



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Facultatea de Automatică și Calculatoare - CATEDRA CALCULATOARE

STRUCTURA SISTEMELOR DE CALCUL

**Măsurarea accelerației cu senzorul de pe placa Nexys 4 DDR și
transmiterea ei la un dispozitiv mobil**

Baleanu Sorina-Diana

Ionescu Raluca-Ionela

Grupa: 302310

Prof. îndrumător: Lisman Florin- Dragos

Data: 07.01.2021

Cuprins

Cuprins	2
1.Rezumat	3
2.Introducere	4
3.Fundamentare teoretică	5
3.1 Interfața UART	5
3.2 Interfața de comunicare SPI	6
3.3 Accelerometrul ADXL362	7
4.Proiectare și implementare	9
4.1 Interfața UART	9
4.2 Interfața de comunicare SPI	11
4.2 Unitatea de control al accelerometrului	13
4.3 Implementarea aplicației pe mobil	15
4.4 Manual de utilizare	16
5. Rezultate experimentale	17
6. Concluzii	18
7. Bibliografie	19

1.Rezumat

Proiectul realizat a presupus folosirea plăcii Nexys 4 DDR și a accelerometrului ADXL363 pentru măsurarea unei accelerații și transmiterea acesteia pe un dispozitiv mobil prin intermediul unui cablu USB. S-a realizat de asemenea și o aplicație pe dispozitivul mobil care preia și afișează accelerația măsurată. Proiectul a fost realizat în mediul de proiectare Xilinx Vivado 2020.1 Design Suite în limbajul VHDL, iar aplicația de Android a fost realizată în Android Studio utilizând limbajul Java. Tehnicile de bază folosite pentru acest proiect au fost cele de comunicare asincronă, UART pentru transmiterea datelor furnizate de placă către telefonul mobil, dar și interfața serială sincronă SPI utilizată pentru citirea registrelor accelerometrului.

2.Introducere

FPGA-urile reprezintă, în momentul de față, platforme pentru sisteme digitale complexe, oferind elemente computaționale numeroase și din ce în ce mai sofisticate. Acest termen de FPGA se traduce prin matrice sau arie de porți logice re-programabilă. Actualmente, pe o astfel de placă sunt integrate și blocuri de memorie RAM precum și multiplicatoare hardware.

Cu ajutorul acestor circuite, dezvoltarea aplicațiilor logice de control a devenit una mai rapidă, mai flexibilă și mai optimizată deoarece implementarea se realizează direct la nivelul de hardware.

Tema proiectului constă în realizarea unei comunicări dintre placa fpga Nexys 4 DDR și un mobil Android, cu scopul transmiterii unor date de la accelerometrul încorporat în placa Nexys la aplicația mobil. Accelerația este măsurată de către accelerometru pe cele 3 axe: x,y, z, astfel, pentru stabilirea axei ce se vrea transmisă se vor folosi primele 2 butoane de tip switch. (00 - se va transmite accelerația de pe axa x, 01-axa y, 10-axa z)

Soluția găsită de noi va măsura încontinuu accelerația, prin activarea modului de măsurare. Astfel, nu este necesară apăsarea unui buton pentru citirea din regiștrii.

3. Fundamentare teoretică

Pentru realizarea proiectului am utilizat accelerometru, o componentă integrată deja în placa folosită, Nexys 4 DDR.

Pentru implementarea comunicației seriale am utilizat următoarele interfețe: SPI(Serial Peripheral Interface), UART(Universal Asynchronous Receiver Transmitter).

3.1 Interfața UART

Se ocupă de comunicarea serială asincronă dintre fpga și un dispozitiv periferic conectat la portul serial.

În transferul asincron, emițătorul și receptorul au semnale de tact separate, sincronizarea la nivel de bit este asigurată numai pe durata transmisiei efective a fiecărui caracter. Se folosește însă o rată de eșantionare, generată independent la sursă și destinație, numită baud rate (număr de biți transmiși pe secundă). Fiecare caracter transmis este precedat de un bit de START (linia de comunicație are nivelul logic 0) și este urmat de cel puțin un bit de STOP (linia de comunicație are nivelul logic 1). Fiecare caracter transmis este încadrat de acești biți de start și stop.

Datele transmise prin comunicație serială sunt codificate folosind coduri ASCII.

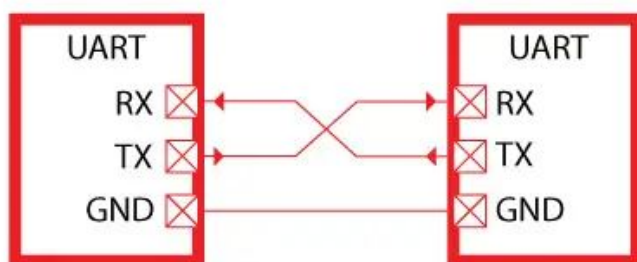


Fig.1

În fig. 1 se poate observa modul în care se realizează conexiunea prin intermediul celor 3 pini: rx,tx, gnd.



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Atunci când receptorul detectează bitul de START (ce indică începutul unui caracter), se pornește un oscilator de ceas, măsurându-se astfel intervalul de timp corespunzător unui bit. Eșantionarea biților se realizează aproximativ la mijlocul intervalului corespunzător fiecărui bit.

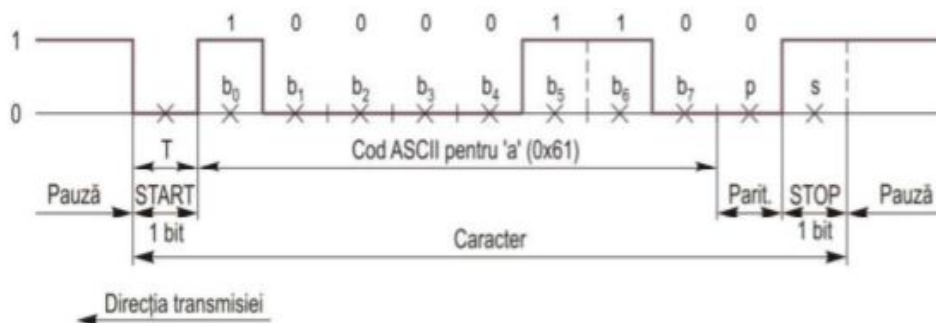


fig.2(sursa: <http://users.utcluj.ro/~baruch/ssc/labor/Testare-Depanare.pdf>)

În fig. 2 este ilustrată transmisia serială asincronă a caracterului cu codul ASCII 0x61.

La protocoalele de tip UART se poate stabili de către programator:

- rata de transfer (denumită baud rate, viteza de transmisie a biților) ce ia valori precum 9600, 19200 sau 115200
- numărul de biți transmiși, bitul de paritate, biții de sincronizare(start, stop)
- fluxul de control

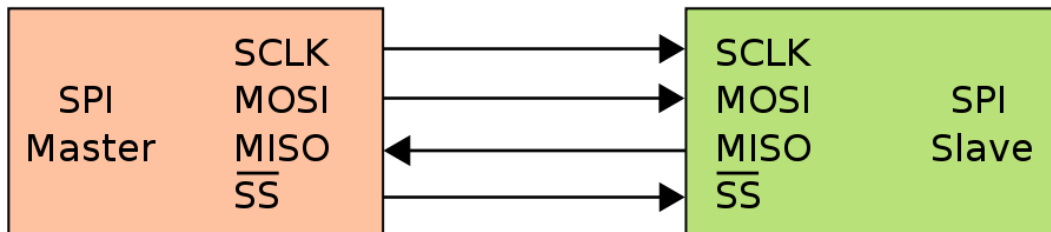
3.2 Interfața de comunicare SPI

SPI este o interfață serială, ce funcționează în modul de comunicație duplex, datele fiind transmise în ambele direcții simultan. Arhitectura acestei interfețe este de tipul master-slave, unde dispozitivul de tip master inițiază toate transferurile.

În proiectul nostru, interfața SPI este utilizată de controllerul de comunicație USB.

Interfața SPI utilizează următoarele semnale:

- SCLK : serial clock, generat de către master
- MOSI : master output, slave input, generat de către master
- MISO : master input, slave output, generat de către slave
- SS : chip select sau slave select



Realizarea transmisiei de date

Înainte de a se începe comunicatia, masterul trebuie să stabilească ce frecvența de ceas se va utiliza, aceasta fiind mai mică sau egală cu maximumul frecvenței suportată de slave (de obicei în intervalul 1-70MHz). Prin activarea semnalului SS se selectează slave-ul.

Transmisia de tip duplex apare în timpul fiecărui ciclu de ceas SPI, iar master-ul transmite un bit pe linia MOSI, fiind citit de către slave. Slave-ul la rândul său transmite un bit pe linia MISO, fiind citit de către master. De asemenea, transmisia poate să fie și unidirecțională.

Pentru transmiterea și recepția datelor se vor utiliza registre de deplasare, dimensiunea lor depinzând de numărul de biți transmiși. Cum primul bit transmis este cel mai semnificativ, se va efectua o deplasare la stnga a fiecărui registru.

Trebuie să se configureze, de asemenea și o polaritate și faza a semnalului de ceas. Polaritatea(CPOL) presupune nivelul logic al semnalului de ceas, iar faza(CPHA) se referă la frontul crescător sau descrescător.

Combinatia dintre polaritatea și faza semnalului determina un mod de comunicare SPI

3.3 Accelerometrul ADXL362

Pentru comunicarea cu accelerometru se va folosi protocolul SPI, pe modul de comunicare 0 (CPOL = 0 și CPHA = 0).

Pentru a configura accelerometrul trebuie să urmăm pașii:

- resetarea, prin scrierea "0x52" în registru la adresa 0x1F (SOFT_RESET)
- activarea modului de măsurare, realizată prin scrierea "0x02" în registru la adresa 0x2D (POWER_CTL)
- citirea a 8 biți



UNIVERSITATEA TEHNICĂ **DIN CLUJ-NAPOCA**

Comenzile recunoscute de către accelerometru sunt:

- 0x0A - scriere registru
- 0x0B - citire registru
- 0x0D - citire memorie FIFO

Pentru a citi datele de pe axele x, y, z se dau următoarele adrese ale registrelor accelerometrului:

- 0x08 - pentru XDATA (axa x)
- 0x09 - pentru YDATA (axa y)
- 0x0A - pentru ZDATA (axa z)

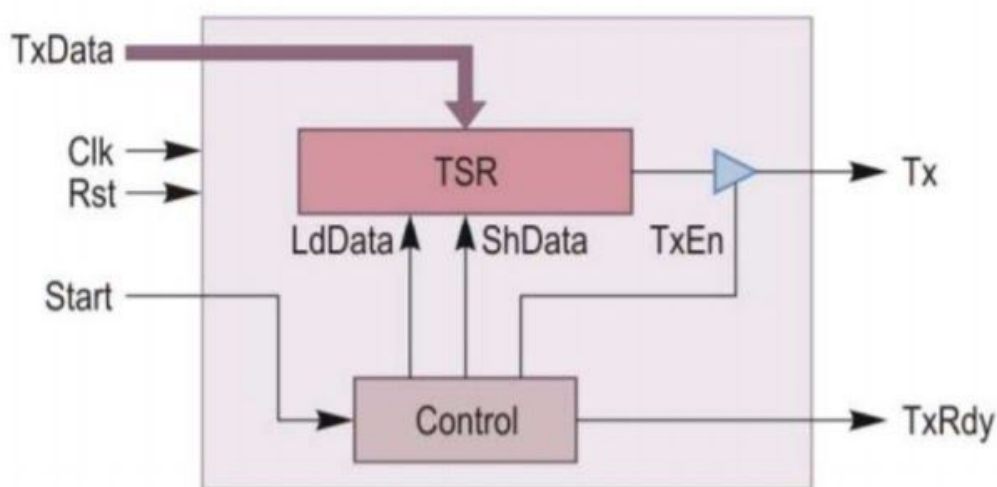
Comunicatia SPI se bazează pe furnizarea unei adrese, unei date, și semnal de scriere sau citire. La fiecare tranzactie se va scrie sau citi 8 biti de data, in functie de semnalul de comanda.

4. Proiectare și implementare

Proiectul este implementat folosind următoarele componente care urmează a fi prezentate în detaliu: o componentă pentru comunicarea serială asincronă UART între placa de dezvoltare FPGA și telefonul mobil și o componentă pentru comunicarea pe baza protocolului SPI între placa Nexys DDR 4 și senzorul de accelerație ADXL362.

4.1 Interfața UART

Intrările modulului uart sunt semnalul de ceas Clk, semnalul de resetare sincronă Rst, vectorul de 8 biți TxData, care reprezintă octetul ce se va transmite și semnalul Start care indică când să se înceapă transmiterea. Ieșirile modulului sunt linia Tx pr care sunt transmise datele în mod serial și semnalul TxRdy care indică faptul că transmiterea a fost terminată atunci când se află în starea sa activă.

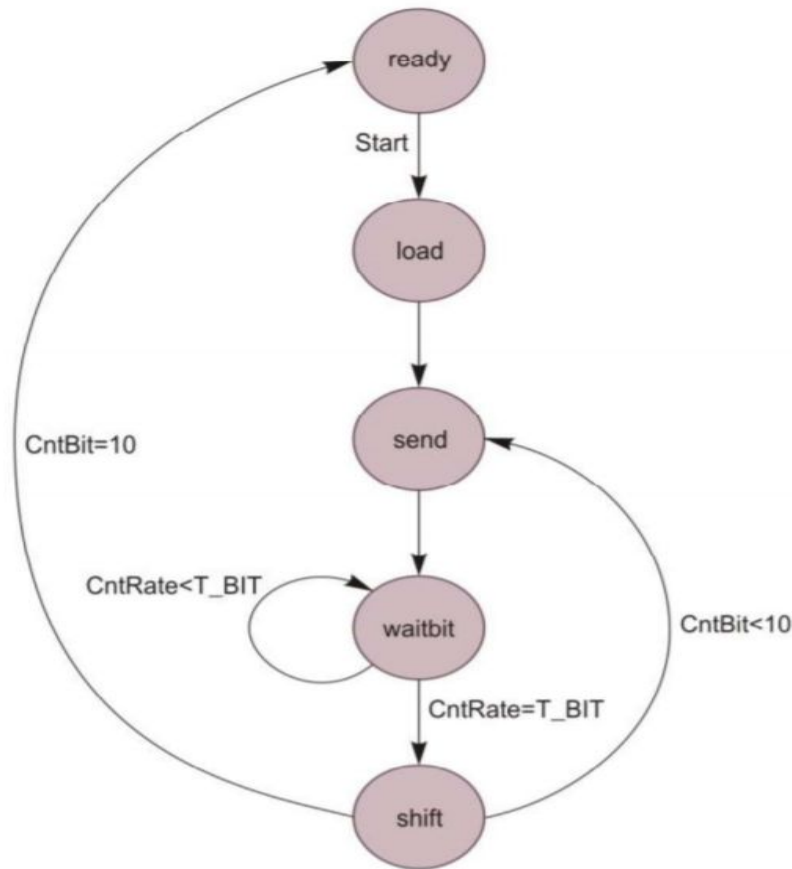


În componența sa acest modul conține un registru de deplasare spre dreapta de 10 biți, unde este încărcat octetul transmis care este completat cu bitul de START ('0') și cu bitul de STOP('1'). Modulul transmitatorului conține și un buffer comandat de semnalul TxEn, atunci când acest semnal este activat ieșirea serială a registrului de deplasare va fi transmisă pe linia Tx.

Unitatea de control a fost implementată cu sub forma unui automat de stare.



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA



Se vor folosi doua semnale $CntBit$ pentru contorizarea numărului de biți transmiși și $CntRate$ pentru contorizarea ciclurilor de ceas.

- **Starea 1 :** *Ready* : Se inițializează la 0 semnalele $CntBit$ și $CntRate$.Dacă $Start$ devine '1', atunci se va intra în următoarea stare.
- **Starea 2 :** *Load*: În aceasta stare se încarcă registrul de deplasare cu octetul care trebuie transmis completat cu biții $START$ și $STOP$, apoi se trece în starea următoare.
- **Starea 3 :** *Send*: În această stare, se incrementează contorul $CntBit$, după care se trece în starea următoare.
- **Starea 4:** *WaitBit*: În aceasta stare se așteaptă trecerea intervalului de timp egală cu durata de transmisie a unui bit (115200). Se incrementează $CntRate$ pana ce ajunge la aceasta valoare, după care se va trece în starea următoare.

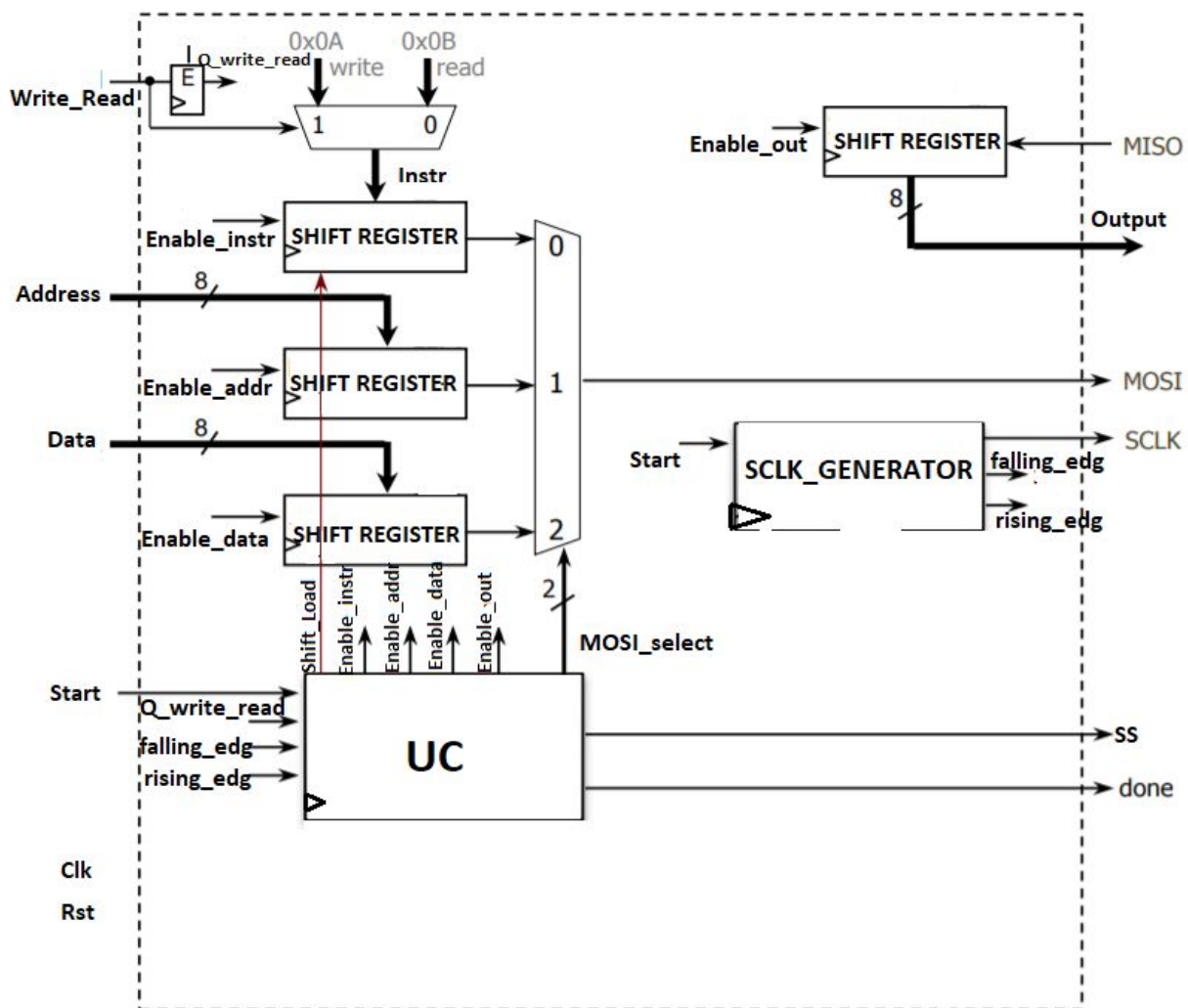


UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

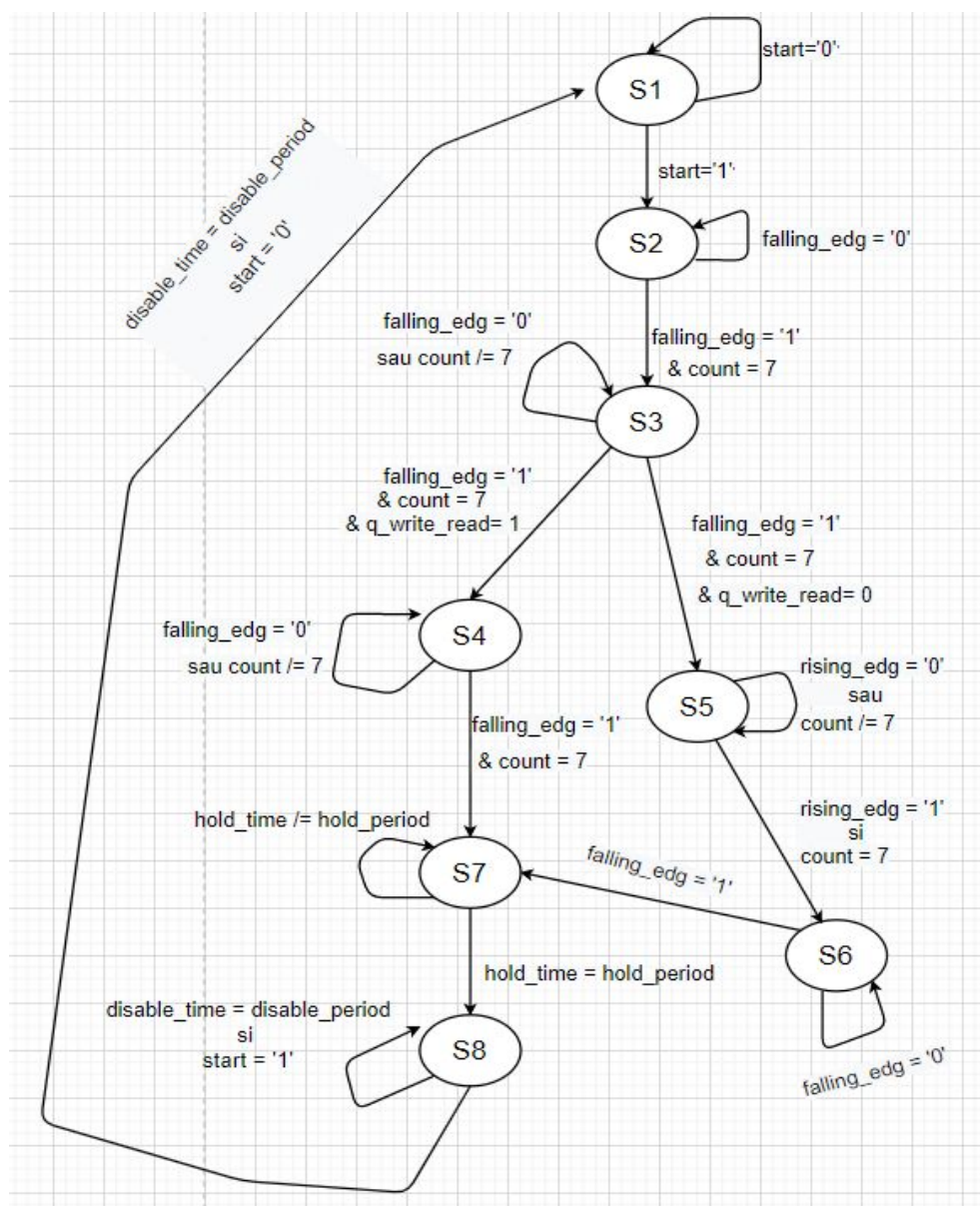
- **Starea 5 – Shift:** În aceasta stare sese deplasează spre dreapta registrul de deplasare. Se verifica contorul CntBit si în cazul în care nu s-au transmis cei 10 biți se va reveni în starea send, iar în caz contrar se revine în starea ready.

4.2 Interfața de comunicare SPI

Intrările modulului care se ocupă cu gestionarea comunicării dintre accelerometrul plăcii și placa de dezvoltare sunt semnalul de ceas Clk, semnalul de resetare sincronă Rst, vectorii de 8 biți Data, Address și Instr care vor fi transmiși de către unitatea principală de control a accelerometrului, semnalul de Start care indică începerea comunicării cu senzorul, semnalul de Write_Read care este transmis de asemenea de unitatea principală de control a accelerometrului și semnalul MOSI, utilizat pentru transmisia datelor de la dispozitivul master la dispozitivul slave. Ieșirile modulului sunt semnalul MISO utilizat pentru transmisia datelor de la un dispozitiv slave la un dispozitiv master, semnalul SS folosit pentru selecția dispozitivului slave și semnalul SCLK.



În componenta acestui modul s-au utilizat 4 registre de deplasare la stanga de 8 biți pentru instrucțiune, adresa, date și ieșire. Funcționalitatea acestor registre este determinată de unitatea de control a modului care este implementată printr-un automat de stări finite. Se mai folosesc de asemenea și două multiplexoare, unul pentru selecția instrucțiunii, iar celălalt pentru a selecta dacă accelerometrului i se transmite instrucțiunea, adresa sau datele. Automatul de stare funcționează de asemenea cu ajutorul unui generator pentru semnalul de ceas SCLK, care va furniza un semnal de ceas care se presupune a fi de 8 ori mai redus față de semnalul de ceas al plăcii și va activa semnalul rising_edg atunci când se afla pe frontul crescător al semnalului de ceas generat și va activa semnalul falling_edg când este pe front decrescător.





UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

- **Starea 1** : Se oprește comunicarea cu dispozitivul slave si se așteaptă activarea semnalului Start, când acesta devine activ se vor încarcă datele pe registrul de deplasare spre stanga si se va trece în starea următoare.
- **Starea 2** : Pe frontul descrescător al semnalului de ceas generat SCLK se verifică dacă s-au transmis cei 8 biti catre dispozitivul slave, in caz afirmativ se trece în starea următoare. În aceasta stare se transmit către accelerometru biții corespunzători instrucțiunii, astfel ca selecția pentru MOSI este "00" si se vor shifta către stanga datele din registrul de deplasare pentru instrucțiune.
- **Starea 3** : Pe frontul descrescător al semnalului de ceas generat SCLK se verifica dacă s-au transmis cei 8 biti catre dispozitivul slave, in caz afirmativ se trece în starea următoare. În aceasta stare se transmit către accelerometru biții corespunzători adresei, astfel ca selectia pentru MOSI este "01" si se vor shifta către stanga datele din registrul de deplasare pentru adresa.
- **Starea 4**: Pe frontul descrescător al semnalului de ceas generat SCLK se verifica dacă s-au transmis cei 8 biti catre dispozitivul slave, in caz afirmativ se trece în starea următoare. În aceasta stare se transmit către accelerometru biții corespunzători datelor, astfel ca selecția pentru MOSI este "10" si se vor shifta către stanga datele din registrul de deplasare pentru instrucțiune.
- **Starea 5** – Pe frontul crescător al semnalului de ceas generat SCLK se verifica dacă s-au primit cei 8 biti de la dispozitivul slave, in caz afirmativ se trece în starea următoare. În aceasta stare se primesc pe linia de date MISO datele furnizate de către accelerometru si se va shifta spre stanga registrul de deplasare corespunzător ieșirii controlerului SPI.
- **Starea 6**: Pe frontul descrescător al semnalului de ceas se va trece în starea 7.
- **Starea 7** – Se verifica dacă hold time-ul pentru accelerometru este atins după care se trece în starea următoare.
- **Starea 8** – Se verifica dacă disable time-ul pentru accelerometru este atins după care se așteaptă activarea semnalului de start, pentru a se întoarce înapoi în starea 1.

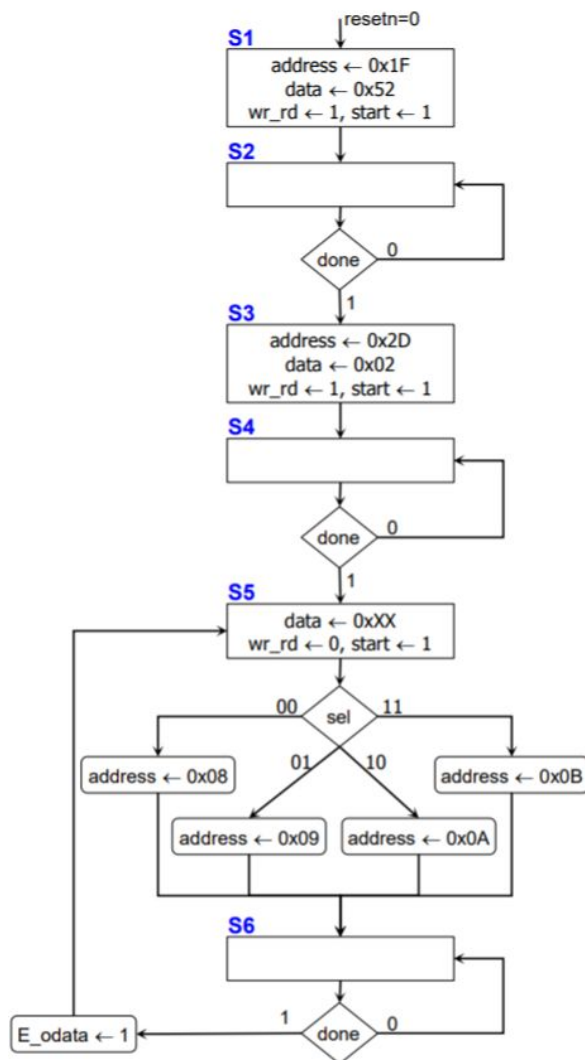
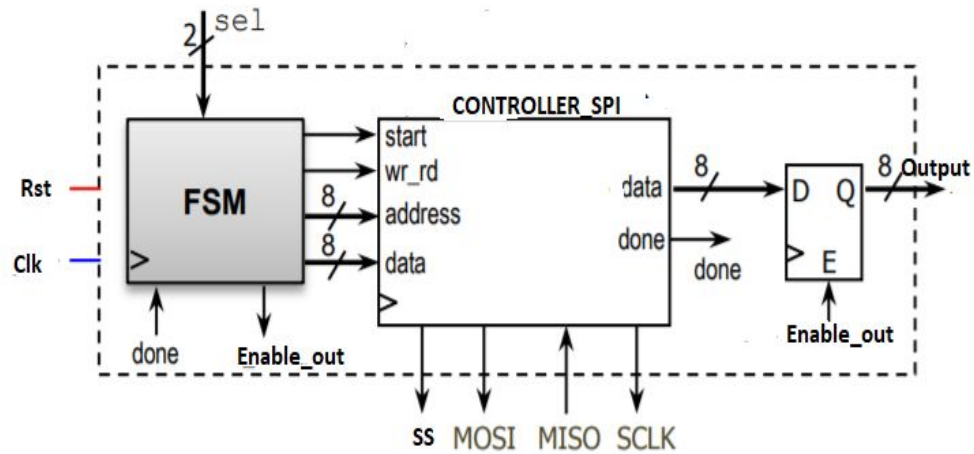
4.2 Unitatea de control al accelerometrului

Intrările modului care se ocupă cu gestionarea comunicarii dintre accelerometrul plăcii și placa de dezvoltare sunt semnalul de ceas Clk, semnalul de resetare sincronă Rst, semnalul de selecție Sel, semnalul MISO, iar ieșirile sunt SS, MOSI, SCLK si Output.



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Unitatea de control a accelerometrului este alcatuita din controller-ul SPI, un registru care pastreaza datele furnizate de accelerometru si un automat cu stări finite.



Starea 1 : În aceasta stare se va face o scriere pe registrul de soft reset a accelerometrului, adică se va scrie pe registrul x"1F", valoarea x"52". Semnalul Write_Read va avea valoarea '1' deoarece dorim sa executam o scriere , iar instrucțiunea va fi selectata în controller-ul SPI, cu ajutorul acestui semnal se va selecta instrucțiunea de scriere care este x"0A". După aplicarea acestor comenzi se trece în starea 2.

· **Starea 2 :** Se asteapta activarea semnalului Done, care reprezinta faptul ca datele au fost transmise cu succes dispozitivului slave după care se trece în starea 3.

· **Starea 3 :** În aceasta stare se activa modul de masurare al



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

accelerometrului, adică se va scrie pe registrul x"2D", valoarea x"02"(registrul POWER_CTL). Semnalul Write_Read va avea valoarea '1' deoarece dorim sa executam o scriere , iar instrucțiunea va fi selectata în controller-ul SPI, cu ajutorul acestui semnal se va selecta instrucțiunea de scriere care este x"0A".Se va trece mai apoi in starea 4.

- **Starea 4:** Se asteapta activarea semnalului Done, care reprezinta faptul ca datele au fost transmise cu succes dispozitivului slave după care se trece în starea 5.

- **State 5** – În aceasta stare se vor executa în funcție de selecție citiri ale diferitelor regiștrii ai accelerometrului..Semnalul Write_Read va avea valoarea '0' deoarece dorim sa executam o citire , iar adresa va fi selectata in de semnalul Sel. Pentru selecția "00" se va citi registrul x"08" care corespunde celor mai semnificativi 8 biti si aceleratiei pe axa X, pentru selecția "01" se va citi registrul x"09" care corespunde celor mai semnificativi 8 biti si accelerației pe axa Y, pentru selecția "10" se va citi registrul x"0A" care corespunde celor mai semnificativi 8 biti si aceleratiei pe axa Z, iar pentru selecția "11" se va citi registrul x"00" care corespunde indicatorului specific firmei Analog Devices .

- **Starea 6:** În aceasta stare se va ramane cat timp semnalul Done este inactiv, în momentul în care acesta se activează se face scrierea în registrul cu datele de ieșire din accelerometru si apoi se revine în starea 5 pentru citirea următorii valori.

4.3 Implementarea aplicației pe mobil

Proiectul constă în transmiterea datelor de la accelerometrul la un dispozitiv Android. Pentru realizarea acestui lucru, am creat o aplicație, dezvoltată în mediul de programare Android Studio, bazata pe comunicarea prin USB. Limbajul de programare utilizate este java.

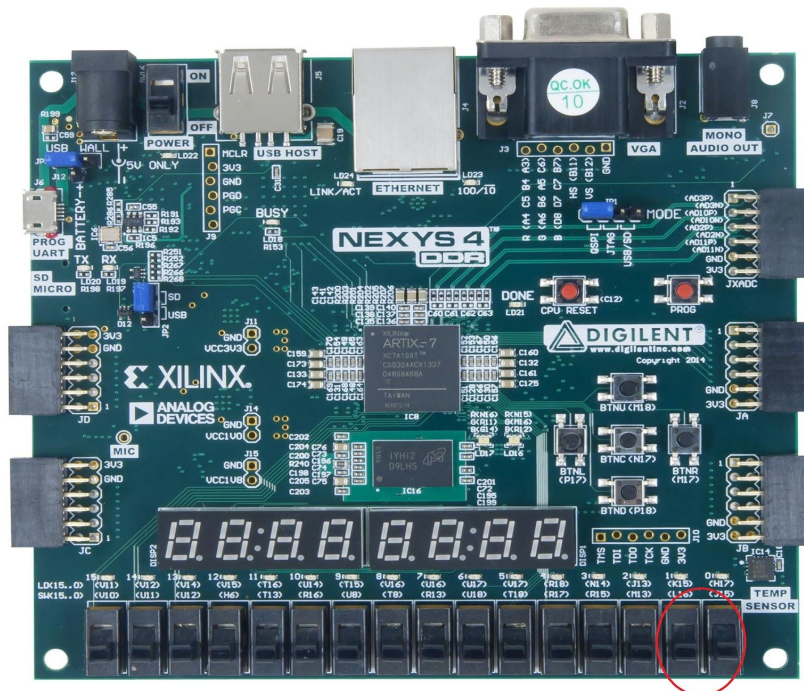
Interfața acesteia consta într-un progressbar, în care se poate observa mărirea sau încetinirea accelerației. De asemenea, este afișată și valoarea curentă a accelerației, ce este transmisă de la placa fpga cu ajutorul unui cablu USB.

Pentru comunicarea prin USB-Serial channel, clasa principala a implementat interfata USBSerialListenet, suprascriind metodele din aceasta.



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

4.4 Manual de utilizare



Pentru ca orice utilizator să poată rula și verifica funcționalitatea proiectului acesta trebuie să urmeze următorii pași:

1. Utilizatorul trebuie să instaleze mediul de proiectare Vivado pe calculatorul și instalarea aplicației de mobil pe un dispozitiv Android
2. Se deschide proiectul “ Accelerometru” in Vivado
3. Se conectează placa FPGA la calculator cu un cablu corespunzător.
4. Se configurează placa cu fișierul bitstream generat din proiect.
5. Se conectează placa FPGA la telefon cu un cablu corespunzător.
6. Se porneste aplicația pe Android.
7. Se alege data care dorește a fi furnizată de către accelerometru prin utilizarea celor două switch-uri încercuite în imaginea de mai sus.
 - pentru selecția “00” se va furniza accelerația pe axa X.
 - pentru selecția “01” se va furniza accelerația pe axa Y.
 - pentru selecția “10” se va furniza accelerația pe axa Z.
 - pentru selecția “11” se va furniza indicatorul specific firmei Analog

Devices.

8. Datele vor fi disponibile atât pe display-ul plăcii, pe primele 7 led-uri, cât și în aplicația mobilă, modificarea accelerației fiind ușor de vizualizat în aplicație pe bara de progres.

5. Rezultate experimentale



Aplicația de pe Android primește datele prin usb de la placa de dezvoltare și le afișează folosind un progressBar care are valoarea maximă de 255.

De asemenea am testat funcționalitatea proiectului prin afișarea pe display-ul plăcii a valorilor corespunzătoare a datelor primite de la accelerometru și am folosit selecția "11" pentru a testa corectitudinea datelor deoarece aici vom citi valoarea registrului x"00" al

accelerometrului care are mereu valoarea x''_{AD} corespunzătoare indicatorului specific firmei Analog Devices.

6. Concluzii

Realizarea acestui proiect a presupus implementarea interfețelor de comunicare UART (pentru comunicarea dintre placa FPGA și aplicația mobil) și SPI (pentru accelerometrul încorporat în placa), precum și dezvoltarea unei aplicații android.

Pe baza schemei bloc și a diagramelor de stare prezentate anterior, am implementat modulele ce se ocupă de comunicare și controlere pentru aceste module.

În urma realizării acestui proiect am reușit să avem o imagine mai clară asupra modului în care funcționează atât accelerometrul, cât și a comunicărilor seriale sincrone și asincrone.

În ceea ce privește o posibilă dezvoltare ulterioară, o îmbunătățire majoră ar fi citirea și transmiterea accelerației de pe toate cele 3 axe, fără a mai fi necesară selecția acesteia. De asemenea, s-ar putea îmbunătăți și aplicația de pe mobil, care să permită selecția axei ce se vrea preluată de la plăcuță și afișată pe ecran, sau în cazul în care date de pe mai multe axe, acestea să fie afișate sub forma unui grafic.

7. Bibliografie

[2]

https://www.unitbv.ro/documente/cercetare/doctorat-postdoctorat/sustinere-teza/2019/szabolcs-hajdu/Rezumat_Hajdu_Szabolcs.pdf
<http://users.utcluj.ro/~baruch/ssc/labor/Proiectare-Interfata.pdf>

[3]

<https://fpgalover.com/software/coding-usb-serial-using-android-studio/>

[4]

<https://learnappmaking.com/how-to-make-an-app/>
<https://mashable.com/article/build-mobile-apps/?europe=true>
<https://www.secs.oakland.edu/~llamocca/Courses/ECE4710/Notes%20-%20Unit%203.pdf>
<https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/start/>
<https://developer.android.com/training/basics/firstapp/creating-project>
<https://www.livescience.com/40102-accelerometers.html>
<https://vhdlguide.readthedocs.io/en/latest/>
<https://socratic.org/questions/how-is-acceleration-measured>
http://ww1.microchip.com/downloads/en/DeviceDoc/bluetooth_cr_UG-v1.0r.pdf
<https://www.digikey.com/eewiki/pages/viewpage.action?pageId=90243412>
<https://reference.digilentinc.com/learn/programmable-logic/tutorials/pmod-ips/2018.2>
<https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-gpio-demo/start?redirect=1>
<https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf>
http://users.utcluj.ro/~onigaf/files/teaching/AC/AC_indrumator_laborator.pdf