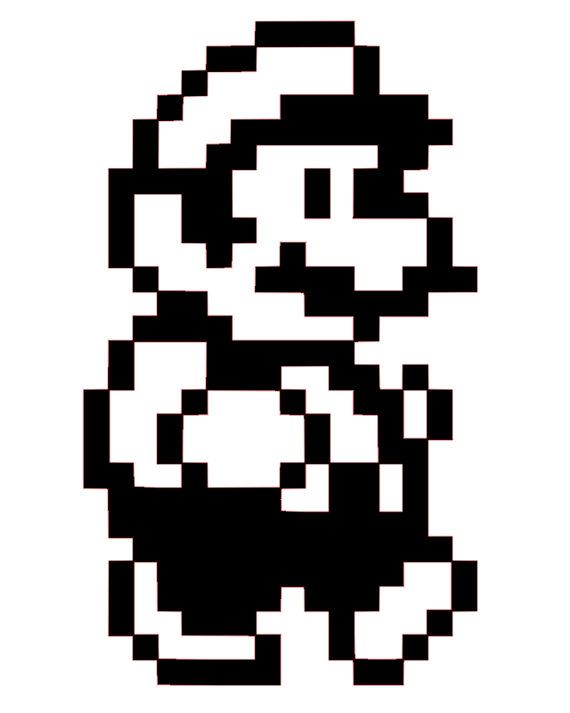
|  |  |  |
| --- | --- | --- |
| A picture containing clipart  Description automatically generated  Universitatea Tehnica din Cluj-Napoca | Student: **Ossian Adrian Bogdan**  Grupa: 30233  Profesor indrumator: **Fati Daniela**  Data : 4 ian. 2020 |  |
| Catedra: Dr. **Baruch Zoltan Francisc** |

Mario Animation

Using Digilent’s PmodOLED



Contents

[1. Rezumat 3](#_Toc29075567)

[2. Introducere 4](#_Toc29075568)

[2.1 Descrierea contextului temei si a tendintelor tehnologice legate de aceasta tema 4](#_Toc29075569)

[2.2 Definirea domeniului de studiu, explicarea tehnologiei de bază și explicarea importanței acestui domeniu 5](#_Toc29075570)

[2.3 Obiectivele principale ale proiectului 6](#_Toc29075571)

[2.4 Soluția propusă 6](#_Toc29075572)

[3. Fundamentare teoretică 8](#_Toc29075573)

[3.1 Interfața SPI 8](#_Toc29075574)

[3.2 VHDL și mediul de proiectare Vivado 9](#_Toc29075575)

[3.3 Resursele folosite 9](#_Toc29075576)

[4. Proiectare și implementare 10](#_Toc29075577)

[4.1 Descrierea componentelor principale 10](#_Toc29075578)

[4.2 Descrierea librăriei MarioLib și a modului de contruire a caracterelor prezentate 12](#_Toc29075579)

[4.3 Schema generată 14](#_Toc29075580)

[4.4 Manual de utilizare 15](#_Toc29075581)

[5. Rezultate experimentale 16](#_Toc29075582)

[5.1 Descrierea animației obținute 16](#_Toc29075583)

[5.2 Detalii tehnice despre implementare 18](#_Toc29075584)

[6. Concluzii 19](#_Toc29075585)

[7. Bibliografie 20](#_Toc29075586)

# Rezumat

Tema aleasa a fost selectata cu scopul de a invata cum functioneaza protocolul de comunicatie SPI si cu scopul de a reproduce o secventa din jocurile retro ale copilariei. Obiectivele principale ale acestui proiect au fost implementarea unui protocol de comunicatie SPI si a unei librarii cu caractere pentru a face posibila afisarea unei animatii pe display-ul monocromatic cu LED-uri organice denumit Pmod OLED, produs de compania Digilent. Metoda de rezolvare a implicat utilizarea resurselor oferite de compania Digilent incepand de la obiectele fizice precum display-ul si placa FPGA, pana la resursele software precum editorul Vivado. In plus, limbajul ales pentru realizarea acestei descrieri hardware a fost VHDL. Acest proiect are ca si scop principal de utilizare demonstratia didactica, adica demonstrarea capacitatilor acestui display de ultima generatie. Rezultatele obtinute au fost pe masura asteptarilor si am ramas placut surprins de simplitatea acestui proiect dupa ce m-am informat corespunzator despre toate resursele necesare. Principalele concluzii la care am ajuns sunt: protocolul de comunicatie SPI isi are locul in multe aplicatii in dispozitivele hardware unde alte protocoale de comunicatie esueaza[;](https://www.dictionarroman.ro/?c=esueaza) interfata Pmod a placutei FPGA Nexys 4 ofera posibilitatea de a conecta o multitudine de periferice; realizarea animatiilor este o procedura ce necesita multa rabdare si desigur, talent artistic.

# Introducere

## 2.1 Descrierea contextului temei si a tendintelor tehnologice legate de aceasta tema

Contextul în care se plasează această temă este a display-urilor de ultima generație care folosesc tehnologia OLED. **Display OLED** – unde fiecare “pixel” al ecranului se poate aprinde și stinge individual, în diverse culori, în funcție de imaginea pe care trebuie să o afișeze. Asta înseamnă că, spre deosebire de display-urile numite uzual LED și LCD, fiecare pixel al ecranului produce lumină și se și poate stinge complet, generând astfel un negru absolut în acea zonă.

Din acest motiv și din alte detalii constructive ale ecranului favorizate de această tehnologie, display-urile OLED au în general culori mai vii, mai plăcute, și un negru mult mai intens ce duce la un contrast total mai mare. În plus, este eliminată problema *bleedingului* de lumină, iar ecranul poate fi mult mai subțire deoarece dispar straturile de cristale lichide prezente pe celelalte tehnologii. În imaginea de mai jos este evidențiată diferența masivă de culori între cele două tehnologii.

A picture containing building, factory, indoor

Description automatically generated

Fig 1. LED vs OLED

## Definirea domeniului de studiu, explicarea tehnologiei de bază și explicarea importanței acestui domeniu

Un OLED (prescurtare din engleza de la *Organic Light-Emitting Diode*) este o componentă electronică în forma de folie foarte subțire si luminoasă făcută dintr-un material semiconductor organic. Comparând cu tehnologia LED, tehnologia OLED este ieftină pentru că nu are nevoie de un monocristal scump. La început această tehnologie nu permitea display-urilor să ajunga la luminozitatea standard a display-urilor LED, dar in ultimii ani s-au facut progrese majore. Spre deosebire de un display LED care are nevoie de iluminare suplimentara a fundalului (backlight), un display OLED ofera control la nivel de fiecare pixel si astfel se pot obtine cele mai profunde culori de negru (deoarece pixelul este stins), si desigur cel mai inalt nivel de contrast.



Fig 2. OLED vs LED assembly

Există două tipuri de tehnologii OLED: matrice pasivă (**PMOLED**) și matricea activă (**AMOLED**). Matricea activă necesită componente electronice pentru a comuta și deconecta fiecare pixel individual. Această tehnologie este mult mai bună pentru afișarea negrului profund și de aceea este folosită în televizoarele OLED pe care le putem cumpăra astăzi.

Un panou OLED este în mod normal compus din patru straturi primare: Substratul, care acționează ca structură de bază; anodul, care atrage electroni; catodul, care oferă electroni; și stratul organic dintre. Stratul organic este împărțit în continuare într-un strat conductiv - care oferă "găurile de electroni" cu ajutorul cărora, electronii care trec prin strat se pot aprinde, producând energie în proces; - și un strat emisiv în care se produce de fapt lumina. De aici și până la producerea unei culori reale, este doar o chestiune de adăugare a straturilor de plastic roșu, verde și albastru la substrat.

Display-urile OLED au un consum redus de energie, calitate mai bună a imaginii, timp de răspuns mai rapid, unghi perfect de vizualizare (până la 178 de grade) și greutate redusă datorită reducerii numărului de componente.

## Obiectivele principale ale proiectului

Obiectivele principale ale acestui proiect sunt:

* Informarea cu privire la tehnologiile folosite pentru display-uri ;
* Informarea cu privire la protocoalele de comunicație folosite în ziua de azi ;
* Realizarea unui controller SPI pentru comunicarea cu display-ul Pmod OLED al companiei Digilent ;
* Dezvoltarea unui mod de lucru în condiții ostile (precum împrumutarea resurselor fizice timp de doar câteva zile pentru testare și depanare) ;
* Demonstrarea capacităților tehnologiei Oled pentru dispozitive portabile.

## Soluția propusă

Soluția propusă de mine folosește un clock de 32 de ori mai mic decât frecvența standard a plăcuței FPGA; un state-machine cu stările Idle, Send, HoldCS1, HoldCS2, HoldCS3, HoldCS4, Done ; un proces pentru transmiterea unui byte folosind un registru de shiftare. În plus, un alt detaliu important al implementării mele este folosirea unei librării de caractere custom numită MarioLib, ce conține peste 15 caractere folosite în animația realizată de mine.

Spre deosebire de alte implementări ale protocolului de comunicare SPI care folosesc un clock de 10MHz si au un machine state de doar 5 stări, implementarea realizată de mine funcționează la o frecvență de 3MHz oferind siguranță la transmiterea datelor.

În cele ce urmează, secțiunile ce vor fi prezentate ulterior conțin prezentarea fundamentului teoretic (tehnic) al acestui proiect și desigur proiectarea si implementarea propriu zisă urmată de rezultate experimentale și concluziile la care am ajuns.

# Fundamentare teoretică

## Interfața SPI

Această interfață a fost dezvoltată de Motorola, fiind adoptată ulteriori de numeroși producători ca o interfață de comunicație între microcontrolere și diferite circuite periferice din sistemele existente. Abrevierea SPI vine de la Serial Peripheral Interface. Interfața serială periferică (SPI) este o specificație sincronă de interfață de comunicare serială folosită pentru comunicarea pe distanțe scurte, în principal în sistemele încorporate. Aplicațiile tipice includ carduri digitale sigure și afișaje cu cristale lichide.

Dispozitivele SPI comunică în modul full duplex folosind o arhitectură master-slave (terminologia alternativă fiind principală și secundară) cu un singur master. Dispozitivul principal generează cadrul pentru citire și scriere. Mai multe dispozitive slave sunt acceptate prin selecție cu selectare individuală a sclavilor (SS), uneori numite chip select (CS), linii.

Uneori SPI este numit bus serial cu patru fire, în contrast cu autobuzele seriale cu trei, două și unu-fire. SPI poate fi descris cu exactitate ca o interfață serială sincronă, dar este diferit de protocolul SSI (Synchronous Serial Interface Interface), care este, de asemenea, un protocol de comunicare sincronă cu patru fire. Protocolul SSI utilizează semnalizare diferențială și oferă doar un singur canal de comunicare simplex. SPI este o comunicație mutli-slave cu single master.

A close up of a sign

Description automatically generated

Fig 3. Protocol de comunicatie SPI

Pentru mai multe detalii despre istoria și aplicabilitățile acestui protocol de comunicare, accesați referința [[1].](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface)

## VHDL și mediul de proiectare Vivado

VHDL este un limbaj de descriere hardware apărut în anul 1987, denumit de IEEE. Este utilizat în concepția asistată pe calculator a circuitelor integrate sau pentru configurarea FPGA-urilor. Alături de Verilog, este cel mai popular limbaj de descriere hardware și este una din principalele metode de proiectare a circuitelor integrate moderne.

Vivado Design Suite este un produs software realizat de Xilinx pentru sinteza și analiza proiectelor HDL, care înlocuiește Xilinx ISE cu funcții suplimentare pentru sistem pe dezvoltarea de cipuri și sinteză la nivel înalt. Vivado reprezintă o rescriere și re-gândire la nivelul întregului flux de proiectare (în comparație cu ISE) și a fost descris de recenzori drept „bine conceput, bine integrat, aprins rapid, scalabil, întreținător și intuitiv”.

Pentru mai multe detalii despre cele două, articolul atașat este la dispoziția cititorului [[2].](https://en.wikipedia.org/wiki/Xilinx_Vivado)

## Resursele folosite

În realizarea acestui proiect au fost folosite atât resursele de la laborator [[3]](http://users.utcluj.ro/~baruch/ssc/labor/Proiectare-Interfata.pdf), cât și resursele oferite de producătorul Digilent [[4]](https://store.digilentinc.com/pmod-oled-128-x-32-pixel-monochromatic-oled-display/).

Pinii pentru descrierea conectorului se află în figura de mai jos:

A screenshot of a computer

Description automatically generated

Fig 4. Conectorul SPI

# Proiectare și implementare

## Descrierea componentelor principale

Componenta **SpiMarioLib** este componenta SPI folosită în implementare având ca porturi intrările Clk, Rst, SPI\_EN, SPI\_DATA și ca ieșiri CS, SDO, SCLK, SPI\_FIN. Descrierea acestei componente se face folosind 3 procese și un state-machine. În primul proces se realizează un divizor de frecvență ce împarte frecvența la 32 obținând o viteză de operare de 3MHz. Dacă starea curentă este de transmisie, se incrementează numărătorul, altfel se resetează. Procesul de trasnziție a stărilor numit STATE\_MACHINE introduce starea de IDLE dacă intrarea rst are valoarea HIGH. Altfel, când este în starea IDLE va aștepta după semnalul SPI\_EN și va intra în starea SEND. Când se află în starea Send se va începe transmisia unui cuvânt de un byte trecând prin stările HoldCS1,2,3,4. Starea de finalizare este Done, unde dacă SPI\_EN este LOWm se va trece direct în starea IDLE din nou, asteptând alt cuvânt. Procesul final este SPI\_SEND\_BYTE, unde se realizează transmisia propriu zisă. Dacă starea curentă este Idle, numărătorul de shiftare este resetat si registrul de shiftare este initializat cu SPI\_DATA (byte-ul ce trebuie trimis). Dacă starea curentă este Send și ne aflăm pe frontul de cădere al clock-ului, vom trimite cel mai semnificativ bit si vom shifta datele din registrul de shiftare și reținem câți biți am shiftat prin incrementarea shift\_counter. La SLCK high, vom reseta flag-ul pentru falling.

Componenta **MarioLib** este componenta care construiește caracterele. Are ca intrare o adresa, iar ca iesire o valoare. Primii 3 biti sunt folosiți pentru a specifica numărul liniei ce trebuie trimisă din caracterul x, iar urmatorii 8 biți specifică caracterul dorit. Astfel pentru ca un caracter sa fie transmis in totalitate, sunt nevoie de 8 valori din această componentă. Vom explica în detaliu această componentă în cele ce vor urma.

Componenta **Delay** este componenta folosită pentru sincronizarea animației. Are ca porturi de intrare CLK, RST, DELAY\_MS, DELAY\_EN, iar ca port de ieșire un flag numit DELAY\_FIN. Principiul de funționare este simplu. Avem un state machine care iterează prin stările IDLE, HOLD și DONE. În momentul în care se ajunge in starea HOLD, sistemul va rămâne în această stare până când counterul creat în procesul al doilea va ajunge la valoarea dorită. Astfel obținem un delay exprimat în milisecunde.

Componenta **OledMarioAnimation** este componenta folosită pentru realizarea animației propriu zise. Este de fapt un state-machine care este capabil să comunice cu toate componentele definite anterior. Are ca porturi de intrare Clk, Rst, AllChars (buton care este folosit pentru afișarea tuturor caracterelor din librărie), Animation (buton pentru a afișa animația), EN, iar ca ieșiri CS, SDO, SCLK, DC, FIN(flag pentru finish). După cum am spus, partea cea mai importantă a acestei componente este state-machine-ul, care are stări ce interacționează cu constante ale ecranului, precum și stări care ajută la trimiterea caracterului către display, citiri din memorie și multe altele. La linia 130 începe o declarație de constante ce reprezintă imaginile care vor fi afișate pe ecran. La linia 280 începe procesul propriu zis de tranziție, unde sunt testate butoanele menționate mai sus, iar apoi este derulată animația. Starea de UpdateSceen ia toate valorile ascii din ecranul curent pentru a lucra cu ele. Odată accesată, această valoare este folosită de 8 ori în stările SendChar1,2,3,4,5,6,7,8 unde se trimit concatenate valoarea caracterului și coloana dorită din acel caracter. Este nevoie de 2 tranziții unde se dă valoare semnalului spi\_en și se așteaptă după spi\_fin. După fiecare stare de trimitere a caracterului, este necasară o citire din memoria creată, adică ieșirea din MarioLib.

Componenta **OledInit** este componenta pentru inițializarea display-ului OLED. Inițializarea lui este obligatorie înainte fiecărei folosiri. Această componentă este proprietatea firmei Digilent și este disponibilă pe site-ul acestora.

Componenta **PmodOLED** controlează toată activitatea display-ului și tot ceea ce face este să unească resursele fizice ale plăcii FPGA de partea de descriere hardware realizată anterior. În plus, mai are un state machine care trece prin procesul de inițializare a display-ului, iar apoi va trece în starea de afișare a animației.

## Descrierea librăriei MarioLib și a modului de contruire a caracterelor prezentate

Caracterele au fost desenate folosind grupuri de pătrate de dimensiuni 8x8 biți. De exemplu, varianta mare a lui Mario este constituită de fapt din 9 caractere, un grid de 3x3 caractere fiecare caracter având 64 de biți. Aceste caractere au fost desenate de mine personal și au necesitat numeroase ore de muncă și testare. A screenshot of a computer

Description automatically generated

Fig 5. MarioLib

Caracterele se observă mult mai bine dacă sunt prezentate în dimensiuni mai mici precum în următoarea imagine.

A close up of a logo

Description automatically generated

Fig 6. MarioLib part 2

Caracterele sunt trimise către SPI coloană cu coloană, de jos în sus. Astfel sunt nevoie de 8 accese la această librărie pentru trimiterea unui caracter, deci pentru desenarea lui Mario în versiunea mare sunt nevoie de 8 x 3 x 3 accese. MarioLib nu este altceva decât o memorie care are biți stocați în ea.

## Schema generată

A screenshot of a cell phone

Description automatically generated

## Manual de utilizare

Conectați display-ul în portul A pentru perifericele Pmod și programați plăcuța FPGA Nexys 4 DDR. Butonul din centru este folosit pentru resetarea animației, Butonul din stânga este pentru afișarea tuturor caracterelor pe ecran, iar butonul din dreapta este folosit pentru a afișa animația propriu zisă.

A circuit board

Description automatically generated

Fig 7. Modul de conectare a display-ului

A circuit board

Description automatically generated

Fig 8. Butoanele folosite

# Rezultate experimentale

## Descrierea animației obținute

Inițial sunt afișate toate caracterele pe ecran, după care se face un delay pentru a putea fi observate. Apoi ecranul este golit și apare pe ecran Mario în varianta mică și o ciupercă. Sunt derulate în cele ce urmează 10 imagini cu delay de 200ms între ele pentru a oferi senzația de fluiditate. Mario înghite ciuperca și se face mare. Înainte de a se face mare, are loc o tranziție repetată între stările lui când era mic și când este mare, exact ca și în clasicul joc Mario. Mai jos este o imagine cu toate caracterele.

A circuit board

Description automatically generated

Fig 9. Caracterele desenate

În continuare este o imagine cu Mario în varianta mică și ciuperca.

A circuit board

Description automatically generated

Fig 10. Mario si ciuperca

Iar în final, Mario când este mare (după ce a înghițit ciuperca).

A circuit board

Description automatically generated

Fig 11. Mario mare

## Detalii tehnice despre implementare

Pentru a obține rezultatul dorit, s-au folosit 681 de celule, 11 porturi de Input/Ouput și 766 de nets. Frecvența maximă de funcționare este de 100MHz, iar a SPI-ului este de 3.125MHz. Procedura de testare utilizată a fost una practică pe display-ul propriul zis, încercând în repetate rânduri sincronizarea dorită pentru animație.

# Concluzii

Conluziile la care am ajuns în urma realizării acestui proiect sunt următoarele. În primul rând tehnologia OLED este de-a dreptul fascinantă și pot spune concret că acesta este viitorul. Realizarea SPI-ului a fost complicată, iar realizarea librăriei a fost muncitorească.

Contribuțiile originale ale acestei lucrări sunt SPI-ul, librăria de caractere și componenta folosită la derularea animației. Avantajul proiectului este că oferă la dispoziția utilizatorilor posibilitatea de a crea un posibil joc, aceasta fiind desigur o posibilitate de dezvoltare ulterioară. Modularitatea librăriei create reprezintă și ea un avantaj.

Aplicațiile proiectului meu au scop pur didactic demonstrativ, arătând posibilitățile acestui display. Adevărata aplicație este dezvoltarea ulterioară. Spre exemplu se poate reproduce jocul clasic de mario pe acest display, însă pentru realizarea acestei idei este nevoie de o echipă întreagă de designeri. Acest display se poate folosi si pe dispozitive wearables, de exemplu pentru un smartwatch, întrucât este suficient de subțire.

# Bibliografie

[1] <https://en.wikipedia.org/wiki/Serial_Peripheral_Interface>

[2] <https://en.wikipedia.org/wiki/Xilinx_Vivado>

[3] <http://users.utcluj.ro/~baruch/ssc/labor/Proiectare-Interfata.pdf>

[4] <https://store.digilentinc.com/pmod-oled-128-x-32-pixel-monochromatic-oled-display/>