

Grupa od dva ili tri studenta realizuje ceo sistem. Potrebno je isprojektovati 32-bitni procesor opšte namene. Procesor se sastoji od jednog jezgra. Interfejs procesora prema okolini su magistrala, RESET signal i signal kloka. Prilikom startovanja sistema RESET signal se postavlja na aktivnu vrednost i nakon nekog vremena se vraća na neaktivnu vrednost. Smatrati da je trajanje RESET signala dovoljno da se ceo sistem resetuje. Magistrala se sastoji od 32 adresne linije, 32 linije podataka i nekoliko kontrolnih linija. Smatrati da pristup memoriji traje 13 signala takta (jedan takt za postavljanje zahteva, jedanaest za dohvaćanje podataka i jedan takt za vraćanje rezultata). Adresibilna jedinica je reč. Reč je veličine 4 bajta. Procesor ima sledeće elemente:

- registarski fajl
- load/store arhitektura sa RMW instrukcijama
- adresiranje je bazirano na vrednostima u registrima
- uniformna instrukcijska reč, veličine 32 bita

Jezgro ima 16 registara opšte namene širine 32-bita. Registri su obeleženi sa R0 – R15. Registar R13 je softverski pokazivač na stek (SP). Registar R14 je link registar. Registar R15 je pokazivač na sledeću instrukciju (PC). Registar specijalne namene je statusni registar CSR. Instrukcijski set dat je u sledećem poglavlju. Statusni registar ima format prikazan na Slika 1. Značenje pojedinih bitova dato je u Tabela 1. Bitove za uslov skoka setuju instrukcije poređenja i druge aritmetičke instrukcije. U slučaju greške (npr. nepostojeća instrukcija) izvršiti instrukciju stop.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSR	N	Z	C	V	Reserved																											

Slika 1 – CSR registar

Bit	Značenje
N	Rezultat je negativan
Z	Rezultat je nula
C	Postoji prenos u sledeći razred
V	Došlo je do prekoračenja

Tabela 1 – Bitovi CSR registra

Kao rešenje potrebno je priložiti sledeće:

1. VHDL kod koji opisuje dati procesor
2. Testove i VHDL kod testbench-a

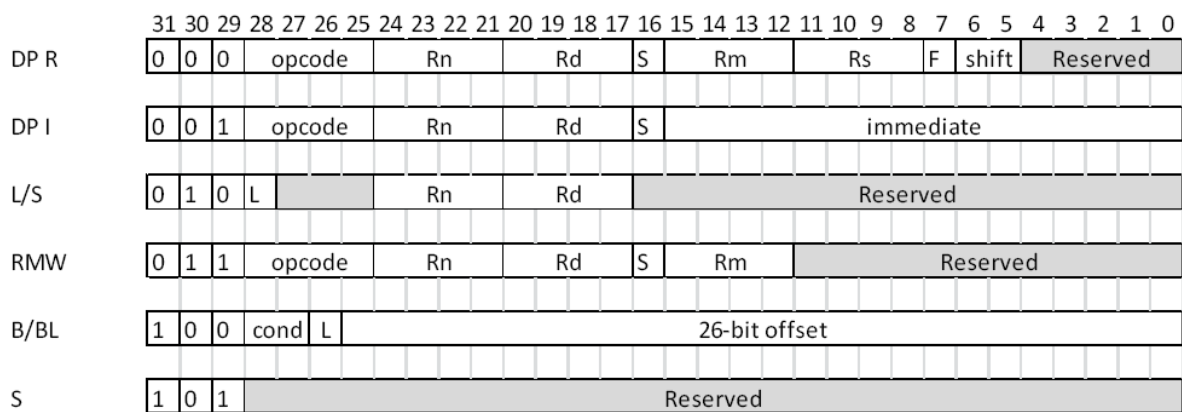
Na odbrani se očekuje sledeće:

1. Da se opis datog procesora može sintetizovati i da sintetizovani model ima sve tražene funkcionalnosti
2. Prikaz simulacije studentskih testova
3. Prikaz simulacije testova koje obezbedi predmetni asistent
4. Ispitivanje u vezi realizacije procesora

Jedan test se sastoji iz dva tekstualna fajla. Jedan fajl inicijalizuje memoriju. Drugi fajl sadrži očekivano stanje memorije podataka nakon izvršavanja programa. Fajl sadrži vrednosti memorijskih lokacija. Format fajla je sledeći: u jednoj liniji nalazi se adresa zapisana kao heksadecimalni broj i vrednost lokacije koja se nalazi na toj adresi zapisana kao binarni broj. Ulazni fajl na početku sadrži dve adrese, zapisane kao heksadecimalni broj, koja će biti smeštena na početku simulacije u PC registre jezgara. Potrebno je dostaviti sledeće tipove testova: test svih instrukcija, test izvršavanja proizvoljnog programa po želji. Kada se procesor zaustavi potrebno je uporediti stanje memorije podataka i vrednosti iz fajla koji sadrži očekivano stanje memorije podataka. U slučaju nepodudaranja prijaviti grešku. Za potrebe simuliranja i testiranja napisati kod za memoriju (ne treba da se vrši sinteza memorije).

Instrukcijski set

Procesor ima 6 tipova instrukcija prikazanih na Slika 2. Tabela 2 prikazuje skraćenice tipova instrukcija. Dužina instrukcije je fiksna i iznosi 32 bita.



Slika 2 - Tipovi instrukcija

Skraćenica tipa instrukcije	Tip instrukcije
DP R	Obrada podataka sa registarskim operandima
DP I	Obrada podataka sa neposrednom vrednošću
L/S R	Load/Store
RWM	Read Modify Write
B/BL	Instrukcije skoka
S	Stop instrukcija

Tabela 2 – Tabela skraćenica tipova instrukcija

cond	Skraćenica	Značenje
00	EQ	Jednako
01	GT	Označen veći
10	HI	Neoznačen veći
11	AL	Bezuslovno

Tabela 3 – Uslovni kodovi

Instrukcije za obradu podataka

Postoje dva tipa instrukcija za obradu podataka. Tip instrukcije određuje šta će biti drugi operand. Instrukcija za obradu podataka postavlja bitove u CSR registru. Tabela 4 prikazuje kodove operacija za različite instrukcije. Instrukcije koje upisuju u PC registar vrše običan skok. Instrukcije za premeštanje podataka imaju samo drugi operand. Ostale instrukcije imaju dva operanda. Rn specificira prvi operand. Rm specificira drugi registarski operand. Immediate specificira drugi neposredni operand. Rd specificira određeno mesto. Instrukcija SWAP postoji samo u prvom tipu instrukcije i menja vrednosti registara Rd i Rm.

opcode	Skraćenica	Instrukcija
0000	AND	Logičko I
0010	SUB	Oduzimanje
0100	ADD	Sabiranje
0101	ADC	Sabiranje sa bitom prenosa
0110	SBC	Oduzimanje sa bitom prenosa
1000	SWAP	Zamena vrednosti registara
1010	CMP	Upoređivanje
1101	MOV	Premeštanje drugog operanda
1111	NOT	Logičko ne

Tabela 4 – Instrukcije za obradu podataka

U slučaju instrukcije sa registarskim operandima prvi operand je vrednost iz registra Rn pomerena ili rotirana za onoliko mesta kolika je vrednost registra Rs. Tip pomeranja ili rotiranja se određuje na osnovu shift bitova, prikazanih u Tabela 5. Bit F određuje da li se radi o pomeranju (vrednost 0) ili o rotiranju (vrednost 1). U slučaju instrukcije sa neposrednim podatkom drugi operand je immediate vrednost koja se proširuje na

32 bita kao neoznačen broj u slučaju da je vrednost S bita 0 ili kao označen broj u slučaju da je vrednost S bita 1. Bit S određuje da li se aritmetičke operacije rade nad neoznačenim brojevima (vrednost 0) ili nad označenim brojevima (vrednost 1).

shift	Skraćenica	Tip
00	LL	Logičko pomeranje/rotiranje ulevo
01	LSR	Logičko pomeranje/rotiranje udesno
10	ASR	Aritmetičko pomeranje/rotiranje udesno
11	-	Nema pomeranja/rotiranja

Load/Store instrukcije

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L/S	0	1	0	L	Rn				Rd				Reserved																			

Load/Store (L/S) instrukcije mogu da pristupaju rečima u memoriji. Adresa podatka kojem se pristupa je u registru Rn. Prilikom Load instrukcije podatak se smešta u Rd, a prilikom Store instrukcije podatak iz Rd se smešta u memoriju. Bit L određuje da li je instrukcija Load (L = 1) ili Store (L = 0).

Read Modify Write instrukcije

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMW	0	1	1	opcode				Rn				Rd				S	Rm				Reserved											

Read Modify Write (RMW) instrukcije omogućavaju atomičnu izmenu memorijskih lokacija. Operacija učita podatak iz memorije sa adrese koja je smeštena u registru Rn. Podatak smesti u registar Rd. Izvrši operaciju koja je data poljem opcode (Tabela 4). Prvi operand je registar Rd, drugi operand je registar Rm. Odrediste je memorijska lokacija čija se adresa nalazi u registru Rn. Bit S određuje da li se aritmetičke operacije rade nad neoznačenim brojevima (vrednost 0) ili nad označenim brojevima (vrednost 1).

Istrukcije skoka

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B/BL	1	0	0	cond	L	26-bit offset																										

Postoje dve instrukcije skoka. Bit L određuje da li se radi o običnom skoku Branch (L = 0) ili o skoku sa pamćenjem povratne adrese u registru R14 Branch and Link (L = 1). Adresa skoka se dobija sabiranjem tekuće vrednosti PC registra (adresa instrukcije skoka plus 1) i proširene označene vrednosti 26-bitnog offset-a na 32 bita. Uslov skoka je određen na osnovu bitova cond, prikazanih u Tabela 3.

Stop instrukcija

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S	1	0	1	Reserved																												

Stop instrukcija zaprljane blokove iz keš memorije jezgra vraća u memoriju i zaustavlja jezgro.

Pogledati obavezno 22 poglavlje knjige "The designer's guide to VHDL" by Ashenden and Lewis.