

Lucrarea de laborator 1

Elaborarea unui proiect schematic

Scopul lucrării:

Proiectarea, verificarea și simularea circuitelor logice în mediul de proiectare software Altera Quartus II, utilizând descrierea schematică a circuitului.

Considerații teoretice și practice

Fiecare circuit logic, proiectat de utilizator în Quartus II poartă numele de proiect. Doar un singur proiect poate fi activ la un moment dat. Toate fișierele de lucru, necesare pentru proiectul activ se păstrează într-un director (mapă). De aceea, prima etapă în elaborarea unui nou proiect este crearea directorului pentru păstrarea fișierelor de lucru.

1. Crearea unui proiect nou

Pentru crearea unui nou proiect se selectează opțiunea *New Project Wizard* din meniul *File*.

File → *New Project Wizard*

În fereastra care se deschide se selectează directorul de lucru pentru proiect, se introduce un nume pentru proiectul curent care, în mod automat se înscrie și pentru entitatea principală a proiectului (Figura 1).

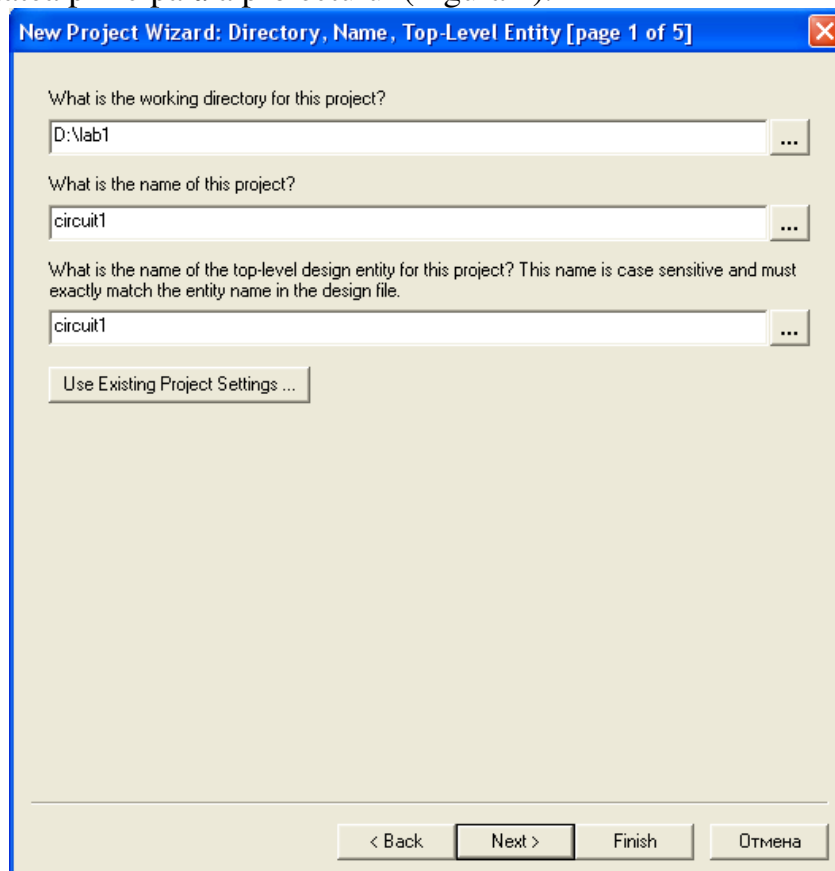


Figura 1. Directorul de lucru și numele proiectului

În fereastra următoare se pot include în proiect eventualele fișiere existente și/sau să se specifice librării noi, diferite de cele implicite (figura 2).

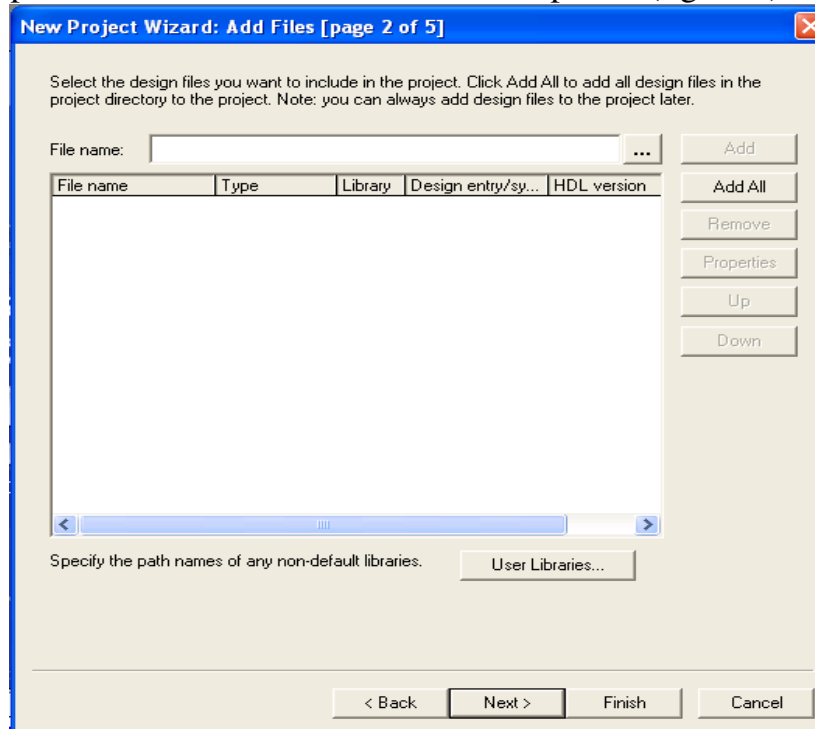


Figura 2. Adăugarea de fișiere noi sau selectarea altor librării

În cea de-a treia fereastră se selectează dispozitivul hardware utilizat. O posibilă variantă a selecției este arătată în Figura 3.

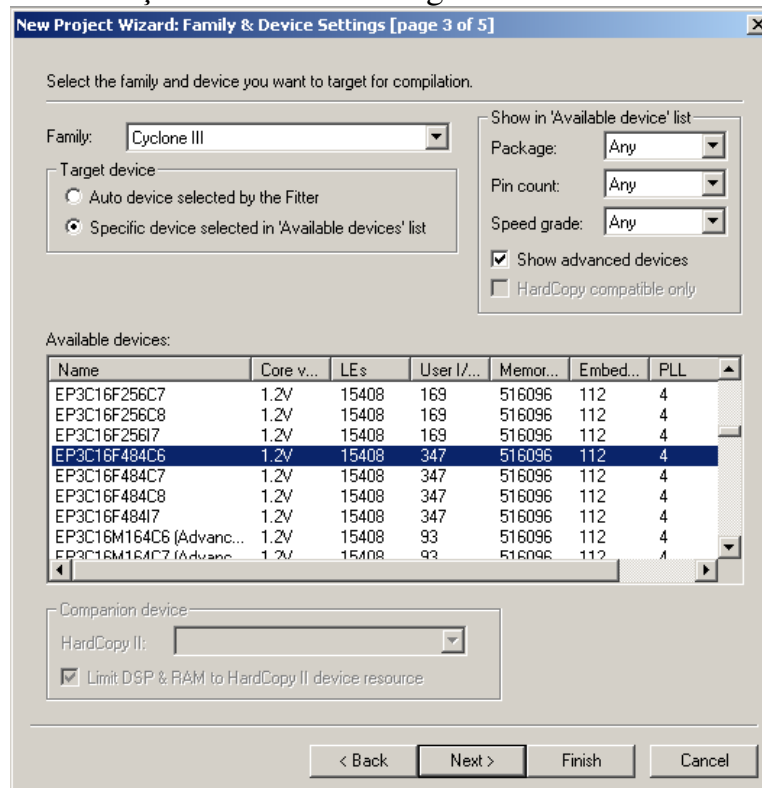


Figura 3. Selecția circuitului FPGA

Fereastra următoare oferă posibilitatea de a selecta unelte EDA (Electronic Design Automation) adiționale față de cele incluse deja în mediul de dezvoltare Quartus II. Astfel se pot selecta unelte pentru introducerea schemelor și sinteză, pentru simulare sau pentru analiza temporară. Deoarece nu vom utiliza astfel de unelte, în această fereastră nu vom face nici o setare. Fereastra este prezentată în figura 4.

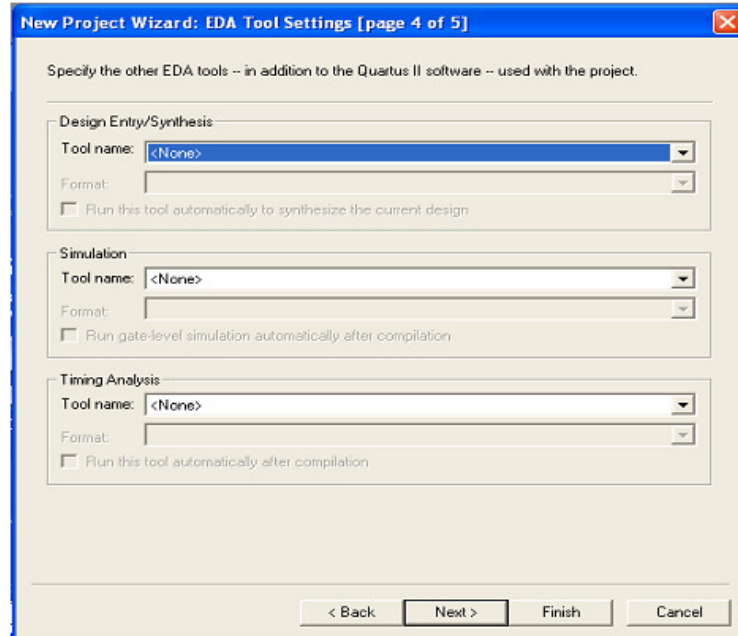


Figura 4. Selecția uneltelor EDA externe

În ultima fereastră a wizardului (Fig. 5) avem prezentat un sumar al selecțiilor făcute până acum. Pentru a finaliza procesul de creare al unui proiect nou se dă clic pe butonul *Finish*.

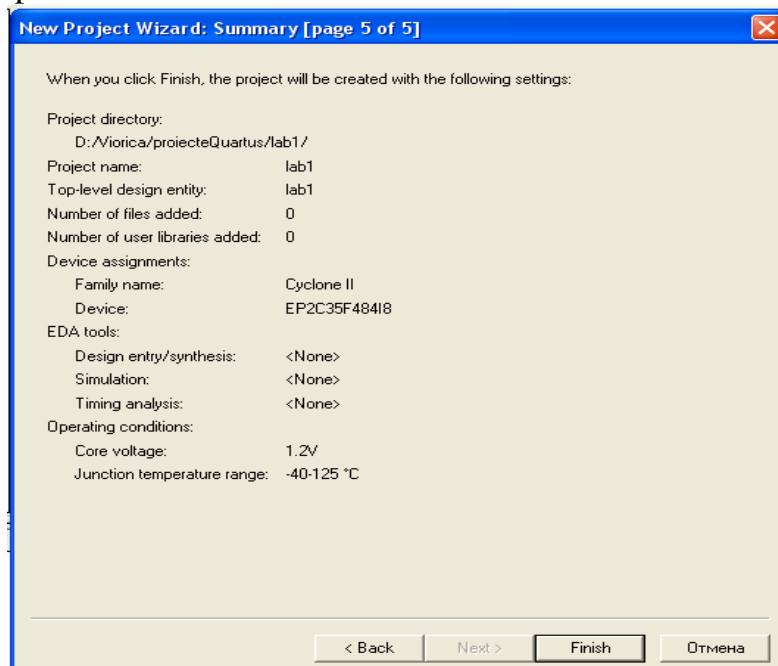


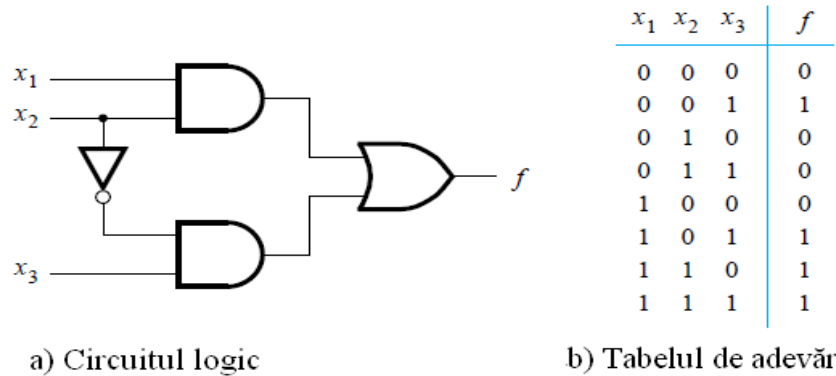
Figura 5. Sumarul setărilor efectuate

2. Implementarea schemei

Mediul de proiectare software Altera Quartus II oferă posibilitatea de a elabora proiecte schematice sau în limbajele de programare hardware AHDL, VHDL și Verilog.

Etapele de implementare a unui proiect schematic vor fi analizate în baza următorului exemplu:

Fie dată funcția $f = x_1x_2 + \bar{x}_2x_3$. Circuitul logic și tabelul de adevăr sunt prezentate în figura 6.



a) Circuitul logic

b) Tabelul de adevăr

Figura 6. Circuitul logic și tabelul de adevăr

Pentru a putea implementa schema, se va crea un fișier nou, de tipul **Block Diagram / Schematic File**. Pentru aceasta, din meniul **File** se alege opțiunea **New**. Din fereastra nou deschisă se alege opțiunea **Block Diagram / Schematic File** din cadrul opțiunii **Design Files**, după cum se poate observa în figura 7.

File → **New** → **Block Diagram / Schematic File**

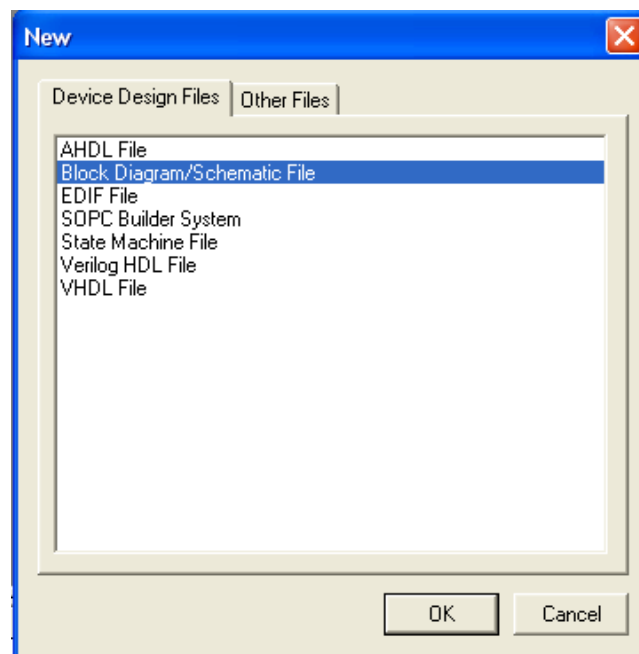


Figura 7. Selecția tipului de fișier

Dând clic pe butonul *OK* se va deschide fereastra de lucru pentru introducerea schemei (Figura 8).

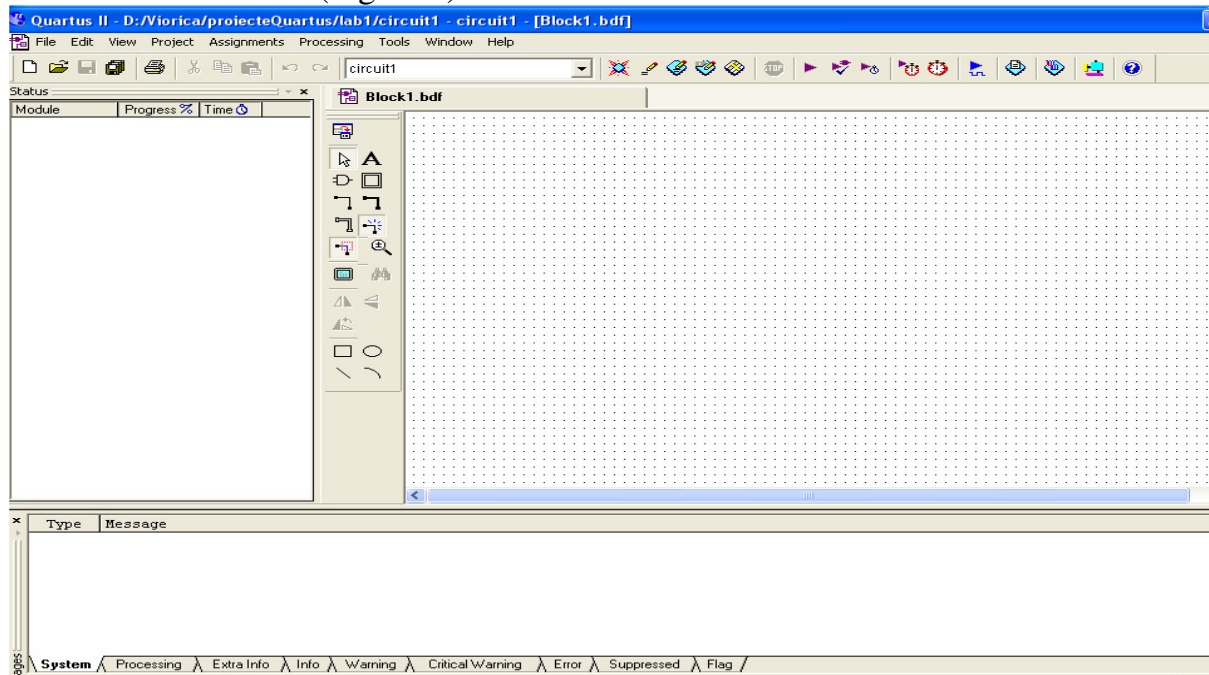



Figura 8. Fereastra de lucru pentru fișierul cu diagrama bloc.

Se va salva fișierul cu schema bloc prin selecția opțiunii *Save as...* din meniul *File*.

De reținut că numele fișierului .bdf (block diagram file.) trebuie să fie numele declarat la crearea proiectului pentru entitatea principală.

După salvarea fișierului cu schema bloc se poate trece la implementarea schemei. Pentru a se introduce o componentă în schemă, se dă dublu clic pe foaia de lucru sau se tastează butonul .

Se deschide astfel fereastra de selecție a simbolurilor din cadrul bibliotecilor. Se selectează din lista de biblioteci *primitives* → *logic* și se aleg porțile logice necesare pentru implementarea circuitului.

De exemplu, pentru a selecta poarta logică *and2*, se face dublu-clic pe ea. Simbolul porții logice va apărea în fereastra curentă. Tastând butonul *OK*, ea poate fi importată în fereastra de lucru. Pentru a face mai simplă poziționarea porților logice în fereastra de lucru se poate afișa grila de lucru, selectând *View* → *Show Guidelines*. Pentru a schimba direcția de orientare a porții logice *Not*, se selectează poarta și se alege opțiunea *Edit* → *Rotate by Degrees* → *270*.

După plasarea tuturor porților logice, este necesar de a importa și pinii pentru conectarea circuitului în exterior. Pentru aceasta vom da dublu clic pe suprafața liberă a foii de lucru, intrând astfel în fereastra de selecție a simbolurilor din biblioteci. De această dată vom selecta din lista de biblioteci *primitives* → *pin* → *input* pentru a selecta pinii de intrare, respectiv *output* pentru pinii de ieșire. Se dă clic pe butonul *OK* și se revine în foaia de lucru, cu pinul activ. Pentru plasarea pinului, se dă clic pe foaia de lucru în locul dorit. Se repetă operațiunea până sunt amplasați toți pinii.

După plasarea tuturor pinilor pe foaia de lucru se trece la conectarea lor cu terminalele porților logice. Pentru aceasta se dă clic pe butonul **Orthogonal Node Tool** aflat în lista de butoane din partea stângă a foii de lucru și se unesc, pe rând, terminalele porților logice cu terminalele pinilor corespunzători. Pentru schimbarea numelor pinilor, se dă dublu clic pe fiecare pin în parte și, în fereastra de configurare ce se deschide, se introduce numele dorit pentru fiecare pin.

După ce circuitul este implementat, fișierul se salvează, selectând **File**→**Save**.

Circuitul logic obținut este prezentat în figura 9.

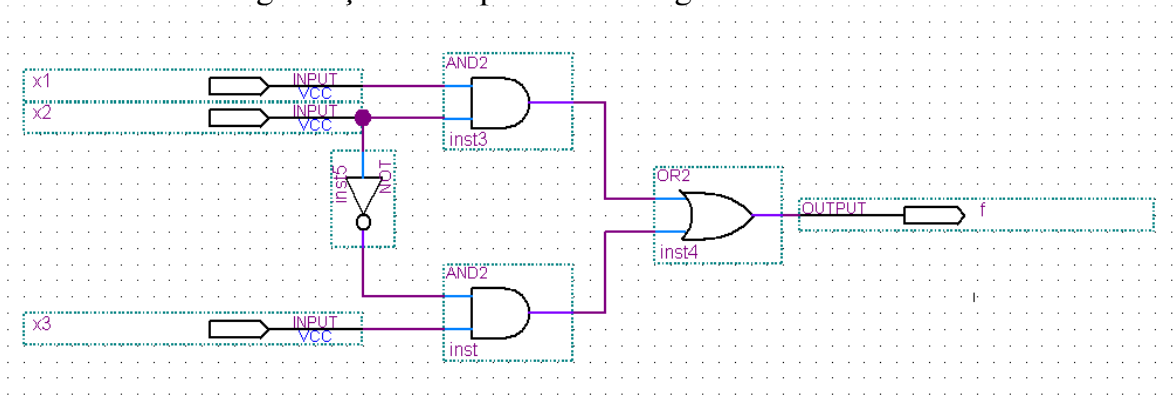


Figura 9. Circuitul logic implementat

O altă posibilitate de a selecta porți logice și elemente funcționale mai complexe este utilizarea bibliotecii **megafunctions**→**gates**, care permite configurarea componentei selectate.

3. Compilarea proiectului

După ce a fost creat fișierul cu extensia .bdf , proiectul trebuie compilat. Compilatorul reprezintă o aplicație program care gestionează mai multe module CAD (Computer Aided Design).

Compilarea proiectului

Accesând **Processing**→**Compiler Tool**, se deschide o fereastră, în care pot fi vizualizate patru module CAD de bază (Figura 10). Modulul **Analysis & Synthesis** efectuează operația de sinteză și generează un circuit format din elemente logice care pot fi direct implementate pe chipul ales. Modulul **Fitter** alege locația exactă pe chip, unde elementele sintetizate vor fi implementate.

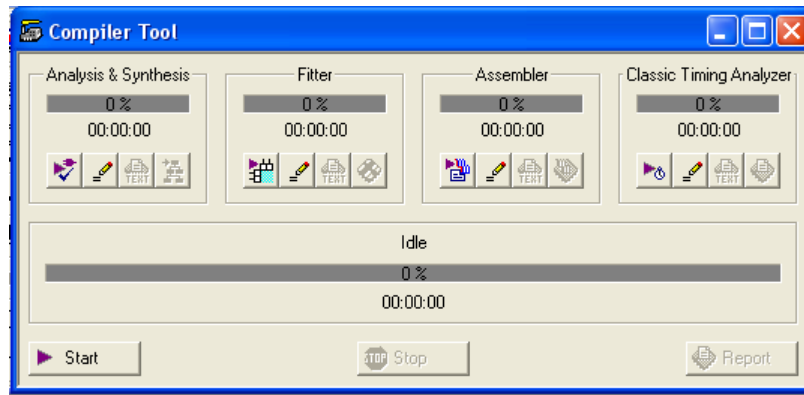



Figura 10. Module CAD

Selectând **Processing** → **Start Compilation** sau tastând butonul **Start Compilation**  se inițiază procesul de compilare, care utilizează, în mod consecutiv, modulele din Figura 10. La diferite etape ale proiectului pot fi necesare doar unele instrumente de compilare.

În mediul de proiectare Quartus II pot fi efectuate două tipuri de simulare:

- **simularea funcțională** (Functional Simulation), în care se analizează doar logica circuitului, fără a lua în considerare întârzierile reale pe chip;
- **simulare temporală** (Timing Simulation), care este efectuată, ținând cont de întârzierile reale pe chip.

În cazul simulării funcționale, procesul de compilare poate să se limiteze doar la analiză și sinteză. Pentru a face acest lucru selectăm:

Processing → **Start** → **Start Analysis & Synthesis**

După compilare se generează raportul de compilare (Figura 11). Raportul conține informația utilă pentru proiectant. În cazul circuitului analizat, se poate observa că sunt utilizați doar patru pini și un singur element logic din circuitul Cyclone II FPGA.

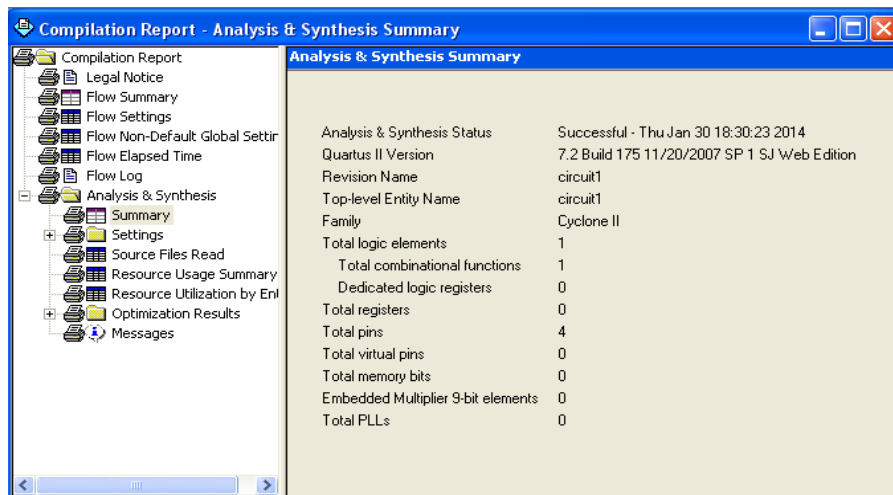


Figura 11. Raportul de compilare

4. Simularea proiectului

Pentru a putea realiza simularea funcțională trebuie să creăm un fișier cu stimuli pentru intrări. Pentru a realiza acest lucru, din meniul **File** se selectează opțiunea **New**. În fereastra care apare selectăm opțiunea **Vector Waveform File** din submeniul **Other Files**, apoi facem clic pe butonul **OK**.

File → **New** → **Vector Waveform File**

Adăugarea semnalelor pentru simulare se face prin selectarea:

Edit → **Insert** → **Insert Node or Bus**

Se tastează butonul **Node Finder** în fereastra **Insert Node or Bus** (Figura 12).

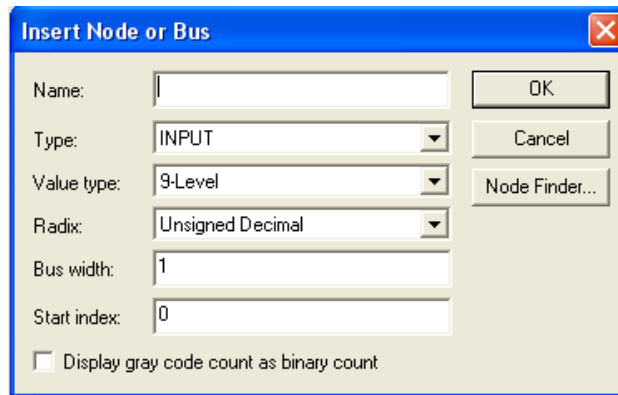


Figura 12. Fereastra pentru specificarea nodurilor și magistrelor.

Va apărea fereastra **Node Finder** (Figura 13). În meniul **Filter** se selectează **Pins: all** și se tastează butonul **List**. Pentru a transfera toți pinii circuitului, în fereastra **Selected Nodes** se tastează butonul **>>**. Apoi se tastează **OK**.

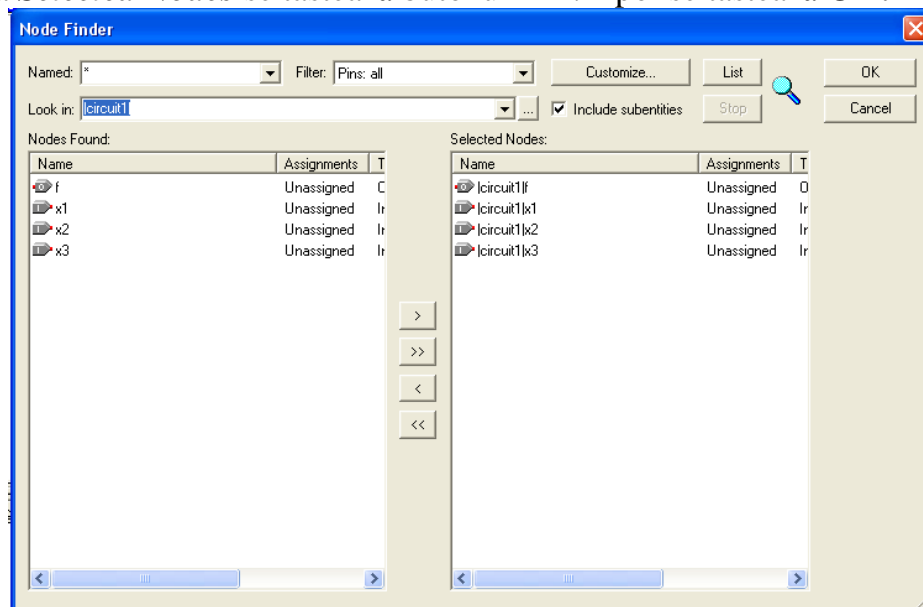


Figura 13. Selectarea nodurilor pentru simulare

În fereastra **Insert Node or Bus** se selectează **Binary** în opțiunea **Radix** (Figura 14) pentru a specifica tipul semnalului vizualizat la simulare.

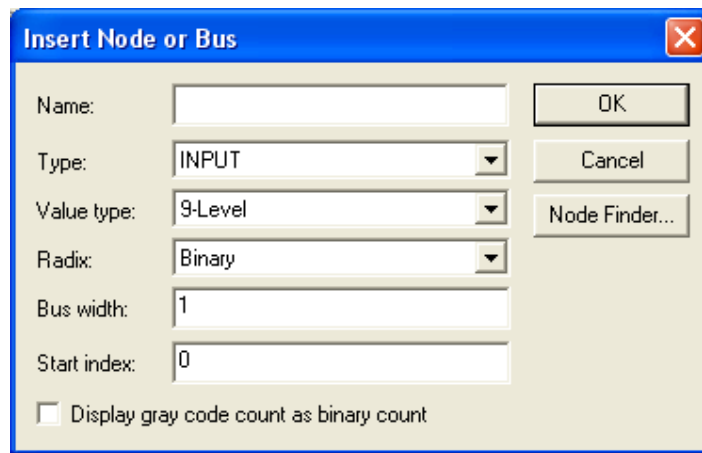


Figura 14. Specificarea tipului semnalului.

Pentru a alege timpul de simulare, se selectează **Edit** → **End Time**. Se va deschide fereastra **End Time** (Figura 15). Se alege timpul de simulare de 400 ns.

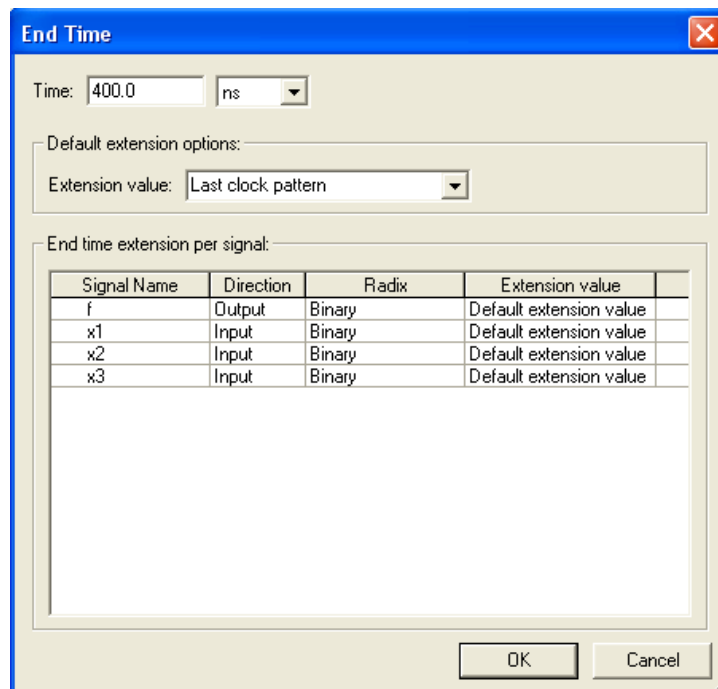


Figura 15. Alegerea timpului de simulare.

Se generează forma de undă dorită pentru fiecare intrare în parte, folosind butoanele din partea stângă a ferestrei de lucru (figura 16).

După setarea semnalelor pentru cele trei intrări, se salvează fișierul, selectând opțiunea **Save as...** din meniul **File**. Numele fișierului **.vwf** va fi același cu al schemei **.bdf**.

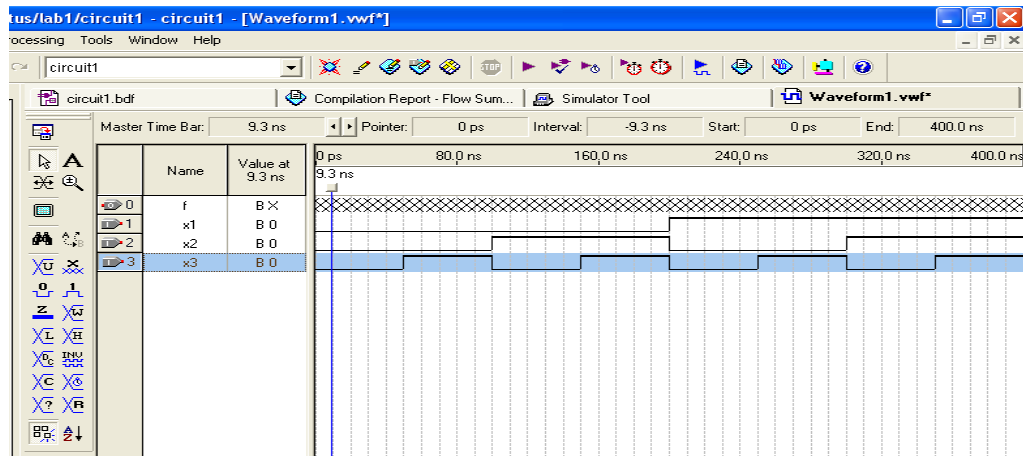



Figura 16. Fișierul cu stimuli de intrare.

Pentru simularea funcțională a schemei se va deschide simulatorul încorporat în mediul de dezvoltare Quartus II prin selectarea opțiunii **Simulator Tool** din meniul **Processing**. Se va deschide fereastra simulatorului.

La opțiunea **Simulation mode** se va selecta **Functional**, iar la opțiunea **Simulation input** se va selecta fișierul cu stimuli pentru intrări creat anterior. Apoi se va da clic pe butonul **Generate Functional Simulation Netlist**. După terminarea generării listei de conexiuni funcționale se poate porni simularea dând clic pe butonul **Start Simulation** .

După încheierea simulării se poate vedea raportul simulării dând clic pe butonul **Circuit1.vwf**. Se va deschide o nouă fereastră în care apar semnalele de intrare generate anterior și semnalul de ieșire obținut în urma simulării (figura 17). Se poate observa astfel dacă schema funcționează corect.

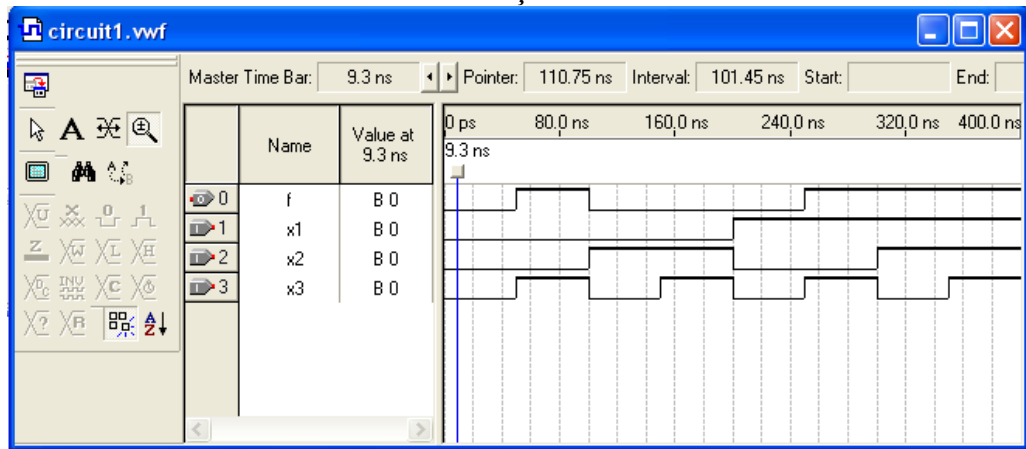



Figura 17. Raportul simulării funcționale

Pentru a efectua simularea temporală se va selecta **Assignments** → **Settings** pentru a deschide fereastra **Settings**. În fereastră se selectează **Simulator Settings** și se setează **Simulation Mode** la **Timing**.

Se efectuează, de această dată, compilarea completă și se pornește simularea, dând clic pe butonul **Start Simulation** . Din raportul simulării se poate observa că funcția **f** va reacționa la stimuli de intrare cu o reținere de

aproximativ 6 ns din cauza întârzierilor din circuitul real. Tot din cauza întârzierilor din circuitul real se pot observa și anumite hazarduri de semnal.

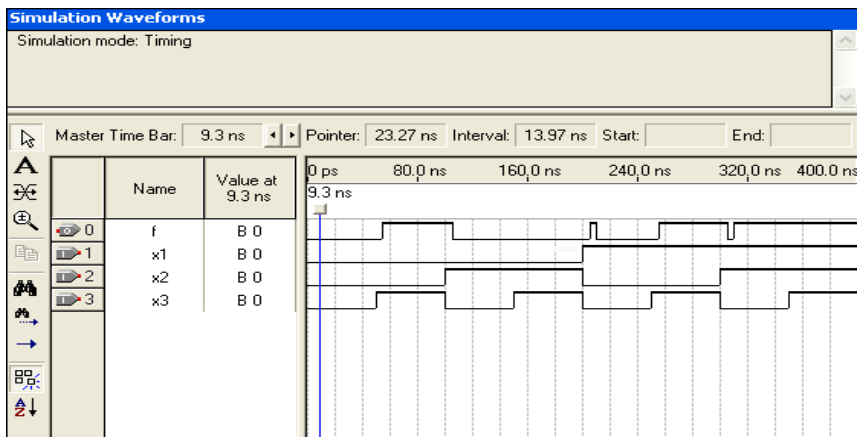


Figura 18. Raportul simulării temporale

5. Asignarea pinilor

Pentru a efectua implementarea practică a schemei în circuitul FPGA de pe placa de dezvoltare DE0 este necesar ca pinii din schemă să fie atribuiți unor pini fizici ai circuitului FPGA Cyclone III EP3C16F484C6. Acest lucru se realizează prin selectarea opțiunii **Pins** din meniul **Assignments**. Se va deschide o fereastră în care vom putea selecta ce pini fizici vor corespunde intrărilor și ieșirilor schemei.

Lista pinilor de pe circuitul FPGA Cyclone III EP3C16F484C6 conectați la întrerupătoarele plăcii de dezvoltare DE0 este prezentată în tabelul 1. În tabelul 2 este dată lista pinilor conectați la cele trei butoane de pe placă. Tabelul 3 conține lista pinilor conectați la led-uri.

Tabelul 1. Asignarea pinilor pentru întrerupătoare

Numele semnalului	Nr. Pinului de pe FPGA	Descriere
SW[0]	PIN_J6	Slide Switch[0]
SW[1]	PIN_H5	Slide Switch[1]
SW[2]	PIN_H6	Slide Switch[2]
SW[3]	PIN_G4	Slide Switch[3]
SW[4]	PIN_G5	Slide Switch[4]
SW[5]	PIN_J7	Slide Switch[5]
SW[6]	PIN_H7	Slide Switch[6]
SW[7]	PIN_E3	Slide Switch[7]
SW[8]	PIN_E4	Slide Switch[8]
SW[9]	PIN_D2	Slide Switch[9]

Tabelul 2. Asignarea pinilor pentru butoane

Numele semnalului	Nr. Pinului de pe FPGA	Descriere
BUTTON [0]	PIN_H2	Pushbutton[0]
BUTTON [1]	PIN_G3	Pushbutton[1]
BUTTON [2]	PIN_F1	Pushbutton[2]

Tabelul 3. Asignarea pinilor pentru led-uri

Numele semnalului	Nr. Pinului de pe FPGA	Descriere
LEDG[0]	PIN_J1	LED Green[0]
LEDG[1]	PIN_J2	LED Green[1]
LEDG[2]	PIN_J3	LED Green[2]
LEDG[3]	PIN_H1	LED Green[3]
LEDG[4]	PIN_F2	LED Green[4]
LEDG[5]	PIN_E1	LED Green[5]
LEDG[6]	PIN_C1	LED Green[6]
LEDG[7]	PIN_C2	LED Green[7]
LEDG[8]	PIN_B2	LED Green[8]
LEDG[9]	IN_B1	LED Green[9]

Pentru a putea observa funcționarea schemei după încărcarea ei în circuitul FPGA, se vor conecta cele trei intrări ale circuitului la trei întrerupătoare, iar ieșirea acestora se va conecta la unul din LED-urile programabile de către utilizator.

După încheierea etapei de asignare a pinilor (figura 19), pentru a putea încărca și rula aplicația pe circuitul FPGA, trebuie realizată compilarea proiectului în vederea obținerii fișierului *.sof* (SRAM Object File). Pentru aceasta se va selecta opțiunea **Start Compilation** din meniul **Processing**.

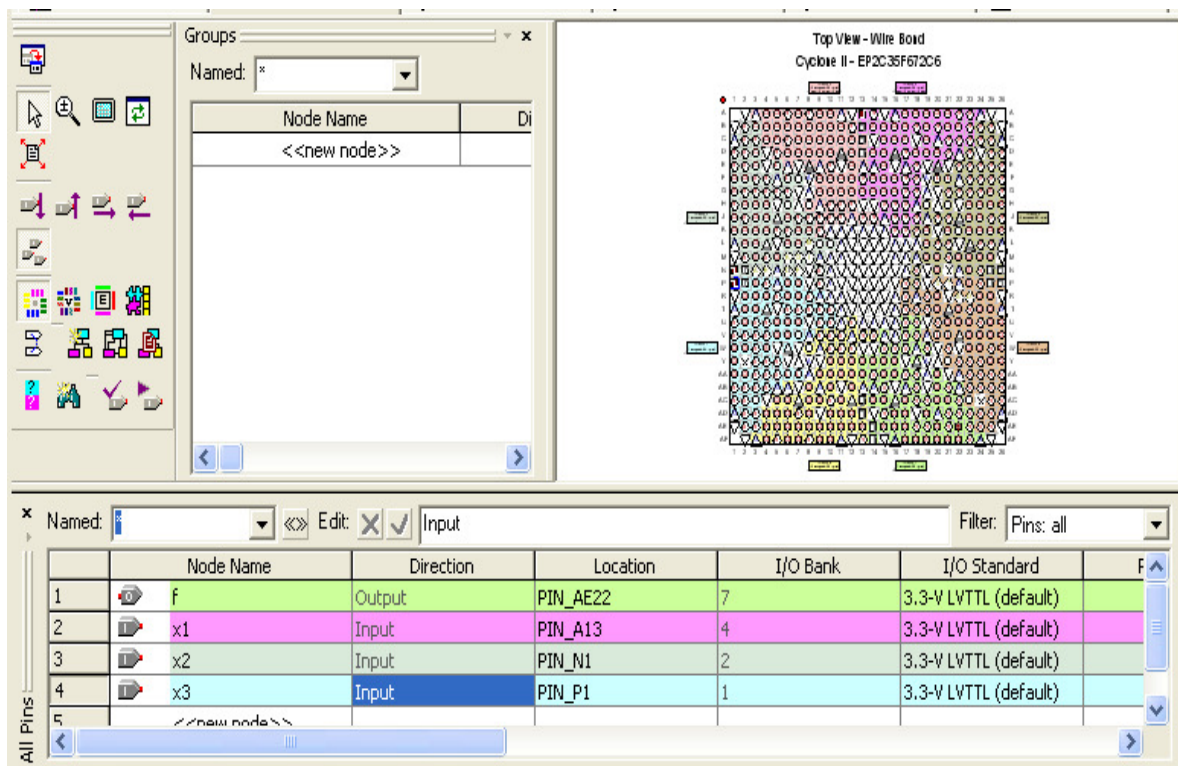


Figura 19. Asignarea pinilor

6. Implementarea circuitului pe placa de dezvoltare DE2

După terminarea compilării și obținerea fișierului *.sof*, se selectează din meniul *Tools* opțiunea *Programmer*. În fereastra care se deschide se apasă butonul *Hardware Setup* pentru a se configura interfața *USB-Blaster* necesară comunicării cu kitul de dezvoltare. În acest moment, placa trebuie să fie conectată pe USB și alimentată, butonul *RUN/PROG* pentru modul de lucru *JTAG/AS* trebuie să fie în poziția *RUN*. Se selectează interfața *USB-Blaster* și se închide fereastra apăsând butonul *OK*. În fereastra programatorului se selectează modul de conectare ca fiind modul *JTAG*, se încarcă dacă este necesar fișierul *.sof* obținut la compilare și se dă clic pe butonul *Start*. După încărcarea totală a programului se poate testa funcționarea schemei implementate pe placă.

Desfășurarea lucrării

1. Se va crea proiectul pentru circuitul logic, conform variantei din tabelul 2. Se va utiliza biblioteca *primitives*. Se va compila și se va simula circuitul realizat.
2. Se va crea un alt proiect pentru circuitul logic. Pentru sinteza circuitului se va folosi biblioteca *megafunction*. Circuitul inițial va fi suplimentat cu o intrare de selecție, care va transmite la ieșire funcția *f* în formă directă sau inversă. Se va compila și se va simula circuitul realizat.

Tabelul 2.

Nr. var.	Funcția logică	Nr. var.	Funcția logică
1.	$F = \sum(0,1,2,3,7,8,9,15)$	13.	$F = \sum(0,1,5,9,11,13)$
2.	$F = \sum(2,6,7,8,9,10,14,15)$	14.	$F = \sum(2,3,6,7,9,10,13)$
3.	$F = \sum(4,6,7,9,11,12,14,15)$	15.	$F = \sum(3,5,7,13,14,15)$
4.	$F = \sum(3,7,8,10,11,14,15)$	16.	$F = \sum(1,8,9,11,12,13)$
5.	$F = \sum(0,1,2,3,7,8,9,15)$	17.	$F = \sum(0,2,4,5,6,10)$
6.	$F = \sum(3,4,5,6,7,8,9,11,15)$	18.	$F = \sum(1,8,9,11,13,15)$
7.	$F = \sum(0,1,5,7,8,9,12,13)$	19.	$F = \sum(6,7,11,12,14,15)$
8.	$F = \sum(2,8,9,10,11,13)$	20.	$F = \sum(3,5,7,9,13,15)$
9.	$F = \sum(2,4,6,8,10,12,14)$	21.	$F = \sum(0,1,2,8,9,12)$
10.	$F = \sum(1,2,5,8,9,10,11,13)$	22.	$F = \sum(1,4,5,12,13,14,15)$
11.	$F = \sum(0,2,4,5,8,9,10,11)$	23.	$F = \sum(2,4,6,10,12,14,15)$
12.	$F = \sum(4,5,6,7,8,10,11,12)$	24.	$F = \sum(1,3,5,9,12,13,14)$