Truța Andrei

GRUPA 30211   
ÎNDRUMĂTOR: ING. DIANA POP

Aplicație FPGA și MOUSE

Proiect PSN

[1. Considerații teoretice 3](#_Toc136267926)

[1.1. Specificații 3](#_Toc136267927)

[1.2. Protocol PS2 3](#_Toc136267928)

[2. Proiectare 5](#_Toc136267929)

[2.1. Schema bloc 5](#_Toc136267930)

[2.2. Unitatea de control și unitatea de execuție 6](#_Toc136267931)

[2.3. Organigrama UC 7](#_Toc136267932)

[2.4. Schema de implementare 8](#_Toc136267933)

[2.5. Resurse 8](#_Toc136267934)

[2.5.1. Click Counter 8](#_Toc136267935)

[2.5.2. SSD Manager 9](#_Toc136267936)

[2.5.2.1. Frequency Divider 10](#_Toc136267937)

[2.5.2.2. Ring Shifter 11](#_Toc136267938)

[2.5.2.3. Priority Decoder 11](#_Toc136267939)

[2.5.2.4. Digit Splitter 12](#_Toc136267940)

[2.5.2.5. MUX 4:1 12](#_Toc136267941)

[2.5.2.6. Convertor BCD-7SEG 13](#_Toc136267942)

[3. Justificarea deciziilor 13](#_Toc136267943)

[4. Posibilități de dezvoltare 13](#_Toc136267944)

[6. Bibliografie 14](#_Toc136267945)

# Considerații teoretice

## 1.1. Specificații

Să se implementeze o aplicație care permite utilizatorului contorizarea numărului de click-uri ale mouse-ului.

Cerințe funcționale:

1. Existența unui buton de RESET care va goli afișajul SSD (starea ’’0000’’)
2. Starea curentă este afișată pe SSD
3. La acționarea butonului stânga al mouse-ului starea curentă se va incrementa
4. La acționarea butonului dreapta al mouse-ului starea curentă se va decrementa
5. Un LED IS\_LEFT este aprins, marcând faptul că butonul stâng incrementează iar butonul drept decrementează
6. Un buton/switch REVERSE care este folosit pentru a inversa funcțiile mouse-ului; la acționare, LED-ul IS\_LEFT se va stinge

Cerințe non-funcționale:

1. Implementare pe plăcuță
2. Utilizare SSD
3. Utilizare LED, switch/butoane
4. Utilizare mouse

## 1.2. Protocol PS2

Interfața PS/2, folosită de multe dintre mouse-urile și tastaturile moderne, a fost dezvoltată de IBM și a apărut inițial în Manualul de Referință Tehnică IBM.

Mouse-ul și tastatura PS/2 implementează un protocol serial sincron bidirecțional. Magistrala este "inactivă" când ambele linii sunt pe High. Aceasta este singura stare în care tastatura/mouse-ul poate începe să transmită date. Host-ul are controlul absolut asupra magistralei şi poate inhiba comunicarea în orice moment trăgând linia Clock la nivel scăzut.

Dispozitivul generează întotdeauna semnalul de ceas. Dacă host-ul dorește să trimită date, trebuie mai întâi să inhibe comunicarea de la dispozitiv setând Clock-ul pe Low. Host-ul trage apoi linia de date la nivel scăzut și eliberează Clock-ul. Aceasta este starea "Request-to-Send" și semnalează dispozitivul să înceapă să genereze impulsuri de ceas.

Toate datele sunt transmise in mod serial, câte un bit pe rând, iar fiecare byte este trimis într-un cadru format din 11-12 biți. Acești biți sunt:

1. 1 bit de start; este întotdeauna 0
2. 8 biți de date; cel mai puțin semnificativ este primul
3. 1 bit de paritate; folosește logica parității impare
4. 1 bit de stop; este întotdeauna 1
5. 1 bit de confirmare; numai in comunicarea Host to Device

Datele trimise de dispozitiv către gazdă sunt citite pe frontul descendent al clock-ului; datele trimise de gazdă către dispozitiv sunt citite pe frontul ascendent. Frecvența ceasului trebuie să fie în intervalul 10kHz – 16.7kHz. Aceasta înseamnă că semnalul de clock trebuie să fie High timp de 30 - 50 de microsecunde și Low timp de 30 - 50 de microsecunde.

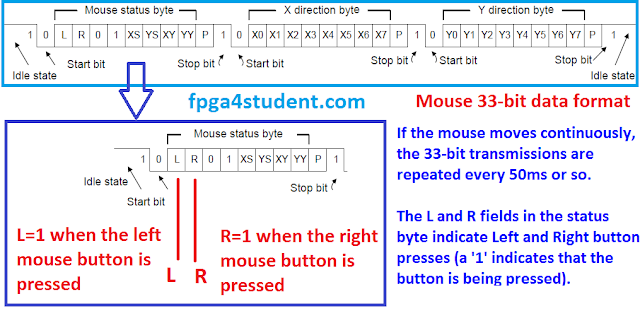
A row of black hexagons

Description automatically generated with low confidence

Figura : Un pachet de un byte transmis Host to Device

În cazul comunicării cu un mouse simplu, sunt necesari 3 bytes de date pentru comunicarea tuturor informațiilor necesare. Primul byte conține informații referitoare la starea butoanelor (buton stânga apăsat, buton dreapta apăsat, etc.), al doilea byte conține date despre mișcarea mouse-ului pe axa X și al treilea byte, date despre mișcarea pe axa Y. Totalul de biți necesari la citire, împreună cu cei de start/stop/paritate, devine 33.

În cadrul acestui proiect am folosit un mouse Logitech G203 care este dotat cu 5 butoane, lucru care necesită o suplimentare cu un byte la pachetul complet de informații, aducând totalul la 43 de biți (1 bit de start, 8 biți de date, 1 bit de stop).





O extensie populară pentru mouse-ul standard PS/2 este Microsoft Intellimouse. Aceasta include suport pentru un total de cinci butoane de mouse și trei axe de mișcare (dreapta-stânga, sus-jos și o roată de scroll). Aceste caracteristici suplimentare necesită utilizarea unui pachet de date de mișcare de 4 octeți în loc de pachetul standard de 3 octeți.

# 2. Proiectare

## 2.1. Schema bloc

A picture containing text, screenshot, font, line

Description automatically generated

Figura 1: Blackbox-ul sistemului cu intrările și ieșirile stabilite

## 2.2. Unitatea de control și unitatea de execuție

A picture containing diagram, text, plan, technical drawing

Description automatically generated

Figura 2: Map-ul intrărilor și ieșirilor cutiei negre după prima descompunere

## 2.3. Organigrama UC

A diagram of a flowchart

Description automatically generated with medium confidence

Figura 3: Organigrama UC

## 2.4. Schema de implementare

A picture containing diagram, text, plan, technical drawing

Description automatically generated

Figura 4: Schema de implementare a proiectului

## 2.5. Resurse

Urmează o prezentare detaliată a resurselor folosite în cadrul acestui proiect, inclusiv modul în care acestea sunt implementate. Vom explora modul în care aceste resurse sunt definite și utilizate pentru a crea funcționalitatea dorită a sistemului.

### 2.5.1. Click Counter

A picture containing text, font, screenshot, line

Description automatically generated

Figura 5: Schema bloc a click counter-ului

Click counter-ul, după cum numele sugerează, reprezintă un numărător. Acesta se ocupă de stocarea și actualizarea stării curente a numărului de click-uri recepționate. Ca intrări acesta primește un semnal de ceas, fiind o componentă sincronă, un semnal de reset și două semnale de comandă: incrementare și decrementare. Modulul utilizează semnalul de ceas pentru a sincroniza operațiile de numărare; la fiecare front crescător valoarea internă este actualizată în funcție de semnalele de incrementare/decrementare. În cazul în care semnalul de reset este activ, valoarea numărătorului revine la 0.

Datorită limitării fizice a plăcuței Basys 3 care prezintă doar 4 afișoare, intervalul la care numărul de click-uri va fi restrâns este [0, 9999]. Astfel, dimensiunea output-ului este de 16biți, pentru a putea acomoda numerele de care este nevoie. Deși 14 biți ar fi suficienți, am ales o putere a lui 2 din motive subiective care țin de aspect.

### 2.5.2. SSD Manager

A white rectangular object with black text

Description automatically generated with low confidence

Figura 6: Schema bloc a managerului afisoarelor

Manager-ul de afișor (Seven Segment Display) este o componentă mai complexă din punct de vedere intern, dar care însă are un rol banal: de a pune pe afișoare numărul pe care îl primește ca input. Este bineînțeles o componentă sincronă care beneficiază de un semnal de reset, iar acțiunile acesteia au loc pe frontul ascendent al semnalului de ceas.

Datorită limitării hardware de accesare a afișoarelor pe un anod comun, este nevoie de 4 semnale de ceas pentru a afișa un număr întreg (câte un front pentru fiecare cifră).

Separarea cifrelor se face cu ajutorul unui algoritm simplu care constă în operații repetate de împărțire cu rest la numărul 10. Fiecărei cifre îi este atribuită un cod pentru catozi care este mai apoi transmis mai departe către afișor. Mai departe va fi descrisă în detaliu structura internă a resursei, care poate fi observată în schema de mai jos.

A diagram of a computer

Description automatically generated with low confidence

Figura 7: Structura internă a managerului de afișoare

#### 2.5.2.1. Frequency Divider

A white rectangular sign with black text

Description automatically generated with low confidence

În cadrul managerului de afișor, datorită faptului că lucrăm cu o frecvență de 100MHz a clock-ului intern de pe placuța Basys 3, vom avea nevoie de un divizor de frecvență deoarece o schimbare mult prea rapidă a afișoarelor va duce la o luminozitate redusă și la neclaritatea cifrelor care, pentru ochiul nostru, vor deveni suprapuse și indescifrabile. Intern, divizorul de frecvență constă într-un numărător sincron cu frontul crescător al intrării de clock. Intrarea de Reset este asincronă și odată adusă la High, va reseta numărătoarea.

Dorind să reducem suficient de mult frecvența astfel încât să fie adecvată pentru ochiul uman, am ajuns prin experimentare la o frecvență de aproximativ 1.5kHz pentru divizor. Aceasta este realizată prin numărarea pană la 65535, iar pentru a menține factorul de umplere de 50% a semnalului, ieșirea este legată la bit-ul 16 al numărului în reprezentarea sa binară. Umplerea este asigurată datorită proprietăților numerelor binare.

#### 2.5.2.2. Ring Shifter

A white rectangle with black text

Description automatically generated with low confidence

Pentru a selecta cifra care urmează sa fie afișată pe afișorul ei corespunzător folosim un registru de deplasare în inel cu valoarea internă inițială 1 (’’0001’’). La fiecare impuls, bit-ul ’’1’’ se va deplasa pe următoarea poziție, ciclic, parcurgând astfel fiecare index de cifră. Aducerea intrării asincrone de reset la High va aduce registrul la starea inițială 1 (’’0001’’). Alegerea de a folosi un registru de 4 biți se datorează limitării hardware a plăcuței Basys 3, care beneficiază doar de 4 afișoare.

#### 2.5.2.3. Priority Decoder

A picture containing text, font, screenshot, line

Description automatically generated

Aceasta este o componentă asincronă cu o singură intrare și o singură ieșire. Decodificatorul prioritar ne va da index-ul bitului ’’1’’ de pe intrarea RAW care provine de pe ieșirea registrului de deplasare. Este pe 4 biți și prioritizează valorile în sensul acelor de ceasornic; Valoarea ’’0100’’ va avea același rezultat ca și valoarea ’’0111’’, adică valoarea 2 (însă nu este cazul în sistemul nostru ca o valoare cu mai mult de un singur bit de ’’1’’ sa ajungă aici). Astfel vom putea determina care din cifrele numărului trebuie selectată pentru a fi afișată. Rezultatul acestei operații va deveni semnal de selecție pentru un multiplexor.

#### 2.5.2.4. Digit Splitter

A white rectangle with black text

Description automatically generated with medium confidence

Această componentă asincronă are rolul de a desface numărul dat în 4 cifre. Această operație se realizează aritmetic și are ca scop pregătirea numărului pentru a fi afișat pe afișoare. Operațiile folosite constă în împărțiri cu rest repetate, folosindu-ne de algoritmul clasic întâlnit in programare de inversare a unui număr. Prin împărțirea cu rest la 10 a unui număr putem păstra ultima cifră a acestuia în același timp în care o eliminăm din număr. Putem privi această eliminare ca pe o deplasare a stânga in baza 10 a numărului.

#### 2.5.2.5. MUX 4:1

A picture containing text, diagram, line, screenshot

Description automatically generated

Această componentă asincronă reprezintă un multiplexor 4:1. Are ca selecție rezultatul decodificatorului prioritar și ca intrări, cele 4 cifre ale numărului. În funcție de semnalul de selecție, una dintre cifre va fi aleasă.

#### 2.5.2.6. Convertor BCD-7SEG

A white rectangle with black text

Description automatically generated with medium confidence

Ultimul pas în acest sistem al managerului SSD este convertirea cifrei a cărei rând este de a fi afișată în codul corespunzător 7SEG. Nu există logică complexă în spatele operației, iar componenta este una asincronă. Folosindu-ne de descrierea comportamentală am implementat tabelul de decodificare BCD -> 7SEG.

# 3. Justificarea deciziilor

Totalitatea deciziilor luate au avut ca sursă limitări hardware. Pentru început, orice registru care intră în componența SSD Manager are doar 4 biți datorită faptului că plăcuța Basys 3 beneficiază doar de 4 afișoare.

În componenta PS2 READER registrul este de 43 de biți. Această decizie a fost influențată de mouse-ul folosit pentru proiect, Logitech G203. Necesitatea celor 43 de biți provine din faptul că mouse-ul beneficiază de 5 butoane în loc de cele 3 clasice, astfel intervine nevoia a încă 8 biți de date însoțiți de biții de start și paritate, adăugându-se astfel încă 10 biți la cei 33 care sunt specificați în protocol.

# 4. Posibilități de dezvoltare

În etapa curentă, proiectul verifică și interpretează doar 2 biți din tot pachetul de date transmis: cel pentru buton stânga si cel pentru buton dreapta. Totuși, este citit un pachet întreg, care ar putea fi mai apoi validat și folosit pentru a extrage și alte date, precum starea curentă a altor butoane sau mișcările mouse-ului.

Cea mai directă și ușoară posibilitate de dezvoltare este cea de a îmbunătăți resursa PS2 READER pentru a interpreta întreg pachetul de date primit. Această modificare reprezintă pragul tuturor celorlalte posibilități de dezvoltare. Odată ce avem acces la totalitatea informației transmise de mouse, vom putea extinde funcționalitatea proiectului.

O altă cale ar fi implementarea unui modul care să se ocupe de transmis date către host. Astfel, am putea sa interceptăm pachetele de date provenite de la mouse și să le alterăm în drumul lor către host.

# 6. Bibliografie

1. PS2 Controller - <https://www.eecg.utoronto.ca/~jayar/ece241_08F/AudioVideoCores/ps2/ps2.html>
2. PS2 Protocol - <http://pcbheaven.com/wikipages/The_PS2_protocol/>
3. Basys 3 FPGA Board Reference Manual - <https://digilent.com/reference/_media/basys3:basys3_rm.pdf>
4. The PS2 Mouse Interface - <http://users.utcluj.ro/~baruch/media/sie/labor/PS2/PS-2_Mouse_Interface.htm>