

**UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**  
**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**PROIECTAREA SISTEMELOR NUMERICE -  
PROIECT**  
**”SONNY”, THE ATM**



**COORDONATOR:**

**CIREAP GABRIEL**

**REALIZAT DE:**

**VLAD URSACHE**

**DENISA HERLEA**

**AN I, SEMESTRUL II, GRUPA 30218**

**2021**

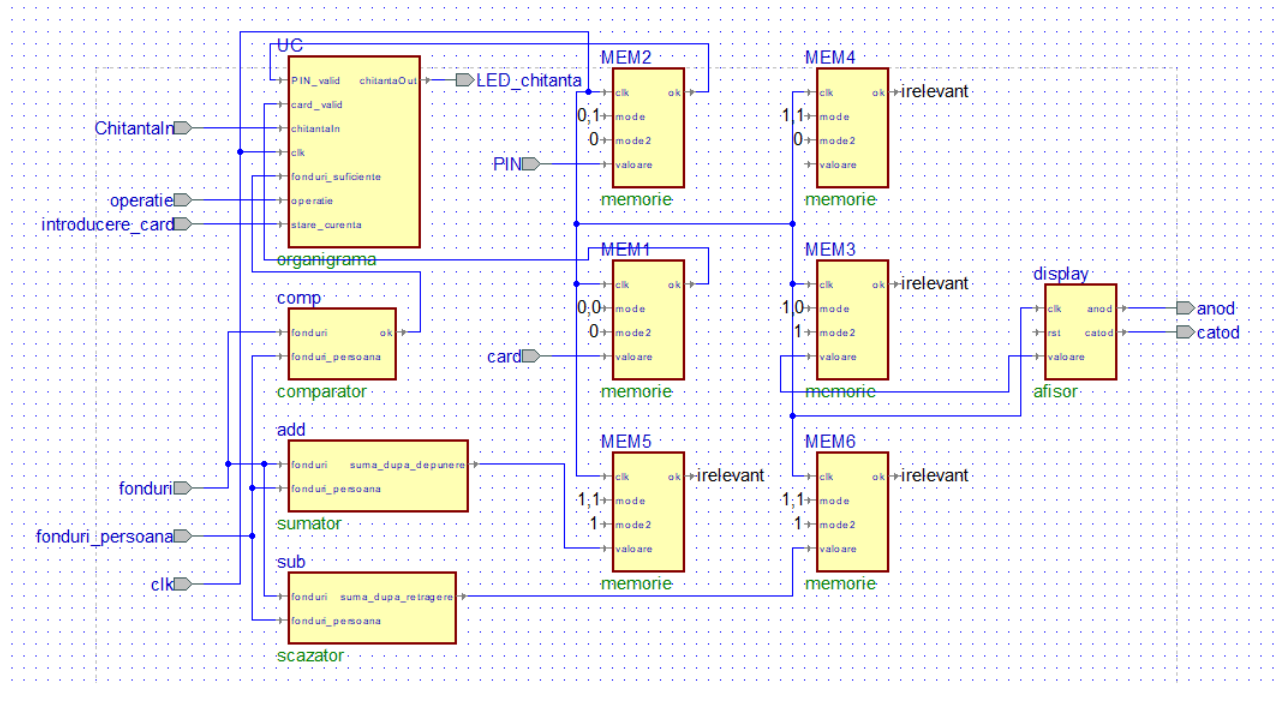
## CUPRINS

1. Prezentare proiect
2. Schema bloc cu componentele principale
3. Organigrama
4. Unitatea de comanda
5. Etape de proiectare
  - a) Afisorul*
  - b) Memoria*
  - c) Comparator*
  - d) Sumator*
  - e) Scazator*
  - f) Sonny*
6. Justificarea soluției alese
7. Instrucțiuni de utilizare și întreținere
8. Posibilități de dezvoltare ulterioare

## **1.PREZENTARE PROIECT A8**

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. Inițial se efectuează identificarea cardului și se alege operația. Vor fi suportate minim 4 carduri/conturi diferite și se vor implementa minimum 4 operații diferite. Automatul dispune de o casă în care inițial se introduce o anumită sumă (număr de bancnote de diferite valori). În cazul cererii de eliberare de numerar se introduce suma, se verifică existența sumei cerute, se vizualizează tipurile de bancnote emise și se actualizează contul. Apoi se eliberează cardul, suma și, eventual, chitanța. Proiectul va fi realizat de 2 studenți.

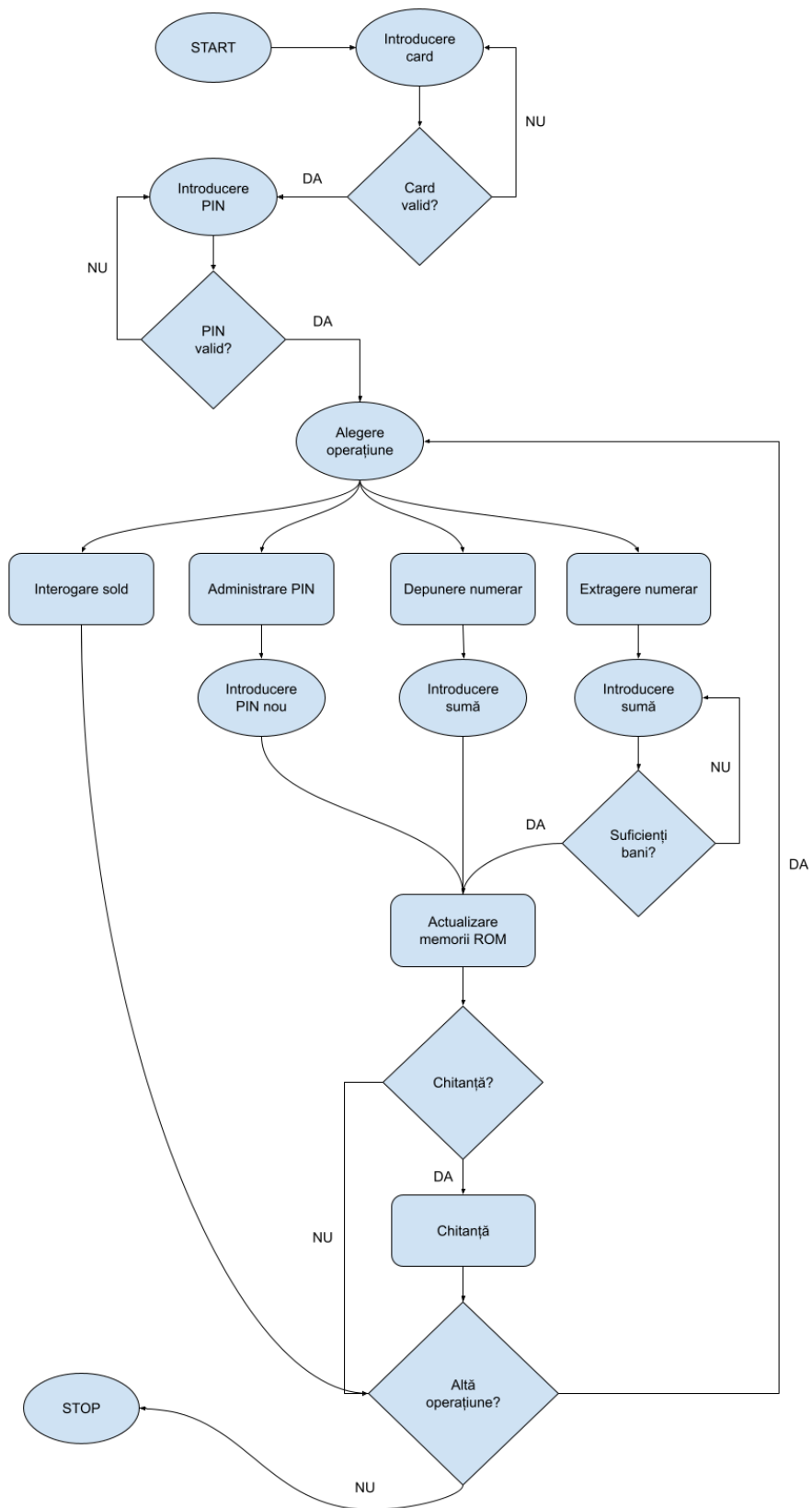
## 2.SCHEMA BLOC CU COMPONENTELE PRINCIPALE



### Componentele principale:

- 1 unitate de control = organigrama.vhd (inspirata din organigrama din imaginea urmatoare)
- 6 instante de memorie
- un afisor
- 2 sumatoare
- 2 scazatoare
- un comparator

### 3.ORGANIGRAMA



## 4.UNITATEA DE COMANDA

**Unitatea de comanda e reprezentata de organigrama.vhd si cuprinde urmatoarele stari:**

Primeste ca input un numar de card. In cazul in care acesta este identic cu unul dintre cele patru stocate in memorie, va fi considerat valid si se va trece in starea urmatoare. De asemenea, se va tine minte cardul folosit. Se va introduce apoi un PIN. Se va verifica daca acesta corespunde cu cel din adresa de memorie a cardului validat anterior. In caz contrar, se va introduce alt PIN. Primeste mai departe un alt input (selectiile) si se va trece la una din cele 4 operatiuni codificate

Prima operatiune este interogarea soldului. Se va extrage din memorie suma de pe card si va fi afisata pe 4-digit-7-segment display.

Cea de-a doua operatie este schimbarea PIN-ului, prin care se va suprascrie PIN-ul cardului cu noua valoare primita (input).

A treia operatiune este depunerea de numerar prin care se va suprascrie suma de pe card cu suma dintre cea deja existenta si cea introdusa de utilizator.

Ultima operatiune este retragerea numerar. Se va introduce suma care se doreste a fi extrasa, verificandu-se daca aceasta exista in bancomat. In caz afirmativ, suma va fi actualizata cu diferenta dintre cea existenta pe card si cea introdusa ulterior.

## 5.ETAPE DE PROIECTARE

*Sonny, the ATM* cuprinde urmatoarele componente:

- sonny.vhd
- organigrama.vhd
- memorie.vhd
- pachet\_stari.vhd
- afisor.vhd
- sumator.vhd
- scazator.vhd
- comparator\_mic.vhd

**Starile** se gasesc in pachet\_stari.vdh si sunt denumite sugestiv:

- introducere\_card
- introducere\_PIN
- alegere\_operatiune
- interogare\_sold
- introducere\_PIN\_nou
- actualizare PIN
- introducere\_suma\_depunere
- actualizare\_cont\_depunere
- introducere\_suma\_retragere
- actualizare\_cont\_retragere
- emitere\_chitanta
- alta\_operatiune

## a) AFISORUL

Pentru a afisa valorile s-a folosit un numarator. Bitii cei mai semnificativi sunt folositi ca selectii pentru a improspata anozii si valorile de pe ei. Catozii sunt setati cu ajutorul unui decodificator, totul conform documentatiei de la “Arhitectura Calculatoarelor”.

## b) MEMORIA

In memorie sunt stocate 4 numere de card, 4 PIN-uri si 4 sume de bani in formatul urmator:

```
"1001111010010011", "1011101000100010", "0110010100001010", "1111010000011100", -- carduri  
"1001011101010111", "1101000011001100", "0011101010001101", "1110001010011100", -- PIN-uri  
"0000101101001101", "0010101001001110", "0001000100010001", "0011100110100111" -- sume
```

In structura sa se disting 4 secvente:

- Secventa „00” face ca memoria sa citeasca numarul de card spre a-l valida ulterior
- Secventa „01” face ca memoria sa citeasca PIN-ul pentru a-l valida ulterior
- Secventa „10” face ca memoria sa citeasca suma de pe card, pe care o foloseste ulterior la sumator, scazator, comparator pentru actualizarea unitatii de control
- Secventa „11” realizeaza, in functie de urmatoarea selectie, „mode2”, suprascrierea peste PIN-ul actual in cazul in care se doreste schimbarea lui (daca „mode2” e 0) sau suprascrierea peste suma in cazul restragerii sau depunerii altei sume (cand „mode2” este 1).



## c) ORGANIGRAMA

Organigrama.vhd este implementarea in VHDL a unitatii de control, care cuprinde intrarile „card\_valid”, „PIN\_valid”, „fonduri\_suficiente”, „chitanta”, „revenire\_operatiuni”, „operatiune”, „LED\_chitanta”, „stare\_curenta”, legate la diverse componente, ca in schema bloc.

Modul prin care aceasta functioneaza este urmatorul: cand reset este setat pe 1, automatul se reseteaza. Starea curenta va fi introducere\_card, iar cand clock-ul este pe 1 si este pe front ascendent, unitatea de control isi va schimba starea.

```
when introducere_card =>
    if card_valid = '1' then stare_urmatoare <= introducere_PIN;
    else stare_urmatoare <= introducere_card;
    end if;
```

Se introduc pe rand numarul cardului si PIN-ul. Daca sunt corecte, se trece la selectarea operatiunii bancare. In caz contrar, se revine la reintroducerea lor pe rand. Cele patru operatii sunt: interogare sold, administrare PIN, depunere numerar si retragere numerar. La finalul ultimelor 3 se va elibera la alegerea utilizatorului, chitanta prin intermediul unui LED. La finalul sesiunii, utilizatorul poate alege sa efectueze o alta operatie bancara sau sa revina la inceput/ sa permita altui utilizator sa opereze automatul.

## d) COMPARATORUL

Comparatorul este pe 16 biti si se foloseste pentru a verifica daca exista fonduri suficiente pe cardul curent, comparand suma din bancomat cu cea care se doreste a fi extrasa. In cazul in care nu exista fonduri, semnalul de iesire va fi 0.

```
entity comparator_mic is
    port(A, B: in std_logic_vector(0 to 15);
          rez: out std_logic);
end comparator_mic;

architecture comportamental of comparator_mic is
begin
    process(A, B)
    begin
        if A < B then
            rez <= '0'; -- pentru fonduri suficiente vom returna 1
        else
            rez <= '1';
        end if;
    end process;
end comportamental;
```

## e) SUMATORUL

Sumatorul este, asemenea comparatorului, pe 16 biti. Acesta se foloseste pentru a aduna sumei de pe card noua suma care urmeaza a fi depusa in cont.

```
entity sumator is
    port(A, B: in std_logic_vector(0 to 15);
          S: out std_logic_vector(0 to 15));
end sumator;

architecture comportamental of sumator is
begin
    S <= A + B;
end comportamental;
```

## f) SCAZATORUL

Scazatorul este folosit pentru a actualiza suma ramasa in bancomat, dar si in cont in urma extragerii sumei. Este de asemenea pe 16 biti.

```
entity scazator is
    port(A, B: in std_logic_vector(0 to 15);
         D: out std_logic_vector(0 to 15));
end scazator;

architecture comportamental of scazator is
begin
    D <= A - B;
end comportamental;
```

## g) SONNY

„Sonny.vhd” reprezinta partea din proiect in care leaga componentele.

In prima parte, sunt prezentate toate componentele folosite (cele mentionate si in schema bloc), in modul urmator:

```
component afisor
    port(valoare: in std_logic_vector(15 downto 0);
         clk: in std_logic;
         catod: out std_logic_vector(6 downto 0);
         anod: out std_logic_vector(3 downto 0));
end component;
```

Cea de-a doua parte a codului cuprinde semnalele intermediare

```
signal card_valid: std_logic;
signal PIN_valid: std_logic;
signal cardsemnal: std_logic_vector(15 downto 0);
signal PINsemnal: std_logic_vector(15 downto 0);
signal fonduri_suficiente: std_logic;
signal stare_curenta: type_states;
signal val_de_afisat: std_logic_vector(15 downto 0);
signal PIN_nou_semnal: std_logic_vector(15 downto 0);
signal suprascriere_suma_semnal_depunere: std_logic_vector(15 downto 0); --
signal suprascriere_suma_semnal_retragere: std_logic_vector(15 downto 0); --
```

In ultima parte se initializeaza semnalele, pentru a respecta sintaxa limbajului (pentru a evita erorile cauzate de porturile inout):

```
begin
    cardsemnal <= card;
    PINsemnal <= PIN;
    card_valid <= '0';
    PIN_valid <= '0';
    fonduri_suficiente <= '0';
    stare_curenta <= introducere_card; -- nu sensibilitate
    PIN_nou_semnal <= PIN_nou;
end process;
```

## 6.JUSTIFICAREA SOLUTIEI ALESE

Dorinta de a nu avea un automat bancar care sa faca totul pe un singur front de clock ne-a impins catre proiectarea unei unitati de comanda (organigrama.vhd). Celelalte componente urmaresc firul logic al functionarii bancomatului.

## 8.POSIBILITATI DE DEZVOLTARE ULTERIOARE

Automatul bancar are diverse posibilitati de dezvoltare ulterioare: adaugarea de noi carduri, PIN-uri si sume, memorarea unui nou card in memorie atunci cand este folosit pentru prima data sau o diversitate mai mare in ceea ce priveste operatiunile care pot fi efectuate.

Contrar proiectului Sonny, the ATM, bancomatele actuale au posibilitatea de a alege limba in care se doreste sa se efectueze operatiunile sau de a memora mai multe conturi pe acelasi card, sub acelasi nume si numar, oferind utilizatorului posibilitatea de a alege contul asupra caruia doreste sa actioneze.

De asemenea, bancomatele actuale percep comisioane pentru retragerile de numerar, depinzand de suma extrasa sau banca aleasa de utilizator. O posibilitate de dezvoltare a proiectului este calculul unui comision imaginar, care, mai apoi, va fi scazut din suma ramasa pe card in momentul efectuarii retragerii.

Pe langa acestea, fiecare bancomat afiseaza mesaje precum „Doriti sa continuati?” pentru fiecare operatiune realizata.