



Catedra: Structura Sistemelor de Calcul

Comunicația dintre placa Nexys 4 DDR și un mouse USB HID

Studenti: Strujan Florentina, Stroe Mădălina

Îndrumător proiect: Dragoș-Florin Lișman

Data: 18.10.2020



Cuprins

1.Rezumat.....	2
2.Introducere.....	3
3.Fundamentare teoretică	4
3.1. VHDL	4
3.2. FPGA	5
3.3. Nexys 4 DDR	5
3.4. Mouse USB HID	7
3.5. Protocol de comunicare PS/2	7
3.6. Xilinx Vivado Design Suite	8
3.7.Eclipse.....	8
4.Proiectare și implementare	9
4.1 Diagrama de stări	9
4.2.Diagrama bloc.....	10
5.Rezultate experimentale	13
6.Concluzii	18
Bibliografie	19



1.Rezumat

Tema proiectului realizat este Comunicația dintre placa Nexys 4 DDR și un mouse USB HID, temă ce a implicat anumite dificultăți dar a și adus un nou bagaj de cunoștințe, fiind o temă nouă și diferită față de ce am realizat până în momentul de față.

Tema propusă spre implementare a necesitat folosirea limbajului VHDL, dar și folosirea unui al doilea limbaj pentru realizarea aplicației pe calculator, aplicație necesară pentru afișarea cursorului și a stării butoanelor. Implementarea s-a făcut pe placa Digilent Nexys 4 DDR, fiind nevoie și de un mouse USB HID.

Ca rezultate avem simulările care funcționează conform cerințelor, dar și aplicația Java care afișează rezultatele în funcție de informația primită.



2.Introducere

Un FPGA este un tip de circuit logic programabil, configurabil de către utilizator, cu ajutorul unui limbaj de descriere hardware HDL. Un mouse este un dispozitiv hardware periferic foarte important pentru introducerea de comenzi. Principiul acestuia de funcționare este bazat pe recunoașterea de către computer a mișcării sale. Mișcarea este detectată de un senzor din partea inferioară, preluată, digitalizată și transmisă computerului printr-o interacțiune potrivită cu un FPGA(interfața). Protocolul de comunicație (PS2) utilizat presupune un canal de comunicare bidirecțională serială. Interfața are 2 semnale principale: Data și Clock. Avem noțiunile de host(plăcuță sau pc) și device(care e, în principiu, mouse-ul). Device-ul generează mereu un semnal de ceas, dar host-ul are control asupra comunicației PS2. Dacă host-ul vrea să trimită date, întrerupe comunicarea cu un clock de nivel low. Pentru a reveni la comunicare de la device la host, host-ul trebuie să aibă Data low și să elibereze clock-ul, ceea ce duce într-o stare de "Request to Send" care semnalează device-ului că poate să înceapă să transmită iar semnal de clock.

Starile sunt următoarele:

Data=H , Clock=H --> Disponibil

Data=H , Clock=L --> Comunicare întreruptă

Data=L , Clock=H --> Request To Send

Tema propusă spre implementare este: Comunicația dintre placa Nexys 4 DDR și un mouse USB HID.

Obiectivele principale ale acestei teme sunt:

- Proiectare utilizând limbajul VHDL
- Implementare pe placa Digilent Nexys 4 DDR
- Aplicație pe calculator pentru afișarea cursorului și a stării butoanelor

Soluția aleasă pentru rezolvarea temei alese, și anume Comunicația dintre placa Nexys 4 DDR și un mouse USB HID este folosirea protocolului de comunicare PS/2 (IBM Personal System/2) utilizat de mouse, protocol serial bidirecțional. Dispozitivele care utilizează acest protocol folosesc two-wire serial bus (clock și data), datele fiind transmise bit cu bit, cuvintele fiind de 11 biți astfel: primul bit de start (mereu 0) , un bit de paritate(odd parity) , un bit de stop (mereu 1) și restul biților de date(8). Interfața PS / 2, utilizată de multe tipuri de mouse-uri și tastaturi moderne, a fost dezvoltată de IBM și a apărut inițial în IBM Technical Reference Manual.



Fundamentele teoretice legate de proiect, cum ar fi modele, metode și tehnologii vor fi detaliate în secțiunea „Fundamentare teoretică”. De asemenea, tot în secțiunea „Fundamentare teoretică” vor fi descrise și semnalele de control și utilizarea acestora pentru realizarea proiectului. Secțiunea „Proiectare și implementare” descrie etapele parcurse în realizarea obiectivelor proiectului, dar și o soluție a acestei implementări în limbajul VHDL. Rezultatele obținute prin implementarea proiectului sunt prezentate în secțiunea „Rezultate experimentale”. În secțiunea „Concluzii” vor fi enunțate concluziile care au fost trase în urma implementării și, de asemenea, vor fi prezentate și posibile dezvoltări ulterioare.

3. Fundamentare teoretică

Pentru proiect, va fi necesar:

- Hardware:
 - Placă Nexys4
 - Cablu micro-USB
 - Mouse USB
- Software:
 - Vivado Design Suite
 - Eclipse

3.1. VHDL

VHDL este un limbaj de descriere a hardware-ului (VHSIC Hardware Description Language), destinat descrierii comportamentului și arhitecturii unui modul electronic logic, cu alte cuvinte, al unei funcțiuni logice combinatorii sau secvențiale. Alături de Verilog, este cel mai utilizat limbaj de proiectare a sistemelor electronice digitale. Este una din uneltele principale pentru proiectarea circuitelor integrate moderne, aplicat cu succes în câmpul microprocesoarelor, în telecomunicații, automobile. Acest limbaj este utilizat în concepția asistată pe calculator (CAD) a circuitelor integrate sau pentru configurarea FPGA-urilor.



3.2. FPGA

Un FPGA (Field Programmable Gate Array) este un circuit integrat digital care poate fi configurat de către utilizator după ce a fost fabricat. FPGA-urile sunt alcătuite din blocuri logice configurabile care comunică unele cu altele prin conexiuni configurabile. Un bloc logic FPGA clasic este compus, după cum se observă și în figura de mai jos, dintr-un Look-Up Table cu 4 intrări, un bistabil și un multiplexor care selectează ieșirea.

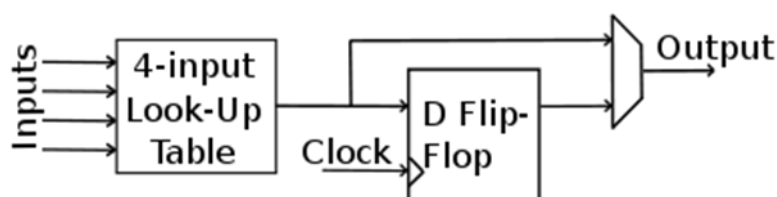


Fig1. Bloc logic FPGA clasic

Configurarea FPGA se face prin intermediul unui limbaj de descriere hardware (HDL) cum sunt VHDL și Verilog. Acest proiect a fost implementat în limbajul VHDL utilizând mediul de dezvoltare Xilinx Vivado.

FPGA-urile sunt folosite în foarte multe domenii, de la industria auto până la domeniul medical, securitate și sisteme audio/video.

3.3. Nexys 4 DDR

Placa este o platformă completă, gata de utilizare pentru dezvoltarea de circuite digitale bazate pe cele mai recente FPGA-uri Artix-7 de la Xilinx®. Cu o mare capacitate FPGA, memorii externe generoase, precum și o colecție de USB, Ethernet și alte porturi, Nexys4 DDR poate găzdui modele variind de la circuite combinaționale introductive la procesoare puternice încorporate. Mai multe tipuri de periferice încorporate, inclusiv un accelerometru, senzor de temperatură, MEMS microfon digital, un amplificator, și mai multe dispozitive I / O permit utilizarea plăcii Nexys4 DDR pentru o gamă largă de modele, fără a avea nevoie de alte componente. Sistemul implementat, realizat utilizând mediul de proiectare Vivado, va fi încărcat în circuitele plăcii Nexys4 DDR și va fi verificată funcționarea acestuia digital în condiții reale.

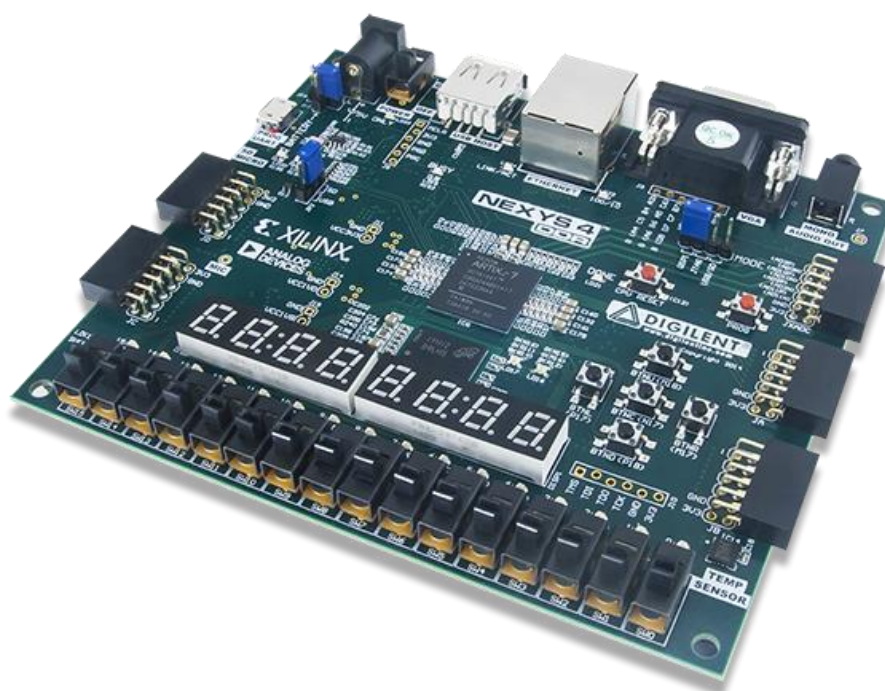


Fig2. Placa Nexys4 DDR

Placa Nexys4 oferă porturi și periferice:

- 16 switch-uri
- 16 led-uri
- 2 afișoare 7-segmente
- Conector card microSD
- Microfon PMD
- 4 Porturi PMOD
- USB HID pentru mouse, tastatura și stick-uri de memorie
- Senzor de temperatură
- ieșire VGA 12 biți
- 2 led-uri tri-color
- Flash serial
- Port USB Digilent Adept pentru programare și date
- 10/100 Ethernet PHY

Pentru tema dezvoltată am folosit doar USB HID pentru mouse, tastatura și stick-uri de memorie și ieșirea VGA 12 biți.



3.4. Mouse USB HID

Mouse-ul este un dispozitiv comun din clasa USB HID. Mouse-ul USB HID poate varia de la dispozitive simple cu un singur buton la dispozitive compuse cu mai multe butoane. Majoritatea sistemelor de operare moderne sunt livrate cu drivere pentru modele standard de mouse HID (cel mai obișnuit design modern al mouse-ului are două butoane dedicate și o roată a mouse-ului care se dublează ca al treilea buton); mouse-urile cu funcționalitate extinsă necesită drivere personalizate de la producător.

Protocolul de comunicare utilizat de mouse este PS/2 (IBM Personal System/2). Datele sunt transmise bit cu bit (cuvinte de 11 biți).

Cuvintele sunt formate din:

- Un bit de start(mereu 0)
- Un bit de paritate(odd parity)
- Un bit de stop(mereu 1)
- Restul biților de date(8)

Când mouse-ul intră în modul stream și data reporting, transmite spre ieșiri clock și semnal de date când e mișcat, altfel semnalele rămân la 1 logic.

De fiecare dată când e mișcat mouse-ul, 3 cuvinte de 11 biți sunt transmise de la mouse la dispozitivul gazdă, deci fiecare transmisiune are 33 de biți. Datele sunt valide la falling edge și perioada clock-ului este de la 20 la 30 de KHz.

Lucrând cu un sistem de coordonate relative, când mouse-ul este mișcat la dreapta, se generează un număr pozitiv, iar dacă e mișcat la stânga, se generează un număr negativ(similar și pe axa Y).

3.5. Protocol de comunicare PS/2

Protocolul de comunicație (PS/2) utilizat presupune un canal de comunicare bidirecțională serială. Interfața are 2 semnale principale: Data si Clock. Avem noțiunile de host(plăcuță sau pc) și device(care e, in principiu, mouse-ul). Device-ul generează mereu un semnal de ceas, dar host-ul are control asupra comunicației PS2. Dacă host-ul vrea sa trimită date, întrerupe comunicarea cu un clock de nivel low. Pentru a reveni la comunicare de la device la host, host-ul trebuie sa aibă Data low și să elibereze clock-ul, ceea ce duce într-o stare de "Request to Send" care semnalează device-ului că poate să înceapă să transmită iar semnal de clock.

Starile sunt urmatoarele:

Data=H , Clock=H --> Disponibil

Data=H , Clock=L --> Comunicare intrerupta

Data=L , Clock=H --> Request To Send

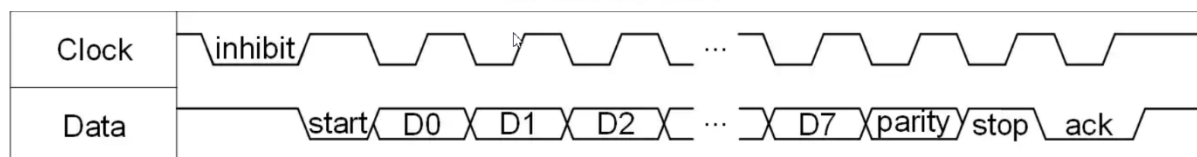


Fig3. Receptie PS/2

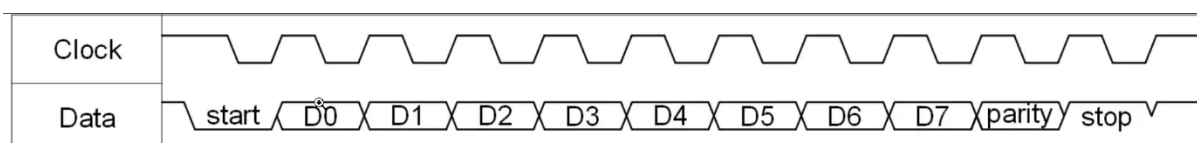


Fig4. Transmisie PS/2

3.6. Xilinx Vivado Design Suite

Mediul de proiectare Xilinx Vivado Design Suite integrează toate utilitățile necesare pentru proiectarea sistemelor digitale utilizând circuitele FPGA Xilinx din seria 7.

Interfața grafică Vivado IDE permite invocarea utilităților de proiectare corespunzătoare pentru executarea operațiilor necesare în diferitele etape ale fluxului de proiectare cu circuite FPGA, cum sunt descrierea sistemului, sinteza, implementarea, analiza proiectului, configurarea circuitului, verificarea și depanarea. Facilitățile de analiză a proiectului cuprind simularea logică în diferitele etape ale fluxului de proiectare, definirea constrângerilor, verificarea regulilor de proiectare, analiza de timp, analiza puterii consumate și vizualizarea logicii proiectului.

3.7. Eclipse

Eclipse este un mediu de dezvoltare open-source scris preponderent în Java. Acesta poate fi folosit pentru a dezvolta aplicații Java și, prin intermediul unor plug-in-uri, în alte limbaje, cum ar fi C, C++, COBOL, Python, Perl și PHP.



4. Proiectare și implementare

4.1 Diagrama de stări

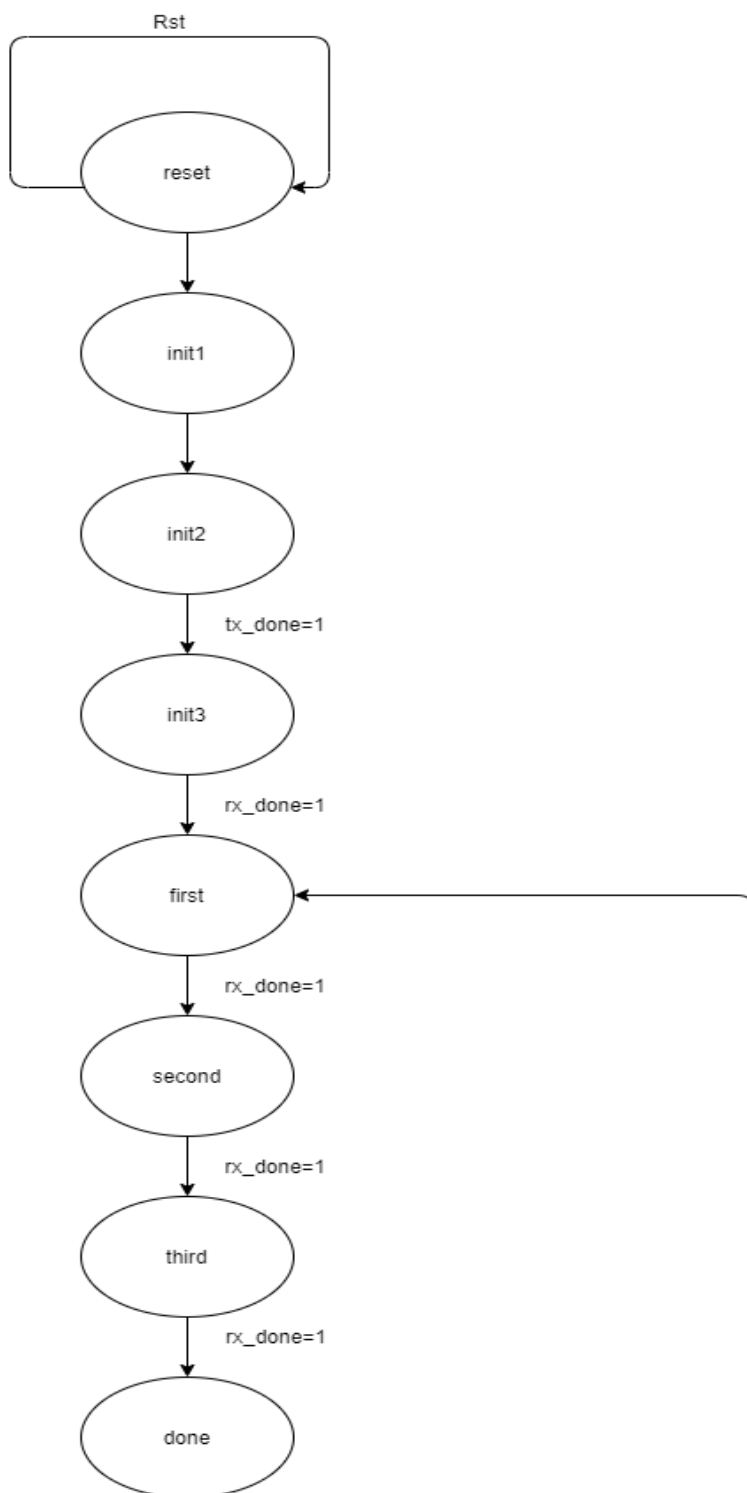


Fig5. Diagrama de stări



4.2.Diagrama bloc

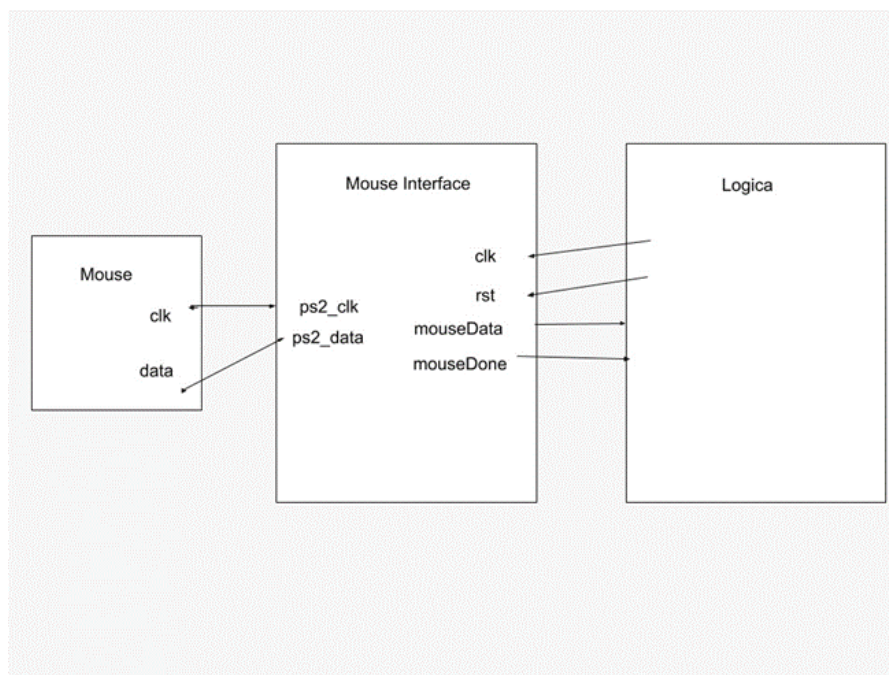


Fig6.Diagrama bloc

Mai sus avem diagrama bloc a proiectului, mouse-ul cu semnalul de clock si data, logica urmează a fi implementată mai târziu, iar mouse interface este interfața (implementează o comunicare bidirecțională sincronă) care conține un transmițător și un receptor de informație (se pot vedea în schemele de mai jos).

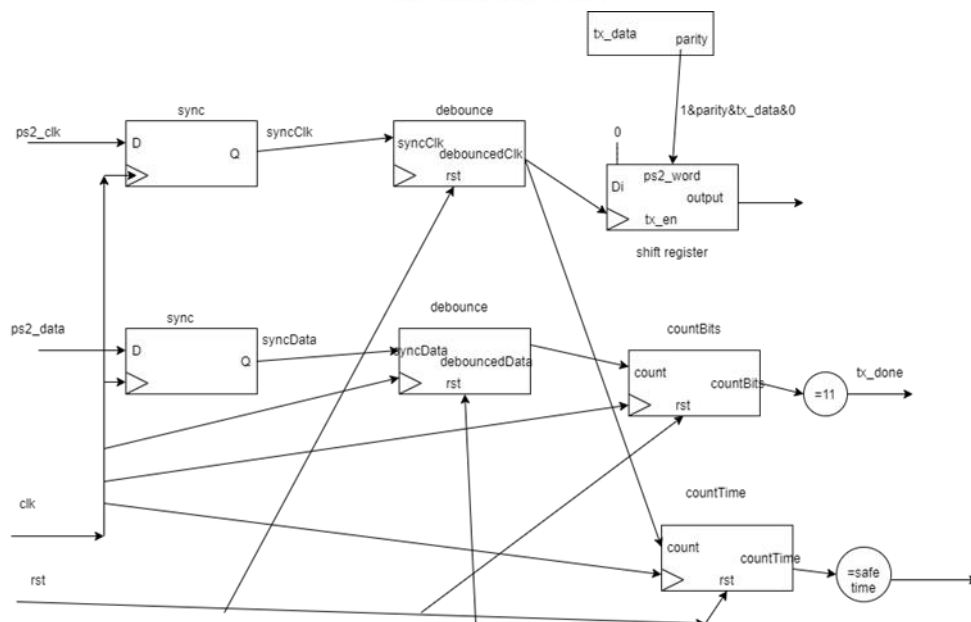


Fig7.

Ps2_clk si ps2_data sunt de tip in-out pentru mouse. Aceste semnale urmează să fie sincronizate, așadar le vom trece prin bistabile de tip D(sync pe schema). Ieșirile bistabilelor reprezintă intrări pentru debouncer, iar ieșirile lor sunt semnalele debouncedCik și debouncedData.

Cuvântul care se transmite este încărcat într-un shift register, după ce i se adaugă bitul de paritate(care se calculează anterior), biții de start și stop ai informației. Serial, se va transmite la mouse bitul cel mai nesemnificativ, după care se face o deplasare si se înlocuiește cu 0. Mai departe, counterul de timp incrementează până ajunge într-o stare sigură în care se poate trece în starea de transmitere. Counterul de biți va număra pana la 11 pentru a se asigura de corectitudinea informației, apoi se va activa semnalul de tx_Done.

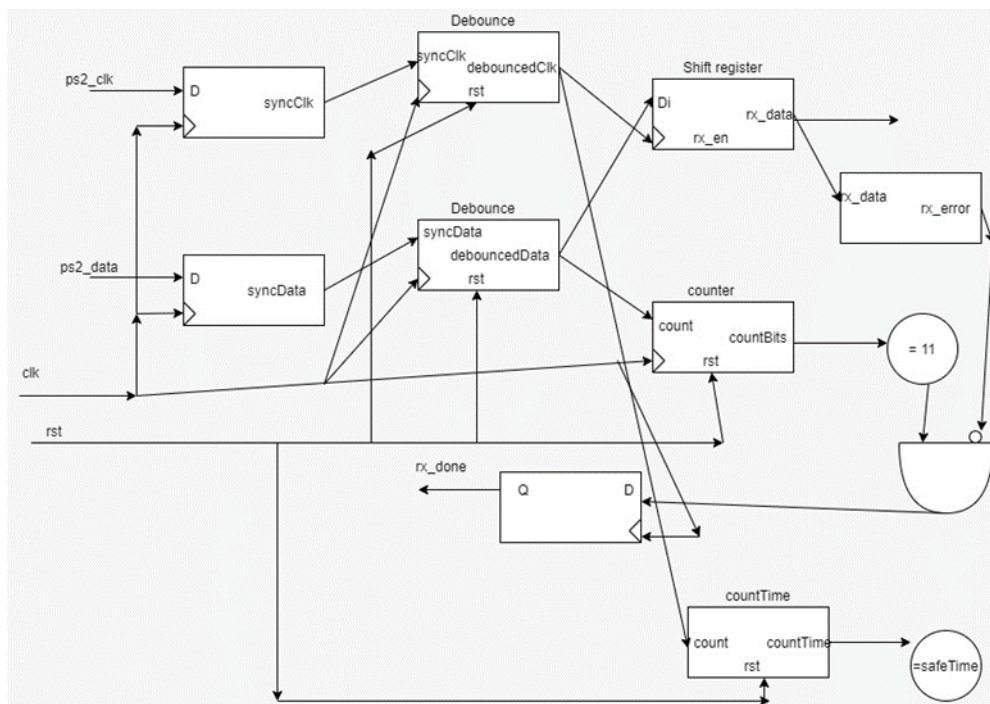


Fig8.

Ps2_clk si ps2_data sunt de tip in-out pentru mouse. Aceste semnale urmează să fie sincronizate, așadar le vom trece prin bistabile de tip D(sync pe schema). Ieșirile bistabilelor reprezintă intrări pentru debouncer, iar ieșirile lor sunt semnalele debouncedClk și debouncedData.

În registrul de deplasare ajung datele de la debounceData, serial. Se activează semnalul rx_en atunci când se pot primi date. Ieșirea rx_data este trecută printr-o verificare a corectitudinii. În cazul în care informația este corectă(rx_error = 0) si numărul de biți primiți este 11, recepția este finalizată, așadar rx_done este activat.



5.Rezultate experimentale

În figura de mai jos sunt rezultatele simulării componenteii uart:

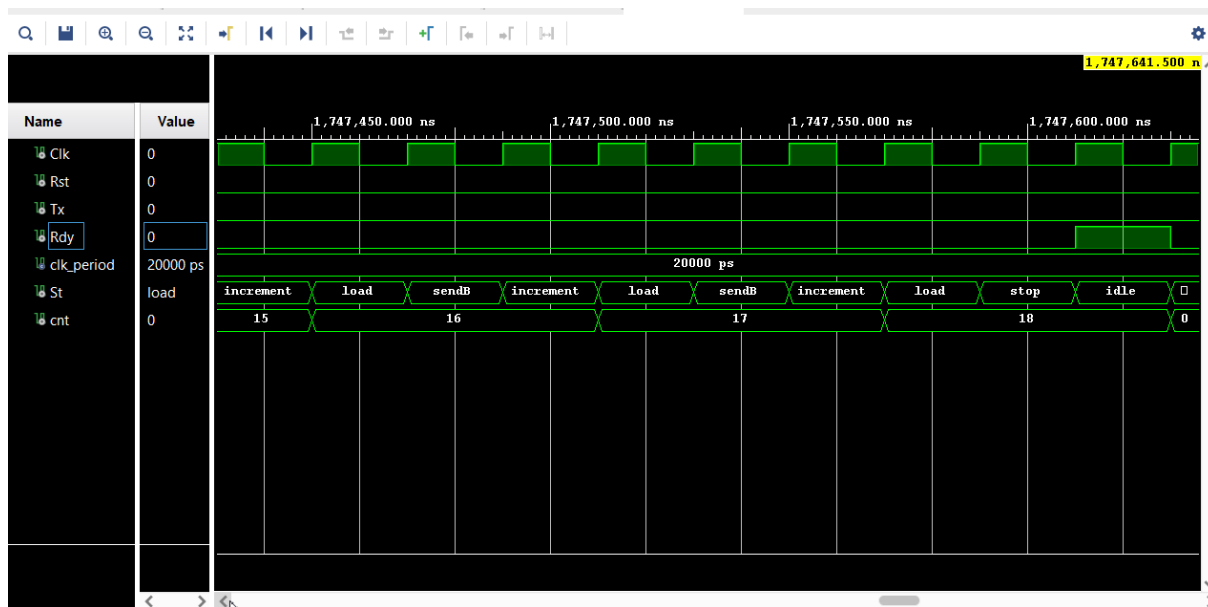


Fig9. Rezultatul simulării pentru uart

În figurile de mai jos sunt rezultatele simulării componenteii rx:

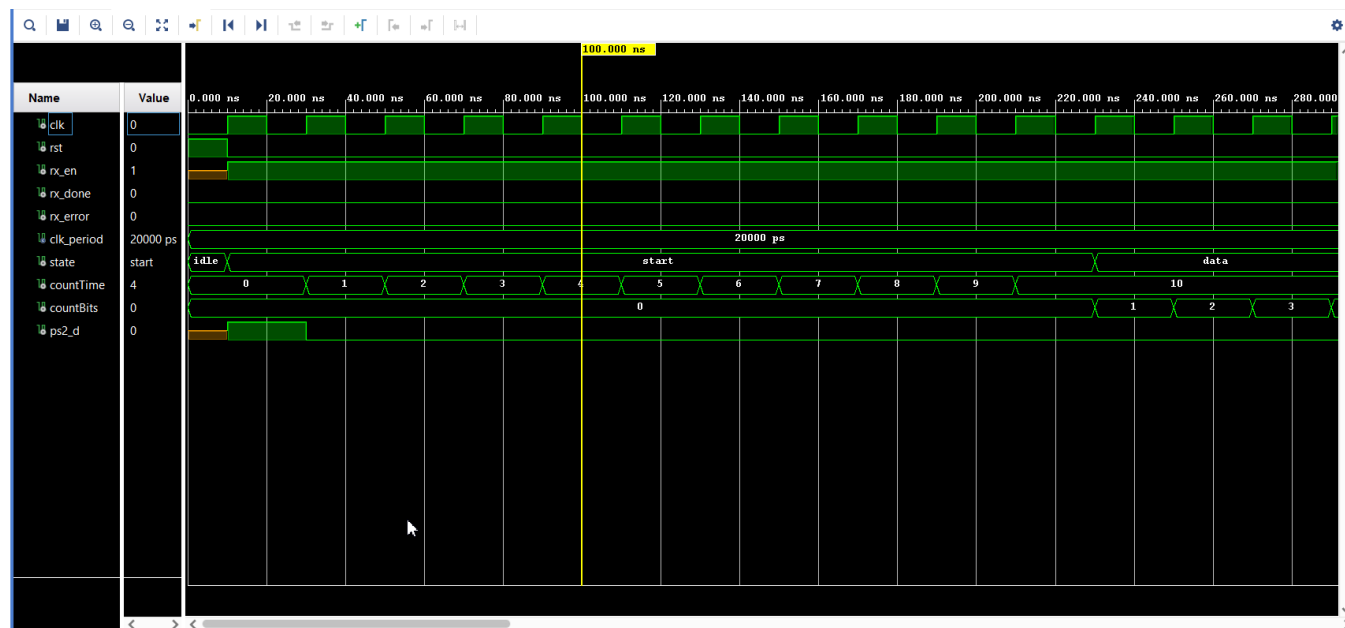


Fig10.1. Rezultatul simulării pentru rx

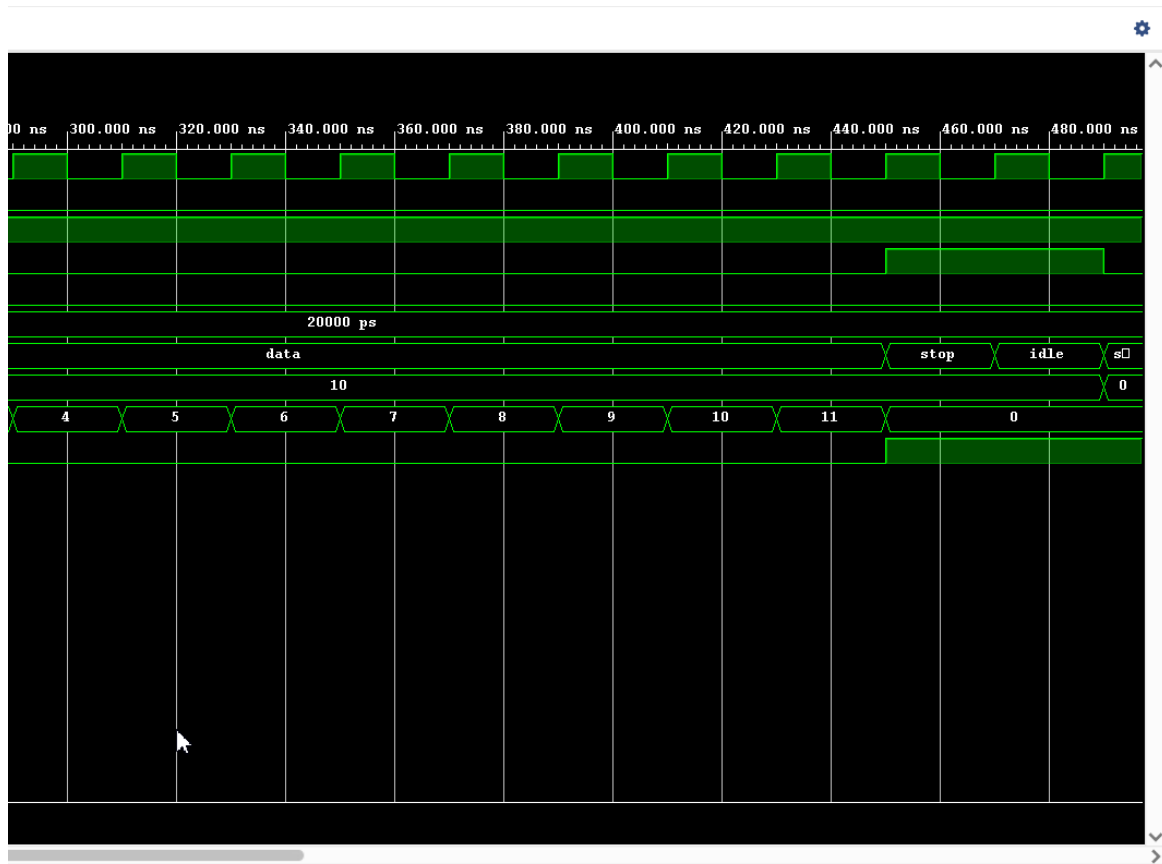


Fig10.2. Rezultatul simulării pentru rx



În figurile de mai jos sunt rezultatele simulării componentei tx:

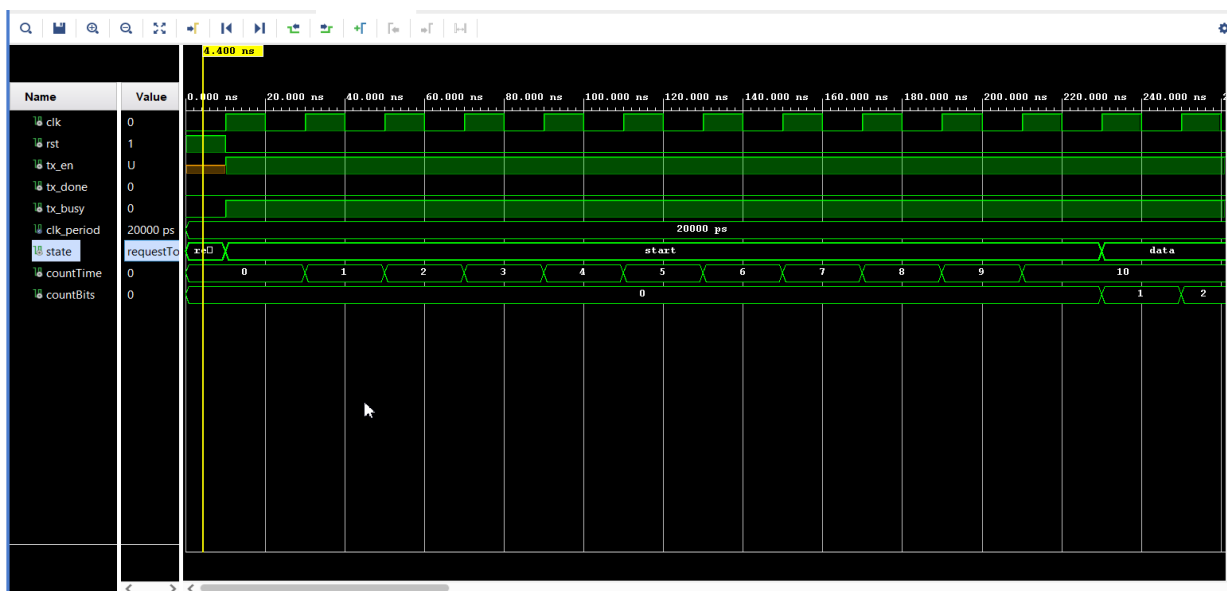


Fig11.1. Rezultatul simulării pentru tx

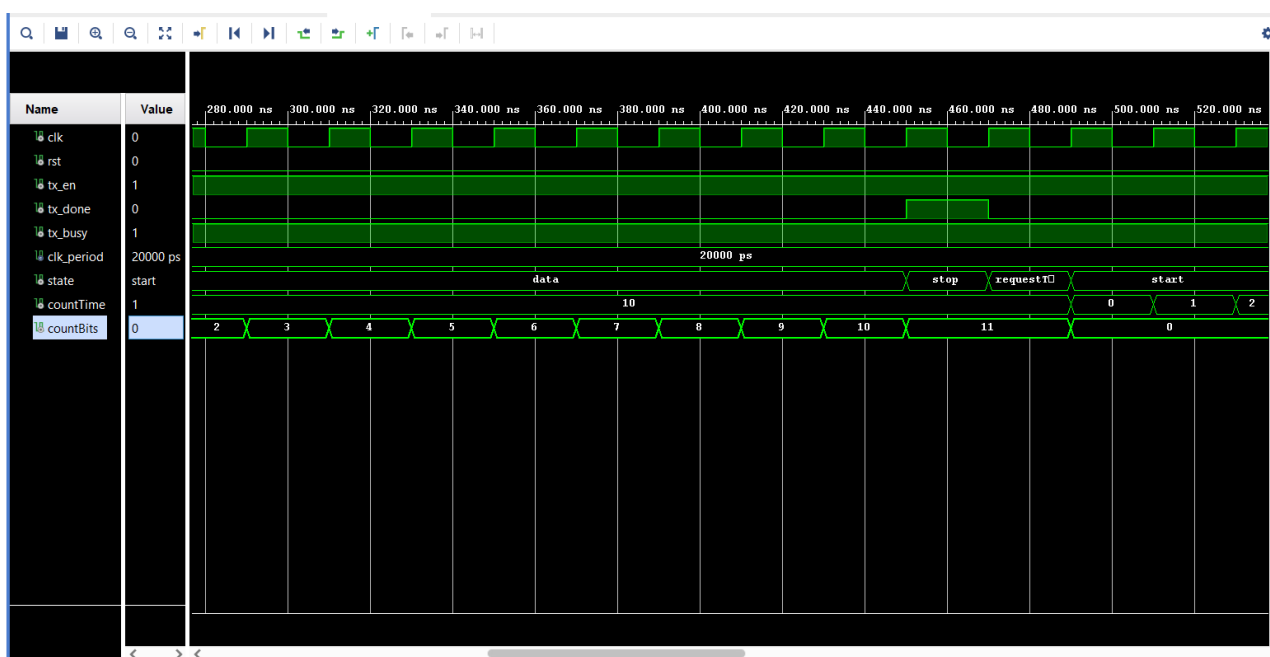


Fig11.2. Rezultatul simulării pentru tx



Modulul UART a fost realizat în cadrul unui laborator din acest semestru. Pentru a vedea funcționalitatea acestuia, atunci când un buton de pe placa se apasă, se afișează un mesaj în terminal.

În cadrul proiectului, între cei 33 de biți transmiși de mouse, 3 biți arată ce butoane au fost apăstate de la mouse. În funcție de valorile acestora, ar trebui să apară în terminal un mesaj corespunzător.

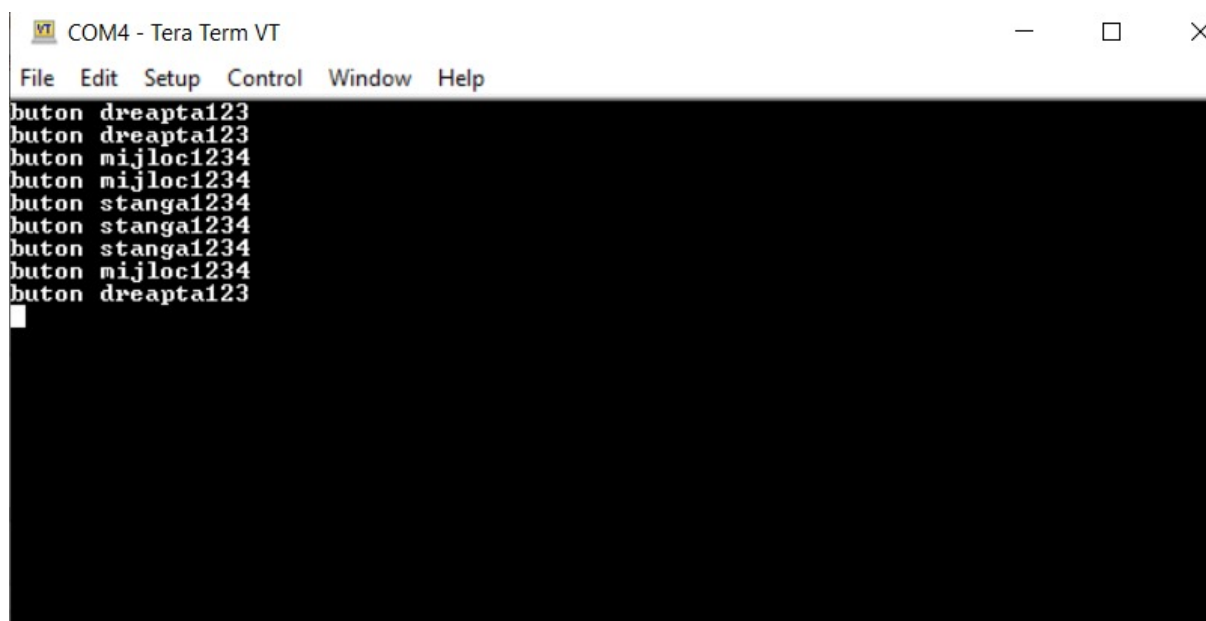


Fig12. Terminalul Tera Term demonstrând funcționarea uart-ului



Am realizat o aplicație pentru afișarea rezultatelor pe calculator. În funcție de poziția mouse-ului, cursorul va fi poziționat pe ecran, iar forma cursor-ului va fi schimbată la un click dreapta, stânga sau mijloc.

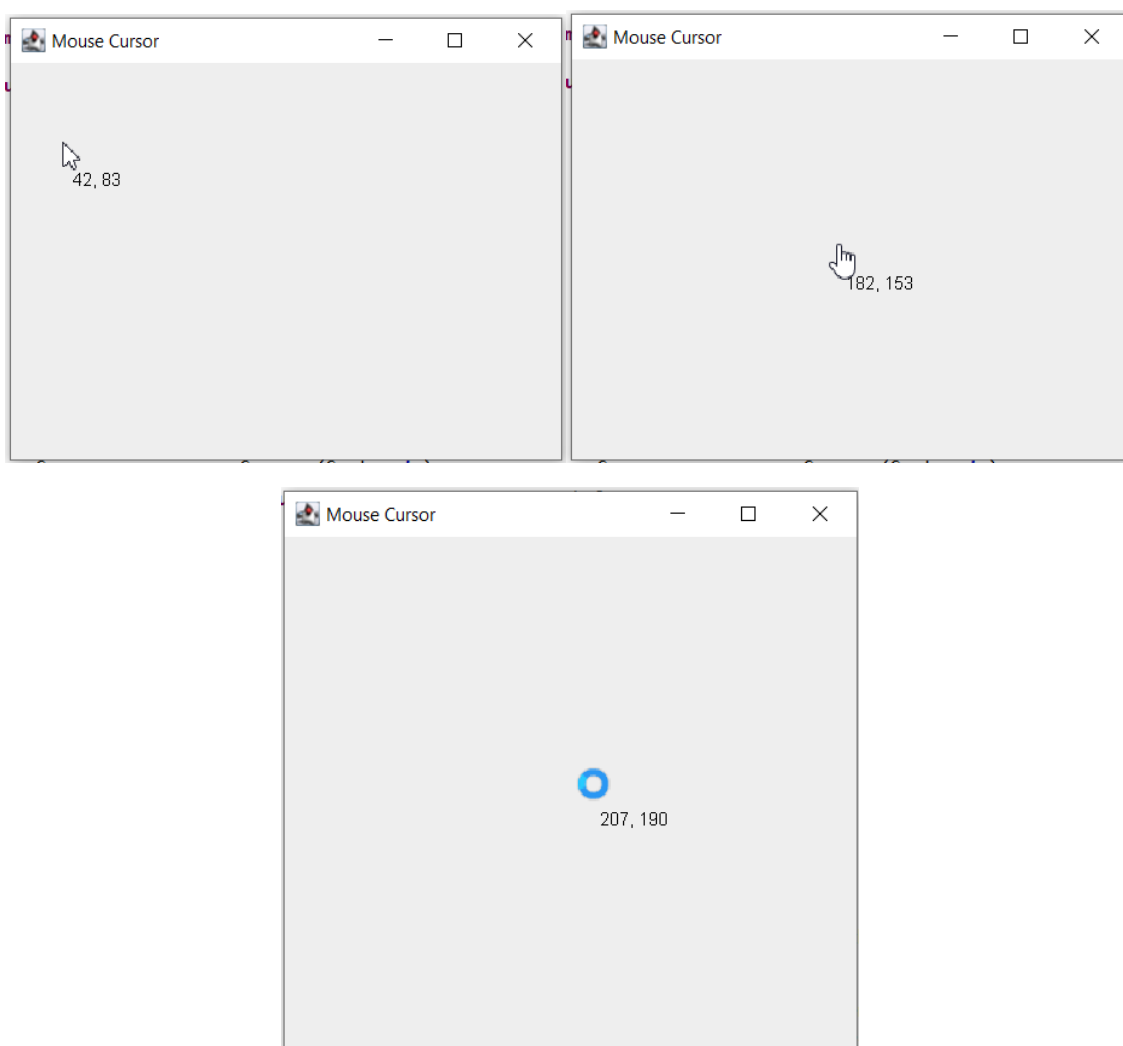


Fig13.Exemple de funcționare ale aplicației Java



6. Concluzii

Problema rezolvată de acest proiect este Comunicarea dintre placa Nexys 4 DDR și un mouse USB HID. Testarea pe placă a proiectului nu a fost realizată cu succes, însă test bench-urile funcționează conform cerințelor. Aplicația Java funcționează și afișează rezultatele dorite.

Pentru viitor aplicația ar putea fi reluată pentru găsirea și remedierea problemelor, pentru funcționarea inclusiv pe placă.

Rezolvând acest proiect ne-am îmbunătățit abilitățile de lucru cu Vivado, am aprofundat cunoașterea limbajului vhdl și ne-am confruntat cu altfel de probleme față de cele de la laborator, având o temă complexă, ce ține și de alte componente hardware.



Bibliografie

<https://reference.digilentinc.com/reference/programmable-logic/nexys-4/reference-manual>

http://www.burtonsys.com/ps2_chapweske.htm

<https://www.digikey.com/eewiki/pages/viewpage.action?pageId=70189075>

https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-vga-test-pattern-with-mouse-overlay/start?_ga=2.141608374.526639414.1603008038-251961870.1603008038

<https://www.digikey.com/eewiki/pages/viewpage.action?pageId=70189075>

http://www.burtonsys.com/ps2_chapweske.htm

<https://hw-server.com/connection-over-ps2-port>

https://www.youtube.com/watch?v=A1YSbLnm4_o&t=111s&ab_channel=LBEbooks

<https://silo.tips/download/ps2-mouse-interface-using-fpga>

http://radio.ubm.ro/EA/Documente/Cursuri_Laboratoare/Sisteme%20dedicate/carte.pdf