

# Circuite Logice Programabile

## LABORATOR 1

### INTRODUCERE ÎN MEDIUL INTEGRAT XILINX ISE. PROIECTAREA UNUI SUMATOR PE UN BIT

#### INTRODUCERE

Softwarele CAD (Computer Aided Design) de proiectare cu circuite logice programabile (FPGA) permit în prima etapă alegerea unei metode de descriere a proiectului: captură schematică, descriere VHDL sau diagrame de stare, în următoarea etapă se poate face o simulare funcțională în urma operației de sinteză și de timp în urma operației de traducere. În cele din urmă, dacă specificațiile de bază sunt întrunite se poate face implementarea și testarea proiectului în hardware, vezi diagramele de mai jos.

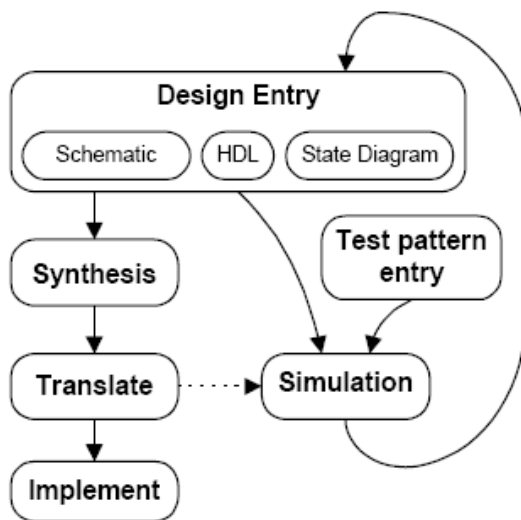


Diagrama I

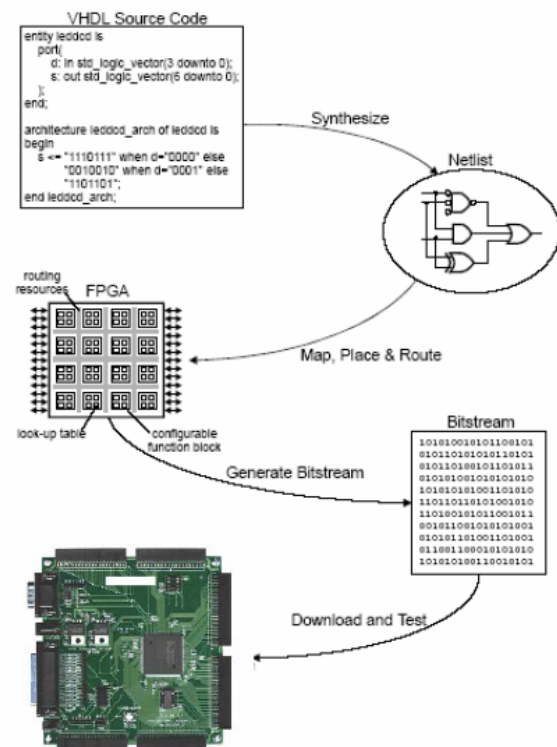


Diagrama II

## SCOPUL LUCRĂRII

ISE – Integrated Software Environment – este un mediu integrat care permite descrierea și implementarea unei multitudini de blocuri logice. Implementarea se face cu FPGA Spartan sau Virtex sau cu CPLD-uri din seria 9500. Circuitele integrate de tip FPGA sau CPLD sunt, de asemenea fabricate de compania Xilinx. La nivel didactic, pentru testarea corectitudinii blocurilor logice proiectate, dispunem de o placa de test D2SB, figura 1, produsă de firma Digilent, bazată pe modulul FPGA de tip Spartan II cu codul XC2S200E.

Pentru a putea exploata mai eficient resursele plăcii D2SB la aceasta se conectează placa cu circuite periferice DIO4, figura 2. Placa DIO4 este produsă de firma Digilent și conține majoritatea dispozitivelor de intrare/ieșire date prezente în orice sistem digital. Placa de dezvoltare este dotată cu comutatoare, taste, LED-uri și afișoare 7 segmente cu LED-uri.



Figura 1. Modulul de dezvoltare D2SB

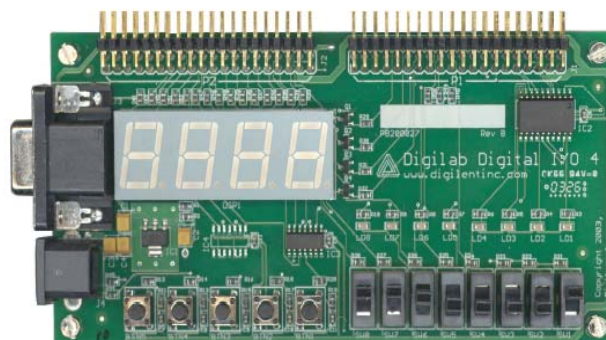


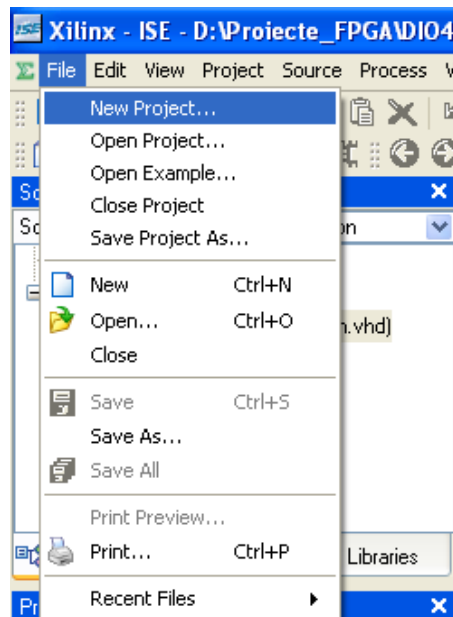
Figura 2. Placa cu dispozitive de intrare/ieșire DIO4

Scopul lucrării constă în implementarea unui sumator elementar descris în VHDL, pentru familiarizarea studenților cu proiectele de tip HDL.

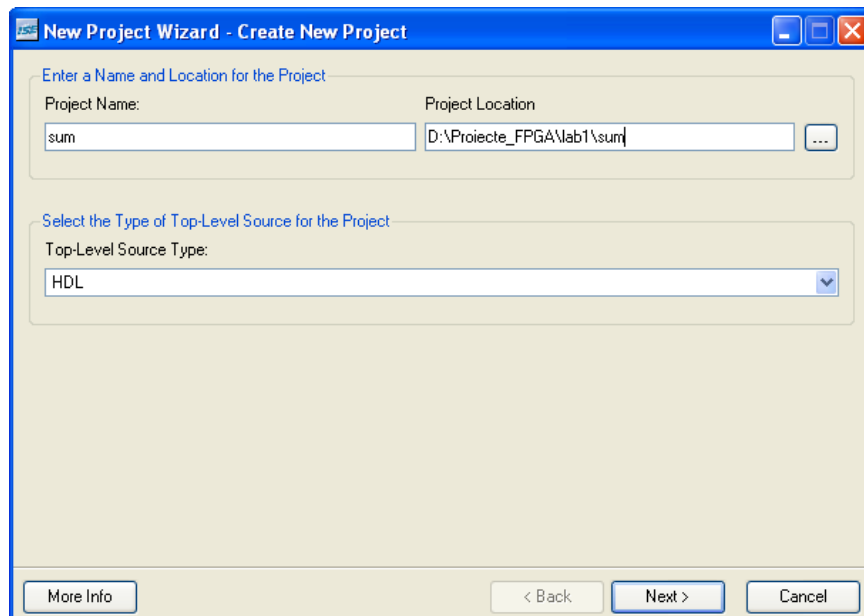
## Desfășurarea lucrării

### Pasul 1: Crearea proiectului.

Se lansează în execuție ISE prin intermediul icoanei

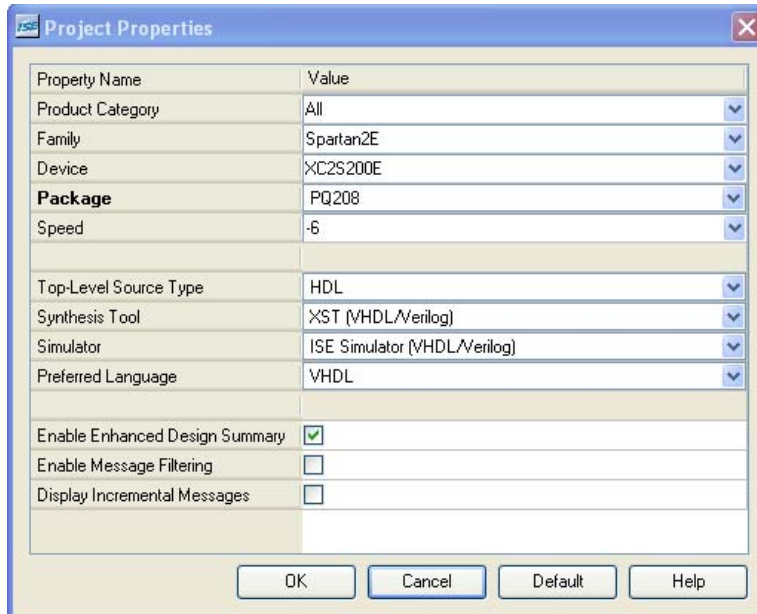


Se creează un nou proiect,  
cu următoarele specificații:



**Figura 3**

ISE creează câte un folder pentru fiecare proiect. Folderul va avea același nume ca și proiectul. Din acest motiv, mai întâi se specifică folderul **lab1**, iar apoi numele proiectului, **sum**. După ce ați particularizat și completat toate informațiile conform figurii 3, apăsați butonul **Next**. Va apare următoarea fereastră:



**Figure 4**

Modulul în jurul căruia este construită placa de dezvoltare D2SB aparține familiei **Spartan2**, are codul **XC2S200E** (200 înseamnă ca modulul conține echivalentul a 200000 de porți logice) și împachetarea este de tip **PQ208**.

Ca și în cazul proiectelor software, un proiect este compus din mai multe fișiere. Funcționalitatea proiectelor hardware se poate specifica fie prin intermediul schemelor logice fie prin intermediul unui limbaj de descriere hardware, cum ar fi VHDL sau Verilog. Așa cum în cazul unui proiect scris în C există o funcție care se execută prima, și anume **main**, în cazul proiectelor hardware rolul lui „main” este jucat de modulul din vârful ierarhiei, și anume **Top Level Module**. Descrierea modulul din vârful ierarhiei poate fi de tip schematic sau HDL (HDL=Hardware Description Language = Limbaj de descriere hardware). Pe parcursul acestui laborator se va folosi numai descrierea de tip HDL. Mai mult despre structura ierarhică a proiectelor hardware, în laboratoarele următoare.

În cazul sumatorului de un bit, descrierea funcționalității se face prin intermediul unui singur fișier HDL și automat aceasta va fi în vârful ierarhiei. Din acest motiv alegem **Top Level Module Type** de tip **HDL**.

Celelalte două câmpuri se setează la valorile din figura 4 for fi detaliate în următoarele laboratoare, iar apoi se apasă butonul **Next**.

Acțiunile aferente următoarelor două ferestre, **Create a New Source** și **Add Existing Sources**, sunt opționale, și din acest motiv se apasă **Next** pentru fiecare în parte, fără să se completeze nimic. În final, apare o fereastră de informare, pentru care se apasă **Finish**. După executarea tuturor acțiunilor descrise mai sus trebuie să se obțină următorul ecran:

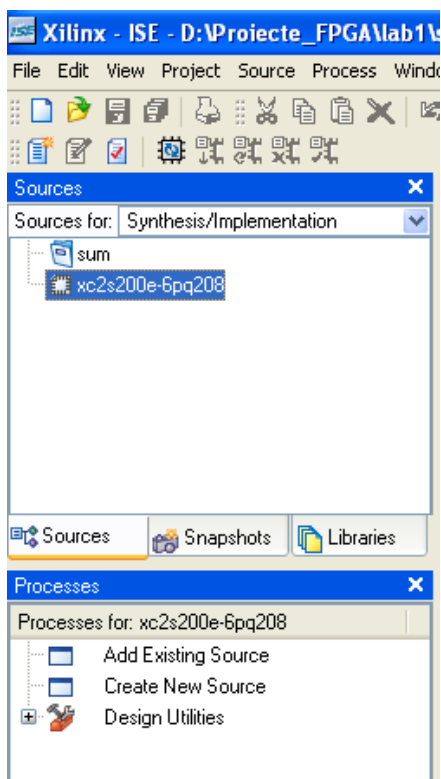


Figura 5

## Pasul 2: Crearea fișierului VHDL care va conține descrierea sumatorului de un bit.

Se face clic dreapta pe numele circuitului (xc2s200e), și din meniul contextual apărut se selectează **New Source**. În fereastră New Source, se selectează tipul **VHDL Module** iar apoi se completează numele fișierului (figura 6). În continuare se apasă **Next**, apare o fereastră care permite definirea semnalelor prin care modulul descris se conectează cu exteriorul (figura 7)

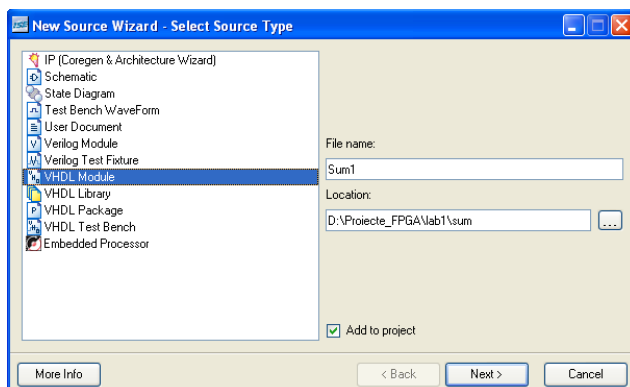


Figura 6

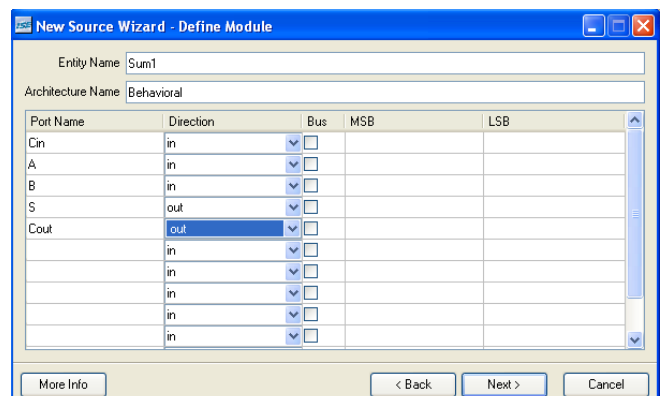


Figura 7

Intrările sumatorului sunt semnalele A, B, și Cin iar ieșirile S și Cout. Se completează fereastra din figura 7 cu aceste informații, apoi se apasă **Next**. Ca urmare, apare o fereastră de informare care sumarizează informațiile introduse și permite revenirea în caz că există o greșeală. Dacă totul este corect, se apasă butonul **Finish**.

Ca efect al apăsării lui **Finish**, **sum.vhd** este construit și adăugat la proiect. După cum se remarcă ISE a creat automat entitatea din descrierea furnizată prin intermediul figurii 7.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Sum_s4 is
  Port ( Cin : in std_logic;
        A : in std_logic;
        B : in std_logic;
        S : out std_logic;
        Cout : out std_logic);
end Sum_s4;
```

.....  
În corpul arhitecturii între *begin* și *end Behavioral* scrieți expresiile logice S și Cout.

```
S <= A xor B xor Cin;
Cout <= (A and B) or (B and Cin) or (A and Cin);
```

### Pasul 3: Crearea fișierului de constrângeri și specificarea acestora.

Pentru a verifica funcționarea sumatorului intrărilor A, B și Cin li se asignează pini FPGA care în exterior sunt conectați la comutatoare iar ieșirilor S și Cout li se asignează pini FPGA conectate la LEDuri. Această asignare se numește constrângere. Există mai multe tipuri de constrângeri, aceasta fiind una dintre ele.

Din documentațiile plăcilor D2SB și DIO4 rezultă următorul tabel:

| RESURSE D2SB ȘI DIO 4 | PINI FPGA |
|-----------------------|-----------|
| "MCLK"                | "P182";   |
| "D2SB_BTN"            | "P187";   |
| "D2SB_LED"            | "P154";   |
| "DIO4_LEDG"           | "P45";    |
| "DIO4_LED<0>"         | "P111";   |
| "DIO4_LED<1>"         | "P109";   |
| "DIO4_LED<2>"         | "P102";   |
| "DIO4_LED<3>"         | "P100";   |
| "DIO4_LED<4>"         | "P98";    |
| "DIO4_LED<5>"         | "P96";    |

|               |         |
|---------------|---------|
| "DIO4_LED<6>" | "P94";  |
| "DIO4_LED<7>" | "P89";  |
| "DIO4_BTN<0>" | "P3";   |
| "DIO4_BTN<1>" | "P206"; |
| "DIO4_BTN<2>" | "P44";  |
| "DIO4_BTN<3>" | "P43";  |
| "DIO4_BTN<4>" | "P42";  |
| "DIO4_SW<0>"  | "P23";  |
| "DIO4_SW<1>"  | "P21";  |
| "DIO4_SW<2>"  | "P18";  |
| "DIO4_SW<3>"  | "P16";  |
| "DIO4_SW<4>"  | "P11";  |
| "DIO4_SW<5>"  | "P9";   |
| "DIO4_SW<6>"  | "P7";   |
| "DIO4_SW<7>"  | "P5";   |
| "DIO4_AN<0>"  | "P41";  |
| "DIO4_AN<1>"  | "P40";  |
| "DIO4_AN<2>"  | "P36";  |
| "DIO4_AN<3>"  | "P35";  |
| "DIO4_SSG<0>" | "P22";  |
| "DIO4_SSG<1>" | "P20";  |
| "DIO4_SSG<2>" | "P17";  |
| "DIO4_SSG<3>" | "P15";  |
| "DIO4_SSG<4>" | "P10";  |
| "DIO4_SSG<5>" | "P8";   |
| "DIO4_SSG<6>" | "P6";   |
| "DIO4_SSGDP"  | "P4";   |

Astfel se vor atribui următorii pini FPGA pentru A,B,Cin, S și Cout: Cin – **P23** (SW1), A – **P21** (SW2), B – **P18** (SW3), conform tabelului anterior. Lui S i se asignează **P111** (LD1), iar lui Cout **P109** (LD2).

Mai întâi se creează fișierul de constrângeri. Se face clic dreapta pe numele circuitului (xc2s200e) sau pe numele fișierului VHDL sum1.vhd și din meniul contextual apărut se selectează **New Source**. În fereastră New Source, se selectează tipul **Implementation Constraints File** iar apoi se completează numele fișierului. Fie numele acestui fișier **sum1cf**. Se apasă **Next**, în următoarea fereastră **Next**, iar apoi în fereastra de informare se apasă **Finish**. Dacă totul a decurs conform celor explicate anterior se va obține situația din figura 8.

Se observă ca procesele posibile pentru un anumit fișier sunt diferite în funcție de fișierul selectat în fereastra **Sources in Project**. Dacă în fereastra **Sources in Project** se selectează **sum1cf.ucf**, în fereastra **Processes for ...** apar procesele din figura 8.

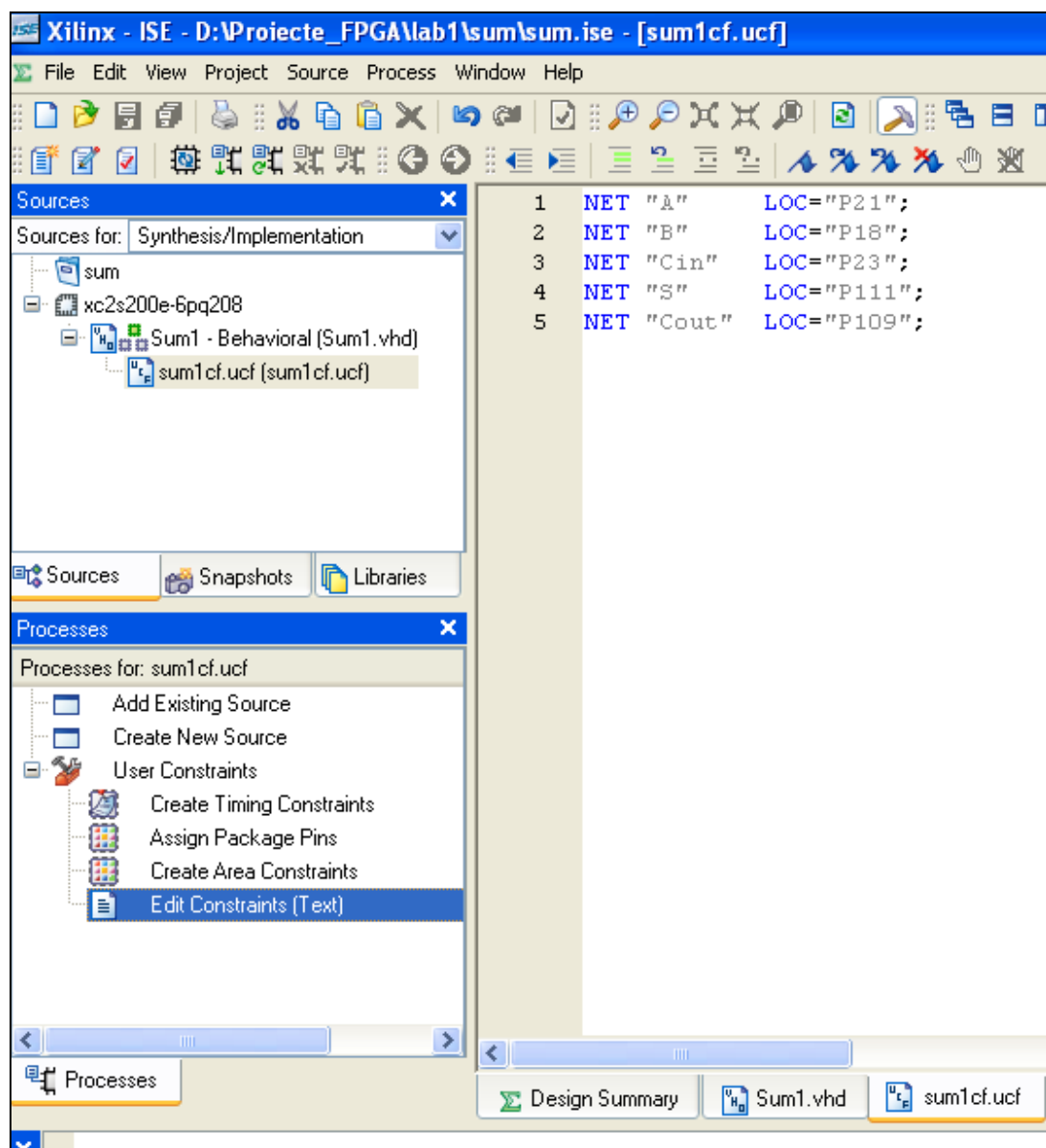
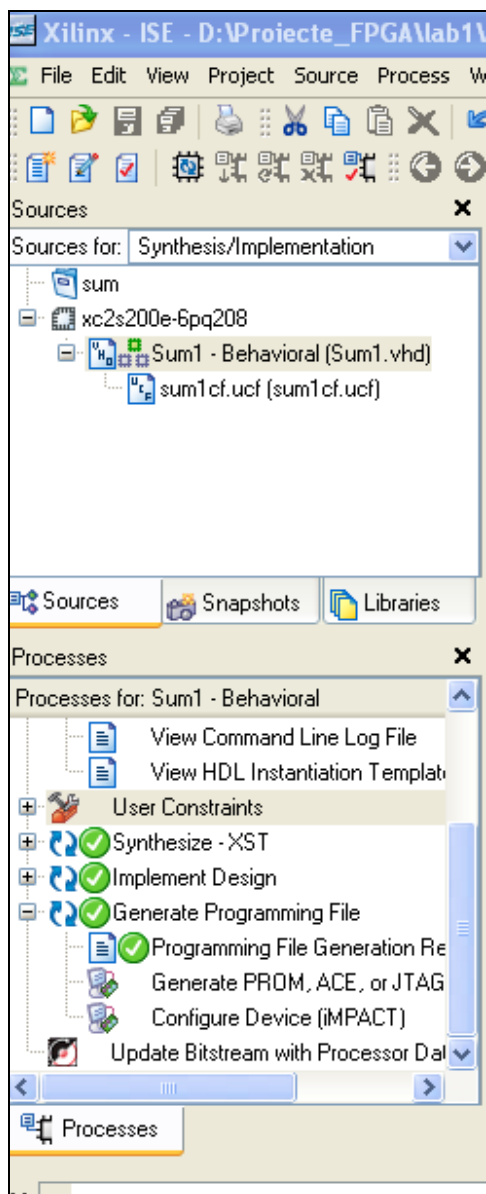


Figura 8

Înainte de a crea fișierul de constrângeri selectați **Edit -> Preferences -> Editor** și apăsați radio butonul **Constraints Editor**. Pentru crearea/editarea constrângerilor, pornind de la configurația din figura 8, se face dublu clic pe procesul **Edit Constraints (Text)**, pentru a lansa în execuție editorul de constrângeri și se introduc liniile de configurare din figura 8, după care se salvează.

#### Pasul 4: Crearea fișierului de configurare și verificarea funcționalității.

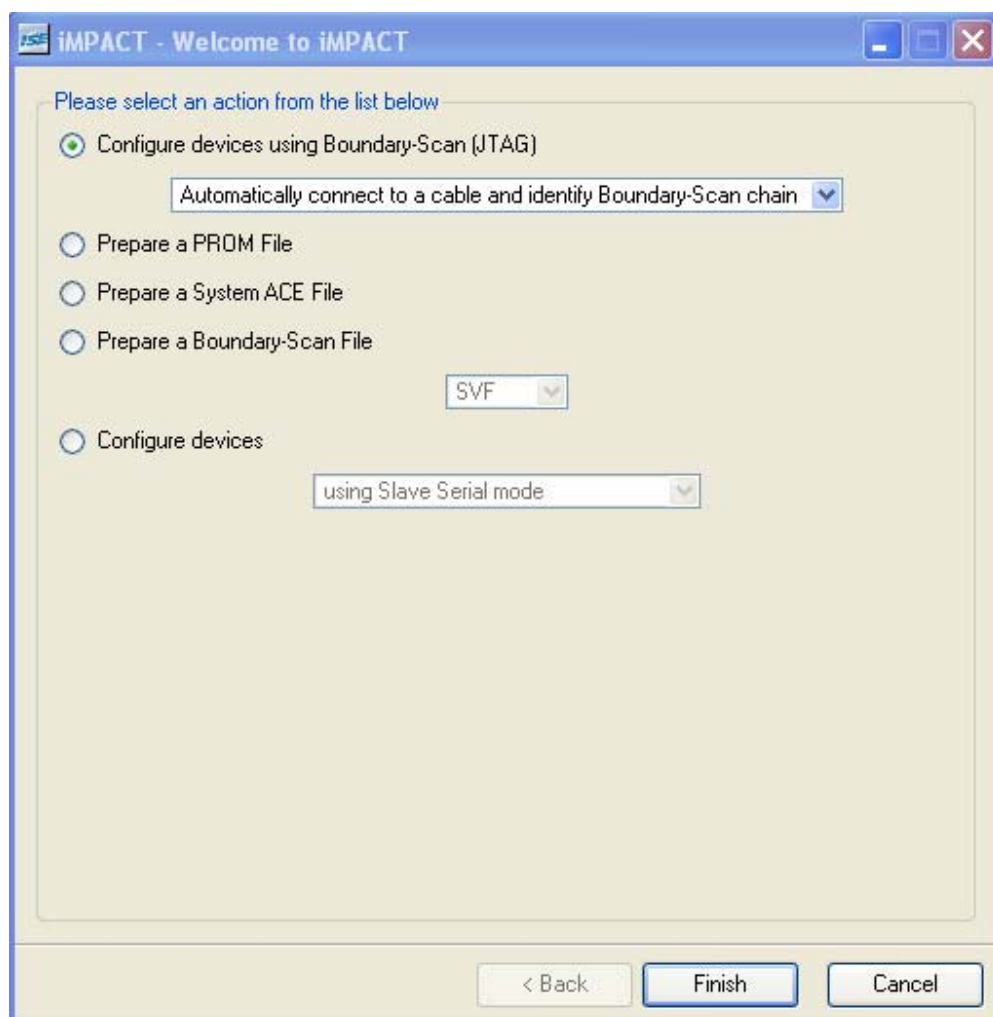
Plecând de la configurația din figura 8, se face click pe numele proiectului *Sum1- Behavioral..* după care în fereastra *Processes* se face dublu clic pe procesul **Generate Programming File**, vezi figura 9.



**Figura 9**

#### Pasul 5: Configurarea FPGA Spartan 3 de pe placa de dezvoltare

Odată generat fișierul bit sub procesul **Generate Programming File** faceți dublu clic pe opțiunea **Configure Device (iMPACT)** aflată sub procesul **Generate Programming File**. Se va obține fereastra din figura 10, se face clic pe Finish.



**Figura 10**

După care se obține figura fereastra din figura 11, după ce se face în care se selectează fișiereul de configurare de tip bit, în cazul nostru *sum1.bit*

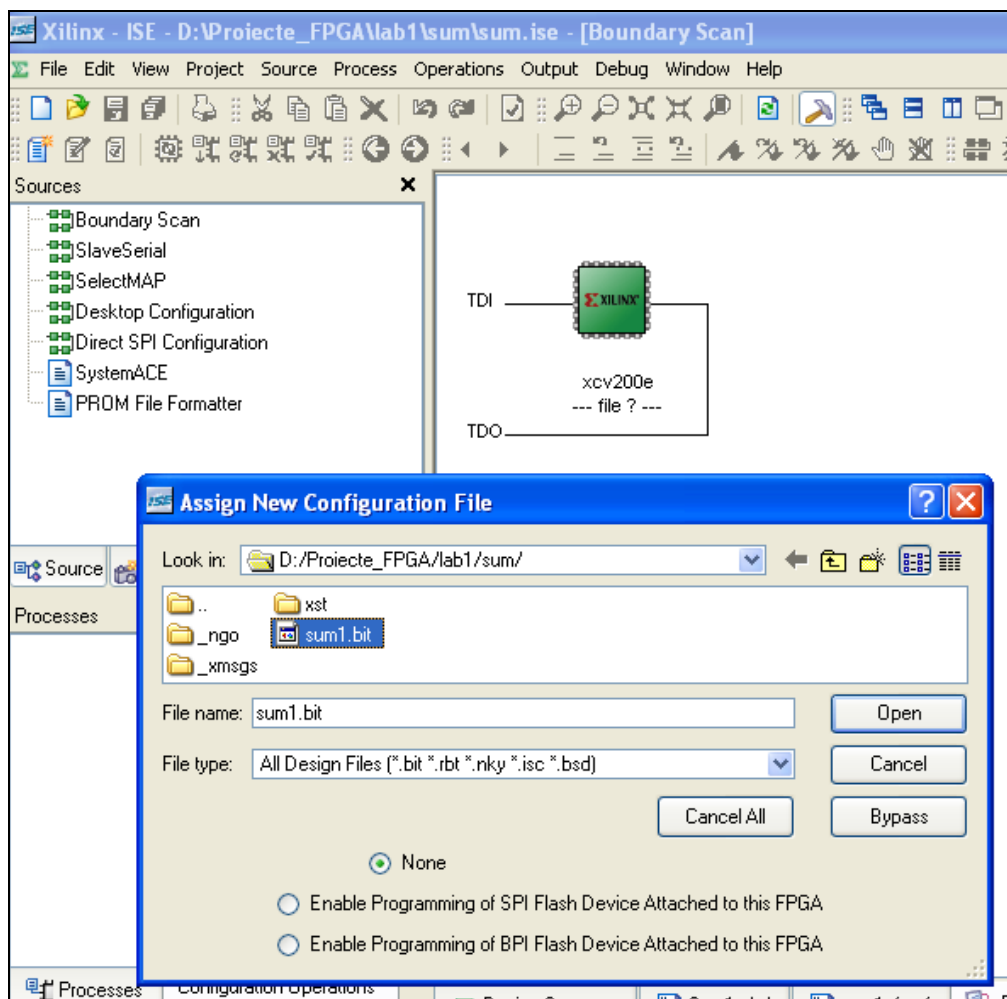
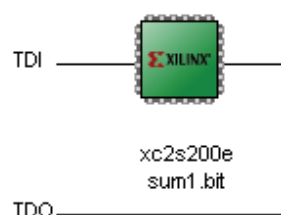


Figura 11



Se face clic dreapta pe dispozitivul Xilinx și se alege opțiunea *Program...* în fereastra care apare se face clic pe OK. Apare o fereastră *Progress Dialog* după care

**Program Succeeded**

ar trebui să se obțină mesajul profesorul!!!

în caz contrar, chemați

#### Pasul 6: (opțional, pentru notă mai mare ca 5)

1. Pentru ca datele să fie transferate imediat de la intrare la ieșire este necesar ca latchurile prezente la fiecare LED să primească semnal de validare, astfel conform documentației plăcii D2SB se recomandă ca semnalul Leds să fie ținut în 1 logic. Pinul FPGA aferent acestui semnal este P45.

Introduceți linia corespunzătoare în dul VHDL și în fișierul de constrângeri UCF, astfel încât afișarea rezultatului la LED-uri să aibă loc imediat ce au fost modificate valorile de la intrarea sumatorului.

2. Modificați proiectul existent astfel încât să se implementeze un sumator pe doi biți.

Intrările vor fi:

- Cin,
- A1, A0 primul operand, A0 LSB
- B1, B0 al doilea operand, B0 fiind LSB

Ieșirile vor fi:

- S1, S0 suma, S0 LSB
- Cout

Intrările și ieșirile se vor conecta la comutatoare și Leduri astfel:

Cin – SW0, A0-SW1, A1-SW2, B0-SW3, B1-SW4, S0-LD0, S1-LD1, Cout-LD2