Tema 4 – Microblaze

Digori Gheorghe

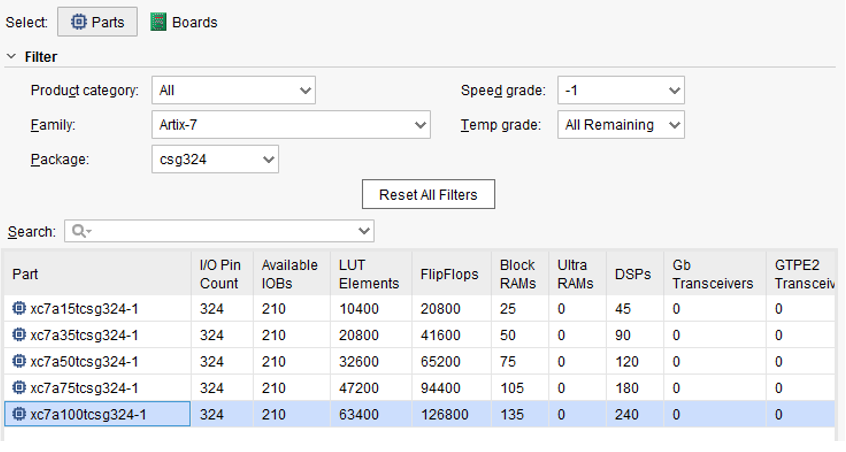
Neciu Laurentiu-Florin

331CC

Scopul lucrarii:

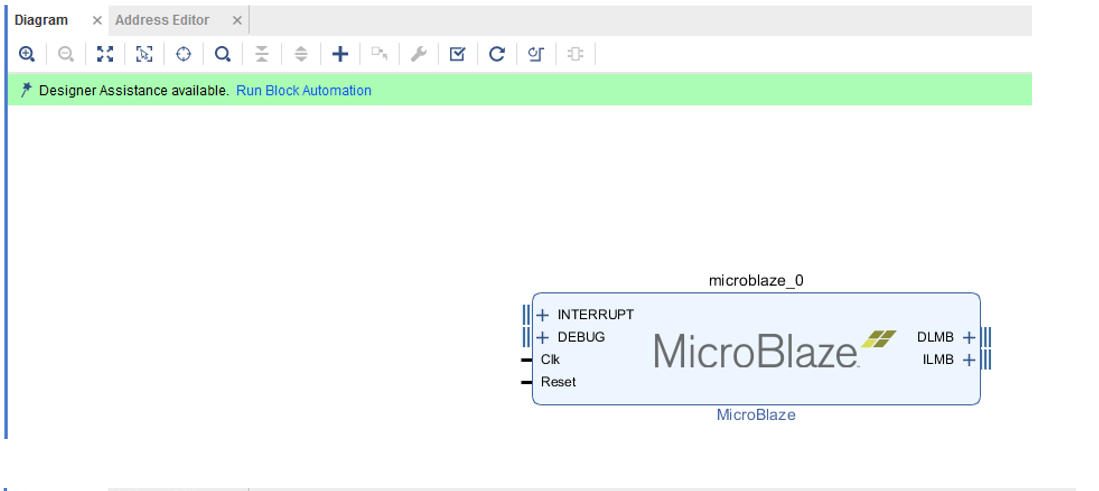
Scopul lucrarii de fata este implementarea unui program (simplistic, pentru testare) scris in Limbajul C pe o placa FPGA folosing modulul Microblaze.

Initial vom crea un proiect cu specificatiile din poza de mai jos.



Imediat cum am selectat create block design se va deschide o fereastra “diagram”. In cadrul acelei ferestre se apasa butunul “+”si se selecteaza Microblaze.

Va apărea componenta în diagramă. Ulterior, selectăm “Run block automation”. Setam memoria locala la 32KB, Debug la “Debug & UART” și Axi port la “Enabled”, după care confirmam setarile apasand pe butonul OK. Observam ca ne apare o diagrama ca in poza.



Programul Vivado ne da posibilitate de a adauga anumite secțiuni folosind apasand pe butoanele “+” din diagramă.

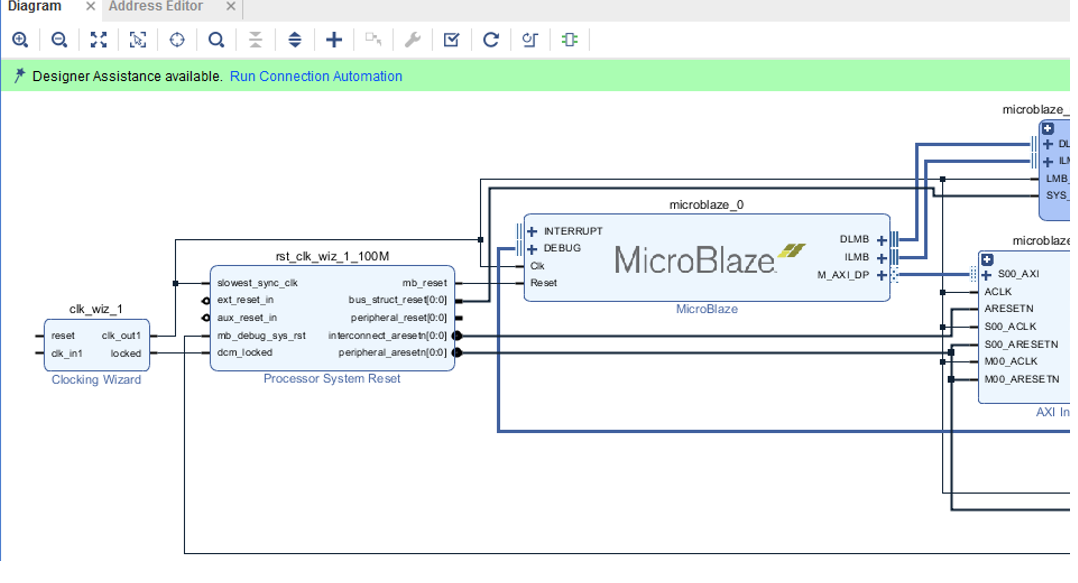
Se executa click drepata pe spațiul gol al diagramei și dăm “regenerate layout”.

Facem dublu click în diagramă pe componenta clocking wizard. În “input clock information” la primary -> source vom modifica in Single ended non clock pin.

Rulăm “Run clock automation”. Se bifeaza CLK\_in1.

Apăsăm din nou pe run connection Automation, și setăm reset-ul pe active\_high. Setăm ext\_reset\_in pe active\_low.

Salvăm setarile facute pana acum.

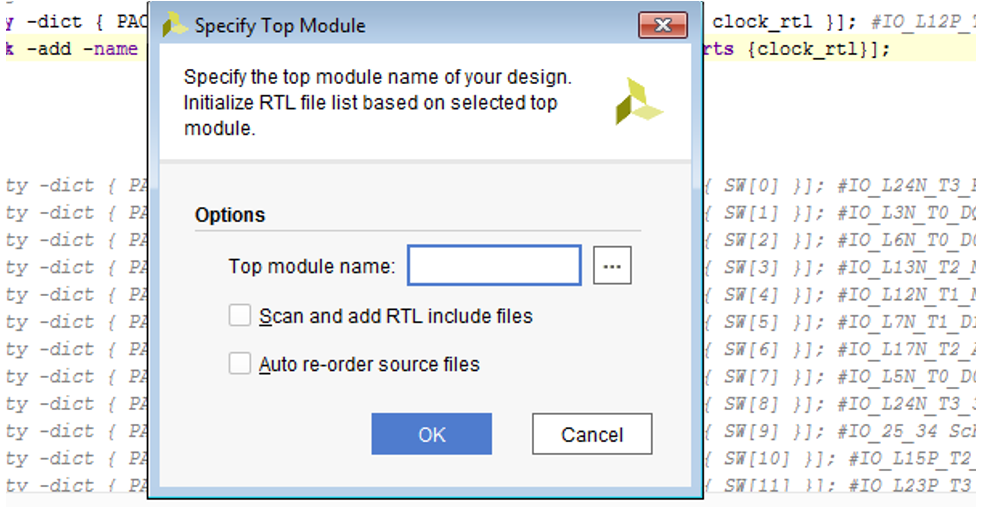


Deschidem fișierul și decomentăm primele doua linii de la “Clock”. Se inlocuieste “CLK100MHZ” cu “clock\_rtl”. Vom decomenta si liniile Switch > J15 > reset\_rtl Button > N17 > reset\_rtl\_0.

Aceasta decomentare are rolul de la a seta butonul de reset pe N17. Va trebui sa tinem apasat pe acest buton cand vom programa placa.

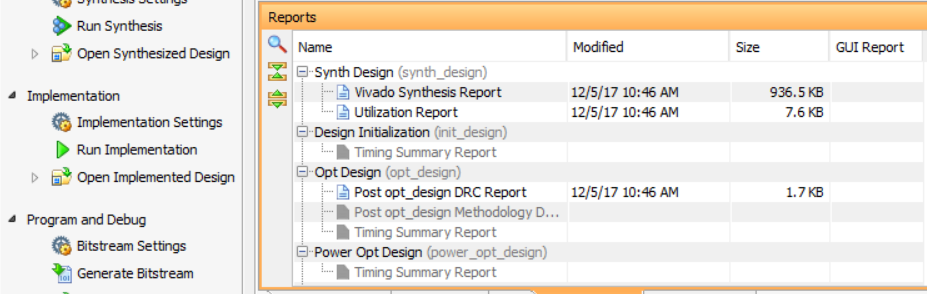
Salvăm fisierul.

Rulăm “Run Synthesis”, și introducem numele modulului Top (in laborator l-am denumit design\_1). După ce termină se selecteaza view reports.



Dupa, ne ducem la Synthesis si in Open Elaborated Design.

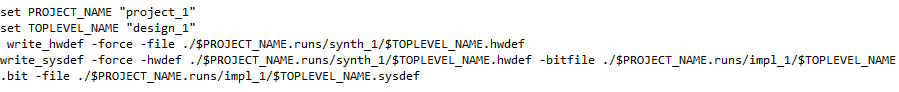
Mergem în colțul din sus-drepata, Default Layout , IO Planning , FPGA. Intrăm în TCL Console și verificăm dacă toate porturile reset au opțiunea Fixed bifată.

După, se selecteaza “Generate bitstream”. Selectăm “Open Implemented design” și apăsăm pe Yes. 

Din sources, selectăm design\_1, Facem click dreapta si alegem “Creade HDL wrapper”. Selectăm design\_1, click dreapta și generate output products.

Intram in File, Export, Export hardware (și includem bitsteam), Ok.

Va da o eroare. Pentru remediere avem în owncloud un fișier care conține comenzile necesare pentru a remedia eroarea.

Odata ce am rezolvat eroarea, putem sa exportam deisgn-ul in SDK-ul xilinx in care vom scrie un program in C++.

Pentru a exporta intram in:

File, Launch SDK.

Ni se va deschide SDK-ul, daca acesta a fost instalat. Pentru a crea si rula o noua sursa helloworld pentru testare in limbajul C cu care vom programa placa FPGA, putem sa intram in urmatoarele

File, New, Application Project, helloworld, next & finish.

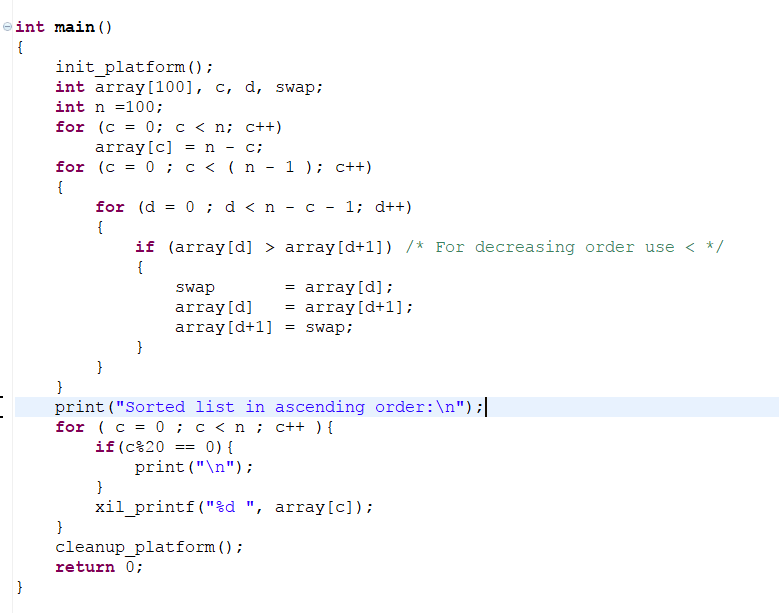
Pentru a rula codul, intram in helloworld , src , helloworld.c apoi in run , run configurations, Xilnix C++ app GDB.

Vom face acum o noua configuratie de run.

New Bsfile , select design1.bit Application , browse PR name (helloworld) STDIO Connection , Connect to stdio , PORT: JTAG (UART) Run.

Pentru a genera mesajul “Helloworld” pe ecran, se conecteaza placa la calculator prin USB, si dupa ce dam run tinem apasat pe butonul de reset care a fost desemnat ca fiind N17.

Acum vom implementa pe placa un algoritm de sortate, anume Selection sort. Algoritmul genereaza automat intr-un vector numere in ordine descendenta de la 100 la 1 si le sorteaza cf. algoritmului in ordine ascendeta. Codul este prezentat mai jos.



Iar rezultatul este urmatorul:

