UNIVERSITATEA TEHNICA CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA SI CALCULATOARE

AUTOMAT BANCAR

Disciplina: Proiectarea Sistemelor Numerice Nume prenume studenti:

Profesor coordonator: Vlad Cristian Miclea Chetan Anca

Boca Bianca Malvina

Grupa 30216

**1. Cerinta si specificatii**

**Să se proiecteze un automat bancar pentru extrageri de sume în EURO. Se presupune că suma maximă care poate fi extrasă o dată este de maximum 1.000 euro. Iniţial se efectuează identificarea cardului şi se alege operaţia. Vor fi suportate minim 4 carduri/conturi diferite şi se vor implementa minimum 4 operaţii diferite. Automatul dispune de o casă în care iniţial se introduce o anumită sumă (număr de bancnote de diferite valori). În cazul cererii de eliberare de numerar se introduce suma, se verifică existenţa sumei cerute, se vizualizează tipurile de bancnote emise şi se actualizează contul. Apoi se eliberează cardul, suma şi, eventual, chitanţa.**

pin

Automatul bancar va realiza urmatoarele operatiuni:

optiune

* Retragere numerar
* Depunere numerar

suma

* Interogare sold

Automat

Bancar

sold

* Schimbare pin

Pin\_nou

Intrarile bancomatului sunt urmatoarele:

* Pinul: intrare pe 16 biti

chitanta

* Optiunea aleasa de utilizator: intrare pe 2 biti
* Suma (pentru retragere sau depunere): intrare pe 16 biti
* Pin\_nou (pentru optiunea de schimbare pin): intrare pe 16 biti
* Chitanta: intrare pe un bit care semnaleaza daca se doreste eliberarea chitantei sau nu

Iesirile bancomatului sunt urmatoarele:

* Soldul (pentru retragere si depunere): iesire pe 16 biti

Se verifica atat in automat (memoria ce retine numarul de bancnote), cat si in cont(memoria ce retine suma din cont)

0

1

Validare pin

Mesaj eroare

Introducere pin

Alegere optiune

11

10

01

00

Accesarea memoriei cu pinurile cardurilor si schimbare pin

Introducere suma pentru depunere

Introducere suma pentru retragere numerar

0

Verificare existanta suma ceruta

1

Actualizare cont folosind scazator

Mesaj eroare

Accesarea memoriei cu sumele cardurilor si afisare sold

Actualizare cont folosind sumator

1

0

Afisare pin nou

Chitanta

**Componentele folosite vor fi:**

* **3 memorii RAM**
* **Un ALU – sumator/scazator**
* **O componenta care va efectua validarea pinului**

1

1

0

0

Eliberare chitanta

Chitanta

Eliberare chitanta

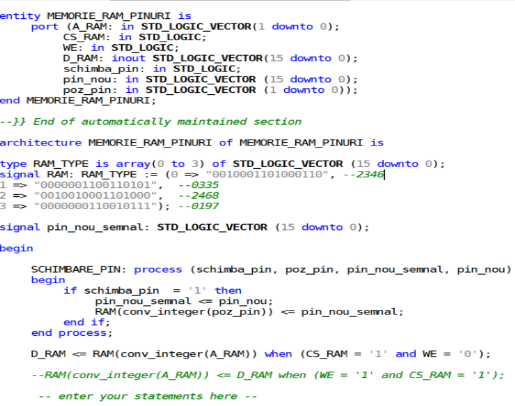
Eliberare chitanta

Chitanta

**2. Componentele utilizate:**

**Vom avea urmatoarele semnale:**

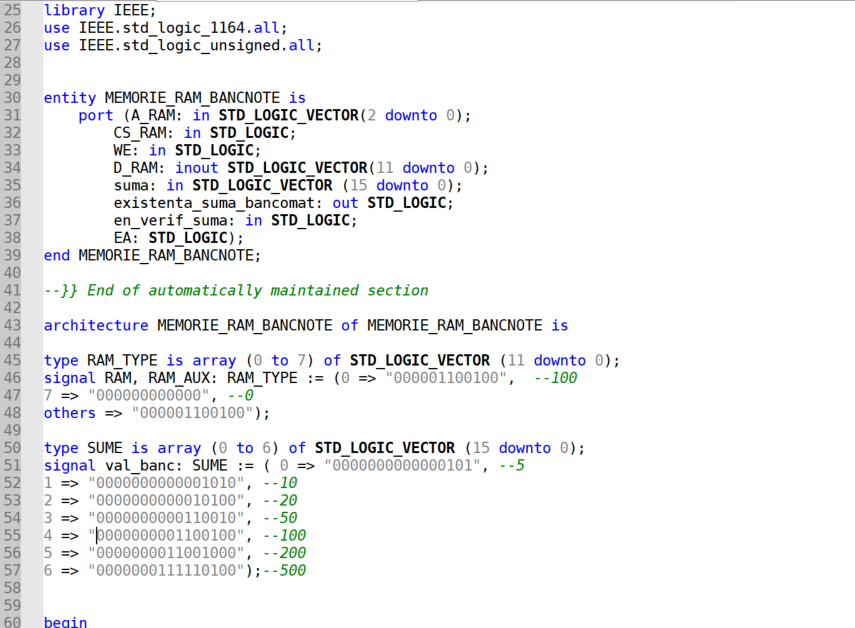
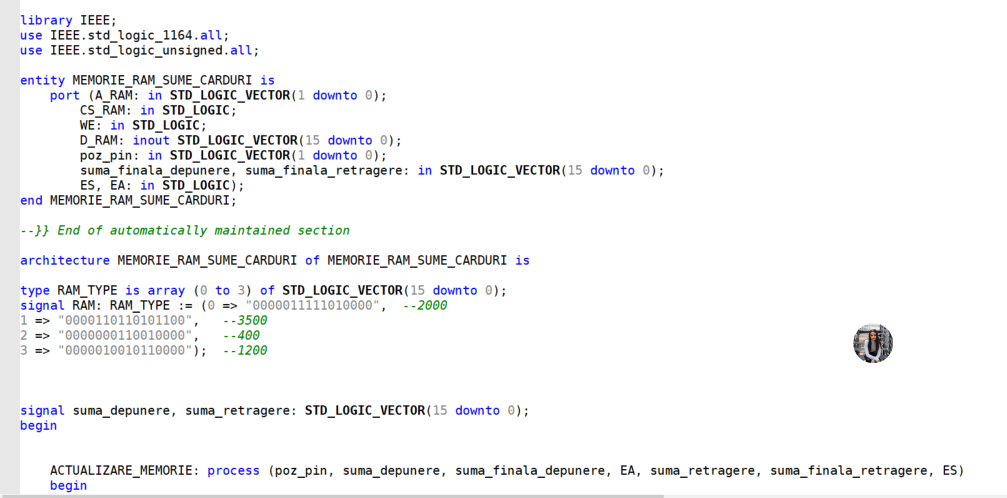
* WE1 – enable ce permite scrierea sau citirea in memoria RAM ce retine numarul de bancnote
* WE2 – enable ce permite scrierea sau citirea in memoria RAM ce retine pinurile cardurilor
* WE3 – enable ce permite scrierea sau citirea in memoria RAM ce retine sumele din conturi
* EA – enable pentru ALU, pentru efectuarea adunarii.
* ES – enable pentru ALU, pentru efectuarea scaderii
* En\_verif\_suma – enable pentru verificarea sumei cerute de retragere atat in memoria cu sumele din conturi, cat si in memoria cu bancnote
* En\_validare\_pin – enable transmis pentru validarea pinului
* Schimba\_pin – semnal transmis pentru schimbarea pinului
* Afisare\_sold – semnal transmis pentru afisarea soldului
* Vom folosi o memorie RAM pentru a retine numarul de bancnote existente in bancomat, astfel memoria va avea 8 locatii si va fi codificata astfel:

La adresa:

* + 000 - nr. bancnote de 5 EURO
  + 001 – nr. bancnote de 10 EURO
  + 010 – nr. bancnote de 20 EURO
  + 011 – nr. bancnote de 50 EURO
  + 100 – nr. bancnote de 100 EURO
  + 101 – nr. bancnote de 200 EURO
  + 110 – nr. bancnote de 500 EURO
  + 111 – 0
* Vom folosi alte doua memorii RAM pentru a retine datele de identificare pentru carduri. Aceaste memorii vor dispune de 4 locatii pentru a putea retine datele a 4 carduri. Prima memorie va retine pinurile cardurilor, iar cea de-a doua sumele existente in conturi.

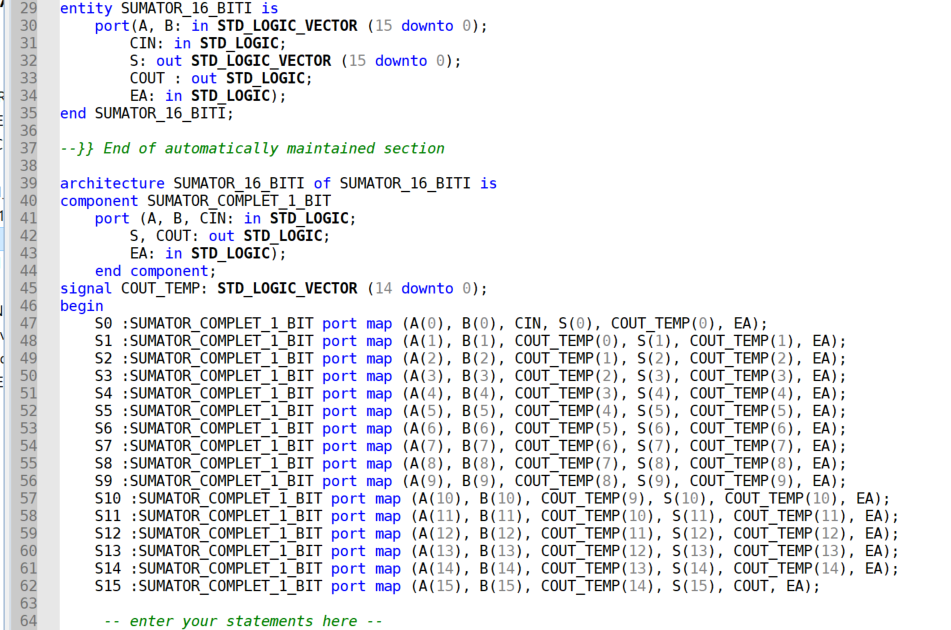
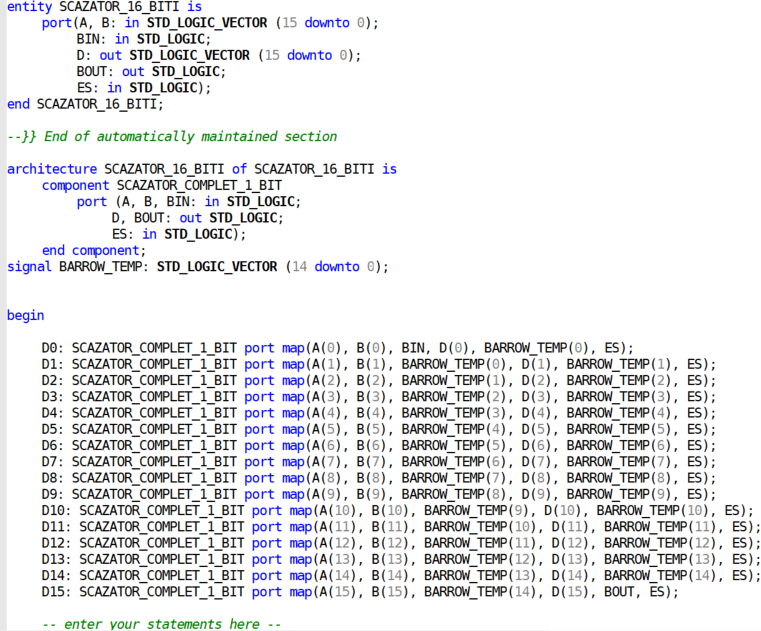
La adresele din memorii se vor putea accesa urmatoarele:

* + 00 – pin/suma card 1
  + 01 – pin/suma card 2
  + 10 – pin/suma card 3
  + 11 – pin/suma card 4

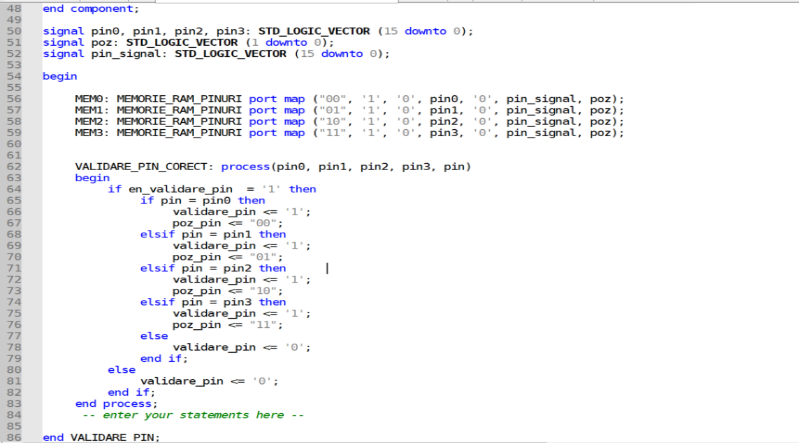


Se va folosi un multiplexor pentru a alege optiunea dorita, codificat astfel:

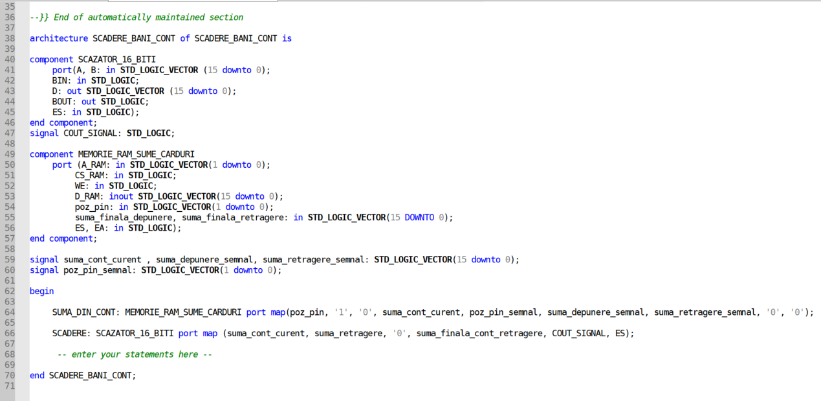
* 00 – retragere numerar
* 01 – depunere numerar
* 10 – interogare sold
* 11 – schimbare pin
* Pentru actualizarea conturilor vom folosi un ALU care va efectua adunarea si scaderea.



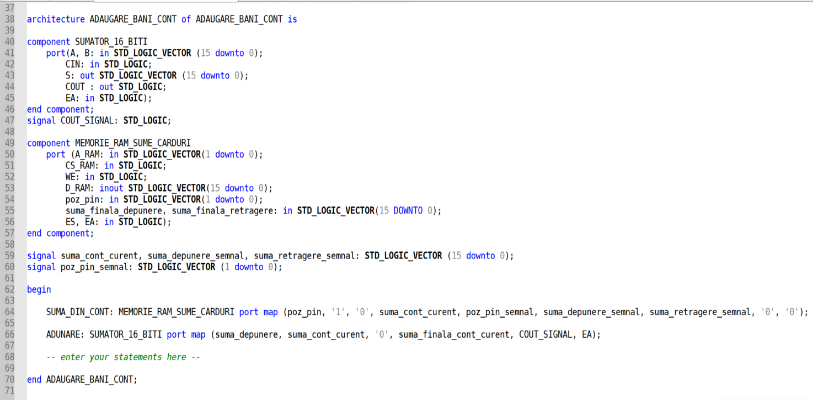
* Componenta de validare pin care realizeaza cautarea in memorie a pinului, iar daca il gaseste retine pozitia acestuia si transmite prin semnalul validare\_pin 0 daca nu a gasit pinul si 1 daca l-a gasit in memorie.

****

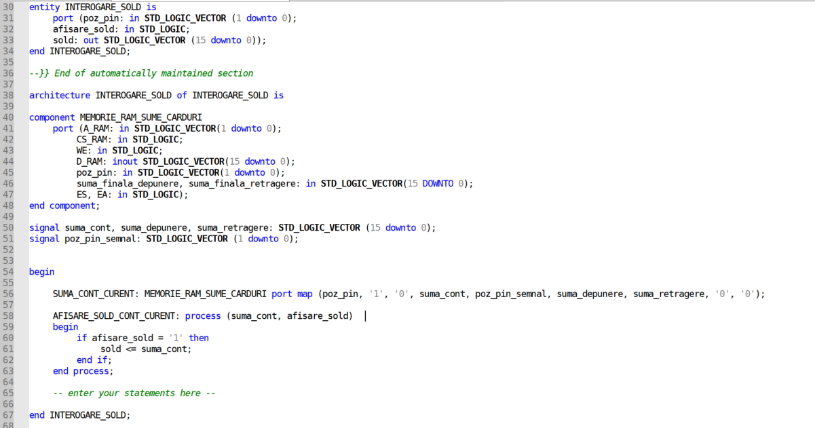
* Componenta de retragere numerar, care primeste suma de retragere, iar cu ajutorul scazatorului se obtine suma de dupa retragere, care se modifica apoi si in memorie.

****

* Componenta de depunere numerar, care primeste suma de depunere, iar cu ajutorul sumatorului se obtine suma de dupa depunere, care se modifica apoi si in memorie.



* Componenta de interogare sold, care extrage soldul de la pozitia retinuta de componenta validare pin si trimite soldul intr-un output, pentru a fi vizibil utilizatorului.



* ****Componenta de verificare a existentei sumei cerute in memoria ce retine sumele de pe carduri, pentru optiunea retragere numerar.

**3. Instructiuni de utilizare:**

* Utilizatorul va introduce mai intai pinul, prin semnalul en\_validare\_pin = ‘1’, acesta va fi validat sau nu in componenta care face verificarea pinului. Doar daca pinul e valid se va trece mai departe.
* Utilizatorul va introduce apoi una dintre cele 4 optiuni posibile.
* Pentru retragere numerar se va introduce suma, se va verifica existenta sumei cerute prin enable-urile la memorii WE1, WE2 si semnalul en\_verif\_suma, atat pe card cat si in automat si se va actualiza contul prin semnalul ES la ALU care va permite efectuarea scaderii. Apoi se va elibera cardul, suma si la cerere chitanta.
* Pentru depunere numerar in cont se va efectua adunarea la suma deja existenta in cont prin semnalul EA la ALU care permite actualizarea contului. Se va elibera cardul si la cerere chitanta.
* Pentru interogare sold se va accesa memoria prin semnalul WE3 si prin semnalul afisare\_sold = ‘1’ se va activa componenta care va trimite soldul intr-un output.
* Pentru schimbare pin se va transmite semnalul de WE2 la memorie si prin semnalul schimba\_pin = ‘1’ se va face modificarea in memorie.
* Dupa efectuarea oricarei operatiuni alese de utilizator se va putea reveni in starea initiala, iar procesul se poate relua pentru alegerea unei noi optiuni.

**4. Justificarea solutiei alese:**

Am ales sa proiectam automatul utilizand diferite componente pe care apoi le-am legat intr-o componenta principala cu ajutorul semnalelor. Am ales sa utilizam organigrama pentru a transmite cat mai usor semnalele la diferite componente si pentru a fi foarte clar cum se trece dintr-o stare in alta. Am optat pentru o descriere atat structurala cat si comportamentala dupa cum s-a putut observa pentru sumator si scazator am ales o descriere structurala, iar pentru componentele care realizeaza efectiv operatiile automatului, cum ar fi validare pin sau schimbare pin am ales o descriere comportamentala. Am incercat sa imbinam aceste doua tipuri de descriere pentru a face cat mai clara solutia aleasa. Componenta principala este realizata doar prin atribuiri de semnale spre diferite componente, pentru a fi cat mai lizibil ce componenta se utilizeaza pentru fiecare optiune. Comportamentul automatului am ales sa il reprezentam printr-o organigrama, deoarece ni s-a parut ca in acest fel se evidentiaza cel mai bine toti pasii pe care automatul bancar ii efectueaza in orice situatie.

**5. Posibilitati de dezvoltare:**

O prima posibilitate de dezvoltare a automatului bancar ar fi extinderea capacitatii memoriei pentru a retine mai mult de 4 carduri si datele despre acestea. O alta imbunatatire ar fi posibilitatea ca la retragere si depunere numerar sa se poata extrage orice suma de bani, nu doar bancnote fixe asa cum realizeaza automatul in acest moment. Transferul bancar intre doua conturi ar putea fi o alta functionalitate pe care automatul ar putea sa o realizeze.