



Universitatea Tehnică “Gheorghe Asachi” din Iași



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

ELECTRONICĂ DIGITALĂ

Proiect

Tema: MODUL ALU-v5

Oancea Cosmin

Coordonator:
Asist. Drd. Ionica Pletea

2024

Tema proiectului:

ALU – v5

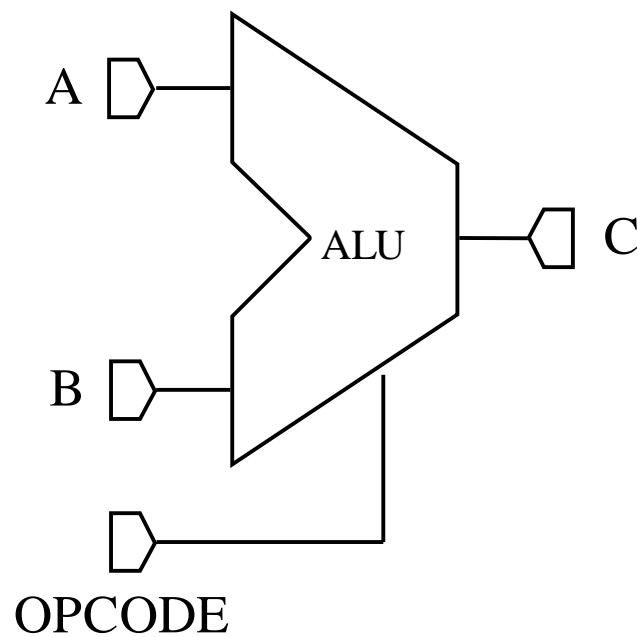
1. Specificațiile proiectului:

Să se implementeze în FPGA prin descriere în limbaj VHDL, utilizând programul VIVADO, modulul prezentat în figura 1 care este descris prin următoarele specificații:

- a) operandii A și B au dimensiunea de 4 biți
- b) operațiile vor fi stabilite prin portul de intrare OPCODE
- c) lista de operații aritmetice: *, /, +, -

Rezultatele vor fi asignate la portul C și vor fi vizualizate pe Displayul 7 segmente de pe placa de dezvoltare.

Descrierea va fi făcută în mod comportamental.



Fișierul bitstream rezultat în urma procesului de implementare va fi verificat utilizând placa de dezvoltare BASYS3

2. Modulul ALU_v5

Unitatea aritmetică și logică (ALU) este responsabilă pentru realizarea operațiilor aritmetice și logice în cadrul unui procesor (CPU) al unui calculator. Funcția sa esențială constă în preluarea datelor de intrare sub formă binară, executarea operațiilor specificate și generarea, stocarea și distribuirea rezultatelor sub formă binară. ALU_v5 efectuează calcule aritmetice, cum ar fi adunări, scăderi, înmulțiri și împărțiri.

Pentru a realiza aceste operații, ALU preia datele de intrare fie din memoria calculatorului, fie din registre. Aceste date sunt procesate conform instrucțiunilor furnizate de către CPU, iar rezultatele sunt generate și utilizate în diverse moduri în cadrul CPU. Aceste moduri includ manipularea datelor, efectuarea comparațiilor și luarea deciziilor necesare.

Unitatea aritmetică și logică reprezintă un element esențial al funcționalității generale a CPU și joacă un rol cheie în execuția instrucțiunilor unui program de calculator. Colaborează cu alte componente, precum registrele și unitățile de control, pentru a îndeplini sarcinile necesare pentru calcul și prelucrare de date într-un sistem informatic.

3. Metoda de implementare

Metoda de implementare aleasă pentru modulul ALU este una comportamentală, folosind resurse precum un circuit FPGA, limbajul de descriere hardware VHDL și software-ul de sinteză Vivado.

Un circuit FPGA (Field-Programmable Gate Array) reprezintă un dispozitiv electronic cu circuite integrate, furnizând flexibilitate în proiectarea și implementarea circuitelor digitale. Aceasta permite configurarea și reconfigurarea circuitelor digitale în funcție de necesități.

Limbajul VHDL (VHSIC Hardware Description Language) este utilizat pentru a descrie și modela comportamentul sistemelor electronice digitale și al circuitelor integrate într-un mod electronic. Acest limbaj facilitează simularea comportamentului circuitelor într-un mediu virtual, contribuind la identificarea și remedierea erorilor înainte de implementarea fizică.

Vivado este un mediu integrat de dezvoltare folosit pentru proiectarea, implementarea și programarea dispozitivelor FPGA produse de Xilinx. Acest mediu furnizează instrumente și fluxuri de lucru care simplifică procesul de dezvoltare a sistemelor digitale bazate pe FPGA, contribuind la eficiența și fiabilitatea implementării.

4. Descrierea (scurtă) a sistemului de dezvoltare BASYS 3

Basys 3 reprezintă o placă de dezvoltare FPGA extrem de versatilă. Echipată cu un FPGA Xilinx Artix-7, această platformă oferă flexibilitate și configurabilitate pentru învățarea și dezvoltarea de circuite digitale. Placa dispune de o gamă diversă de interfețe, inclusiv butoane, comutatoare, LED-uri și porturi USB, facilitând astfel prototiparea rapidă și experimentarea în cadrul proiectelor.

Software-ul utilizat pentru programarea și implementarea FPGA este Vivado, asigurând un mediu integrat și eficient pentru dezvoltare. Basys 3 se remarcă în special în domeniul educațional, fiind potrivită pentru proiecte didactice, prototipare și dezvoltare personală în electronica digitală. Cu resursele sale ample și ușurința de

utilizare, această placă se adresează atât studenților cât și profesioniștilor din domeniul electronicelor digitale, oferind o platformă robustă pentru experimentare și învățare.

5. Editarea fișierului VHDL

Pentru început , am declarat cele 2 intrări A și B pe 2 biți fiecare , OPCODE pe 2 biți și ieșirea C pe 4 biți . Așadar , A - ul va fi de la switch- ul 0- 1 , B- ul de la switch- ul 2 - 3 și OPCODE - ul va fi de la 4-5 . Utilizăm un bloc “ process(A, B, OPCODE) ” , folosit pentru a descrie comportamentul reactiv al unui circuit digital, în funcție de semnalele date: A, B și OPCODE. Astfel , se execută codul în interiorul procesului . Avem de testat 4 operații aritmetice (adunare , scădere , înmulțire , împărțire) cu ajutorul unei structuri de tipul switch/ case știm ce operații avem de făcut (“00” = adunare , ”01” = scădere , ”10” = înmulțire , ”11” = împărțire) .

6. Editarea fișierului de constrângeri

Am comentat 6 pin-uri Switch (pentru A,B si OPCODE) si 7 pin-uri pentru segmentul de display.

7. Descrierea pașilor de sinteză și testarea circuitului rezultat

Procesul de dezvoltare a început prin crearea unui fișier în programul Vivado, la care am atașat fișierele tutorial.vhd și Basys3_Master.xdc, și am selectat tipul de dispozitiv. Ulterior, am efectuat modificări în fișierele vhd și xdc, după care am rulat sinteza și implementarea, urmate de generarea bitstream-ului. Conectarea plăcii la computer s-a realizat prin intermediul unui cablu USB, iar apoi am deschis Open Hardware Manager și am selectat Open Target -> AutoConnect, urmat de Program Device. Astfel, am conectat placa și am testat programul.

Sinteza, primul pas, transformă descrierile în limbaj de înalt nivel ale circuitelor într-un set de porți logice și elemente de interconectare. Importanța acestui pas constă în furnizarea unei imagini a resurselor hardware necesare și a performanței estimate a designului. Sinteza asigură corectitudinea și eficiența codului VHDL și îndeplinește cerințele de resurse ale FPGA-ului.

Implementarea, etapa următoare, implică plasarea și rutarea componentelor pe FPGA după sinteză. Plasarea determină alocarea componentelor logice în FPGA, iar rutarea stabilește interconexiunile dintre acestea. Importanța acestei etape rezidă în eficientizarea plasării și rutării pe FPGA, oferind informații despre consumul de resurse, timpul de propagare și alte metrice de performanță.

Generarea bitstream-ului, ultimul pas, rezultă într-un fișier binar care conține informațiile necesare pentru a programa FPGA și a-i configura blocurile logice. Această configurare specifică FPGA-ului permite realizarea funcționalității dorite. Importanța generării bitstream-ului constă în faptul că acesta este rezultatul final al procesului de dezvoltare, iar fără el FPGA nu poate realiza configurația necesară pentru a-și îndeplini funcționalitatea.

Parcurgerea acestor pași esențiali este crucială pentru a asigura implementarea corectă a designului hardware pe FPGA și pentru a realiza cu succes sarcinile planificate. Fiecare etapă oferă informații esențiale pentru optimizarea resurselor, depistarea erorilor și asigurarea performanței dorite a sistemului.

Bibliografie:

1. VHDL Reference Manual, <http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf>
2. BASYS 3 Reference Manual, <https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual>