

PROIECT: Automat pentru cumpărarea biletelor de tren

Grupa 30216

Facultatea de Automatică și calculatoare

An studiu: 2018-2019

**Profesor indrumător laborator: Student:**

***Pop Diana Irena Arseniuc Anamaria***

Cuprins

**1. Specificatia proiectului**

**2. Proiectare**

a) Cutia neagra a automatului

b) Organigrama

c) Schema bloc

**3. Lista componentelor utilizate**

Descriere formala a componentelor

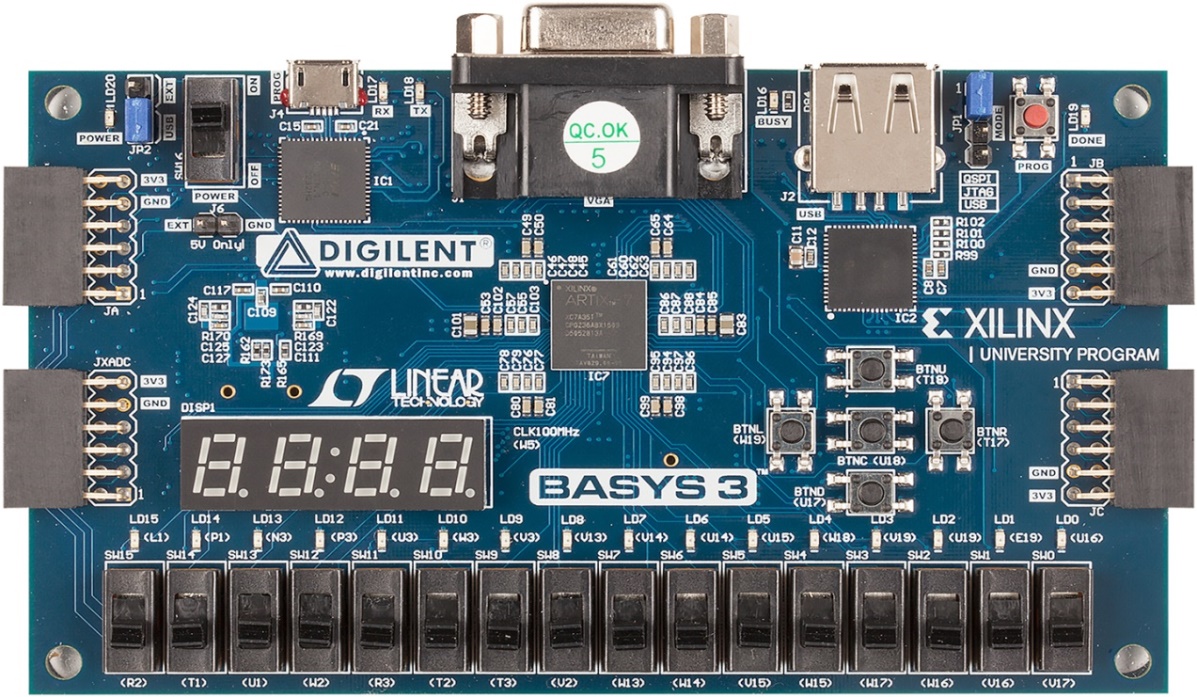
**4. Justificarea solutiei alese**

# 5. Instructiuni de utilizare si întretinere

# 6. Posibilitati de dezvoltare ulterioara

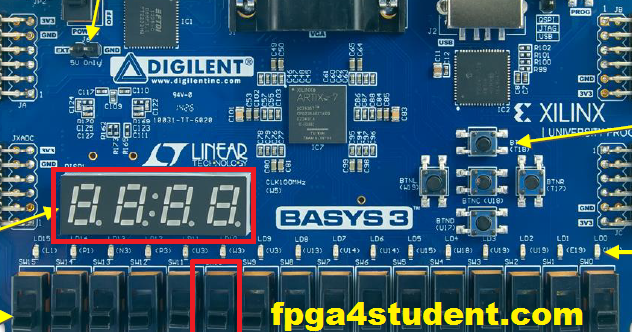
1.Specificație

**Cerința:** Să se proiecteze un automat pentru **cumpărarea biletelor de tren** pentru trafic intern. Cumpărătorul introduce distanţa până la destinaţie (în zeci de km). Costul biletului si sumele introduce sunt afişate pe afişoare 7 segmente. Moneda utilizată este EURO. Preţul maxim pentru un bilet este de 100 EURO. Automatul primeşte suma necesară în hârtii sau monede şi eliberează biletele şi, eventual, restul. Lipsa de bilete, introducerea unei sume mai mici decât costul biletului sau imposibilitatea restituirii restului se semnalizează luminos. Proiectul va fi realizat de **1 student.**



= > **Proiectul l-am realizat pe placuta** **Basys 3** care este o placă de dezvoltare FPGA la nivel de intrare proiectată exclusiv pentru Vivado Design Suite, cu arhitectura Xilinx Artix-7 FPGA.

Dupa cum se vede in imaginea de mai sus, aceasta detine :

* 4 afisoare pe 7 segmente
* 16 switch-uri
* 16 led-uri
* 5 butoane
* 

LED-URI

SWITCH-URI

AFISOARELE

BUTOANEE

**2. PROIECTARE**

1. Cutia neagra a automatului



4

ANULARE

6

Introducere

km

1. Organigrama

000



0

0

0

0

0

1

Btn 0

001

1

1

1

1

Introducere suma

Btn 0

Btn 0

Btn 0

Btn 0

Calcul

pret

011

Eliberare rest

100

Eliberare

bilet

101

Reset

010

Anulare

Pentru a înțelege mai ușor funcționalitatea automatului se poate construi o organigramă (sau diagrama bloc) deoarece se pot vizualiza etapele prin care trece automatul. Acestea trec dintr-o stare intr-alta doar daca sunt indeplinite anumite conditii: in cazul de fata, butonul 0 (cel din mijloc) face legatura dintre stari.

Daca btn0=1 se trece, in starea urmatoare, iar daca btn0=0, automatul sta in aceeasi stare in care se afla la momentul respective.

Starea 000: Starea in care se introduc banii

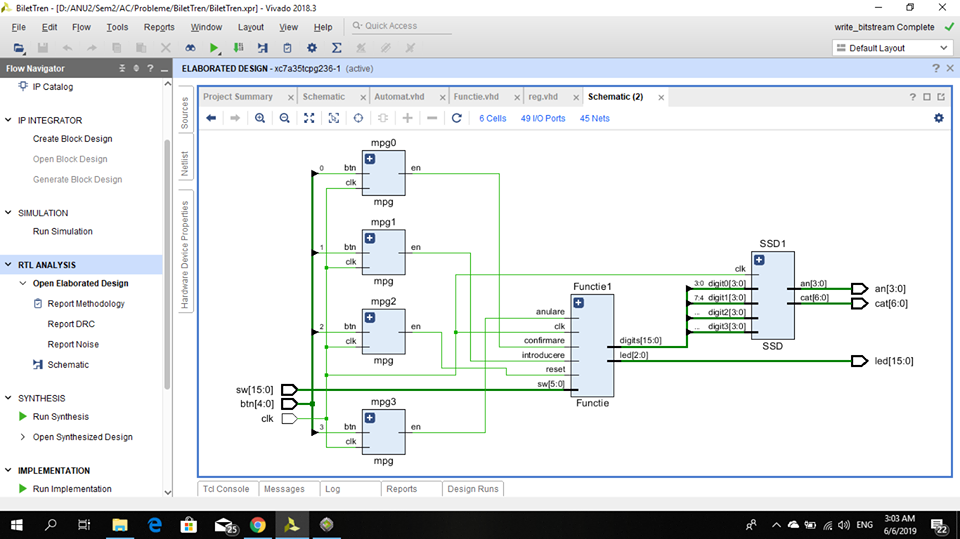
Starea 001: Starea in care se calculeaza pretul

Starea 010: Se introduc bancnotele

Starea 011: Se da restul

Starea 100: Se da biletul, in caz ca e cazul

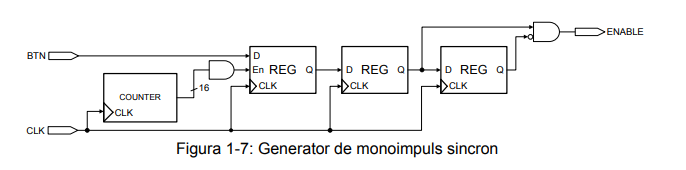
Starea 101: Este starea de RESET a automatului in care se revine la ce a fost inainte de realizarea operatiilor.

1. **Schema BLOC**

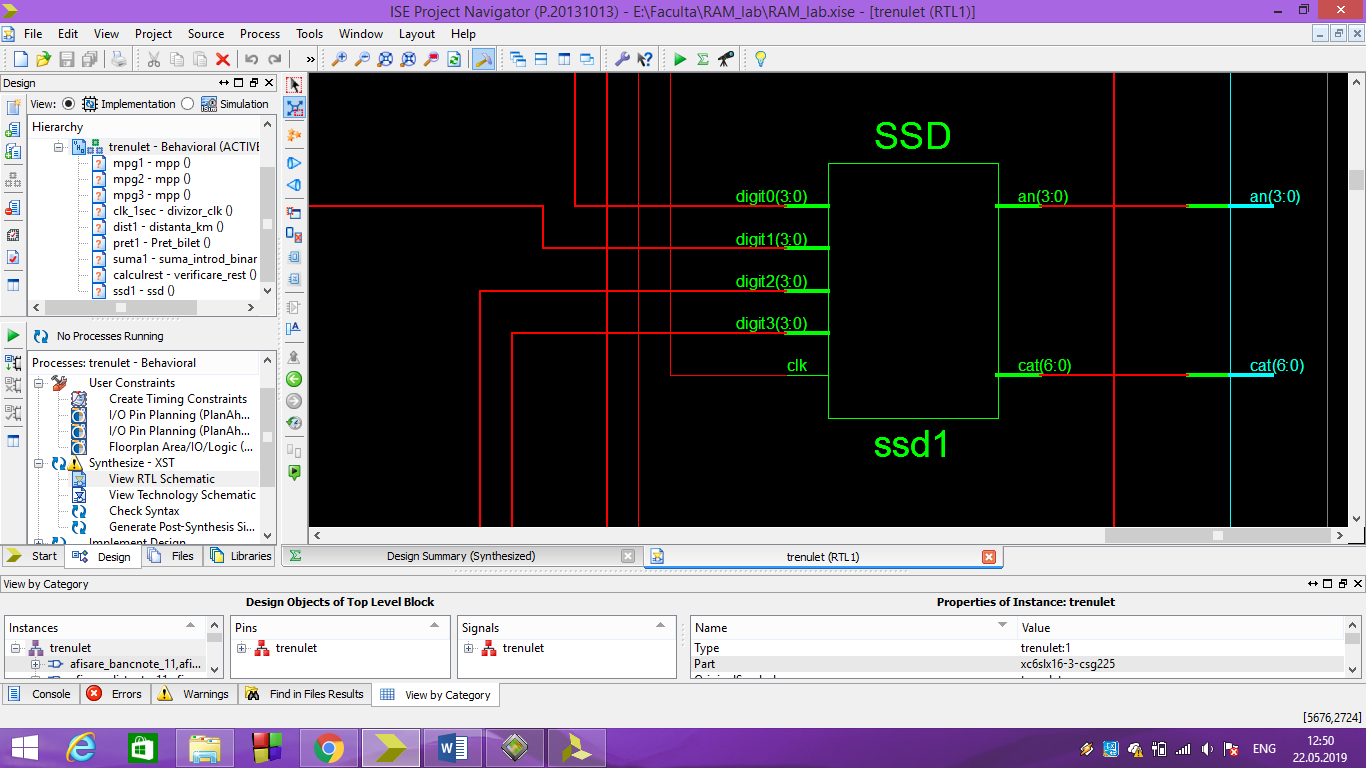
**3. LISTA COMPONENTELOR UTILIZATE**

* **Componenta MPG (**MonoPulse Generator**)**

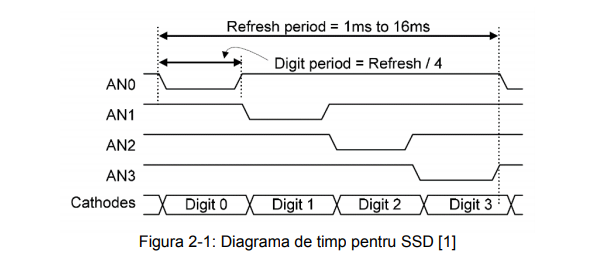
Circuitul necesar, care activează un semnal de ENABLE o singură dată la o apăsare a butonului, este prezentat în figura următoare:

****

Rolul numărătorului, împreună cu primului bistabil, este de a asigura robustețe la utilizarea butoanelor uzate fizic, când pot să apară activări multiple ale semnalului ENABLE la o apăsare de buton. În funcție de uzură, este posibil să fie nevoie de mai mulți biți ai numărătorului (17-20+) pe care să se aplice un ŞI logic, astfel încât să se mărească intervalul de eșantionare al butonului.

* **Afișorul pe 7 segmente cu 4 cifre** (Seven Segment Display - SSD)

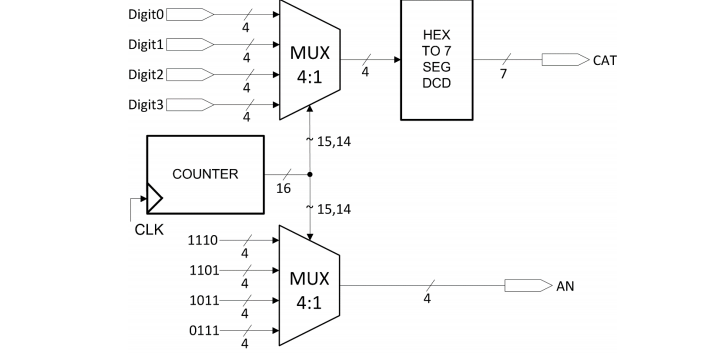
Această interfață folosește șapte leduri pentru fiecare cifră; fiecare cifră este activată de un semnal de anod. Toate semnalele interfeței SSD (7 semnale comune de catod și 4 semnale distincte de anod) sunt active pe 0. Semnalele de catod controlează ledurile care se aprind de pe acele cifre care au semnalul de anod activ (de exemplu dacă se activează toate 4 anodurile, atunci se va afișa aceeași cifră pe cele 4 poziții).



*Diagrama de timp SSD*

Pentru a afișa 4 cifre diferite pe SSD, este necesară implementarea unui circuit care trimite cifrele pe semnalele de catod ale SSD în concordanță cu diagrama de timp a SSD. Perioada maximă de reîmprospătare (refresh) este astfel calculată încât ochiul uman să nu perceapă aprinderea și stingerea succesivă a fiecărei cifre de pe SSD (16 ms <=> 60 Hz).

Intrările sunt 4 semnale de câte 4 biți (cifrele de afișat) și semnalul de ceas al plăcii; ieșirile sunt reprezentate de semnalele de anod (an) și semnalele de catod (cat), toate active pe 0.

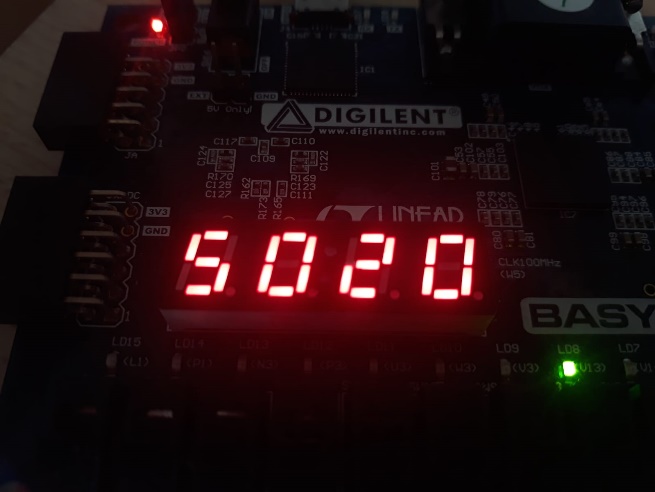


**1110- Se activeaza primul anod si se pune valoarea lui Digit0 pe acesta**

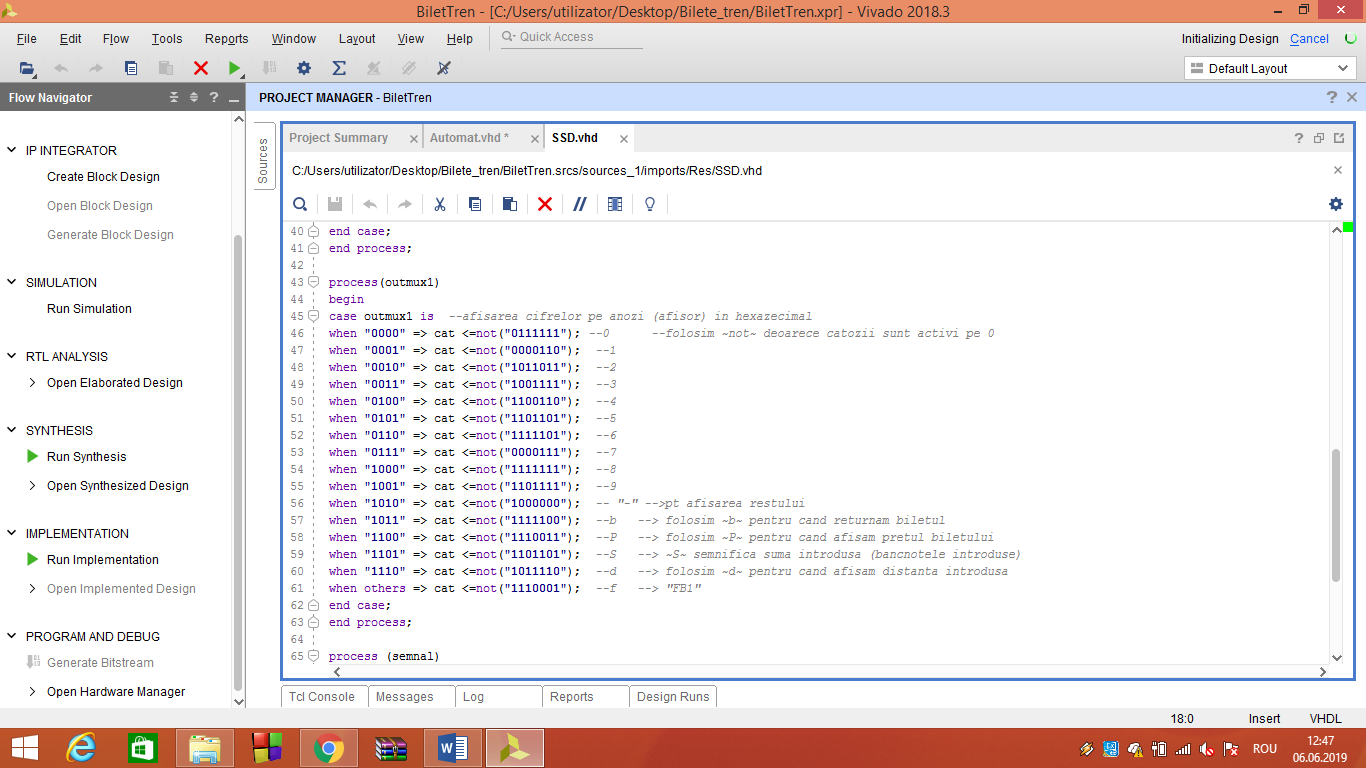
**1101- Se activeaza al doilea anod si se pune valoarea lui Digit1**

**1011- Al treilea anod e activat si se pune valoarea lui Digit2 pe el**

**0111- Se activeaza al patrulea anod si se pune valoarea lui Digit3**

 (semnificatie: S=20)

In afisorul din proiect, am pus ca pentru anumite “digit-uri” ( iesirea din primul MUX ), sa imi afiseze anumite caractere pentru a se intelege mai clar atunci cand se afiseaza distanta, pretul, restul etc.

De exemplu, pentru codul “1011” se va afisa pe primul afisor (cel din stanga) litera “d” (de la distanta) , iar pentru “1011, litera “S”, reprezentand suma pe care o introduce utilizatorul.

* **Distanta introdusa de utilizator prin switch-uri**

Switch-urile care se folosesc pentru introducerea distantei sunt SW(1), SW(0). Distanta maxima e de 990.

* **Calcularea pretului biletului**

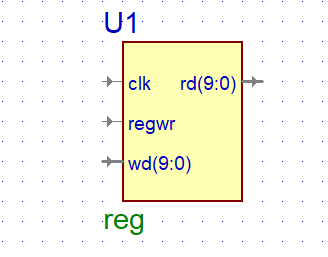
=> Am luat , astfel daca, DE EXEMPLU, distanta e de 530 km, pretul unui bilet ar iesi 53.

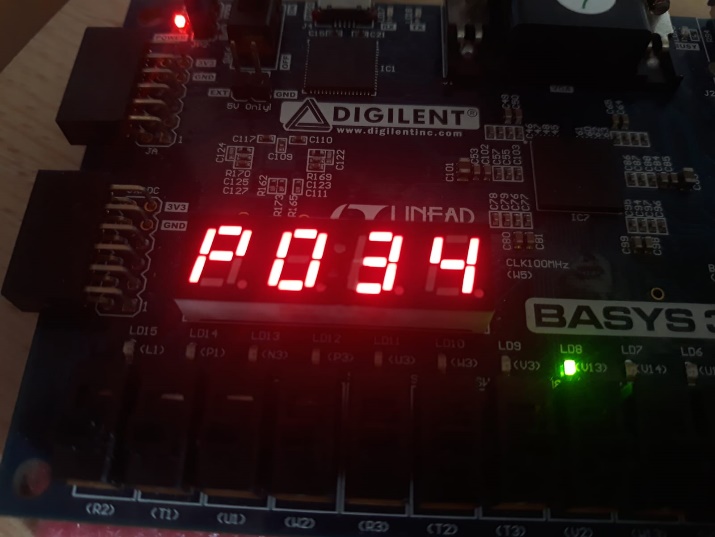
10 km=1 euro

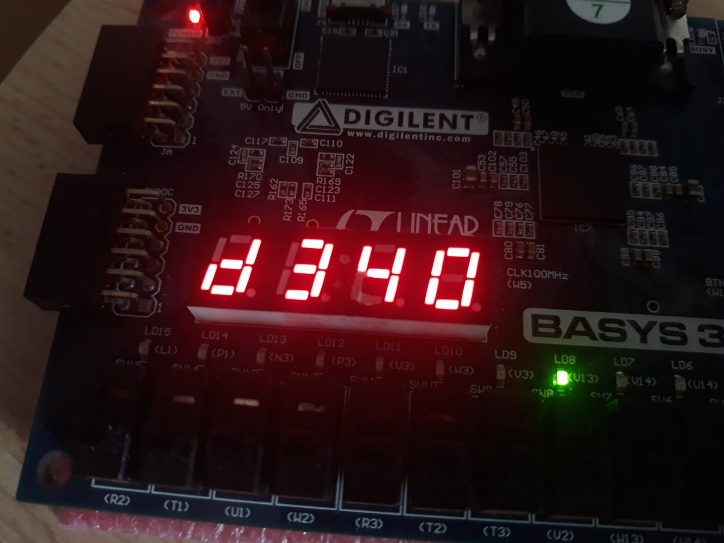
=> Asadar, pretul biletului poate fi luat ca distanta/10, sau primele 2 cifre din distanta-- dist(11 downto 4).

* REGISTRU

Acesta are rolul de a-mi pastra anumite semnale cu scopul de a nu se pierde datele.







* **Suma introdusa de catre utilizator**
* Bancnotele pentru platirea biletului sunt introduse de catre utilizator cu ajutorul switch-urilor astfel:

SW(0)- reprezinta bancnota de 1 euro

SW(1)- reprezinta bancnota de 2 euro

SW(2)- reprezinta bancnota de 5 euro

SW(3)- reprezinta bancnota de 10 euro

SW(4)- reprezinta bancnota de 20 euro

SW(5)- reprezinta bancnota de 50 euro

* In variabilele nr1, nr2, nr5, nr10, nr20, nr50 se retin numarul bancnotelor de fiecare tip introduse de persoane.

**4. Justficarea solutiei alese**

Pentru proiectul de fata, am ales un cod simplu, cu comentarii pentru a fi inteles de alte persoane. Am separat codul in mai multe componente (calcularea distantei, pretului, restului in fisiere separate). Am folosit mai multe butoane, fiecare cu o functionalitate specifica, clara,.

# 5. Instructiuni de utilizare si întretinere

-dupa punerea proiectului in Vivado si generarea programului (\*.bit), se porneste placa

-se introduce distanta de catre utilizator cu ajutorul ultimelor 2 switch-uri din partea dreapta astfel: se pune pe valoare ‘1’ switch-ul corespondent zecilor/sutelor (doar unul dintre ele) si se apasa pe **butonul 1** pentru incrementare

-dupa ce avem distanta dorita, apasam pe **butonul 0** pentru a trece in urmatoarea stare unde ne va afisa costul biletului

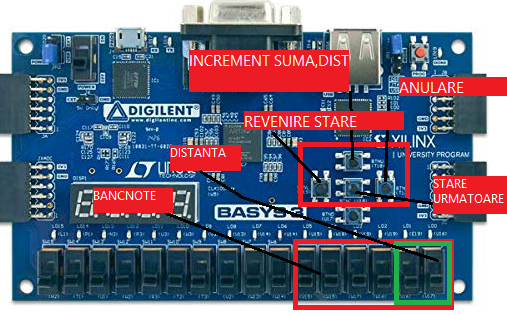
-se apasa iar pe butonul 0 pentru a putea plati biletul: la fel ca si la distanta, se pune switch-ul pentru bancnota pe care o vrem sa o introducem pe valoarea “1” si se apasa butonul 1 pentru incrementare (sw0-bancnota1, sw1-bancnota2, …, sw5-bancnota50)

-daca s-a terminat de introdus banii, se apasa butonul 1 pentru a ne da restul si biletul

-se poate renunta oricand la operatie, apasand **butonul 3** (din dreapta) ducandu-ne la golirea sumei introduce, apoi, apasand iar butonul 0, putem reveni in starea 0, de unde putem introduce din nou distanta.

-**butonul 2**, apasat in orice moment, ne duce tot in starea 0, insa diferenta e ca e pastrata valoarea distantei pentru a o putea modifica

-prin aprinderea ledurilor suntem anuntati ca nu am introdus bancnote suficiente sau ca nu mai sunt bilete



**6. Posibilitati de dezvoltare ulterioara**

Proiectul dat poate fi dezvoltat prin adăugarea de noi funcționalități ca: folosirea a mai putinor butoane,pentru o utilizare mai accesibila.