

# Cola vending machine

**Cuprins:**

1. Cerința – pagina 2
2. Placa Nexys4DDR – pagina 2
3. Schema Bloc – pagina 3
4. Organigrama – pagina 5
5. Unitatea de control și unitatea de execuție – pagina 7
6. Componentele folosite – pagina 8
7. Manual de utilizare – pagina 18
8. Justificarea soluției alese – pagina 19
9. Posibilități de dezvoltări ulterioare – pagina 20

## **1. Cerința**

Să se proiecteze un automat distribuitor de Coca Cola. Prețul este de 1 leu. Se acceptă monede de 5, 10 și 50 bani. Sistemul este prevăzut cu 5 fotocelule:

- F0 – pentru moneda de 5 bani;
- F1 – pentru moneda de 10 bani;
- F2 – pentru moneda de 50 bani;
- F3 – pentru respingere monedă (alta decât cele acceptate) sau corpuri străine;
- F4 – pentru semnal de acceptare a monedei.

Dacă nu există Coca Cola atunci nu se acceptă nici un tip de monezi . Se face verificarea pentru suma totală și monezile sunt returnate dacă suma nu este completă. Se eliberează rest, dacă este cazul.

Se generează semnale și se semnalizează pentru acceptarea unei monezi, a totalului și pentru eliberarea de Coca Cola.

## **2. Placa Nexys4DDR**

Placa folosită pentru realizarea acestui proiect este Nexys4DDR, a cărei imagine este prezentată mai jos.

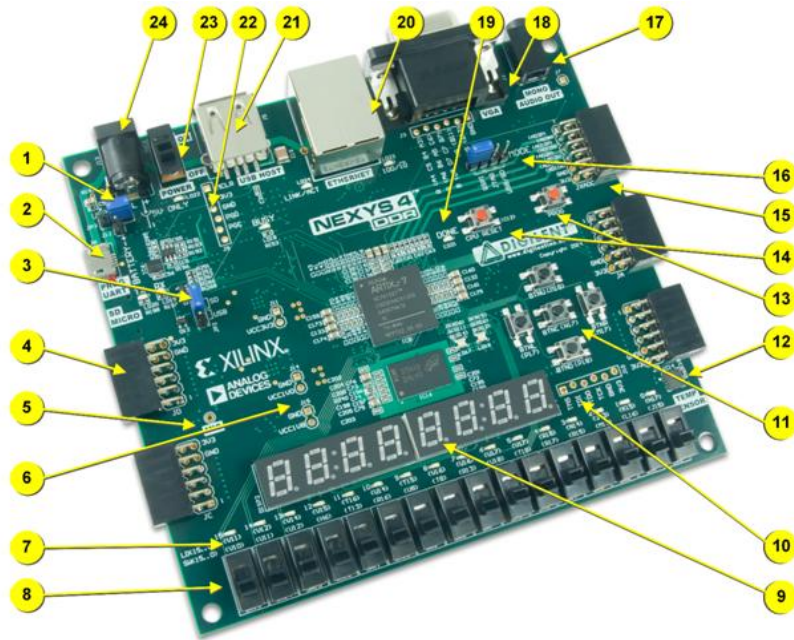
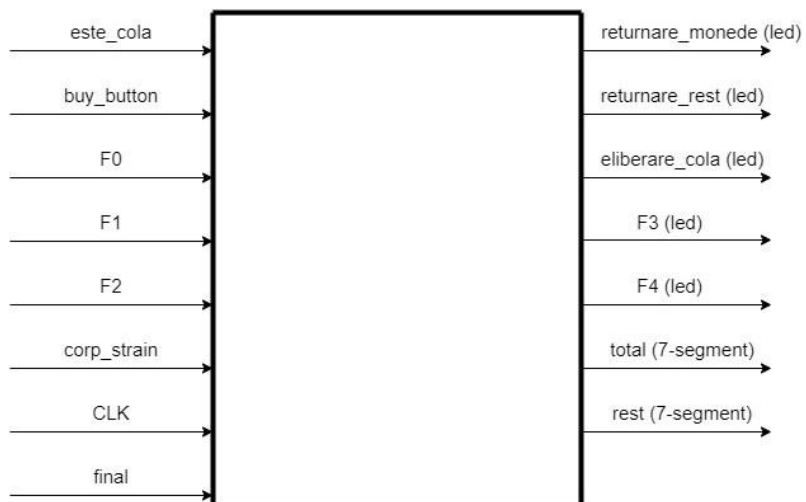


Figure 1. Nexys4 DDR board features.

Aceasta dispune de 16 switch-uri (numărul 8), 16 led-uri (numărul 7), 5 butoane (numărul 11) și 8 afișoare 7-segment (numărul 9). În realizarea proiectului am folosit 2 switch-uri, 5 led-uri, toate cele 5 butoane, precum și 4 afișoare 7-segment din cele 8 disponibile.

### 3. Schema Bloc



În imaginea de mai sus este prezentată schema bloc a distribuitorului (cutia neagră a acestuia). Aici se pot observa intrările pe care le primește automatul și ieșirile pe care le generează.

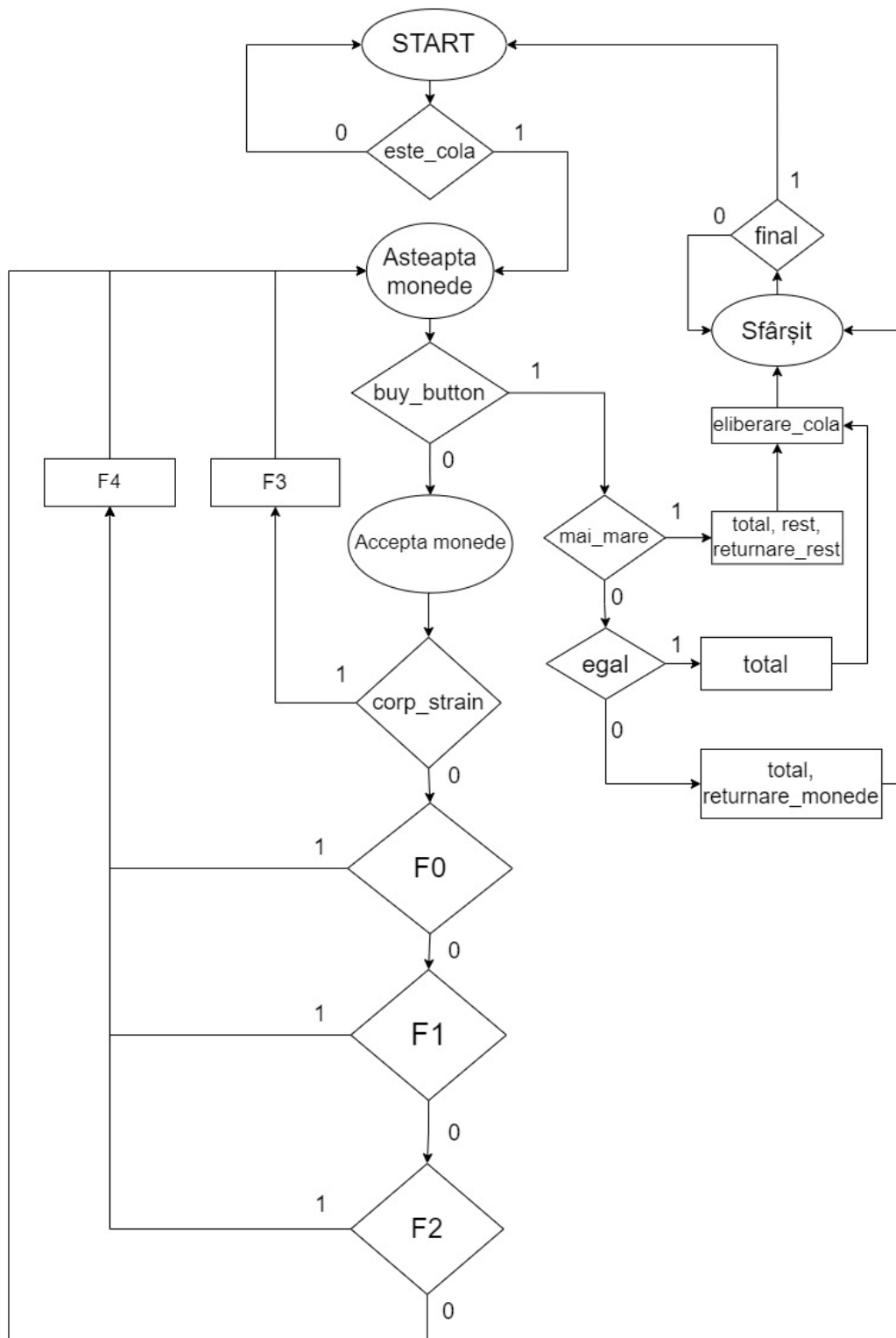
#### Intrări:

- este\_colă – acest semnal specifică dacă mai există Coca-Cola în distribuitor;
- buy\_button – semnal primit la apăsarea butonului de cumpărare de către utilizator
- F0, F1, F2 – semnale primite la introducerea unei monede de 5, 10 sau 50 de bani;
- corp\_străin – semnal primit la introducerea unui corp străin sau a unei monede care nu este acceptată
- CLK – semnalul de tact
- final – semnal primit de la utilizator după primirea dozei de Coca-Cola și a restului (dacă este cazul).

#### Ieșiri:

- returnare\_monede – semnal generat la returnarea monedelor (sumă introdusă insuficientă);
- returnare\_rest – semnal generat la returnarea restului;
- eliberare\_colă – semnal generat la eliberarea dozei de Coca-Cola;
- F3 – semnal generat la respingerea unei monede care nu este acceptată sau a unui corp străin;
- F4 – semnal generat la acceptarea unei monede;
- total – suma introdusă este afișată pe 2 afișoare 7-segment;
- rest – restul ce trebuie eliberat este afișat pe 2 afișoare 7-segment.

## 4. Organigrama



Mai sus se află organigrama distribuitorului de Coca-Cola. Stările sunt reprezentate în cerceulețe. Deciziile luate în fiecare stare sunt reprezentate prin romb. Ieșirile generate de fiecare decizie și tranziție dintr-o stare în alta sunt reprezentate în dreptunghiuri. În interiorul dreptunghiurilor sunt enumerate ieșirile adevărate în acel moment.

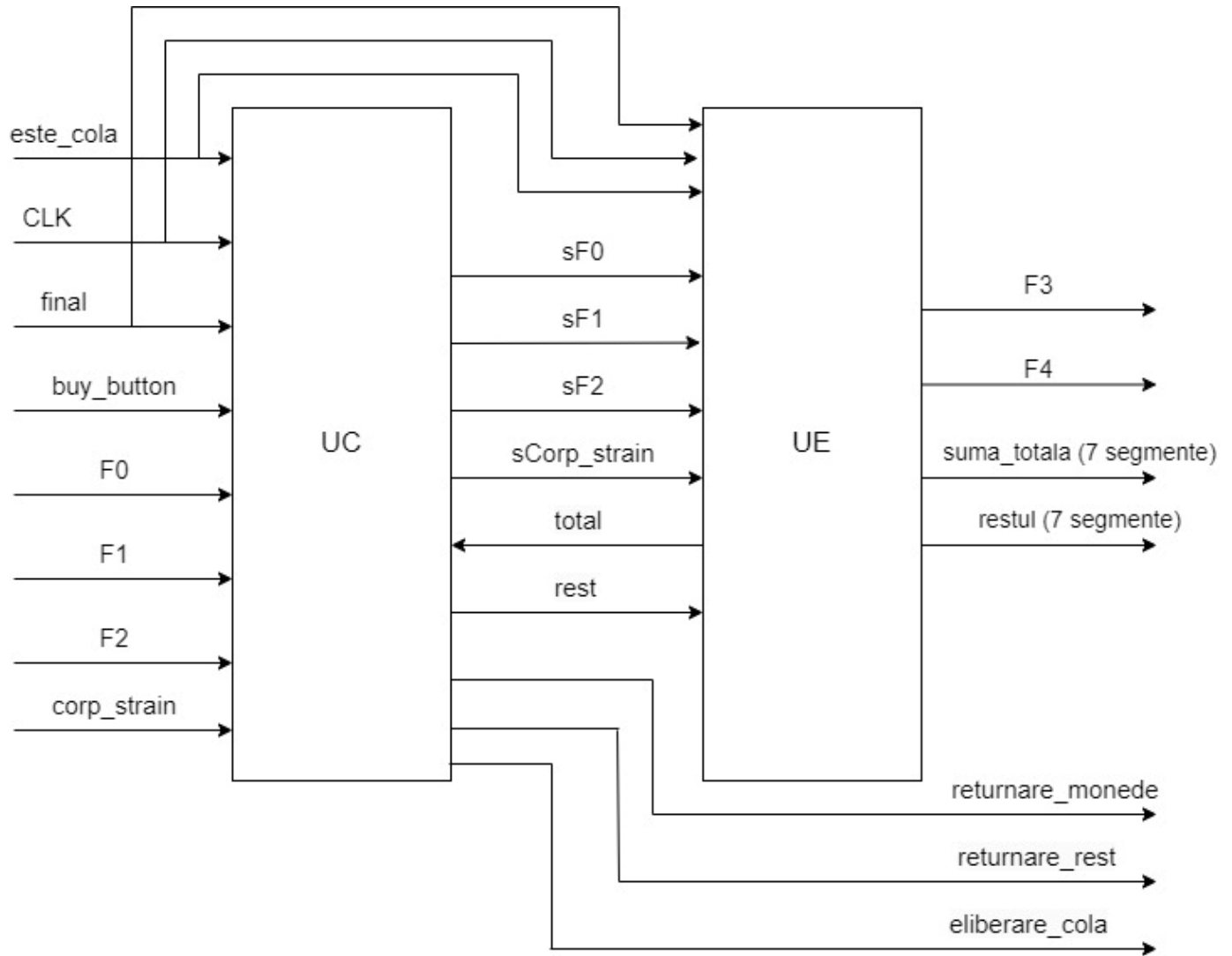
Distribuitorul pleacă din starea START. Dacă  $este\_cola = '0'$ , atunci nu există Cola în distribuitor. În acest caz, nu se acceptă niciun tip de monezi, revenindu-se în START.

Dacă  $este\_cola = '1'$  (este Cola în distribuitor), se face tranziția în starea ASTEAPTA MONEDE. În cazul în care butonul de cumpărare ( $buy\_button$ ) nu s-a activat, se trece în starea ACCEPTĂ MONEDE. La fiecare introducere a unei monede (5 bani, 10 bani, 50 bani sau corp străin/altă monedă decât cele acceptate) se generează semnalele F4/F3, iar apoi se revine în starea ASTEAPTA MONEDE.

În momentul în care se activează butonul de cumpărare, suma acumulată până în acel moment se compară cu prețul unei doze (1 leu). Dacă suma este mai mare, se afișează totalul, restul, se returnează restul și se eliberează Cola. Se face apoi tranziția în starea SFÂRȘIT. Dacă suma este egală cu prețul unei doze, se afișează totalul, se eliberează Cola și se trece în starea SFÂRȘIT. Dacă suma este mai mică, se afișează totalul iar monedele introduse sunt returnate. Se ajunge în starea SFÂRȘIT.

Se așteaptă activarea semnalului *final*. Activarea acestuia provine de la utilizator și confirmă faptul că a primit doza de Cola și restul corespunzător, ori i-au fost returnate monedele în cazul în care a introdus o sumă insuficientă de bani. La activarea acestui semnal, se ajunge înapoi în starea START. Ciclul se va relua la folosirea distribuitorului de către următorul utilizator.

## 5. Unitatea de control și unitatea de execuție



În imaginea de sus este prezentată descompunerea proiectului în unitatea de control (UC) și unitatea de execuție (UE). Se pot observa intrările și ieșirile, precum și semnalele prin care comunică între ele cele două unități. Acestea sunt esențiale pentru funcționarea corectă a întregului sistem.



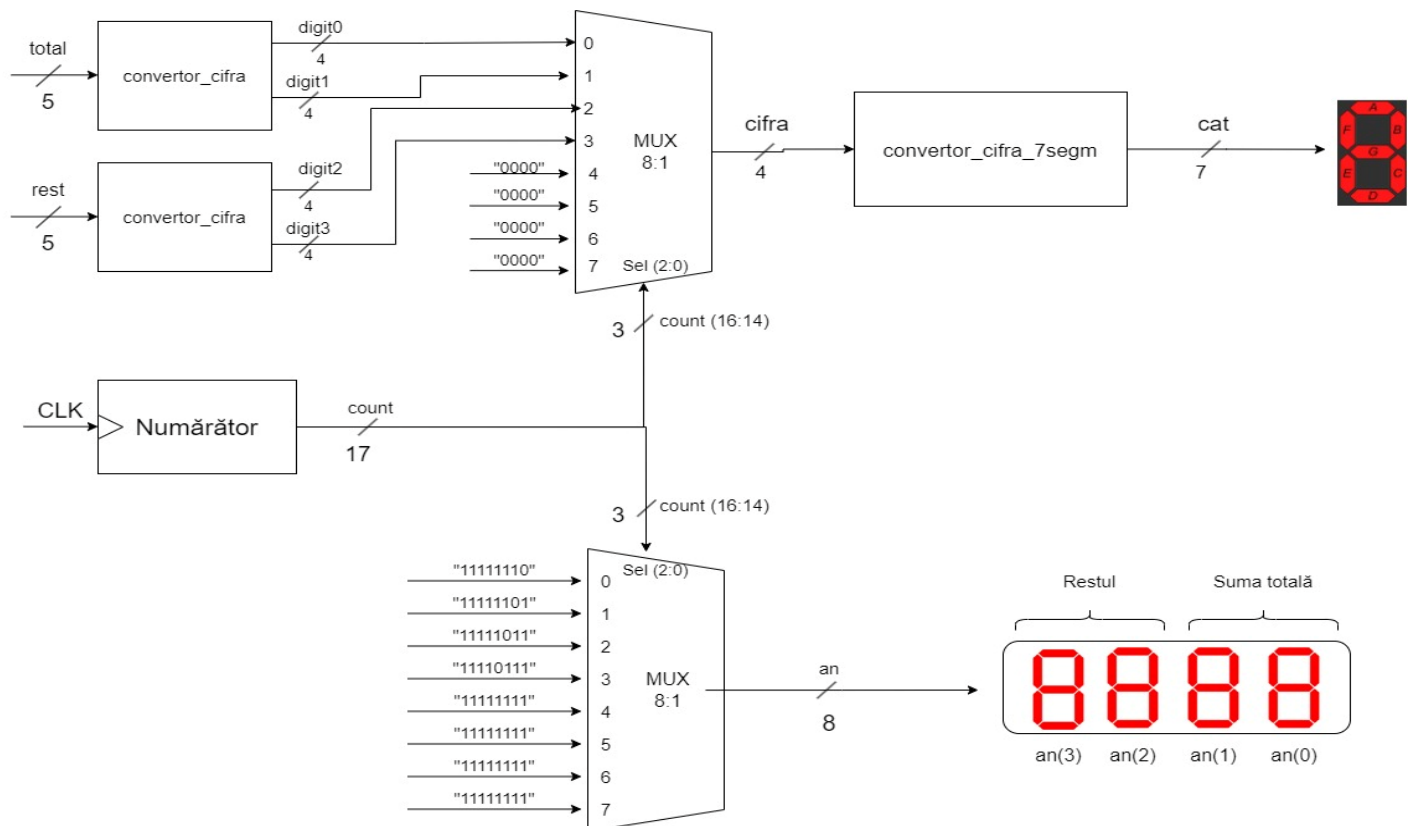
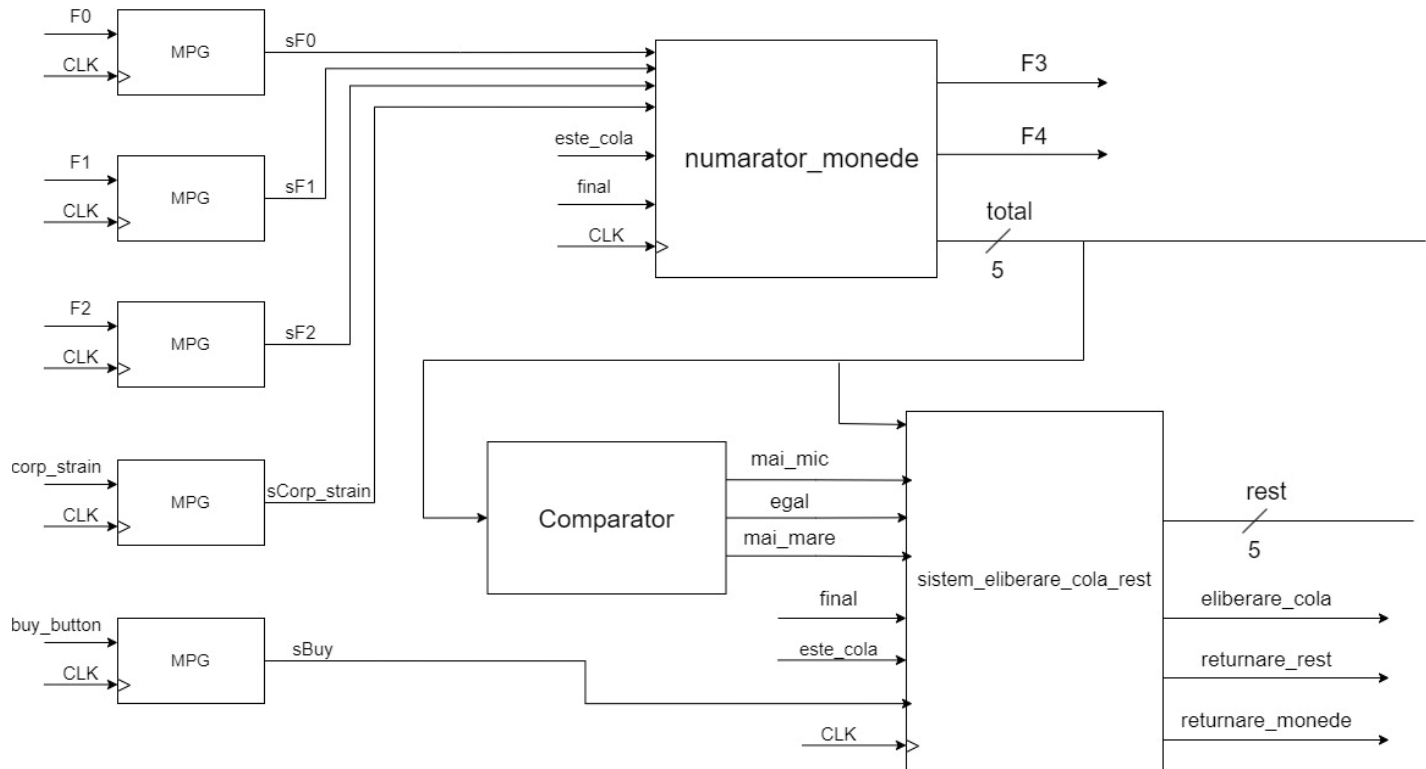
Putem împărți atât intrările cât și ieșirile în 2 categorii: de date și de control.

- Intrări de date: *F0*, *F1*, *F2*, *corp\_strain* (pentru introducerea monedelor), *este\_cola*;
- Intrări de control: *buy\_button* (buton pentru cumpărarea unei doze de Cola), *final* (semnal pentru confirmarea eliberării dozei de Cola și a restului / pentru confirmarea returnării monedelor în cazul introducerii unei sume insuficiente de bani);
- Ieșiri de date (valori de afișat pentru utilizator): suma totală de bani introdusă, restul datorat (afișate folosind afișoarele 7-segment);
- Ieșiri de control (semnale de avertizare sau atenționare a utilizatorului, prin care noi putem să controlăm și să îndrumăm utilizatorul prin funcționarea sistemului): *F3*, *F4* (pentru a avertiza utilizatorul dacă moneda introdusă a fost acceptată sau respinsă), *returnare\_monede*, *returnare\_rest*, *eliberare\_cola* (pentru a informa utilizatorul asupra comportamentului distribuitorului – returnează restul, eliberează doza de Cola, returnează monedele).

## 6. Componentele folosite

În următoarea parte sunt prezentate componentele folosite pentru implementarea proiectului, precum și o scurtă descriere a funcționalității acestora, însoțită de câteva snippet-uri de cod.

(Semnalele cu același nume provin de la aceeași sursă. Am folosit această convenție pentru a ușura realizarea desenului).

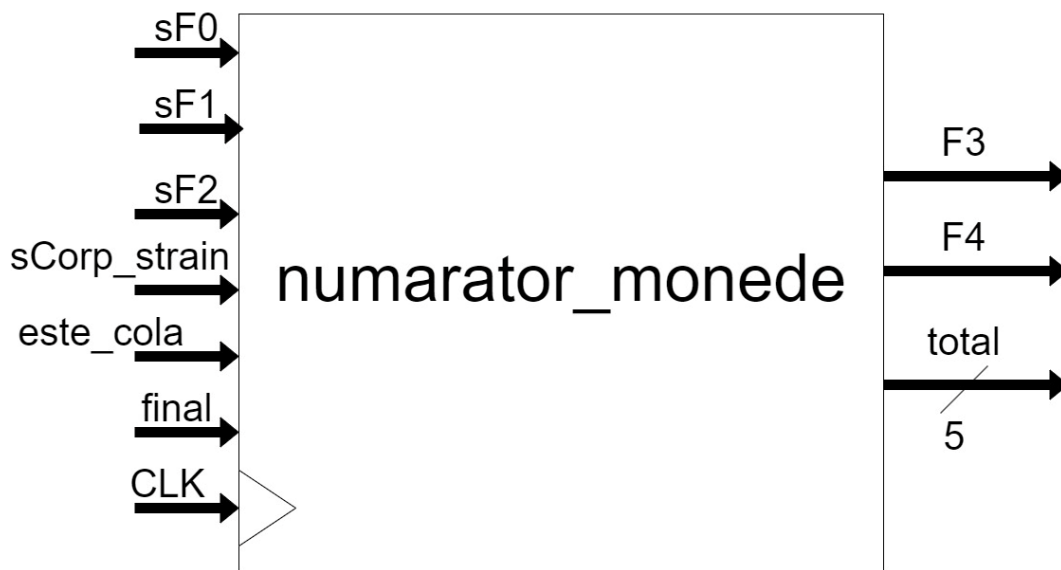


### 6.1. MPG



Am utilizat 5 MPG-uri pentru a filtra semnalele primite de la butoane, folosite pentru introducerea monedelor ( $F0, F1, F2, corp\_strain$ ) și pentru a cumpăra o doză ( $buy\_button$ ). Semnalele filtrate sunt  $sF0, sF1, sF2, sCorp\_strain$  și  $sBuy$ .

### 6.2. Numarator\_monede



Componenta *numarator\_monede* este responsabilă de calcularea sumei totale de bani introduse de către utilizator în distribuitor. În cazul în care semnalul *este\_cola*=0, niciun tip de monedă nu este acceptat. Dacă este Coca-Cola în distribuitor, atunci la fiecare introducere a unei monezi, semnalată de  $sF0, sF1$  și

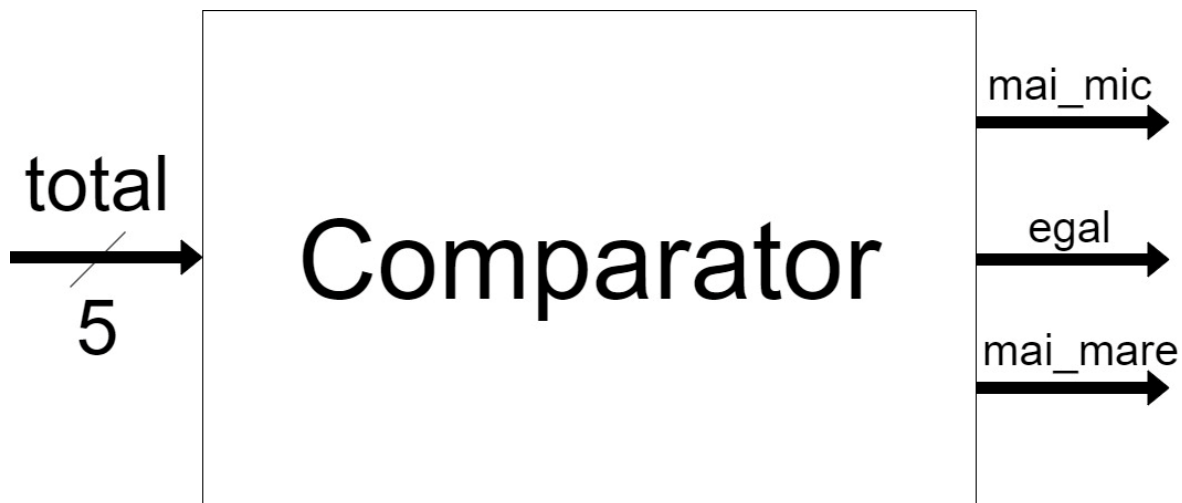
$sF2$ , suma totală va fi actualizată. De asemenea, ieșirea  $F4$  va determina aprinderea unui led care anunță utilizatorul că moneda i-a fost acceptată. În cazul respingerii unei monezi care nu este acceptată de către distribuitor (semnalul  $sCorp\_strain$ ),  $F3$  determină aprinderea unui led care avertizează utilizatorul asupra acestui fapt.  $Total$  este o valoare binară pe 5 biți, reprezentând suma de bani introdusă.

La activarea semnalului  $final$  ( $final = 1$ ), totalul revine la valoarea 0. Astfel, distribuitorul se resetează și își reia ciclul de funcționare la următoarea utilizare a acestuia.

Liniile de cod de mai jos exemplifică comportamentul acestei componente:

```
if final = '1' then
    count <= "00000";
    F3 <= '0';
    F4 <= '0';
elsif esteCola = '1' then
    if CLK = '1' and CLK'event then
        if F0 = '1' then
            count <= count + "1"; -- moneda de 5 bani introdusa
            F4 <= '1';
            F3 <= '0';
```

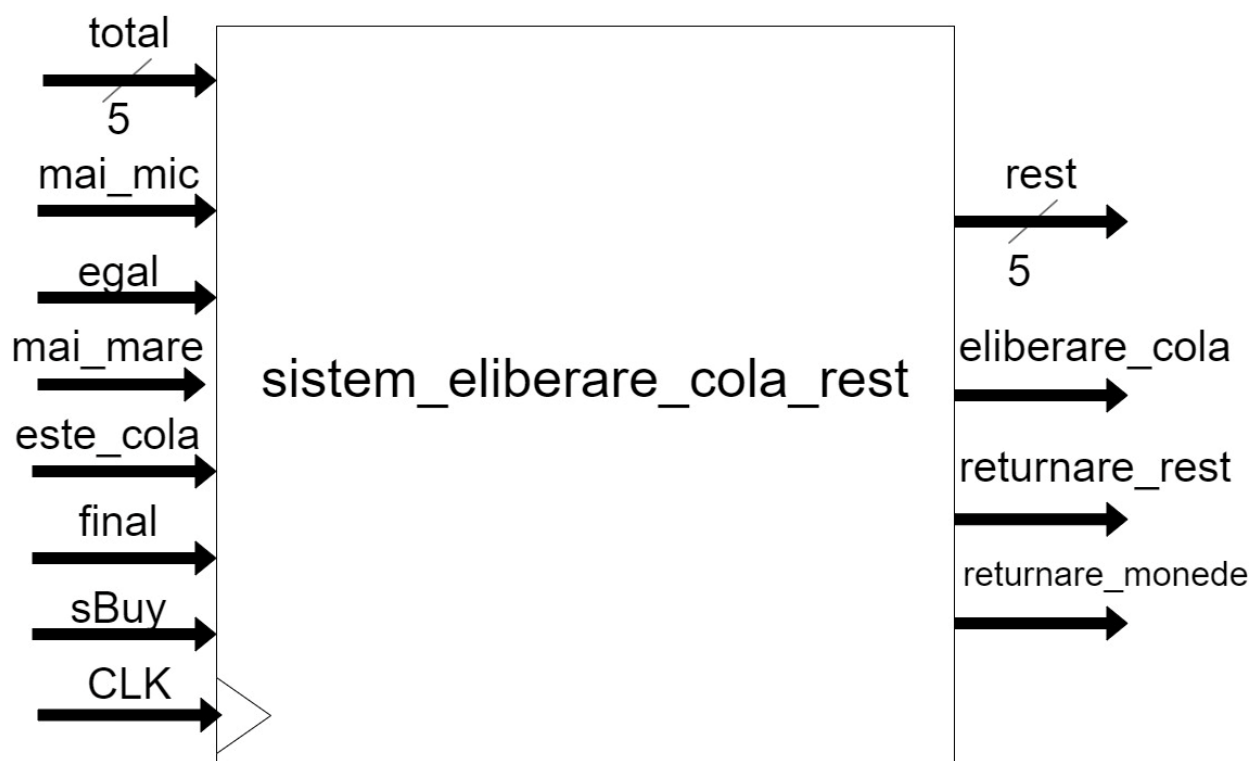
### 6.3. Comparator



Această componentă realizează comparația dintre suma de bani introdusă de către utilizator și prețul unei doze (1 leu). Valoarea semnalelor de ieșire *mai\_mic*, *egal* și *mai\_mare* reprezintă rezultatul comparației.

```
mai_mic <= '1' when (X < pretCola) else '0';
egal <= '1' when (X = pretCola) else '0';
mai_mare <= '1' when (X > pretCola) else '0';
```

#### 6.4. Sistem\_eliberareColaRest



Componenta este responsabilă de eliberarea dozei de Coca-Cola și a restului, respectiv de returnarea monedelor către utilizator în cazul în care suma introdusă este insuficientă pentru achiziționarea unei doze.

Dacă în distribuitor nu există Cola, achiziția unei doze nu este permisă. În caz contrar (*esteCola* = '1'), componenta va aștepta activarea semnalului *sBuy*. La

activarea acestuia, în funcție de intrările *mai\_mare*, *egal* și *mai\_mic*, sistemul va determina ceea ce are de făcut.

Dacă suma de bani introdusă este mai mare decât prețul unei doze, atunci se va calcula restul, iar semnalele de ieșire *eliberareCola* și *returnareRest* vor determina aprinderea a 2 led-uri care vor îndruma utilizatorul în acceptarea dozei și a restului. *Rest* este o valoare binară pe 5 biți ce memorează valoarea restului de eliberat.

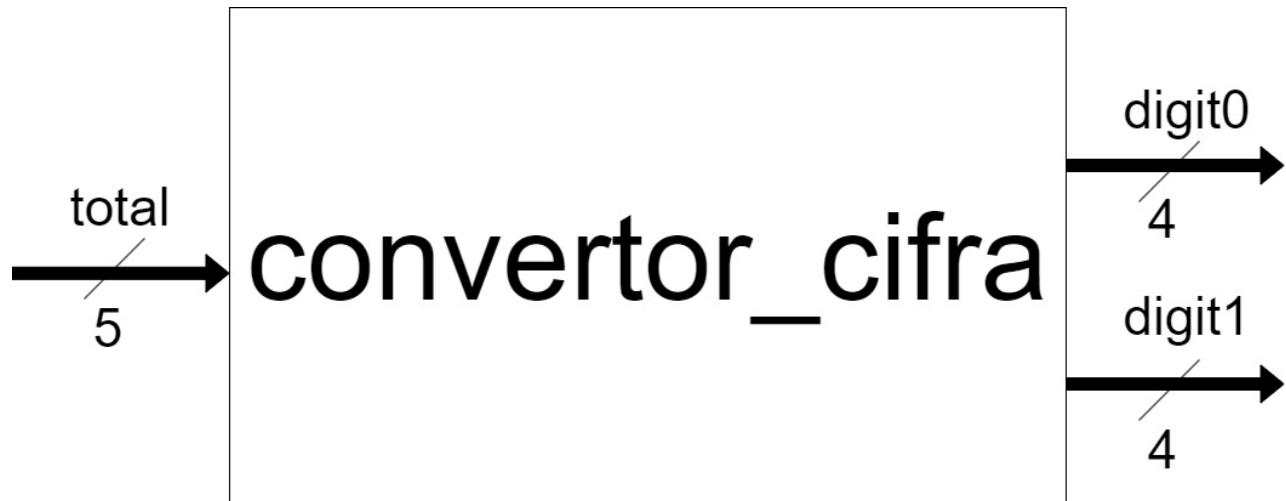
Dacă suma de bani este egală cu prețul unei doze, se va aprinde led-ul corespunzător eliberării de Coca-Cola pentru a informa utilizatorul.

Dacă suma de bani este mai mică decât prețul unei doze, utilizatorului nu i se va elibera Coca-Cola. Monedele introduse vor fi returnate iar semnalul *returnareMonede* determină aprinderea unui led care va informa utilizatorul asupra acestui fapt.

La activarea semnalului *final* (*final* = 1), ieșirile *eliberareCola*, *returnareRest* și *returnareMonede* primesc valoarea 0. De asemenea, restul devine și el 0. Astfel, distribuitorul revine în starea inițială.

```
elsif buy = '1' and rising_edge(CLK) then
    if esteCola = '1' then
        if maiMare = '1' then
            rest <= total - pretCola;
            eliberareCola <= '1';
            returnareMonede <= '0';
            returnareRest <= '1';
```

### 6.5. Convertor\_cifra

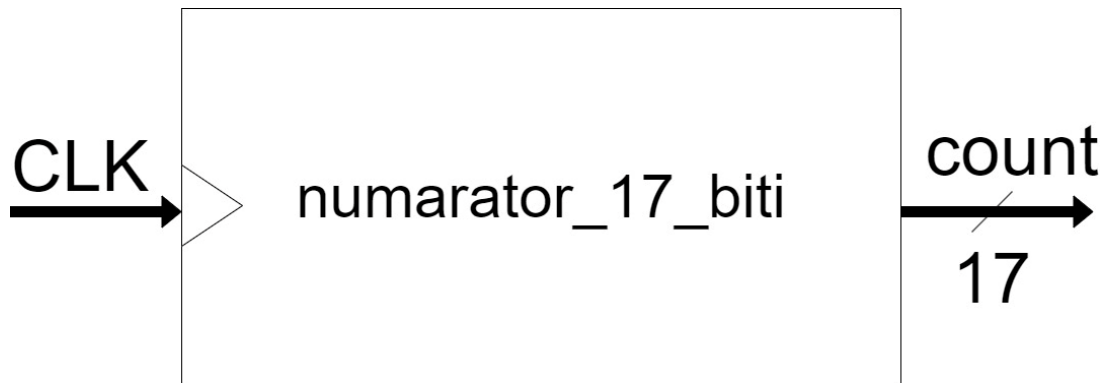


Această componentă realizează conversia unui număr binar în cifra unităților și cifra zecilor, acestea fiind reprezentate tot în binar. Un exemplu concret: dacă totalul este egal cu “01101” (13 în zecimal), *digit0* (cifra unităților) va primi valoarea “0011” (3 în zecimal), iar *digit1* (cifra zecilor), va primi valoarea “0001” (1 în zecimal).

```
when "01101" =>
    digit0 <= "0011";
    digit1 <= "0001";
```

În cadrul proiectului am folosit două astfel de convertoare, atât pentru *total* cât și pentru *rest*.

## 6.6. Numărător (pe 17 biți)

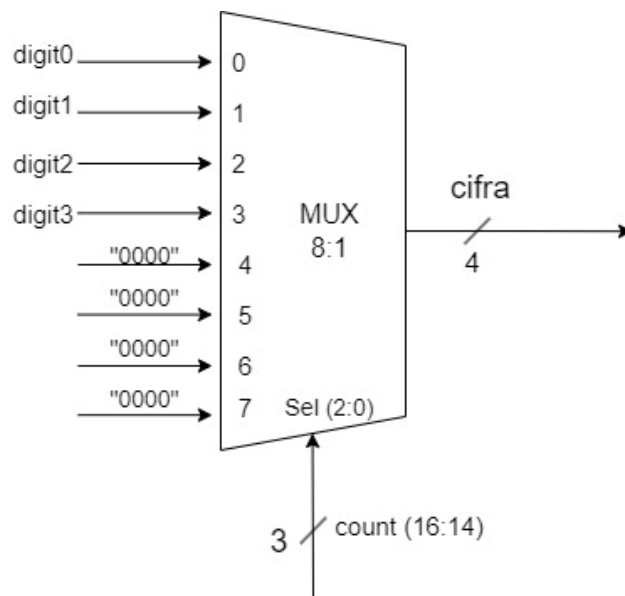


Acesta este un numărător pe 17 biți ce numără în bucla “000...0” – “111...1”.  
Ultimii trei biți ai ieșirii *count* (*count 16:14*) vor fi folosiți ca biți de selecție pentru cele 2 multiplexoare prezentate mai jos.

```
process(CLK)
begin
    if CLK = '1' and CLK'event then
        count <= count + 1;
    end if;

    num <= count;
end process;
```

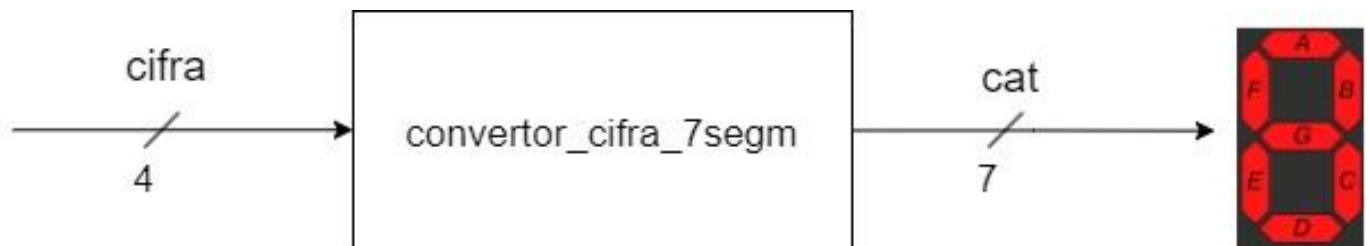
## 6.7. Multiplexor 8:1 (cu calea de date pe 4 biți)





În funcție de biții de selecție (*count 16:14*), acest multiplexor 8:1 cu calea de date pe 4 biți va selecta cifra care trebuie afișată pe display-ul 7-segment corespunzător, alegând dintre *digit0*, *digit1*, *digit2* și *digit3*. Celelalte intrări au primit valoarea "0000" deoarece vrem să folosim doar 4 din cele 8 afișoare 7-segment ale plăcii NexysDDR4.

#### 6.8. Convertor\_cifra\_7segm

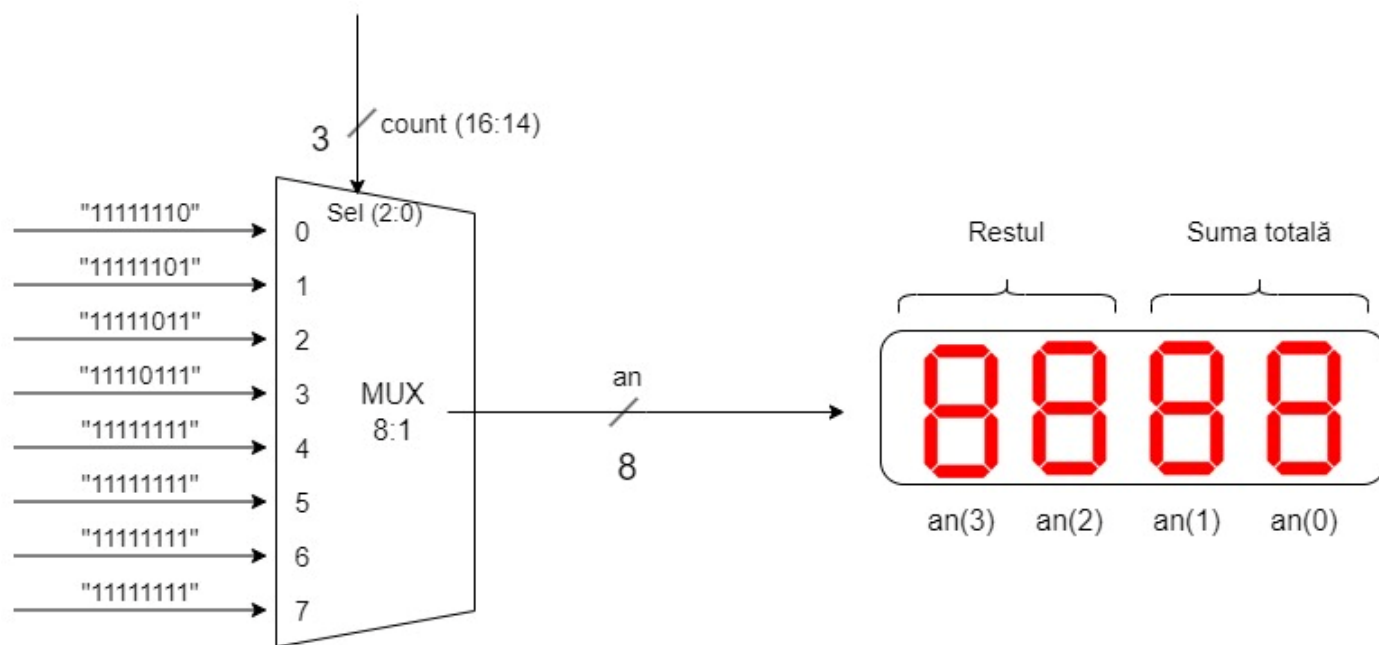


În funcție de valoarea binară a cifrei primite la intrare, acest convertor va genera valorile corespunzătoare ale vectorului de catozi (*cat 6:0*) pentru a reprezenta corect cifra pe afișorul 7-segment.

```

cat <= "1000000" when cifra="0000" else
      "1111001" when cifra="0001" else
      "0100100" when cifra="0010" else
      "0110000" when cifra="0011" else
      "0011001" when cifra="0100" else
      "0010010" when cifra="0101" else
      "0000010" when cifra="0110" else
      "1111000" when cifra="0111" else
      "0000000" when cifra="1000" else
      "0010000" when cifra="1001" else
      "1111111";
  
```

## 6.9. Multiplexor 8:1 (cu calea de date pe 8 biți)



Acest multiplexor cu calea de date pe 8 biți este folosit pentru a afișa fiecare cifră pe anodul corespunzător. Astfel, primele 2 cifre, de pe  $an(0)$  și  $an(1)$ , vor afișa suma totală introdusă în distribuitor. Următoarele 2 cifre, de pe  $an(2)$  și  $an(3)$  vor afișa restul ce trebuie eliberat. Ceilalți anodi vor fi stinși și nu vor afișa nimic, primind valoarea '1' (inactivi).

Pentru afișare am folosit următoarea convenție: am considerat că o monedă de 5 bani reprezintă o unitate.

```
-- 5 bani -> unitatea de masura
-- 5 bani -> 1 unitate
-- 10 bani -> 2 unitati
-- 50 bani -> 10 unitati
-- 20 unitati -> 1 cola
```

Astfel, toate valorile afișate au fost reprezentate în aceste unități. Pentru a converti o valoare afișată din unități în lei, aceasta trebuie înmulțită cu 0.05, obținându-se valoarea monetară reală. Spre exemplu,  $20 \text{ unități} * 0.05 = 1 \text{ leu}$ .

De asemenea, am folosit numărătorul pe 17 biți și cele 2 multiplexoare 8:1 pentru a putea afișa simultan valori diferite pe display-urile 7-segment.

!!! Datorită reprezentării valorilor totalului și restului pe 5 biți, valoarea maximă acceptată de către distribuitor este de 1.55 lei (31 de unități).

## 7. Manual de utilizare

Prețul unei doze este de 1 leu. Suma maximă care poate fi introdusă în distribuitor este de 1.55 lei. !!! În cazul în care nu există Coca-Cola în distribuitor, nu se acceptă niciun tip de monezi. Pentru a putea folosi distribuitorul de Coca-Cola, utilizatorul trebuie să respecte următorii pași:

1. Utilizatorul va introduce monedele în distribuitor. Monedele acceptate sunt cele de 5, 10 și 50 de bani. Se apasă butonul M18 pentru a introduce o monedă de 5 bani, butonul P17 pentru a introduce o monedă de 10 bani și butonul M17 pentru a introduce o monedă de 50 de bani. În momentul introducerii unei monezi acceptate, led-ul 4 se va aprinde pentru a confirma acceptarea monedei. Apăsarea butonului P18 va determina introducerea unei monezi care nu este acceptată. Led-ul 3 se va aprinde pentru a semnaliza acest fapt. La fiecare introducere a unei monezi, totalul se va actualiza și va fi afișat pe primele 2 display-uri 7-segment în unități (5 bani = 1 unitate). După finalizarea introducerii monedelor, utilizatorul trebuie să apese butonul N17. Acesta este butonul de cumpărare a unei doze (buy\_button).
2. În funcție de suma introdusă de către utilizator în distribuitor, există 3 cazuri:
  - Suma introdusă este mai mare decât prețul unei doze (1 leu). După apăsarea butonului de cumpărare, restul ce trebuie eliberat va fi afișat pe următoarele 2 display-uri 7-segment, lângă total. Led-ul 1 se va

aprinde pentru a atenționa utilizatorul asupra returnării restului de către distribuitor. Aprinderea led-ului 2 va confirma eliberarea dozei de Coca-Cola. Pentru a confirma primirea dozei și a restului, utilizatorul trebuie să folosească switch-ul 1. Astfel, distribuitorul va reveni în starea inițială, putând fi folosit de către următorul utilizator.

- Suma introdusă este egală cu prețul unei doze (1 leu). După apăsarea butonului de cumpărare, led-ul 1 se va aprinde, atenționând utilizatorul asupra eliberării dozei de Coca-Cola. Pentru a confirma primirea dozei, utilizatorul trebuie să folosească switch-ul 1. Astfel, distribuitorul va reveni în starea inițială, putând fi folosit de către următorul utilizator.
- Suma introdusă este mai mică decât prețul unei doze (1 leu). După apăsarea butonului de cumpărare, led-ul 0 se va aprinde. Astfel, utilizatorul va fi atenționat asupra introducerii unei sume insuficiente și asupra returnării monedelor de către distribuitor. Pentru a confirma faptul că monedele i-au fost returnate, utilizatorul trebuie să folosească switch-ul 1. În acest mod, distribuitorul va reveni în starea inițială, putând fi folosit de către următorul utilizator.

## **8. Justificarea soluției alese**

Am ales să realizez proiectul folosind o arhitectură structurală deoarece această metodă este concisă și clară, evidențiind intrările, ieșirile, precum și parcursul semnalelor interne care determină luarea unor decizii și efectuarea unor tranziții. Fiecare componentă este clar specificată, având o funcție precisă. Funcționarea distribuitorului este concepută cât mai simplu pentru utilizator, astfel încât să nu aibă probleme în folosirea acestuia.

O alternativă ar fi fost utilizarea unei arhitecturi comportamentale, descriind funcționarea distribuitorului cu stări. Totuși, am considerat implementarea structurală a acestui proiect mai facilă și mai eficientă.

## **9. Posibilități de dezvoltări ulterioare**

O posibilă dezvoltare a proiectului este înlocuirea semnalului de intrare *este\_cola* cu un semnal intern al distribuitorului, reprezentând numărul de doze de Coca-Cola rămase. În momentul în care acest semnal ajunge la valoarea 0, nu mai există doze în distribuitor, iar niciun tip de monedă nu mai este acceptat. Utilizatorul nu va putea achiziționa Coca-Cola până când distribuitorul nu este realimentat. În acest mod se poate adăuga o funcționalitate și o îmbunătățire distribuitorului deja realizat.