

Unitate de control pentru un lift al unui hotel cu P+12 etaje

Indrumator laborator:

Miclea Vlad

Realizatori:

Bogdan Stefan

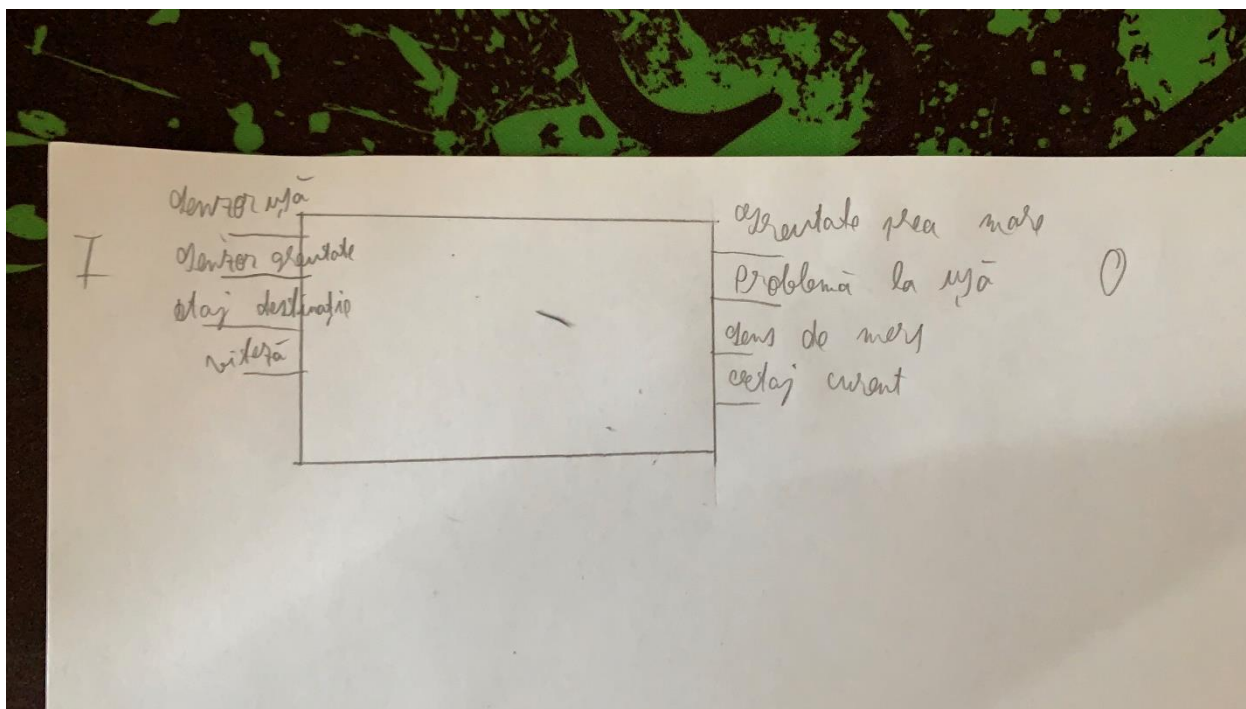
Fleser Mihai

I) Intorudcere proiect

Proiectul nostru a fost sa proiectam un automat care comanda liftul unui hotel cu P+12 etaje. Liftul raspunde la solicitari atat din afara, cat si dintruntru. Acesta onoreaza cererile in ordinea etajelor, indiferent de provenienta. Liftul este echipat cu senzori care detecteaza daca greutatea maxima admisa este depasita si daca usile nu sunt inchise. Usile nu se pot inchide daca un om este detectat in usa. Initial, se presupune ca liftul se afla la parter, iar usile sunt deschise. Viteza liftului este selectabila, iar timpul de deschidere al usilor este programabil

II) Descriere proiect

Vom incepe prin prezentarea proiectului sub forma de schema bloc



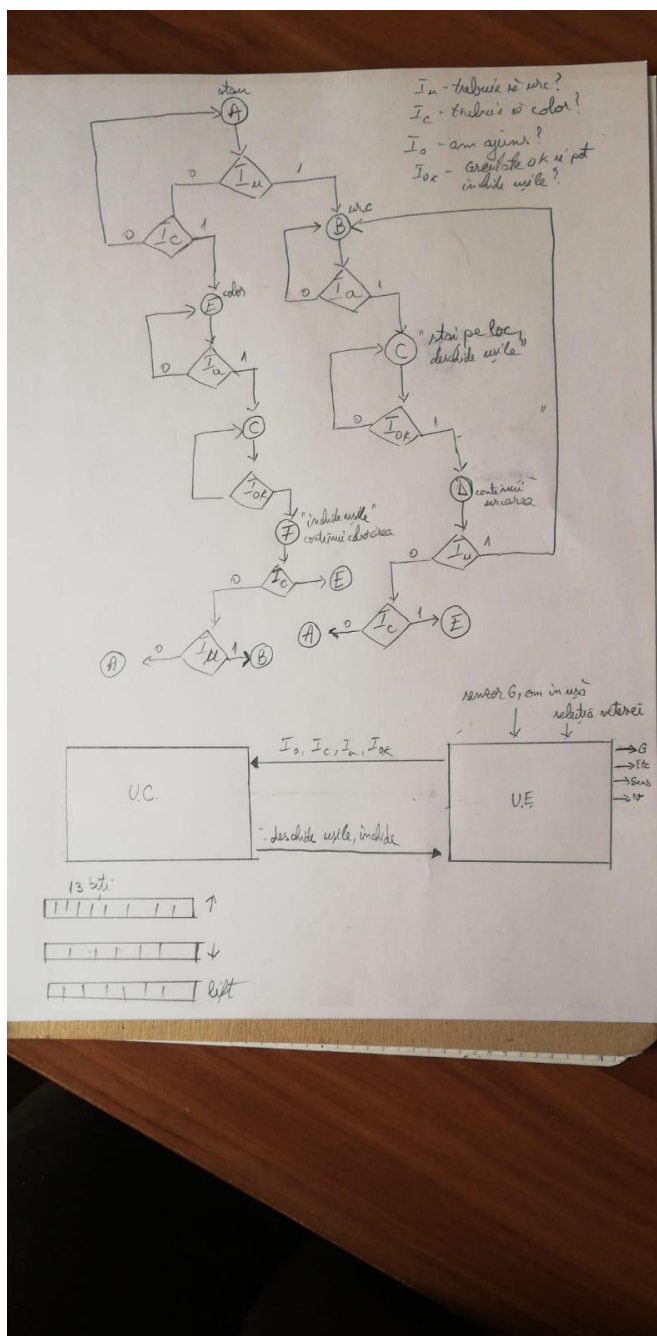
Liftul are 4 intrari:

- 1) Senzor usa, intrare de la senzorul usii liftului, ne spune daca usile sunt deschise
- 2) Senzor greutate, intrare de la senzorul de greutate al liftului, care ne informeaza daca a fost depasita greutatea maxima admisa a acestuia
- 3) Etaj destinatie, intrare primita de la utilizatorul liftului, fie el in interiorul sau exteriorul liftului, ne spune unde vrea acesta sa se deplaseze
- 4) Viteza, intrare primita de la utilizatorul liftului, ne da viteza cu care se va deplasa liftul, adica daca va lua 1 secunda sau 3 secunde sa treaca de la un etaj la altul

Cele 4 iesiri ale liftului reprezinta urmatoarele lucruri:

- 1) Greutate prea mare, ne atentioneaza ca a fost depasita greutatea maxima admisa
- 2) Probleme la usa, ne atentioneaza ca fie este un om in usa, fie usile sunt deschise
- 3) Sens de mers, ne spune daca liftul urca, coboara sau stationeaza la un moment dat
- 4) Etaj curent, atentioneaza utilizatorul la ce etaj se afla la un anumit moment

Organigrama:



Organigrama de mai sus prezinta modul de functionare al automatului nostru. Starile prin care acesta trece sunt urmatoarele:

A – stare de standby, automatul se afla in aceasta stare pana primeste o comanda dinauntru sau dinafara

B- „urc”, automatul se afla in aceasta stare in timp ce urca

C- „stai pe loc, deschide usile”, automatul se afla in aceasta stare cand ajunge la un anumit etaj

D- „continua urcarea”, automatul se afla in aceasta stare cand continua „bucula de urcare”, ca sa nu isi schimbe brusc sensul de mers

E- „cobor”, automatul se afla in aceasta stare in timp ce coboara

F- „continua coborarea”, analog starii D cand automatul se afla in stare de coborare

Iar semnalele care comuta automatul intre starile sale sunt umatoarele:

Iu – „trebuie sa urc”, verificam daca avem o cerere de la un etaj de deasupra noastra

Ic – „trebuie sa cobor”, verificam daca avem o cerere de la un etaj de sub noi

Ia – „am ajuns?”, verificam daca am ajuns la etajul de la care am primit cererea

Iok – „ok?”, verificam daca usile sunt inchise si greutatea liftului e in parametri normali

Implementarea propriu-zisa:

Etajele destinatie sunt introduse de catre utilizatori, iar pozitiile efective ale etajelor sunt memorate in 3 memorii RAM separate. Datele din aceste memorii sunt gestionate de catre 2 componente, floor handler si detectorul de miscare. Acestea au rolul de a calcula etajul la care se afla liftul la un moment dat, respectiv de a lua datele propriu-zise din memorie si de a le manipula. In rest, ne-am folosit de diverse countere pentru a face componentele sa mearga, de o componenta care detecteaza potentiale erori de la senzori si opreste liftul in caz ca le detecteaza, si o componenta care se ocupa numai de deschiderea si inchiderea usilor.

Componentele:

1) Memoria RAM 4X13

Memoriile RAM sunt conectate direct la butoanele din interiorul liftului, respectiv de la etajele hotelului. RAM-ul are ca si semnale de input un write enable, pentru a permite scrierea, un clock pentru functionare sincrona, semnal de reset pentru stergerea continutului si un semnal, „toBeWritten”, care reprezinta inputul primit de la utilizator. Semnalul „toBeWritten” e folosit pentru scrierea continutului RAM-ului

2) Detectorul de miscare

Detectorul de miscare primeste ca si input etajul curent la care se afla liftul, impreuna cu un clock. Acesta se foloseste de aceste semnale, in combinatie cu continuturile RAM-urilor pentru a trimite unul din trei semnale (Iu, Ic, Ia) catre unitatea de control, semnale care determina schimbarea starilor automatului

3) Detectorul de erori

Detectorul de erori primeste ca si input semnale de la senzorii liftului, in combinatie cu clock-ul. Daca detecteaza vreo problema venind de la vreunul din acestia opreste functionarea liftului, prin intermediul semnalelor „PR_USA”, respectiv „PR_GR”. Daca nu exista probleme, semnalul „OK” este activat, care permite functionarea normala a liftului.

4) Floor Handler

Componenta floor handler primeste ca si input viteza liftului, un clock si statutul curent al liftului (urcare/coborare). Acesta se foloseste de 2 countere interne pentru a da etajul curent al liftului, print intermediul semnalului „etajCurent”

5) Usi

Componenta usi primeste 5 inputuri, enable, care permite comutarea intre starile componentei, reset, care blocheaza componenta, clock, SenzOm, care ne spune daca este un om in usa si timpDeschidere, un input programabil care ii spune liftului cat ar trebui sa tina usile deschise cand ajunge la un etaj. Componenta, cu ajutorul a 2 countere trece usile prin 5 stari distincte:

- 1- Idle, cand liftul se afla in repaus
- 2- deschideUsi, cand usile ar trebui sa se deschida, cand liftul ajunge la un anumit etaj
- 3- staiDeschis, perioada cand usile raman deschise, determinata de inputul programabil
- 4- Verifica, o perioada scurta, cand liftul se asigura ca nu e un om in usa
- 5- inchideUsi, perioada de inchidere a usii, cand automatul verifica continuu daca e cineva in usa

Ca si output aceasta componenta ne spune daca usile liftului sunt inchise sau nu, lucru esential pentru alte componente ale automatului

6) States

Componenta principala a automatului, se ocupa de tranzitia dintre starile principale. Acesta primeste semnale ca liftul urca, coboara sau a ajuns la un anumit etaj, iar cu acestea anunta celalalte elemente daca liftul se afla in coborare sau urcare si daca usile pot fi deschise sau inchise.

III) Codul pentru componente si simularile

```
*****  
*****  
*****
```

IV) Concluzii

Concluzionand, consider ca acest proiect m-a ajutat atat pe mine cat si pe colegul meu sa ne dezvoltam cunostiintele in legatura cu limbajul VHDL. Desi am intampinat dificultati, de exemplu in cadrul componentei detectorMiscare la capitolul semnale, am reusit sa trecem peste ele.

Rezolvarea problemei am facut-o treptat, cand am intampinat o problema cu unul din aspectele liftului am implementat componente care sa ne ajute cu depasirea acelor obstacole.