## 

CALCULATOARE

**Interfață de utilizator bazată pe tastatură**

Proiect realizat pentru disciplina Structura Sistemelor de Calcul

Student: Iurea Mihaela-Cezara

Grupa 30238

Profesor îndrumător: Șopterean Andrei Mihai

Data: 8 ianuarie 2025

**Cuprins**

1. Rezumat ……………………………………………………………………….. 3
2. Introducere …………………………………………………………………….. 4
   1. Contextul temei proiectului și tendințele tehnologice …………………….. 4
   2. Domeniul de studiu și terminologia de bază ……………………………… 4
   3. Problema de rezolvat și obiectivele proiectului …………………………... 4
   4. Soluția propusă ……………………………………………………………. 4
   5. Conținutul raportului ……………………………………………………… 5
3. Fundamentare teoretică ………………………………………………………... 6
   1. Tastaturile matriciale ……………………………………………………… 6
   2. Comunicarea UART ……………………………………………………….. 6
   3. Utilizarea plăcii Zybo Z20 ………………………………………………… 6
4. Proiectare și implementare …………………………………………………….. 7
   1. Metoda experimentală utilizată …………………………………………… 7
   2. Soluția aleasă ……………………………………………………………… 7
   3. Arhitectura generală a sistemului …………………………………………. 7
   4. Algoritmul implementat …………………………………………………... 8
   5. Detalii de implementare …………………………………………………... 8
5. Rezultate experimentale ……………………………………………………….. 9
   1. Instrumente utilizate pentru proiectare și implementare ………………….. 9
   2. Procedura de testare ………………………………………………………. 9
   3. Rezultatele testării ………………………………………………………… 9
   4. Comparații ………………………………………………………………… 9
6. Concluzii ………………………………………………………………………. 10
7. **Rezumat**

Acest proiect avut la bază o interfață cu utilizatorul bazată pe o tastatură matricială Pmod KYPD conectată la platforma Zybo Z20. Scopul principal a fost de a crea un sistem capabil să citească apăsările de taste și să afișeze caracterul corespunzător în timp real, utilizând comunicarea UART. Problema rezolvată a constat în implementarea unei metode eficiente de interacțiune între utilizator și sistem printr-un dispozitiv hardware dedicat.

Proiectul a fost realizat prin configurarea unei arhitecturi hardware în Vivado, utilizând GPIO-uri pentru gestionarea semnalelor între rândurile și coloanele tastaturii. În Vitis, a fost implementat un algoritm software care activează pe rând liniile tastaturii, citește semnalele de pe coloane și determină tasta apăsată pe baza unei matrice de mapare predefinite. Rezultatul a fost transmis către un terminal serial care demonstreaza functionalitatea proiectului .

Rezultatele obținute au confirmat funcționarea corectă a sistemului , tastatura fiind capabilă să detecteze și să afișeze fiecare caracter apăsat. Proiectul evidențiază avantajele oferite de combinația FPGA-procesor a platformei Zybo Z20 și poate fi extins pentru aplicații mai complexe precum sisteme de securitate sau interfețe utilizator personalizate.

1. **Introducere**
   1. **Contextul temei proiectului și tendințele tehnologice**

În era tehnologiei moderne, interfețele de utilizator bazate pe dispozitive hardware, precum tastaturile fizice, joacă un rol esențial în aplicațiile de calculatoare. Tastaturile reprezintă o metodă simplă și eficientă de introducere a datelor în sistemele digitale, fiind utilizate pe scară largă în diverse domenii, inclusiv automatizări industriale, educație și interacțiuni cotidiene cu dispozitive electronice. În paralel, progresele în dezvoltarea microcontrolerelor și plăcilor de dezvoltare, precum platforma Zybo Z20, au permis integrarea tastaturilor cu funcționalități avansate, extinzând astfel posibilitățile de utilizare.

**2.2 Domeniul de studiu și tehnologia de bază**

Proiectul se încadrează în domeniul interfețelor om-mașină și al comunicațiilor hardware-software, cu un accent deosebit pe utilizarea tehnologiei UART (Universal Asynchronous Receiver-Transmitter). Acest protocol permite transmisii rapide de date între dispozitive, fără a necesita echipamente complexe suplimentare.

Platforma utilizată, Zybo Z20, este o placă de dezvoltare bazată pe arhitectura Zynq-7000, combinând un procesor ARM Cortex-A9 cu un FPGA programabil. Această combinație oferă flexibilitate maximă în proiectarea și implementarea interfețelor personalizate, cum ar fi cea realizată în cadrul acestui proiect. Tastatura Pmod KYPD, conectată prin interfața GPIO, permite introducerea de date utilizând o matrice de 16 taste.

**2.3 Problema de rezolvat și obiectivele proiectului**

Problema abordată în acest proiect constă în crearea unei interfețe de utilizator accesibile, bazată pe o tastatură matricială, care să permită transmiterea datelor către un computer. În acest sens s-au urmărit următoarele obiective :

* Integrarea tastaturii Pmod KYPD cu platforma hardware Zybo Z20;
* Implementarea unei aplicații care să permită citirea și interpretarea corectă a tastelor apăsate;
* Transmiterea datelor către un computer prin protocolul UART, pentru afișare în timp real într-un terminal serial (Hterm);

**2.4 Soluția propusă**

Soluția propusă constă în utilizarea mediilor de dezvoltare Vivado și Vitis:

1. Configurarea hardware-ului în Vivado:
   * Proiectarea unui block design pentru configurarea GPIO pe portul JB al plăcii Zybo Z20.
   * Exportarea designului către mediul Vitis.
2. Implementarea aplicației în Vitis:
   * Dezvoltarea codului sursă în limbajul C pentru a citi și interpreta semnalele provenite de la tastatura Pmod KYPD.
3. Testarea funcționalității:
   * Testarea interfeței utilizator-tastatură prin afișarea datelor în timp real într-un terminal serial (Hterm).
   * Validarea corectitudinii și a vitezei de răspuns a sistemului.

Această soluție evidențiază avantajele utilizării platformei Zybo Z20, datorate flexibilității oferite de combinația procesor-FPGA, și demonstrează o implementare eficientă a unei interfețe de utilizator hardware-software.

**2.5 Conținutul raportului**

Structura raportului este organizată astfel:

* Secțiunea 1 – Rezumat : Prezintă o privire de ansamblu asupra proiectului, incluzând obiectivele, soluția propusă, rezultatele obținute și concluziile principale.
* Secțiuna 2 – Introducere : Oferă contextul temei proiectului, descrie domeniul de studiu și tehnologia utilizată, clarifică problema de rezolvat și obiectivele, precum și soluția propusă.
* Secțiunea 3 – Fundamentare teoretică: Prezintă detalii despre tehnologiile utilizate, inclusiv GPIO, UART și modul de funcționare al tastaturii Pmod KYPD.
* Secțiunea 4 – Proiectare și implementare: Detaliază pașii hardware și software pentru integrarea componentelor proiectului.
* Secțiunea 5 – Rezultate experimentale: Include capturi de ecran și interpretări ale rezultatelor obținute în timpul testării.
* Secțiunea 6 – Concluzii: Rezumă principalele realizări ale proiectului și propune direcții de dezvoltare viitoare.

1. **Fundamentare teoretică** 
   1. **Tastaturile matriciale**

Tastaturile matriciale sunt dispozitive de intrare simple și eficiente care funcționează prin conectarea tastelor într-o matrice de rânduri și coloane, reducând numărul de pini necesari pentru a detecta apăsarea unei taste. În cazul proiectului de față, am utilizat tastatura matricială Pmod KYPD, care permite identificarea rapidă și precisă a tastelor apăsate prin intermediul interfețelor GPIO.

* 1. **Comunicarea UART**

Protocolul UART (Universal Asynchronous Receiver-Transmitter) este o metodă folosită pentru comunicarea serială. Acesta permite transferul datelor între dispozitive utilizând un număr minim de pini TX și RX. În cadrul proiectului, UART a fost utilizat pentru a trimite datele de la tastatura Pmod KYPD către terminalul serial al unui computer, fiind cel utilizat de pe placa Zybo Z20.

* 1. **Utilizarea plăcii Zybo Z20**

Platforma Zybo Z20 combină flexibilitatea procesoarelor ARM Cortex-A9 cu puterea de calcul paralelă a FPGA-urilor. Această combinație a permis implementarea eficientă a unei interfețe hardware-software pentru citirea tastaturii matriciale și transmiterea datelor prin UART. Pentru folosirea Zybo Z20 si Pmod KYPD, am folosit:

* [1] Documentația oficială Digilent pentru Pmod KYPD.
* [2] Manualul de utilizare al plăcii Zybo Z20

1. **Proiectare și implementare**

Această secțiune prezintă în detaliu toate etapele parcurse pentru realizarea proiectului, de la proiectarea hardware, implementarea software-ului, la integrarea și validarea soluției finale.

* 1. **Metoda experimentală utilizată**

Proiectul a fost realizat utilizând o combinație de:

* Implementare hardware: Configurarea modulelor necesare în Vivado pentru interfațarea cu tastatura Pmod KYPD.
* Implementare software: Dezvoltarea codului în Vitis pentru citirea tastelor și afișarea caracterelor

Metoda experimentală a implicat configurarea perifericelor GPIO și UART pe platforma hardware, urmată de implementarea unui algoritm software pentru gestionarea tastaturii matriciale.

* 1. **Soluția aleasă**

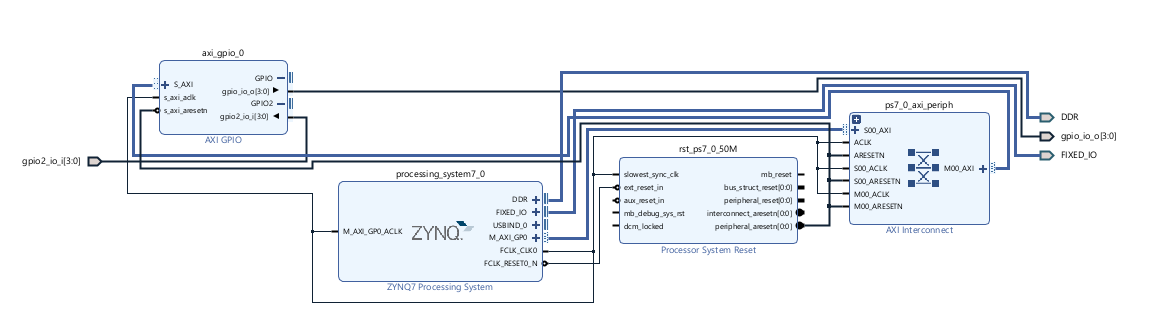
Am ales soluția de implementare bazată pe utilizarea interfețelor GPIO pentru citirea tastelor și UART pentru afișarea datelor. Alegerea acestei soluții se bazează pe următoarele avantaje:

* Eficiență: Utilizarea modulelor preexistente în Vivado și Vitis reduce complexitatea implementării.
* Module deja integrate: Platforma Zybo Z20 include deja un protocol UART, eliminând folosirea unor module suplimentare cum ar fi Pmod USBUART.
* Flexibilitate: Soluția poate fi extinsă sau modificată ușor pentru alte tipuri de periferice

**4.3 Arhitectura generală a sistemului**

* Arhitectura hardware

Schema bloc hardware a fost realizată în Vivado și include următoarele componente principale:

1. Zynq Processing System: execută aplicația software.
2. AXI GPIO: Permite interfațarea cu tastatura Pmod KYPD.
   * GPIO\_io\_o: Controlează rândurile tastaturii.
   * GPIO2\_io\_i: Citește semnalele de pe coloane.
3. Reset System: Gestionează resetarea perifericelor.
4. AXI Interconnect: Conectează procesorul cu perifericele.

* Arhitectura software

Software-ul implementează un algoritm care:

1. Activează pe rând fiecare linie a tastaturii.
2. Citește semnalele generate pe coloane.
3. Mapează coordonatele detectate (rând, coloană) la caracterul corespunzător.
4. Afișează rezultatul în terminal prin UART.
   1. **Algoritmul implementat**

Tastatura Pmod KYPD este o matrice 4x4, iar caracterele sunt mapate conform imaginii de mai jos:

Dupa ce designul hardware a fost generat și exportat către Vitis, implementăm algoritmul de citire:

Pentru a determina ce tastă este apăsată pe tastatura Pmod KYPD, sistemul activează pe rând fiecare dintre cele patru linii ale matricei de taste folosind GPIO\_io\_o. Când o linie este activată, sistemul citește semnalele generate pe cele patru coloane folosind GPIO2\_io\_i pentru a detecta dacă o tastă este apăsată. Coordonatele rând-coloană corespunzătoare tastei apăsate sunt apoi mapate la caracterul asociat folosind funcția mapKey, care utilizează o matrice predefinită pentru această asociere. În cele din urmă, caracterul detectat este trimis către un terminal serial prin UART, unde este afișat utilizatorului. Acest proces se repetă continuu pentru a monitoriza apăsarea tastelor.

**4.5 Detalii de implementare**

* Tastatura matricială Pmod KYPD este conectată la portul JB al plăcii Zybo Z20.
* Rândurile tastaturii (R0, R1, R2, R3) sunt configurate ca ieșiri GPIO și sunt responsabile pentru activarea secvențială a fiecărui rând. Aceasta se face prin setarea unui rând la nivel logic scăzut (0), în timp ce celelalte rânduri rămân dezactivate (1).
* Coloanele tastaturii (C0, C1, C2, C3) sunt configurate ca intrări GPIO pentru a detecta semnalele generate de apăsarea tastelor. Dacă o tastă de pe rândul activ este apăsată, coloana corespunzătoare va semnala acest lucru.
* Terminalul Hterm este utilizat pentru afișarea în timp real a tastelor apăsate. Acesta este configurat să primească date prin UART de la placa Zybo Z20.

1. **Rezultate experimentale**

În cadrul acestui proiect, am utilizat o abordare practică pentru a testa funcționalitatea și performanța sistemului, utilizând instrumente software și hardware avansate. Detaliile sunt prezentate mai jos.

**5.1 Instrumentele utilizate pentru proiectare și implementare**

Am utilizat:

- Vivado 2024.1 pentru proiectarea hardware și configurarea conexiunilor GPIO

- Platformă hardware: Placa de dezvoltare Zybo Z20, bazată pe procesorul Zynq-7000.

- Sistem de operare: Windows 10.

- Instrumentul de programare: Vitis IDE

- Terminal pentru afișare: Hterm, configurat pentru recepția datelor transmise prin UART

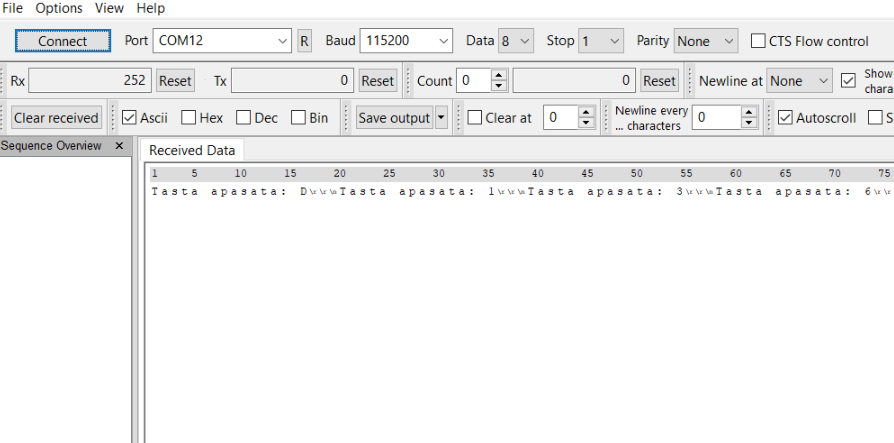
* 1. **Procedura de testare**

Sistemul a fost testat utilizând o combinație de metode:

* + Testare hardware: După implementarea designului în Vivado, tastatura a fost conectată la portul JB al plăcii Zybo Z20. Fiecare tastă a fost apăsată pentru a verifica corespondența dintre caracterul apăsat și cel afișat în terminalul Hterm.
  + Testare software: Algoritmul de citire a fost testat în Vitis utilizând funcții de debugging pentru a verifica corectitudinea valorilor citite din GPIO și maparea acestora.
  + Validarea interfeței UART: Terminalul Hterm a fost configurat cu un baud rate corespunzător pentru a asigura comunicarea corectă între placa Zybo Z20 și computer.

**5.3 Rezultatele testării**

Sistemul a afișat cu succes caracterele tastate pe Pmod KYPD în terminalul Hterm, după cum se poate observa în capturile de ecran de mai jos:



* 1. **Comparații**

Comparând această implementare cu soluții similare, precum utilizarea unui modul Pmod USBUART, abordarea finală s-a dovedit a fi mai simplă și mai eficientă, eliminând necesitatea unui modul suplimentar și reducând astfel complexitatea hardware.

1. **Concluzii**

Proiectul acesta a reușit să rezolve problema citirii tastelor de pe tastatura matricială și să trimită datele către un computer folosind platforma Zybo Z20. Toate obiectivele propuse au fost atinse, iar sistemul funcționează corect.

Am învățat cum să folosesc GPIO pentru a controla tastatura și cum să implementez o comunicare serială prin UART. Proiectul are avantajul că este simplu și eficient, dar este limitat la tastaturi de tip matrice și necesită hardware specific. Acest sistem poate fi folosit în proiecte precum sisteme de acces, autentificare, introducerea datelor etc. În viitor, aș putea să îmbunătățesc proiectul adăugând funcții precum detectarea mai multor taste apăsate simultan sau o conexiune wireless pentru transmiterea datelor.

**Bibliografie**

[1] Digilent, Inc., *Zybo Z20 Reference Manual*, Digilent Documentation, 2019. Disponibil online: https://digilent.com/reference/\_media/zybo-z7:zybo\_z7\_rm.pdf

[2] Digilent, Inc., *Pmod KYPD Reference Manual*, Digilent Documentation, 2018. Disponibil online: https://digilent.com/reference/\_media/pmod:pmod:kypd\_rm.pdf

**Anexa A**

 #include <stdio.h>

#include "platform.h"

#include <xgpio.h>

#include "xparameters.h"

#include "sleep.h"

char identificare\_tastatura(int rand, int coloana) {

    const char tastatura[4][4] = {

        {'1', '2', '3', 'A'},

        {'4', '5', '6', 'B'},

        {'7', '8', '9', 'C'},

        {'0', 'F', 'E', 'D'}

    };

    return tastatura[rand][coloana];

}

int main() {

    XGpio gpio;

    int rand, coloana;

    char tasta;

    int valoriRanduri[4] = {0b1110, 0b1101, 0b1011, 0b0111};

    int valoriColoane[4] = {0b1110, 0b1101, 0b1011, 0b0111};

    int colVal;

    init\_platform();

    XGpio\_Initialize(&gpio, XPAR\_AXI\_GPIO\_0\_BASEADDR);

    XGpio\_SetDataDirection(&gpio, 1, 0x0); // randurile ca iesire

    XGpio\_SetDataDirection(&gpio, 2, 0xF); // coloanele ca intrare

    while (1) {

        for (rand = 0; rand < 4; rand++) {

            XGpio\_DiscreteWrite(&gpio, 1, valoriRanduri[rand]);

            usleep(1000);

            colVal = XGpio\_DiscreteRead(&gpio, 2);

            for (coloana = 0; coloana < 4; coloana++) {

                if (colVal == valoriColoane[coloana]) {

                    tasta = identificare\_tastatura(rand, coloana);

                    printf("Tasta apăsată: %c\r\n", tasta);

                    while (XGpio\_DiscreteRead(&gpio, 2) != 0xF);

                    break;

                }

            }

        }

        usleep(20000);

    }

    cleanup\_platform();

    return 0;

}