

**FACULTATEA: Automatică și Calculatoare  
SPECIALIZAREA: Calculatoare și Tehnologia Informației  
DISCIPLINA: Proiectarea sistemelor numerice  
PROIECT: AUTOMAT BILETE DE TREN**

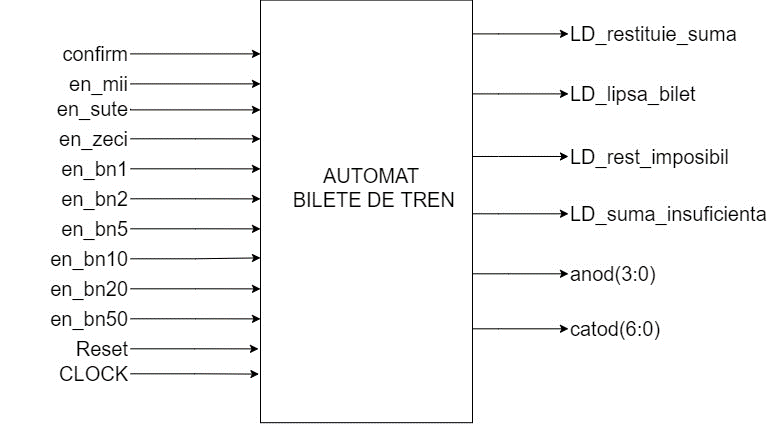
**Îndrumător laborator Student dr. ing. Diana Irena Pop Ioana Vijoli**

**Cuprins**

1. Specificația proiectului .............................................. 3
2. Proiectare .................................................................. 3
   1. Schema bloc........................................................ 3
      1. Schema........................................................ 3
      2. Lista detaliată cu intrări şi ieşiri ................. 4
   2. Descompunerea în UC şi UE ............................. 5
      1. Lista completă a resurselor.......................... 6
   3. Descrierea detaliată a resurselor.......................... 6
   4. Organigrama UC................................................ 16
3. Justificarea soluţiei alese ........................................... 17
4. Manual de utilizare şi întreţinere ............................... 17
5. Posibilităţi de dezvoltare ulterioară ........................... 17
6. Bibliografie ................................................................ 18
7. **Specificația proiectului**

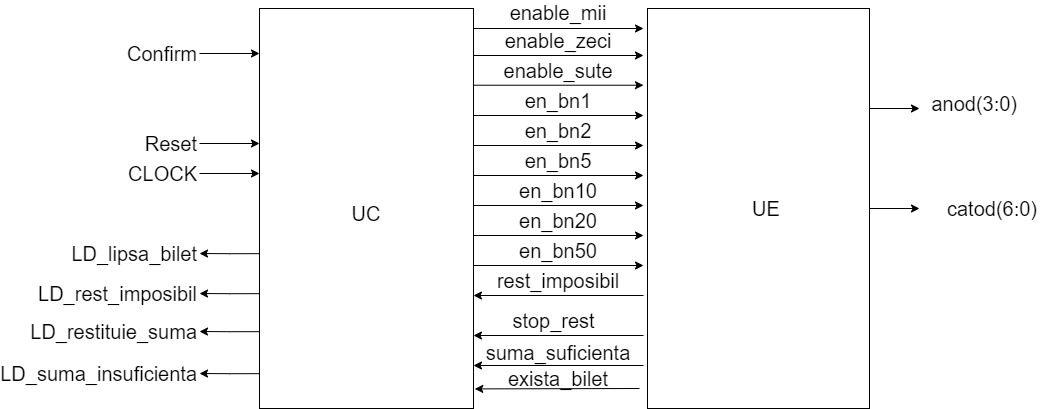
**Cerința:** Să se proiecteze un automat pentru cumpărarea biletelor de tren. Cumpărătorul introduce distanța până la destinație (în zeci de km). Costul biletului și sumele introduse sunt afișate pe afișoare 7 segmente. Moneda utilizată este EURO. Prețul maxim pentru un bilet este de 100 Euro. Automatul primește suma necesară în hârtii sau monede și eliberează biletele și, eventual, restul. El dispune de o casă de bani care se încarcă la începutul funcționării cu un număr de hârtii și monede (toate posibilitățile între 1 euro și 50 de euro). Lipsa de bilete, introducerea unei sume mai mici decât costul biletului sau imposibilitatea restituirii restului se semnalizează luminos. Se poate renunța în orice moment la operație, cu restituirea sumei introduse, dacă este cazul.

1. **Proiectare**
   1. Schema bloc

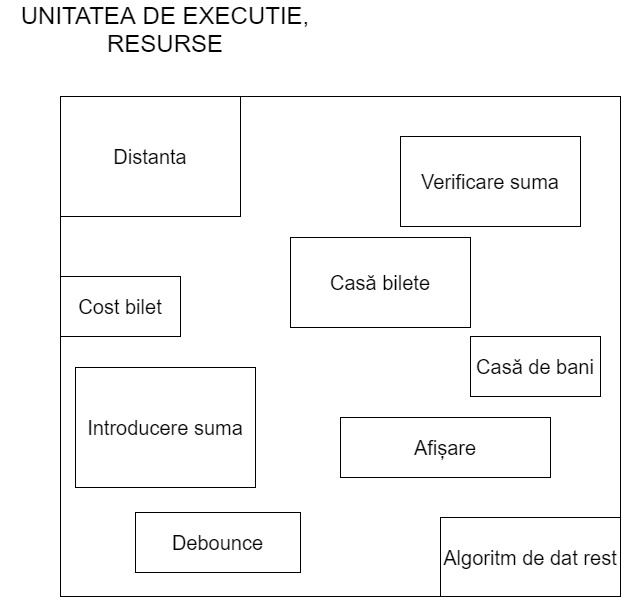
2.1.1. Schema

2.1.2. Lista detaliată cu intrări şi ieşiri

* confirm = intrare, pe 1 bit, prin intermediul acesteia, utilizatorul confirmă că doreşte să treacă la pasul următor
* en\_mii = 1 bit, cu ajutorul ei, utilizatorul va introduce cifra miilor din distanţa pe care vrea să o parcurgă
* en\_sute = 1 bit, cu ajutorul ei, utilizatorul va introduce cifra sutelor din distanţa pe care vrea să o parcurgă
* en\_zeci = 1 bit, cu ajutorul ei, utilizatorul va introduce cifra zecilor din distanţa pe care vrea să o parcurgă
* en\_bn1 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 1 euro în automat
* en\_bn2 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 2 euro în automat
* en\_bn5 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 5 euro în automat
* en\_bn10 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 10 euro în automat
* en\_bn20 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 20 euro în automat
* en\_bn50 = 1 bit, cand această intrare este activate, se semnalează că utilizatorul introduce o bancnotă de 50 euro în automat
* reset = 1 bit, va readuce automatul în starea iniţială, însemnând totodată şi renunţarea la operaţie
* CLOCK = 1 bit, semnalul de tact al plăcii FPGA
* LD\_restituie\_suma = 1 bit, aprinderea acestui led înseamnă că utilizatorul primeşte înapoi suma de bani introdusă
* LD\_lipsă\_bilet = 1 bit, aprinderea acestui led semnalează că automatul nu poate să dea biletul cerut
* LD\_rest\_imposibil = 1 bit, aprinderea acestui led semnalează faptul că în automat nu există suficiente bancnote ca să se poată elibera restul, iar suma introdusă va fi restituită
* LD\_suma\_insuficienta = 1 bit, acest led va sta aprins până în momentul în care utilizatorul a introdus destule bancnote pentru a plăti biletul
* anod = 4 biţi, folosit pentru a afişa datele pe display
* catod = 7 biţi, utilizat în afişare

2.2. Descompunerea în UC şi UE

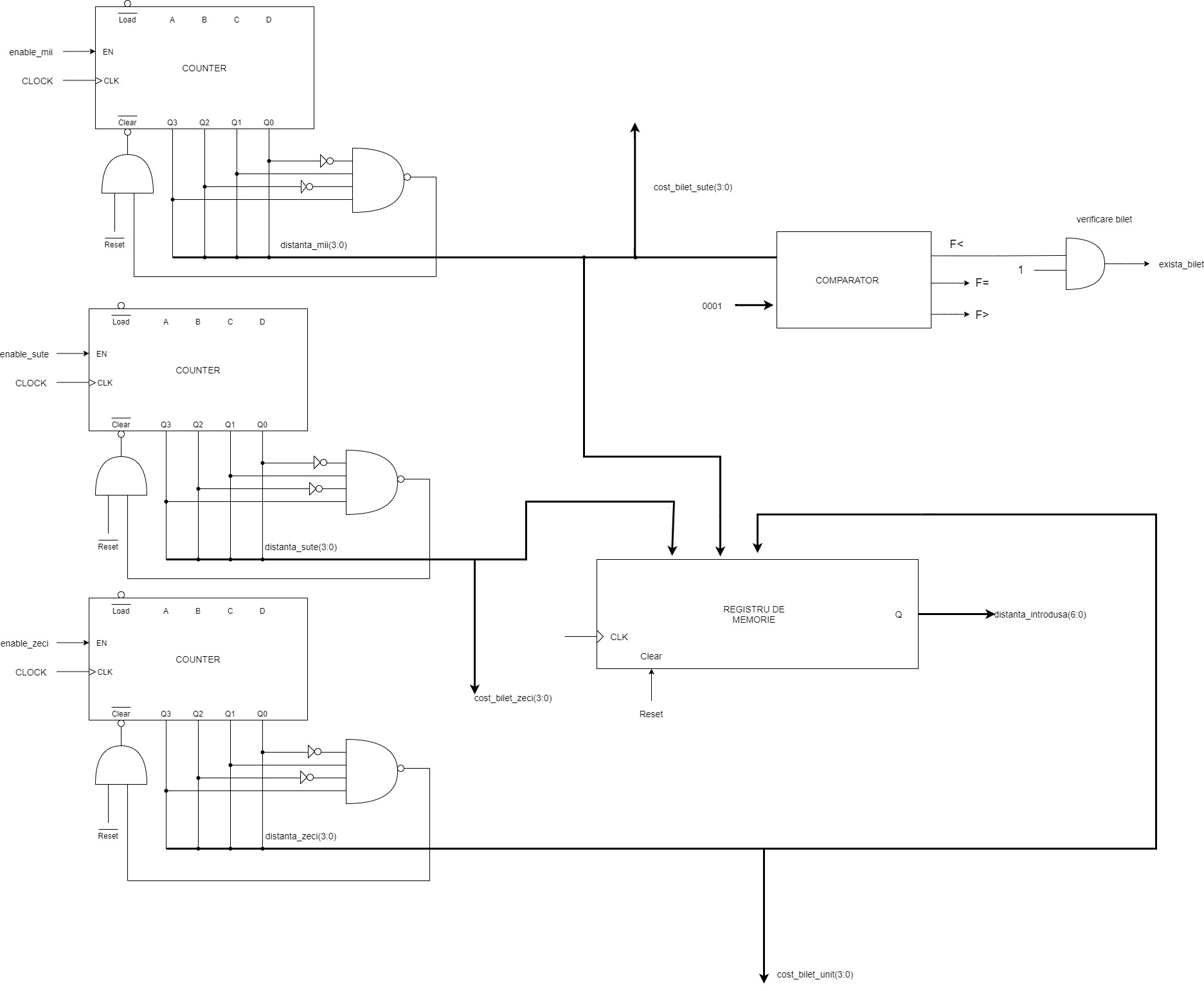
2.2.1. Lista completă a resurselor



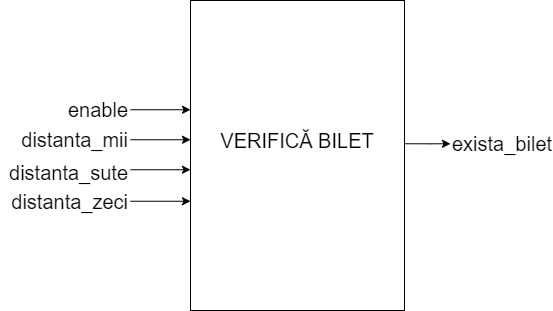
* 1. Descrierea detaliată a resurselor
* DISTANŢA



Distanţa este o resursă prin intermediul căreia se stabileşte distanta introdusă de utilizator până la destinaţie. Astfel, distanta\_introdusa este o variabilă pe 7 biţi, pe când distanta\_mii, distanţa\_sute şi distanta\_zeci sunt variabile pe 4 biţi. Vom folosi un numărător zecimal, cu ajutorul căruia vom stabili distanta\_mii, distanţa\_sute şi distanta\_zeci. Numărătorul va începe numărarea când enable\_mii, enable\_sute, sau enable\_zeci este 1 şi are şi un semnal de reset. Distanţa introdusă se pastreaza intr-un registru de memorie.



* VERIFICĂ BILET



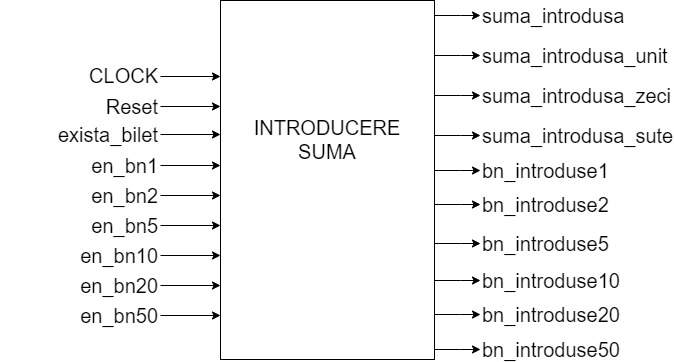
Componeta de verificare bilet primeşte pe intrare distanta\_introdusa şi verifică dacă există sau nu bilet până la destinaţie. În caz afirmativ exista\_bilet va fi 1, iar altfel variabila va avea valoarea 0. Biletul există şi se poate elibera, în cazul în care distanţa introdusă este mai mică decât 1000 de km. Altfel, automatul nu va putea elibera bilet şi va semnala acest lucru printr-un led.

* COST BILET



Pentru costul biletului, am stabilit regula ca preţul unui bilet să fie egal cu numărul format din cifra miilor şi cifra sutelor din distanta\_introdusa. Astfel, cost\_bilet va fi o variabilă pe 7 biţi, pe cand cost\_bilet\_unit, cost\_bilet\_zeci şi cost\_bilet sute vor fi variabile pe 4 biţi. Enable este folosit pentru a stabili momentul în care se porenşte execuţia acestei componente, el provenind de la UC.

* INTRODUCERE SUMA



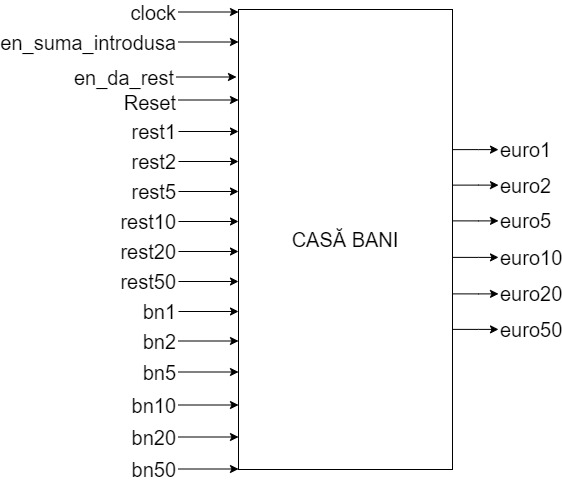
Vom folosi un numărător cu semnal de Enable, pentru a calcula câte bacnote din fiecare tip sunt (en\_bn1, en\_bn2, en\_bn3,.... vor activa sau dezactiva numărătorul). Apoi, cu ajutorul unui sumator se va calcula suma\_introdusa, care este o variabilă pe 7 biţi. De asemenea vom reţine fiecare cifră din numărul ce reprezintă suma, pentru a putea să o afişăm ulterior. Output-urile bn\_introduse1, bn\_introduse2, bn\_introduse5, bn\_introduse10, bn\_introduse20, bn\_introduse50, după cum le spune şi numele, reţin numărul de bancnote de 1 euro, 2 euro, 5 euro, 10 euro, 20 euro, respectiv 50 de euro, introduse de utilizator. Ele vor fi transmise ulterior componentei casa bani, pentru a fi adaugate la numărul de bancnote existent deja în automat.

* VERIFICARE SUMA



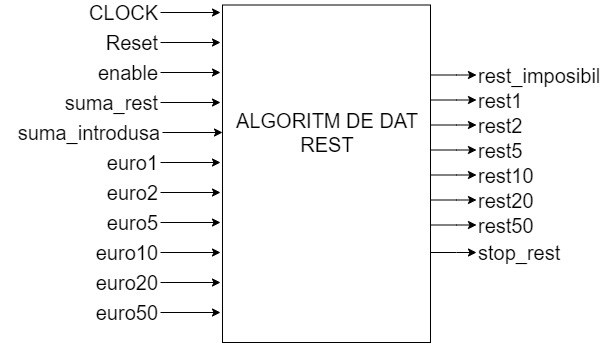
Pentru verificarea sumei se va folosi un comparator, care va stabili dacă suma\_introdusă este mai mică, egală sau mai mare decât cost\_bilet. În cazul în care suma\_introdusa < cost\_bilet, variabila suma\_suficientă va avea valoarea 0, pentru a semnaliza că nu s-au introdus destui bani. Altfel, suma\_suficientă va fi 1, iar dacă suma\_introdusa > cost\_bilet, atunci enable\_rest va fi 1 şi se va calcula suma\_rest, cu ajutorul unui scăzător (suma\_rest = suma\_introdusa – cost\_bilet).

* CASĂ DE BANI

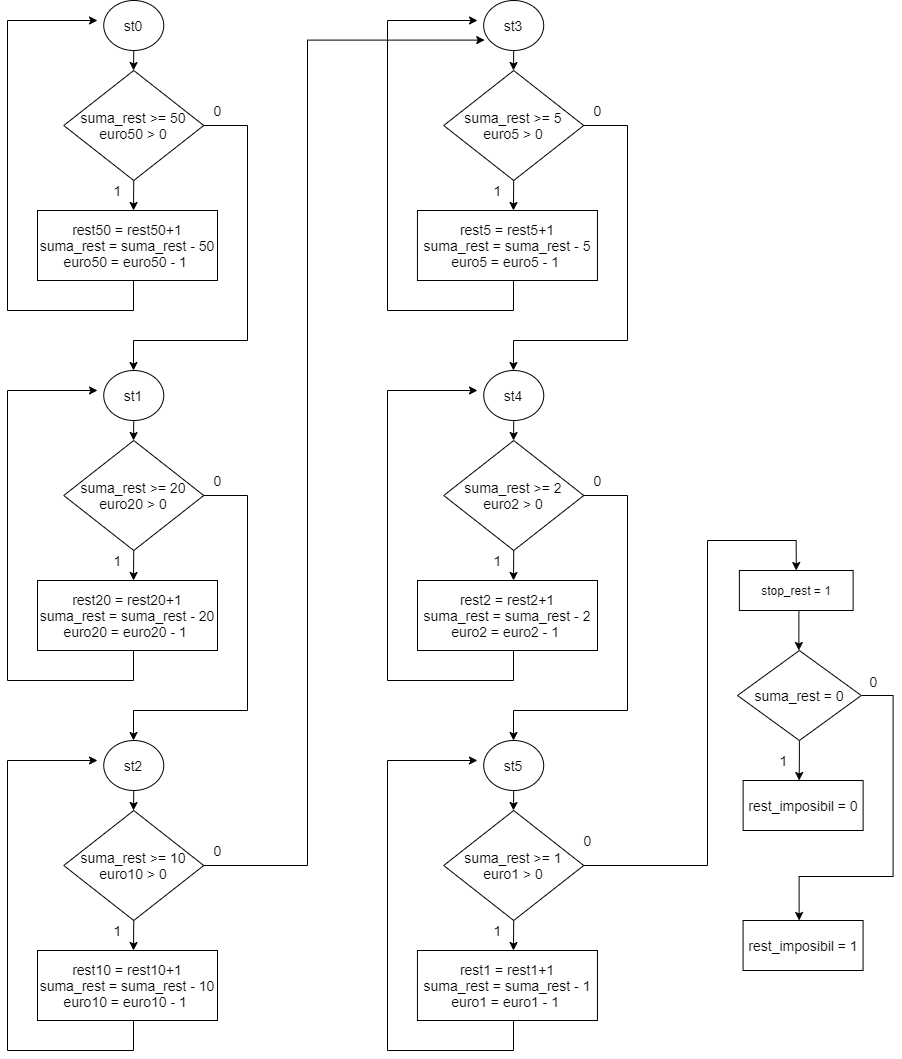


Casa de bani stabileşte câte bacnote de fiecare tip există în automat. Astfel, euro1, euro2, euro5, euro10, euro20 şi euro50 sunt variabile pe 7 biţi, reprezentând numărul de bancnote de 1 euro, 2 euro,... restpectiv 50 euro existente după ce utilizatorul a introdus suma. Cele 2 semnale de enable en\_da\_rest şi en\_suma\_introdusă, pornesc componenta, fie pentru a adăuga bani primiţi de la utilizator, fie pentru a scoate bancnotele necesare eliberării restului.

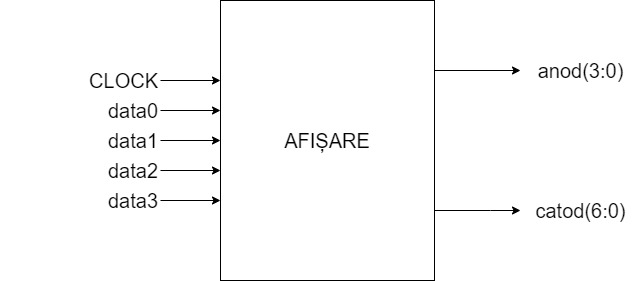
* ALGORITM DE DAT REST

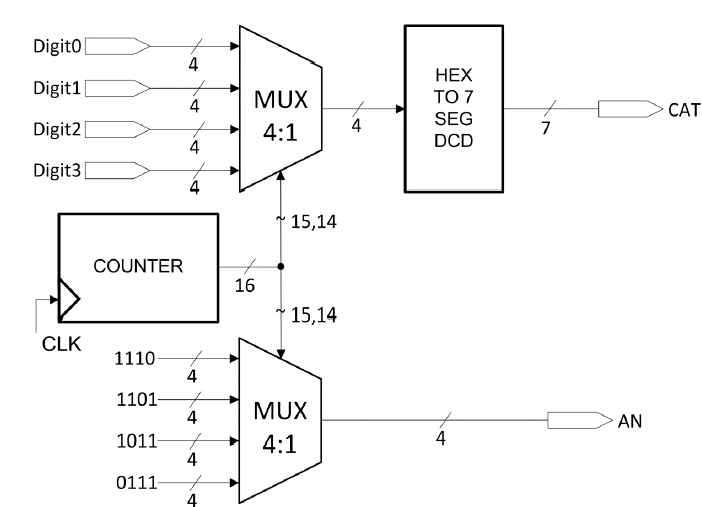


Algoritmul de dat rest primeşte suma introdusă de utilizator, suma de dat rest, dar şi numărul de bancnote din fiecare tip, existente în bancă şi returnează câte bancnote de 1 euro, 2 euro, .. respectiv 50 de euro se vor da înapoi utilizatorului. Semnalul stop\_rest comunică încheierea efectuării algoritmului, pe când rest\_imposibil va aprinde un led, în cazul în care nu putem da rest.

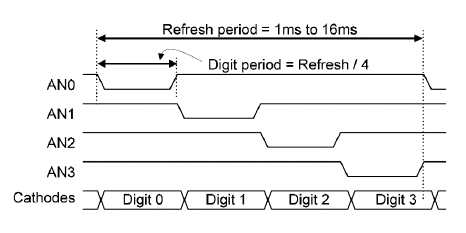


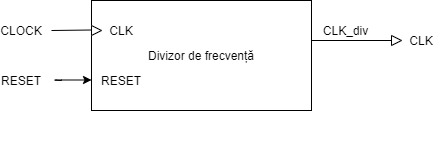
* AFISARE





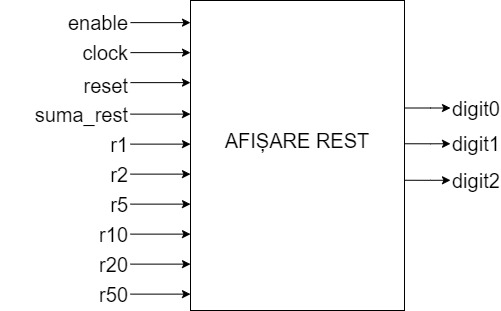
În cadrul componentei de afişare sunt folosite un numărător (pentru divizorul de frecvență) și două multiplexoare, care au ca și selecție ultimii doi biți(15 și 14) a vectorului emis de numărător. De exemplu, dacă se află în cazul 00 atunci primul multiplexor preia valoarea digit și al doilea multiplexor activează primul catod.



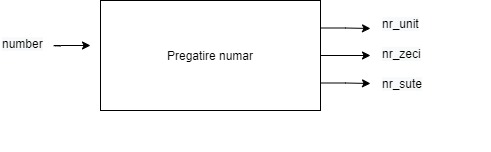


Divizorul de frecvenţă are rolul de a încetini clock-ul de 100 Mhz generat de plăcuța Basys3. Acest lucru este efectuat cu un proces care are în lista de sensibilitate semnalul de CLK, iar cu ajutorul unei variabile de tipul std\_logic\_vector de 16 biți inițializată cu 0, semnalul de clock este încetinit.

* AFIŞARE REST



Această componentă are rolul de a pregăti afişarea restului şi are ca iesire cifra care se afişează în momentul curent. În cadrul acestei resurse se mai afla o componentă, pregătire număr, care transformă un număr pe 7 biţi, într-un număr gata de afişat. De asemenea, am folosit un numărator care să numere timp de o secundă, pentru a putea afişa numere consecutiv pe FPGA.

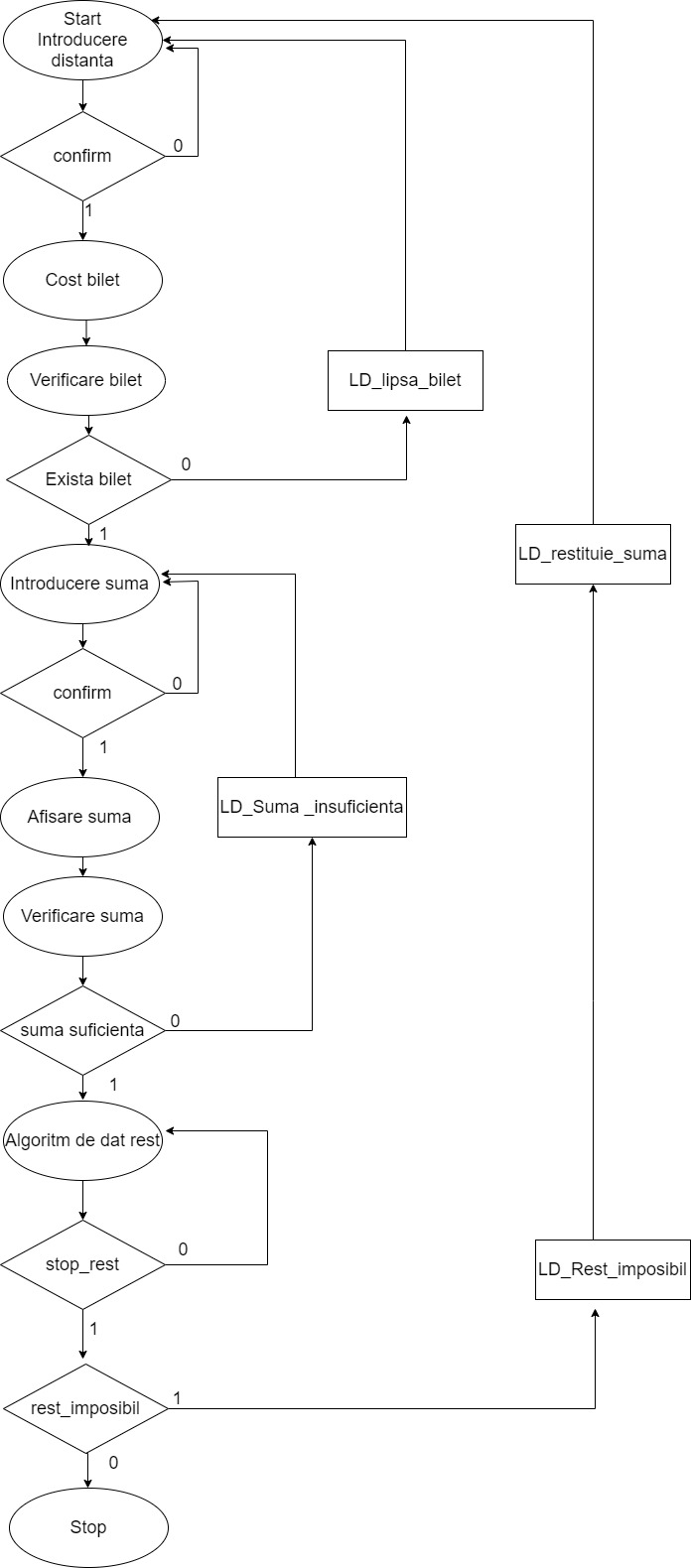


* DEBOUNCE



Această componentă verifică daca switch-ul este activat timp de 50 ns. Este folosită pentru a putea introduce datele pe placa FPGA, cu ajutorul switch-urilor

* 1. Organigrama UC



1. **Justificarea soluţiei alese**

Am ales această soluţie, deoarece este uşor de implemnetat, iar modul de utilizare este destul de intuitiv. Codul din VHDL respecta structura descrisă anterior, care ulterior se poate testa pe placa FPGA. Se folosesc circuite logice combinaţionale şi secvenţiale, cu ajutorul cărora se generează ieşirile corespunzătoare din fiecare componentă. Semnalele de enable care provin din UC, controlează ordinea în care se activează fiecare component. Switch-ul de reset ne permite să renunţăm la operaţie în orice moment.

1. **Manual de utilizare şi întreţinere**

Pentru utilizarea plăcuței FPGA și a automatului de dat bilete de tren este nevoie de programul ISE Design Suite, care poate fi descărcat de pe site-ul Xilinx. După instalarea programului, se începe rularea acestuia.

Se urmăresc următorii pași după ce FPGA-ul a fost progamat.

* Se introduc datele de la butoane (se poate observa cum display-ul se modifică, afişând distanţa introdusă).
* Se apasă butonul de confirmare
* Pe display va apărea costul biletului
* Se apasă din nou butonul de confirm
* Se introduce cu ajutorul switch-urilor suma de bani necesară
* Se confirmă suma
* Pe display va apărea restul

1. **Posibilităţi de dezvoltare ulterioară**

Se poate dezvolta afişarea restului astfel încât să dureze mai mult de o secundă timpul în care numarul este afişat pe placă. De asemenea se pot implementa mai multe tipuri de bilete, cu costuri diferite, iar utilizatorul să poată să aleagă ce tip de bilet doreşte.

1. **Bibliografie**

* Manualul de referinţă al plăcii Basys 3
* <https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/VHDL/docs-pdf/Vivado_Tutorial.pdf>
* Cursul de Proiectare a sistemelor numerice
* https://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf
* https://www.arl.wustl.edu/projects/fpx/class/resources/Libraries%20and%20Packages%20in%20VHDL.htm