# Structura sistemelor de calcul

## Memory game - numbers in binary

Studenți: Manoilescu Victor
Pricop Petru Adelin
Grupa: 30238 .
Îndrumător proiect: .
Lisman Dragos Florin
Data: 18.12.2023

# Cuprins

# 1 Rezumat:

În această lucrare se studiază functionalitatea unui joc de memorie contracronometru cu numere in binar folosindu-ne de un Display PMOD OLED cu 128 x 32 pixeli. Din perspectiva utilizatorului, jocul poate fi impartit in 4 stari principale, acestea avand propriul lor comportament in program:

- Start
- Display
- Wait for answer + Check answer
- Increment score
- Game over

# 2 Introducere:

## 2.1 START:

Prima stare cu care utilizatorul va interactiona este starea START. Aceasta stare este accesibila doar prin apasarea butonului de reset, si se poate iesi din aceasta doar prin apasarea butonului de start. In aceasta stare nu au loc procese, aceasta avand ca scop principal asteptarea inceperii jocului.

## 2.2 DISPLAY:

Una dintre principalele stari ce apar recurent in pe parcursul jocului este starea DISPLAY. Aceasta este accesata prima data prin apasarea butonului start si este re-accesata de fiecare data cand utilizatorul introduce un raspuns corect. In aceasta stare principalele procese ce iau loc sunt stabilirea numarului aleator prin generarea unei adrese cu ajutorul generatorului de numere pseudoaleatoare [1] si afisarea acestuia cu ajutorul afisorului pe OLED [2]. Iesirea din aceasta stare se face automat dupa trecerea unei durate de timp masurata de un timer [3].

## 2.3 WAIT FOR ANSWER + CHECK ANSWER:

In starea WAIT FOR ANSWER se va astepta raspunsul utilizatorului urmat de apasarea butonului check, in functie de corectitudinea raspunsului mergandu-se ori in starea INCREMENT SCORE ori in GAME OVER. In aceasta stare se afiseaza timpul ramas , ce descreste constant pana ajunge la 0, si scorul current pe cate 2 cifre ale afisorului cu 7 segmente, iar la apasarea butonului check sau la terminarea timerului [3], inputul utilizatorului va fi comparat cu numarul aleator stabilit in starea anterioara. In caz de egalitate se merge in INCREMENT SCORE iar in caz contrar se merge in GAME OVER.

### 2.4 INCREMENT SCORE:

Starea INCREMENT SCORE este o stare intermediara in care se vor face modificari asupra scorului si timpului initial de raspuns. Scorul va fi incrementat in functie de cat de repede a fost dat raspunsul, astfel incrementul fiind direct proportional cu timpul ramas, iar timpul initial de raspuns este decrementat cu o secunda, pentru a creste dificultatea odata cu cresterea nivelului. Iesirea din aceasta stare se face automat, mergandu-se inapoi in starea DISPLAY.

### 2.5 GAME OVER:

Finalul jocului este marcat de starea GAME OVER. In aceasta stare se va afisa scorul complet pe toate cele 4 cifre ale afisorului cu 7 segmente. Iesirea din aceasta stare este posibila doar prin apasarea butonului de reset.

## 3 Fundamentare teoretica:

In aceasta sectiune vor fi descries mai in detaliu componentele utilizate in crearea jocului.

### 3.1 Generatorul de numere pseudoaleatoare:

Prima componenta discutata va fi generatorul de numere pseudoaleatoare. Acesta a fost create prin adunarea succesiva a unui numar arbitrar cu 7 si selectarea bitilor din interiorul acestui numar.

### 3.2 Afisorul pe Display PMOD OLED 128 x 32 pixeli:

Afisorul a fost facut cu ajutorul manualului de referinta PMOD OLED. Acest afisor primeste ca intrare o adresa, si in functie de aceasta schimba ecranul in ecranul corespunzator adresei sau il pastreaza pe cel current, daca adresa primita este deja cea a ecranului current. Prin activarea semnalului de reset pe Display va fi afisat un ecran gol.

### 3.3 Timer:

Timer-ul scade o valoare initiala, calculata pe baza unui nivel dat, in conditiile in care are semnalul enable active, la fiecare secunda, oferind ca output numarul de secunde ramase din valaorea initiala si un flag ce devine activ odata ce valoarea ajunge la 0. Prin activarea semnalului de reset valoarea timerului ajunge la valoarea sa initiala. Numararea secundelor a fost facuta cu ajutorul unui clock divizat, cu fregventa de jumatate de secunda [3.1]

### 3.3.1 Clock divizat:

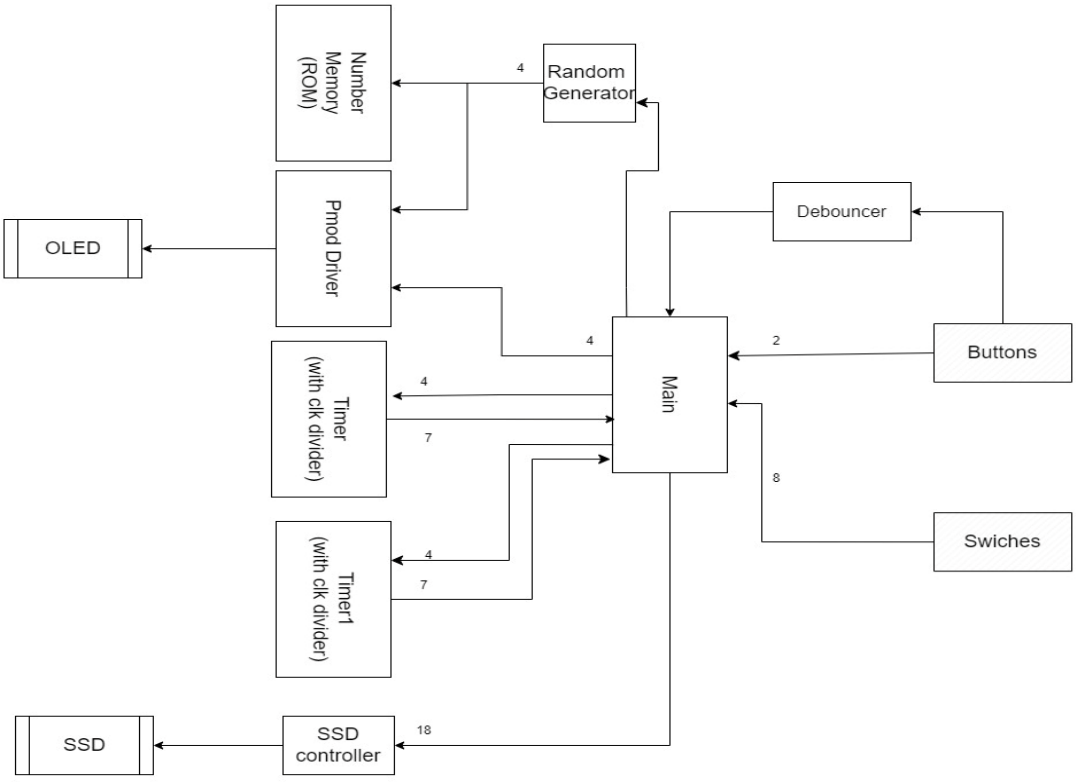
Clock-ul divizat foloseste un counter ce se incrementeaza la fiecare rising-edge al clockului intern. Odata ce valoarea counterului ajunge la 50000000, valoarea de output devine valoarea anterioara de output negata, iar counterul este adus inapoi la 0. Prin activarea semnalului de reset valoarea clock-ului divizat si a counterului devin 0.

# 4 Proiectare si implementare

4.1 Metoda experimentala utilizata este implementarea hardware alcatuita dintr-o placuta Basys 3 si un Pmod OLED 128*32 Pixel Monochromatic OLED Display pe care sunt afisate date despre joc.

4.2 A fost aleasa aceasta solutie deoarece fiind vorba despre un joc, este mai usor de observat si de testat rezultatele FSM-ului.

## 4.3 Schema bloc




8

## 4.4 Module:

- ***Main*** avem legate intre ele toate modulele hardware necesare realizarii proiectului si automatul cu stari finite care trece jocul dintr-o stare in alta;
- ***Timer*** reprezinta un counter cu semnal de Reset si Enable care permit oprirea sau resetarea numaratorului, incrementarea facandu-se la aproximativ o secunda cu ajutorul modulului ***Clk_Div*** care este un divizor de freventa simplu;
- ***ROM*** este modulul hardware in care sunt stocate numerele care urmeaza sa fie afisate in joc;
- ***Debouncer*** realizeaza actiunea de debounce, stabilizarea semnalelor emise la apasarea unui buton si este folosit pentru butonul de check;
- ***Displ7seg*** este modulul hardware care ne ajuta sa afisam datele dorite pe afisorul cu 4 pozitii a cate 7 segmente;
- ***RandomGenerator*** este un mmodul care genereaza numere pseudo-random in functie de momentul la care se face citirea;
- ***Pmod_driver*** este un modul mai complex alcatuit din alte doua module: ***OLEDInit*** care initializeaza display-ul cu ajutormul modulelor ***SPI_COMP*** si ***DELAY_COMP*** si din modulului ***Disp***(in care mai avem un FSM care ajuta la afisarea informatiilor dorite pe pmod OLED) format la randul lui din modulele ***SPI_COMP, DELAY_COMP*** si ***CHAR_LIB***(in care sunt mapate caractererele ascii pentru a fi afisate pe display).

## 4.5 Manual de utilizare:

1. Conectam placuta la o sursa de alimentare;
2. Apasam butonul RESET(butonul de jos → );
3. Cand suntem pregatiti, apasam butonul START(butonul de sus → );
4. Citim si retinem numarul si asteptam afisarea mesajului „Waiting for your answer”
5. Introducem pe ultimele 8 switch-uri(de la stanga la dreapta) numarul citit anterior in binar(cel mai semnificativ bit fiind corespunzator celui mai din stanga switch → );
6. Pe display-ul cu 7 segmente putem observa pe cele doua cifre din stanga nivelul curent iar, pe cele doua cifre din dreapta timpul ramas in secunde;
7. Dupa ce am introdus numarul, apasam butonul CHECK(butonul din mijloc → ) pentru a verifica daca numarul introdus este corect;
8. Daca pe display apare un alt numar inseamna ca numarul intreodus este corect si am trecut la nivelul următor unde avem cu o secunda mai puțin timp la dispoziție sa introducem numărul si revenim la pasul 4;
9. In cazul in care pe display apare mesajul „GAME OVER” înseamnă ca numărul introdus a fost greșit sau timpul s-a scurs si combinația de switch-uri la acel moment era greșită iar pe afișorul cu 7 segmente (pe toate cele 4 cifre);

# 5 Rezultate experimentale

- Instrumente de proiectare utilizate:
  - Limbaj: VHDL;
  - Mediu software: Vivado 2016.4

Experimental, soluția implementata a fost testata fizic, fără ajutorul unor software-uri, unele dintre module(e.g. modulul timer, modulul clk_div si modulul rom) fiind simulate individual in mediul Vivado înainte de a fi integrate in proiectul final.

Totodată, au fost testate toate funcționalitățile proiectului :

- Funcția de start;
- Funcția de reset;
- Funcția de check;
- Funcția de auto-check la terminarea timpului alocat;
- Funcția de afișare a numarului;
- Functia de afisare a mesajului de asteptare;
- Functia de afisare a mesajului de terminare a jocului;
- Functia de afisare a nivelului si a timpului ramas;
- Functia de afisare a scorului obtinut la finalul jocului.

# 6 Concluzii

Concluzionând, mai sus am prezentat un joc simplu dar captivant si rar intalnit numit „Memory game - numbers in binary” implementat cu ajutotul unei placi FPGA Basys 3 si al unui pmod Oled 128*32 monochromatic in limbajul VHDL si in mediul de dezvoltare Vivado 2016.4. Jocul consta in memorarea si transpunerea contra-cronometru a unor numere afisate pe oled in binar cu ajutorul switchurilor si al butoanelor de start, check si reset.

Din acest proiect, cel mai semnificativ aspect invatat este reprezentat de utilizarea dispozitivului pmod oled si protocolul de transmitere de date catre acesta. Totodata, un alt aspect invatat este gestionarea unui proiect cu un numar mai ridicat de module si organizarea lor intr-o forma cat mai optima.

Proiectul poate fi utilizat in scop recreativ dar si didactic, fiind un joc pe cat de simplu pe atat de captivant din care putem sa ne exersam abilitatile de memorare si de transformare a numerelor din baza 10 in baza 2.

O posibila devoltare viitoare a proiectului ar putea fi aducerea lui la o scara mai larga de numere prin posibilitatea de a introduce numere pe 16 biti. O alta dezvoltare viitoare poate fi afisarea numerelor atat pe oled cat si pe un monitor iar introducerea numarului sa se faca comutand switchuri pe acel ecran cu ajutorul mousului.

# 7 Bibliografie

[1] Basys 3™ FPGA Board Reference Manual, 2016, https://digilent.com/reference/_media/basys3:basys3_rm.pdf.

[2] PmodOLED™ Reference Manual, 2016, https://digilent.com/reference/_media/reference/pmod/pmodoled/pmodoled_rm.pdf.