A picture containing text, clipart

Description automatically generated

Universitatea Tehnica din Cluj-Napoca

Facultatea de automatica si calculatoare

**Poriect Structura Sistemelor de Calcul**

**GPS**

**Oprian Ianis Stefan**

**Indrumator de proiect:**

**Filip Marian Gherman**

Contents

[1. Introducere: 2](#_Toc125480188)

[2.Fundamentare teoretica: 2](#_Toc125480189)

[3.Proiectare si implementare: 3](#_Toc125480190)

[5.Bibliografie: 7](#_Toc125480191)

# 

# 1. Introducere:

Proiectul referit in aceasta documentatie consta in afisarea pe terminalul virtual al aplicatiei Tera Term, o aplicatie care ajuta la cititirea de pe porturi serial, coordonatelor transmise de catre modulul GPS NEO 6M, in format nmea folosind IDE-ul Vivado 2020.2 pentru realizarea diagramei block, pe care s-a incarcat microprocesorul MicroBlaze, tehnologie ce permite compilarea si rularea de cod in limbajul C/C++.

Bitstreamul diagramei block a fost generat si trimis catre cel de-al doilea IDE vital proiectului, si anume Vitis, unde este permisa scrierea de cod C++, cod ce a fost rulat de microblaze. S-a folosit o librarie numita “xuartlite.h”, cu ajutorul careia s-au prelucrat datele obtinute dupa pornirea montajului, si s-au obtinut coordonatele corecte, cu o acuratete remarcabila.

# 2.Fundamentare teoretica:

In cadrul proiectului s-au folosit numeroase tehnologii hardware de trimitere, primire si prelucrare a datelor, acestea fiind vitale montajului cu ajutorul caruia s-a realizat manipularea informatiei in software. Aceste tehnologii sunt:

* O placuta FPGA digilent basys3, care este create special pentru design suite-ul Vivado impreuna cu arhitectura Xilinx Artyx 7 FPGA. S-au folosit in special pinii de pe PMod-ul JA al acestuia, pentru transmiterea de date de la modulul gps catre portul serial.
* Un cablu de date micro-usb.
* Un modul gps NEO 6M care foloseste un format special nmea, pe care il afisam cu ajutorul Vitis in terminalul oferit de Tera Term. Din toate informatiile, pentru extragerea coordonatelor in spatiu este nevoie doar de campul GPGGA care continue latitudinea si longitudinea.
* Un laptop cu ajutorul caruia putem trimite curent catre placuta si modulul GPS

Montajul cu care s-a lucrat a fost realizat cu ajutorul materialului oferit de Digilent, care confera informatii despre asezarea pinilor de pe PMod. S-au folosit 4 cabluri jumper, de tip tata-mama, 2 rezervate pentru pinii JA6 si JA5 unde s-au conectat in ordine VCC respective GND-ul modulului, iar apoi au fost conectati RX-ul modulului la J1 iar TX-ul modulului la L2.

Conexiunea realizata :

A picture containing text, electronics

Description automatically generated

# 3.Proiectare si implementare:

Realizarea diagramei block:

Pentru realizarea diagramei block a proiectului s-a adaugat ca IP un microcontroller microblaze, pentru care vivado a facut automat conexiunile si pregatit circuitul pentru folosirea acestuia in mod corect.

Diagram, schematic

Description automatically generated

Aditia la aceasta diagrama block facuta in schimb este a componentei “axi\_uartlite\_1”, ale carei semnale rx sit x au fost mapate separat strict pentru a citi date de pe Pmod-ul JA al placutei basys3. Dupa ce acesti pasi au fost setati, s-a realizat un wrapper pentru acest proiect si am generat bitstream-ul, care urma sa fie exportat astfel incat sa fie integrat intr-un proiect Vitis. Odata exportat, IDE-ul Vitis va ajuta la scrierea de cod C++ pentru citirea si afisarea coordonatelor de pe modulul GPS.

Modulul UART AXI UART LITE:

Modulul folosit pentru primirea datelor este AXI URAT Lite are ca caracteristici:

* Face conversia parallel to serial pe caractere primate prin interface AXI4-Lite si conversia serial to parallel pe caractere primate prin periferica seriala.
* Transmite si primeste de un range de 5-8 caractere, continand un bit de stop si biti de paritate.
* Generaza intreruperi pe rising\_Edge cand receiverul FIFO primeste date sau cand acesta ajunge sa fie gol.

Diagrama block a AXI UART Lite:

Diagram

Description automatically generated

Interfata Axi ajuta la implementarea interfetei slave pentru access sau transmitere de date “AXI4-Lite ”.

Registrii UART Lite continue registrii de control, de status, si o pereche de registrii de trimitere/primire FIFO, fiecare de lungime de 16 caractere.

Uart control continue RX si TX control, care copiaza date primite la baud rate-ul configurat de user si scrie aceste date in Receive Data FIFO, respective citeste date din Transmit Data FIFO si trimite catre interfata UART Tx. Acesta mai continue blocul BRG (Baud Rate Generator) care genereaza variat baud rate-uri programabile, si Interrup Control, cu ajutorul caruia se pot realiza intreruperi pe rising\_edge atunci cand FIFO devine ocupat sau gol.

Formatul NMEA:

NMEA este un acronym pentru National Marine Electronics Association si a existat chiar mult inainte sa fi existat GPS. Scopul NMEA este de a da echipamentului electronic abilitatea de a configura hardware-ul deopotriva cu software. GPS-urile cu format NMEA fac viata mai usoara software developerilor, deoarece acesta face mai usor scrisul de software in general pentru majoritatea tipurilor de GPS de pe piata. Structura este in felul urmator:

$GPGGA,181908.00,3404.7041778,N,07044.3966270,W,4,13,1.00,495.144,M,29.200,M,0.10,0000\*40

Unde:

* **GP** reprezinta faptul ca este o pozitie GPS.
* **181908.00** este timestamp-ul UTC: ore minute si secunde.
* **3404.7041778** este latitudinea in format DDMM.MMMMM.
* **N** latitudinea Nordica .

**07044.3966270** este longitudinea in format DDDMM.MMMMM format.

* **W** longitudine vestica.

Proiectarea in Vitis:

Proiectarea in vitis a constat in scrierea de cod C++ cum a fost mentionat mai sus.Pentru aceasta parte s-a scris cod cu ajutorul unei librarii speicale pentru uart si anume “xuartlite.h” cu ajutorul careia s-a initializat un vector care primeste date de la pinii unde s-a realizat conexiunea pe placuta basys3. In final, s-a afisat tot in terminalul tera term. Codul folosit este urmatorul:

**#include** "xparameters.h"

**#include** "xstatus.h"

**#include** "xuartlite.h"

**#include** "xil\_printf.h"

**#include** "String.h"

**#include** "stdio.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Constant Definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#define** UARTLITE\_DEVICE\_ID\_0 XPAR\_UARTLITE\_0\_DEVICE\_ID

**#define** UARTLITE\_DEVICE\_ID\_1 XPAR\_UARTLITE\_1\_DEVICE\_ID

**#define** TEST\_BUFFER\_SIZE 16

**int** **UartLitePolledExample**(u16 DeviceId);

**char** lat[100];

**char** lng[100];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variable Definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

XUartLite UartLite\_0; /\* Instance of the UartLite Device \*/

XUartLite UartLite\_1; /\* Instance of the UartLite Device \*/

/\*

\* The following buffers are used in this example to send and receive data

\* with the UartLite.

\*/

u8 SendBuffer[TEST\_BUFFER\_SIZE]; /\* Buffer for Transmitting Data \*/

u8 RecvBuffer[TEST\_BUFFER\_SIZE]; /\* Buffer for Receiving Data \*/

**int** **main**(**void**) {

**int** Status;

**void** **getLocation**(u8 Recv[TEST\_BUFFER\_SIZE]);

/\*

\* Run the UartLite polled example, specify the Device ID that is

\* generated in xparameters.h

\*/

Status = XUartLite\_Initialize(&UartLite\_0, UARTLITE\_DEVICE\_ID\_0);

**if** (Status != XST\_SUCCESS) {

**return** XST\_FAILURE;

}

**if** (Status != XST\_SUCCESS) {

xil\_printf("Uartlite polled Example Failed\r\n");

**return** XST\_FAILURE;

}

Status = XUartLite\_Initialize(&UartLite\_1, UARTLITE\_DEVICE\_ID\_1);

**if** (Status != XST\_SUCCESS) {

**return** XST\_FAILURE;

}

**if** (Status != XST\_SUCCESS) {

xil\_printf("Uartlite polled Example Failed\r\n");

**return** XST\_FAILURE;

}

xil\_printf("Successfully ran Uartlite polled Example\r\n");

//XUartLite\_Send(&UartLite\_0, SendBuffer, TEST\_BUFFER\_SIZE);

**int** temp = 80000;

**int** simplecounter = 0;

**char** links[] = "DIGILENT DIGILENT\n\0";

**char** linksx[] = "ARTY ARTY ARTY ARTY\n\0";

**while** (1) {

**if** (1) {

//xil\_printf("Cha Cha Cha.... %d\r\n", simplecounter++);

temp = 80000;

XUartLite\_Recv(&UartLite\_1,RecvBuffer,1);

xil\_printf("%s", RecvBuffer);

//getLocation(RecvBuffer);

//xil\_printf("Latititude: %s\n", lat);

//xil\_printf("Longitude: %s\n", lng);

}

}

**return** XST\_SUCCESS;

}

**bool** **split**(u8 Recv[TEST\_BUFFER\_SIZE]){

**char** str[100];

**sprintf**(str,"%s",Recv);

**char** \*ptr; // declare a ptr pointer

ptr = **strtok**(str, " , "); // use strtok() function to separate string using comma (,) delimiter.

//cout << " \n Split string using strtok() function: " << endl;

// use while loop to check ptr is not null

**while** (ptr != NULL)

{

//cout << ptr << endl; // print the string token

**if** (**strcmp**(ptr,"$GPGGA"))

**return** **true**;

ptr = **strtok** (NULL, " , ");

}

**return** **false**;

}

**void** **getLocation**(u8 Recv[TEST\_BUFFER\_SIZE]){

**int** i=0;

**if**(split(Recv)){

**char** str[100];

**sprintf**(str,"%s",Recv);

**char** \*ptr; // declare a ptr pointer

ptr = **strtok**(str, " , "); // use strtok() function to separate string using comma (,) delimiter.

//cout << " \n Split string using strtok() function: " << endl;

// use while loop to check ptr is not null

**while** (i < 4)

{

**if**(i = 1)

**strcpy**(lat,ptr);

**if**(i = 3)

**strcpy**(lng,ptr);

ptr = **strtok** (NULL, " , ");

i++;

}

}

}

Iar output-ul generat de acest cod este in poza de mai jos:

Text

Description automatically generated

Cum s-a mentionat si mai sus, Informatiile de care avem nevoie din output se afla pe linia cu GPGGA care este al treilea de sus in jos, fiind prost generat deoarece placuta nu prindea semnal in momentul in care s-a realizat screenshot-ul. In mod normal, o linie GPGGA ar arata ca in poza atasata mai jos.



# 4.Bibliografie:

* <https://docs.xilinx.com/v/u/en-US/pg142-axi-uartlite>
* <https://forum.digilent.com/topic/4753-multiple-uartlite-instantiation-w-microblaze/>
* <https://github.com/Xilinx/embeddedsw/tree/master/XilinxProcessorIPLib/drivers/uartlite/examples>
* <https://support.xilinx.com/s/question/0D52E00006nbR1MSAU/implementation-fails-with-file-or-directory-does-not-exist-on-windows-10-due-to-windows-user-name-including-a-space?language=en_US>
* <https://www.youtube.com/watch?v=vgGKC3DPGCk&list=PLrkc7rkl1XyjmpqShS31FeF9ZKmyMT8-A&index=1>