**FACULTATEA: Automatica si Calculatoare**

**SPECIALIZARE: Calculatoare si tehnologia informatiei**

**DISCIPLINA: DIGITAL SYSTEM DESIGN**

**PROIECT: CHICKEN INVADERS**

**Student:**

Pirvulescu Carmen-Gabriela

Risteiu Ioana-Stefania

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**Prof. indrumator:**

Ileana Blaj

**2023**

[1.Specificatii 3](#_Toc134554093)

[2.Design .........................................................................................................4](#_Toc134554094)

[2.1 Black box 5](#_Toc134554095)

[2.2 Diagrama detaliata 5](#_Toc134554096)

[3.Structura si functionalitate 5](#_Toc134554097)

[3.1 Resurse 5](#_Toc134554098)

[3.2 Flowchart/fsm 9](#_Toc134554099)

[4.Utilitate si rezultate 10](#_Toc134554100)

[5.Justificare tehnica pentru design si implementare ulterioara 12](#_Toc134554101)

# Specificatii

**TEMA PROIECTULUI:**

Să se proiecteze jocul CHICKEN INVADERS pe o placă de dezvoltare FPGA, astfel încât să se păstreze ideea principală a jocului, folosind elementele esențiale ale acestuia, într-o dezvoltare cât mai apropiată de forma inițială.

Proiectul trebuie sa cuprindă:

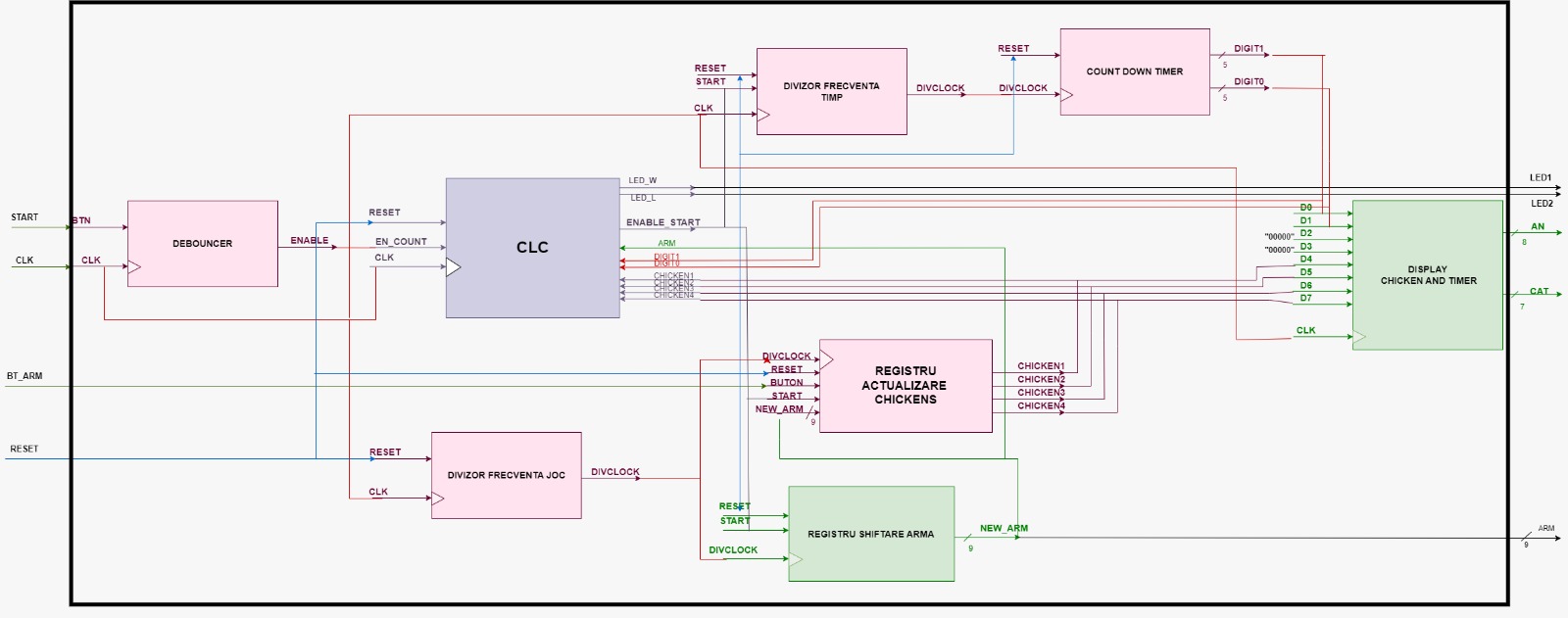
* Un timer;
* Reprezentarea caracterelor pe afișorul SSD;
* Reprezentarea armei pe LED-uri;
* Butonul de start trebuie să fie folosit cu scopul de a porni jocul;
* Butonul de reset trebuie să reseteze întreg jocul;
* Butonul de armă trebuie sa fie folosit pentru împușcarea caracterelor;
* Modalități de anunțare a pierderii sau a câștigului.

# Design

## Black box



## Diagrama detaliată



# 3.Structura si functionalitate

## Resurse

* Schema pentru fiecare resursa alaturi de o scurta descriere:

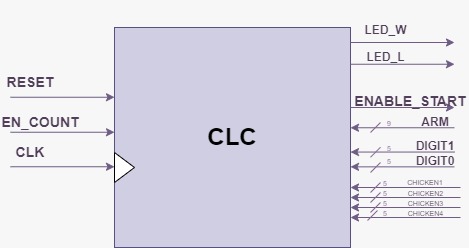
1. **CLC**

În primul rând, componenta ia semnalul de apasare a butonului de start joc, ce intra pe input ca **EN\_COUNT**, si il transforma intr-un enable pentru procesele viitoare, **ENABLE\_START**. **CLK** este intrare și reprezintă semnalul de tact la frecvența de **1Hz.**

In al doilea rand, componenta stabileste starea de WIN si LOSE in functie de starea caracterelor din joc care intra pe input-urile **CHICKEN1**, **CHICKEN2**, **CHICKEN3**, **CHICKEN4**. Daca acestea au fost shiftate in intregime, jucatorul pierde. Totodata, daca numaratoarea inversa ajunge la 00 (**DIGIT1** = 0 si **DIGIT0** = 0 ) jucatorul pierde. In caz contrar, el castiga.

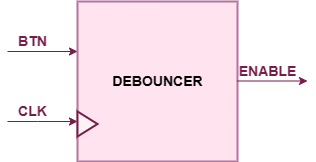
Pentru a semnala starea de WIN sau LOSE, se folosesc doua leduri de culoare verde ( **LED\_L** ) si rosie ( **LED\_W** ).

Componenta poate fi resetata, folosind intrarea **RESET** pe **1**, de pe un switch.



1. **DEBOUNCER**

Pentru a evita anumite erori cauzate de suprapunerea nedorită a unor comenzi sau de apasărea într-o manieră repetată a aceluiași buton, se folosește un **debouncer** pentru butonul de **START**(pentru începerea jocului). **CLK** este intrare și reprezintă semnalul de tact la frecvența de **1Hz**, **BTN** primește semnalul butonului **START** conectat de pe placăși **ENABLE** reprezintă ieșirea butonului care va fi folosită în procesele proiectului.

****

1. **REGISTRU SHIFTARE ARMA**

Primul registru este folosit pentru procesul de shiftare a celor 9 LED-uri, care reprezinta arma. Registrul primeste semnalul de enable dupa apasarea butonului **start**, care intra pe input-ul **START**. Deplasarea LED-urilor se face in functie de un clock divizat, care intra pe input-ul **DIVCLOCK** si care este un semnal de tact la frecventa de 1 HZ. Shiftarea se realizeaza de la stanga la dreapta si de la dreapta la stanga, in functie de directia decisa de intrarea **SEL**( SEL=’0’ pentru shift left, SEL=’1’ pentru shift right). Registrul poate fi resetat folosind intrarea **RESET** pe **1**, astfel ca deplasarea LED-urilor va incepe de la dreapta la stanga.



1. **REGISTRU ACTUALIZARE CHICKENS**

Al doilea registru este folosit pentru procesul de actualizare a caracterelor jocului, adica a celor 3 catozi dispusi orizontal pe fiecare dintre cei 4 anozi, in functie de mai multe input-uri. **START** de pe intrare reprezinta enable-ul registrului, adica apasarea pe buton pentru inceperea jocului. Intrarea **BUTON** activa pe **1** constituie intentia de impuscare a caracterelor, ceea ce implica shiftarea lor cu 0, in sus cu o pozitie. In functie de pozitia LED-ului activ ce reprezinta **arma** ( **NEW\_ARM** ), se alege pe care dintre anozi sa se produca shiftarea ( **CHICKEN1**, **CHICKEN2**, **CHICKEN3** sau **CHICKEN4** ).

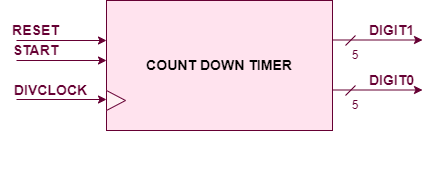
Actualizarea caracterelor se face in functie de un clock divizat, care intra pe input-ul **DIVCLOCK** si care este un semnal de tact la frecventa de 1 HZ. Registrul poate fi resetat folosind intrarea **RESET** pe **1**, care va aprinde catozii dispusi orizontal de pe cei 4 anozi.



1. **COUNT DOWN TIMER**

Componenta numara de la 40 la 0 si transmite valoarea codificata catre un afisor SSD. Intrarea **START** constituie enable-ul counter-ului, ea reprezentand de fapt apasarea butonului de start.

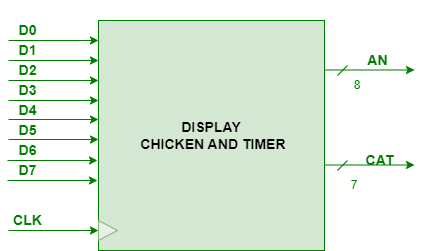
Numaratoarea se face in functie de un clock divizat, care intra pe input-ul **DIVCLOCK** si care este un semnal de tact la frecventa de 1 HZ. Counter-ul poate fi resetat, folosind intrarea **RESET**, activa pe 1. Pentru Output-uri ( **DIGIT1**, **DIGIT0** ) se folosesc 2 vectori de 5 biti, dintre care primul bit este o codificare pentru afisorul SSD, iar ceilalti 4 biti sunt folositi pentru reprezentarea din binar a cifrei.



1. **DISPLAY CHICKEN AND TIMER**

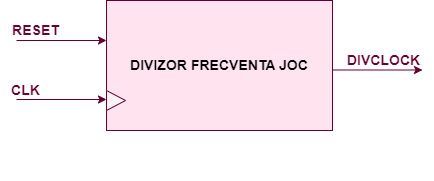
Componenta reprezinta un 7-Segment Display care se foloseste de toti cei 8 anozi si 7 catozi pentru a afisa de la stanga la dreapta valorile primite ca input-uri. Pe primii 4 anozi ( **D4,D5,D6,D7** ) sunt afisate pe cate 3 catozi dispusi orizontal, caracterele jocului, anume gainile. Urmatorii 2 anozi primesc o valoare nula pe intrare ( **D2, D3** ). Ultimii 2 anozi ( **D0, D1** ), primind valorile de la counter-ul anterior, afiseaza numaratoarea inversa. Componenta mai primeste ca intrare **CLK**, care reprezinta semnalul de tact la frecventa de 1 HZ.

**AN** si **CAT** sunt vectori de 8, respectiv 7 biti, corespunzatori numarului de anozi folositi pentru afisare si numarului de catozi ai fiecarui anod.



1. **DIVIZOR FRECVENTA JOC**

Primul divizor este folosit pentru procesele interne de shiftare a armei si de actualizare a caracterelor jocului, care reprezinta gainile, emițând un semnal de tact pentru fiecare 9.000.000 de semnale emise de clock-ul sistemului. Ca intrare va fi **CLK** care reprezintă semnalul de tact la frecvența de **100Mhz** al plăcii. **DIVCLOCK** este ieșirea componentei, adică semnalul de tact divizat la frecvența de **1Hz.** Divizorul poate fi resetat, folosind intrarea **RESET**  pe **1.**

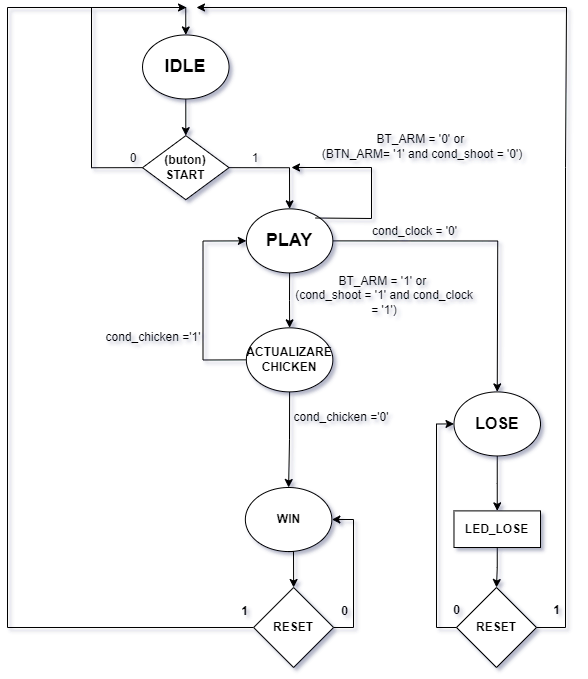


1. **DIVIZOR FRECVENTA TIMER**

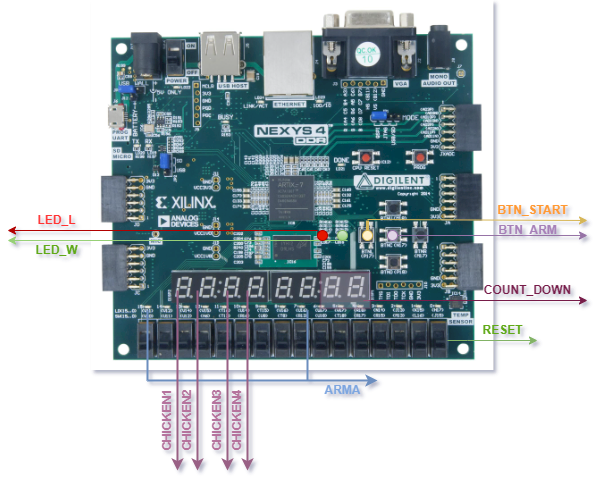
Al doilea divizor este folosit pentru procesul de numaratoare inversă ( 40-0), emițând un semnal de tact pentru fiecare 9.999.999 de semnale emise de clock-ul sistemului. Ca intrare va fi **CLK** si **START** inceperea functionarii divizorului de frecventa. **DIVCLOCK** este ieșirea componentei, adică semnalul de tact divizat la frecvența de **1Hz.** Divizorul poate fi resetat, folosind intrarea **RESET**  pe **1.**



## 3.2 Flowchart/fsm ( finite state machine)



# 4. Utilitate si rezultate



Jocul funcționează după principii simple. Jucătorul trebuie să “împuște” toate caracterele

de pe afisor înainte că timpul să se scurgă. Dacă reușește, el câștigă. În cazul în care numărătoarea ajunge la 0 și mai sunt caractere rămase, el pierde.

La început, jocul este într-o stare inițială **IDLE**. După apăsarea butonului de START, se

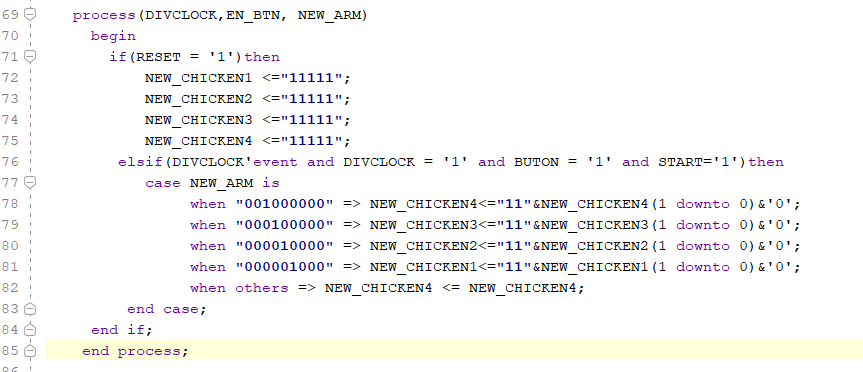
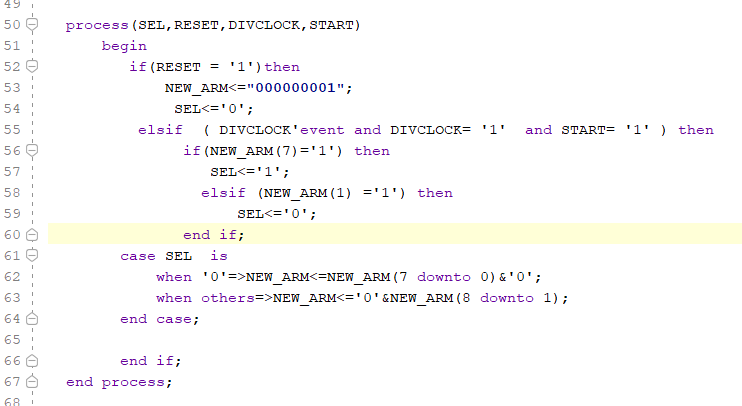
trece în starea următoare de **PLAY**.

Starea de **PLAY** constă în începerea jocului. Începe numărătoarea inversă de la 40 la 0, se

afișează caracterele jocului pe afisorul SSD și începe deplasarea LED-ului aprins de la dreapta la stânga. Această reprezintă desfășurarea normală a jocului. Apăsarea butonului de armă și numărătoarea inversă influențează trecerea în altă stare sau rămânerea în aceeași stare.

Jocul poate evolua în mai multe feluri. Dacă jucătorul nu apasă pe butonul de armă, se

revine în starea de PLAY. De asemenea, dacă el apasă pe butonul de armă, dar armă nu este în dreptul caracterelor, se revine în starea de PLAY. În situația în care jucătorul apasă pe butonul de armă și reușește să țintească un caracter al jocului, se trece în starea de ACTUALIZARE CHICKEN.



Starea de **ACTUALIZARE CHICKEN** presupune „împușcarea” unui caracter, adică

catodul specific acelui caracter se stinge. Această operație îl duce pe jucător mai aproape de câștig. Pe acest principiu, „dispar” pe rând toate caracterele jocului la apăsarea butonului de armă, dacă acesta este orientat către ele. Dacă jucătorul reușește să „împuște” toate „găinile” din joc înainte că numărătoarea inversă să se termine, el câștigă, iar jocul trece în starea de WIN.

Starea de **WIN** presupune aprinderea unui led verde pe placă de dezvoltare, care

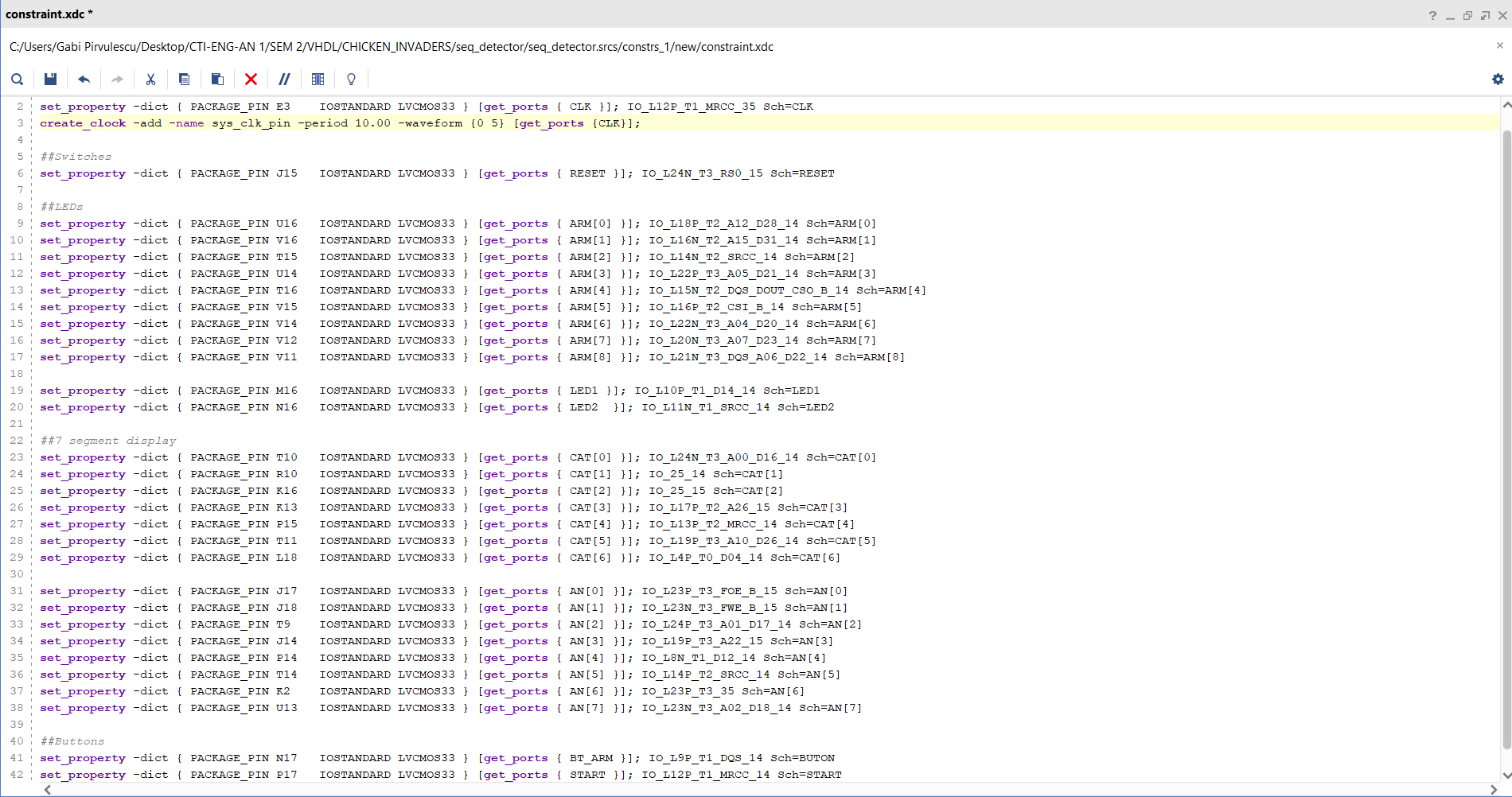
semnalează câștigul. Prin setarea switch-ului de **RESET** pe 1 se revine în starea inițială IDLE.

În cazul în care numărătoarea inversă se termină înainte că jucătorul să fi reușit să

„împuște” toate caracterele înseamnă el a pierdut, iar jocul trece în starea de LOSE.

Starea de **LOSE** presupune aprinderea unui led roșu pe placă de dezvoltare, care semnalează pierderea. Prin setarea switch-ului de RESET pe 1 se revine în starea inițială IDLE.

Pentru a integra proiectul pe placă este nevoie de un program de sinteză și mapare a semnalelor folosite în cod, iar noi am decis să folosim Vivado 2018.3. În Default Part se va alege family: Artix 7, package: csg324 și speed: -1. Este nevoie de un fișier de constrângeri a semnalelor din entitate.



# 5. Justificare tehnica pentru design si implementare ulterioara

Inspiratia pentru acest proiect a fost generată de un alt joc, iar abordarea noastră a fost una minimalista în realizarea sa. Am căutat să aducem caracteristicile esențiale și să eliminăm orice elemente suplimentare, astfel încât să ne concentrăm pe esența și experiența de bază a jocului.

Pentru a face jocul ușor de înțeles și de jucat, am făcut alegeri strategice în ceea ce privește reprezentarea armelor și a personajelor. Am căutat să utilizăm o abordare simplă și intuitivă, astfel încât jucătorii să poată înțelege rapid și să se familiarizeze cu mecanicile de joc.

Ca implementare ulterioara, afisorul pe SSD poate fi inlocuit de un display pentru mai mult entertainment. Acest pas ar duce jocul la un alt nivel. De asemenea, arma care tinteste catre caracterele jocului ar putea fi controlata de pe taste sau „impuscarea” s-ar putea realiza de pe un mouse sau alt dispozitiv. S-ar putea implementa cresterea dificultatii pe niveluri, un scor total si afisarea jocului.