LABORATORIJSKI ZADATAK 5

Strukture za računanje

Potrebno predznanje

- Urađena pokazna vežba 6
- Arhitekture procesora

Šta će biti naučeno tokom izrade vežbe?

Nakon urađene vežbe bićete u mogućnosti da:

- Projektujete složenu strukturu za računanje koja može da vrši aritmetičke i logičke operacije
- Kombinujete standardne kombinacione i sekvencijalne komponente, koje su same po sebi veoma jednostavne, u veoma složeni sistem čija primenjivost značajno prevazilazi primenjivost pojedinih komponenti
- Projektujete aritmetičko-logičke jedinice
- Projektujete upravljačke jedinice struktura za računanje
- Napravite vezu između znanja iz programske podrške i fizičke arhitekture digitalnih sistema.

Apstrakt i motivacija

Mozak vašeg ličnog računara je procesor – univerzalna struktura za računanje koja može da izvršava složene aritmetičke i logičke operacije. U ovoj vežbi projektovaćete složenu strukturu za računanje, preteču procesora, koja može da izvršava osnovne operacije – sabiranja, oduzimanja, uvećanja, umanjenja, logičke operacije i pomeranja. Naučićete kako da koristite multipleksere za izbor operanada, kako da projektujete aritmetičko-logičku jedinicu, kako da računane statusne bite (zero, carry, sign) i kako da koristite registre za čuvanje izračunatih vrednosti. Konačno, implementiraćete i upravljačku jedinicu koja izvršava jedan program na vašem procesoru.

Šta treba doneti na termin laboratorijske vežbe?

- VHDL opis krajnjeg koraka u realizaciji sistema (međukoraci nisu potrebni) – krajnji korak je poslednji korak koji uspete da realizujete.
- Testbench za krajnji korak sistema.

ZADACI

1. Aritmetičke komponente strukture za računanje

Za početak, napravite projekat u kome ćete realizovati zadati procesor. Ovaj projekat ćete dopuniti tokom rada na finalnom projektu, nakon ove vežbe. Krajnji sistem će biti opisan u više VHDL datoteka, a one sve treba da budu unutar ovog projekta.

1.1. Registar

U datoteci *reg.vhd* implementirajte 16-bitni registar sa dozvolom upisa. Registar treba da čuva vrednost sa svog ulaza ukoliko je signal dozvole aktivan (na vrednosti 1), u suprotnom čuva staru vrednost.

Tabela 1-1. Prolazi registra

Prolaz	Smer	Funkcija
iCLK	in	signal takta
inRST	in	signal reseta, aktivan na niskom logičkom nivou
iD [15:0]	in	ulazni podatak
iWE	in	signal dozvole upisa
oQ [15:0]	out	vrednost registra

1.2. Multiplekser

U datoteci *mux.vhd* realizovati 16-bitni 9:1 multiplekser. Ovaj multiplekser treba da prosledi jednu od devet mogućih vrednosti ulaza na izlaz, u zavisnosti od vrednosti selekcionog ulaza.

Tabela 1-2. Prolazi multipleksera

Prolaz	Smer	Funkcija
iD0 [15:0] – iD8 [15:0]	in	devet 16-bit ulaza multipleksera
iSEL [3:0]	in	selekcija multipleksera
oQ [15:0]	out	izlaz multipleksera

1.3. Aritmetičko-logička jedinica

U datoteci *alu.vhd* realizovati aritmetičko-logičku jedinicu (ALU) procesora.

ALU treba da računa rezultat operacije nad svojim ulaznim operandima, u zavisnosti od selekcionih bita. Podržane operacije su sumirane u tabeli 1-3.

ALU treba da bude implementirana kao kombinaciona mreža. Pretpostaviti da sve operacije rade sa označenim brojevima u II komplementu.

Tabela 1-3. Operacije ALU

Kod operacije	Operacija
0000	A
0001	A + B
0010	A – B
0011	A and B
0100	A or B
0101	not (A)
0110	A + 1
0111	A – 1
1000	shl (A)
1001	shr (A)
1010	ashl (A)
1011	ashr (A)

Tabela 1-4. Prolazi ALU

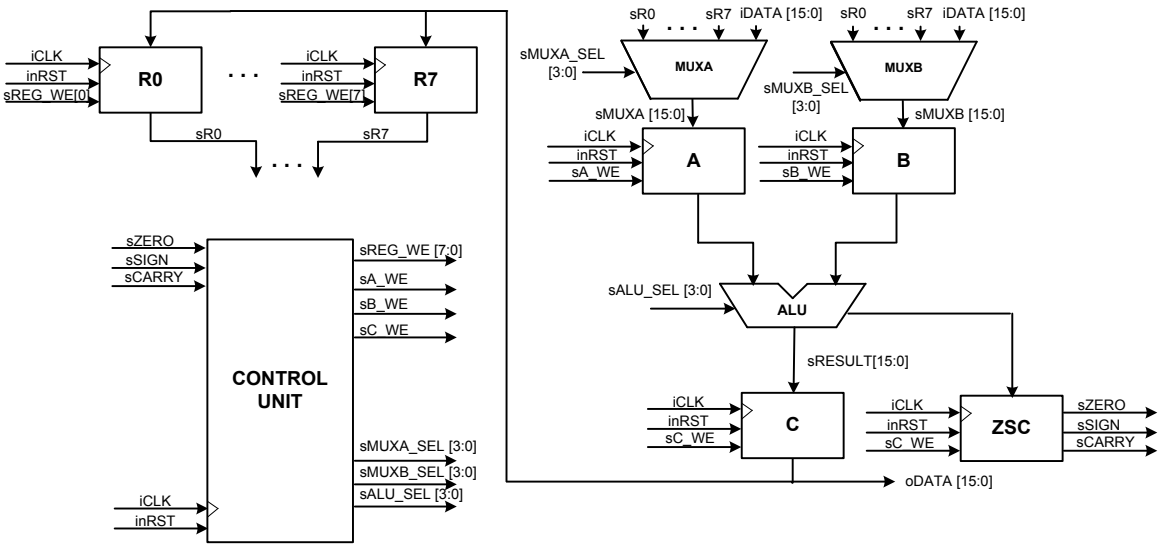
Prolaz	Smer	Funkcija
iA [15:0]	in	prvi ulazni operand
iB [15:0]	in	drugi ulazni operand
iSEL [3:0]	in	selekcija operacije
oC [15:0]	out	rezultat operacije
oZERO	out	statusni bit, rezultat jednak nuli
oSIGN	out	statusni bit, rezultat negativan
oCARRY	out	statusni bit, rezultat ima prenos

Statusni biti takođe treba da budu realizovani kombinaciono, odn. kao funkcije rezultata iz ALU. Aktivna vrednost statusnih bita treba da bude na visokom nivou.

2. Vrh hijerarhije procesora

Vrh hijerarhije procesora je prikazan na slici 2-1.

Komponente realizovane u ranijim delovima vežbe ćemo iskoristiti kako bi realizovali procesor. Vaš zadatak je da instancirate sve potrebne komponente, prema Slici 2-1, kao i da ih povežete internim signalima takođe prema Slici 2-1. Komponente procesora su:



Slika 2-1. Vrh hijerarhije procesora

- R0 : R7 – osam registara opšte namene,
- A, B – registri za ulazne operande u ALU,
- C – registar za rezultat računanja ALU,
- ZSC – tri flip-flopa za registrovanje statusnih bita (realizovati ih u jednom procesu u vrhu hijerarhije),
- MUXA, MUXB – multiplekseri koji selektuju operande za smeštanje u registre A i B,
- ALU – aritmetičko-logička jedinica,
- CONTROL_UNIT – upravljačka jedinica.

Vrh hijerarhije opišite u posebnoj datoteci, nazvanoj *cpu_top.vhd*.

Vrh hijerarhije treba realizovati tako da podrazumeva postojanje opisa upravljačke jedinice sa prolazima prikazanim na slici (prilikom imenovanja prolaza postavite odgovarajuće prefikse). Upravljačka jedinica će biti vaš zadatak na terminu vežbi, a za sada je napravite praznu.

Prolazi procesora su dati u Tabeli 2-1, a upravljačke jedinice u Tabeli 2-2.

Tabela 2-1. Prolazi vrha hijerarhije procesora

Prolaz(i)	Smer	Funkcija
iCLK	in	signal takta
inRST	in	signal reseta, aktivan na niskom logičkom nivou
iDATA [15:0]	in	ulazni podatak
oDATA [15:0]	out	izlazni podatak

Jedini izlaz procesora treba da bude trenutna vrednost 16-bitnog registra rezultata (C). Ovo omogućuje da se posmatraju rezultati operacija koje procesor izvršava. U simulaciji moguće je posmatrati i vrednosti internih signala. Ulaz iDATA [15:0] se može iskoristiti za unos konstanti u procesor, po potrebi.

Tabela 2-2. Prolazi upravljačke jedinice

Prolaz(i)	Smer	Funkcija
iCLK	in	signal takta
inRST	in	signal reseta, aktivan na niskom logičkom nivou
iZERO	in	statusni bit zero
iSIGN	in	statusni bit sign
iCARRY	in	statusni bit carry
oREG_WE [7:0]	out	kontrola registara opšte namene
oA_WE	out	kontrola registra prvog operanda
oB_WE	out	kontrola registra drugog operanda
oC_WE	out	kontrola registra rezultata
oMUXA_SEL [3:0]	out	selekcija za multiplekser prvog operanda
oMUXB_SEL [3:0]	out	selekcija za multiplekser drugog operanda
oALU_SEL [3:0]	out	kontrola aritmetičko-logičke jedinice

OPŠTE NAPOMENE

Prilikom crtanja šeme na papiru ili računaru, koristiti blokove za standardne kombinacione i sekvencijalne mreže – nema potrebe da poznate komponente crtate na nivou logičkih kola. Bitno je da logička šema ispravno opiše logiku sistema. Sve nestandardne komponente koje koristite morate na neki način definisati – bilo istinitosnom tablicom ili opisom na nivou logičke funkcije.

Simulaciju treba raditi posle svakog koraka, jer na sledeći korak treba preći jedino ukoliko je prethodni funkcionalno proveren. U ovoj vežbi ne radimo implementaciju za E2LP platformu.

Na termin vežbe doneti jedan VHDL opis i jedan testbench (nije neophodno donositi ni opis ni testbench međukoraka). Međukoraci su tu da bi vama olakšali put ka kompletnom sistemu i omogućili da vaš sistem projektujete inkrementalno – počev od jednostavnijih komponenata ka složenijim.

Broj poena koje možete osvojiti na vežbama zavisi od koraka do kojeg ste doveli sistem – svaki korak nosi 1 poen (do maksimalno 2), a viši poeni (3, 4) se mogu dobiti izradom modifikacija i malih dopuna vašeg sistema koje ćete dobiti na terminu vežbe. Dodatni zadatak na terminu vežbe vam može doneti i dodatni, peti poen.

ZAKLJUČAK

Sada bi već trebali da imate jasno razumevanje kako rade procesori i kako izvršavaju programe. Razlika između “stvarnog” procesora i ovog koji ste sada realizovali (osim složenosti i količine operacija) je u tome što procesori većinom imaju opšte upravljačke jedinice koje odgovaraju na instrukcije i mogu biti programirane da izvršavaju različite programe. Struktura koju ste realizovali u ovoj vežbi je ograničena na jedan program, onaj koji ste definisali u dodatnom zadatku. Ukoliko se odlučite da radite finalni projekat, imaćete priliku da ovaj procesor proširite sa opštom upravljačkom jedinicom i memorijama, kao i da proširite skup instrukcija koje podržava, uvodeći instrukcije skoka i rada sa memorijom za podatke. Nakon toga, vaš procesor će biti programabilan i moći ćete pisati programe za njega.