Proiectarea Sistemelor Numerice

Tema Proiect: Lift Parter + 12 etaje

Profesor coordonator: Diana Irena Pop

Studenti: Silviu Grumazescu, Andrei-Denis Alexandru

Anul 1, Seria A, Grupa 30211

Cuprins

1. Specificatie proiect
2. Descriere schema bloc.

2.1. Schema bloc

2.2. Intari

2.3. Iesiri

1. Organigrama
2. Lista componentelor folosite

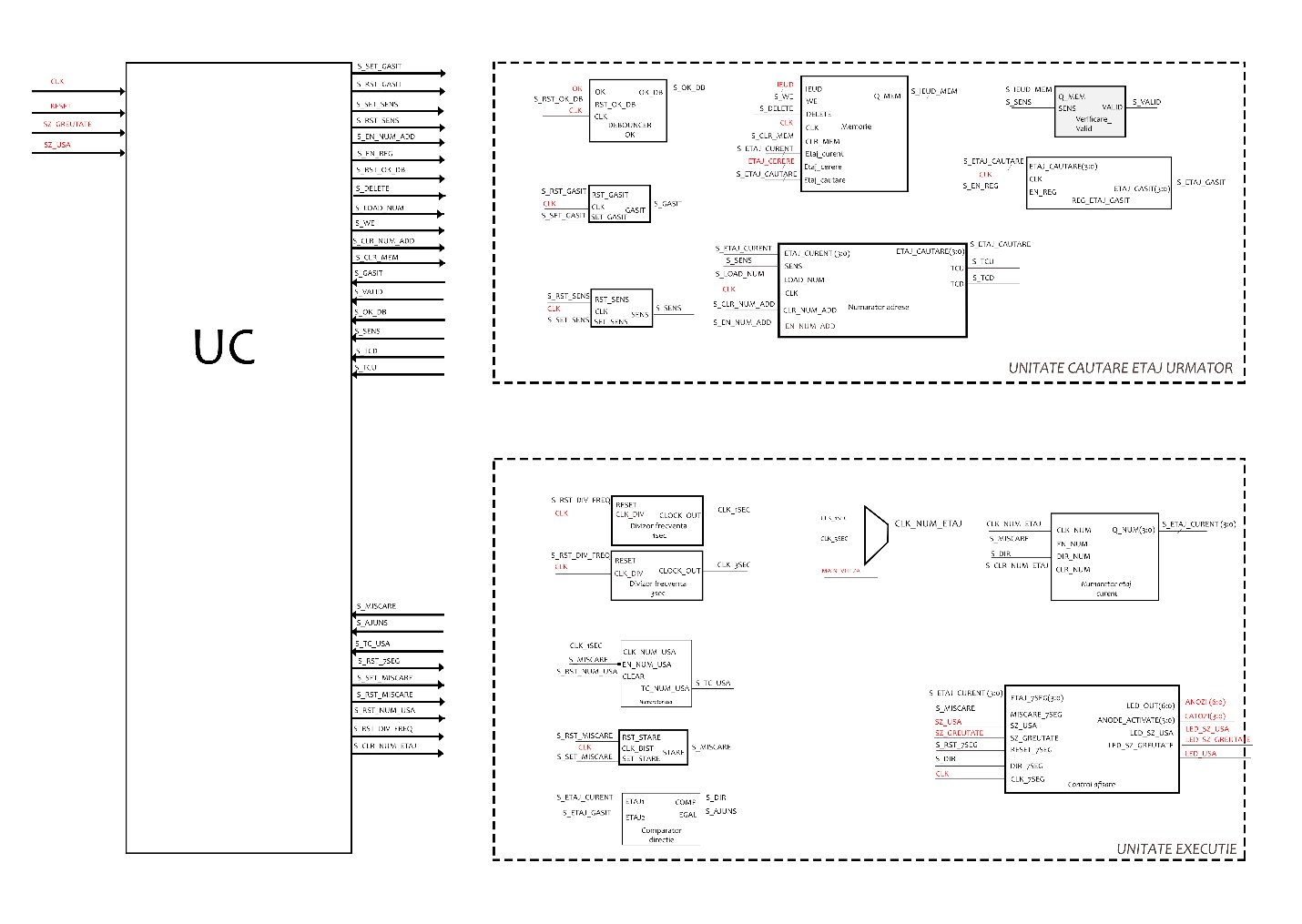
4.1 Unitatea de control

4.2 Unitatea de executie

4.3 Unitatea auxiliara

1. Proiectare, implementare si justificarea solutiei alese
2. Instructiuni de utilizare
3. Posibilitati de dezvoltare
4. Bibliografie
5. Specificatie proiect

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.

1. Descriere schema bloc
   1.  Schema bloc
   2. Intrari

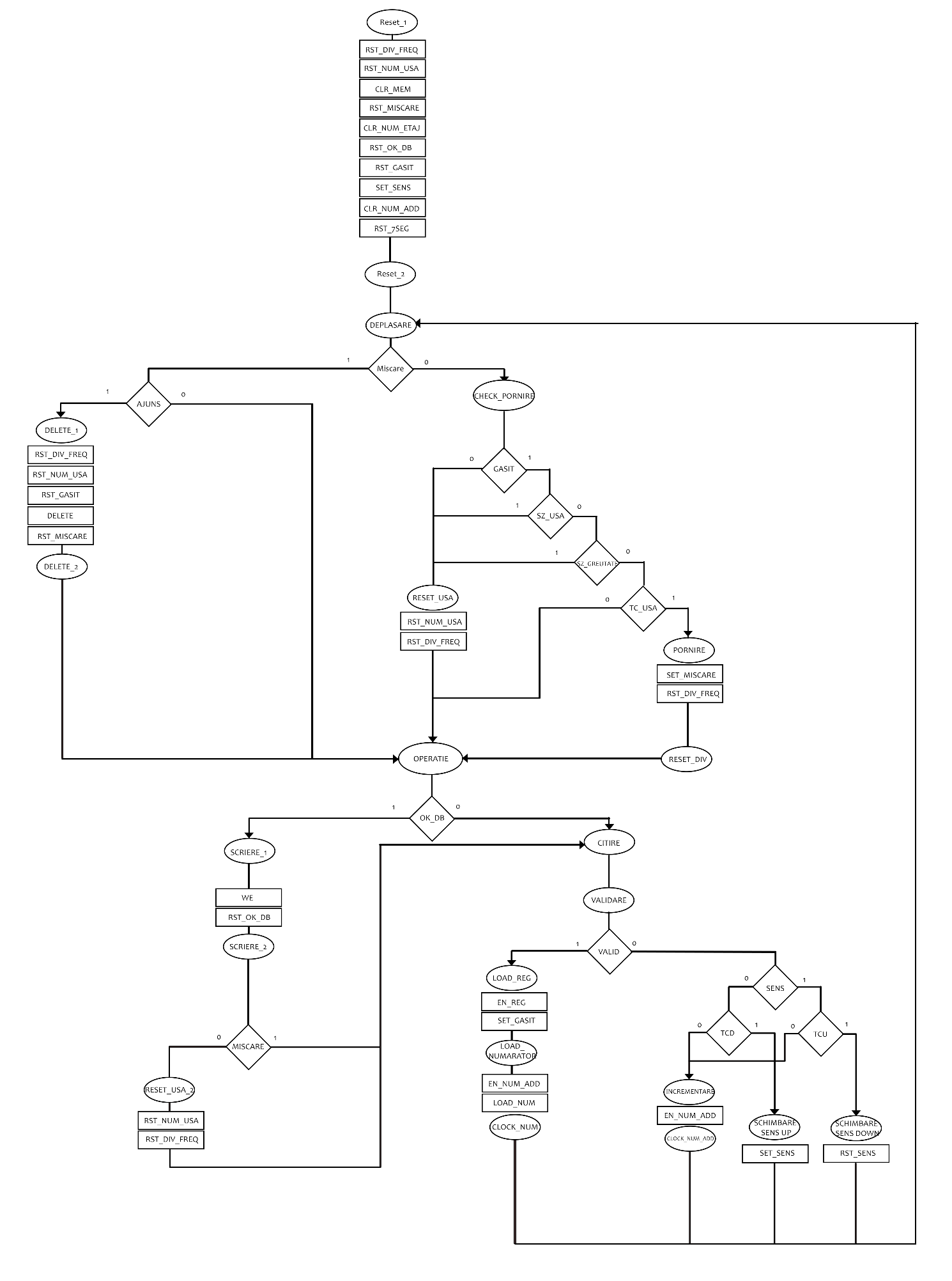
* CLK (1 bit) – Reprezinta intrarea clock-ului liftului
* MAIN\_VITEZA (1 bit) – Selectorul de viteza : 0 pentru viteza de 1 secunda / etaj (rapid) si 1 pentru viteza de 3 secunde / etaj (lent)
* SZ\_GREUTATE (1 bit) – 1 daca greutatea maxima este depasita si 0 in caz contrar
* SZ\_USA (1 bit) –1 daca o persoana blocheaza usa si 0 in caz contrar
* IEUD( 4 biti) – Intrare care specifica tipul cererii : Interior - Exterior – Up – Down
* ETAJ\_CERERE(4 biti) – Reprezinta numarul etajului de la care s-a facut cererea
* OK (1 bit) – Semnaleaza trimiterea unei cereri catre lift
  1. Iesiri
* ANOZI (7 biti) & CATOZI (4 biti) – Controleaza cele 4 afisoare pe 7 segmente pentru reprezentarea etajului curent si a directiei de mers
* LED\_SZ\_USA (1 bit) – LED ce semnaleaza daca usa este blocata
* LED\_SZ\_GREUTATE( 1 bit ) – LED ce semnaleaza daca greutatea maxima este sau nu depasita
* LED\_USA (1 bit) – LED ce semnaleaza daca usa este sau nu deschisa ( aprins cand usa este deschisa )

(Silviu & Denis)

1. Organigrama

Urmatoarea organigrama prezinta pasii pe care ii face automatul pentru functionarea sa corecta. Initial, liftul se gaseste la parter, cu usile deschise, intr-o stare de repaus.

(Silviu & Denis)



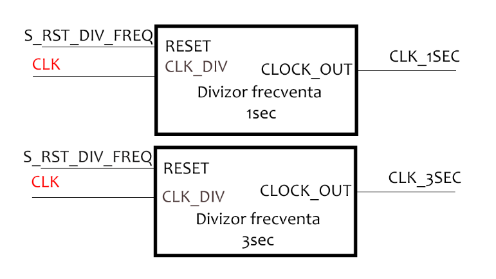
1. Lista componentelor folosite
   1. O imagine care conține masă

      Descriere generată automatUnitatea de control

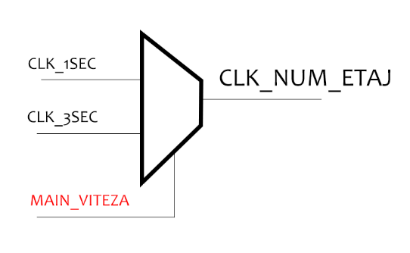
Unitatea de control reprezinta implementarea organigramei care asigura sincronizarea tutoror elementelor din unitatea de executie si unitatea auxialiara. (Silviu & Denis)

* 1. Unitatea de executie

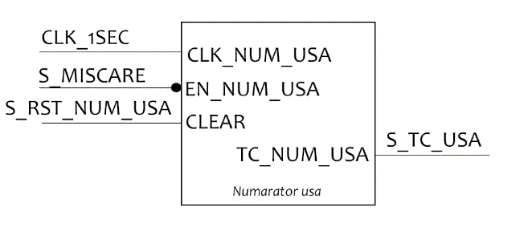
(Denis)

* 2 Divizoare de frecventa pentru clock

Numaratorul pentru contorizarea etajului curent si numaratorul pentru temporizarea deschiderii usii se folosesc de un clock pe o frecventa de 1 sau 3 secunde, iar clock-ul placii functioneaza pe o frecventa de 100 Mhz, de aceea am avut nevoie de doua divizoare de frecventa.

* Multiplexor pentru viteza

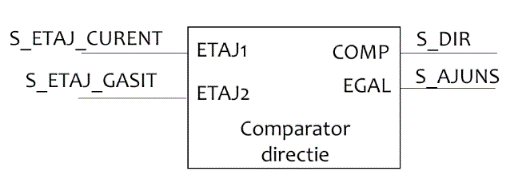
Viteza liftului va fi selectata cu ajutorul unui multiplexor, a carui selectie va fi intrarea de viteza, iar intrarile vor fi iesirile divizoarelor prezentate anterior.

* Numarator ușă

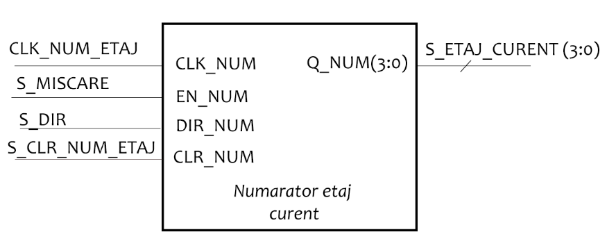
Acest numarator are rolul de a determina durata de timp in care usile sunt deschise. In cazul in care o persoana blocheaza usa sau este depasita greutatea maxima admisa, acest numarator se va reseta. Intrarea de enable este data de semnalul **S\_MISCARE**, ce semnaleaza starea de repaus sau miscare a liftului. Iesirea **TC\_NUM\_USA** va deveni 1 dupa 5 secunde de la ultimul reset, daca intrarea de enable este activa.

* Bistabil pentru starea de miscare

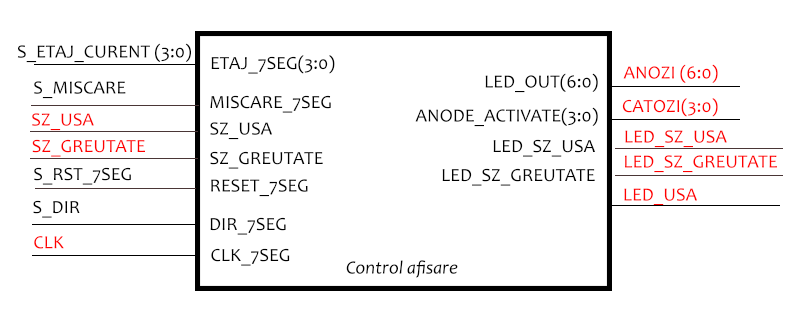
Acest bistabil RS este controlat de unitatea centrala, in care se va stoca starea curenta a liftului, fie de miscare ( 1 ) , fie de repaus ( 0 ).

* Comparator pentru determinarea directiei

Directia liftului va fi data de pozitia etajului curent fata de cel gasit ( **COMP** <= 1 pentru ETAJ1 < ETAJ2, 0 in caz contrar) , iar in cazul in care acestea sunt egale, iesirea **EGAL** va lua valoarea 1 si va semnala ca liftul a ajuns la destinatia ceruta.

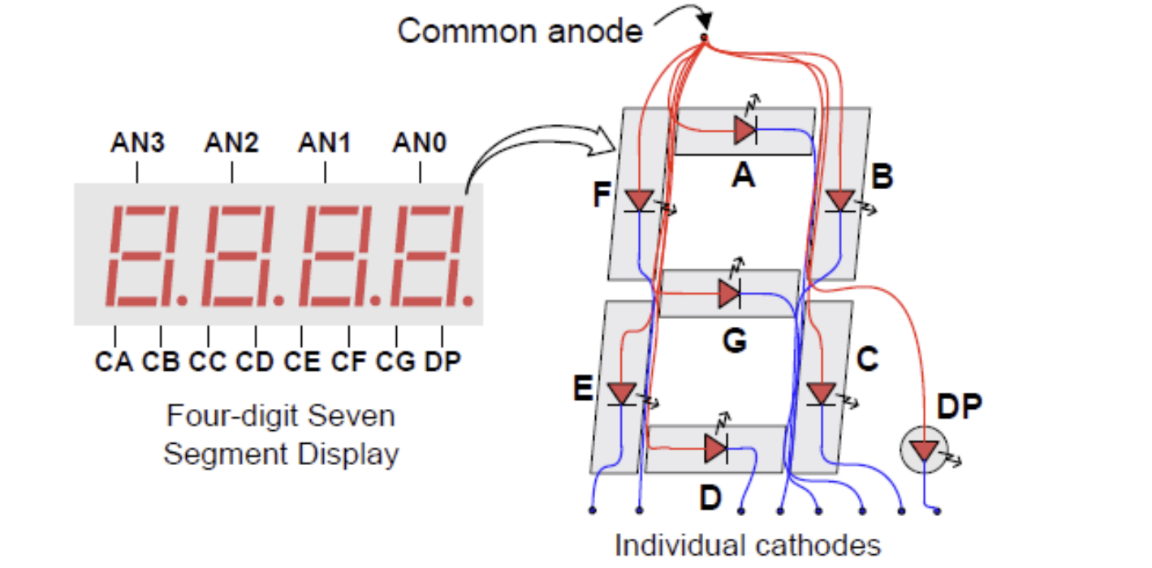
* Numarator pentru etaj curent

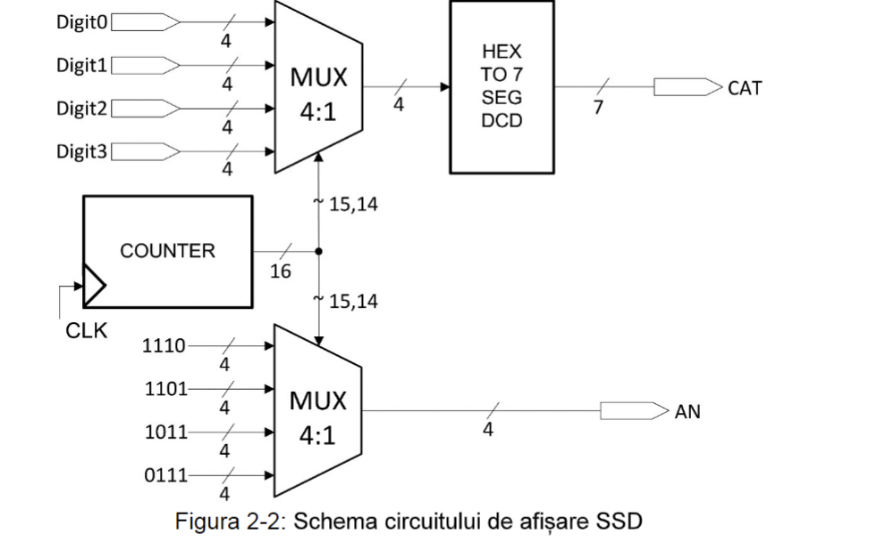
Acest numerator tine evidenta etajului curent la care se afla liftul, sensul de numarare fiind dat de comparatorul descris anterior. Intrarea de enable este data de semnalul **S\_MISCARE** care indica daca liftul se deplaseaza sau nu.

* Controlul afisajului

Aceasta unitate se ocupa de controlarea afisoarelor pe 7 segmente si a ledurilor. Cei 7 catozi ai fiecarui afisor sunt conectati la un anod comun activ pe 0.

(Silviu & Denis)

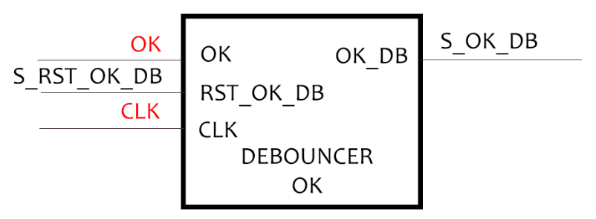


Pentru a reprezenta informatii diferite pe cele 4 afisoare, acestia vor trebui activati pe rand, la o frecventa foarte mare pentru ca utilizatorul sa nu sesizeze schimbarea. Pentru a realiza acest lucru am utilizat urmatoarea schema, folosindu-ne de doua multiplexoare, un numarator si un decodificator pe 7 segmente.

* 1. Unitatea auxiliara

(Silviu)

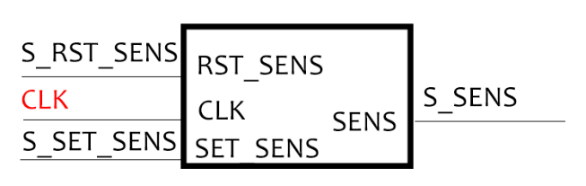
Pentru a implementa algoritmul de cautare a etajului urmator, avem nevoie de unitate auxiliara impartita in mai multe componente

* Debouncer pentru acceptarea cererii

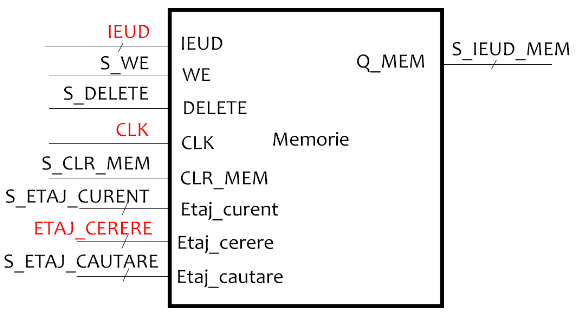
Intrarea de **OK** este data de un buton de pe placuta FPGA, iar pentru a evita fenomenul de bouncing odata cu apasarea acestuia, avem nevoie de folosirea unui debouncer care activeaza iesirea OK\_DB doar daca intrarea **OK** este stabila pentru o anumita durata de timp.

* Bistabil pentru gasirea etajului cautat

Acest bistabil este controlat de Unitatea Centrala si retine daca unitatea auxiliara a gasit un etaj potrivit la care se poate deplasa liftul

* Bistabil pentru sensul de miscare a liftului

Acest bistabil este controlat de Unitatea Centrala si retine sensul in care se vor onora cererile, fie de urcare (1), fie de coborare (0), informatie folosita pentru numaratorul de cautarea al adresei urmatoare.

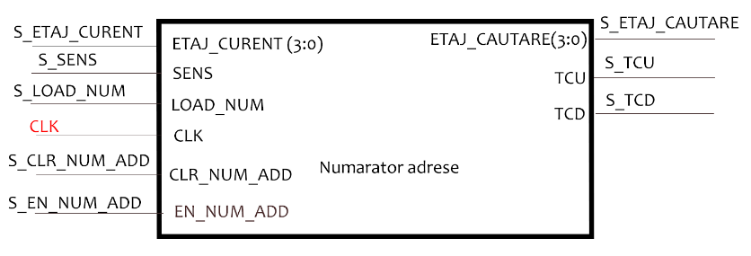
* Unitate de retinere a cererilor

Aceasta unitate are rolul de a inregistra, sterge sau citi dintr-o memorie cererile utilizatorilor. Dimensiunea memoriei este de 12 cuvinte a cate 4 biti, initial toti bitii acesteia fiind initializati cu 0.

In momentul in care se face o cerere, unitatea de control va activa intrarea de Write-Enable(**WE**), iar la adresa **ETAJ\_CERERE**, ce reprezinta codificarea in binar a unuia din cele 12 etaje, se va inregistra valoarea **IEUD**, ce ne ofera informatii despre tipul cererii efectuate (cerere Interioara / Extrioara, de Urcare / Coborare ).

Intrarea de **DELETE** este activata tot de unitatea de control, activandu-se in mometul in care o cererea este indeplinita, valoarea de la adresa **S\_ETAJ\_CURENT** fiind inlocuita cu 0000.

Cand intrarile de Write-Enable si Delete sunt inactive, Unitatea de retinere a cererilor va citi informatia stocata la adresa **S\_ETAJ\_CAUTARE**.

* Numarator pentru adrese

Acesta unitate reprezinta un numarator reversibil pe 4 biti cu rolul de a transmite catre Unitatea de retinere a cererilor urmatoarea adresa la care se va face citirea din memorie.

In momentul in care cererea cautat este valida, numaratorul va face LOAD cu o anumita valoare determinata de sensul de numarare si de cum se pozitioneaza etajul curent la care se afla litful fata de etajul gasit:

* Daca sensul de numarare este **crescator** (urcare):
  + Daca **ETAJ\_CURENT** > **ETAJ\_GASIT**
    - ETAJ\_CAUTARE <= “0000”

(astfel incat intervalul de numarare sa fie 0 - ETAJ\_GASIT)

* + Daca **ETAJ\_CURENT** < **ETAJ\_GASIT**
    - ETAJ\_CAUTARE <= ETAJ\_CURENT

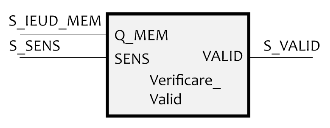
(astfel incat intervalul de numarare sa fie ETAJ\_CURENT – ETAJ\_GASIT )

* Daca sensul de numarare este **descrescator** (coborare):
  + Daca **ETAJ\_CURENT** > **ETAJ\_GASIT**
    - ETAJ\_CAUTARE <= ETAJ\_CURENT

(astfel incat intervalul de numarare sa fie ETAJ\_CURENT - ETAJ\_GASIT)

* + Daca **ETAJ\_CURENT** < **ETAJ\_GASIT**
    - ETAJ\_CAUTARE <= “1100”

(astfel incat intervalul de numarare sa fie 12 – ETAJ\_GASIT )

* Componenta pentru verificarea etajului

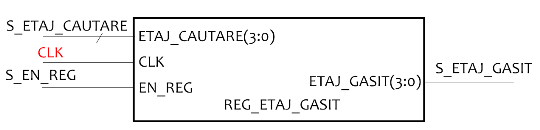
Aceasta componenta va avea iesirea VALID activata daca cererea citita din memorie este valida, adica respecta unul dintre cele 3 cazuri:

* Daca este o cerere interna ( S\_IEUD\_MEM[3] = ‘1’ )
* Daca sensul de miscare este de urcare si cererea este externa de urcare

( S\_IEUD\_MEM[2] = ‘1’ && S\_IEUD\_MEM[1] = ‘1’)

* Daca sensul de miscare este de coborare si cererea este de coborare

(S\_IEUD\_MEM[2] = ‘1’ && S\_IEUD\_MEM[1] = ‘1’)

* Registru pentru etajul gasit

In momentul in care se gaseste o cerere valida, etajul cererii va fi stocat in acest registru, pentru a putea fi folosit de comparatorul pentru determinarea directiei liftului.

1. Proiectare, implementare si justificarea solutiei alese

Proiectarea acestui automat s-a realizat pe baza ideei de a imparti functionarea sa pe doua stari:

* O stare de **miscare**, in care un numarator contorizeaza deplasarea liftului pana la etajul cerut.
* O stare de **repaus**, in care un numarator determina cat va sta liftul la etajul curent.

Unitatea de control are rolul de a determina prezenta sau absenta conditiilor necesare de a trece dintr-o stare in alta, si deasemenea de a sincroniza componentele din unitatea auxiliara.

Unitatea auxiliara este responsabila de determinarea etajului urmator la care se va deplasa liftul, avand la baza urmatoarele idei:

* Cand sensul de mers al liftului este in **sus**, cererile considerate vor fi cele interne sau externe de urcare, prioritara fiind cea mai de jos.
* Cand sensul de mers al liftului este in **jos**, cererile considerate vor fi cele interne sau externe de coborare, prioritara fiind cea mai de sus
* Daca nu se gaseste o cerere pe sensul curent de mers, algoritmul va verifica sensul **opus**.
* Algoritmul ruleaza in continuu astfel ca cel mai optim etaj la care se poate deplasa liftul este **actualizat** in permanenta

(Silviu & Denis)

1. Instructiuni de utilizare

O imagine care conține text, electronice, circuit

Descriere generată automat

Primul pas pentru utilizarea dispozitivului este activarea switch-ului **RESET**, operatie ce reinitializeaza memoria in care sunt stocate cererile si duce liftul la parter cu usile deschise.

Switch-ul **VITEZA** este folosit pentru selectarea vitezei liftului: 0 pentru 1 sec/etaj, 1 pentru 3 sec/etaj

Switch-urile **SZ\_GREUTATE**, **SZ\_USA** vor activa si dezactiva sensorul de greutate respectiv senzorul pentru usa cand liftul se afla in repaus.

Pentru transmiterea unei cereri prima data se va codifica binar numarul etajului folosindu-se cele 4 switch-uri **ETAJ\_CERERE**[3-0], iar apoi se va selecta natura cererii folosind switch-urile **INTERIOR**, **EXTERIOR**, **UP**, **DOWN**.

Pentru finalizarea cererii se va apasa butonul de **OK**.

1. Posibilitati de dezvoltare

Deoarece dorim ca acest proiect sa ia in considerare cat mai circumstante din aplicarea sa practica, ne-am gandit la implementarea pe viitor a urmatoarelor idei:

* Deoarece fizic usa nu isi poate schimba starea imediat, este importanta includerea unei intarzieri ce simuleaza durata pana la deschiderea sau inchiderea sa completa
* Introducerea unui buton pentru anularea unei cereri, care sa poate fi apasat la scurt timp dupa trimiterea comenzii, in cazul in care liftul nu a inceput inca deplasarea catre respectivul etaj.
* Afisarea etajului catre care urmeaza sa se deplaseze liftul.

1. Bibliografie

Pentru realizarea acestui proiect, ne-am folosit de informatiile primite din partea profesorului de laborator, dar si de anumite surse externe:

<https://www.fpga4student.com/2017/09/seven-segment-led-display-controller-basys3-fpga.html>

<https://reference.digilentinc.com/programmable-logic/basys-3/reference-manual>

<https://www.xilinx.com/support.html#documentation>