



**TECHNICAL
UNIVERSITY**
OF CLUJ-NAPOCA
ROMANIA

FACULTATEA: Automatică și Calculatoare
SPECIALIZAREA: Calculatoare și Tehnologia Informației
DISCIPLINA: Structura sistemelor de calcul
PROIECT: Controler UART

Îndrumător proiect:

Fati Daniela

Student:

Galiș George-Laurențiu



Cuprins

1. Specificație și schema bloc.....	3
• Schema bloc generala	
2. Interfata UART.....	4
3. Schema bloc finala a automatului.....	6
4. Componentele automatului.....	7
• Transitator	
• Receptor	
• Debouncer	
5. Mediul de programare folosit.....	11
• Vivado	
• TeraTerm	
6. Testarea pe placuta de dezvoltare Basys3.....	13
7. Concluzie.....	14
8. Webografie.....	14

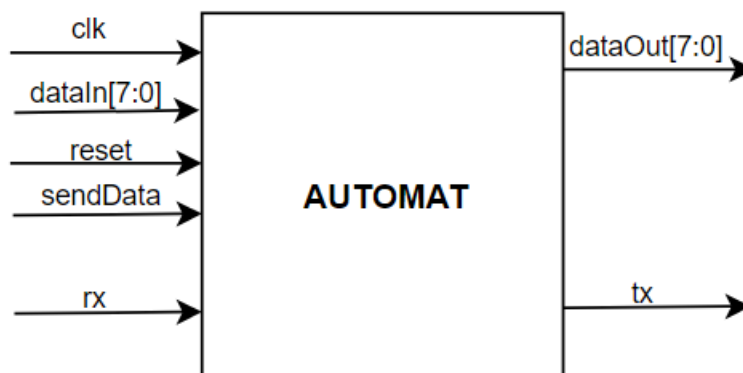
1. Specificație

Proiectul ales are ca cerinta crearea unui controler ce poate face posibila comunicarea unei placute ce contine circuitul integrat FPGA(field-programmable gate array) de la compania Xilinx cu calculatorul prin portul serial. Majoritatea porturilor seriale au la baza interfata UART. Nu este singura interfata ce poate face posibila comunicarea seriala. Pe langa interfata UART mai sunt si interfetele SPI(Serial Peripheral Interface), Microwire, I2C(Inter-Integrated Circuits) si SMBus(System Management Bus).

Cel mai mare avantaj al folosirii interfetei UART este ca fata de celelalte interfete mentionate mai sus, aceasta este o interfata seriala **asincrona**.

Schema bloc

O schema bloc de baza ce reprezinta intrarile si iesirile automatului este urmatoarea.



Din punct de vedere fizic, pe placuta, automatul are urmatoarele intrari si iesiri:

Intrari:

- clk - frecventa de ceas ce este prezenta pe placuta
- dataIn[7:0] - switch-urile care transmit datele la calculator
- reset - un buton care reseteaza afisarea datelor trimise de la calculator
- send data - un buton care atunci cand este apasat, stramite datele la calculator
- rx - tine cont de conectarea la un fir intern ce face posibila transmiterea seriala

Iesiri:

- dataOut[7:0] – ledurile de pe placuta ce afiseaza ce date sunt transmise de la calculator
- tx - tine cont de conectarea la un fir intern ce face posibila transmiterea seriala

2. Interfata UART

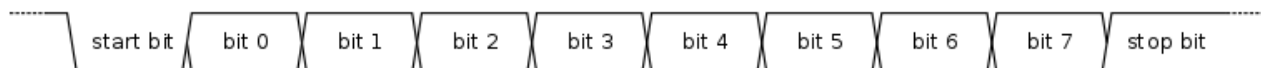
Transmisia de date seriala in tehnologia informatiei, este o tehnologie de comunicatie intre calculator și dispozitivele periferice, in care bitii de date sunt transferati succesiv de-a lungul unui canal de comunicare sau la. Transmisiile seriale de date au la baza transmiterea pachetelor de informații printr-un mediu de transmisie a semnalelor electrice (în general cablu coaxial) într-un singur șir sau serial. Pentru interconectarea dispozitivelor ce constituiau rețeaua nu este necesara folosirea unui switch, conexiunea făcându-se de la un dispozitiv la celalalt, iar la capatul sirului se monteaza un dispozitiv terminal pentru adaptarea de impedanta.

O interfata UART(Universal Asynchronous receiver-transmitter) este o componenta hardware pentru comunicarea serial asincrona in care **datele si viteza de transmitere sunt configurabile**. Acesta trimite datele bit cu bit de la cel mai puțin la cel mai semnificativ bit încadrate de **bitii de start si de stop** care reprezinta "capetele" unei linii de date care urmeaza sa fie transmisa.

In transferul asincron, emitatorul și receptorul au semnale de tact separate, sincronizarea la nivel de bit este asigurata numai pe durata transmisiei efective a fiecarui caracter. O asemenea comunicatie este orientata pe caractere individuale și are dezavantajul că necesită informații suplimentare în proporție de cel puțin 25% pentru identificarea fiecărui caracter. De aceea acest mod de comunicație este numit și start-stop.

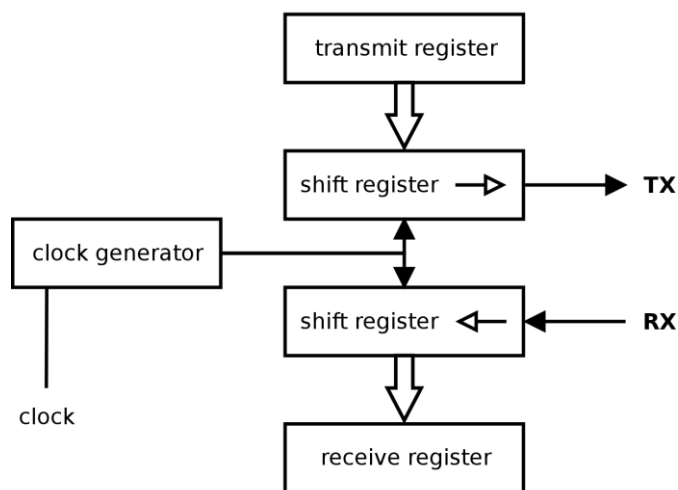
Controlerul UART se ocupa de comunicarea seriala asincrona dintre un computer și un dispozitiv periferic conectat la portul serial al calculatorului și converteste datele de la serial la paralel si invers. Seria de chipuri 16550 este UART-ul cel mai frecvent utilizat.

Impartirea bitilor ce formeaza datele sunt reprezentate in figura de mai jos:



Interfata UART ia octetii de date si ii transmite ca biti individuali intr-o maniera secventiala. La destinatie, un al doilea UART reasambleaza bitii in octeti completi. Fiecare interfata contine un registru de deplasare. Comunicarea poate fi de tipul **simplex**(doar intr-o singura directie), **full duplex**(ambele dispozitice trimit si primesc date in acelasi timp) si **half duplex**(dispozitivele transmit si primesc datele pe rand).

O schema bloc generala a interfetei UART este reprezentata mai jos



Interfata UART implementata de mine contine doua mari componente: **transmitator (transmitter) si receptor (receiver)**. Transmitatorul si receptorul **trebuie** sa aiba acelasi interval de timp corespunzator unui bit, lungime de caracter, bit de paritate si pozitia bitilor de start si de stop. Majoritatea porturilor seriale folosite la calculatoare au 8 biti de date, nu au paritate, si un bit de start si de stop

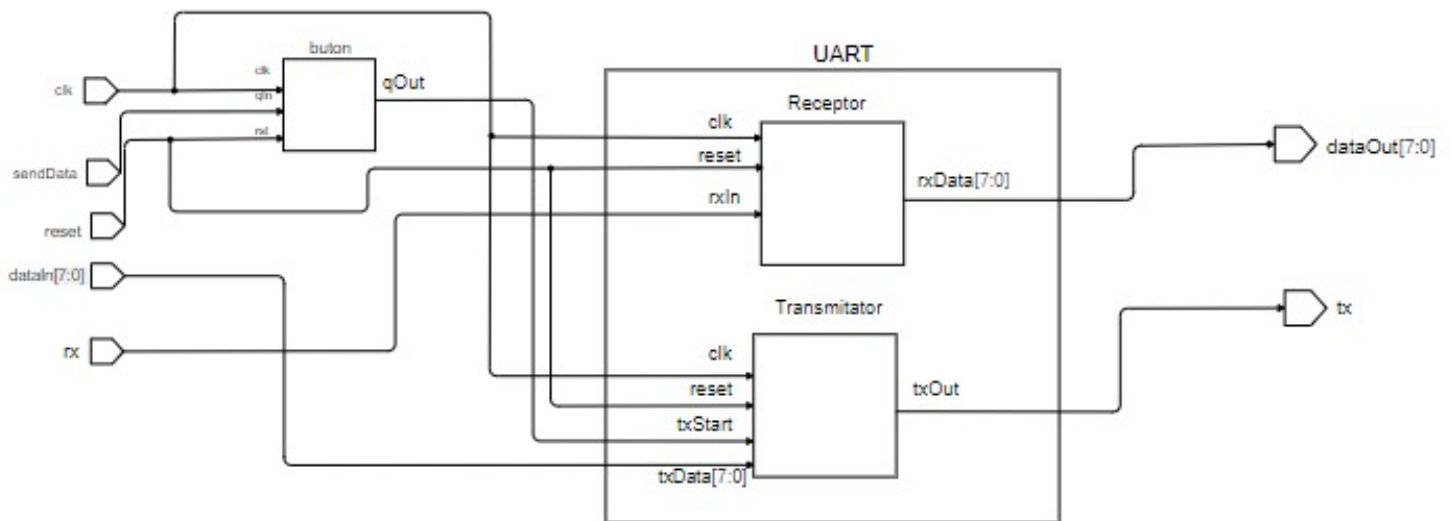
Aceste doua componente vor fi explicate in una din sectiunile urmatoare.

3. Schema bloc finala a automatului

Componentele care se afla in interiorul automatului sunt:

- buton(debouncer) – permite transmiterea unui singur impuls atunci cand un buton e apasat. Atunci cand se apasa un buton pe placuta, se transmit mai multe semnale, dar noi avem nevoie doar de unul singur
- transreceptor(UART) – este format din doua componente: transmitator si receptor. Acesta face posibila transmiterea si primirea datelor de la calculator prin portul serial.

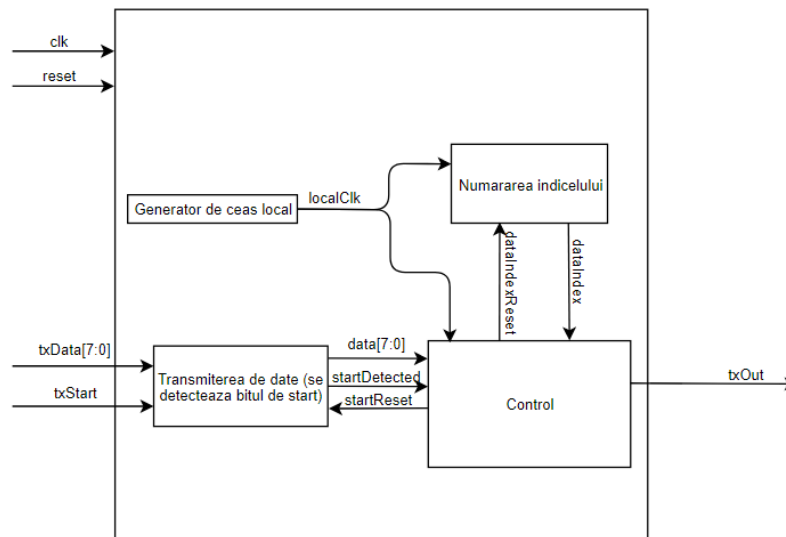
Schema bloc finala a automatului este reprezentata in figura de mai jos:



4. Componentele automatului

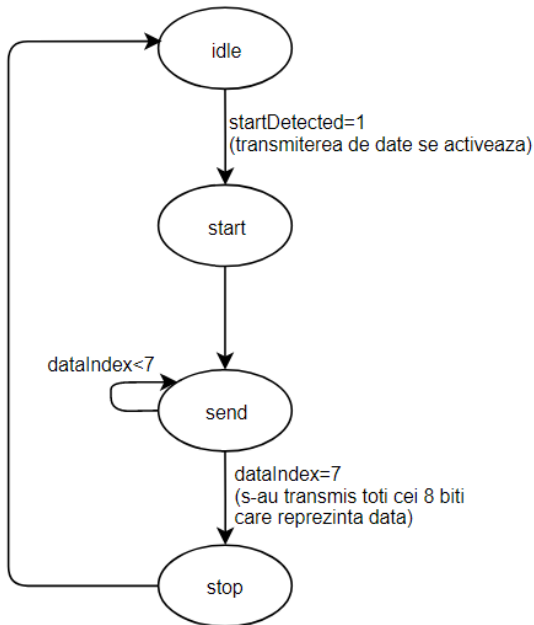
Transmitator

Transmitatorul are ca scop trimiterea datelor prin interfata UART. De exemplu, cand se apasa butonul care face "enable"(sendData) la transmiterea datelor de pe switch-uri, acestea trec prin aceasta componenta si face posibila transmiterea lor la calculator prin portul serial. Datele transmise au o lungime fixa de 8 biti plus doi biti in plus care reprezinta bitii de start si de stop. Bitul de **start** este reprezentat ca fiind '0' logic si bitul de **stop** este reprezentat ca fiind '1' logic. O schema bloc a transmitatorului este reprezentata in imaginea de mai jos:



Componentele care formeaza transmitatorul sunt:

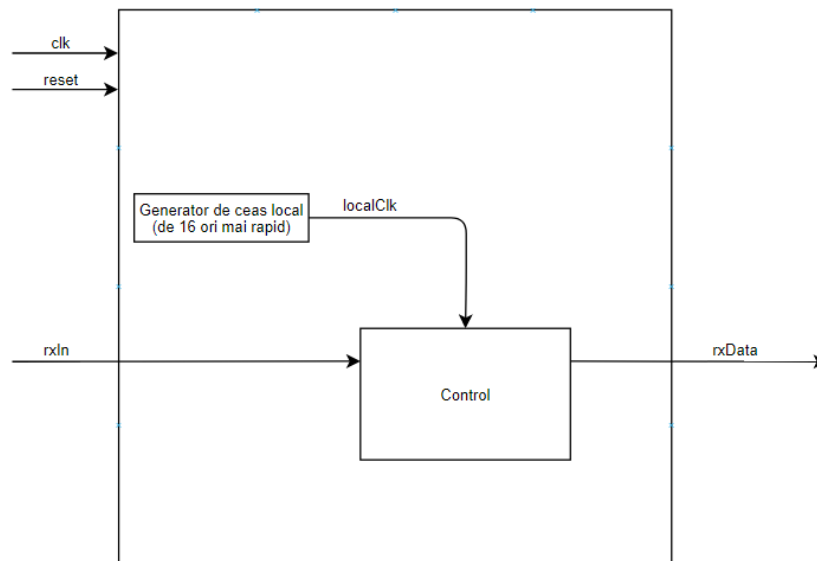
- Generatorul de ceas local – genereaza o frecventa "locala" care permite masurarea intervalului de timp corespunzator unui bit
- Numararea indicelui – este un numarator simplu care numara in intervalul 0-7 penru a determina lungimea datelor care se transmit prin interfata UART. Semnalul dataIndex este folosit in Control pentru a larcure vectorul data si sa trimita bitii unun cate unul
- Transmiterea de tare – reprezinta un registru care transmite datele in Control atunci cand butonul e apasat(txStart). Semnalul startDetected este folosit atunci cand se detecteaza bitul de start '0' si startReset atunci cand se reincepe numararea bitilor(toti cei 8 biti)
- Control – reprezinta unitatea de control a transmitatorului care are urmatoarea diagrama de stare:



- **Starea idle:** reprezinta starea in care se initializeaza semnalele, mai exact dataIndexReset <= '1'; startReset <= '0'; txOut <= '1'. Tot in aceasta stare se asteapta detectarea bitului de start adica startDetected='1' pentru a incepe transmiterea datelor si trecerea in starea "start"
- **Starea start:** reprezinta starea in care activez numararea bitilor adica dataIndexReset<='1' si trimit 0 ca bit de start adica txOut<='0'
- **Starea send:** reprezinta starea in care trimit pe txOut cate un bit din vectorul data. Indexarea bitilor se face cu ajutorul semnalului dataIndex. Tot acest procedeu se realizeaza pana cand dataIndex este mai mic decat 7(adica se transmit exact 7 biti). Atunci cand s-au transmis toti bitii de date se trece in stare de stop
- **Starea stop:** reprezinta starea in care trimit pe txOut '1' logic ce reprezinta bitul de stop si fac semnalul startReset<='1' pentru a pregati citirea urmatorului octet de date.

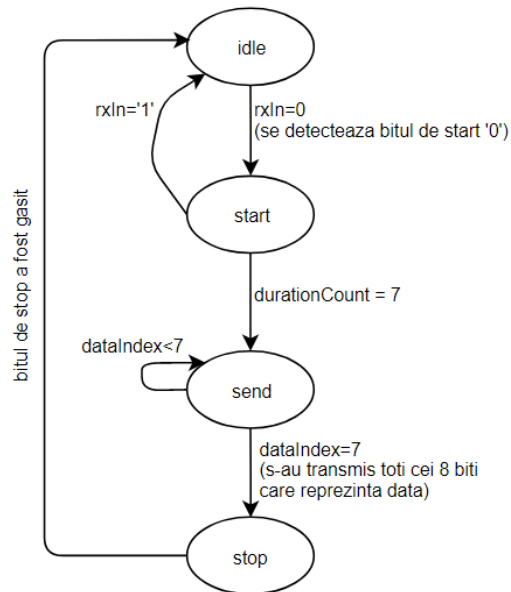
Receptor

Receptorul are ca scop primirea datelor prin semnalul rxIn si transformarea lor in vectori cu dimensiunea de 8 biti(rxData). Afisarea datelor care sunt primite de la calculator in receptor sunt afisate pe cele 8 leduri ale placutei pentru a determina functionalitatea componentei receptor. O reprezentare a acestei componente poate fi vazuta in imaginea de mai jos



Componentele receptorului sunt

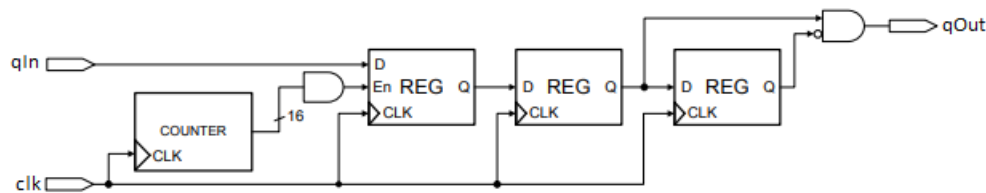
- Generatorul de ceas local – genereaza o frecventa ”locala” care permite masurarea intervalului de timp corespunzator unui bit. Acesta este impartit la valoarea 16 pentru a putea fi de 16 ori mai rapid
- Control – reprezinta unitatea de control a transmitatorului care are urmatoarea diagrama de stare:



- **Starea idle:** reprezinta starea in care se initializeaza semnalele, data <= (others => '0')(curat "registrul" in care se incarca datele primite) durationCount := 0; dataIndex := 0 (resetez numararea bitilor)
- **Starea start:** reprezinta starea in care activez numararea bitilor adica cat timp rxIn este '0'
- **Starea send:** reprezinta starea in care completez in "registrul" in care se primesc datele, bitul primit (rxIn). Atunci cand s-au incarat toti bitii de date se trece in stare de stop
- **Starea stop:** reprezinta starea in care astept pana la terminarea unui ciclu de ceas(localClk) si afizez datele primite(adica vectorul data)

Debouncer

Debouncer-ul are ca scop formarea unui singur impuls la apasarea unui buton. Cand se apasa un buton de pe placuta, oricat de scurta ar fi apasarea, acesta genereaza mai multe semnale. Asadar, chiar daca se tine apasat pe buton, aceasta componenta va genera un singur semnal. Acest lucru face interactionarea cu placuta functionala. O schema bloc a debouncer-ului ar putea fi aceasta:



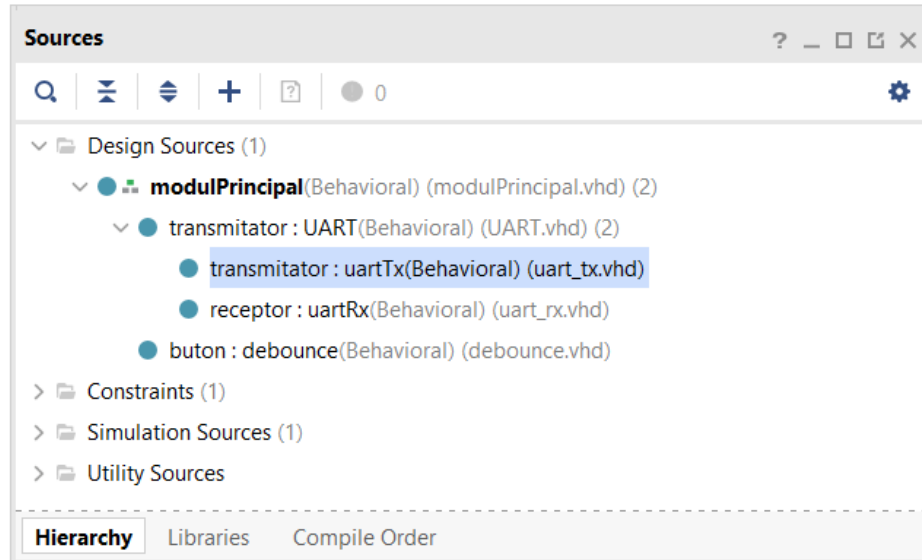
Rolul primului registru, impreuna cu numarul, este de a asigura robustețe la utilizarea butoanelor uzate fizic, cand pot să apară activări multiple ale semnalului qOut la o apăsare de buton. În funcție de uzura, este posibil sa fie nevoie de mai mulți biti ai numaratorului (17-20+) pe care sa se aplice un SI logic, astfel incat să se mareasca intervalul de esantionare al butonului.

5. Mediul de programare folosit

Vivado

Programul folosit pentru implementarea și încărcarea acestuia pe placuta de dezvoltare Basys3 se numește Vivado. Mediul de proiectare Xilinx Vivado Design Suite integrează toate utilitățile necesare pentru proiectarea sistemelor digitale utilizând circuitele FPGA și multe altele. Interfața grafică Vivado IDE gestionează fișierele sursă incluse în proiect, memorează rezultatele lăurii, generează rapoarte ale proiectului și salvează în mod automat starea proiectului.

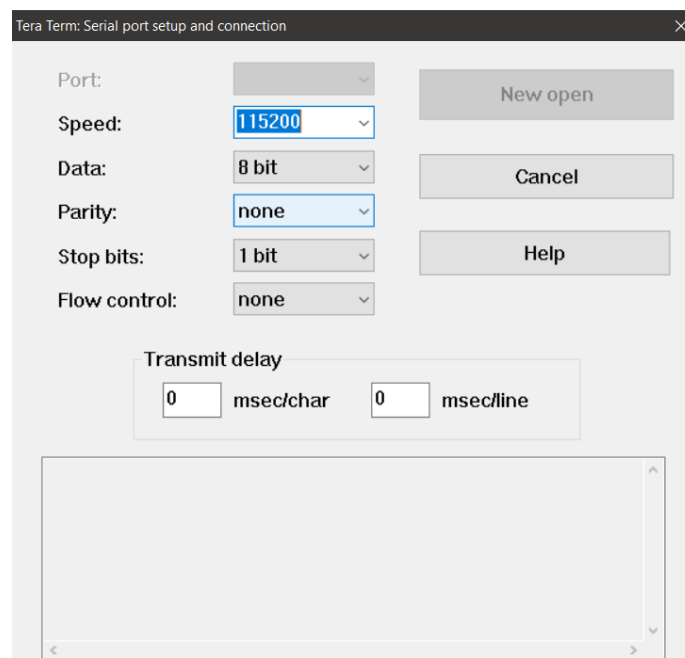
Asadar, fișierele scrise vor fi așezate în mod ierarhic în funcție de scrierea lor. Ierarhia fișierelor poate fi văzută în imaginea de mai jos.



Fisierul modulPrincipal contine fisierele UART si buton. Fisierul UART contine la randul lor fisierele uartTx si uartRx. Acestea sunt asezate in acest mod deoarece in fiecare fisier sunt declarate cu intanta **port map**(de ex, buton: entity WORK.debounce port map(clk => clk rst => reset, qIn => sendData, qOut => enableTx);)

TeraTerm

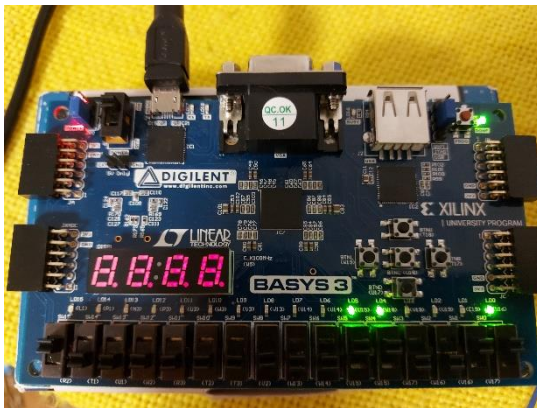
Programul folosit pentru vizualizarea si transmiterea datelor la placuta se numeste TeraTerm. Acest program citeste datele de pe portul serial si le afiseaza sub forma unui termina. Setarile care se fac in acest program sunt urmatoarele. Se modifica viteza la 11520 deoarece implementarea UART a folosit aceasta viteza (generic(rate: integer := 115_200);)



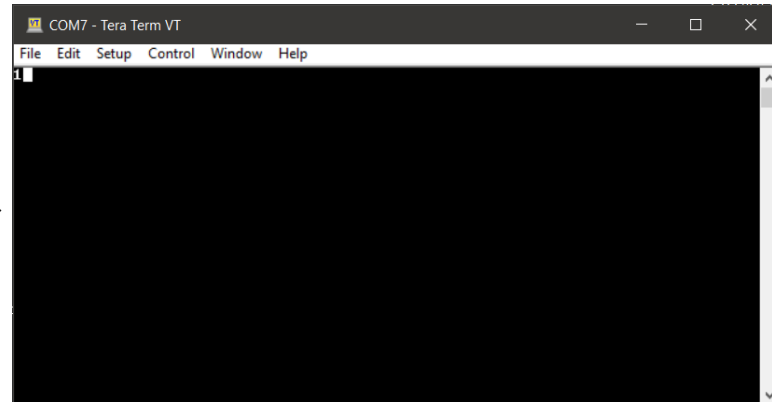
6. Testarea pe placuta de dezvoltare Basys3

Pentru a putea testa implementarea facuta este nevoie de o placuta Basys3 si de programul TeraTerm. Interfat UART este determinata de functionalitatea transmitatorului si a receptorului.

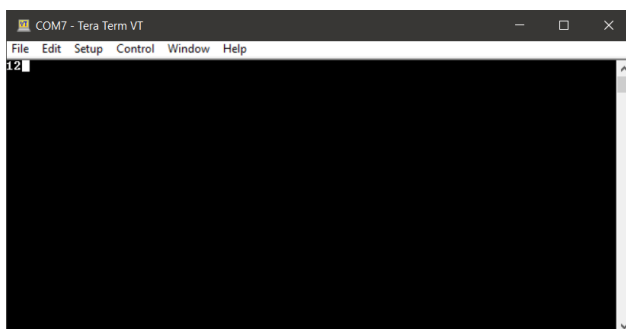
In primul rand pentru a verifica functionalitatea transmitatorului se introduce ce date se doreste sa se transmita prin portul serial si sa se afiseze in terminalul TerraTerm. La calculator o sa se transmita un caracter(adica valoarea caracterului in codul ASCII). De ex daca introduc pe switch-uri bitii "0011 0001"(adica 31 in hexazecimal, adica valoarea caracterului '1') si se apasa pe butonul care trimite datele prin interfata UART, in terminal o sa se afiseze caracterul trimis prin interfata.



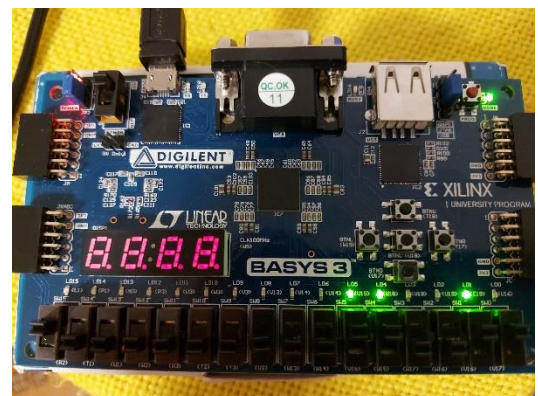
dupa apasare →



In al doilea rand pentru a verifica functionalitatea receptorului se introduce in terminal un caracter ce se doreste a fi citit pe led-urile placutei. De exemplu daca se introduce in terminal caracterul '2', pe placuta o sa se afiseze codul ascii al caracterului (0011 0010, adica 32 in hexazecimal)



dupa introducerea caracterului '2' →





7. Concluzie

În concluzie, implementarea interfeței UART a făcut mult mai ușoară înțelegerea și dezvoltarea transmițătoarelor seriale, cum se transmit datele prin fiecare port existent, ce tipuri de date se transmit și în ce mod sunt formate aceste date. Mai mult de atât a făcut aprofundarea limbajului de programare VHDL.

8. Webografie

<https://www.wikipedia.org/>

<https://www.youtube.com/>

<https://www.circuitbasics.com/>