

LABORATOR 2 – Descrierea sumatorului pe un bit cu porți logice

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 bazată pe modulul FPGA de tip Spartan 3 cu codul XC3S400. Placa de dezvoltare este dotată cu comutatoare, taste, LED-uri și afișoare 7 segmente cu LED-uri. Documentația aferentă plăcii de dezvoltare S3 se găsește în fișierul [S3BOARD-rm.pdf](http://www.cs.ucv.ro/~lemen/Downloads/S3BOARD-rm.pdf), disponibil la <http://www.cs.ucv.ro/~lemen/Downloads/S3BOARD-rm.pdf#>.

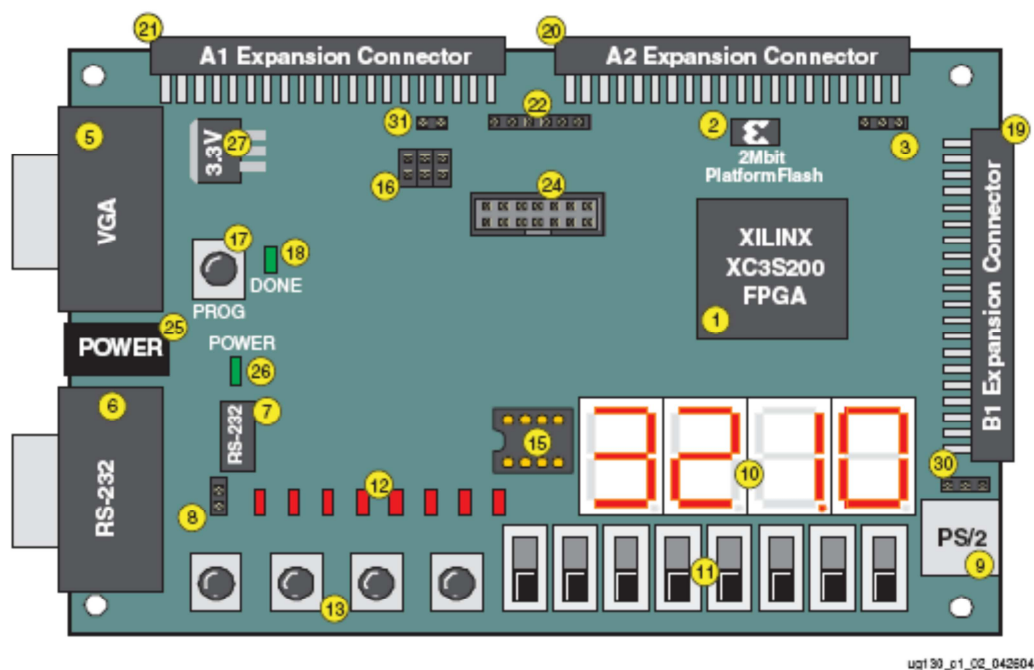


figura 1

Scopul lucrării constă în implementarea unui sumator elementar descris prin intermediul schemei. De asemenea se vor detalia etapele ce trebuie parcurse pentru a obține fișierul de configurare al FPGA-ului.

Desfășurarea lucrării

Pasul 1: Crearea proiectului.



Se lansează în execuție ISE prin intermediul icoanei și se creează un nou proiect cu următoarele specificații:

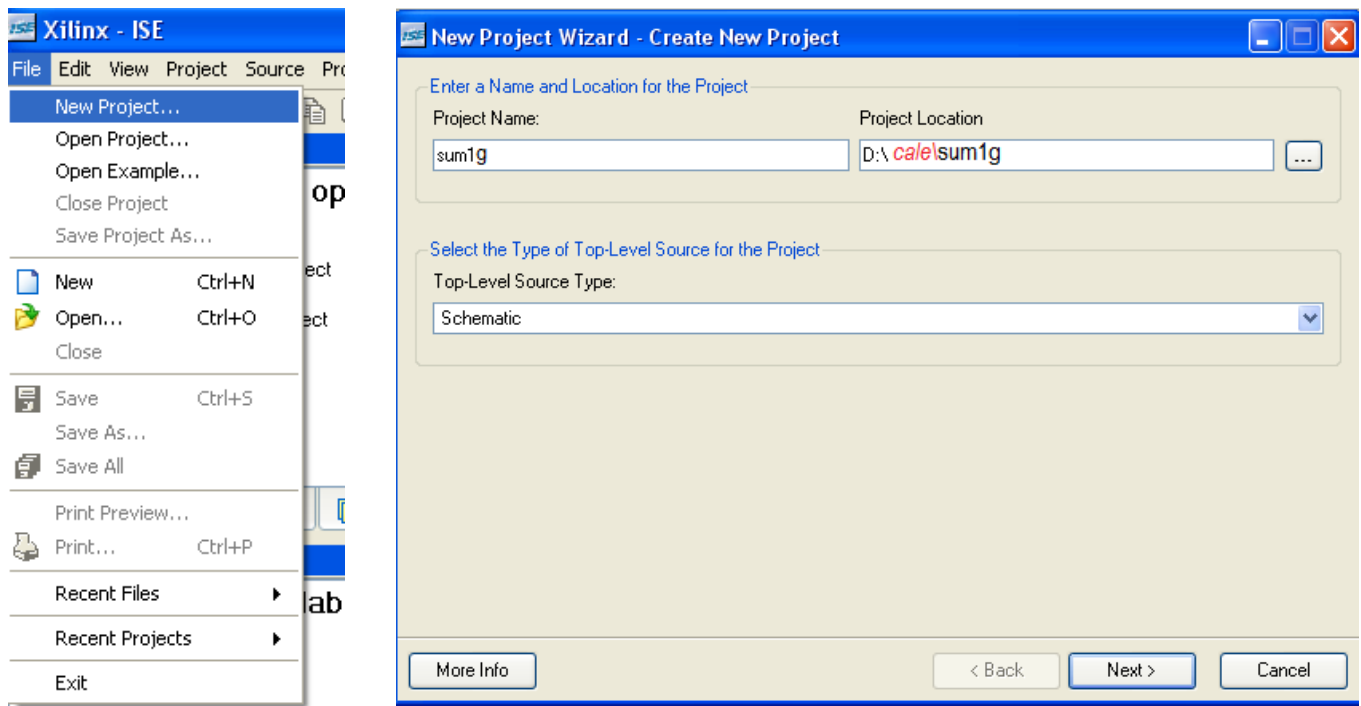


figura 2

ISE creează câte un folder pentru fiecare proiect. Folderul va avea același nume ca și proiectul. Din acest motiv, mai întâi se specifică folderul părinte (adică folderul ce va conține folderul proiectului), iar apoi numele proiectului, **sum1g**. În momentul în care completați numele proiectului, ISE va actualiza automat calea, aceasta devenind: **d:\cale\sum1g**. *Cale* este calea folderului de lucru al fiecărei echipe (D:\oc....).

Atenție: pentru orice proiect ISE, nu folosiți foldere al căror nume conține spații. Astfel de nume nu trebuie folosite nici pentru folderele care alcătuiesc calea, nici pentru numele folderului proiectului nici pentru numele fișierelor din proiect. Datorită unui bug în ISE 9.2 anumite programe funcționează defectuos dacă cerința de mai sus nu este respectată. Un exemplu de folder foarte des utilizat al cărui nume conține spații este „Documents and Settings”.

Ca și în cazul proiectelor software, un proiect ISE 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.

Modulul din vârful ierarhiei poate fi de tip schematic sau HDL (HDL=Hardware Description Language = Limbaj de descriere hardware). Pe parcursul laboratoarelor la disciplina „Organizarea calculatoarelor” se vor folosi atât descrierile de tip schematic cât și cele de tip VHDL. În acest prim laborator **Top-Level Source Type** se alege **Schematic**. Mai mult despre structura ierarhică a proiectelor hardware, în laboratoarele următoare.

După ce ați particularizat și completat toate informațiile din figura 2 apăsați butonul **Next**. Va apare fereastra din figura 3.

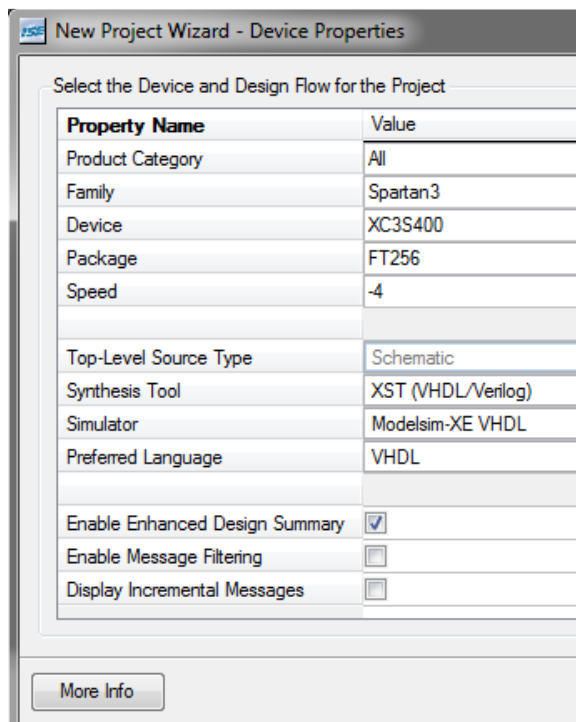


figura 3

Modulul în jurul căruia este construită placa de dezvoltare S3 aparține familiei **Spartan3**, are codul **XC3S400** (400 înseamnă ca modulul conține echivalentul a 400000 de porți logice), împachetarea este de tip **FT** iar numărul de pini este **256**.

Celelalte câmpuri se setează la valorile din figura 3. **Este foarte important ca pentru acest proiect cât și pentru toate proiectele pe care le veți crea ulterior să faceți setările exact ca în figura 3.** În caz contrar implementarea pe placa Spartan 3 nu va fi posibilă.

În continuare 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 situația din figura 4.

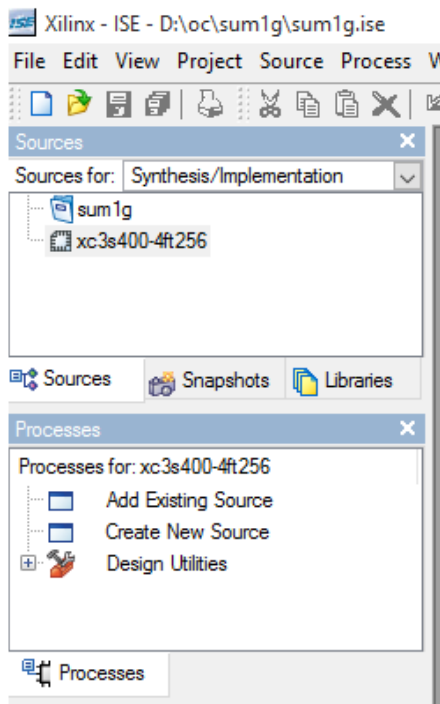


figura 4

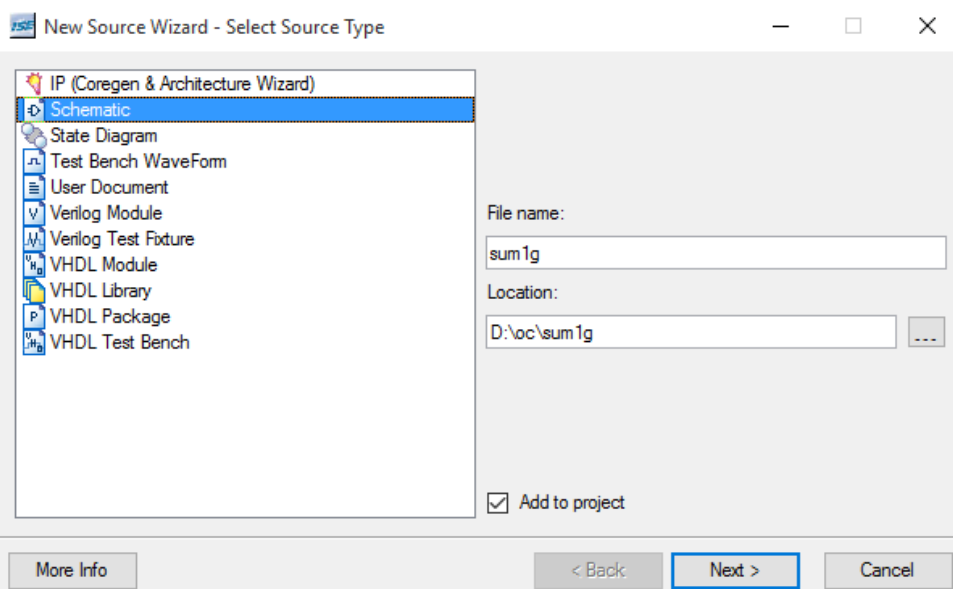


figura 5

Pasul 2: Organizarea spațiului de lucru.

În continuare se adaugă fișierul care va conține schema sumatorului. Se va face clic dreapta pe numele FPGA-ului (xc3s400..), și din meniul contextual apărut se selectează **New Source**. În fereastră New Source, se selectează tipul **Schematic** iar apoi se completează numele fișierului - **sum1g** -, ca în figura 5. Apoi se apasă **Next**, apare o fereastră de informare și în cadrul acesteia se apasă **Finish**. Ca efect al apăsării lui **Finish**, **sum1g** este adăugat la proiect și este lansat în execuție programul care permite desenarea schemei. Acesta se numește (în cadrul ISE) ECS - Engineering Capture System. În general, pentru programele de desenare a schemelor logice, în limba engleză, se folosește termenul de „Schematic Capture”. Veți obține situația din figura 6.

Modul de desenare în ISE este foarte asemănător cu desenarea în OrCAD.

Înainte de a trece la desenarea sumatorului elementar se va reorganiza spațiul de lucru. Mai întâi se vor închide barele de scule folosite rar. Pentru aceasta poziționați cursorul oriunde în zona barelor de scule și faceți clic dreapta. Veți obține o fereastră în care sunt afișate barele de scule active. Această fereastră este încadrată de un dreptunghi roșu în figura 6. Activați numai barele de scule active din dreptunghiul marcat cu roșu în figura 6.

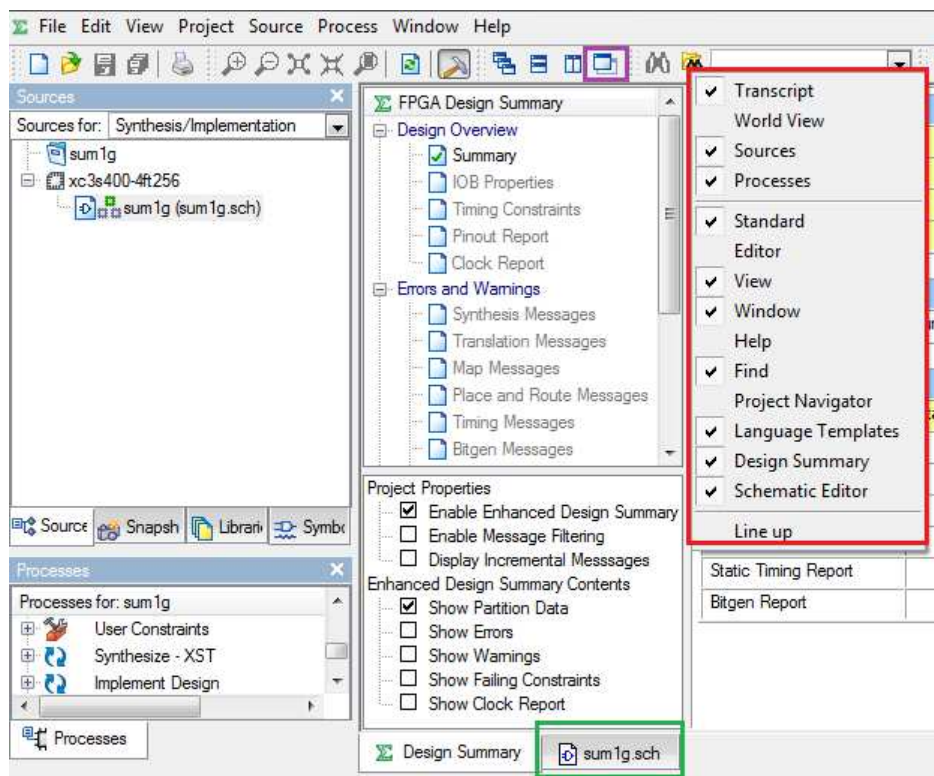


figura 6

Organizarea inițială a interfeței ISE este foarte proastă în cazul proiectelor schematic. Pentru a separa editorul grafic de restul interfeței procedați astfel:

1. Afișați schema sum1g.sch în fereastra principală a ISE. Pentru aceasta faceți clic pe câmpul tab **sum1g.sch** încadrat într-un dreptunghi verde în figura 6 .
2. Din meniul **Window** alegeți opțiunea **Float**, sau apăsați icoana din dreptunghiul violet din figura 6.

După ce se execută cei doi pași, interfața ISE se divide în **două ferestre**: o fereastră principală în care se gestionează proiectul și o fereastră în care se va desena sumatorul elementar.

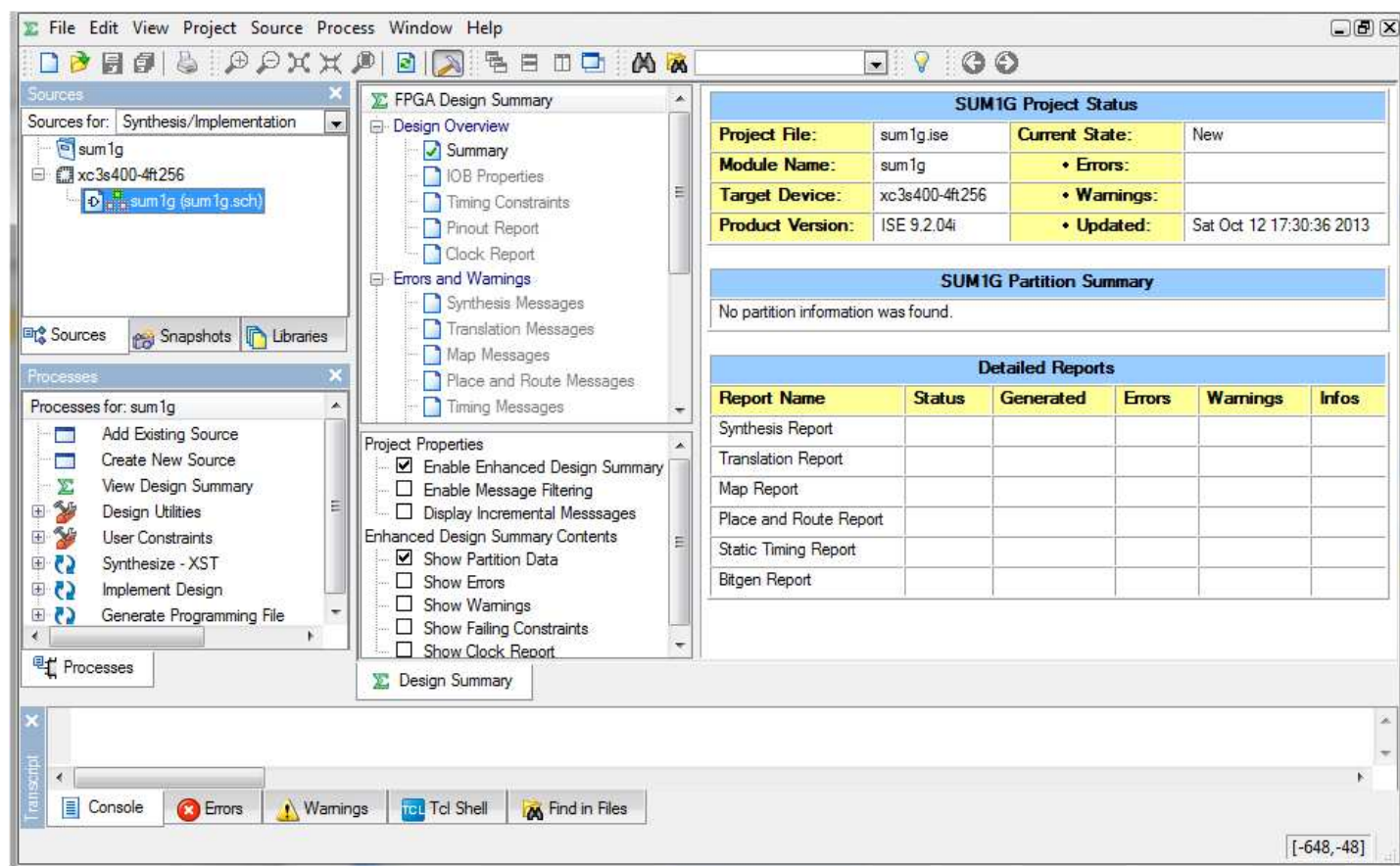


figura 7 – Fereastra proiectului

Fereastra proiectului este prezentată în figura 7 iar fereastra de desenare în figura 8.

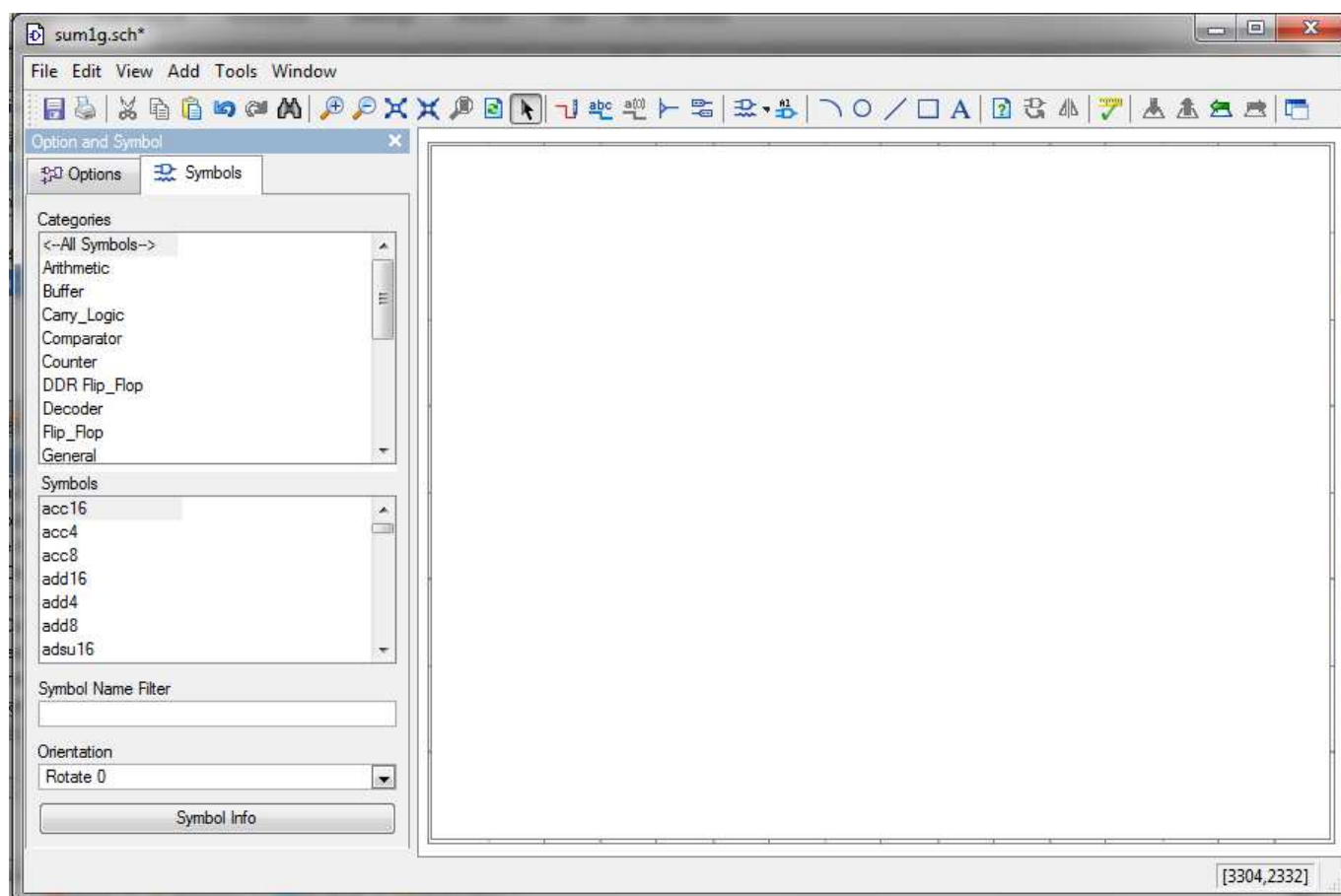


figura 8 - Editorul grafic

În restul acestui laborator, cât și în laboratoarele următoare se recomandă folosirea variantei cu două ferestre.

Pasul 3: Crearea schemei sumatorului elementar.

În laboratorul precedent s-a prezentat structura simplificată a unui FPGA precizându-se ca acesta este alcătuit din LUT-uri și bistabile. Deși LUT-urile sunt cărămizile din care este alcătuit un FPGA, proiectarea cu LUT-uri conduce la scheme neclare: toate funcțiile de 4 variabile vor avea aceeași schemă, și anume un LUT4. Pentru a înțelege ce funcție se implementează, trebuie construită reconstituită tabela de adevăr memorată în LUT.

Pentru ca schemele să fie mai clare ar fi mult mai bine să folosim formele logice. În cazul sumatorului elementar o variantă des întâlnită de forme este:

$$S = A \oplus B \oplus Ci$$

$$Co = AB + ACi + BCi$$

Implementarea cu porți logice a acestor forme necesită un XOR3, 3 porți AND2 și o poartă OR3. Dar FPGA-ul conține doar LUT-uri și bistabile, nu și porți logice. Pentru ca specificarea funcțiilor cu porți logice să fie posibilă **este nevoie de o operație de transformare**: pornind de la schema cu porți trebuie construită o schemă echivalentă cu LUT-uri. Această operație se numește sinteză și se aseamănă conceptual cu compilarea unui program C. Așa cum compilatorul transformă programul C în program cod mașină, așa sintetizatorul transformă o schema cu porți într-o schemă cu LUT-uri. Sinteza se face și dacă specificarea hardware-ului se face folosind un limbaj de descriere hardware (VHDL sau Verilog).

Pentru desenarea sumatorului elementar se vor folosi butoanele din figura 9:

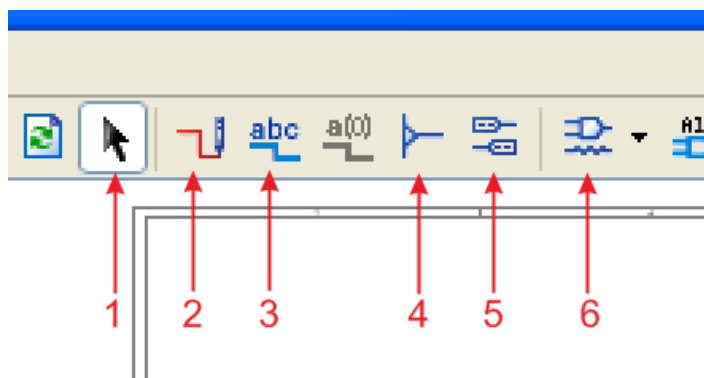


figura 9

Rolul acestor butoane este următorul:

1. Select. Este folosit pentru selectarea obiectelor în vederea ștergerii sau copierii. Acest buton se activează automat atunci când se apasă ESC.
2. Adaugă fir sau magistrală.
3. Adaugă etichetă.
4. Adaugă conexiune fir-magistrală.
5. Adaugă IO marker.
6. Adaugă simbol.

Utilitatea butoanelor va fi detaliată pe măsură ce va fi nevoie de obiectele generate de acestea.

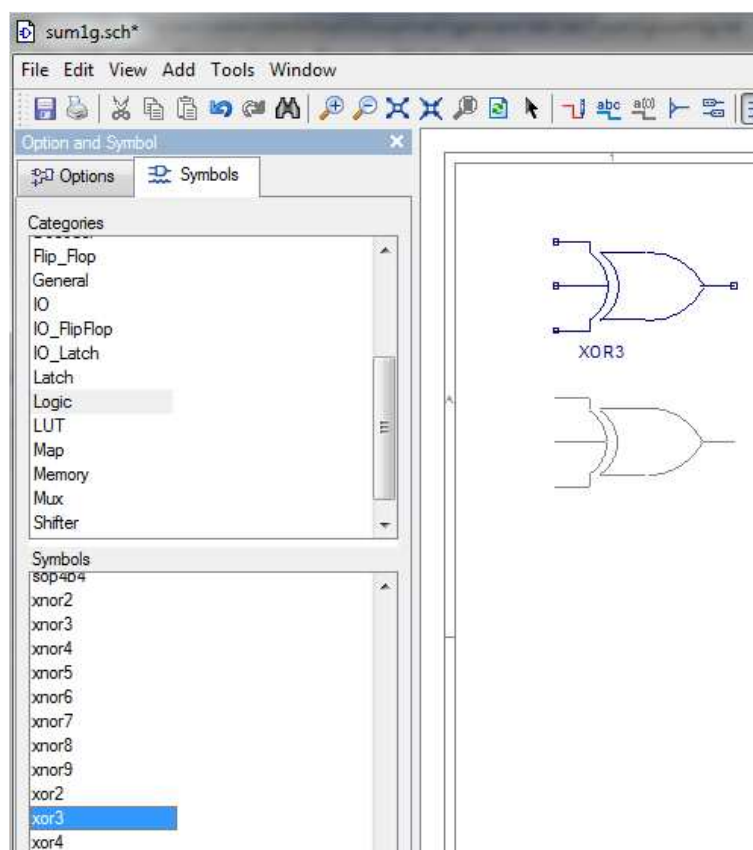


figura 10

În continuare se adaugă celelalte porți.

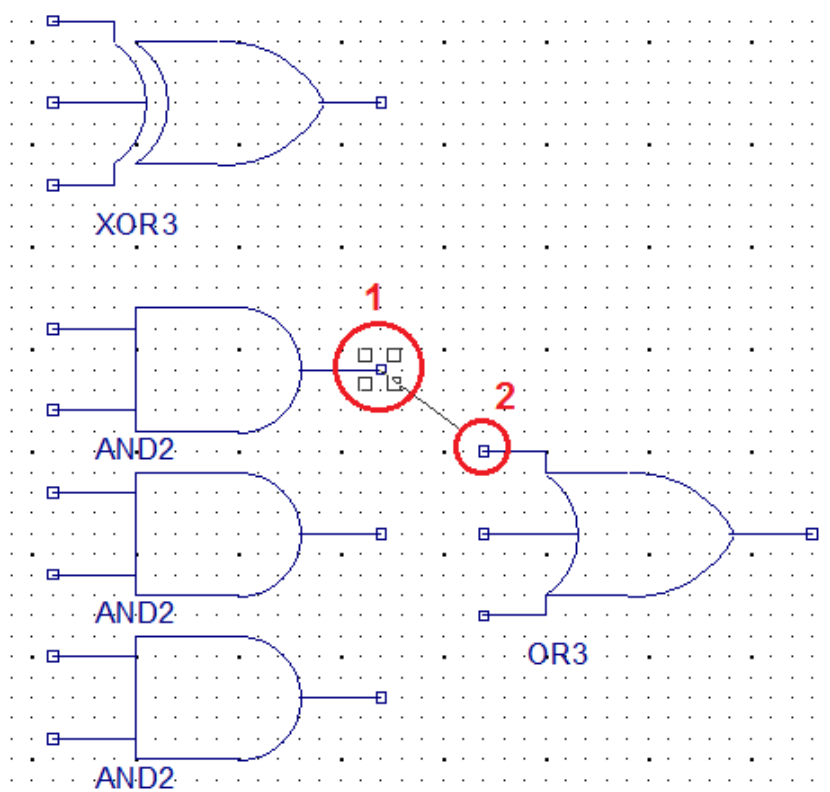


figura 11

Apoi se adaugă conexiunile. Pentru a adăuga conexiunile (firele) dintre porți, se apasă **butonul 2 - Adaugă conexiune** din figura 9. Apoi în subfereastra **Option and Symbol** se selectează câmpul tab **Options**. Sub câmpul Tab **Options** apar două opțiuni: **Autoruter** și **Manual** (figura 12).

Dacă pentru adăugarea de noi conexiuni se folosește opțiunea implicită **Autoruter**, se procedează astfel:

Folosind **butonul 6 - „Adaugă simbol”**, se adaugă porțile XOR3, AND2, și OR3, din care este alcătuit sumatorul elementar.

Apăsarea butonului „Adaugă simbol” face ca în subfereastra **Option and Symbol** să se selecteze automat câmpul tab **Symbols**.

Pentru a plasa poarta XOR3 selectați **Logic** în subfereastra **Option and Symbol –Symbols** iar subfereastra **Symbols** amplasata dedesubt, selectați **XOR3**.

Atunci când se selectează un simbol, respectivul simbol funcționează ca o șampilă atașată de cursor. Plasarea unui obiect se face printr-un singur clic cu butonul stânga al mouse-ului. În acest moment se „ștampilează”. Efectul va fi apariția unui simbol fix – amprenta ștampilei. Ștampila propriu-zisă rămâne atașată de cursor și poate fi folosită în continuare pentru a genera noi simboluri fixe. În figura 10 se observă o poartă XOR3 amplasată și „ștampila” atașată de cursor.

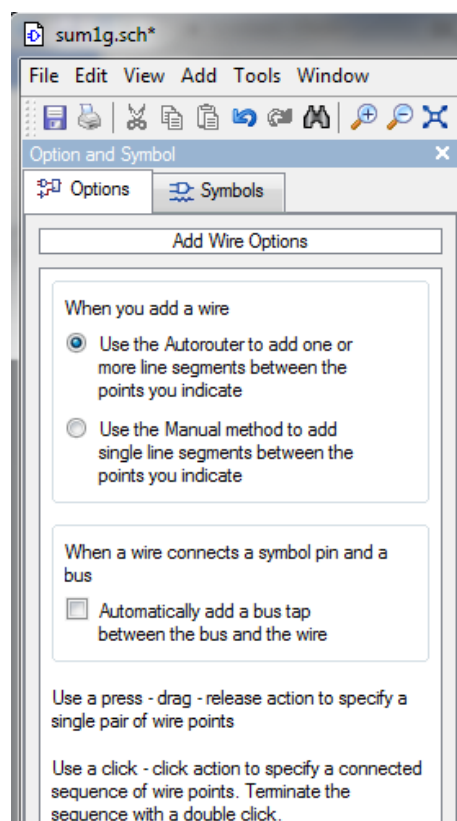


figura 12

1. Se mișcă cursorul deasupra unuia din capetele noii conexiuni. De exemplu, capătul unei conexiuni poate fi terminalul unei porți. Două astfel de terminale sunt marcate cu cerc roșu în figura 11. În momentul în care suntem cu cursorul chiar deasupra terminalului, vor apărea în jurul acestuia 4 pătrate mici (figura 11, cercul roșu marcat cu „1”)
2. Se face clic pe terminalul marcat de cele 4 pătrățele.
3. Se deplasează cursorul până la celălalt capăt al conexiunii. Pe măsură ce cursorul este deplasat, între acesta și punctul de pornire apare o linie. Această linie poate să intersecteze orice alt element grafic din desen, fără ca acest fapt să constituie o eroare.
4. Când cursorul ajunge deasupra terminalului destinație - de exemplu terminalul identificat prin cercul roșu 2 din figura 11- apar cele 4 pătrate mici. Dacă în acest moment se face clic, cele două terminale vor fi unite prin segmente perpendiculare. **Observație: uneori cele patru pătrățele nu apar dar conexiunea se poate face urmând procedura anterioară.**

Conform procedurii descrise anterior, se unesc ieșirile porților AND2 cu intrările porții OR3, obținându-se desenul din figura 13.

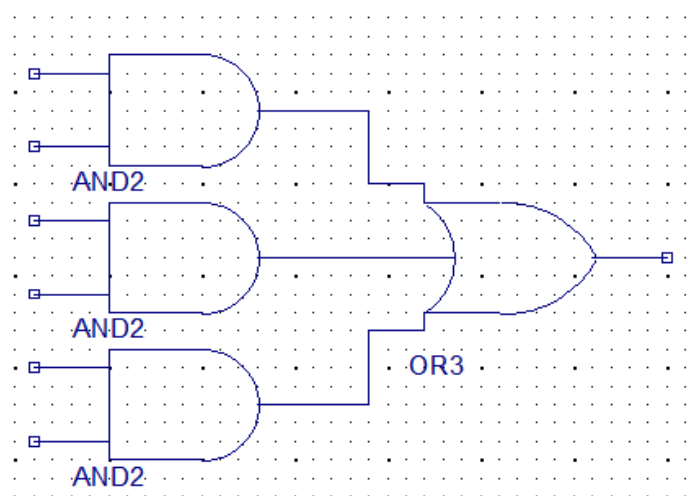


figura 13

se dorește creată. În această situație va apărea mesajul din figura 14.

Observații:

1. Un terminal folosit pentru o conexiune, poate fi folosit pentru o nouă conexiune. Chiar dacă pătrățelele de identificare a terminalului dispăreau după efectuarea primei conexiuni, în momentul în care cursorul ajunge deasupra unui astfel de terminal, cele 4 pătrățele de identificare vor reapărea. Mai mult, orice punct al unei conexiuni existente poate fi folosit ca punct terminal al unei noi conexiuni.
2. În momentul în care există prea multe obiecte pe suprafața de desenare, este posibil ca autorouterul să nu găsească o soluție de desenare pentru conexiunea ce

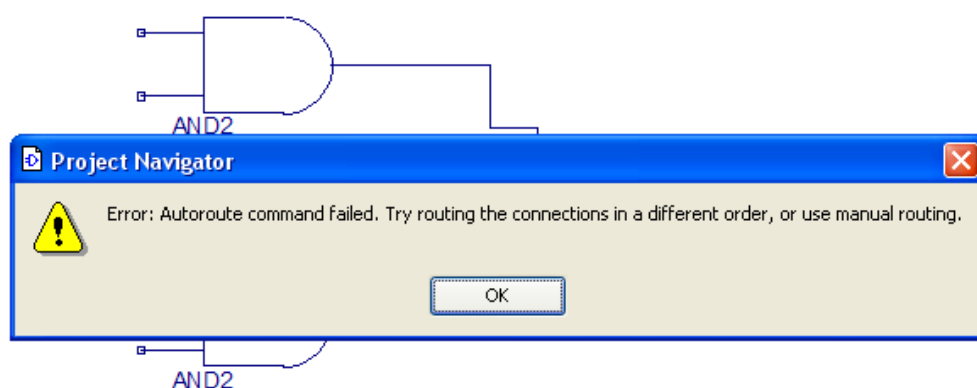


figura 14

Acest mesaj poate să apară și dacă s-a specificat greșit un capăt al conexiunii. Dacă se face clic în interiorul unei porți în loc de terminalul corespunzător, acest mesaj poate să apară. Dacă se reface conexiunea, specificându-se corect ambele capete, mesajul va dispărea.

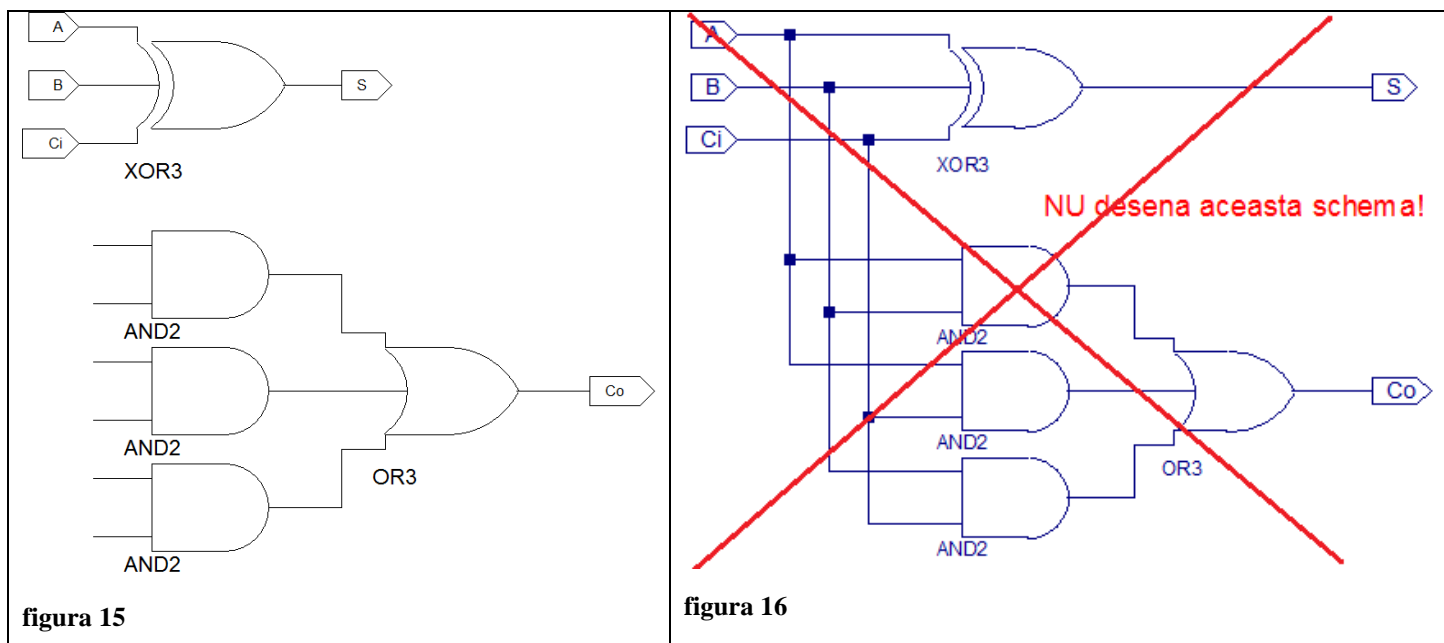
În cazul în care mesajul apare ca urmare a unei densități prea mari de componente, se alege opțiunea de desenare prin metoda manuală (opțiunea **Manual** în figura 12). În această variantă, conexiunea este desenată segment cu segment, după următoarele reguli:

- Se pornește la fel ca în cazul „Autorouter”, adică se mișcă cursorul deasupra unuia din capetele noii conexiuni. În momentul în care suntem cu cursorul chiar deasupra terminalului, vor apărea în jurul acestuia cele 4 pătrate mici
- Pentru a genera un nou segment se face clic stânga o singură dată.
- Pentru a termina conexiunea se face clic pe terminalul unui simbol sau pe o conexiune deja desenată.
- Conexiunea se poate termina în orice moment dacă se face dublu clic stânga. Dacă o conexiune nu se termină pe terminalul unui simbol, ci a fost terminată cu dublu clic, atunci respectiva conexiune se numește **terminată „în aer”** și se evidențiază printr-un pătrățel roșu la capul terminat „în aer”.

În continuare se atașează la terminalele porții **marcheri I/O**. Marcherii I/O reprezintă conexiunile unui bloc logic cu blocul ierarhic superior. În cazul blocului din vârful ierarhiei (Top level) marcherii I/O reprezintă conexiunile cu exteriorul, conexiuni făcute prin intermediul pinilor modulului FPGA.

Pentru a plasa un marcher I/O, se apasă butonul **Add I/O Marker** (marcaj cu 5 în figura 9) și apoi se deplasează cursorul de care este atașat marcherul de intrare în spațiul de desenare până când se atinge de terminalul cel mai de sus al porții XOR3. În acest moment se face clic și marcherul este plasat pe planșă. În același mod se plasează încă doi marcheri pe ceilalți doi pini ai porții XOR3, un marcher pe ieșirea porții XOR3 și încă unul pe ieșirea porții OR3.

În momentul în care a fost creat fiecare marcher primește un nume de tip $XLXN_n$. n este al câtelea marcher IO este marcherul adăugat și diferă de la calculator la calculator. Deoarece aceste nume nu sunt absolut de loc sugestive ele vor fi schimbate în **A, B, Ci, S** și **Co**. Pentru a schimba numele unui marcher **se face dublu clic** pe acesta și se schimbă numele în meniul contextual care apare. După schimbare se obține desenul din figura 15.



În continuare ar trebui adăugate conexiunile de la marcherii IO la intrările porților AND2. Dacă s-ar face acest lucru **s-ar obține** situația din figura 16. **Nu adăugați conexiunile din figura 16. Se va folosi altă metodă!**

În cazul unui număr mare de conexiuni, desenarea schemelor numai cu fire este dezavantajoasă deoarece funcționalitatea este „ascunsă” de numărul mare de conexiuni. Pentru a reduce acest număr, se folosește **conectarea prin nume**.

Într-o schemă, indiferent de programul de desenare, orice conexiune este identificată printr-un identificator sau etichetă. În momentul desenării conexiunii programul atribuie automat un identificator respectivei conexiuni. Acest identificator (etichetă) este „invizibil”. Dacă se face dublu clic pe o conexiune, se poate afla identificatorul asociat acesteia.

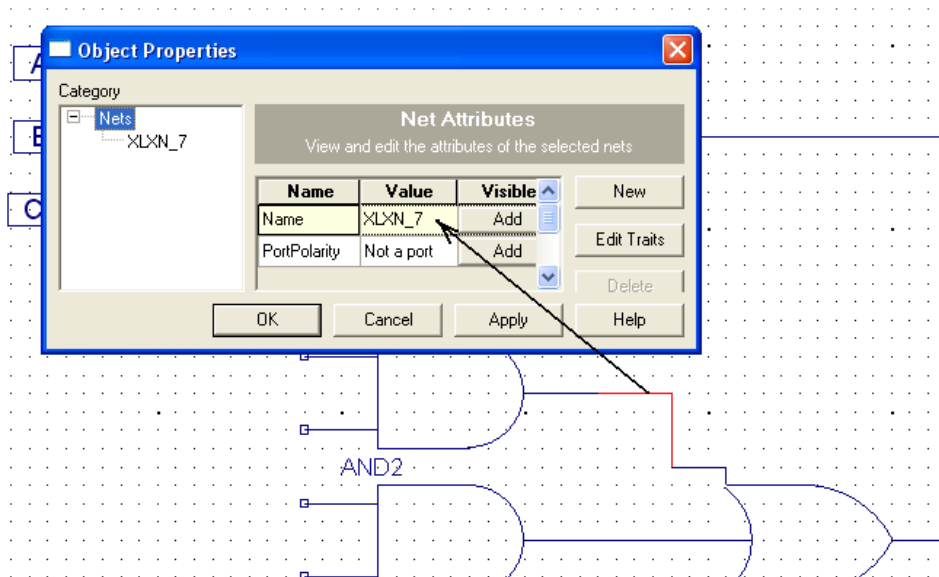


figura 17

De exemplu, dacă se face dublu clic stânga pe conexiunea marcată cu roșu în figura 17, ISE va afișa fereastra „Net Attributes”. De remarcat că în loc de conexiune, ISE folosește „net” sau „wire”. În funcție de ordinea de desenare se poate obține o etichetă diferită de XLNX_7.

Conexiunile între marcherii IO și alte terminale primesc numele marcherului IO. Verificați acest fapt!

Conform procedurii detaliate la pagina 9, adăugați conexiuni **terminated in air** pentru fiecare intrare de poartă AND2. Trebuie să obțineți configurația din figura 18. După ce adăugați prima conexiune, pentru următoarele puteți folosi „Copy and Paste”.

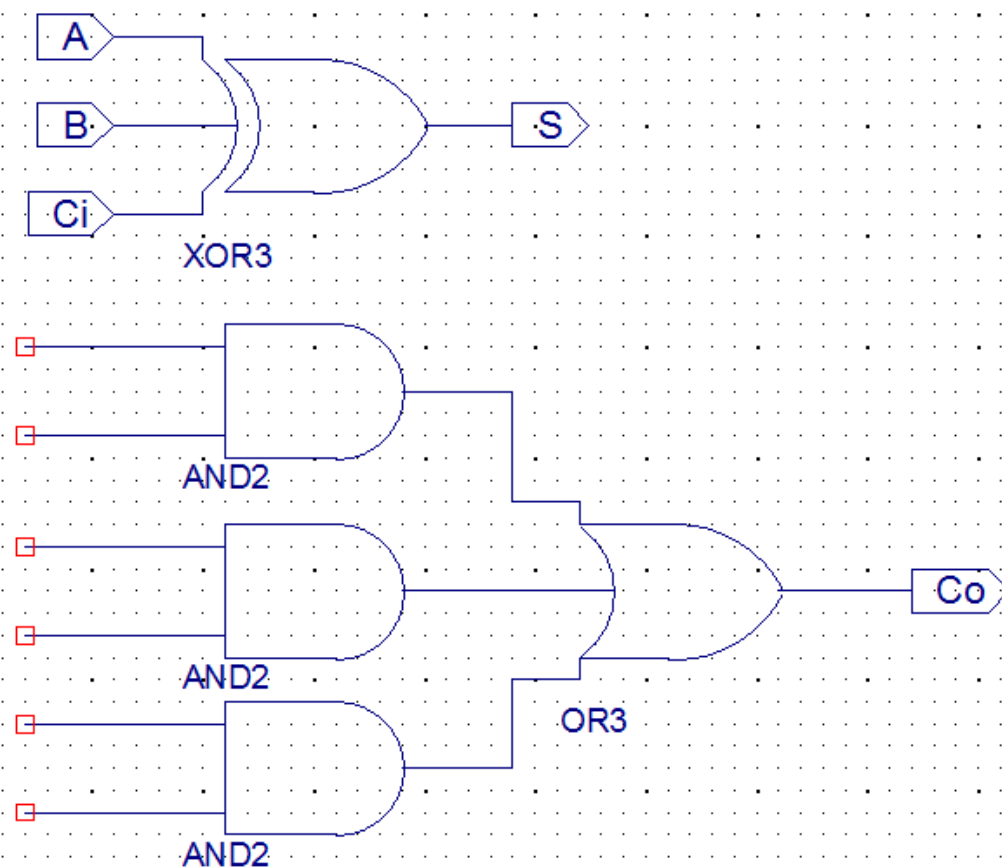


figura 18

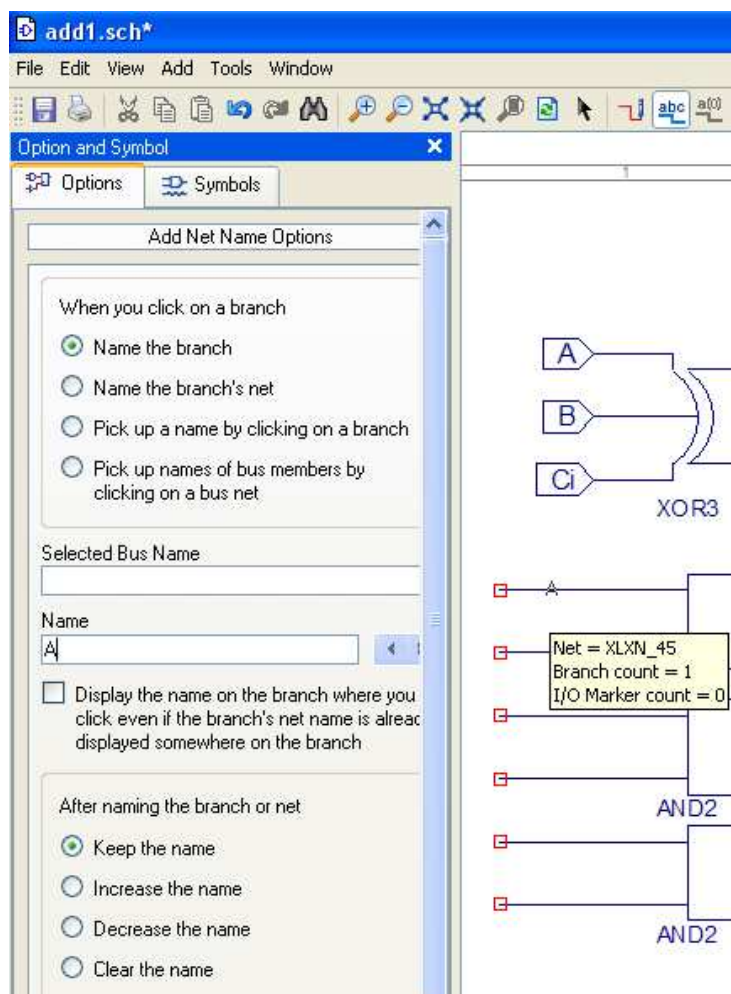


figura 19

În continuare, trebuie modificați identificatorii conexiunile aferente porților AND2 pentru a realiza conexiunile făcute tradițional cu „fire”. Reamintim că în momentul desenării orice conexiune primește un identificator. **Verificați că identificatorii conexiunilor terminate în aer, pe care tocmai le-ați desenat, sunt distincți!**

Există mai multe modalități de a edita numele unui identificator de conexiune. În continuare se va prezenta una dintre acestea. **De acum înainte veți folosi NUMAI această procedură:**

1. Se activează modul de lucru **Add net name** prin apăsarea butonului 3 din figura 9.

2. În câmpul „Name” se introduce numele identificatorului. Vom începe cu „A”.

3. Se mișcă cursorul pe suprafața de desenare din dreapta. Se observă că identificatorul „A” este acum atașat de cursor.

4. Se plasează cursorul deasupra primei conexiuni terminate în aer, ca în figura 19, iar apoi se face clic stânga, **Atenție:** pentru o plasare sigură, în momentul în care faceți clic stânga, punctul de intersecție al segmentelor ce formează cursorul

trebuie să fie chiar pe linia care va primi identificatorul.

5. La fel ca la plasarea simbolurilor, identificatorul acționează ca o „ștampilă”, astfel încât acesta poate fi plasat de mai multe ori, până când se apasă „ESC” sau până când se introduce alt identificator în câmpul „Name”.

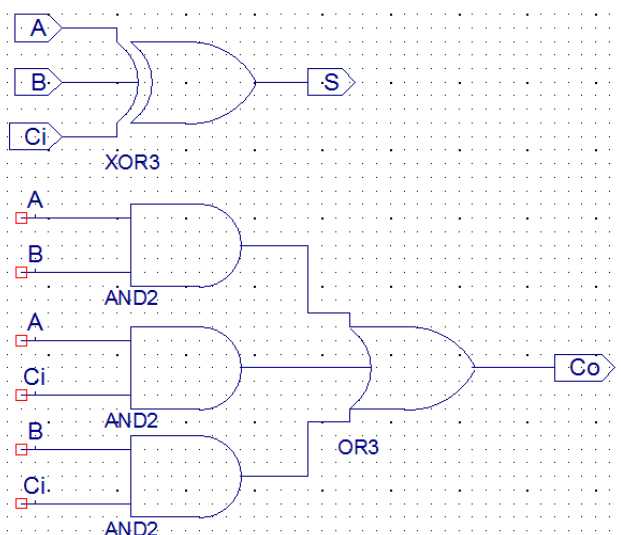


figura 20

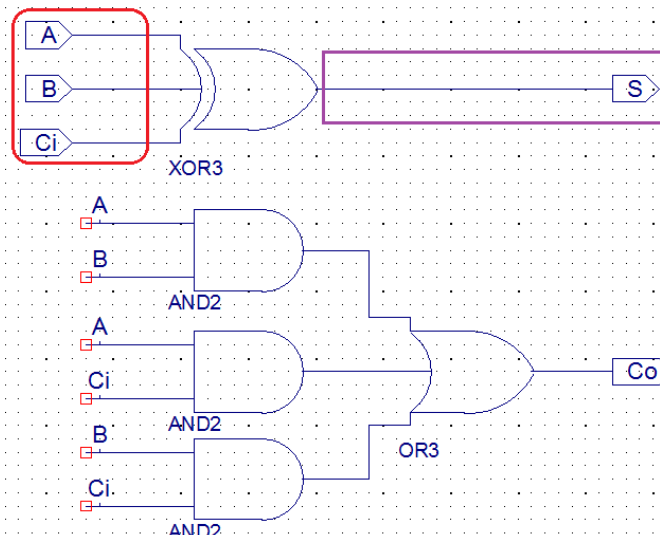


figura 21

Urmând procedura descrisă mai sus, adăugați identificatori de conexiune A, B, Ci. Trebuie să obțineți desenul din figura 20.

În continuare trebuie executată o operație care va contribui la claritatea schemei. Marcherii de intrare A, B, Ci sunt în dreptul numelor iar S nu este aliniat cu C0. Pentru a evidenția marcherii de IO, aceștia se trag ca în figura 21.

Selectați marcherii de intrare A, B și Cin. Selecția se poate face cu CTRL+clic stânga sau prin încadrarea acestora într-un dreptunghi. Dreptunghiul se trasează cu butonul stânga al mouse-ului apăsat. Toate obiectele cuprinse **complet** în acest dreptunghiul se selectează. Un obiect selectat își schimbă culoarea conturului în roșu. Apoi trageți de cei trei marcheri spre stânga. Pentru a trage de un obiect sau grup de obiecte plasați cursorul deasupra obiectului sau grupului și executați „drag”. Pe măsură ce trageți de cei trei marcheri, veți observa apariția a trei conexiuni, câte una per marcher. Procedați la fel pentru a alinia pe S cu Co.

În continuare vom verifica că toate segmentele identificate prin „A” formează o singură conexiune.

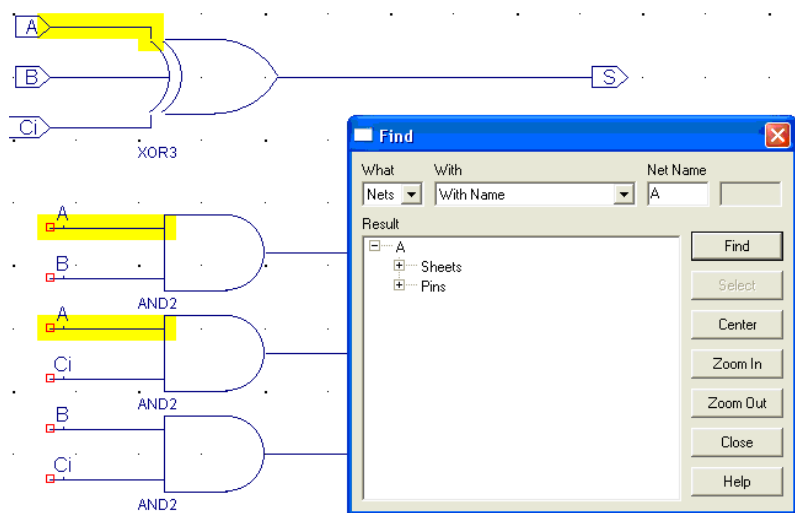
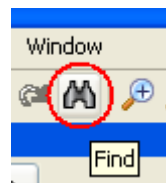


figura 22

Pentru aceasta apăsați butonul „Find”



, iar apoi în fereastra „Find” completați câmpurile „What”, „With” și „Net Name” ca în figura 22. După ce veți apăsa butonul „Find”, toate segmentele cu numele „A” vor fi evidențiate prin desenarea lor pe fond galben, ca în figura 22. Marcajul galben specifică că marcherul IO cu numele „A”, intrarea porți XOR3 și intrarea porții AND2 de sus sunt conectate.



În final **verificați corectitudinea schemei!** Pentru aceasta apăsați butonul . Dacă există erori pe care nu puteți să le corectați sau pe care nu le înțelegeți, **chemați profesorul.** **Salvați** și închideți fereastra de desenare. Apoi reveniți în fereastra proiectului.

Pasul 4: Implementarea proiectului și consumul de resurse.


Implementarea actuală necesită 5 porți. Dacă fiecare poartă ar fi implementată cu un LUT, atunci ar fi nevoie de 5 LUT-uri. Schema este mai clară dar consumul de resurse este aproape dublu. Dacă ar fi necesare 5 LUT-uri, implementarea cu porți ar fi complet nefolositoare deoarece ceea ce contează în final este **prețul produsului și nu frumusețea schemei**. În mod evident, un consum sporit de resurse înseamnă un FPGA mai mare și implicit un preț pe măsură.

Pentru a vizualiza consumul de resurse trebuie parcurse rapoartele de sinteză și mapare. În acest sens trebui ca mai întâi să se înțeleagă procesul de generare a fișierului de configurare FPGA. Acest proces cuprinde următorii pași:

1. **Synthesize** – Sinteza. Programul de sinteză primește la intrare o reprezentare de tip schemă sau HDL și generează o reprezentare echivalentă de tip graf. Acest graf se reprezintă prin intermediul listelor de adiacență și se numește reprezentare netlist. Nodurile grafului sunt de două tipuri: elemente de procesare și conexiuni (fire). Elementele de procesare sunt porți logice și bistabile. Acestea se mai numesc **BELs** – Basic Elements of Logic. În această fază nu contează care sunt de fapt resursele fizice ale dispozitivului programabil, porți logice sau LUT-uri. Această fază este asemănătoare cu **compilarea** programelor. **Deosebirea principală între compilare și sinteză este că în urma compilării se obține o**

succesiune de instrucțiuni cod mașină ce se vor executa secvențial iar în urma sintezei logice se obține un graf.

Pentru a sintetiza sumatorul faceți dublu clic pe procesul **Synthesize**, conform marcajului 1 din figura 23. Sinteza, sau orice alt proces, se mai poate executa făcând clic dreapta pe numele procesului. Va apare un meniu din care se va alege opțiunea **Run**.

După ce sinteza se termină în dreptul acestui proces va apare simbolul .

În continuare se vizualizează raportul de sinteză. Pentru această se selectează **Syntesis Report** din secțiunea **FPGA Design Summary** (vezi marcajul 2 în figura 23). Derulați raportul de sinteză până ajungeți la secțiunea **Final Report**. Din raport rezultă că implementarea necesită o poartă XOR3, o poarta OR3 și 3 porți AND2. Utilizarea resurselor este conform schemei, ceea ce era de așteptat.

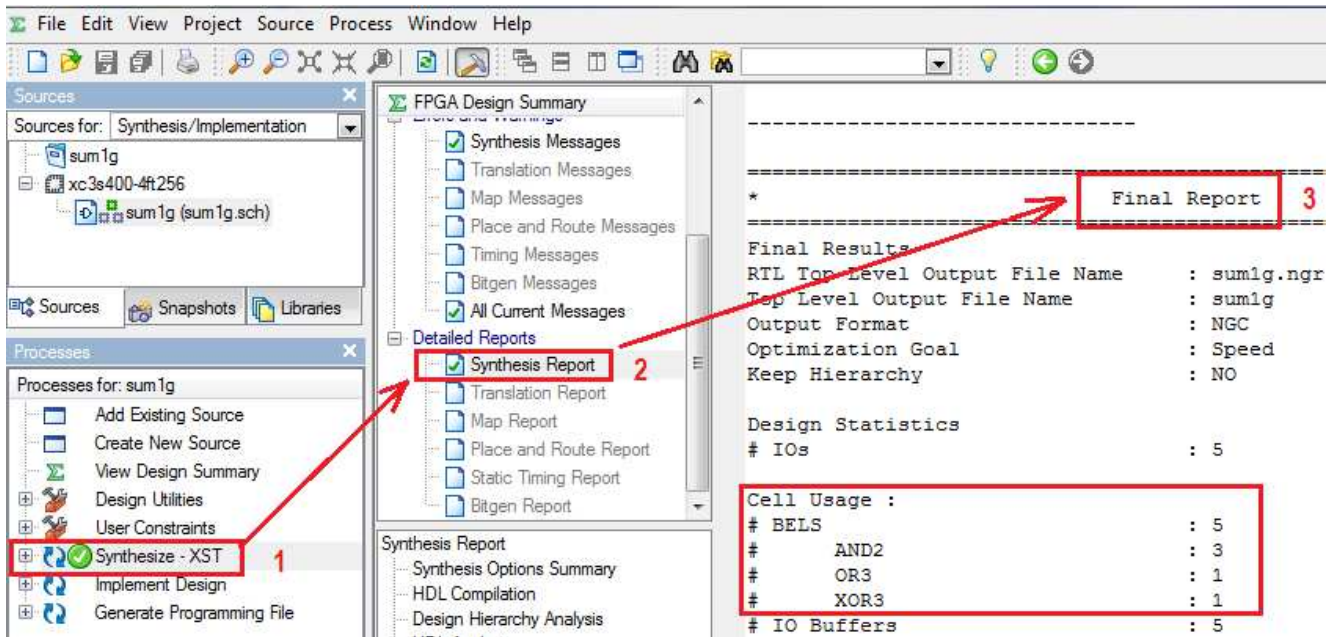


figura 23

Faceți sinteza și verificați că și în cazul proiectului dumneavoastră se obțin rezultatele de mai sus.

2. **Translate** – Traducere. Este posibil ca în proiect să se folosească anumite module pentru care există doar reprezentarea netlist. În această fază toate netlist-urile ce constituie proiectul se contopesc într-o singură reprezentare. Această fază este asemănătoare cu **Linkeditarea** programelor.
3. **Map** – Mapare. În această fază se trece de la BELs la elementele fizice reale. Aici BEL-rile sunt mapate în LUT-uri și bistabile. **Pentru a vedea consumul real de resurse trebuie parcurs raportul de mapare.** Pentru a obține consumul de resurse se parcurg pașii 1, 2, 3 și 4 din figura 24.

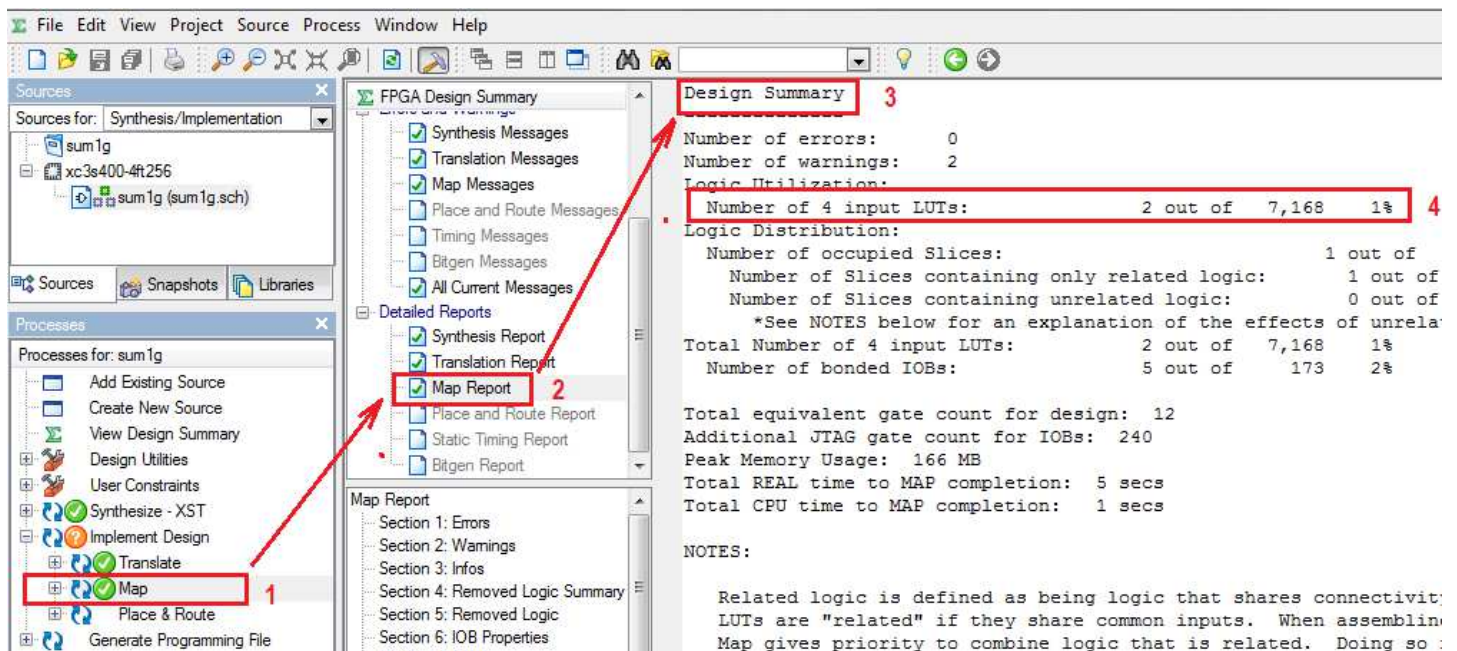


figura 24

Mai întâi se execută maparea făcând clic dreapta pe procesul **Map**. Din meniul contextual care apare se selectează **Run**. Alternativ, se poate face dublu clic pe Map, dar această metodă uneori nu funcționează.

În continuare se vizualizează raportul de mapare (marcaj roșu 2 în figura 24). Pentru această se selectează **Map Report** și se caută secțiunea **Design Summary**. Din raport rezultă că implementare se face doar cu două LUT-uri.

Verificați că și în cazul proiectului dumneavoastră se obțin aceleași rezultate.

Dacă ați obținut rapoartele din figura 23 și figura 24 chemați profesorul!

Concluzie: Nu contează dacă sumatorul elementar se desenează cu porți sau cu LUT-uri. Procesul de mapare face împachetările necesare astfel încât sumatorul elementar se implementează cu doar două LUT-uri.

4. **Place&Route** – Plasare și rutare. LUT-uri și bistabile rezultate în urma mapării se plasează în Slice-uri și CLB-uri și apoi acestea se conectează între ele. Această fază este asemănătoare cu proiectarea unui cablaj.
5. **Generate Programming File** – Generare fișierului de configurare.

Observație: pentru a obține fișierul de configurare ordinea de execuție a proceselor este:

Synthesize → Translate → Map → Place&Route → Generate Programming File

Pentru a executa un proces, toate procesele anterioare trebuie să se fi terminat fără erori. Dacă vreunul din procesele precedente nu a fost rulat, aceasta va fi lansată automat de ISE. În cazul mapării de la punctul 3, procesul translate nu a fost executat explicit dar lansarea în execuție a lui Map face să se execute mai întâi Translate.

În continuare se vor executa procesele necesare generării fișierului de configurare.

Pasul 5: Crearea fișierului de constrângeri și specificarea acestora.

Pentru a verifica funcționarea sumatorului, intrărilor A, B și Ci li se asignează pini FPGA conectați la comutatoare iar ieșirilor S și Cout li se asignează pini FPGA conectați la LED-uri. Această asignare se numește

constrângere de tip LOC (LOC=Locate). Există mai multe tipuri de constrângeri, aceasta fiind una dintre ele. Vom alocă 3 comutatoare glisante pentru A, B, Cin și două LED-uri pentru S și Cout, conform tabelului următor:

Ci	B	A	Co	S
SW2	SW1	SW0	LD1	LD0
		F12		

Din documentația plăcii S3, capitolul *Switches and LEDs*, rezultă că intrarea A este conectată la pinul F12. În plus, pinul FPGA la care este conectată o resursă **este scris chiar pe placă**. Completați tabelul de mai sus.

Correspondența marker I/O - pin se memorează în fișierul de constrângeri. Mai întâi se creează fișierul de constrângeri. Se face clic dreapta pe numele circuitului (xc3s400..) sau pe numele fișierului VHDL sum1g.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 din nou **Next**, iar apoi în fereastra de informare se apasă **Finish**. Dacă totul a decurs conform celor explicate anterior se va obține situația din figura 25. În caz contrar chemați profesorul!

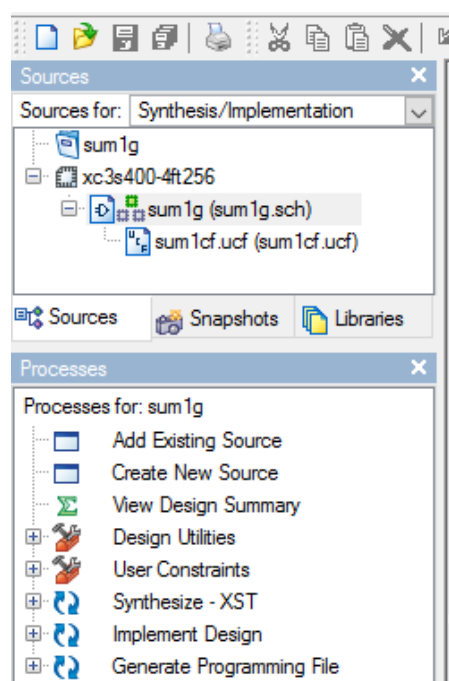


figura 25

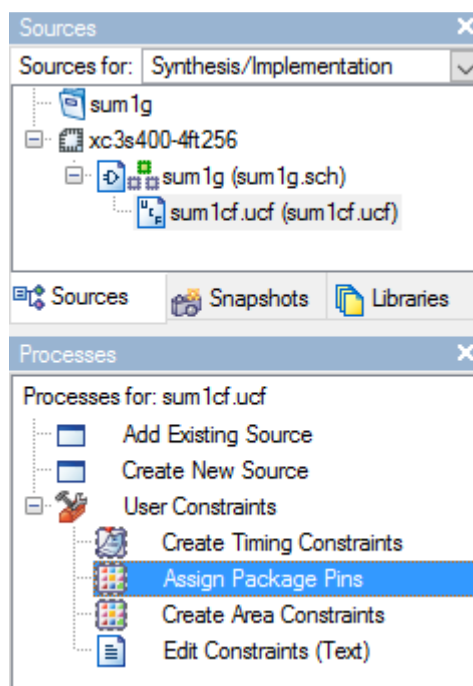


figura 26

Se observă ca procesele posibile pentru un anumit fișier sunt diferite în funcție de fișierul selectat în fereastra **Sources**. Dacă în această fereastră este selectat sum1g.sch, în fereastra **Processes** apar procesele din figura 25. Dacă în fereastra **Sources in Project** se selectează sum1cf.ucf, în fereastra **Processes** apar procesele din figura 26.

Pentru crearea/editarea constrângerilor pornind de la configurația din figura 26, se face dublu clic pe procesul **Assign Package Pins**, pentru a lansa în execuție editorul de constrângeri pentru pini și arie (**PACE=Pin and Area Constraints Editor**) și apoi se completează ca în figura 27.

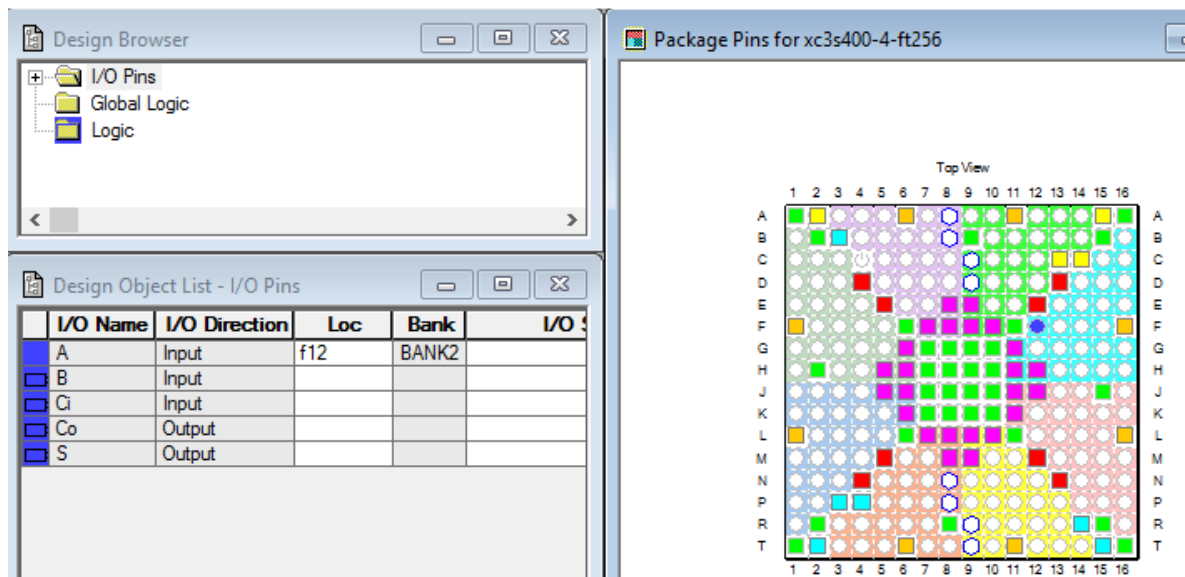


figura 27

Atenție: după ce introduceți ultima valoare LOC apăsați **Enter!**, iar apoi **salvați**.

Pasul 6: Crearea fișierului de configurare și verificarea funcționalității.

Se configurează procesul **Generate Programming File** -> click dreapta, apoi **Properties...** (figura 28).

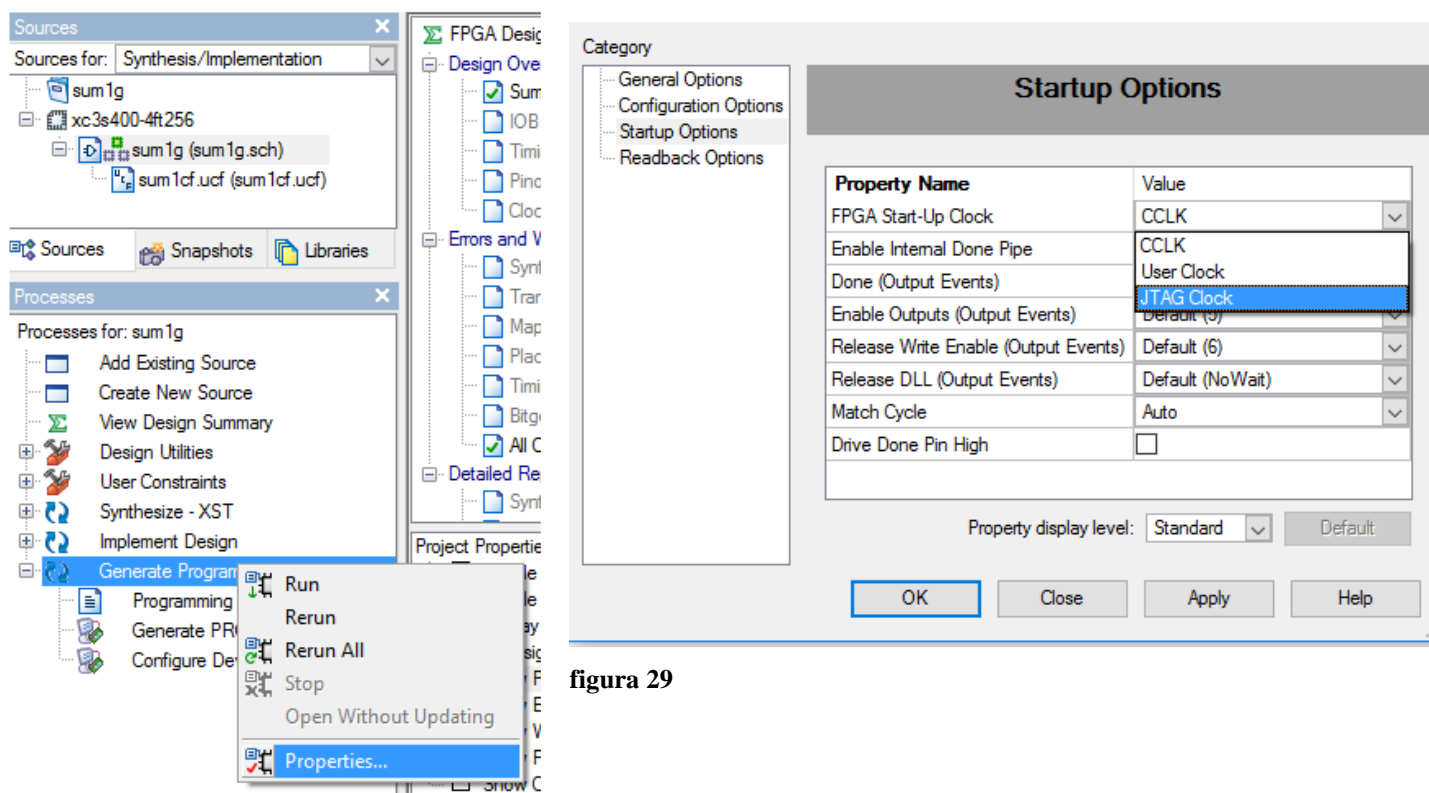


figura 29

figura 28

Modificați Categoria **Startup Option**, Proprietatea **FPGA Start-up Clock**: în loc de **CCLK** se setează valoarea **JTAG Clock** (figura 29).

Există două tipuri de cabluri de conectare: cablu JTAG-USB și cablu paralel JTAG3. Numele cablului este scris pe capătul negru al cablului. **Dacă aveți cablu paralel JTAG3** mai este necesară o setare. Modificați Categoria **Configuration Options**, Proprietatea **Configuration Rate** de la 6 la 3.

În final lansați (se face dublu click) procesul **Generate Programming File**. Ca urmare, în folderul proiectului se va crea fișierul de configurare al FPGA, și anume **sum1g.bit**. Generarea va dura aproximativ 1 minut.

Pasul 7: Configurarea FPGA Spartan 3 de pe placa de dezvoltare

Așa cum s-a precizat anterior, sunt disponibile două tipuri de cabluri de conectare: cablu JTAG-USB și cablu paralel JTAG3. Numele cablului este scris pe capătul negru al cablului. Pașii 1 și 2 se aplică pentru ambele tipuri de cablu. Diferențele apar de la pasul 3.

1. Se conectează cablul de programare ca în figura 30. Atenție la modul de conectare! Este foarte ușor să introduceți cablul decalat! **După ce ați făcut conexiunea chemați profesorul pentru verificare.**



figura 30

2. Se alimentează placa ca în figura următoare:

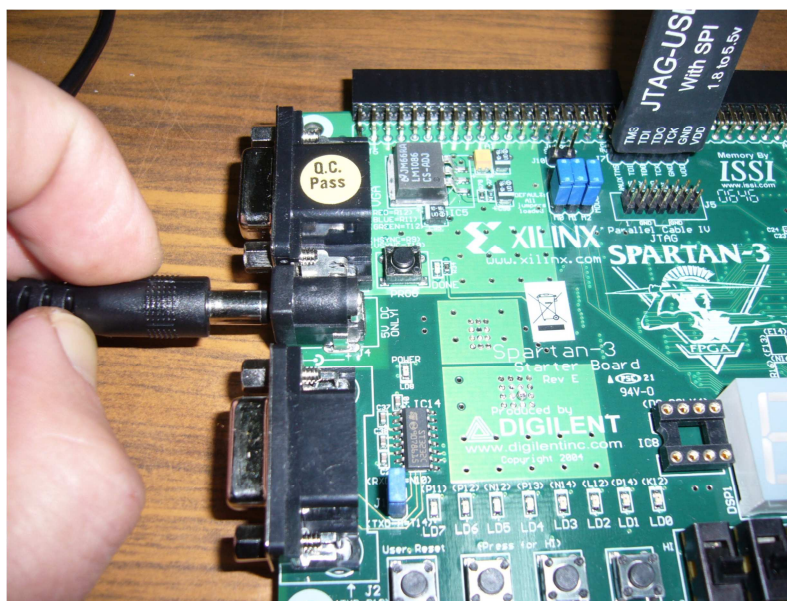


figura 31

3. **Dacă aveți cablu JTAG3 urmați procedura de la pasul 6.**

Dacă aveți cablu **JTAG_USB** se lansează în execuție programul **Adept** care permite programarea FPGA. Va apare fereastra din figura 32.

4. Se apasă butonul **Browse...** corespunzător modulului FPGA. Găsiți fișierul de configurare, **sum1g.bit**, și selectați-l.
5. Se apasă butonul **Program**. Se va face programarea FPGA.

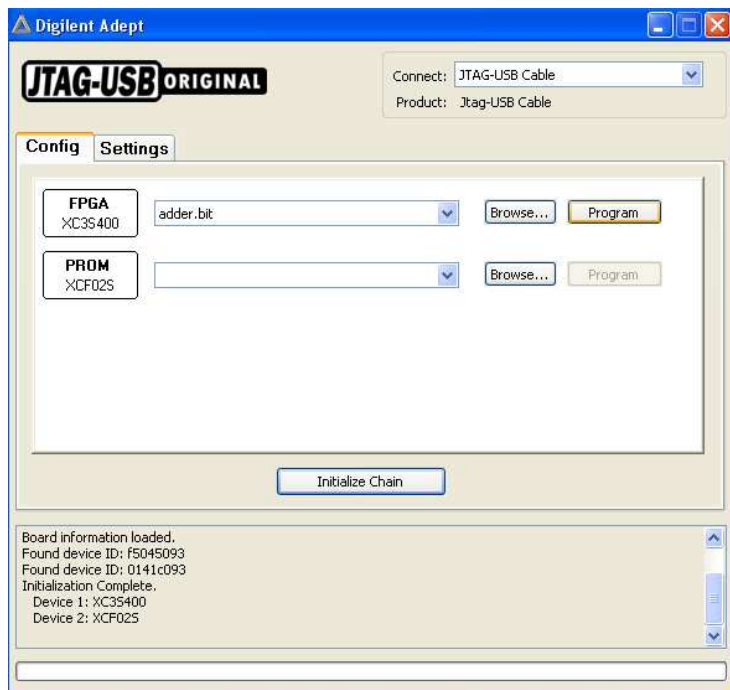
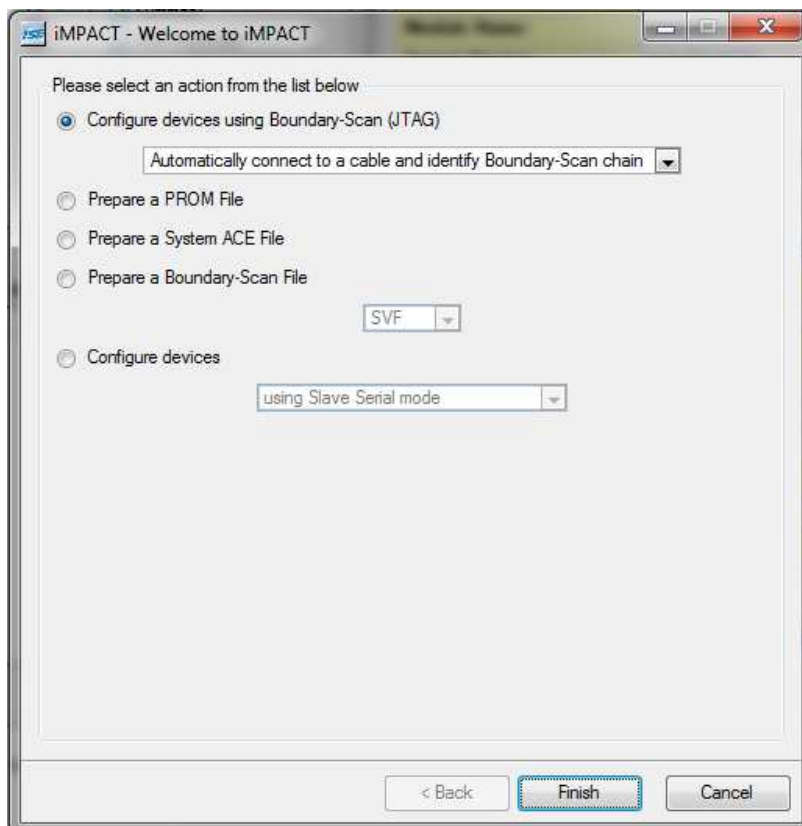
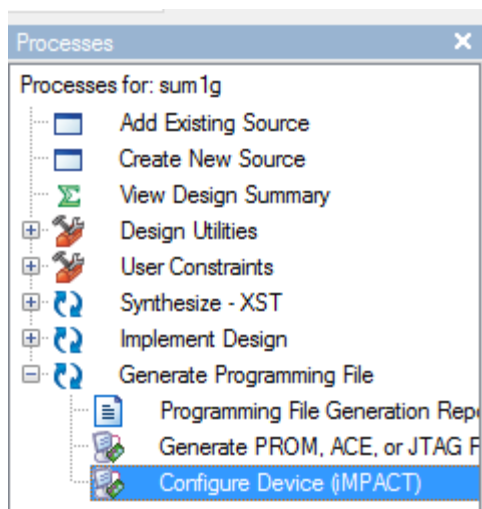


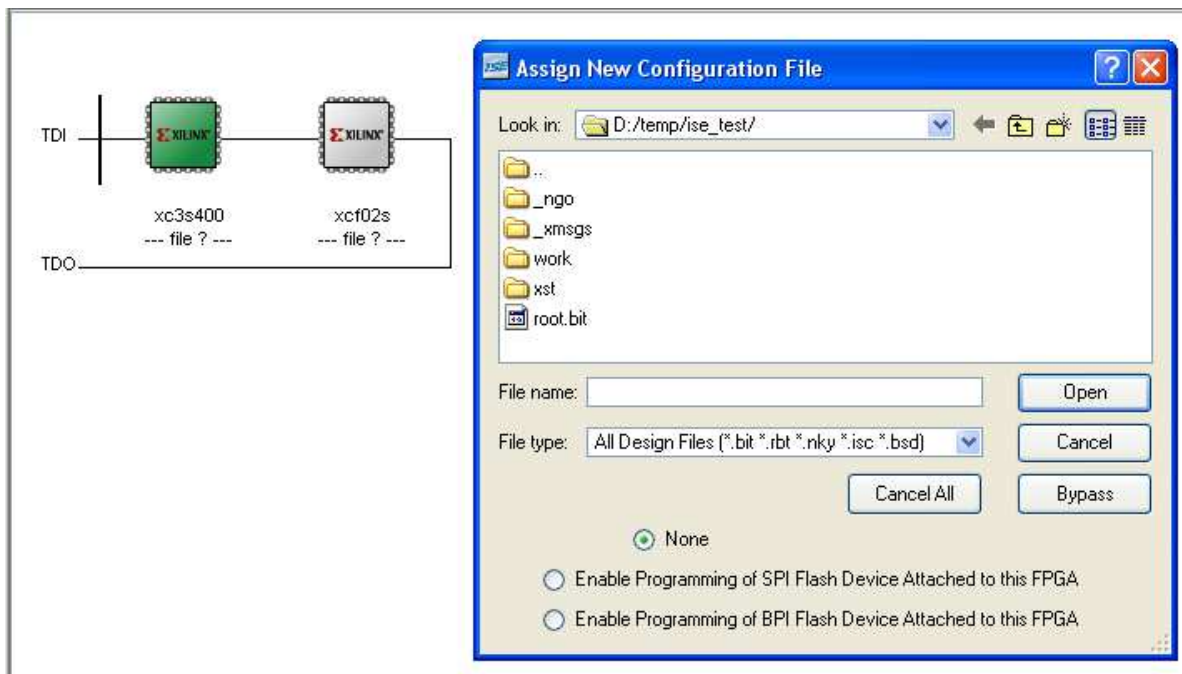
figura 32

Prin acționarea comutatoarelor SW se generează toate combinațiile posibile ale intrărilor și se verifică corectitudinea ieșirilor. **Dacă funcționează corect, chemați profesorul pentru validare!**

6. **Pentru cablu JTAG3:** Lansați în execuție programul **iMPACT**, conform figurii din stânga. Ca urmare va apare figura din dreapta.



În fereastra din dreapta nu modificați nimic, doar apăsați **Finish**. Dacă procesul de conectare cu placa decurge corect va apare următoarea fereastră:



Primul modul care se poate programa este FPGA-ul (xc3s400). Pentru acest modul selectați fișierul cu extensia **bit** din fereastra **Assign New Configuration File**.

În continuare va reapare fereastra **Assign New Configuration File** pentru cel de-al doilea modul (xcf02s). Pentru acest modul **selectați obligatoriu Bypass**.

În continuare faceți clic stânga undeva unde nu există nici un element grafic.

Apoi faceți clic dreapta pe modulul FPGA și din meniul contextual ce va apărea selectați **Program**. Va apărea o nouă fereastră. În această fereastră apăsați OK fără a modifica nimic. Fișierul de configurare se va trimite în FPGA.

Prin acționarea comutatoarelor SW se generează toate combinațiile posibile ale intrărilor și se verifică corectitudinea ieșirilor. **Dacă funcționează corect, chemați profesorul pentru validare!**

Pasul 8 : Sumator pe doi biți – simulare pentru nota >5

Acest pas este opțional și va fi parcurs numai de studenții care doresc o notă mai mare ca 5.

Creați un nou proiect. Proiectul va implementa un sumator pe doi biți. Intrările vor fi:

- $A = a_1a_0$ primul operand, a_0 LSB
- $B = b_1b_0$ al doilea operand, b_0 fiind LSB

Ieșirile vor fi:

- $S = c_{01}s_1s_0 = a_1a_0 + b_1b_0$, c_{01} este ieșirea de transport a sumatorului corespunzător rangului 1.

Intrările și ieșirile se vor conecta la comutatoare și leduri conform tabelului următor:

S			B		A	
c_{01}	s_1	s_0	b_1	b_0	a_1	a_0
LD2	LD1	LD0	SW3	SW2	SW1	SW0
						F12

Obțineți fișierul de configurare și configurați FPGA-ul.

Dacă funcționează corect chemați profesorul pentru validare!