



SISTEM NUMERIC PE FPGA CU MOUSE

Proiect PSN



PĂUCEAN IOANA
GRUPA: 30211
ÎNDRUMĂTOR: ING. DIANA POP

CUPRINS

1.Specificația proiectului	2
2. Proiectare.....	3
2.1 Schema Bloc	3
2.2 Unitatea de Control și Unitatea de Execuție.....	4
2.2.1 Maparea intrărilor și ieseirilor cutiei mari pe cele două componente UC și UE	4
2.2.2 Determinarea resurselor (UE)	5
2.2.3 Schema bloc a primei descompuneri	9
2.2.4 Reprezentarea UC prin diagramă de stări (organigramă)	10
2.2.5 Schema de detaliu a proiectului	12
3. Manual de utilizare și întreținere	13
4. Justificarea soluției alese.....	22
5. Posibilități de dezvoltări ulterioare	23
6. Bibliografie	24

1.SPECIFICAȚIA PROIECTULUI

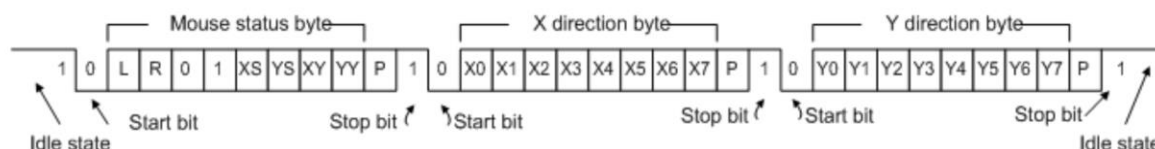
Acest proiect implementează pe FPGA un sistem numeric care permite utilizatorului contorizarea numărului de click-uri ale mouse-ului. Sistemul va afișa numărul de click-uri pe cele 4 display-uri cu 7 segmente (SSD) ale plăcuței. De asemenea incrementarea, decrementarea, reversibilitatea și resetarea numărului afișat sunt permise prin diferite acțiuni ale mouse-ului și ale unor butoane/switch-uri.

Functionarea sistemului:

- Starea inițială este “0000”, această stare indică faptul că nu a fost apăsat niciun click al mouse-ului
- Starea curentă a numărului de click-uri este afișată în timp real pe SSD
- La acționarea butonului stânga al mouse-ului starea curentă se incrementează cu 1
- La acționarea butonului dreapta al mouse-ului starea curentă se decrementează cu 1
- Led-ul “is_left” este aprins pentru a indica faptul că un click stânga incrementează, iar un click dreapta decrementează
- Switch-ul “reverse” permite inversarea funcțiilor butoanelor mouse-ului. La acționarea acestui switch led-ul “is_left” se va stinge, indicând faptul că funcțiile butoanelor sunt inversate (click stânga decrementează, click dreapta incrementează)
- Dacă starea curentă a ajuns la “9999” sistemul va merge înapoi în starea inițială “0000”
- Dacă sistemul se află în starea “0000” și primește comanda să se decrementeze acesta va sări la starea “9999”

Comunicarea mouse-ului cu port-ul USB folosind protocolul PS/2

- Mouse-ul trimite date în pachete de 33 de biți
- Fiecare pachet este compus din trei cuvinte de 11 biți, fiecare conținând:
 - Un bit de start ‘0’
 - 8 biți de date
 - Un bit de paritate impar
 - Un bit de stop ‘1’



Odată ce mouse-ul este introdus în modul de transmitere a datelor și raportarea datelor a fost activată, mouse-ul emite un semnal de ceas și un semnal de date atunci când este mișcat. În caz contrar, aceste semnale rămân la logică '1'. De fiecare dată când mouse-ul este mișcat, trei cuvinte de 11 biți sunt trimise de la mouse la dispozitivul gazdă, așa cum este

arătat în desenul de mai sus. Fiecare dintre cele trei cuvinte de 11 biți conține un bit de start '0', urmat de 8 biți de date (LSB primul), urmat de un bit de paritate impară și terminat cu un bit de stop '1'. Astfel, fiecare transmisie de date conține 33 de biți, unde biții 0, 11 și 22 sunt biți de start '0', iar biții 11, 21 și 33 sunt biți de stop '1'. Cele trei câmpuri de date de 8 biți conțin datele de mișcare, așa cum este arătat în figura de mai jos. Datele sunt valide la marginea descendentă a ceasului, iar perioada ceasului este de 20 până la 30KHz.

Mouse-ul utilizează un sistem de coordonate relative în care mișcarea mouse-ului spre dreapta generează un număr pozitiv în câmpul X, iar mișcarea spre stânga generează un număr negativ. De asemenea, mișcarea mouse-ului în sus generează un număr pozitiv în câmpul Y, iar mișcarea în jos reprezintă un număr negativ (biții XS și YS din byte-ul de status sunt biți de semn – un '1' indică un număr negativ). Magnitudinea numerelor X și Y reprezintă viteza mișcării mouse-ului – cu cât numărul este mai mare, cu atât mouse-ul se mișcă mai repede (biții XV și YV din byte-ul de status sunt indicatori de depășire a mișcării – un '1' înseamnă că a avut loc o depășire). Dacă mouse-ul se mișcă continuu, transmisiile de 33 de biți se repetă aproximativ la fiecare 50 ms. Câmpurile L și R din byte-ul de status indică apăsarea butoanelor Stânga și Dreapta (un '1' indică faptul că butonul este apăsător).

Pentru a realiza comunicarea mouse-ului cu port-ul USB a fost nevoie de:

- Un divizor de frecvență care creează un semnal de ceas mai lent pentru debouncing
- Un debouncing care asigură stabilitatea semnalului de ceas USB
- O mașină de stare finită pentru citirea și parcurgerea datelor de la mouse
 - Starea 0: așteaptă bitul de start '0'
 - Starea 1: citește cei 33 de biți de date
 - Starea 2: emite date valide și așteaptă următorul bit de start

2. PROIECTARE

2.1 SCHEMA BLOC

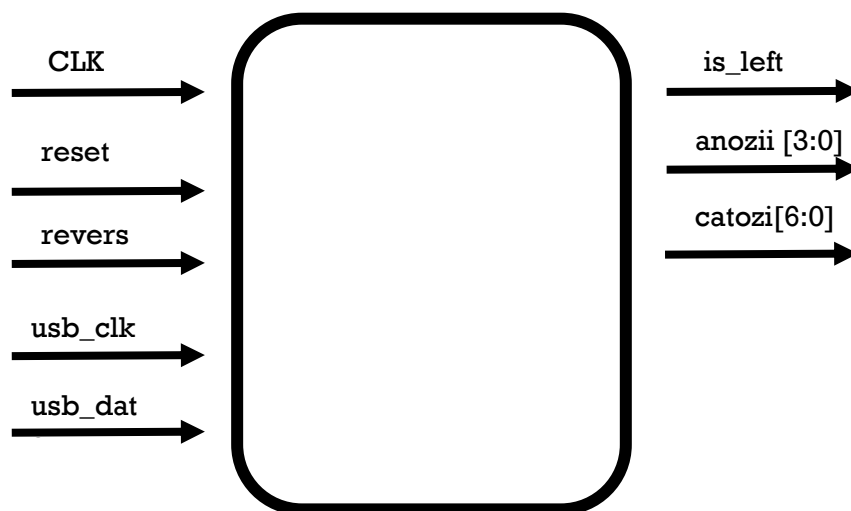


FIG 1: cutia neagra a sistemului

NUME	MODUL PORTULUI	TIPUL	ATRIBUIRE
clk	in	std_logic	semnalul de ceas al sistemului
reset	in	std_logic	resetarea sistemului
revers	in	srd_logic	inversarea controalelor mouse-ului
usb_clk	in	std_logic	semnalul de ceas primit prin USB de la mouse
usb_data	in	std_logic	datele primite prin USB de la mouse
is_left	out	std_logic	marcheaza modul de funcționare al butoanelor de la mouse
anozii	out	std_logic_vector (3 downto 0)	semnale folosite pentru activarea anozilor
catozi	out	std_logic_vector (3 downto 0)	semnale folosite pentru a afisa cifre pe SSD

2.2 UNITATEA DE CONTROL ȘI UNITATEA DE EXECUȚIE

Unitatea de control: responsabilă pentru coordonarea și gestionarea funcționării sistemului, interpretează semnalele de la mouse și inițiază diferite acțiuni

Unitatea de execuție: efectuează operațiile efective cerute de unitatea de control, cum ar fi contorizarea click-urilor, stocarea valorilor și afișarea rezultatelor

2.2.1 MAPAREA INTRĂRILOR ȘI IESIRILOR CUTIEI MARI PE CELE DOUĂ COMPONENTE UC ȘI UE

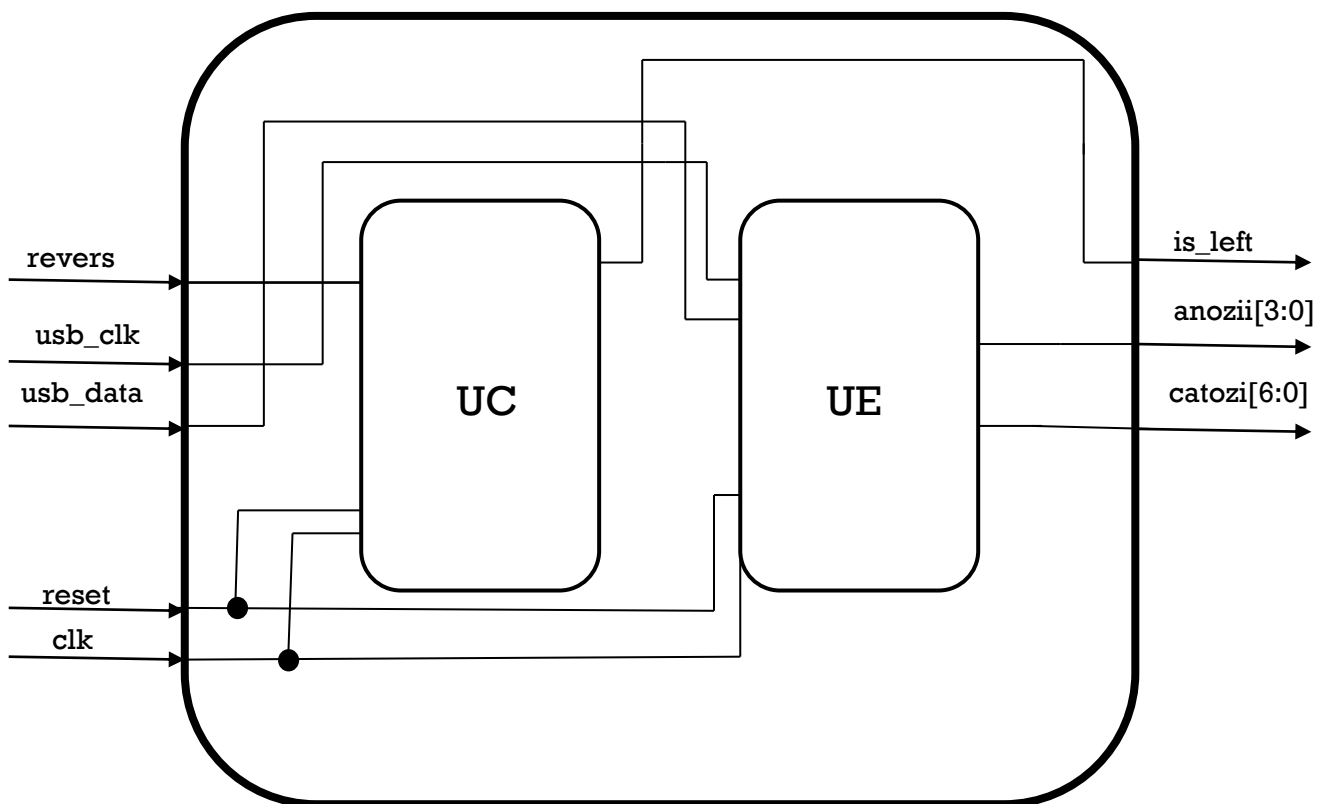


FIG 2: Maparea intrărilor și ieseirilor cutiei negre pe intrările și ieșirile comportamentelor din prima descompunere

Atât intrările cât și ieșirile sunt împărțite în 2 categorii de date: *de date* și *de control*

- **intrări de date:** click-urile mouse-ului
- **intrări de control:** buton de reset, switch de revers
- **ieșiri de date:** valori de afișat pentru utilizator (numărul de click-uri)
- **ieșiri de control:** semnale de avertizare a utilizatorului (led)

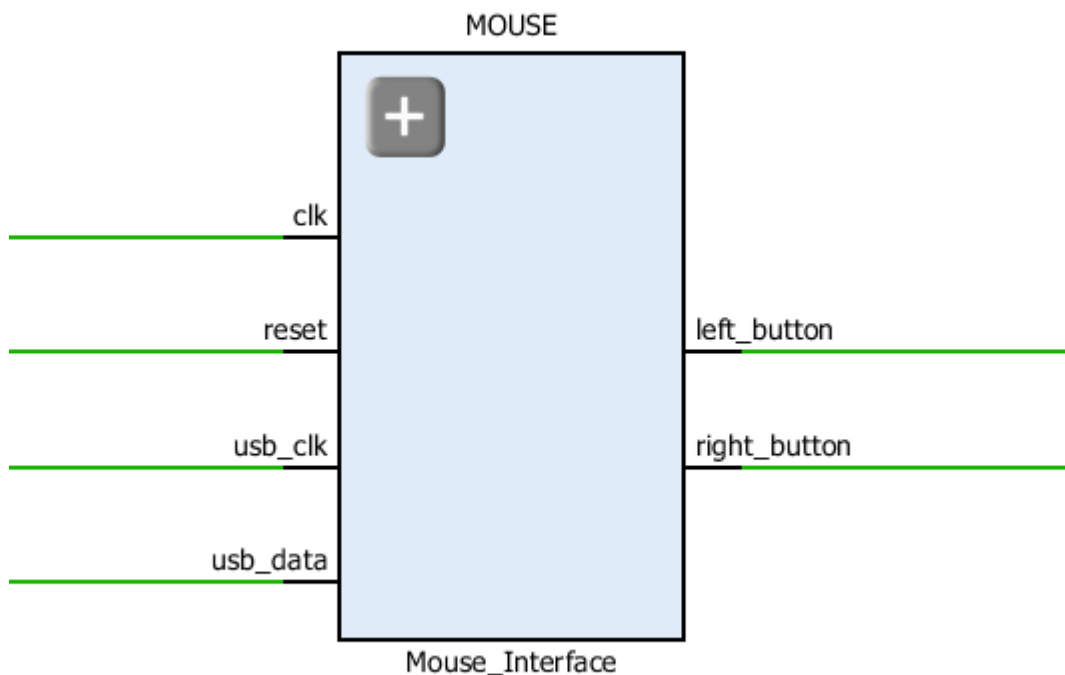
2.2.2 DETERMINAREA RESURSELOR (UE)

RESURSE:

1. Interfața de comunicare a mouse-ului cu plăcuța

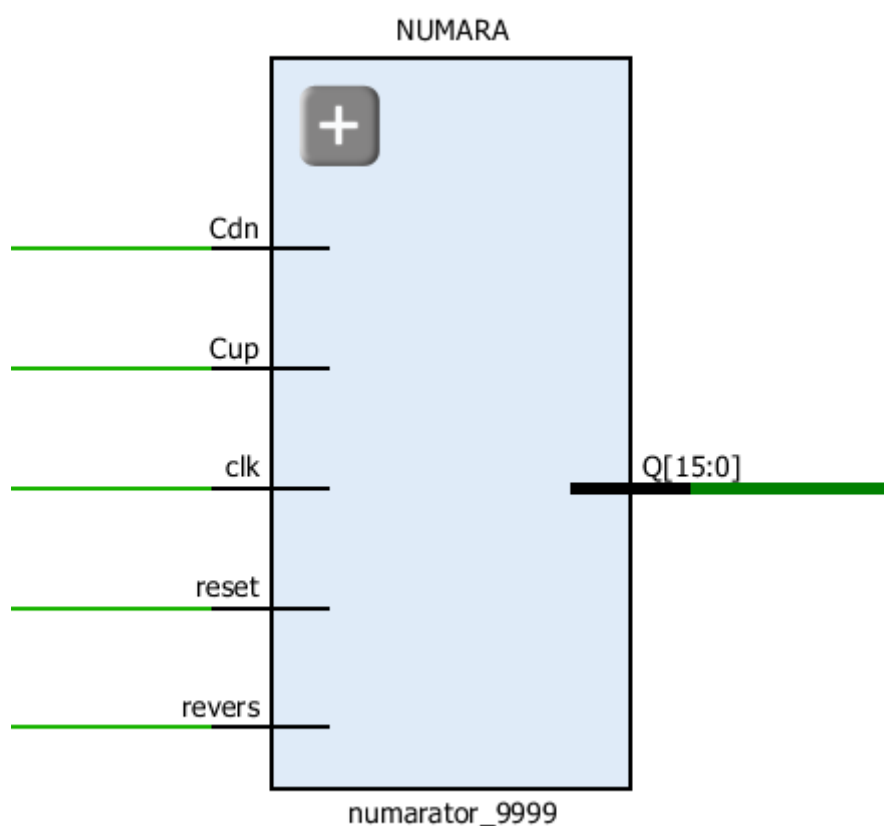
Această componentă implementează un divizor pentru frecvența ceasului ca să sincronizeze semnalul de clock al USB-ului cu al mouse-ului. De asemenea realizează debounce pentru semnalul USB CLK pentru eliminarea bouncing-ului. Citeste datele de la mouse, procesându-se secvențial fiecare bit și formându-se un cuvânt de 32 de biți și determină starea actuală a citirii datelor actualizând starea și variabilele corespunzătoare.

Prin intermediul acestei componente obținem semnale care marchează dacă butoanele mouse-ului au fost apăsate



2. Numatator de la 0 la 9999

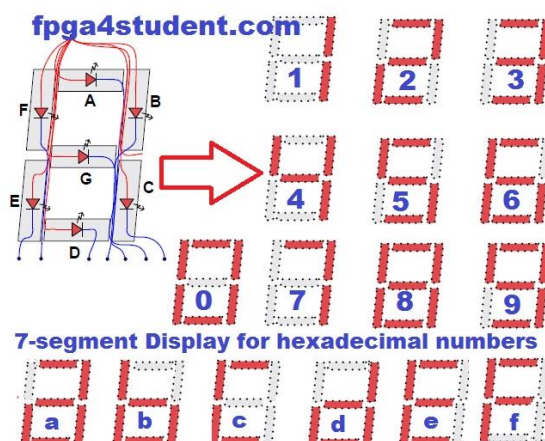
Această componentă este un numărător reversibil pe 16 biți care se resetează când ajunge la valoarea 9999. Acesta are două intrări Cup și Cdn care determină direcția de numărare (Cup incrementează și Cdn decrementează). Dacă intrarea de revers este activă pe '1' cele două intrări Cup și Cdn își inversează funcțiile (Cup decrementează și Cdn incrementează).



Folosind această componentă numărăm de câte ori au fost apăsată butoanele mouse-ului

3. Codificator BCD la SSD

Este un codificator care primește o cifră și o codifică astfel încât aceasta să poată fi afișată pe 7 segment display.



Tabelul de adevăr pentru afișarea pe SSD

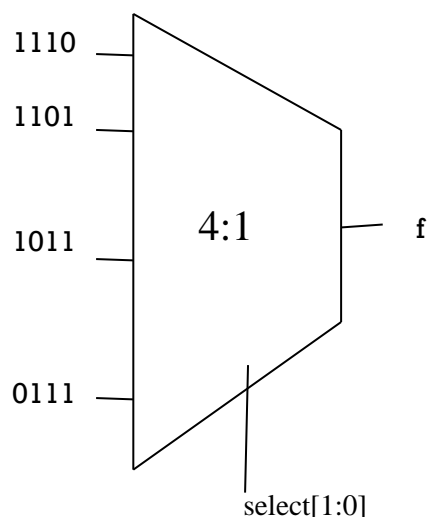
Hex Number	Common-Anode	CA	CB	CC	CD	CE	CF	CG	Cathode[6:0]
0	high	low	low	low	low	low	low	high	0000001
1	high	high	low	low	high	high	high	high	1001111
2	high	low	low	high	low	low	high	low	0010010
3	high	low	low	low	low	high	high	low	0000110
4	high	high	low	low	high	high	low	low	1001100
5	high	low	high	low	low	high	low	low	0100100
6	high	low	high	low	low	low	low	low	0100000
7	high	low	low	low	high	high	high	high	0001111
8	high	low	low	low	low	low	low	low	0000000
9	high	low	low	low	low	high	low	low	0000100
a	high	low	low	low	low	low	high	low	0000010
b	high	high	high	low	low	low	low	low	1100000
c	high	low	high	high	low	low	low	high	0110001
d	high	high	low	low	low	low	high	low	1000010
e	high	low	high	high	low	low	low	low	0110000
f	high	low	high	high	high	low	low	low	0111000

4. Multiplexor 4:1 pentru activarea anozilor

Fiecare anod al LED-urilor este conectat la o intrare diferită a mux-ului. Deci, anodul LED-ului 1 este conectat la intrarea 0 a mux-ului, anodul LED-ului 2 la intrarea 1, și așa mai departe.

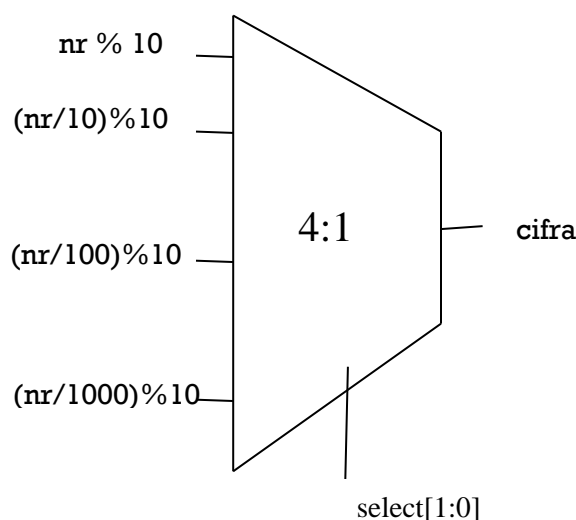
Un semnal de control cu două biți este utilizat pentru a selecta care dintre cele patru intrări ale mux-ului va fi transmisă pe ieșire. Acești doi biți pot fi furnizați de diverse surse, cum ar fi un comutator hardware, un contor sau chiar un alt mux.

Atunci când semnalul de control este setat într-o anumită configurație, mux-ul va selecta intrarea corespunzătoare și va transmite tensiunea de la acea intrare pe ieșire. Deci, dacă semnalul de control este 00, atunci ieșirea mux-ului va fi conectată la anodul LED-ului 1. Dacă semnalul de control este 01, atunci ieșirea mux-ului va fi conectată la anodul LED-ului 2, și așa mai departe.



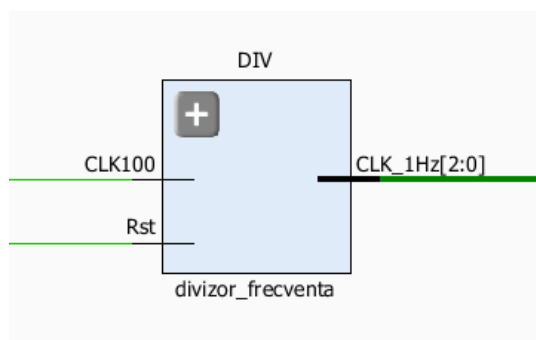
5. Multiplexor 4:1 pentru a prelucra cifrele numarului

Având în vedere anodul selectat trebuie aleasă și cifra corespunzătoare poziției.



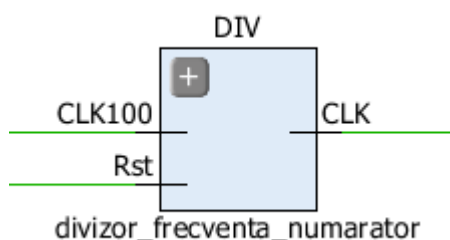
6. Divizor de frecventa pentru SSD

Când se dorește afișarea pe un display cu 7 segmente este necesară utilizarea unui divizor de frecvență pentru a asigura funcționarea corectă a afișajului și pentru a obține o afișare stabilă și clară. Fără un divizor de frecvență, semnalele ar putea fi prea rapide sau instabile, ceea ce ar face afișajul să pară tremurat sau să fie imposibil de citit.



7. Divizor frecvență pentru clock-ul numaratorului

Acesta este un divizor de frecvență care generează un clock de aproximativ 120Hz pentru a sincroniza momentul de apăsare a butonului de la mouse cu frecvența de numărare a Numărătorului.



2.2.3 SCHEMA BLOC A PRIMEI DESCOMPUNERI

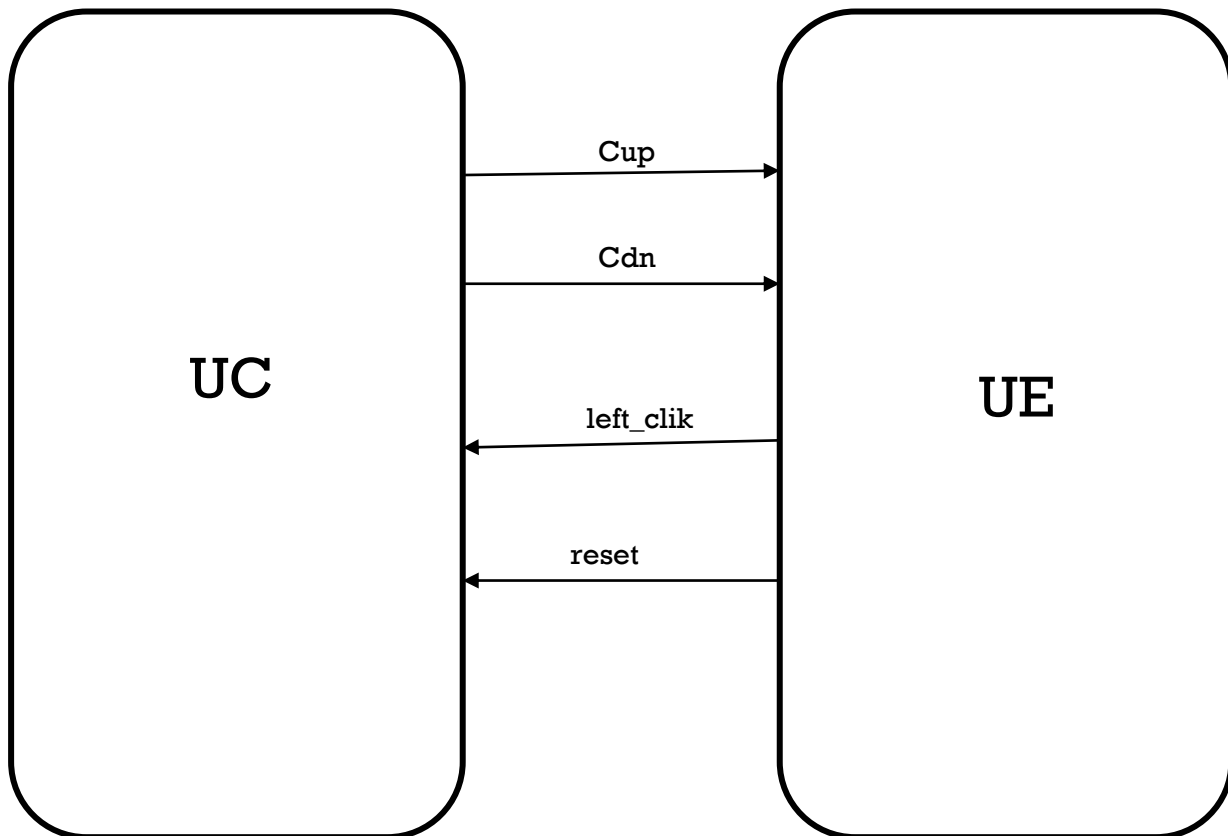


FIG 3: Shema Bloc cu legăturile dintre UC si UE mapate

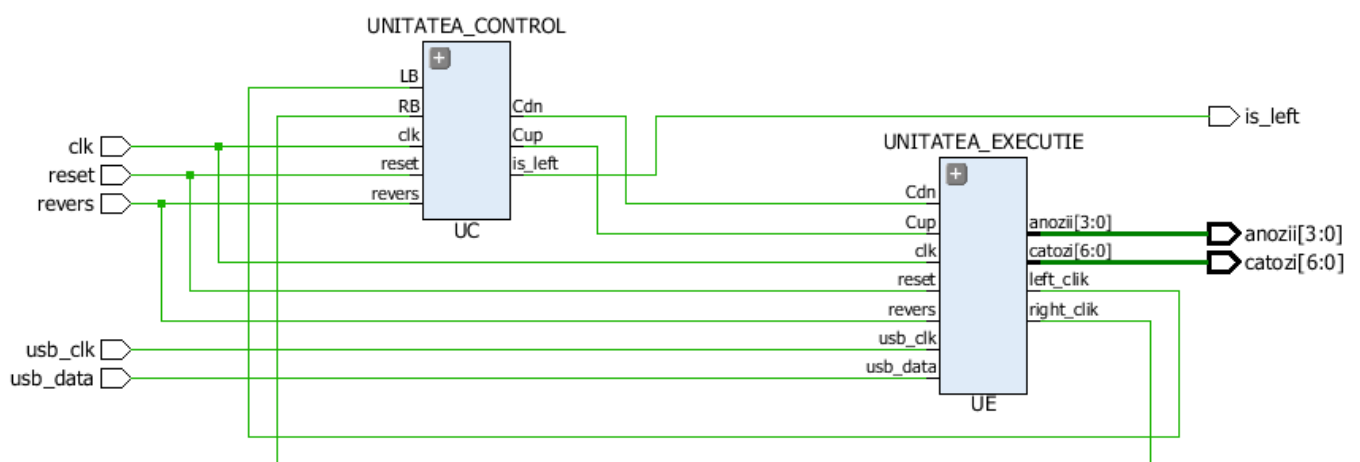
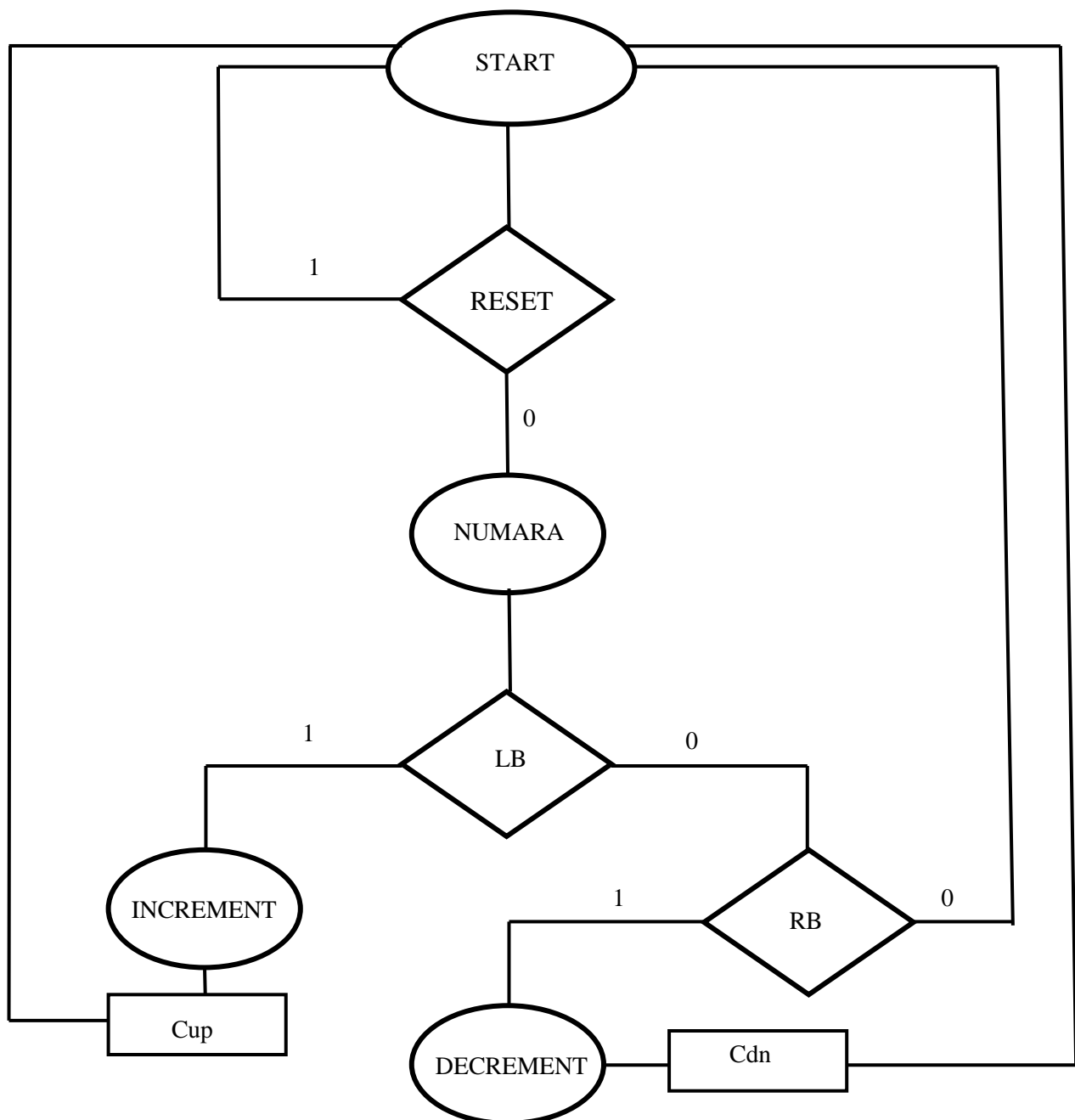


FIG 4: Shema Bloc cu legăturile dintre UC si UE mapate realizata de VIVADO

2.2.4 REPREZENATREA UC PRIN DIAGRAMĂ DE STĂRI (ORGANIGRAMĂ)

O organigramă este o reprezentare grafică a structurii unei organizații, arătând relațiile ierarhice și funcționale dintre diferitele departamente și poziții. Aceasta ilustrează cine raportează cui și cum sunt distribuite responsabilitățile în cadrul organizației, facilitând înțelegerea structurii și comunicării interne.

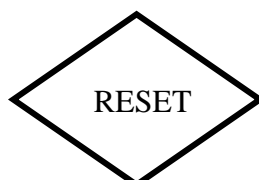


Această structură controlează un contor simplu care poate incrementa sau decrementa bazat pe intrările de la butoanele mouse-ului și condițiile de resetare.

- **STĂRILE** sunt reprezentate prin cerc. O stare reprezintă un moment de timp (o perioadă)



- **DECIZIILE** luate în fiecare stare sunt reprezentate prin romb



- **IEȘIRILE** generate în fiecare stare sunt reprezentate prin dreptunghi. În interiorul dreptunghiului se enumeră ieșirile care sunt adevărate în acel moment

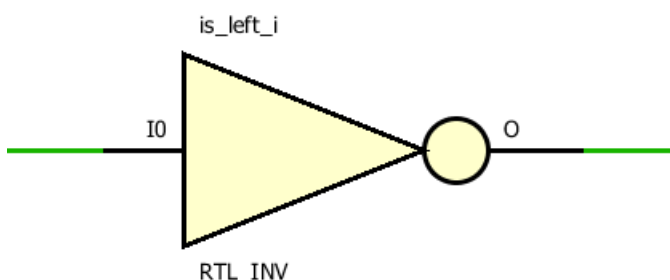


Organigrama modulului UC ilustrează un controler secvențial cu patru stări principale: start, numara, increment și decrement. Intrările principale sunt clk (ceas), reset, revers, LB (buton stânga mouse) și RB (buton dreapta mouse). Ieșirile sunt Cup (semnal de incrementare), Cdn (semnal de decrementare)

La fiecare front ascendent al semnalului clk, procesul sincronizat actualizează starea curentă (stare) la următoarea stare (next_stare). În starea start, atât Cup, cât și Cdn sunt setate la '0'. Dacă semnalul reset este activat, rămâne în starea start. În starea numara, dacă LB este activ, tranzizionează la starea increment; dacă RB este activ, tranzizionează la starea decrement. În starea increment, Cup este setat la '1' și Cdn la '0', apoi revine la start. În starea decrement, Cup este setat la '0' și Cdn la '1', apoi revine la start.

Inversor pentru semnalul “reverse”

Dacă led-ul “is_left” este aprins acesta indică faptul că butonul din stânga al mouse-ului incrementează, iar butonul din dreapta al mouse-ului decrementează, dacă este stîns înseamnă că atribuțiile butoanelor au fost inversate. Prin intermediul acestui inversor se stabilește dacă led-ul “is_left” este aprins, acesta fiind ‘1’ în cazul în care switch-ul de revers nu este activat și ‘0’ în caz contrar.



TABEL DE ADEVĂR	
revers	is_left
0	1
1	0

2.2.5 SCHEMA DE DETALIU A PROIECTULUI

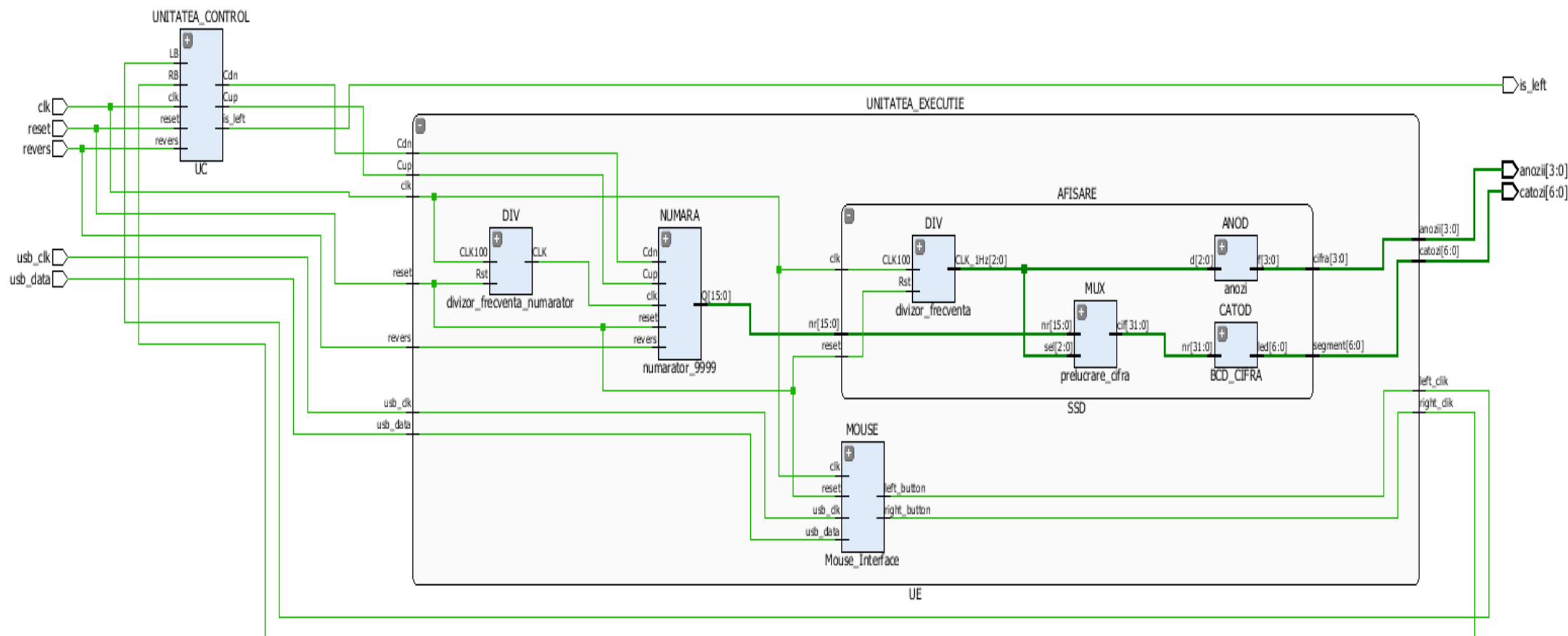


FIG 5: Schema de detalii a proiectului conform implementarii din Vivado

3. MANUAL DE UTILIZARE ȘI ÎNTRETINERE

Proiectul implementează pe FPGA un sistem numeric care permite utilizatorului contorizarea numărului de click-uri ale mouse-ului. Sistemul va afișa numărul de click-uri pe cele 4 display-uri cu 7 segmente (SSD) ale plăcuței. Pentru a utiliza urmăriți pașii de mai jos.

Materiale necesare:

- placă FPGA
- cablu micro USB
- mouse cu fir în 3 butoane

1. Asigurați-vă ca aveți instalat pe calculatorul dumneavoastră Vivado 2016.4 sau orice altă versiune mai recentă.

Tutorial de descărcare Vivado: <https://www.youtube.com/watch?v=Lc2EEbZmlz0>

Link spre site-ul celor de la AMD de unde puteți instala orice versiune de Vivado: <https://www.xilinx.com/support/download.html>

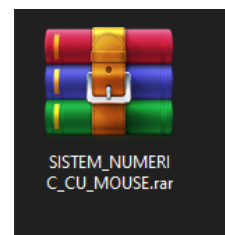
2. După ce ați instalat Vivado descărcați arhiva proiectului

Arhiva proiectului o descărcați de aici:

https://drive.google.com/file/d/1wRgoy97DM6sI4cW0oLSznB3Ja7ju7gKj/view?usp=drive_link

După descărcare ar trebui să aveți această arhivă pe calculatorul dumneavoastră, în caz contrar verificați dacă aveți instalat WinRAR

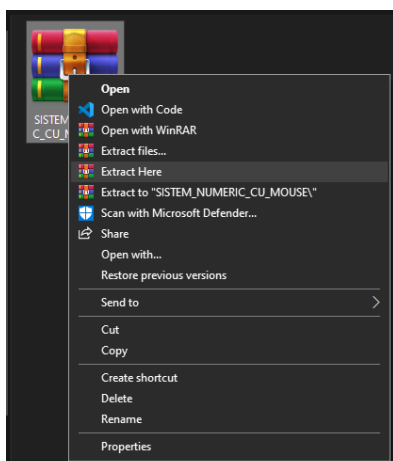
Link descărcare WinRAR: <https://www.win-rar.com/start.html?&L=0>



3. Dezarhivați arhiva pentru a obține materialele proiectului

Dezarhivarea se face apăsând click dreapta pe documentul ce urmează a fi dezarhivat, mai apoi alegeți una din opțiunile: Extract files..., Extract Here, Extract to ...

Dacă nu sunteți siguri ce opțiune să alegeți recomandăm "Extract Here"



4. Odată dezarhivat proiectul aveți următoarele fișiere din poza de mai jos, pentru a implementa pe placuța FPGA proiectul trebuie să dați dublu click pe “SISTEM_NUMERIC_CU_MOUSE.xpr”. Acesta va deschide în Vivado proiectul.

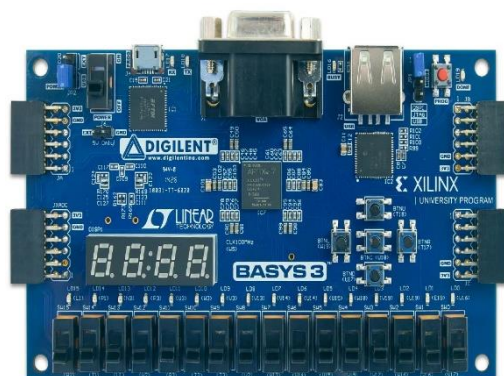
Name	Date modified	Type	Size
.Xil	02.06.2024 13:16	File folder	
SISTEM_NUMERIC_CU_MOUSE.cache	02.06.2024 13:15	File folder	
SISTEM_NUMERIC_CU_MOUSE.hw	02.06.2024 13:15	File folder	
SISTEM_NUMERIC_CU_MOUSE.ip_user_fil...	01.06.2024 21:23	File folder	
SISTEM_NUMERIC_CU_MOUSE.runs	02.06.2024 13:15	File folder	
SISTEM_NUMERIC_CU_MOUSE.sim	02.06.2024 13:15	File folder	
SISTEM_NUMERIC_CU_MOUSE.srscs	02.06.2024 13:15	File folder	
SISTEM_NUMERIC_CU_MOUSE.xpr	02.06.2024 01:06	Vivado Project File	9 KB
vivado.jou	Type: Vivado Project File Size: 8,42 KB	JOU File	2 KB
vivado.log	Date modified: 02.06.2024 01:06	Text Document	3 KB
vivado_8992.backup.jou	02.06.2024 12:51	JOU File	1 KB
vivado_8992.backup.log	02.06.2024 13:14	Text Document	2 KB

5. Conectare plăcă FPGA la calculator

Acum că avem proiectul deschis în Vivado trebuie să conectăm plăcuța la calculator.

ATENȚIE: proiectul are făcut fișierul de constrângeri pentru Basys3, în cazul în care folosiți un alt model de plăcuță FPGA ar trebui să modificați fișierul de constrângeri conform cerințelor plăcii dumneavoastră. Pentru a obține fișierul de constrângeri corespunzător plăcii consultați manualul producătorului. Pentru a vedea cum înlocuiți fișierul de constrângeri mergeți la pagina 18

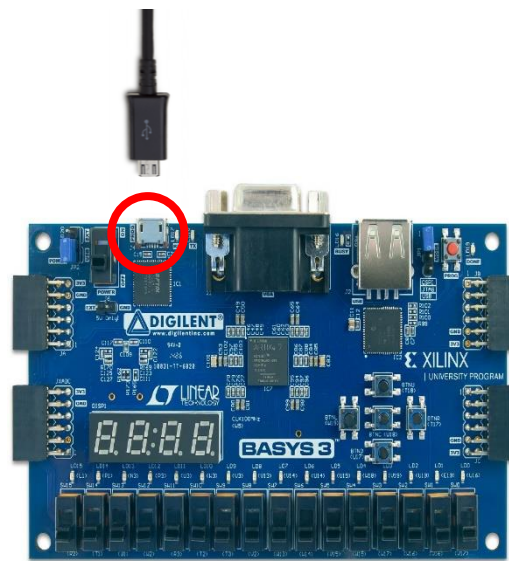
Dacă plăcuța dumneavoastră arată ca cea din imagine nu este necesar să schimbați nimic în proiect.



Plăcuța se conectează la calculator printr-un cablu micro USB care are ca cel din imaginea de mai jos



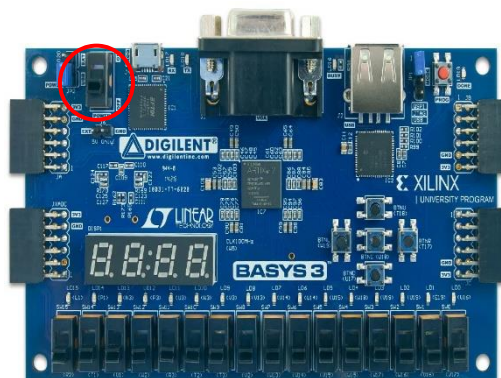
Cablurile sunt conectate astfel: micro USB la plăcuță și USB in calculator. Imaginea de mai jos vă arată exact care capăt al cablului trebuie conectat la plăcuță și care trebuie conectat la calculatorul dumneavoastră



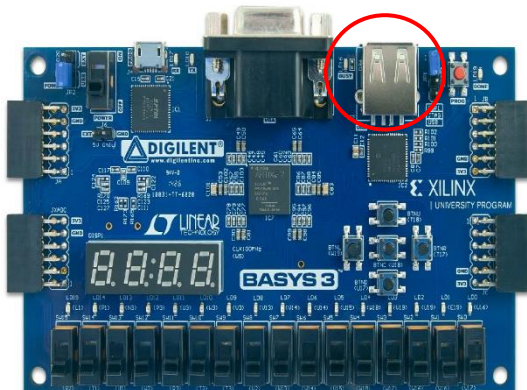
În calculatorul
dumneavoastră



Odată conectată la calculator plăcuța poate fi pornită prin switch-ul de on/off. După acționarea acestui switch ar trebui să observați ca cele 4 afișoare numără de la 0 la 9 cât și că led-ul de power este aprins. Switch-ul de on/off este cel încercuit în imaginea de mai jos.

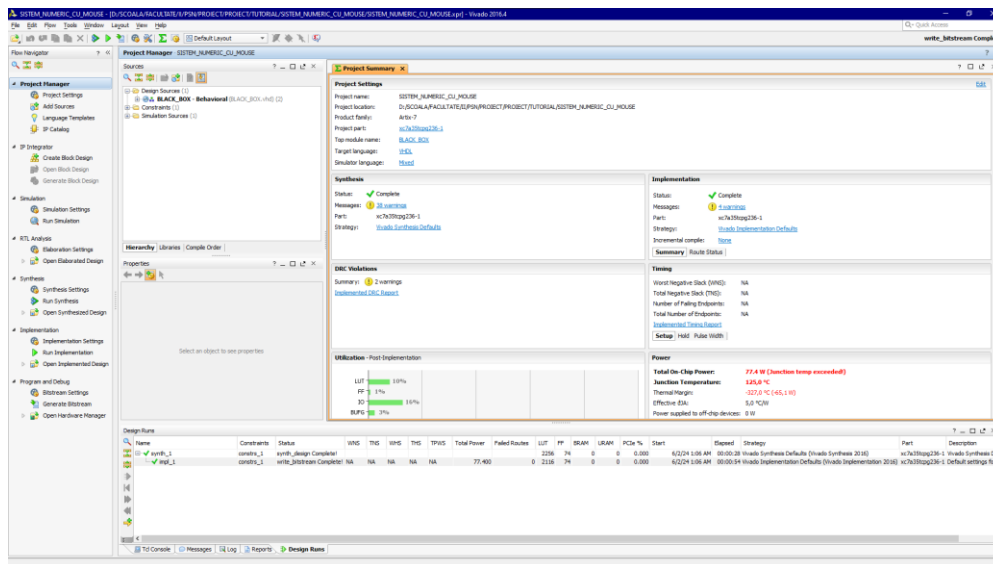


Pentru acest proiect va trebui să conectăm un mouse cu 3 butoane la portul USB al plăcuței

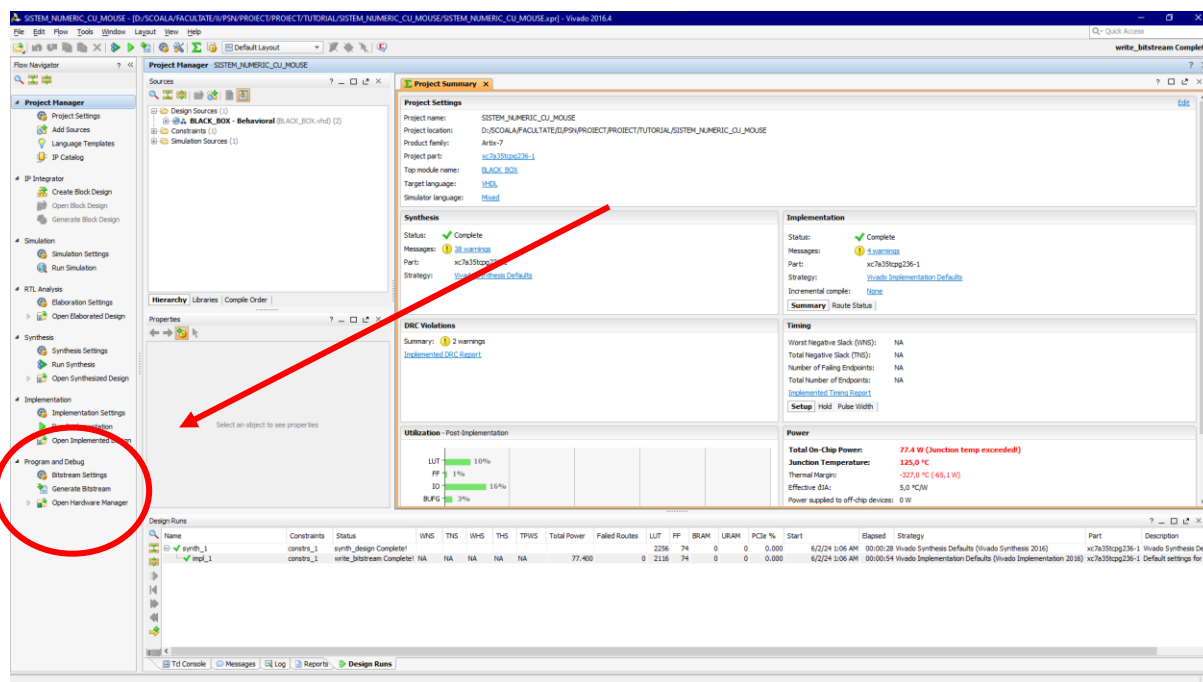


6. Încărcarea proiectului pe plăcuță

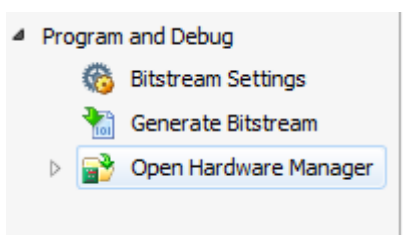
La pasul 4 am deschis proiectul în Vivado. Pe ecranul dumneavoastră ar trebui să aveți ceva asemănător cu imaginea de mai jos. Aspectul poate să difere dacă folosiți o altă versiune de Vivado



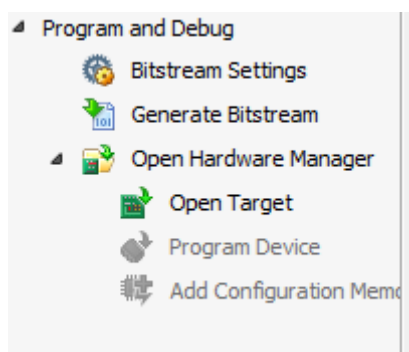
În partea stângă a ecranului este secțiunea “Program and Debug”.



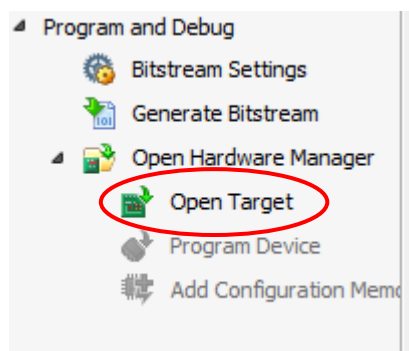
Ne interesează ultima componentă din această secțiune ci anume “Open Hardware Manager”



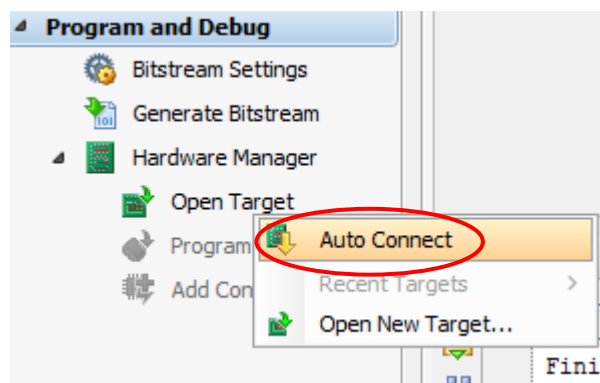
Dacă dați click pe acea componentă va vor apărea următoarele:



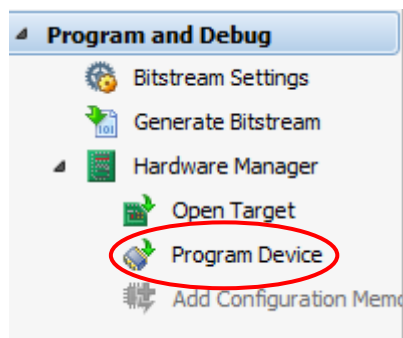
Pentru a conecta plăcuța la proiect apăsați “Open Target”



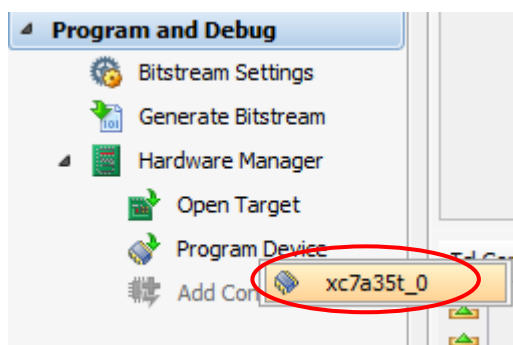
După apăsarea “Open Target” vi se vor afișa următoarele opțiuni dintre care selectați “Auto Connect”



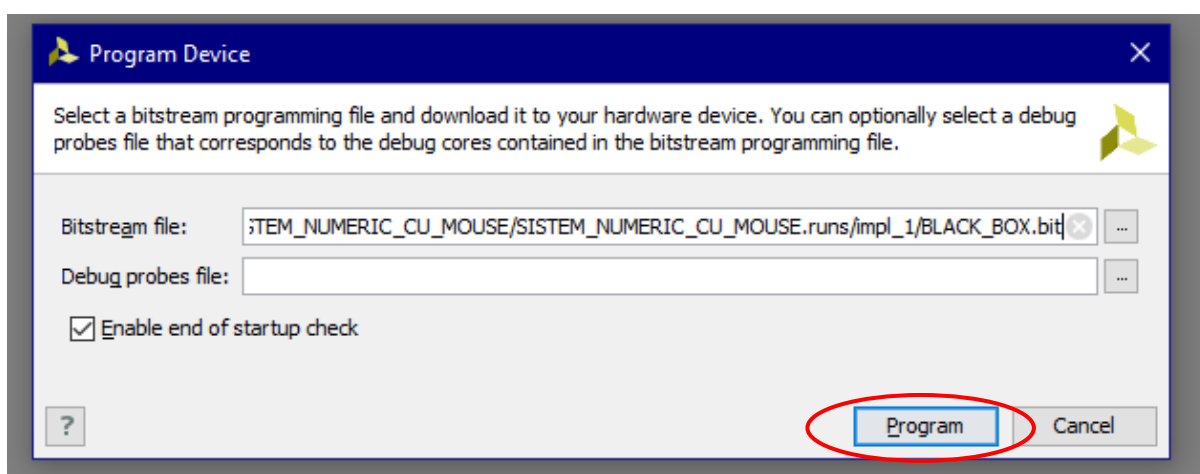
Acum, in secțiunea “Hardware Manager” a devenit disponibilă opțiunea de “Program Device” pe care o vom selecta



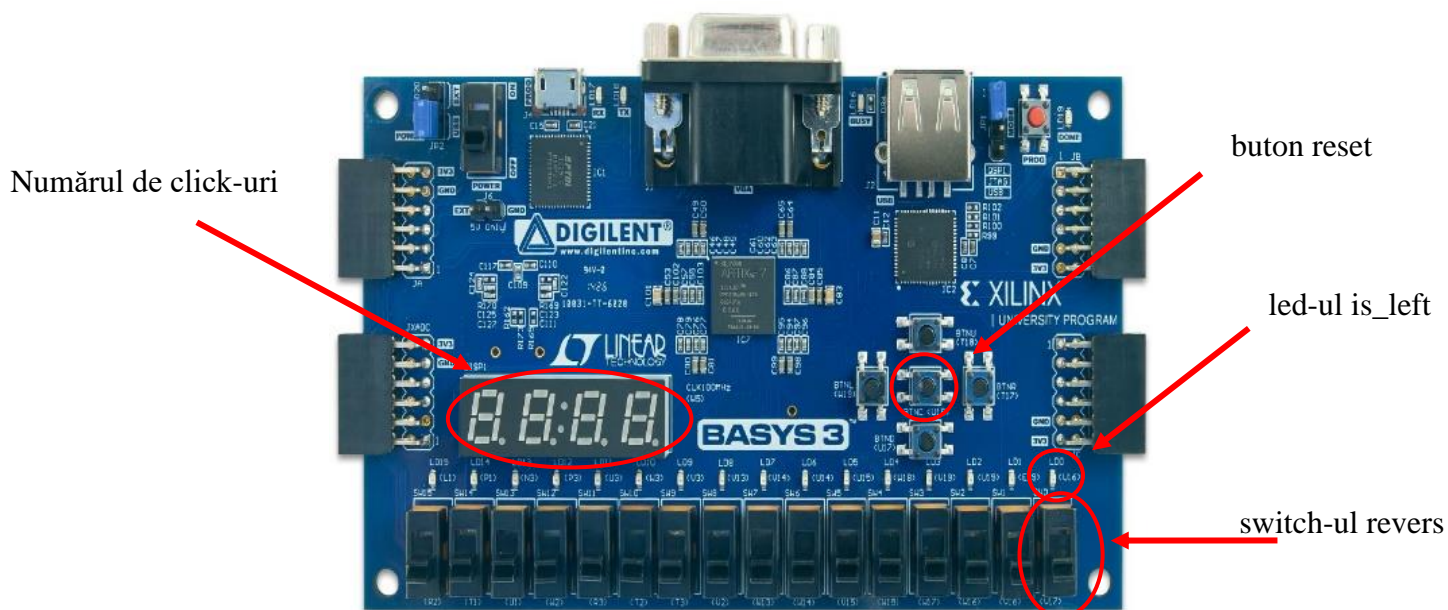
Dupa selectarea obțiunii “Program Device” va apărea încă o obțiune ce ar trebui să corespundă modelului de piesă al plăcii cu care dumneavoastră lucrați. Selectăm acest model. Dacă folosiți un alt model de placă mergeți la pagina 18



Pe ecran va apărea următoarea fereastră unde vom selecta “Program”



Dacă ați făcut fiecare pas corect acum ar trebui să aveți pe plăcuța dumneavoastră un sistem care contorizează click-urile mouse-ului conform cerinței prezentate anterior. Pe plăcuță ar trebui să vedeți următoarele:

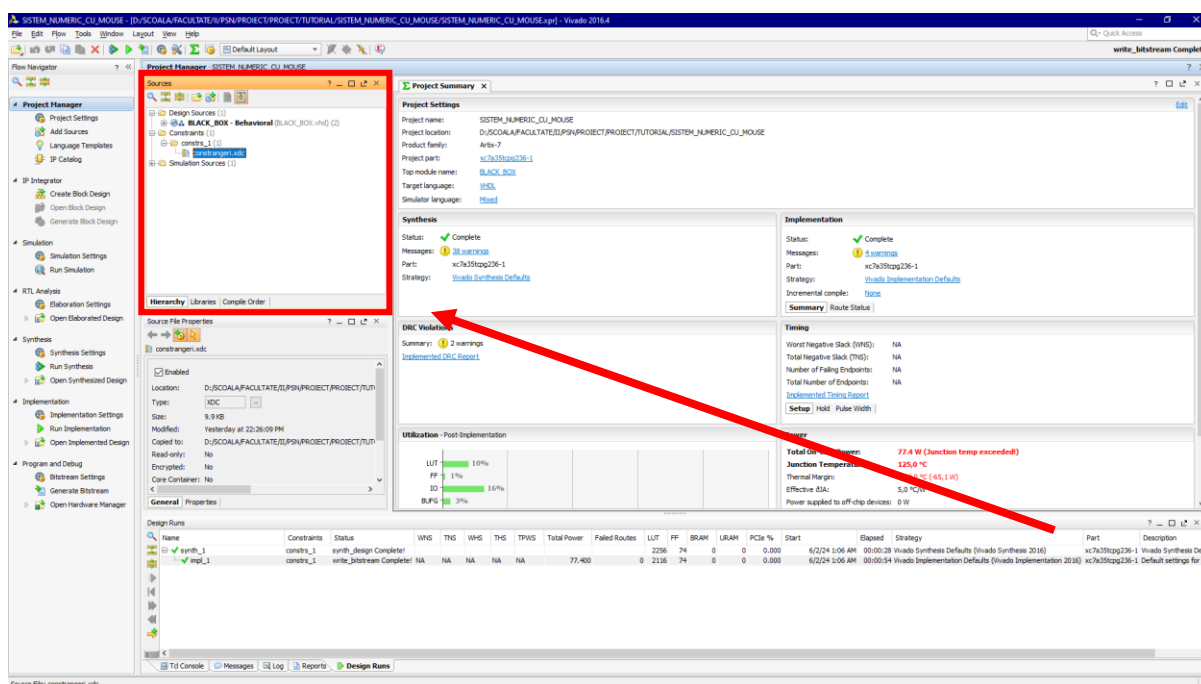


7. Nu folosiți același model de placă FPGA, ce trebuie să faceți:

Dacă nu folosiți un Basys3 sunt necesare următoarele modificări în proiect:

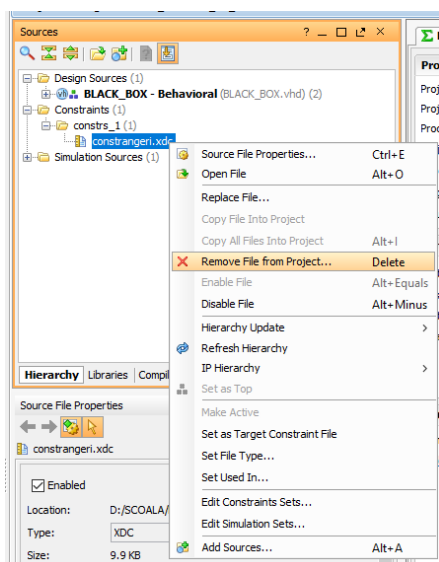
- ștergerea fișierului de constrângeri existent și crearea unui nou ce corespunde plăcii dumneavoastră
- selectarea modelului de placă corespunzător

1) Ștergerea fișierului de constrângeri existent și crearea unui nou ce corespunde plăcii dumneavoastră

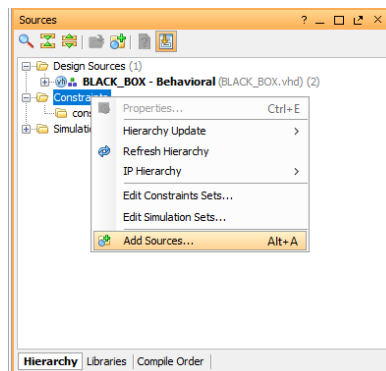


În secțiunea de “Sources” a proiectului apar toate fișierele ce fac parte din acest proiect. La “Constraints” veți observa că există fișierul “constrangeri.xdc”. Acesta trebuie eliminat din proiect și înlocuit cu cel ce corespunde plăcii dumneavoastră despre care aflați mai multe în manualul de utilizare al producătorului.

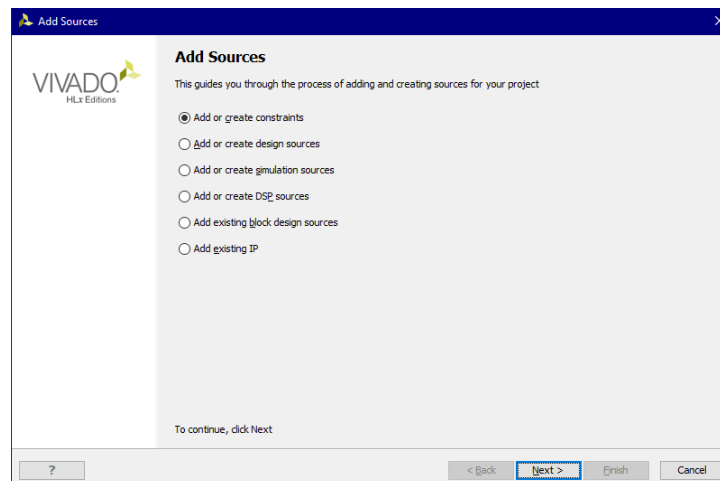
Pentru a-l elimina se apasă click dreapta pe “constrangeri.xdc” după care se selectează “Remove File from Project...”



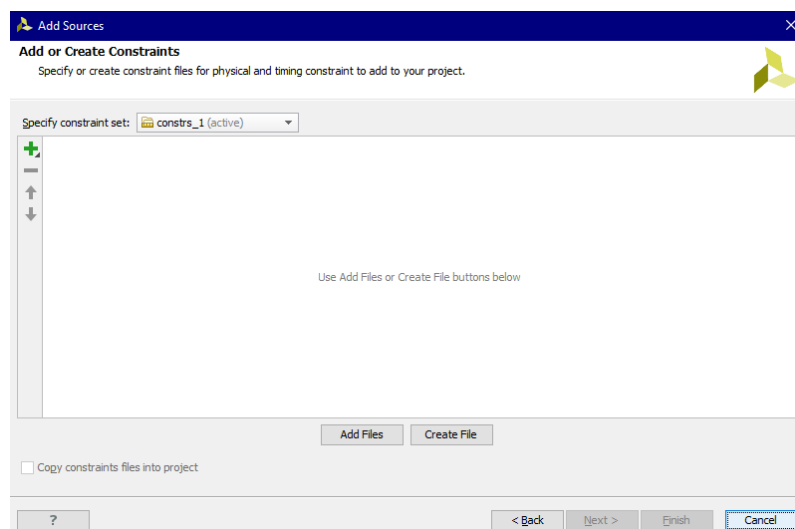
Pentru a adăuga un fișier nou de constrângeri se apasă click dreapta pe “Constraints”, selectați “Add Sources..”



Pe ecran vă va apărea următoarea fereastră unde trebuie să vă asigurați ca aveți bifată opțiunea “Add or create constraints”, după care se apasă next



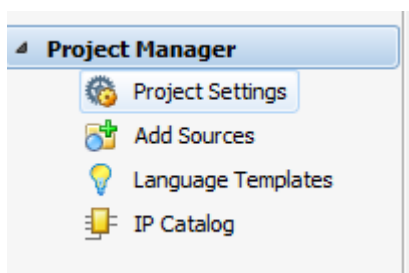
Vă va apărea o fereastră asemănătoare cu cea de mai jos unde aveți posibilitatea să vă creați un fișier de constrângeri sau să adăugați unul deja existent



Indiferent de opțiunea aleasă asigurați-vă ca aveți în fișierul de constrângeri toate datele de intrare ale cutiei negre de la pagina 2

2) Selectarea modelului de placă corespunzător

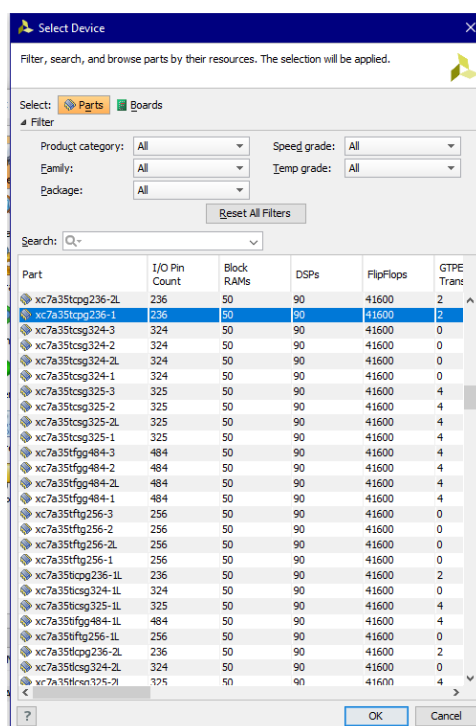
Proiectul este construit pe un Basys3 [XC7A35T-1CPG236C](#). Pentru a modifica acest aspect este necesar să mergem în secțiunea “Project Settings” din partea deraptă a ecranului



Va apărea o fereastră asemănătoare cu cea de mai jos unde ne vom axa atenția pe cea de a doua secțiune “Project device” și apăsați cele 3 puncte incercuite cu roșu din imagine.



Va apărea următoarea fereastră de unde selectați numărul de piesă corespunzător plăcii dumneavoastră



4. JUSTIFICAREA SOLUȚIEI ALESE

Prin utilizarea unui divizor de frecvență și a unui debouncer, soluția aleasă asigură sincronizarea corectă și stabilitatea semnalelor, eliminând zgomotul din ceasul USB. Un automat finit (FSM) gestionat eficient îndeplinește procesul de citire a datelor de la mouse, monitorizând stările și sincronizând semnalele corespunzător. Semnalele interne și ieșirile sunt bine definite și documentate, facilitând înțelegerea și întreținerea codului. Structura modulară permite extinderea ușoară și integrarea de funcționalități adiționale. În ansamblu, codul demonstrează o abordare profesionistă și riguroasă în dezvoltarea interfețelor hardware digitale, asigurând o funcționare corectă și fiabilă în cadrul sistemelor incorporate.

Pentru a număra click-urile de la mouse am implementat un numărător cu o capacitate maximă de 9999, capabil să realizeze și decrementări. Utilizează semnale de intrare pentru a controla direcția de numărare și semnale de reset pentru inițializarea valorii. Pentru a asigura corectitudinea și robustețea funcționării, se utilizează un proces sensibil la flancul ascendent al semnalului de ceas și la semnalul de reset. Numărul este reprezentat printr-un vector de 16 biți, iar incrementarea și decrementarea se realizează corespunzător, verificându-se stările semnalelor de control pentru direcția de numărare. Această implementare este eficientă și clară, asigurând un comportament corect și fiabil în cadrul sistemului integrat.

Pentru a afișa numărul de click-uri ale mouse-ului am ales să respect o structură modulară, împărțind funcționalitățile în componente separate (SSD, divizor_frecventa, prelucrare_cifra, BCD_CIFRA, anoz), ceea ce face codul mai ușor de înțeles, întreținut și reutilizat. De asemenea am utilizat tipuri de date și biblioteci standard conform standardelor IEEE, asigurând portabilitate și compatibilitate cu alte componente și instrumente. Codul folosește structuri de control eficiente și algoritmi optimizați pentru sarcinile specifice, cum ar fi calculul cifrelor BCD și afișarea corespunzătoare pe display.

Am ales să pun un buton pentru, nu pe un switch, deoarece butonul oferă o acțiune momentană, evitând resetarea accidentală, în timp ce un switch poate fi activat în mod continuu sau poate fi comutat în mod accidental, cauzând pierderi de date sau funcționare incorectă a sistemului.

5. POSIBILITĂȚI DE DEZVOLTĂRI ULTERIOARE

În ceea ce privește comunicarea cu mouse-ul o posibilă dezvoltare ar fi sincronizarea procesului de debouncing. Sincronizarea este crucială în sistemele digitale pentru a asigura că semnalele sunt citite sau scrise corect la momentul potrivit în raport cu ciclul de ceas. În cazul interfeței mouse-ului, sincronizarea cu ceasul sistemului este esențială pentru a preveni erorile de sincronizare între semnalele de la mouse și ceasul FPGA. De asemenea, debouncing-ul este o etapă importantă în tratarea fluctuațiilor de semnal, asigurând că semnalele de la mouse sunt citite corect și fără zgomot. De asemenea ar trebui implementate mecanisme de gestionare a erorilor pentru a detecta și a trata corect condițiile neașteptate sau erorile care pot apărea în timpul funcționării sistemului. Acest lucru poate implica, de exemplu, verificarea și validarea semnalelor de la mouse pentru a detecta semnale invalide sau stări de funcționare neașteptate și implementarea unei logici adecvate pentru a gestiona aceste situații, cum ar fi resetarea sau raportarea erorilor într-un mod sigur și fiabil.

În proiect trebuie îmbunătățit divizorul de frecvență al numărătorului pentru o mai bună sincronizare cu semnalul de ceas intern și cu momentul de apăsare pe butoanele mouse-ului. Această îmbunătățire implică ajustarea parametrilor divizorului de frecvență pentru a asigura o sincronizare optimă între semnalul de ceas intern al sistemului și semnalul de ceas al mouse-ului, precum și pentru a asigura că numărătorul reacționează corect la apăsările butoanelor mouse-ului. Este esențială adaptarea acestui divizor pentru a optimiza performanța și fiabilitatea sistemului în ansamblu.

O posibilă dezvoltare ulterioară a proiectului ar putea consta în implementarea unei funcționalități de numărare în sens negativ a clicurilor mouse-ului. Aceasta implică modificarea logicii de numărare astfel încât, atunci când utilizatorul apasă butonul de decrementare la valoarea 0000, numărătoarea să treacă la valoarea negativă, începând de la -1 și continuând în sens descrescător. Această îmbunătățire necesită adaptări în cadrul procesului de numărare și poate implica și ajustări în arhitectura existentă pentru a gestiona corect valori negative și operațiile asociate acestora.

6. BIBLIOGRAFIE

- ❖ Octavian Creț, Lucia Văcariu – Limbajul VHDL. Îndrumător de laborator
- ❖ Baruch, Z. F., *Input/Output Systems* (în limba engleză)
- ❖ <http://www.ijreat.org/Papers%202013/Issue5/IJREATV1I5007.pdf>
- ❖ https://github.com/RobRoyce/fpga_mouse_controller_basys3/blob/master/README.md
- ❖ <https://www.fpga4student.com/2017/12/how-to-interface-mouse-with-FPGA.html>
- ❖ <https://forum.digilent.com/topic/24733-does-basys-3-usb-hid-supports-device-with-mousekeyboard/>
- ❖ <https://digilent.com/reference/programmable-logic/basys-3/demos/gpio>
- ❖ <https://fpga4student.quora.com/FPGA-tutorial-How-to-interface-a-mouse-with-Basys-3-FPGA-http-www-fpga4student-com-2017-12-how-to-interface-mouse>
- ❖ <https://opencores.org>
- ❖ <https://www.youtube.com/watch?v=VtbPexCocVo&list=PLbFgDf51ZkCEJb9MvxKIs-0q2obouwf-f>
- ❖ https://www.youtube.com/watch?v=A1YSbLnm4_o&t=1s
- ❖ <https://www.quora.com/How-do-I-interface-a-PS2-mouse-with-an-FPGA>
- ❖ https://www.ripublication.com/ijaer23/ijaerv18n2_01.pdf
- ❖ <https://www.fpgakey.com/tutorial/chapter56>
- ❖ <https://misterfpga.org/viewtopic.php?t=1949>
- ❖ https://www.amd.com/content/dam/amd/en/documents/university/aup-boards/XUPBasys3/documentation/Basys3_rm_8_22_2014.pdf
- ❖ https://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/2001_w/interfacing/PS2_Mouse_TX/PS2MouseDataTX.htm
- ❖ http://ebook.pldworld.com/_eBook/FPGA/HDL/-Examples-/interfacing%20mouse%20with%20VHDL.pdf
- ❖ https://ftp.intel.com/Public/Pub/fpgaup/pub/Intel_Material/18.1/University_Program_IP_Cores/Input_Output/PS2_Controller.pdf