

Sisteme cu F. P. G. A. și D. S. P.

- Implementarea unui nucleu de micro-controller la nivel de FPGA –
- Cu ajutorul tehnologiei Xilinx MicroBlaze – [1]

I. INTRODUCERE:

Platforma de dezvoltare Basys 3 este realizată pe baza nucleului FPGA Artix – 7 și încorporează o serie de echipamente periferice precum:

- 16 comutatoare bipoziționale;
- 4 butoane cu apăsare și revenire;
- 4 afișaje pe 7 segmente;
- 16 indicatori luminoși pe bază de diode electroluminiscente;
- 4 grupuri de terminale externe, fiecare a câte 12 terminale;
- convertor analog – numeric (digital) jXADC;
- interfață USB – UART + jTAG pentru conectare la calculatorul gazdă;
- interfață video VGA;
- interfață USB – „A” pentru memorii externe, tastatură sau alte periferice de calculator;



Fig. 1 - Platforma de dezvoltare Digilent Basys 3 – FPGA Xilinx Artix – 7 [2]

Aceste echipamente sau interfețe periferice, pot fi gestionate prin intermediul configurației fizice stabilite la nivel de nucleu FPGA. Configurația nucleului FPGA, se poate stabili pe baza fișierului „bit” (eng. BitStream), care se generează în urma compilării sau sintetizării modelului de program VHDL / Verilog. Astfel, pentru a „re-stabili” modul de

funcționare a unei aplicații implementate fizic la nivel de FPGA (prin intermediul fișierului BitStream), este deci necesară re-configurarea structurii interne. Pentru a evita procesul amplu de re-configurare și re-sintetizare a modelului fizic (eng. hardware) implementat la nivel de FPGA, se apelează la nucleul virtual de tip „soft-core-processor” denumit MicroBlaze.

MicroBlaze, este un concept propus de către compania Xilinx, care presupune, implementarea la nivel de FPGA a unei structuri de procesor sau micro-controller. Astfel, realizarea aplicațiilor cu ajutorul perifericelor de uz general (intrări și ieșiri digitale, indicatori luminoși, butoane, porturi și interfețe) nu mai necesită re-configurarea structurii interne a nucleului FPGA. Implementarea nucleului MicroBlaze se va realiza pe baza instrucțiunilor HDL (eng. Hardware Description Language – limbaj de programare pentru implementare fizică). Din acest motiv, nucleul MicroBlaze este considerat un nucleu de tip logic (eng. soft-core);

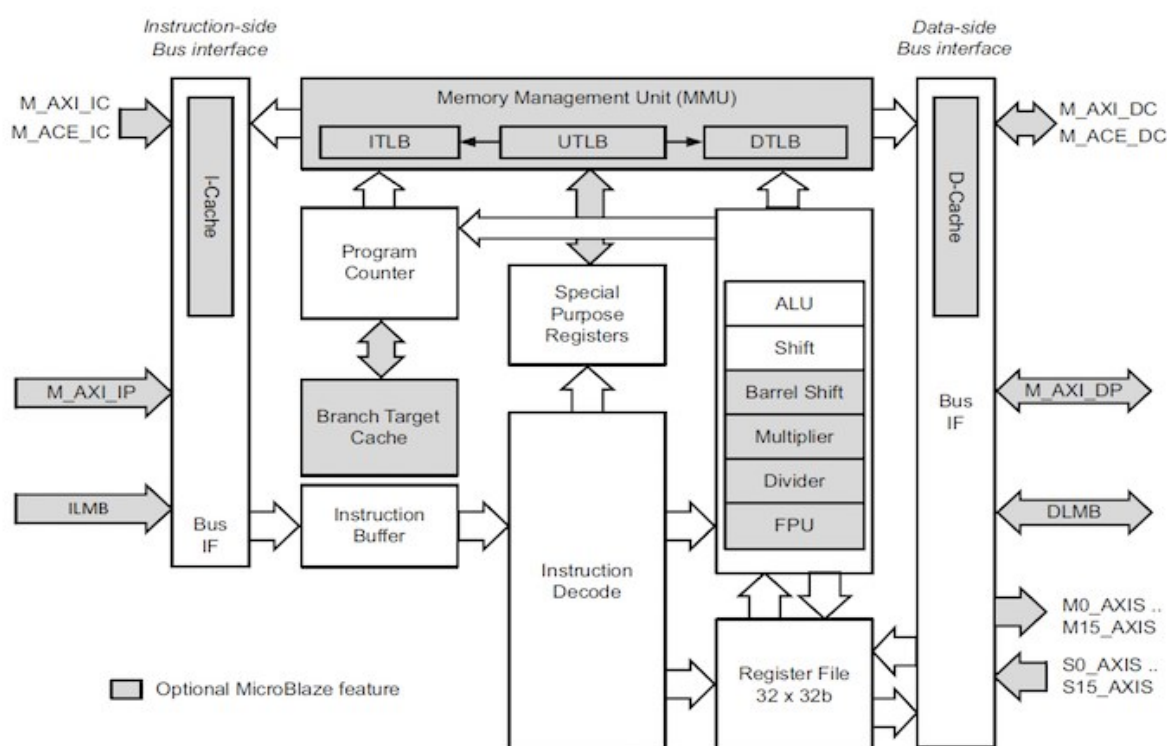


Fig. 2 – Arhitectura nucleului de procesor MicroBlaze [3]

Procesorul virtual MicroBlaze imită comportamentul unui sistem de calcul cu set redus de instrucțiuni (eng. Reduced Instruction – Set Computer – RISC), având capacitatea de reprezentare a instrucțiunilor pe 32 de biți. Scopul pentru care a fost conceput acesta, este de a permite dezvoltarea mai multor aplicații cu micro - controller sau microprocesor în paralel la nivel de FPGA, și pentru a permite accelerarea operațiilor din faza de dezvoltare a aplicației.

Inter-conectarea blocurilor logice și a perifericelor din cadrul nucleului fizic FPGA și cel virtual MicroBlaze, se realizează cu ajutorul interfeței AXI [4] (eng. Advanced eXtensible Interface). Practic, interfața AXI, asigură transferul de informație dintre procesorul virtual și perifericele atașate la acesta. Interfața respectivă prezintă:

- sincronism la transferul de date;
- transfer paralel de informație + suport la conexiuni de tip multi-master și multi - slave;

II. IMPLEMENTARE:

Pe platforma de dezvoltare cu FPGA – Digilent Basys 3, având un nucleul Artix – 7, se va implementa, un exemplu simplu pentru gestionare a intrărilor și ieșirilor digitale de uz general (eng. GPIO – General Purpose Input / Output), cu ajutorul nucleului MicroBlaze.

Pachetele de programe necesare pentru realizarea aplicației sunt:

- Vivado Design Suite 2016.2 (System Edition + Software Development);
- Vivado SDK 2016.2 (Software Development Kit);
- Basys 3 – Digilent Board Support Package files;

A. Implementarea procesorului virtual MicroBlaze cu ajutorul lui Vivado Design Suite:

Pentru început în cadrul programului Vivado 2016.2, se va crea un proiect nou, prin apăsarea butonului „Create New Project” din lista de comenzi rapide „Quick Start”.

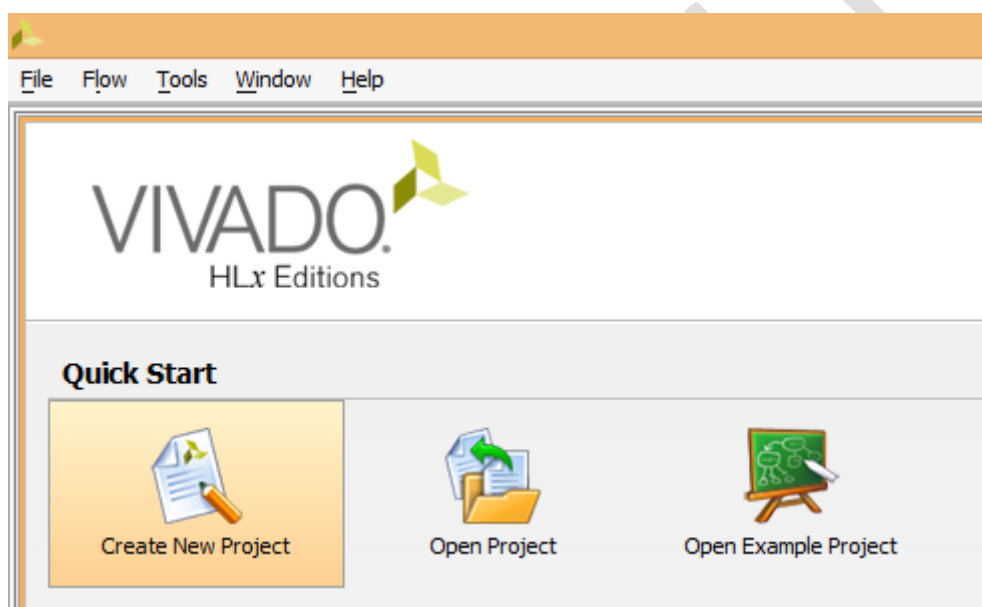


Fig. 3 – Crearea unui proiect nou în cadrul programului Vivado 2016.2

În urma alegerii opțiuni pentru crearea a unui proiect nou, se va deschide o fereastră pentru îndrumare în vederea stabilirii aspectelor de bază ale proiectului nou.

Pentru a parcurge prima etapă (de informare) a procesului de îndrumare, se va alege opțiunea „Next” (următoarea etapă).



Fig. 4 – Crearea unui proiect nou în cadrul programului Vivado 2016.2 (prima etapă)

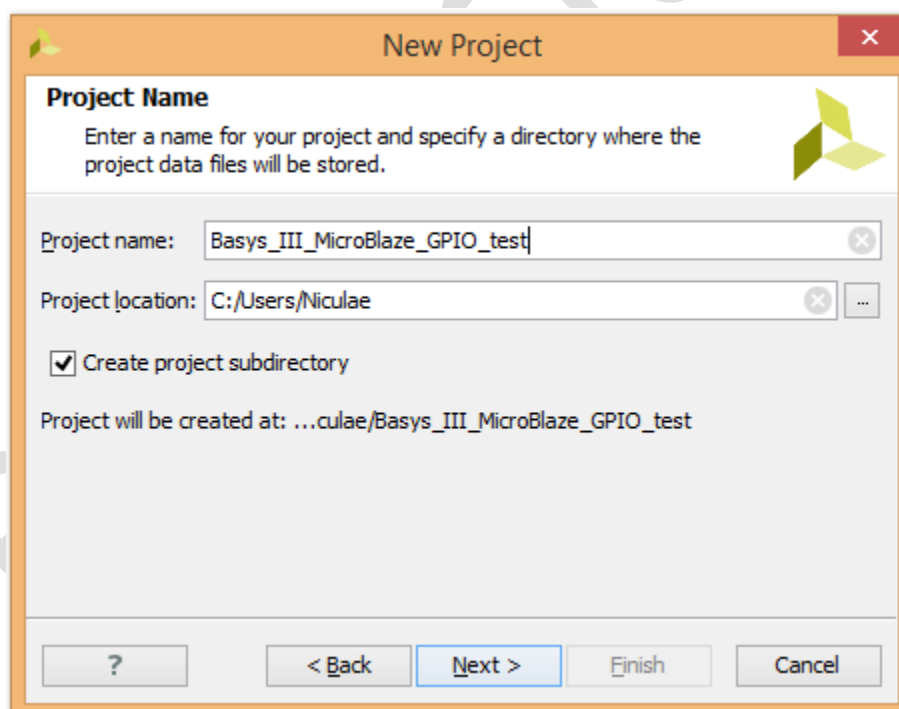


Fig. 5 – Stabilirea denumirii și a locației pentru proiectul nou creat (etapa a doua)

Se va alege denumirea proiectului „Basys_III_MicroBlaze_GPIO_test”, iar locația pentru stocare, se va păstra în mod implicit. De asemenea, se va alege opțiunea pentru crearea unui subdirector pentru proiectul nou conceput.

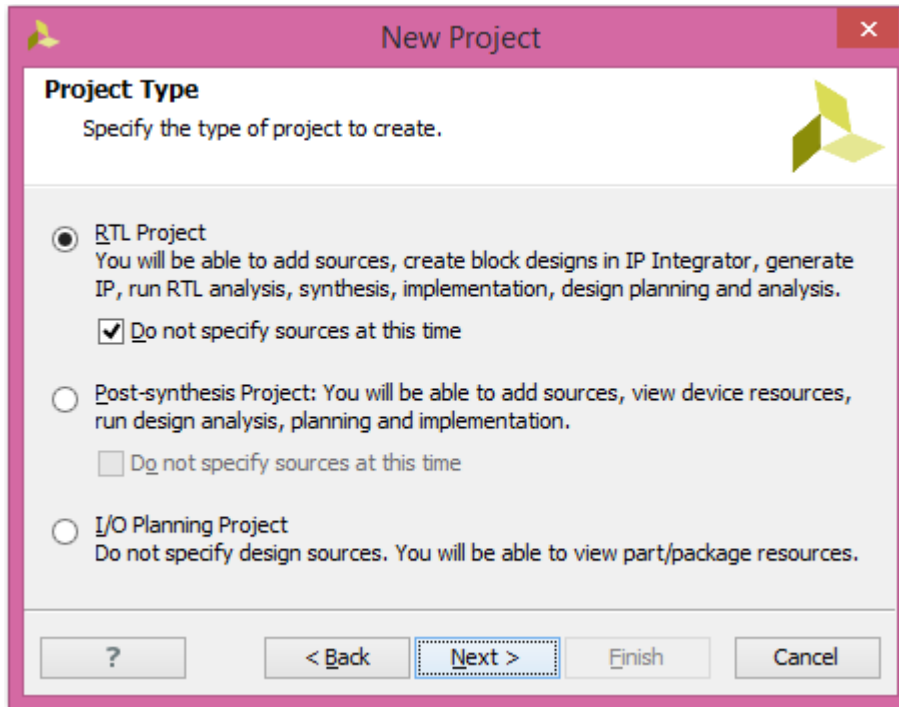


Fig. 6 – Stabilirea tipului de proiect (etapa a treia)

Se va alege „RTL Project” și opțiunea de a amâna specificarea fișierelor sursă.

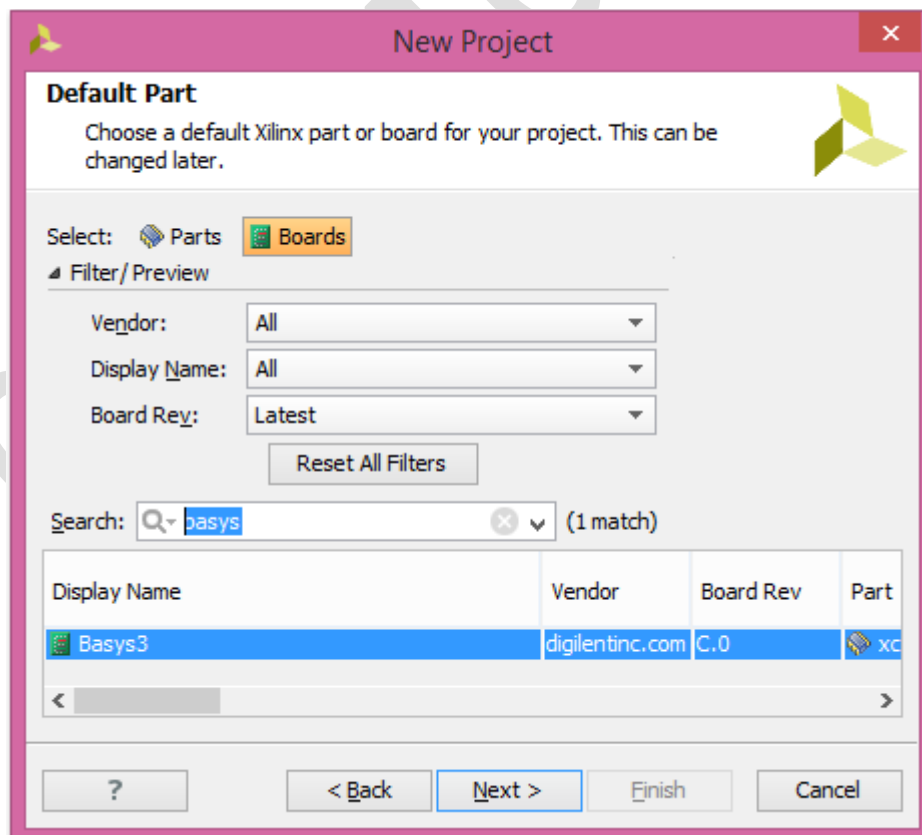


Fig. 7 – Specificarea modelului platformei de dezvoltare (etapa a patra)

În cadrul căsuței pentru căutare, se va introduce numele platformei de dezvoltare „basys”, apoi, din listă, se va alege opțiunea „Basys 3”.

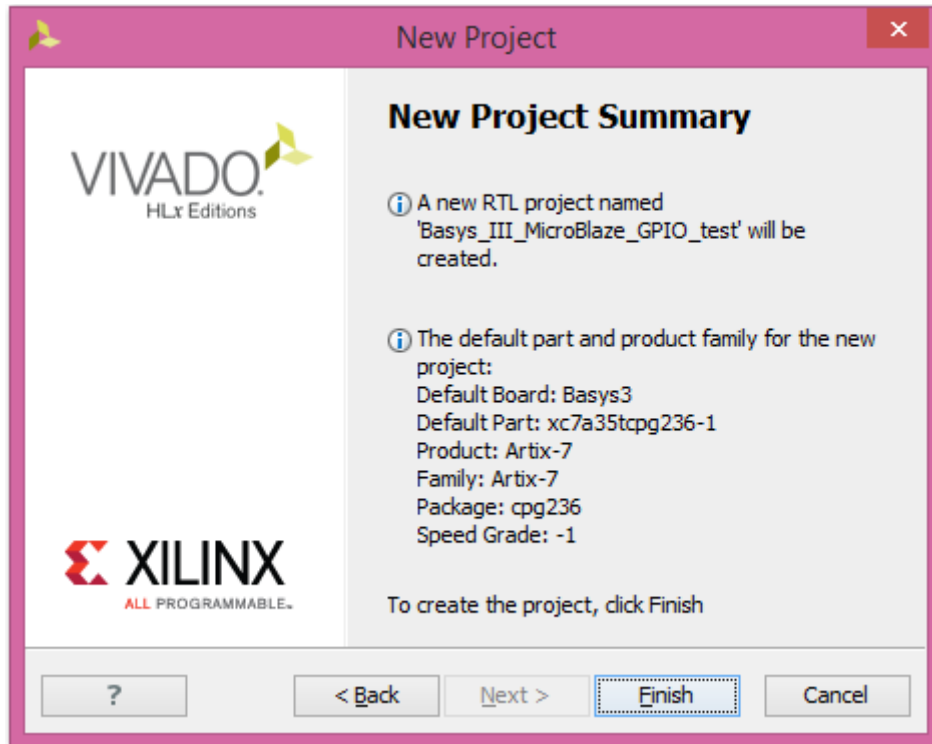


Fig. 8 – Finalizarea procesului de creare proiectului (etapa a cincea)

Se va finaliza procesul de creare a proiectului nou, prin apăsarea butonului „Finish” (finalizare). În urma acestui proces, se va deschide spațiul de lucru în mediul Vivado 2016.2.

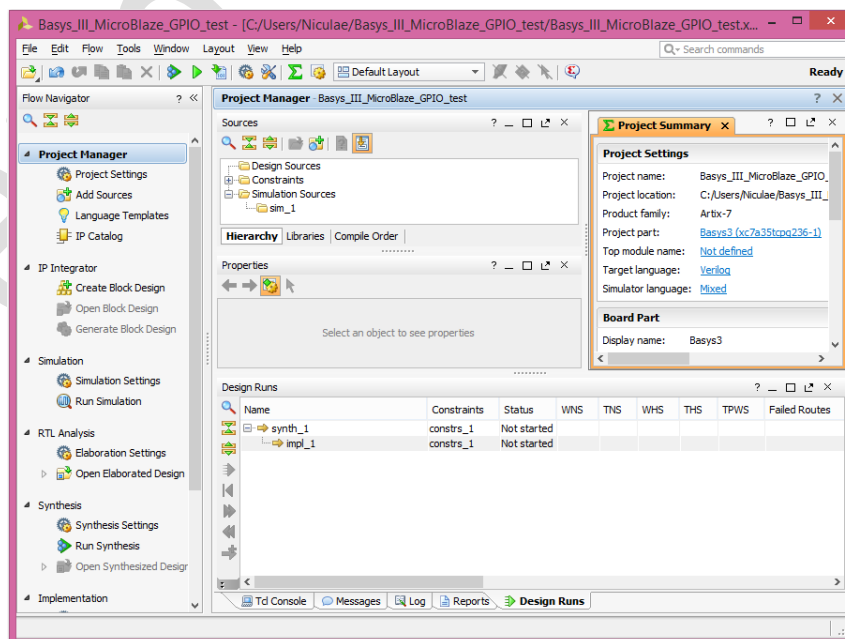


Fig. 9 – Spațiul de lucru al mediului Vivado 2016.2

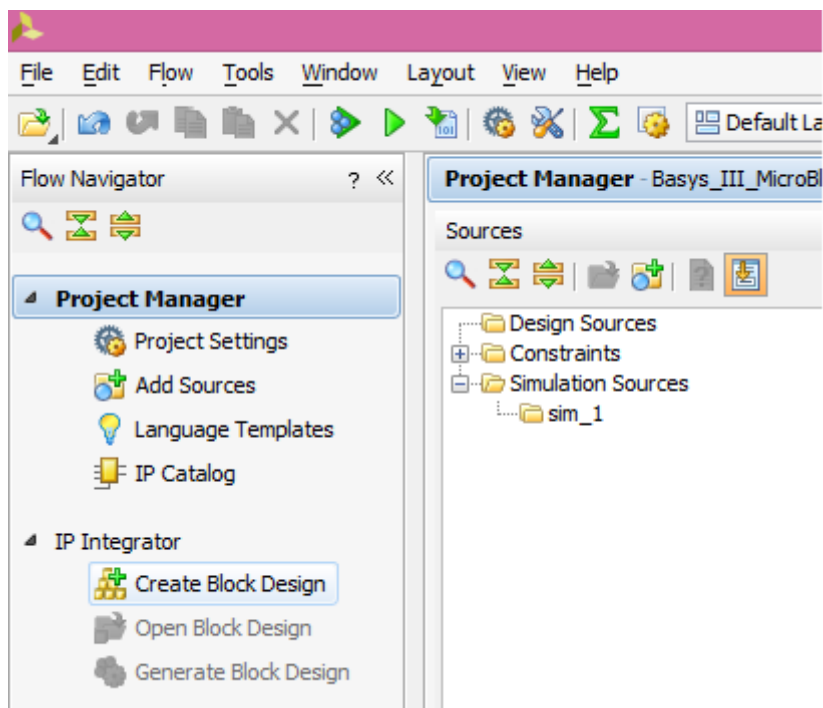


Fig. 10 – Crearea unui model pe baza reprezentării cu blocuri

În spațiul de lucru al mediului Vivado 2016.2, din bara de categorii (Flow Navigator) se va alege opțiunea „Create Block Design” din categoria „IP Integrator”.

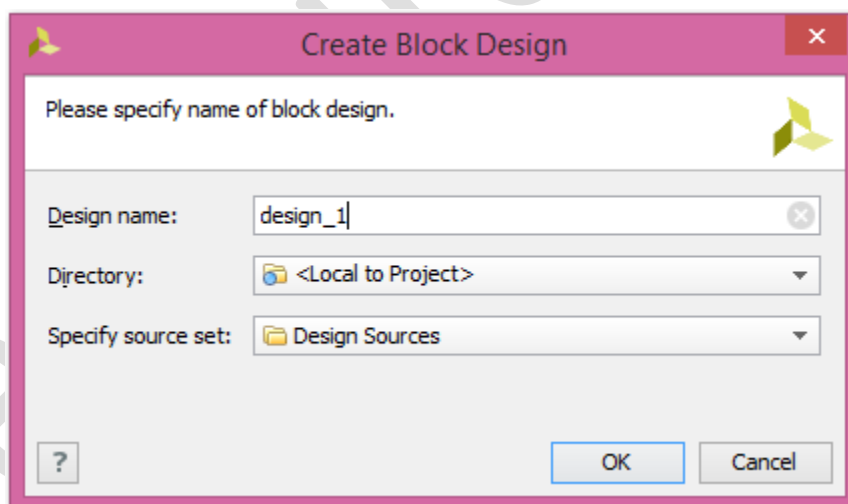


Fig. 11 – Particularizarea denumirii, locației și grupului de apartenență a modelului

Se vor păstra valorile implicite pentru denumire, locație, și grupul de apartenență al modelului conceput pe baza blocurilor. Acest model, practic, permite utilizatorului să particularizeze arhitectura internă a nucleului FPGA, cu ajutorul blocurilor grafice. Tot în cadrul acestei etape, se va stabili arhitectura procesorului virtual și a dotărilor acestuia. De asemenea, dacă se va alege o structură de micro-controller se vor specifica și configurațiile perifericelor atașate la microprocesorul virtual MicroBlaze.

Pentru a adăuga perifericele specifice platformei de dezvoltare în spațiul de lucru, din categoria „Boards”, se va selecta elementul „4 Push Buttons” cu operația click dreapta iar apoi prin accesarea opțiunii „Connect Board Component...” din cadrul meniului contextual.

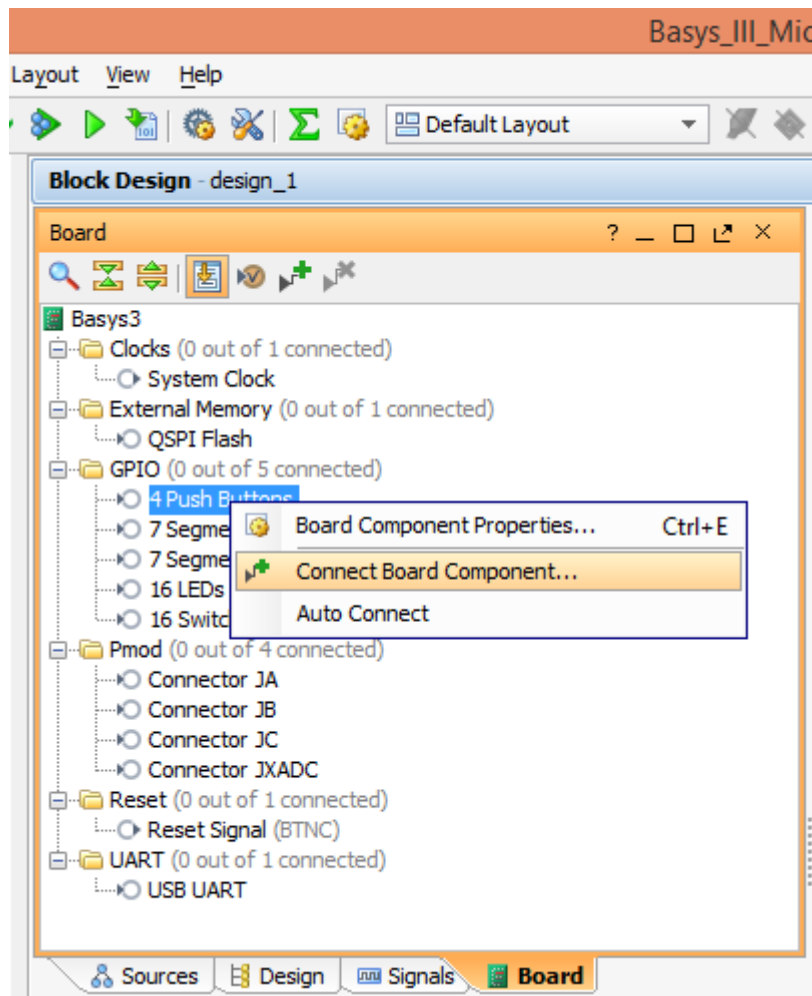


Fig. 12 – Adăugarea componentelor periferice

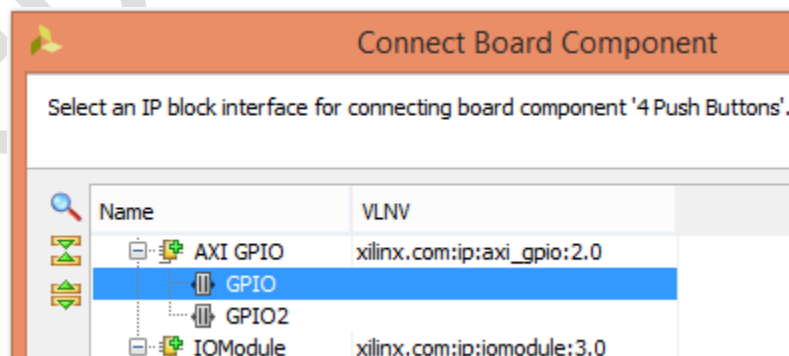
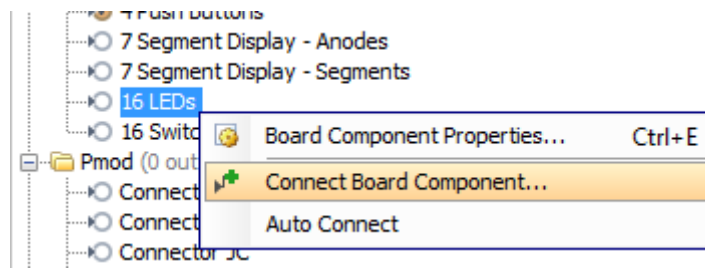


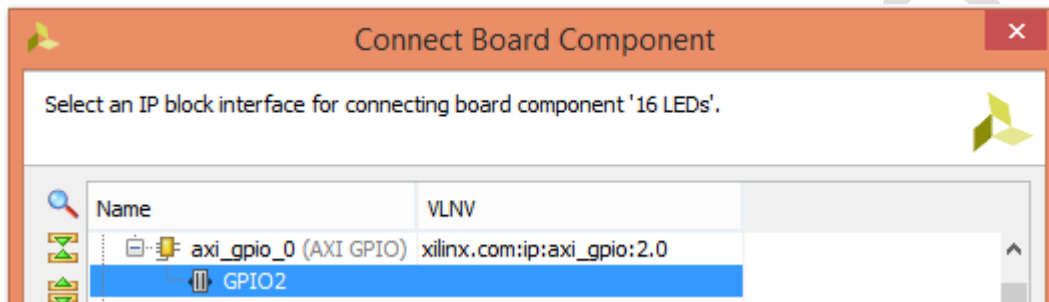
Fig. 13 – Localizarea perifericelor în structura FPGA

În cadrul următoarei etape se va alege opțiunea „GPIO” pentru alocarea perifericelor.

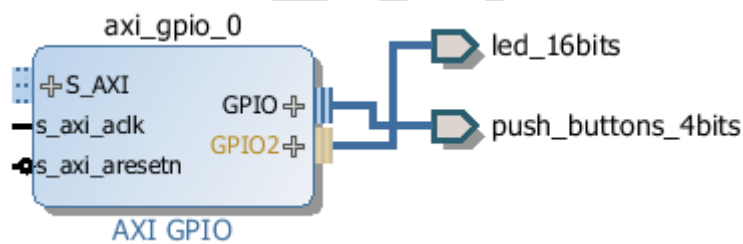
Se va proceda în mod similar pentru elementele „16 LEDs”:



A.



B.



C.

Fig. 14 – Generarea blocurilor pentru conectare a perifericelor

În continuare, se va introduce oscilatorul cu cuarț sau generatorul de tact de ceas:

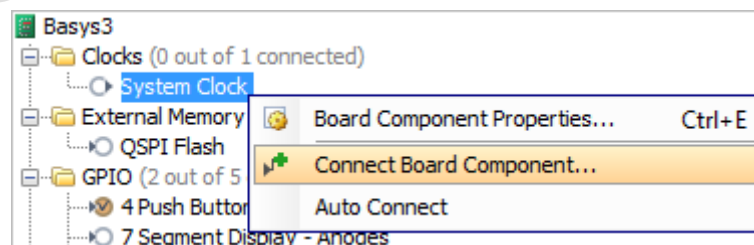


Fig. 15 – Adăugarea oscilatorului pe bază de cuarț

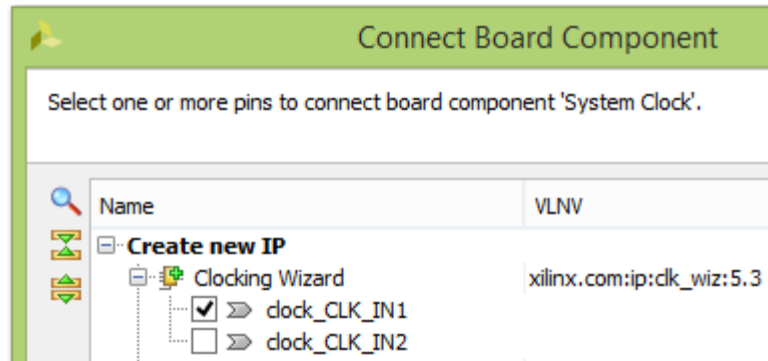


Fig. 16 – Localizarea resurselor în cadrul platformei de dezvoltare (oscilatorul)

Se va alege opțiunea predefinită „clock_CLK_IN1”. În urma acestor operații se va obține blocul „clk_wiz_0”. Pentru a parametriza acest bloc se va efectua operația dublu clic.

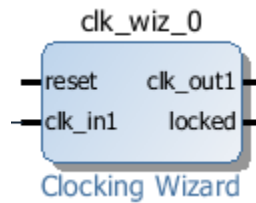


Fig. 17 – Blocul pentru configurarea oscilatorului

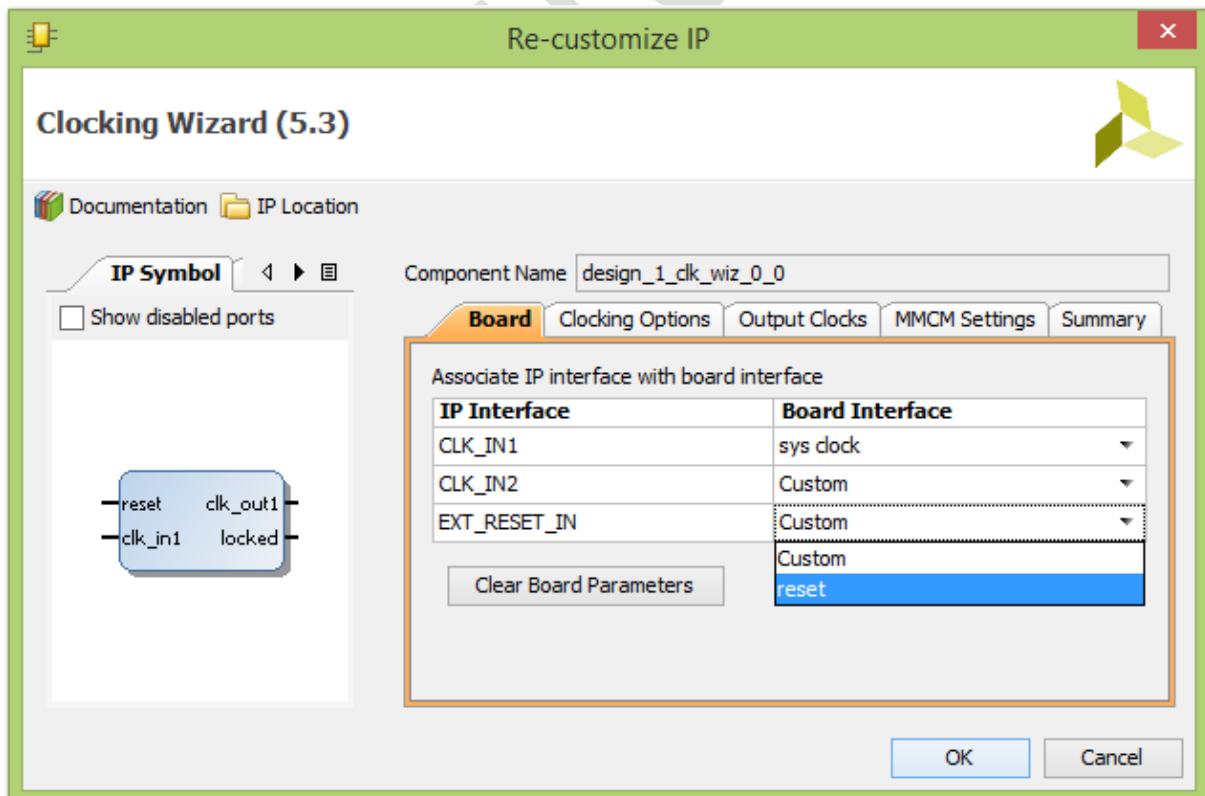
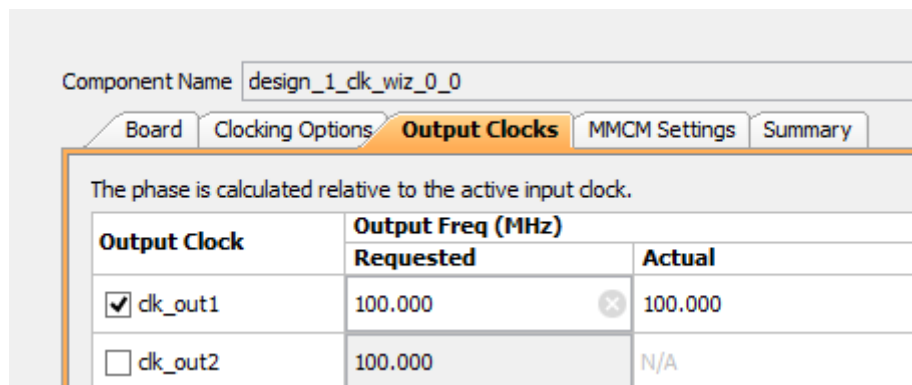


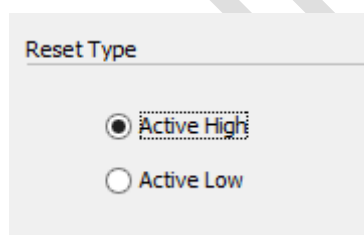
Fig. 18 – Particularizarea oscilatorului – categoria „Board”

În cadrul categoriei „Board” din fereastra pentru îndrumare a configurării oscilatorului, se vor alege următorii parametri: „CLK_IN1” = „sys clock” și „EXT_RESET_IN” = „reset”.

În cadrul categoriei „Output Clocks” se va alege frecvența 100.000 (MHz) pentru „clk_out1” în coloana „Output Freq (MHz)” și sub-categoria „Requested”. Se va verifica de asemenea, dacă polaritatea elementului de resetare este activ pozitivă (eng. Active High).



A.



B.

Fig. 19 – Particularizarea oscilatorului – categoria „Output Clocks”

Tot din cadrul listei de componente periferice, se va adăuga și conexiunea USB – UART.

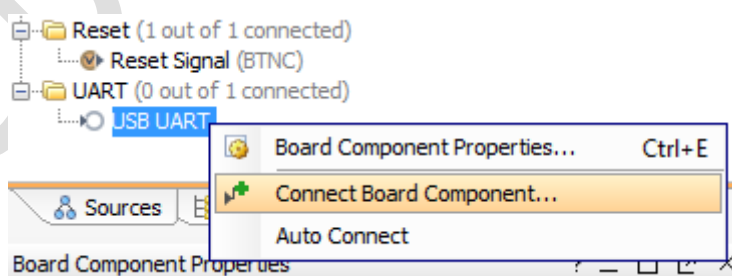


Fig. 20 – Adăugarea conexiunii USB – UART

În următoarea etapă, se va alege opțiunea „AXI Uartlite” – „UART”.

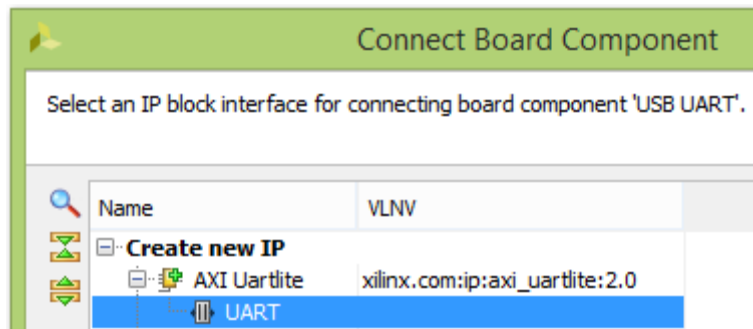


Fig. 21 – Opțiunea „AXI Uartlite” – „UART”

În vederea realizării legăturilor între componente se va alege opțiunea „Run Connection Automation” din bara indicatoare (verde) – „Designer Assistance available”, iar în cadrul ferestrei următoare se va confirma alegerea prin apăsarea butonului „OK”:

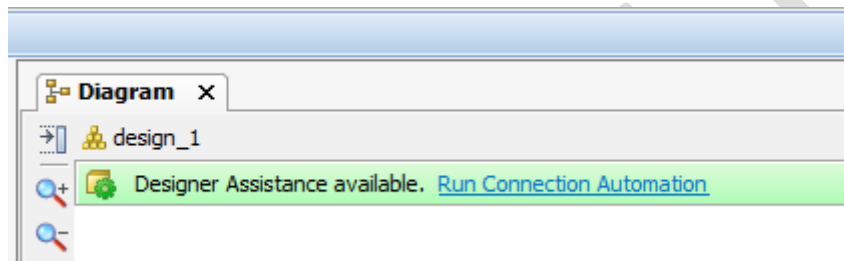


Fig. 22 – Opțiunea „Run Connection Automation”

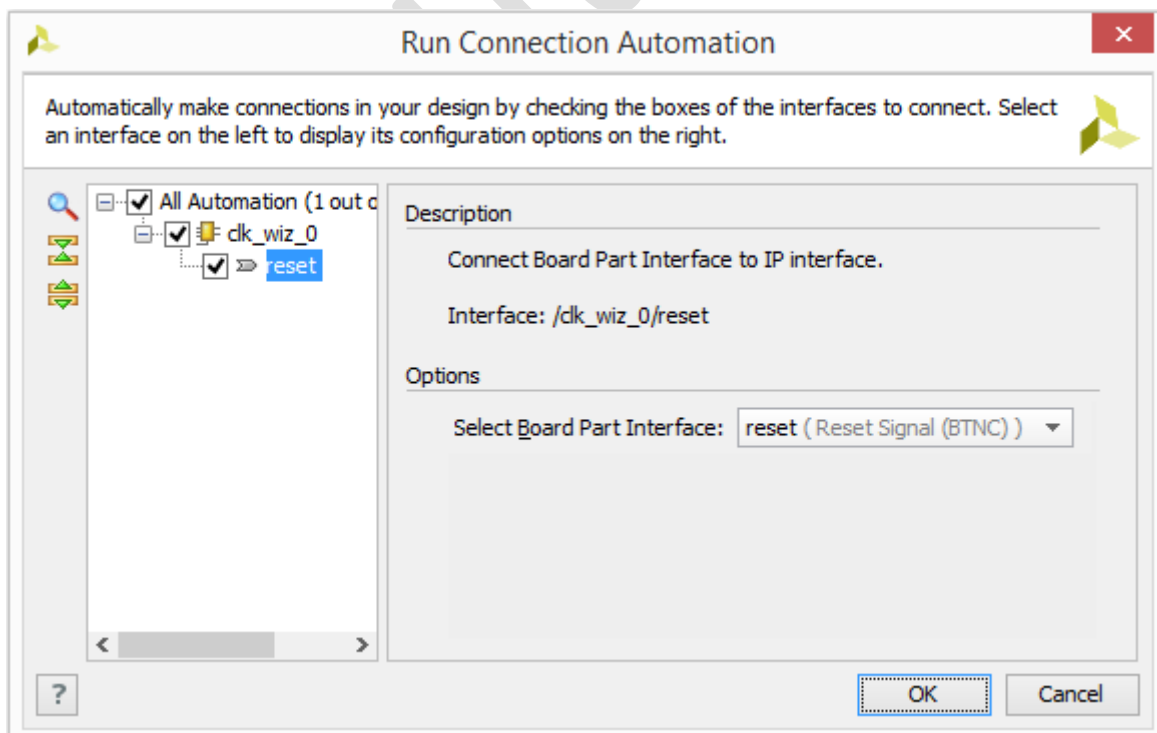


Fig. 23 – Confirmarea alegerii opțiunii „Run Connection Automation”

În spațiul de lucru pentru concepere a modelului se va efectua operația clic dreapta, iar din meniul contextual se va alege opțiunea „Add IP...”. În bara de căutare a ferestrei rezultante se va introduce cuvântul „microblaze”. Se va alege prima opțiune prin efectuarea operației dublu clic. În urma acestor operații se va adăuga în mod automat un bloc de tip „MicroBlaze” în structura modelului dat.

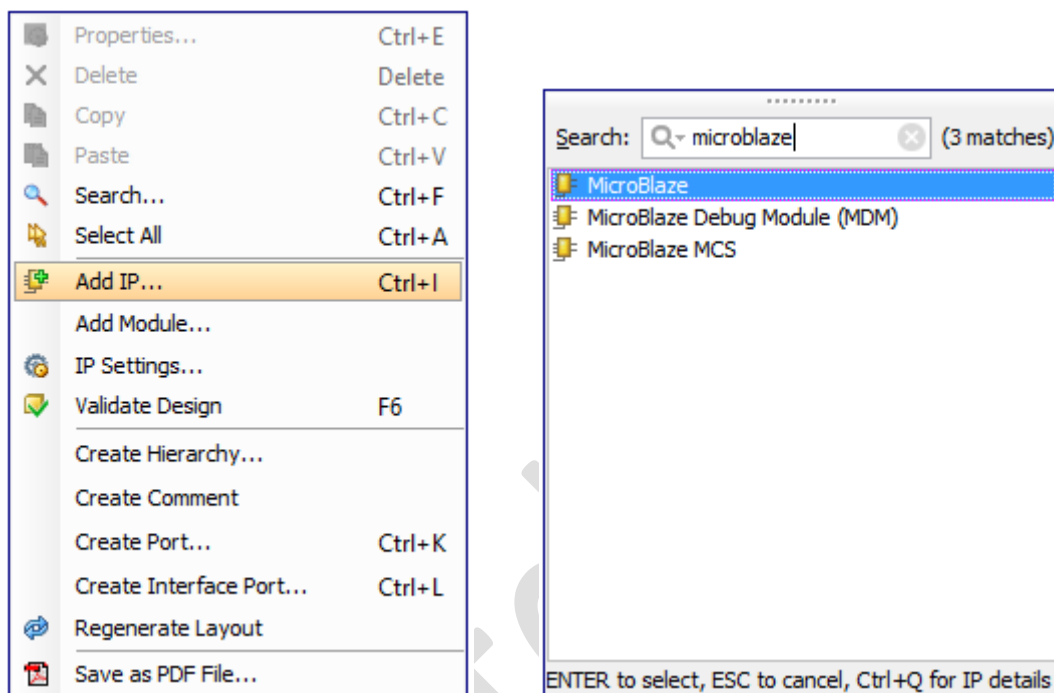


Fig. 24 – Adăugarea blocului specific nucleului MicroBlaze

Pentru a conecta și a particulariza blocul MicroBlaze se va alege opțiunea „Run Block Automation” din bara indicatoare (verde) – „Designer Assistance available”.

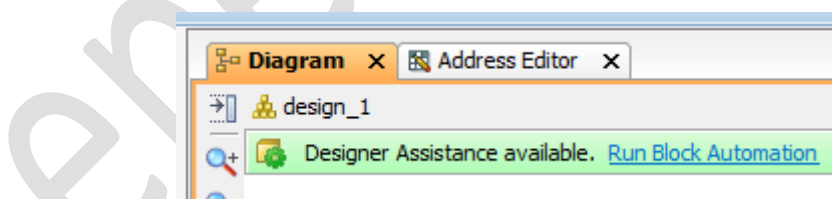


Fig. 25 – Opțiunea „Run Block Automation”

În etapa următoare, se vor alege următorii parametrii:

- Local Memory: 32 KB;
- Local Memory ECC: None;
- Cache Configuration: None;
- Debug Module: Debug Only;
- Peripheral AXI Port: Enabled;
- Interrupt Controller: Nu;
- Clock Connection: /clk_wiz_0/clk_out1 (100MHz);

Realizat de: ing. drd. Pintilie Lucian - Nicolae

Pentru disciplina: „Sisteme cu FPGA și DSP”

Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro



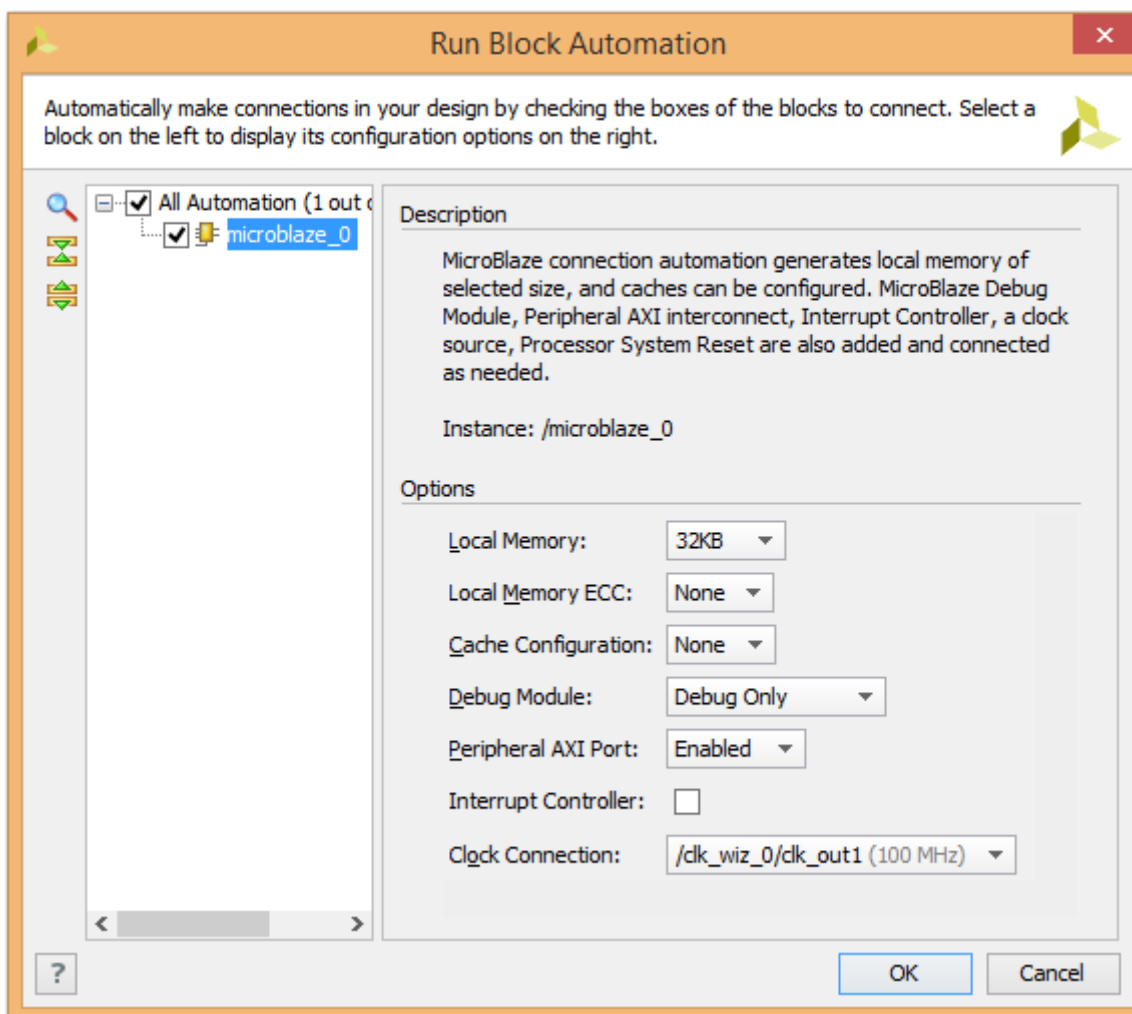


Fig. 26 – Parametrizarea nucleului virtual MicroBlaze

Pentru a finaliza toate conexiunile, se va alege opțiunea „Run Connection Automation”, din bara indicatoare (verde) – „Designer Assistance available”:

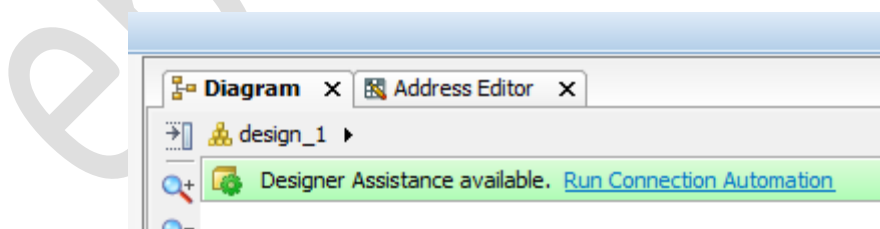


Fig. 27 – Opțiunea „Run Connection Automation”

În urma alegerii opțiunii „Run Connection Automation” se va deschide o fereastră de configurare, în cadrul căreia se va alege opțiunea „All Automation ...”, și se va confirma alegerea prin apăsarea butonului „OK”.

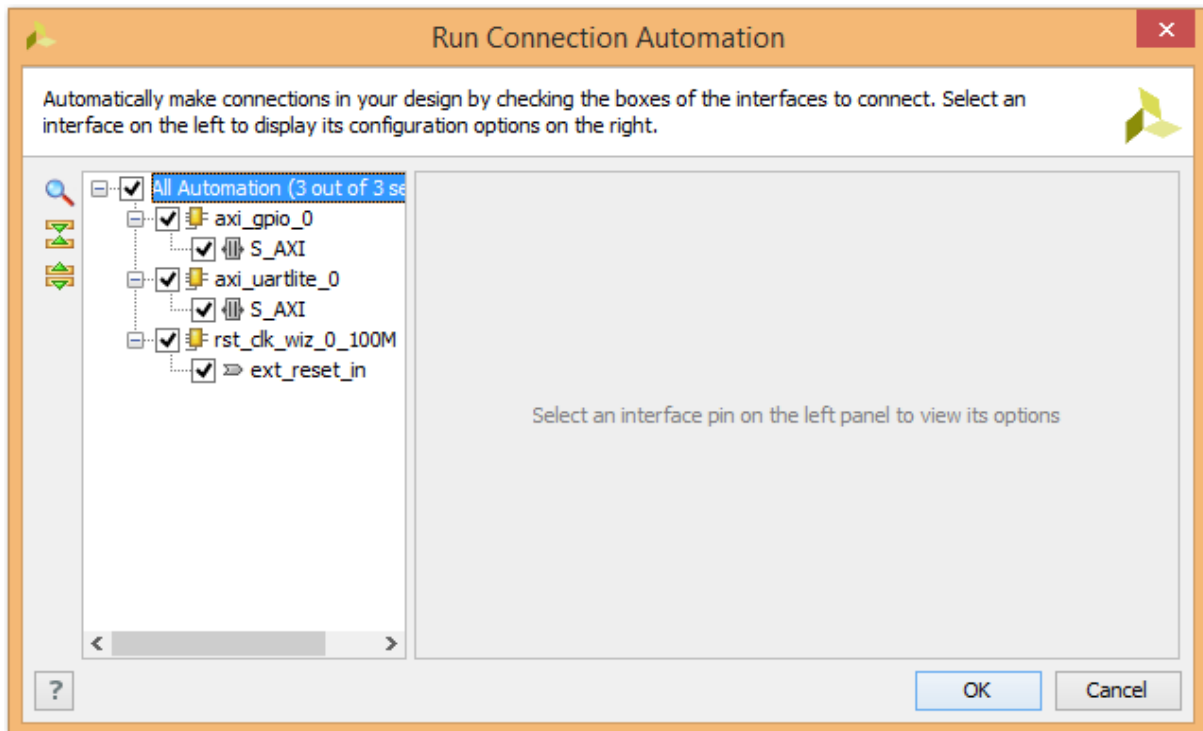


Fig. 28 - Alegerea opțiunea „All Automation ...”

În urma parcurgerii operațiilor amintite, se va obține arhitectura sistemului conceput:

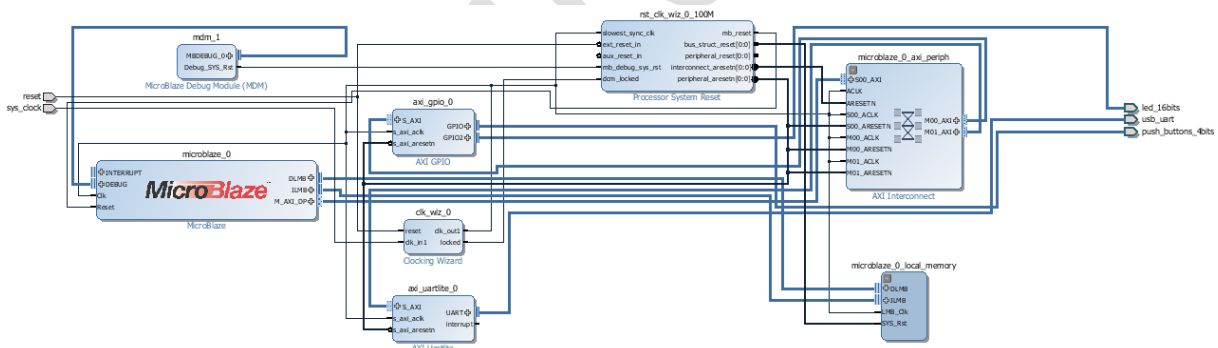


Fig. 29 – Arhitectura sistemului pe bază de procesor MicroBlaze

În continuare, se va verifica integritatea modelului prin efectuarea operației clic dreapta în spațiul de lucru al modelului și alegerea opțiunii „Validate Design” din cadrul meniului contextual. Dacă nu apar erori, se va confirma prin apăsare butonului „OK”.

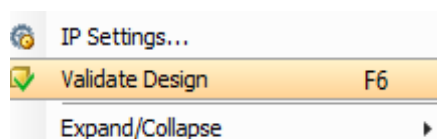


Fig. 30 – Opțiunea „Validate Design”

În continuare, pentru a genera fișierul BitStream pentru reconfigurarea arhitecturii interne a nucleului FPGA, este necesară generarea codului program suport HDL (eng. HDL Wrapper). În cadrul ferestrei „Block Design” în categoria (sau tabul) „Sources” se va selecta din directorul „Design Sources” cu operația clic dreapta din meniul contextual opțiunea „Create HDL Wrapper...”. În cadrul mesajului de confirmare se va alege opțiunea „Let Vivado manage wrapper and auto-update”.

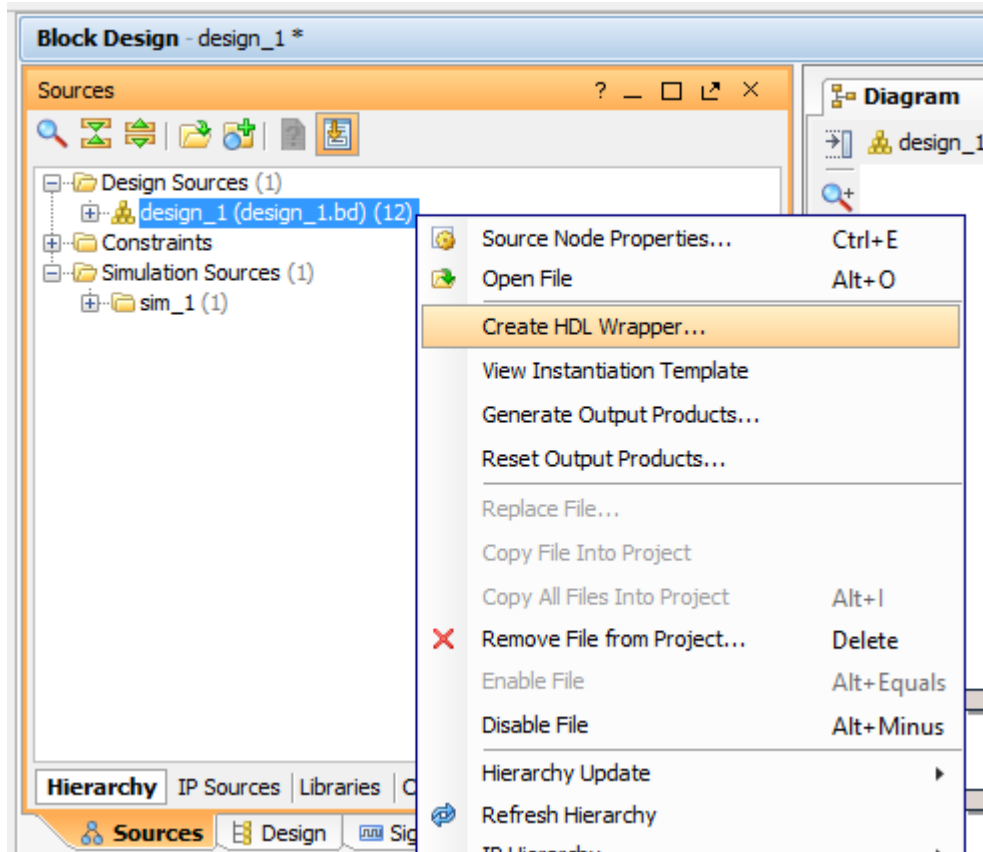


Fig. 31 – Opțiunea „Create HDL Wrapper...”

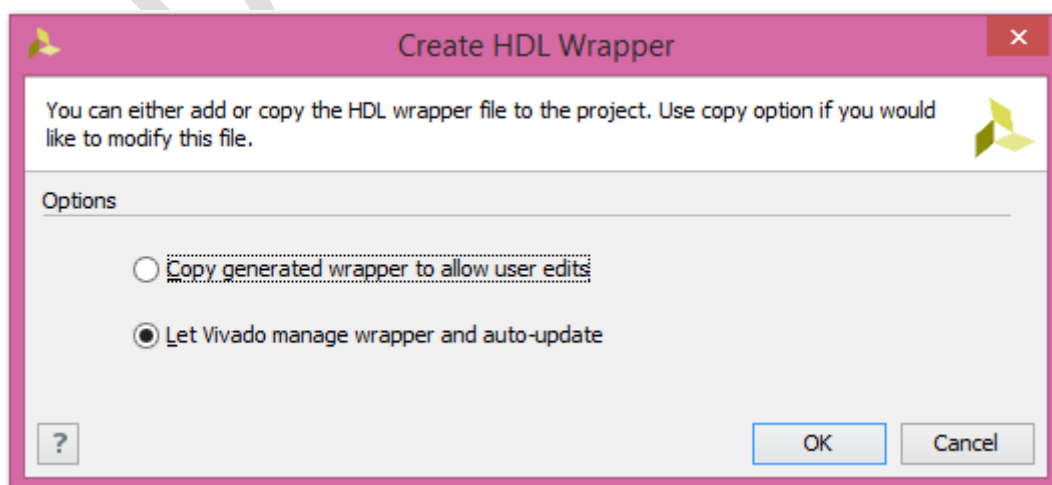


Fig. 32 - Opțiunea „Let Vivado manage wrapper and auto-update”

Următorul pas, constă în generarea fișierului pentru re-configurare a nucleului FPGA, anume, fișierul BitStream. Pentru a realiza această operație, se va alege opțiunea „Generate Bitstream” din bara laterală aflată în partea stângă.

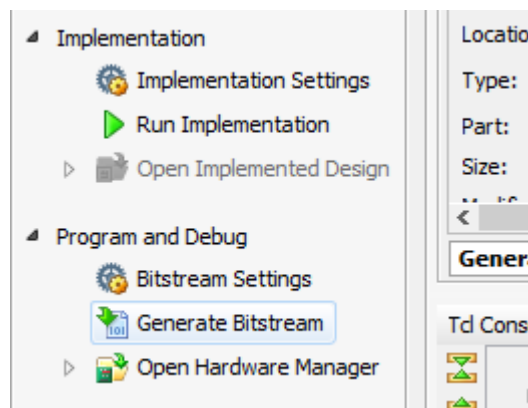


Fig. 33 – Opțiunea „Generate Bitstream”

În cazul în care proiectul nu a fost salvat, mediul Vivado, va cere confirmarea utilizatorului pentru a salva proiectul. Se va alege deci opțiunea „Save”. Mediul Vivado, va solicita de asemenea, confirmarea faptului că, nu există rezultate din faza de implementare. Astfel, se poate trece în mod automat la faza de sintetizare și generare a fișierului BitStream. Se va alege de asemenea, opțiunea „Yes”.

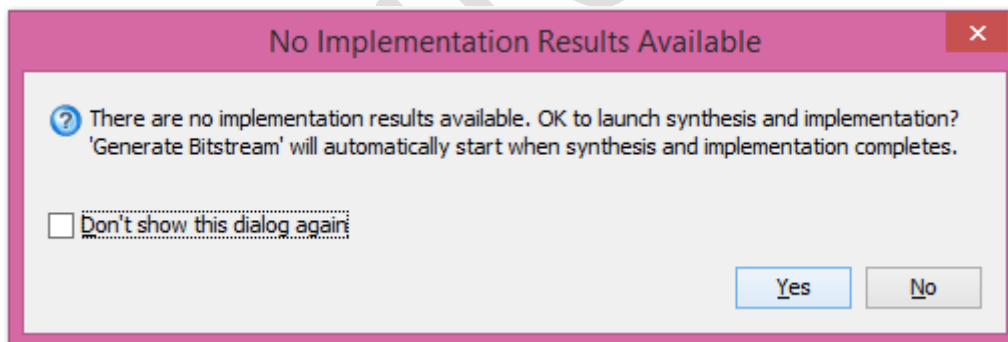


Fig. 34 – Confirmarea etapei de sintetizare și generare a fișierului BitStream

În urma finalizării etapei pentru generare a fișierului BitStream, mediul Vivado va solicita confirmarea următoarei acțiuni. Deoarece nu se dorește nici una din opțiunile afișate în fereastra de dialog, se va alege opțiunea „Cancel” (anulare).

Astfel, etapa de implementare fizică (eng. hardware) a nucleului de procesor MicroBlaze s-a încheiat odată cu generarea fișierului BitStream.

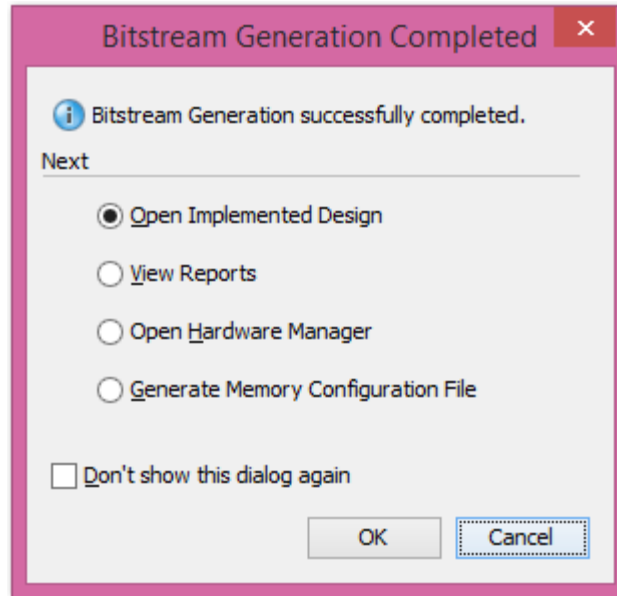


Fig. 35 – Finalizarea procesului pentru generare a fișierului BitStream

B. Implementarea programului la nivel de procesor virtual cu ajutorul mediului Vivado SDK:

Pentru a deschide proiectul creat în cadrul mediului Vivado Design Suite, în mediul de dezvoltare Vivado SDK, din cadrul meniului principal „File” se va alege opțiunea „Export” apoi opțiunea „Export Hardware...”

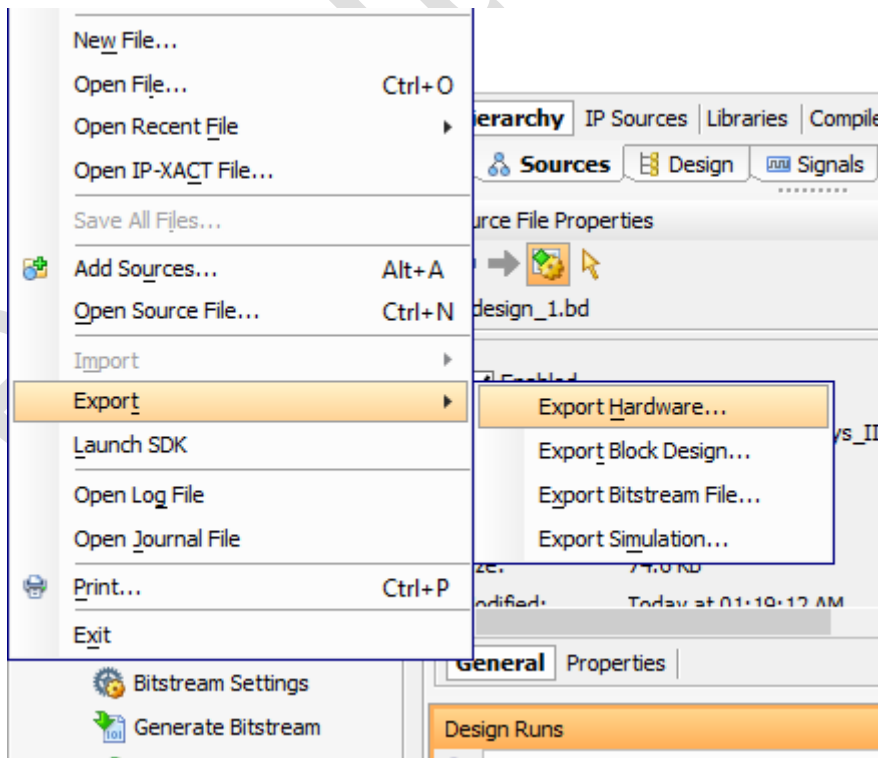


Fig. 36 – Alegerea opțiunii „Export Hardware...”

Pentru a confirma procesul de exportare a arhitecturii concepute anterior, se va selecta și opțiunea „Include bitstream” iar locația se va păstra cea implicită (Local to Project).

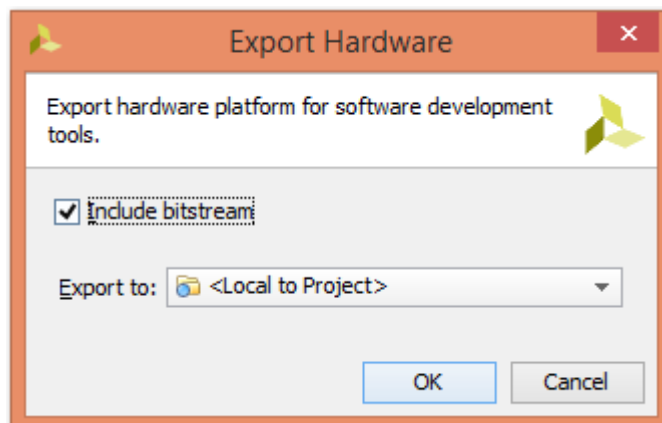


Fig. 37 – Confirmarea procesului de exportare

În urma procesului de exportare a arhitecturii fizice (eng. hardware), se va trece la etapa de lansare în execuție a mediului de dezvoltare Vivado SDK. Această operație se va realiza prin accesarea opțiunii „Launch SDK” din cadrul meniului principal „File”.

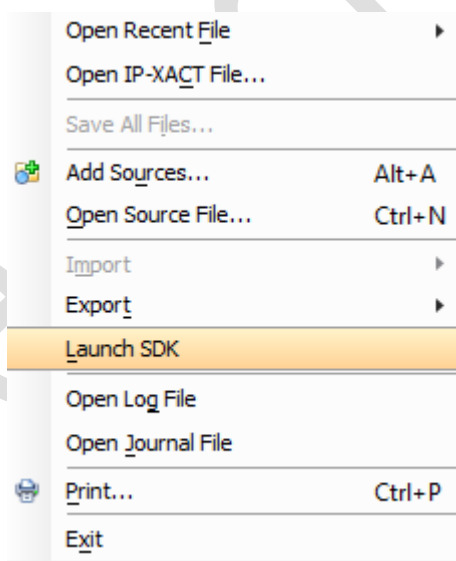


Fig. 38 – Lansarea în execuție a mediului Vivado SDK

Pentru a confirma lansarea în execuție a mediului Vivado SDK, se vor păstra opțiunile implicite oferite de către mediul Vivado Design suite, anume:

- Exported location: <Local to Project>;
- Workspace: <Local to Project>;

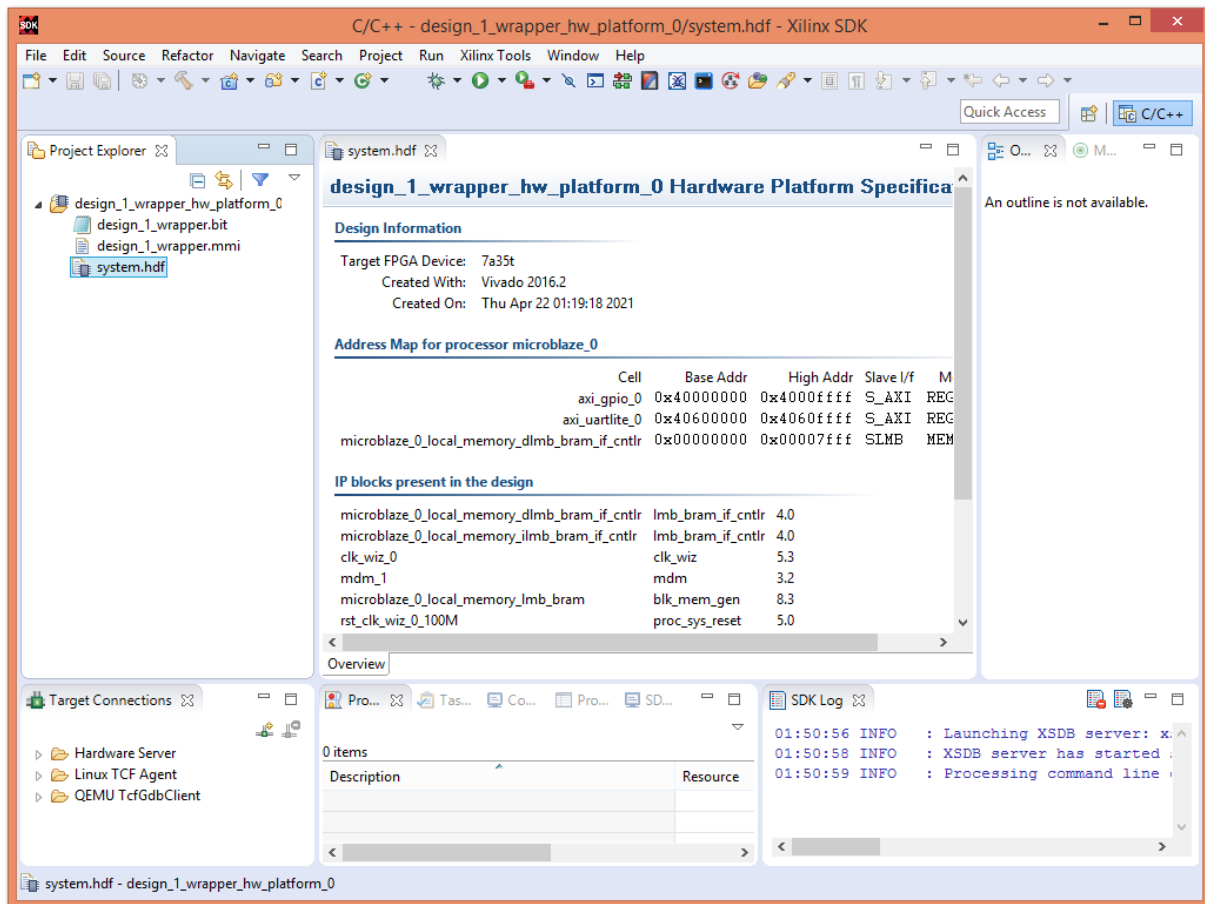


Fig. 39 – Mediul de dezvoltare Vivado SDK

Pentru a crea un proiect, se va alege opțiunea „Application Project” din meniul File.

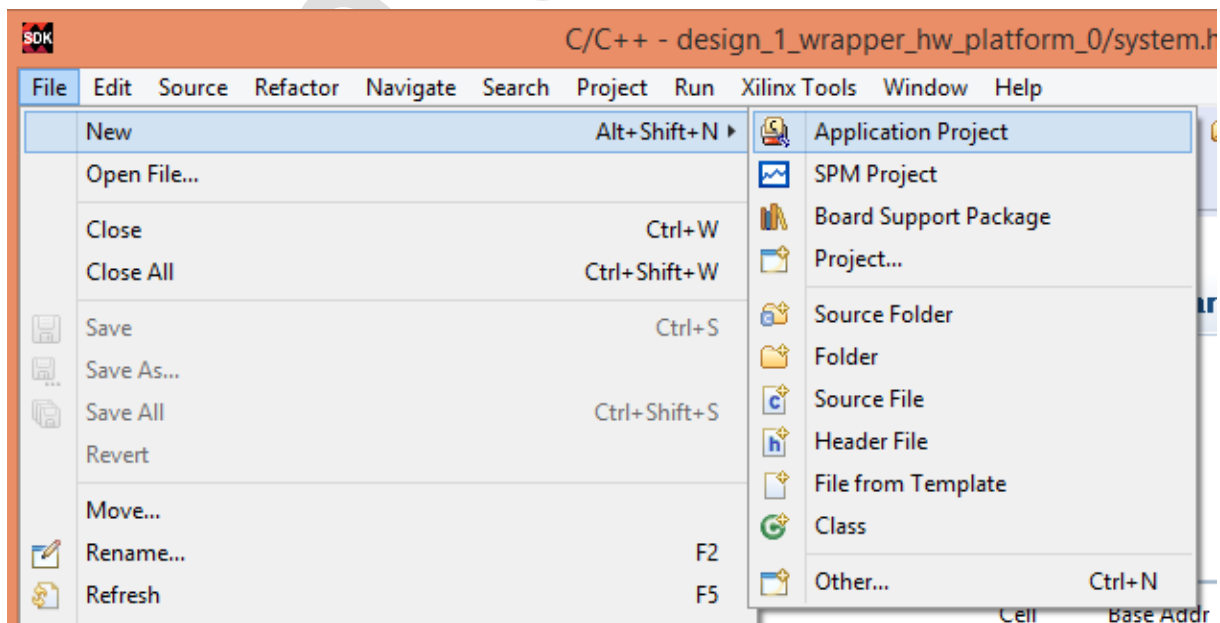


Fig. 40 – Inițializarea unui proiect nou – opțiunea „Application Project”

În continuare se vor specifica următorii parametri:

- Project name: Basy3_MicroBlaze_GPIO_test;
- Use default location: Da;
- OS Platform: standalone;
- Hardware Platform: design_1_wrapper_hw_platform_0;
- Processor: microblaze_0;
- Language: C (standard);
- Board Support Package: Create New: Basy3_MicroBlaze_GPIO_test_bsp

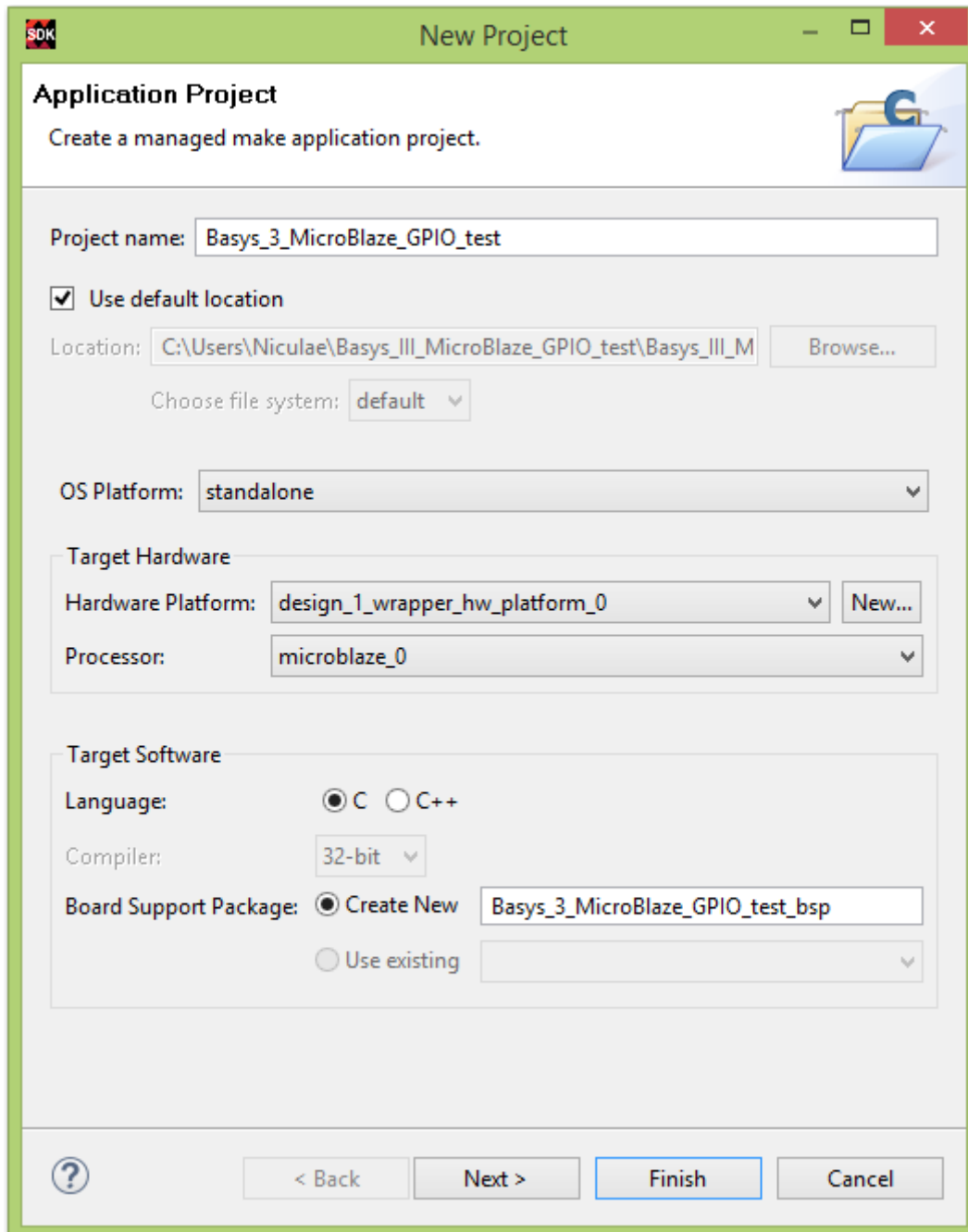


Fig. 41 – Definierea parametrilor unui nou proiect în Vivado SDK

În continuare, se va alege tipul predefinit de aplicație, anume „Empty Application”.

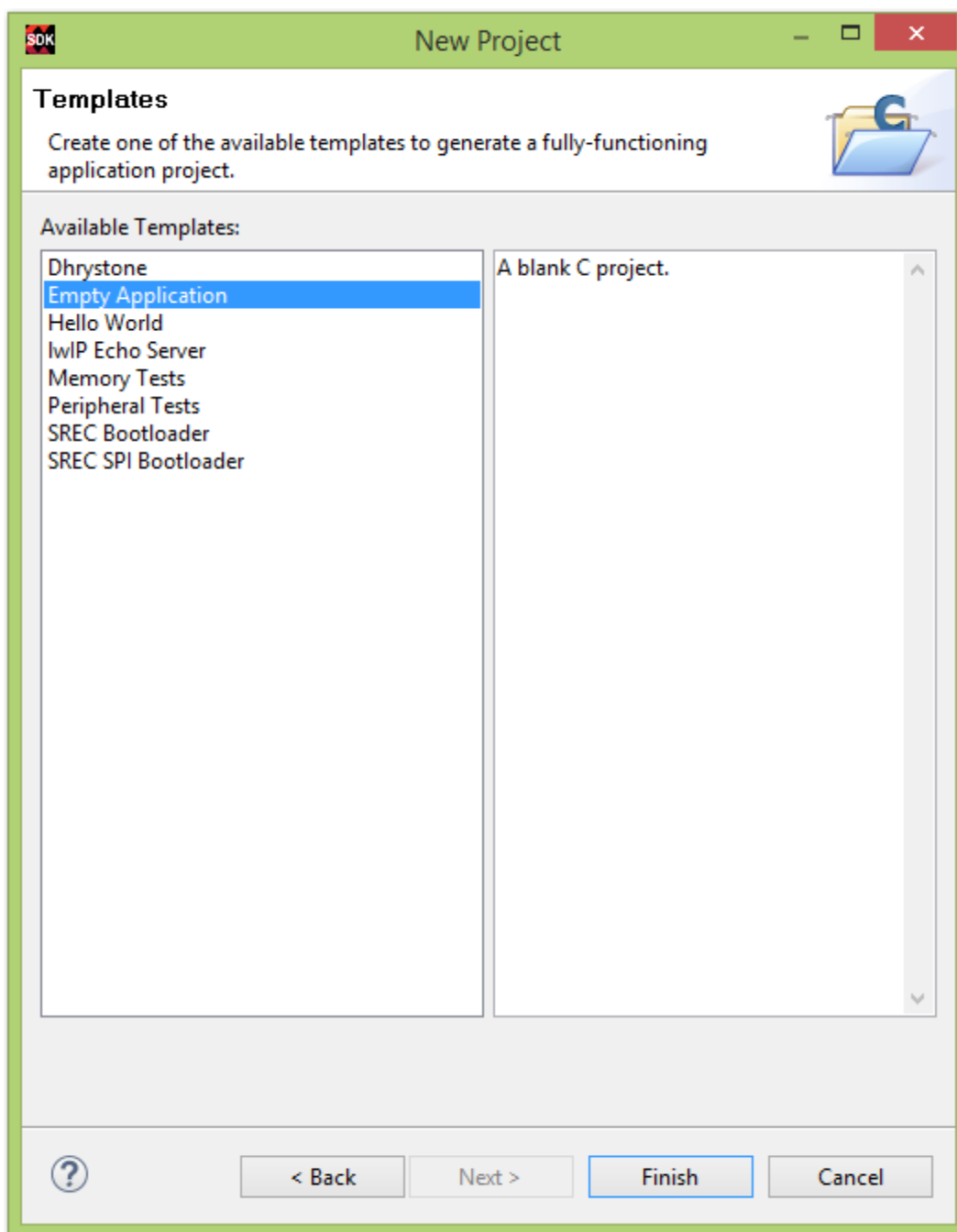


Fig. 42 – Alegerea tipului de aplicație – Empty Application

Se va finaliza partea de inițializare a proiectului prin apăsarea butonului „Finish”. În vederea implementării programului la nivel de procesor virtual, este necesar să se creeze un fișier sursă „C standard” / „C++”. Astfel, în cadrul directorului principal al proiectului, mai precis în subdirectorul „src” (fișiere sursă), cu ajutorul operației clic dreapta, din meniul contextual se va alege opțiunea „New” → „Source File”. În următoarea etapă, se va defini fișierul sursă. Se vor păstra opțiunile implicite. Se va alege denumirea „main.c” a fișierului.

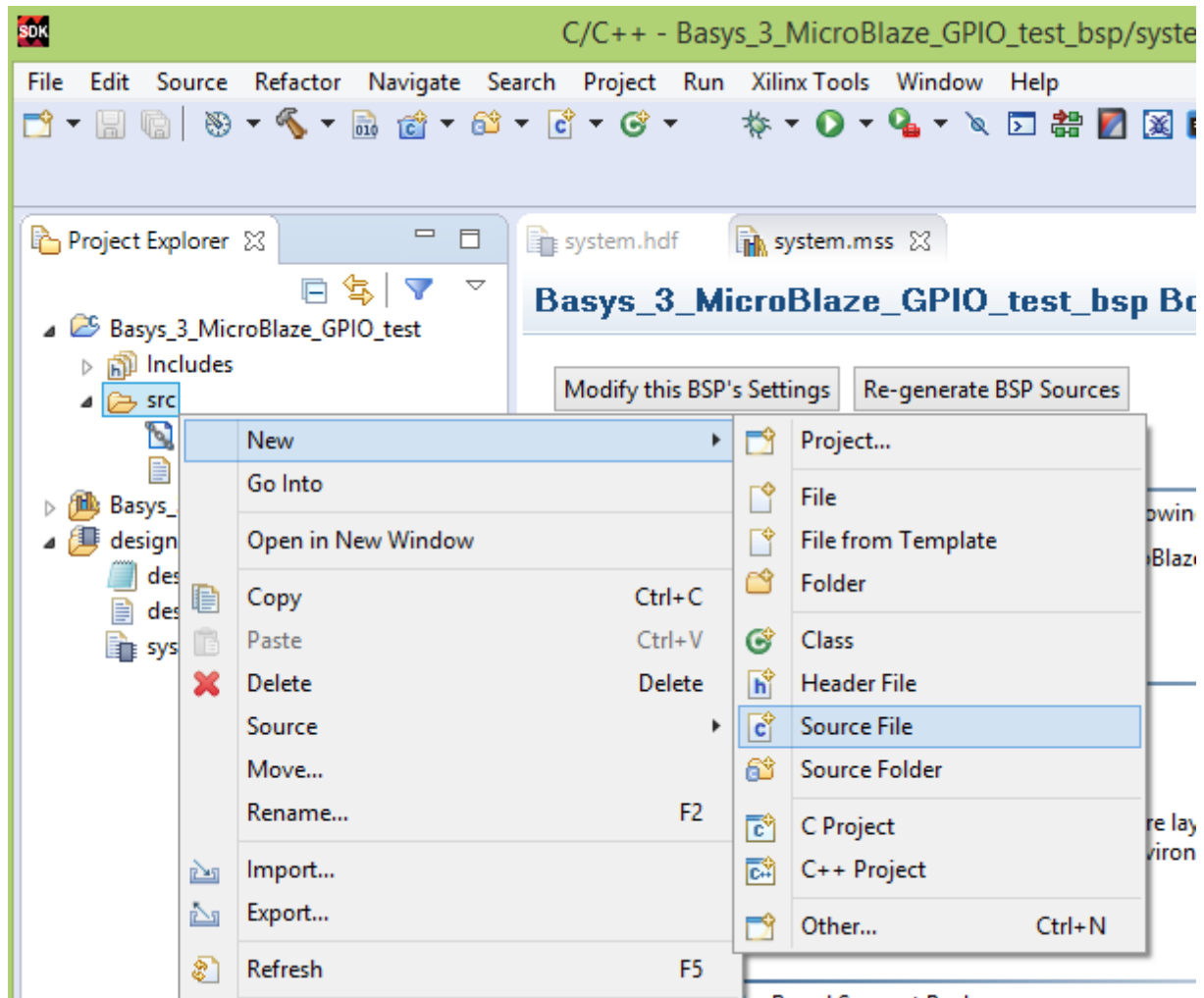


Fig. 43 – Opțiunea „New” → „Source File”

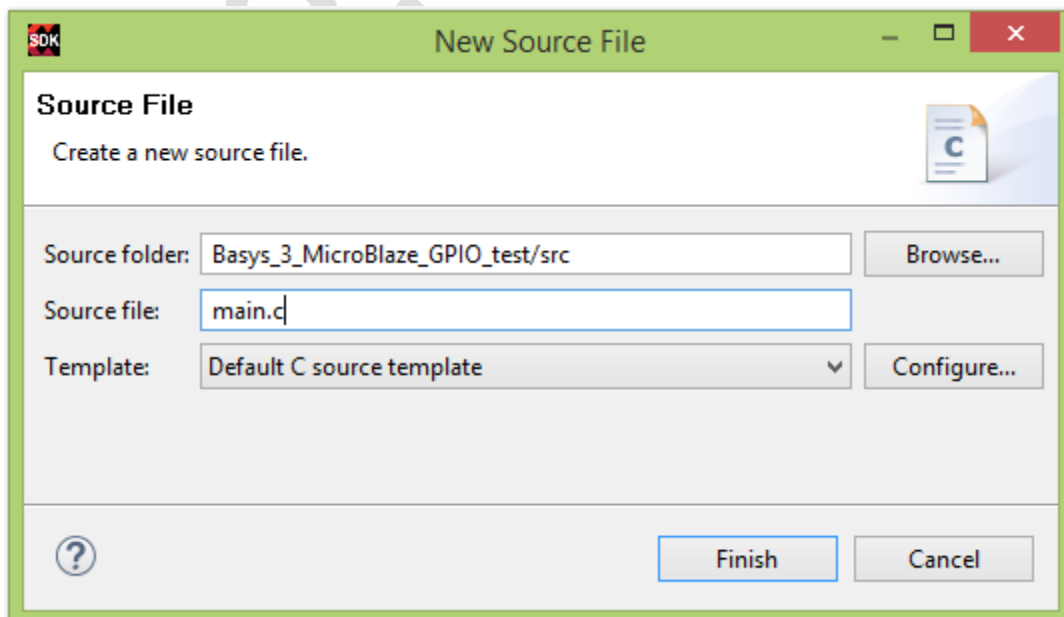


Fig. 44 – Definirea fișierului sursă

Se va utiliza următorul conținut pentru fișierul sursă:

```
//biblioteca pentru interfata AXI GPIO
#include "xgpio.h"

//Biblioteca pentru interfata UART
#include "xil_printf.h"

//Denumirile pinilor pentru interfata AXI
#include "xparameters.h"

int main()
{
    XGpio gpio;
    u32 btn, led;

    XGpio_Initialize(&gpio, 0);

    XGpio_SetDataDirection(&gpio, 2, 0x00000000); //
Echivalent cu pinMode(x, OUTPUT)
    XGpio_SetDataDirection(&gpio, 1, 0xFFFFFFFF); //
ECHIVALENT cu pinMode(x, INPUT)

    while (1)
    {
        btn = XGpio_DiscreteRead(&gpio, 1);

        if (btn != 0) // Aprinderea tuturor indicatorilor la
apasarea butoanelor
            led = 0xFFFFFFFF;
        else
            led = 0x00000000;

        XGpio_DiscreteWrite(&gpio, 2, led);

        xil_printf("\rbutton state: %08x", btn); // Afisare
in consola Serial a starii butonului
    }
}
```

În urma editării fișierului sursă, se va introduce comanda „Ctrl + S” din tastatură pentru a salva conținutul fișierului sursă. Se va proceda mai departe la programarea platformei Basys.

Pentru a încărca fișierul BitStream necesar configurării nucleului FPGA, în structura de procesor virtual MicroBlaze, se va alege din meniul „Xilinx Tools” opțiunea „Program FPGA”. În cadrul ferestrei de confirmare, se vor păstra valorile implicite, și se va alege opțiunea „Program”, pentru a încărca fișierul BitStream în memorie.

Rolul programului implementat este de a prelua starea digitală de la registrul intrărilor digitale (de la butoane) și să transmită valoarea înspre registrul ieșirilor digitale (LED). Totodată, programul afișează rezultatul în consola serial.

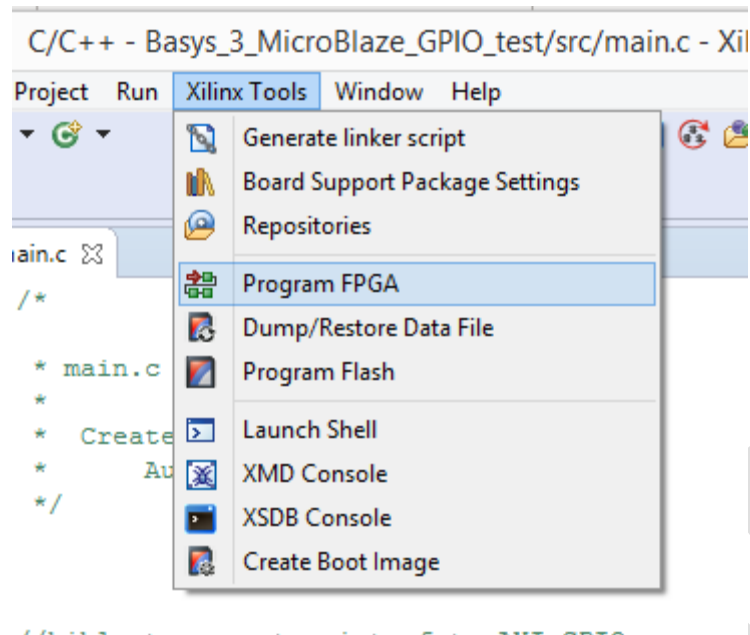


Fig. 45 – Opțiunea „Program FPGA”

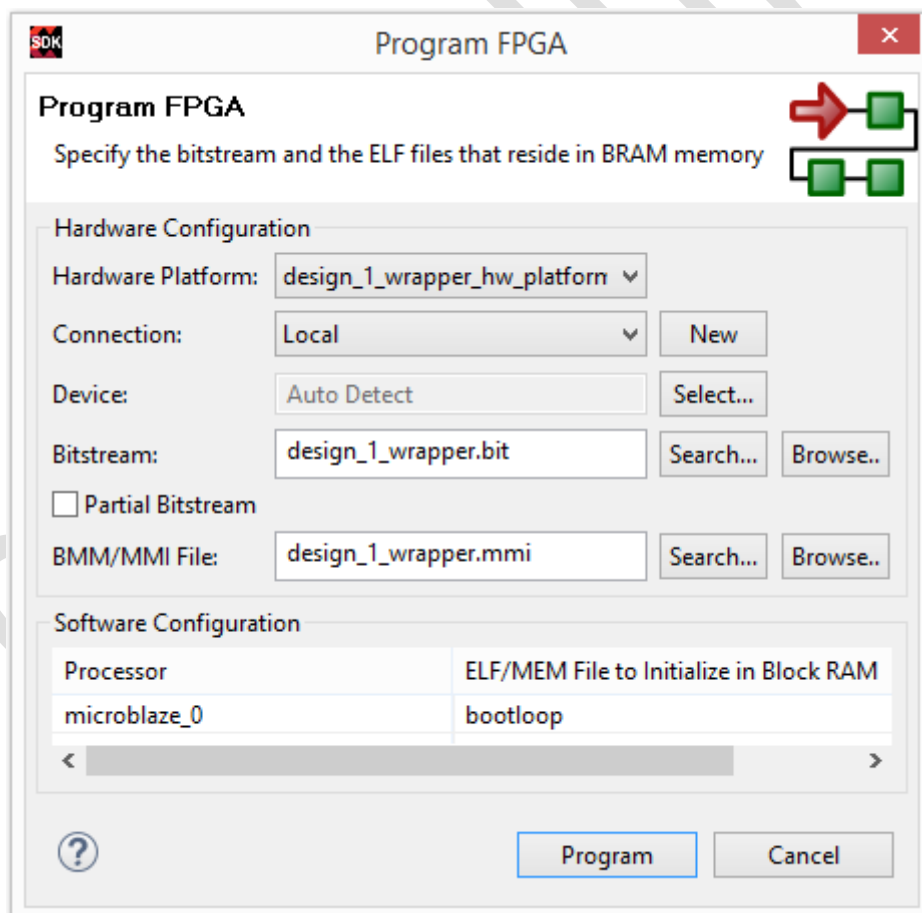


Fig. 46 – Butonul „Program”

În urma acestor operații, procesorul virtual a fost implementat la nivel de FPGA, astfel, programul C / C++ poate fi încărcat și executat la nivel de platformă de dezvoltare. Pentru a realiza acest lucru, din meniul „Run” se va alege opțiunea „Run As” → „Launch on Hardware (System Debugger)”.

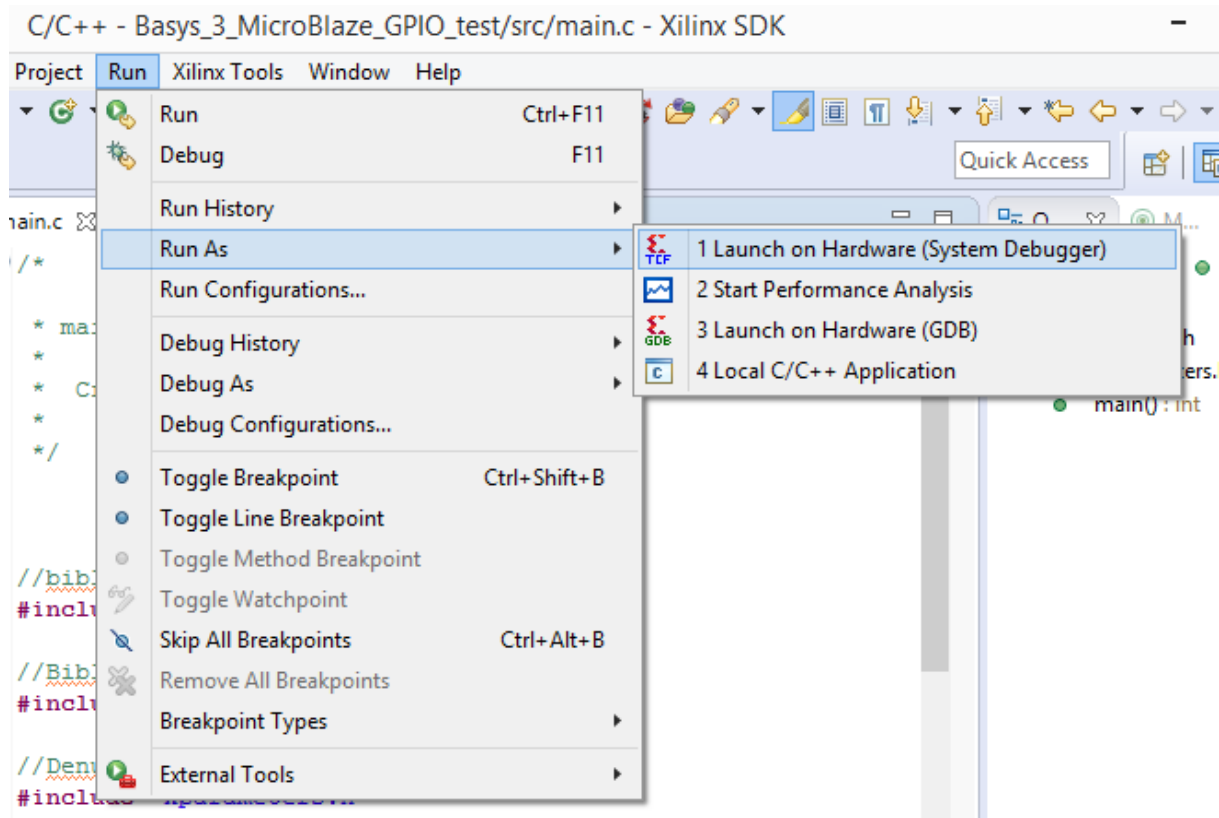


Fig. 47 – Opțiunea - „Launch on Hardware (System Debugger)”



Fig. 48 – Afișarea în consola Serial a stării registrului GPIO la care sunt atașate butoanele

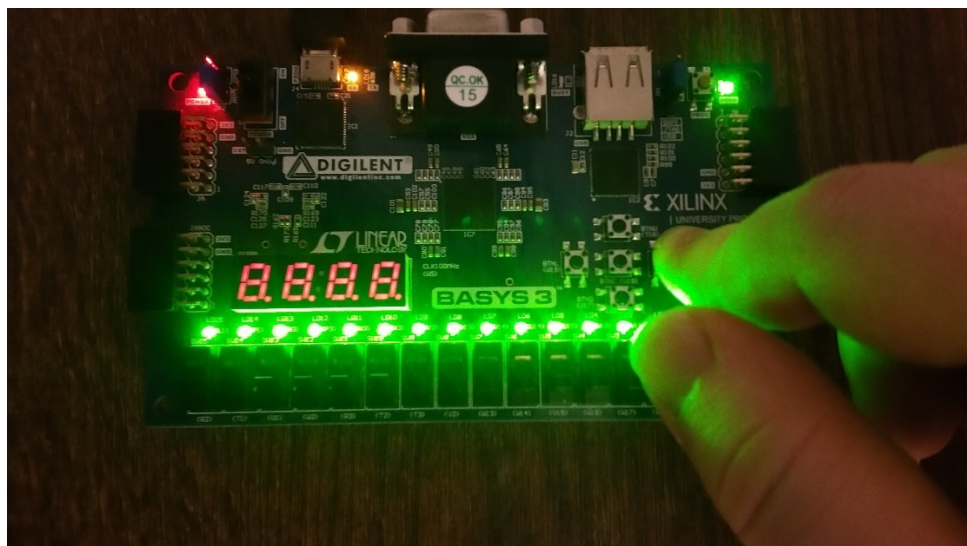


Fig. 49 – Aprinderea indicatorilor luminoși la apăsarea butoanelor momentane

III. CONCLUZIE:

În cadrul unui nucleu FPGA, se poate implementa și o structură logică mai complexă precum arhitectura unui micro-controller sau a unui microprocesor.

IV. BIBLIOGRAFIE:

1. reference.digilentinc.com – „Getting Started with Vivado IP Integrator”:
(<https://reference.digilentinc.com/vivado/getting-started-with-ipi/start>);
2. rsdelivers.com – „Digilent 410-183 Basys Artix-7 Development Board”:
(<https://ro.rsdelivers.com/product/digilent/410-183/digilent-410-183-basys-artix-7-development-board/1346451>);
3. allaboutcircuits.com – „Utilizing Xilinx’s MicroBlaze in FPGA Design - MicroBlaze core architecture”: (<https://www.allaboutcircuits.com/industry-articles/utilizing-xilinx-microblaze-in-fpga-design/>);
4. wikipedia.org – „Advanced eXtensible Interface – (AXI interface)”:
(https://en.wikipedia.org/wiki/Advanced_eXtensible_Interface);
5. Teodor Crișan Pană – „Sisteme de calcul cu microprocesoare, FPGA și DSP” – Editura UTPRESS, Cluj – Napoca, 2016 – ISBN 978-606-737-206-9;
6. Ioana – Cornelia GROS, Lucian – Nicolae PINTILIE, Teodor Crișan PANĂ – „SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ - GHID DE APLICAȚII” – Editura UTPress Cluj – Napoca, 2020 ISBN 978-606-737-431-5:
(<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/431-5.pdf>);

Realizat de: ing. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme cu FPGA și DSP”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

