” pana la X, ENABLE = 1. Cand am indeplinit cererea, etaj\_curent ia valoarea lui X, iar ENABLE devine ‘0’, deoarece numaratorul intra in starea de hold, moment in care pe pozitia X din vector se pune 0 si se cauta urmatoarea valoare a lui X.

Aceasta logica se repeta pana cand am epuizat toate cererile, adica pana cand vectorul Y\_REZ este gol.

Spre exemplu, luam situatia cand liftul se afla pe sensul de urcare. Daca liftul este la etajul 7 si nu mai exista cereri la un etaj superior (>7), atunci sensul de deplasare se schimba automat, in loc sa numere in continuare pana la 12, ca abia apoi sa inceapa sa numere descrescator.

De asemenea, iesirea COUNT este pe 4 biti, reprezentand valoarea etajului curent la care se afla liftul.

CLK

Numarator

reversibil

RESET

COUNT

ENABLE

Y\_REZ

**Bistabil JK**:

In entitatea bistabilului am declarat intrarile: ENABLE, CLK (semnal de tact), J (cu rol de set), K (cu rol de reset) si iesirea Q (toate pe un bit). In implementare vom folosi o variabila ajutatoare x. Aceasta resursa ne va ajuta la formarea panoului interior si panoului exterior al liftului. Componenta functioneaza dupa cum urmeaza:

* Cand intrarea ENABLE este ‘1’ (activa), se activeaza bistabilul si variabila x se va pune pe iesirea Q. In caz contrar, bistabilul este inactiv.
* Daca J = 0 si K = 0, se pastreaza starea anterioara a bistabilului
* Daca J = 0 si K = 1, iesirea Q va lua valoarea 0
* Daca J = 1 si K = 0, iesirea Q va lua valoarea 1
* Daca J = 1 si K = 1, se neaga starea anterioara a bistabilului

Fiecare iesire a decodificatorului 4:13 va corespunde unei intrari J a unui astfel de bistabil.

Daca iesirea decodificatorului este 0, J-ul bistabilului corespunzator va fi ‘0’:

* Daca starea anterioara a bistabilului a fost ‘0’ (nu au existat pana in acel moment cereri la etajul respectiv), atunci K trebuie sa fie 0 pentru ca starea bistabilului sa se pastreze tot ‘0’.
* Daca starea anterioara a bistabilului a fost ‘1’ (a existat pana in acel moment o cerere la etajul respectiv), atunci K trebuie sa fie 0 pentru ca starea bistabilului sa se pastreze tot ‘1’

Daca iesirea decodificatorului este 1, J-ul bistabilului corespunzator va fi ‘1’:

* Daca starea anterioara a bistabilului a fost ‘0’ (nu au existat pana in acel moment cereri la etajul respectiv), atunci K poate sa fie ‘0’ sau ‘1’ pentru ca starea bistabilului sa devina ‘1’
* Daca starea anterioara a bistabilului a fost ‘1’ (a existat pana in acel moment o cerere la etajul respectiv), atunci K trebuie sa fie ‘0’ pentru ca statea bistabilului sa se pastreze ‘1’

ENABLE

Bistabil JK

CLK Q

SET

RESET

**Decodificator 4:13**

Aceasta componenta primeste o intrare pe 4 biti, numita ETAJ si are 13 iesiri, fiecare pe cate un bit. Unul din cei 13 biti se seteaza in functie de valoarea data pe intrare (cererea de la etajul ETAJ al liftului). Fiecare iesire pe un bit corespunde celor 13 intrari ale DMUX-ului 1:2 pe 13 biti.

D0

ETAJ(0)

0

0

D1

1

ETAJ(1)

1

D2

ETAJ(2)

2

2

D3

3

D4

ETAJ(3)

4

3

Decodificator 4:13

D5

5

D6

6

D7

7

D8

8

D9

9

D10

10

D11

11

D12

12

**Panou interior (grup de bistabile JK):**

Aceasta resursa memoreaza cererea venita de pe panoul interior al liftului. Panoul interior cuprinde 13 butoane (parter + butoanele de la etajele 1-12), ceea ce inseamna ca avem nevoie de 13 bistabile pentru a memora starea fiecarui buton in parte (1 daca a fost apasat si 0 daca nu a fost apasat).

De exemplu, cand o persoana din interiorul liftului apasa butonul aferent etajului 5, se seteaza bitul cu numarul 5 din vectorul de iesire Y0. Daca o a doua persoana apasa butonul aferent etajului 8, se seteaza si bitul cu numarul 8 din vector. Astfel, in vectorul Y0 se tine evidenta cererilor aparute din interiorul liftului. Cu alte cuvinte, memoreaza la ce etaje trebuie sa opreasca pentru ca o persoana sa se coboare din lift.

Dupa indeplinirea unei anumite cereri, bitul corespunzator din vector se reseteaza (devine din nou 0).

Acest circuit va avea intrarile: ENABLE (activ pe ‘1’ cand trebuie sa se memoreze cererea), semnalul de tact CLK, o intrare ETAJ pe 1 bit (care reprezinta cererea aparuta, adica singurul bit setat care iese pe una din iesirile decodificatorului) si RESET, care permite circuitului sa stearga cererea dupa ce a fost indeplinita (sa reseteze bitul corepunzator la 0). De asemenea, resursa are iesirea Y0, un vector pe 13 biti, fiecare bit reprezentand starea unui bistabil din cele 13 bistabile RS corespunzatoare etajelor P+12.

Y0

RESET

ETAJ



CLK

ENABLE



### 

Panou interior

**Panou exterior (grup de bistabile JK):**

Aceasta resursa memoreaza cererea venita de la butoanele de pe panoul exterior al liftului. La fiecare etaj, exista cate un buton exterior, prin care o persoana poate cheama liftul. Acest circuit memoreaza cererile aparute din exterior in vederea opririi liftului la etajele respective, pentru ca persoanele respective sa se poata urca.

In exterior exista doar un buton pe etaj pentru chemarea liftului, nu si pentru selectarea directiei de deplasare. Astfel, din exterior nu se poate selecta directia sus/jos in care persoana doreste sa se deplaseze. Acest lucru se stabileste doar ulterior, dupa ce persoana urca in lift si apasa butonul cu numarul etajului dorit de pe panoul interior.

Indeplinirea unei cereri la etajul X este echivalenta cu faptul ca liftul a facut o oprire la etajul X (se afla in hold) si persoana care a initiat cererea coboara. Dupa indeplinirea cererii, bitul corespunzator etajului curent al vectorului se reseteaza la ‘0’.

Circuitul va avea intrarile: ENABLE (activ cand trebuie sa se memoreze cererea), semnalul de tact CLK, o intrare pe un bit numita ETAJ si RESET-ul, care permite circuitului sa stearga cererea indeplinita. De asemenea, circuitul are iesirea Y1, un vector pe 13 biti. Panoul exterior, la fel ca si cel interior, este format din 13 bistabile, fiecare bistabil fiind responsabil de cate un etaj.

ENABLE

CLK

Panou exterior

Y1

ETAJ

RESET

In urma utilizarii resurselor „Panou interior“ si „Panou exterior“ rezulta doi vectori de dimeniunea 13, primul responsabil cu memorarea etajelor corespunzatoare cererilor din interior, iar al doilea cu memorarea cererilor din exterior. Pentru a onora ambele tipuri de cereri, liftul trebuie sa faca acelasi lucru: sa se deplaseze la etajul respectiv pentru ca persoanele sa urce sau sa coboare din lift, dupa caz. Pentru a simplifica modul de lucru, vom construi un al treilea vector de dimensiune 13, care sa contina ambele tipuri de cereri. Acest lucru se poate realiza facand operatia SAU intre bitii cu acelasi indice din cei doi vectori.

Exemplu:

avem cererile interioare pentru etajele 2, 5, 6, 9 si cererile exterioare de la etajele 1, 3, 5, 9, 10.

Vectorul 0:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

Vectorul 1:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

Se realizeaza operatia SAU pe biti intre vectorul 1 si vectorul 2.

Vectorul rezultat:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

Sa presupunem ca liftul stationeaza la etajul 4.

Daca sensul liftului este de urcare, atunci vom verifica ce biti sunt setati din vectorul rezultat, de la indicele 4 pana la 12 (cerererile de la etajele 1, 2, 3 sunt ignorate momentan deoarece pentru a le onora trebuie ca liftul sa coboare, iar acest lucru nu coincide cu sensul de deplasare al liftului).

Astfel, liftul se va opri la etajul 5 (bitul de 1 se va reseta => va deveni „0“ ), apoi se va opri la etajul 6 (bitul de 1 se va reseta), apoi se va opri la 9 (bitul se va reseta), apoi la 10 (bitul se va reseta). Deoarece dupa etajul 10 nu mai exista nicio cerere in vectorul rezultat (adica nici un bit de 1), sensul se schimba automat. Liftul incepe sa coboare, onorand cererile de la etajele 3, 2, 1, in aceasta ordine, resetand bitii corespunzatori la 0.

**Poarta SAU:**

Aceasta componenta are rolul de a construi vectorul rezultat, care contine toate cererile aparute, indiferent de natura lor. Astfel, avem intarile:

* Y0, vector pe 13 biti, in care sunt memorate cererile din exterior (provenit de la iesirea componentei „Panou exterior“)
* Y1, vector pe 13 biti, in care sunt memorate cererile din interior (provenit de la iesirea componentei „Panou interior“)

Iesirea o reprezinta vectorul pe 13 biti Y\_REZ, obtinut in urma operatiei SAU facute intre bitii cu indici corespunzatori din vectorii Y0 si Y1.

Y0

Y\_REZ

Poarta SAU

Y1

DEMULTIPLEXOR 1:2 pe 13 biti:

Demultiplexorul are o selectie pe un bit numita PANOU, cu ajutorul careia se decide pe care panou se inregistreaza cererea pusa pe intrari (PANOU = ‘0’ => iesirile DMUX-ului vor corespunde celor 13 bistabile din panoul interior; PANOU = ‘1’ => iesirile DMUX-ului vor corespunde bistabilelor panoului exterior). Intrarea demultiplexorului este iesirea decodificatorului (numarul etajului reprezentat pe 13 biti). Selectia demultiplexorului o reprezinta intrarea PANOU, cu ajutorul careia se decide carui panou corespunde cererea aparuta.

STROBE

D(0)

BI0

D(1)

BI1

D(2)

BI2

D(3)

BI3

D(5)

D(4)

BI5

BI4

D(6)

BI6

D(7)

BI7

D(8)

BI8

D(9)

BI9

DEMULTIPLEXOR 1:2 pe 13 biti

D(10)

BI10

D(11)

BI11

PANOU

BE12

BE11

BE10

BE9

BI0

BE8

BI0

BE7

BI0

BE6

BE5

BI0

BE4

BI0

BE3

BE2

BE1

BE0

D(12)

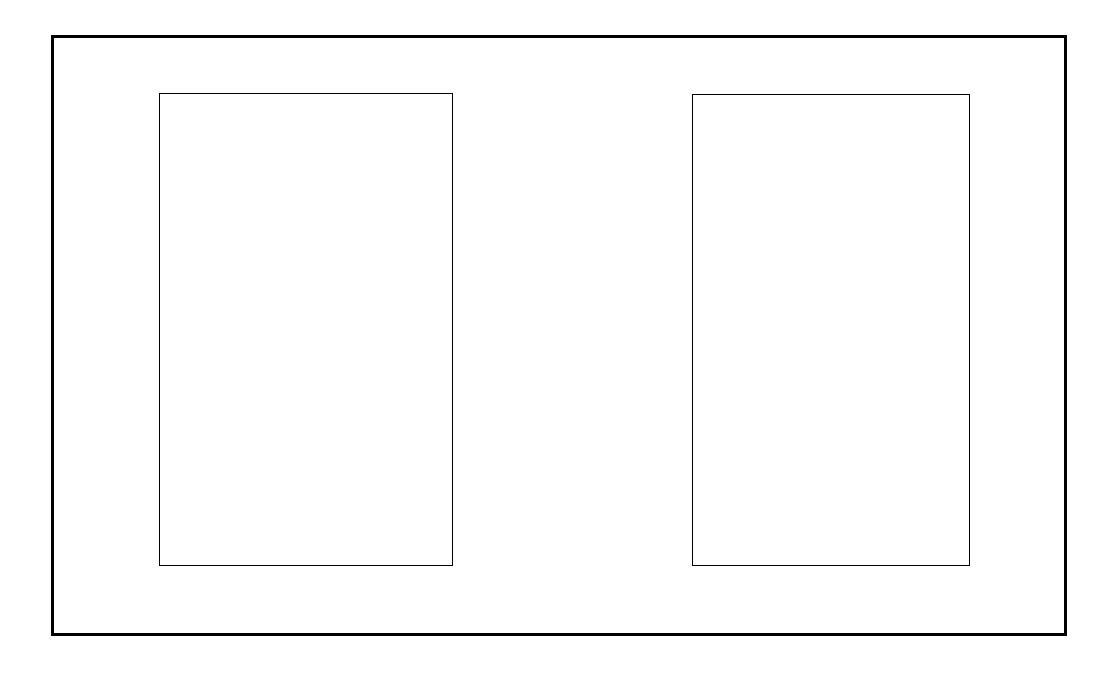
BI12

Intrarile PERS\_IN\_USA, GREUTATE si OK formeaza o logica de control care este conectata la intrarea STROBE a demultiplexorului. Astfel, cand senzorii PERS\_IN\_USA si/sau GREUTATE sunt activi, demultiplexorul se dezactiveaza si se activeaza cand sunt inactivi, iar OK este `1`.

POARTA SAU-NU pe 1 bit:

Aceasta componenta are ca intrari senzorii PERS\_IN\_USA si GREUTATE. Daca cel putin una din aceste intrari este activa atunci iesirea I va fi `0`, in caz contrar va fi `1`.



**

PERS\_IN\_USA

GREUTATE

- Numarator

- Comparator ?

- Multiplexor ?

- Demultiplexor ?

- Porti logice SAU

-Decodificator 4:17

- Decodificator

- Afisor 7 segmente

U.E.

USA\_D

GREUTATE\_D

AFISOR

- Panou interior

- Panou exterior

- Senzor de greutate

- Senzor de om in usa

- Organigrama

U.C.

PANOU

ETAJ

RESET

CLK

Figura 3 Schema Bloc cu legăturile dintre UC și UE mapate

**NOTA:** E de menționat faptul că CLK are o frecvență mai mare la intrare (100MHz), și ar trebui să fie un divizor de frecvență în cadrul UE ca o alta resursă pentru a genera perioada de 1 min pentru cele 2 numaratoare.

### 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.

Aceste ieșiri pot să fie pozitionate **înainte** sau **după decizie**. Dacă sunt înainte de decizie, înseamnă ca ieșirile depind numai de starea internă, deci avem un automat **Moore**. Dacă sunt după decizie, înseamnă că ieșirea depinde și de starea internă și de intrări, deci avem un automat **Mealy**.

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. Dacă se alege această metodă, schema de detaliu a UC trebuie adăugată în documentație împreuna cu toți pașii de sinteză făcuți la acest capitol, următorul subpunct  *1.2.5 Sinteza și schema de detaliu a organigramei*

Graphical user interface, application, Teams

Description automatically generated

Figura Organigrama unității de control

### Diagram, schematic Description automatically generatedSchema de detaliu a proiectului

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. Dacă se decide pentru sinteza organigranei atunci schema de detaliu va conține și schema UC.

# Manual de utilizare și întreținere

# Justificarea soluției alese

# Posibilități de dezvoltări ulterioare

# Bibliografie