

## I. Noțiuni introductive Fibonacci

Numerele Fibonacci sunt constituite ca o secvență definită astfel:

$$fib(i) = \begin{cases} 0 & \text{dacă } i = 0 \\ 1 & \text{dacă } i = 1 \\ fib(i-1) + fib(i-2) & \text{dacă } i > 0 \end{cases}$$

O metodă de a calcula  $fib(i)$  este construcția iterativă a funcției de la 0 la un  $i$  dorit/specificat.

## II. Implementare tradițională Fibonacci în FPGA

Pentru a realiza implementarea circuitului vom folosi pentru intrare/ieșire formatul BCD (a se vedea laboratorul de CN1). Presupunem că intrarea este un semnal de 8 biți în format BCD (vom avea 2 digiți BCD) și ieșirea este afișată ca 4 digiți BCD pe un display 7 segmente.

Mai mult, pe display-ul LED se va afișa „9999” dacă numărul Fibonacci rezultat este mai mare decât 9999 (overflow).

Operația poate fi realizată în 3 pași: conversia intrării către formatul binar, calcularea numărului Fibonacci și conversia rezultatului în format BCD.

Putem urma procedura:

1. Implementarea unui circuit de conversie BCD-binar
2. Modificarea circuitului Fibonacci pentru a include un semnal de ieșire pentru a indica condiția de overflow
3. Proiectarea diagramei bloc – top level
4. Scrierea codului Verilog/VHDL
5. Scrierea unui testbench și utilizarea simulatorului pentru realizarea unei verificări logice
6. Sinteza circuitului, programarea FPGA-ului Nexys4 DDR și verificarea funcționării

## III. Implementare Fibonacci folosind Microblaze

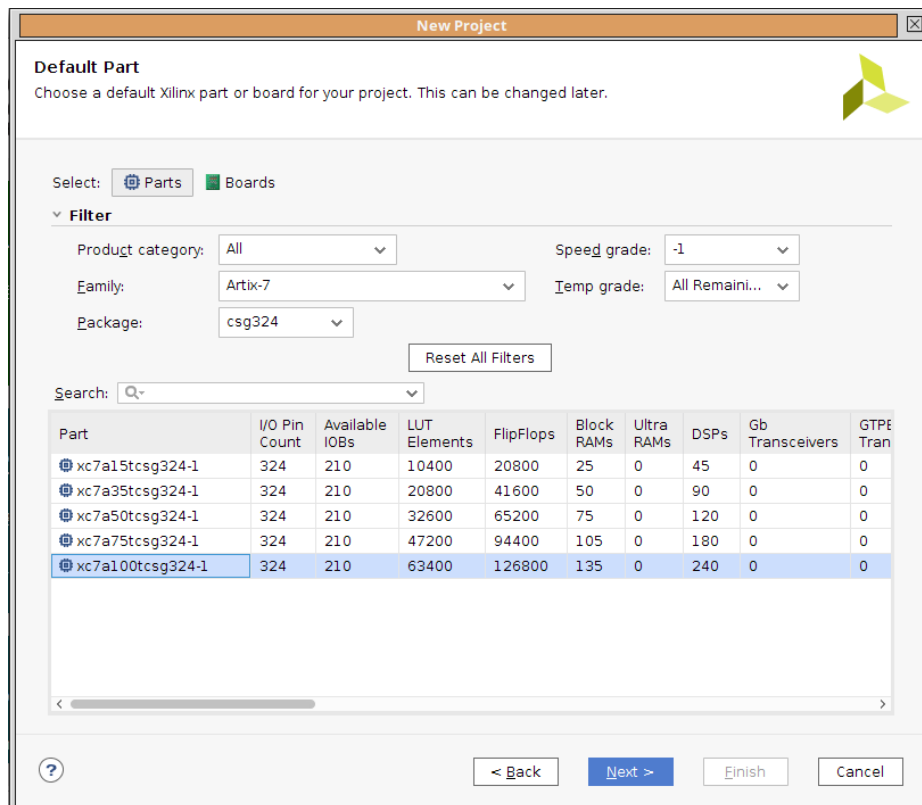
### a). Implementare MICROBLAZE în placa de dezvoltare Nexys4 DDR

*Pasul 1* – Crearea unui nou proiect

- Se va selecta locația noului proiect astfel încât în cale să nu avem spații
- Se va selecta opțiunea „Create project subdirectory”

*Pasul 2* – Se va selecta opțiunea „RTL Project” și opțiunea „Do not specify sources at this time”

*Pasul 3* – Se vor selecta parametrii plăcii de dezvoltare Nexys4 DDR conform figurii de mai jos și apoi se selectează opțiunea FINISH.

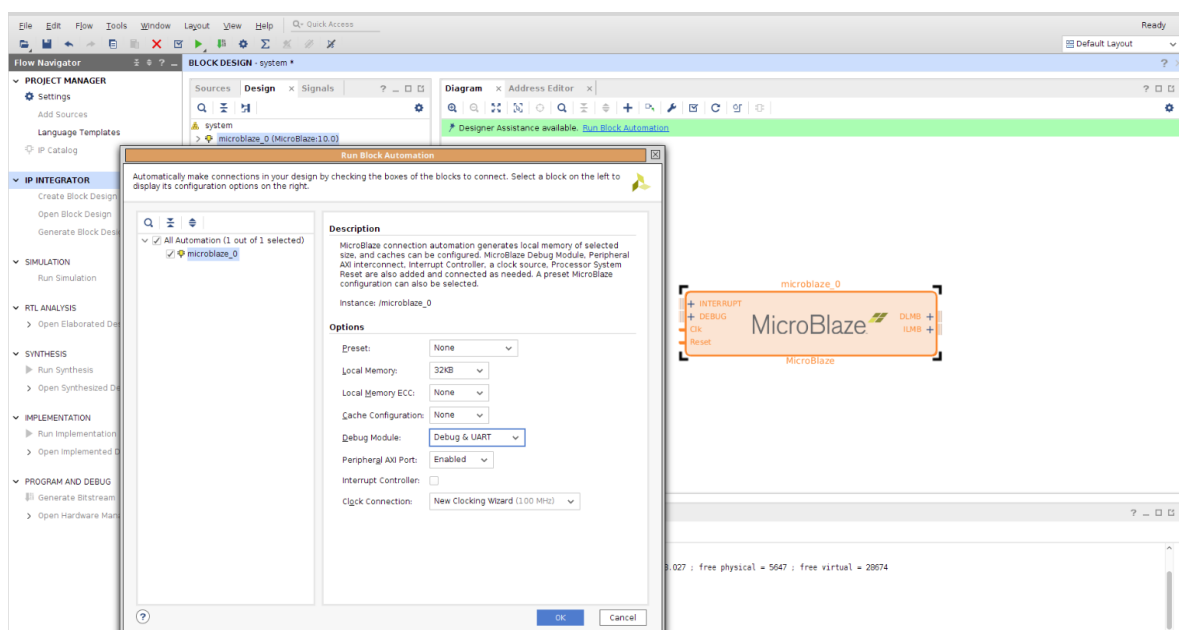


**Pasul 4** – Se va alege opțiunea „Create Block Design” prin dubluclick din fereastra **Flow Navigator**. În noua fereastră vom schimba numele din căsuța de dialog „Design name” cu **system** spre exemplu.

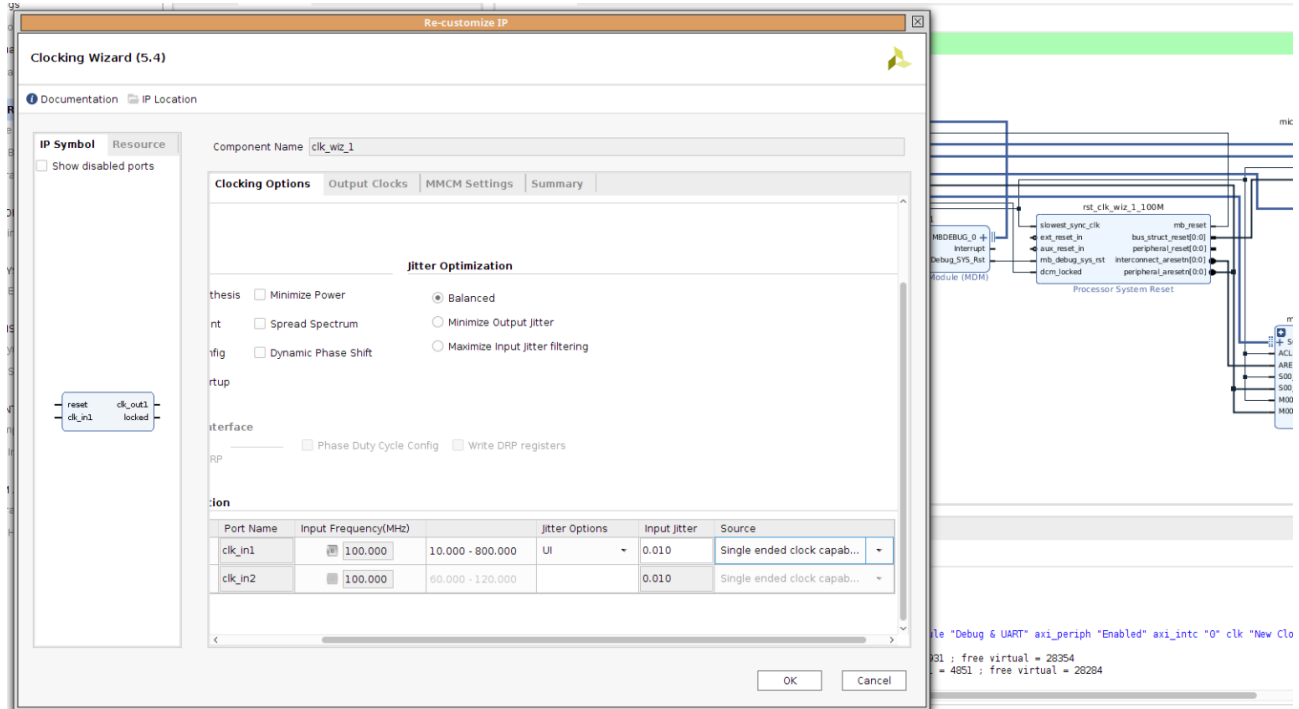
**Pasul 5** – În fereastra „Diagram” apăsăm butonul „+” pentru a adăuga un IP. În noua fereastră, în câmpul „Search” vom scrie microblaze.

**Pasul 6** – Vom selecta opțiunea „Microblaze” care reprezintă implementarea completă a procesorului Microblaze.

**Pasul 7** – Vom selecta opțiunea „Run Block Automation” și vom face modificările conform figurii următoare.



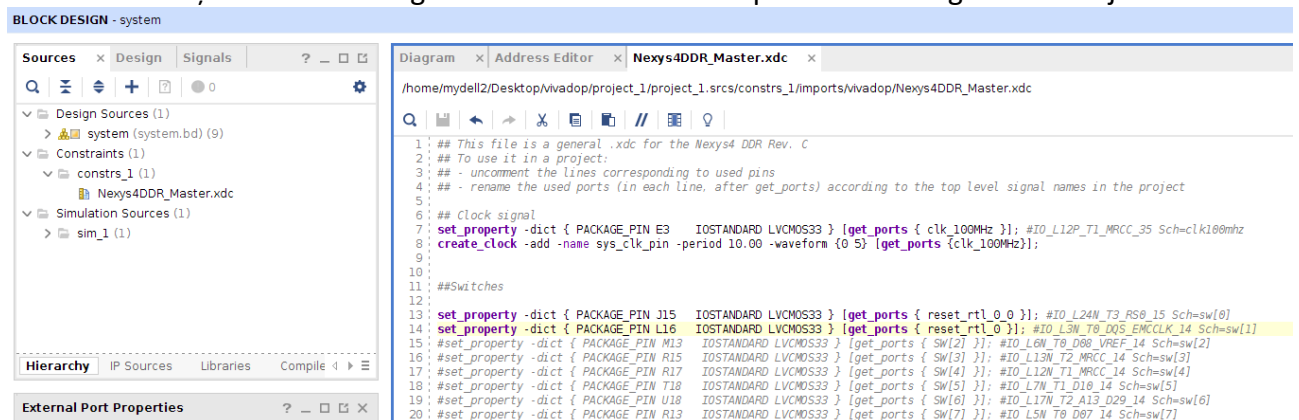
**Pasul 8** – Selectăm modulul „clk\_wiz\_1” printr-un dubluclick și facem o modificare pentru semnalul **clk\_in1**. În câmpul „Source” vom avea: **Single ended clock capable pin**. Restul opțiunilor rămân nemodificate



**Pasul 9** – Vom selecta din nou opțiunea „Run Block Automation”. În noua fereastră vom selecta „All Automation” și apoi OK. Acum au apărut 3 semnale noi: „clk\_100MHz”, „reset\_rtl\_0” și „reset\_rtl\_0\_0”

**Pasul 10** – În fereastra BLOCK DESIGN vom selecta tab-ul „Sources” și vom adăuga fișierul de constrângeri furnizat de producătorul plăcii. Este de dorit să se selecteze opțiunea „Copy constraints files into project”

**Pasul 11** – În fișirul de constrângeri vom face modificările prezentate în figura de mai jos.



**Pasul 12** – Se va selecta „system (system.bd)” din tab-ul „Sources”, click dreapta și alegem opțiunea „Create HDL Wrapper”. În noua fereastră vom selecta opțiunea „Let Vivado manage wrapper and auto-update”

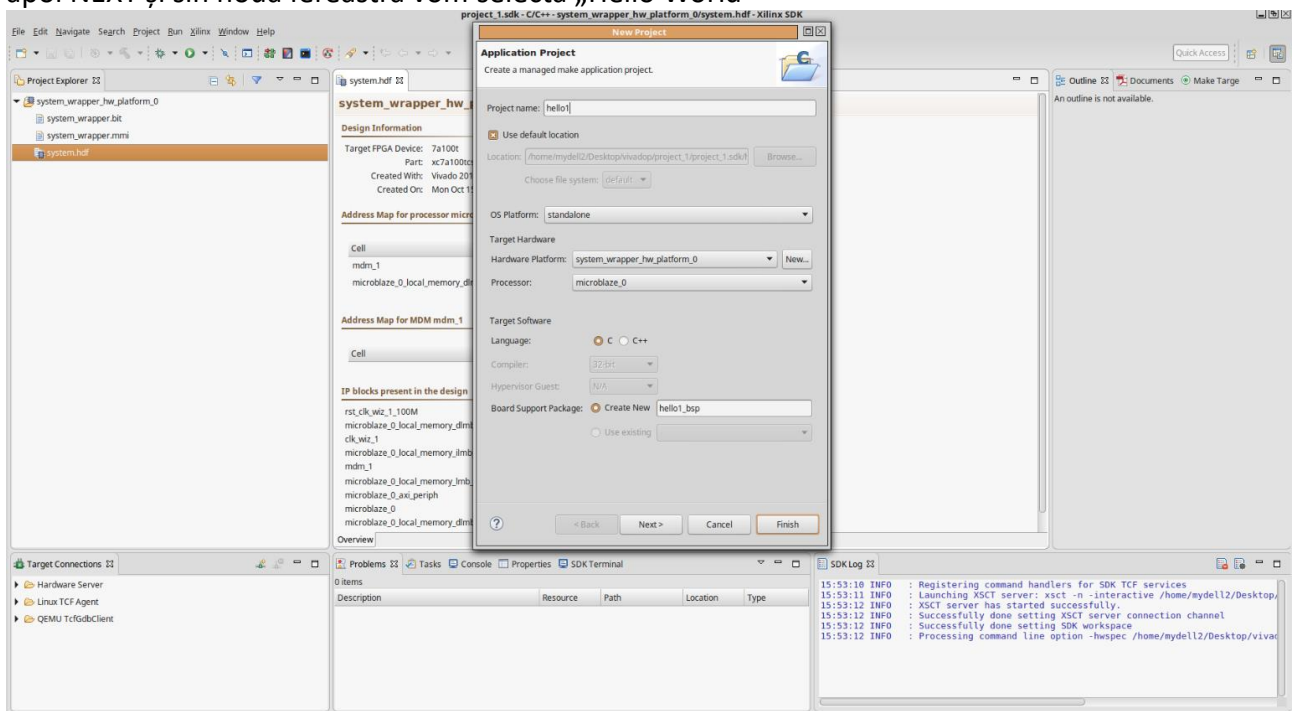
**Pasul 13** – Se va selecta din nou „system.bd”, click dreapta și alegem opțiunea „Generate Output Products ....”. În noua fereastră se va selecta un număr de job-uri cât mai mare și se va apăsa butonul **Generate**.

**Pasul 14** – Vom genera fișierul bitstream selectând opțiunea „Generate Bitstream” din fereastra **Flow Navigator**. Și în acest caz vom selecta un număr de job-uri cât mai mare.

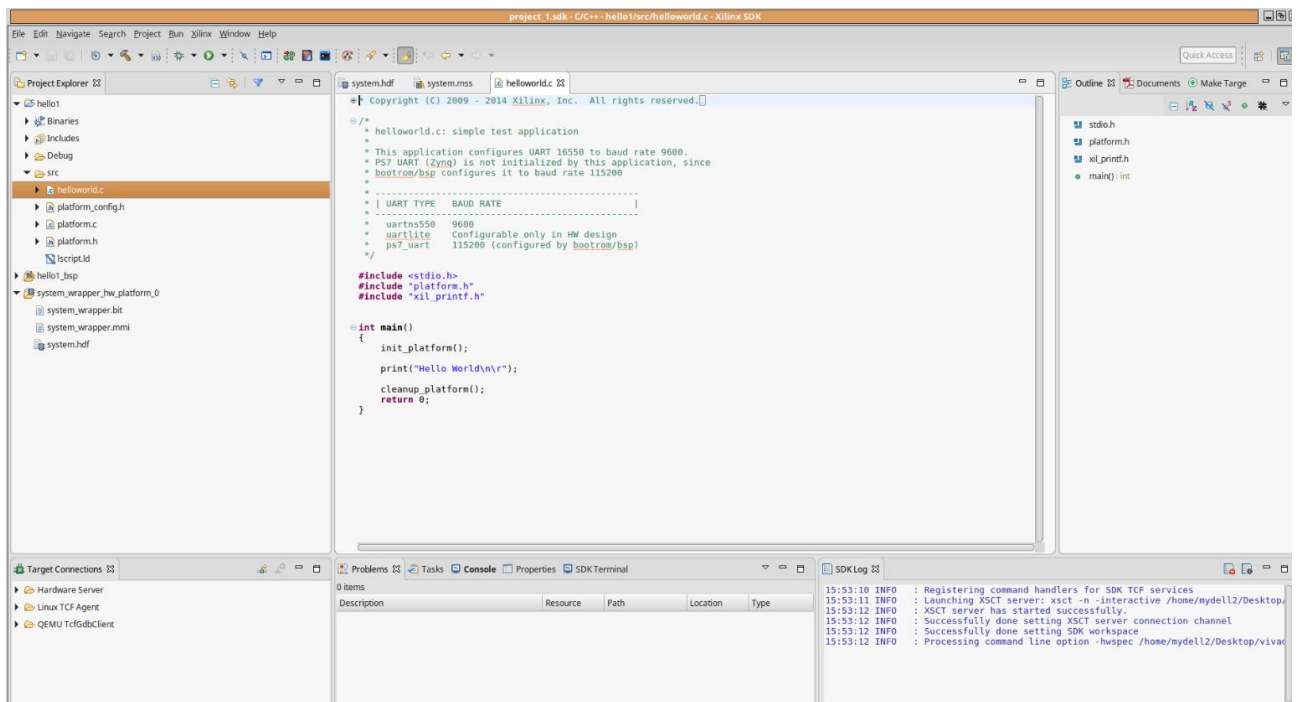
**Pasul 15** – Din meniul **File** vom selecta opțiunea Export și apoi opțiunea Export Hardware. Va trebui în noua fereastră selectată opțiunea „Include bitstream file”

**Pasul 16** – Tot din meniul **File** se va selecta opțiunea „Launch SDK”

**Pasul 17** – Din meniul **File** vom selecta opțiunea „New” și apoi „Application project”. Vom selecta apoi NEXT și sin noua fereastră vom selecta „Hello World”



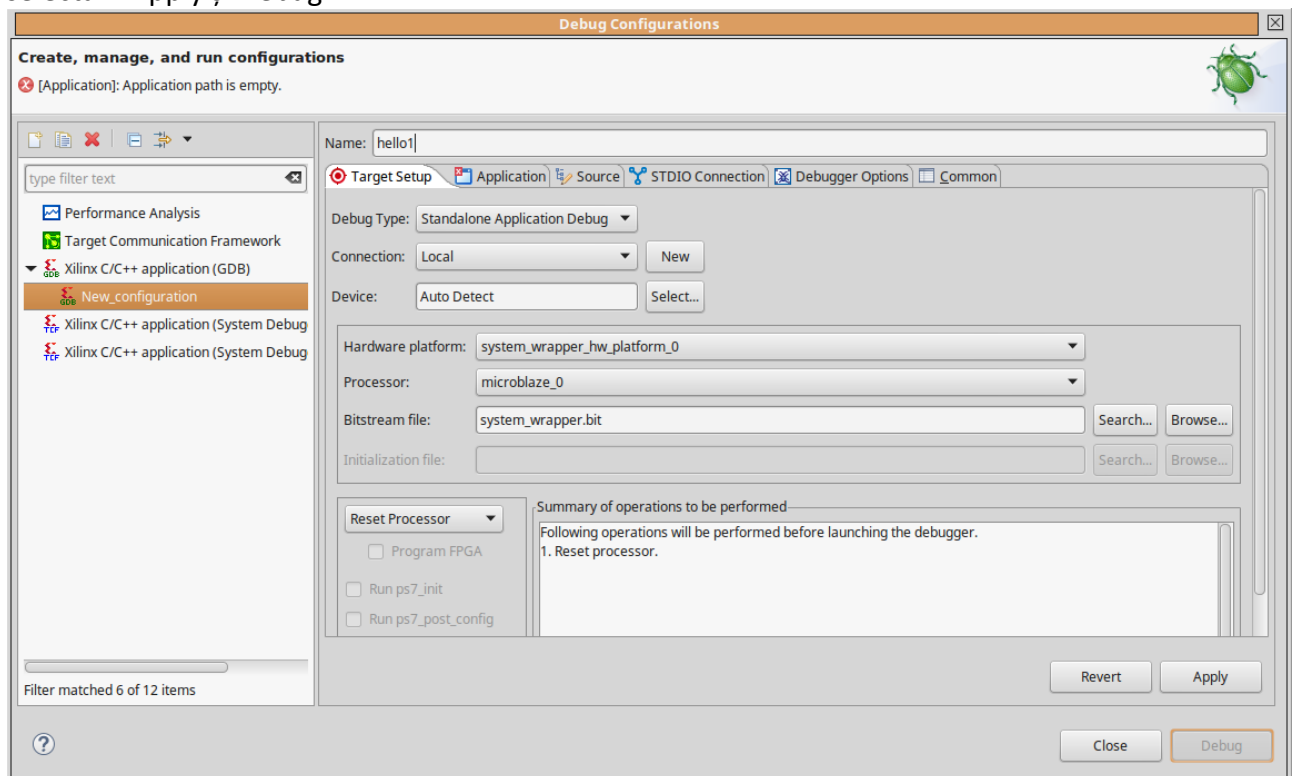
Fișierul helloworld.c va arăta astfel:

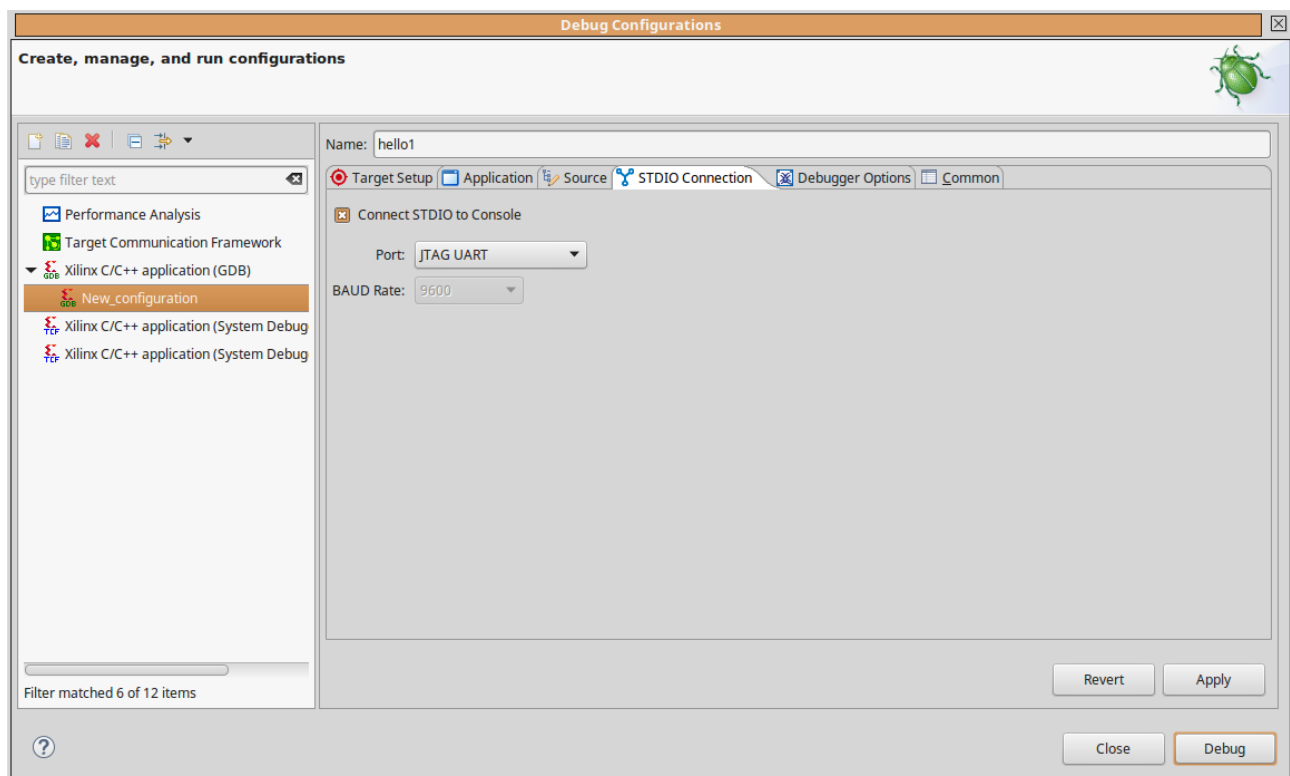
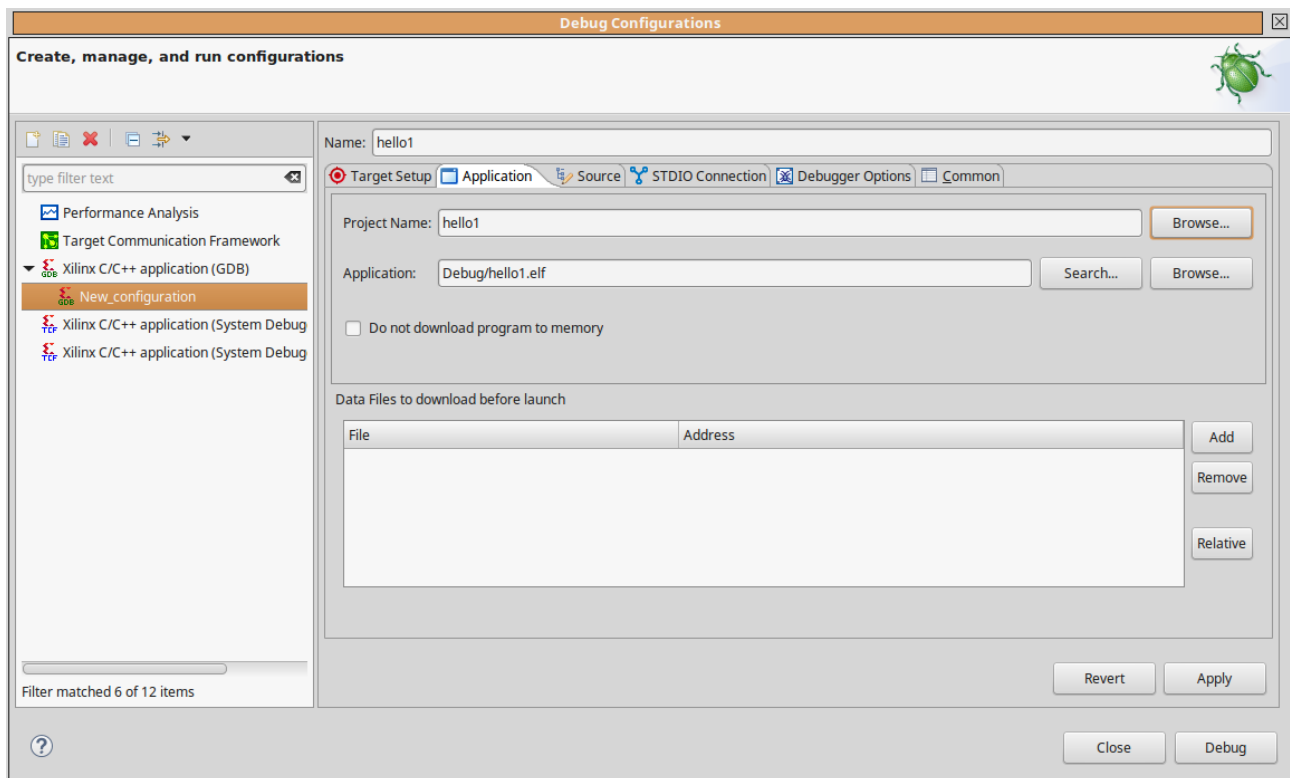


**Pasul 18** – Cuplăm plăcuța la calculator, o pornim și punem primul switch pe 1.

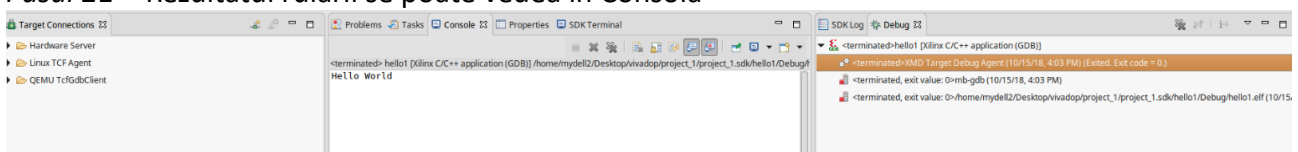
**Pasul 19** – Din meniul Xilinx selectăm opțiunea Program FPGA

**Pasul 20** – Din meniul Run alegem opțiunea Run Configurations și efectuăm următoarele setări. Apoi selectăm Apply și Debug.





**Pasul 21 – Rezultatul r  rii se poate vedea   n Consol  **



**b). Modifica  i programul „Hello World” astfel   nc  t   n consol   s   apar   numerele din   irul lui Fibonacci**

#### **IV. Conținut referat și punctaj**

Referatul va trebui să conțină:

- 1). Implementarea funcției Fibonacci în varianta tradițională cu evidențierea numărului de LUT-uri necesare implementării
- 2). Implementarea funcției Fibonacci folosind Microblaze.

Cele 2 implementări vor fi testate folosind placa de dezvoltare Nexys4 DDR aflată în dotarea laboratorului. Fiecare implementare valorează 0.4 puncte, iar termenul de realizare este de 3 săptămâni începând cu săptămâna 22 – 26.10.2018