

Ispit iz Arhitekture i organizacije računara 2

Opis arhitekture i organizacije procesora

Procesor je jednoadresni i ima samo jedan registar podataka (akumulator, Acc) i 16 adresnih registara, X0 do X15, svi su 32-bitni. Pored ovih registara, postoji registar PSW sa uobičajenim značenjem, koji ima indikatore N, Z, C i V, bit I (*Interrupt Enable*) i bite L1 i L0. Registar X15 služi kao pokazivač steka. Stek raste prema višim adresama, a X15 pokazuje na poslednju zauzetu lokaciju.

Memorijske adrese su širine 32 bita, širina magistrale podataka je 32 bita, adresiranje je na nivou 32-bitnih reči. Procesor operiše samo sa 32-bitnim celobrojnim veličinama (u daljem tekstu *reč* označava 32-bitnu veličinu). Ulazno/izlazni i memorijski adresni prostori su razdvojeni. Vreme odziva memorije je neodređeno, magistrala je asinhrona.

Postoje četiri ulaza za spoljašnje maskirajuće prekide IRQ0...IRQ3, pri čemu je IRQ3 najvišeg prioriteta. U mikroprogramu za obradu prekida na steku se čuvaju Acc, PSW i PC tim redom, a I se briše. Biti L označavaju tekući nivo prioriteta izvršavanja i ažuriraju se u mikroprogramu za obradu prekida. Ne postoji mogućnost selektivnog maskiranja prekida. IV tabela počinje od adrese 0.

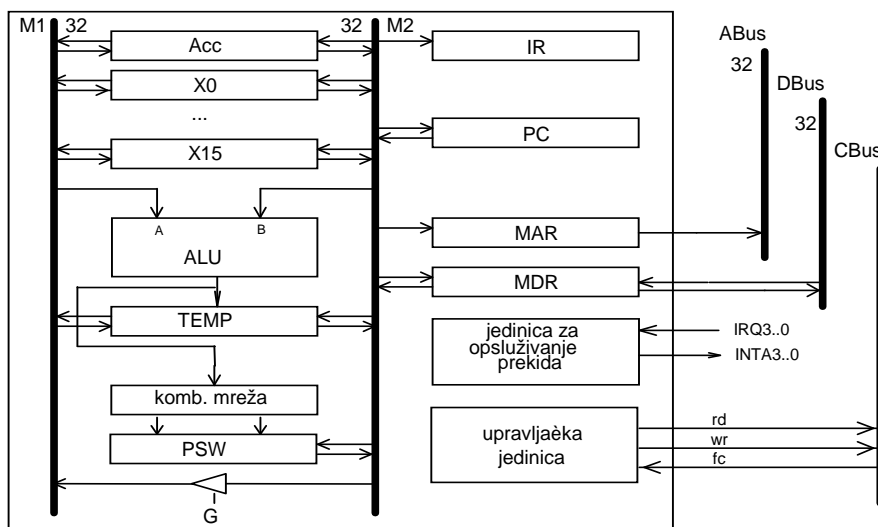
Postoje sledeće grupe instrukcija: bezadresne instrukcije za unarne operacije nad akumulatorom (pomeranja, rotiranja, INC, DEC itd.), jednoadresne instrukcije za operacije nad akumulatorom, instrukcije za rad sa adresnim registrima, instrukcije skokova (bezuslovni i uslovni) i ostale (manipulacije indikatorima, poziv potprograma, povratak iz potprograma ili prekida itd.). Instrukcije su dužine jedne ili dve reči. U prvoj reči su uvek kôd operacije i kôd adresnog registra koji se upotrebljava. U drugoj reči je adresa skoka kod instrukcija skokova, ili konstanta kod instrukcija punjenja adresnog registra.

Jednoadresne instrukcije imaju sledeću sintaksu u assembleru: OP Xi, gde je OP operacija (aritmetička, logička, LOAD ili STORE, itd.), a Xi registar koji učestvuje u adresiranju drugog operanda (prvi je u akumulatoru, kao i rezultat). Jedini način adresiranja za ove instrukcije je registarsko-indirektno adresiranje: adresa operanda dobija se kao sadržaj registra Xi. Operacije nad adresnim registrima date su u tabeli 1.

Sintaksa	Značenje
LOAD Xi, const	Punjenje Xi konstantom
MOV A, Xi	Acc:=Xi
MOV Xi, A	Xi:=Acc

Tabela 1. Operacije nad adresnim registrima

Organizacija procesora data je na slici 1. ALU ima samo sledeće kontrolne ulaze: *add*, *sub*, *incA*, *decA*, *and*, *or*, *notA* i *xor*.



Slika 1. Organizacija procesora

Zadatak (30p):

a) **(5p)** Prikazati strukturnu šemu mreže koja generiše signal logičkog uslova IRQ, koji znači da postoji spoljašnji maskirajući prekid koji treba opslužiti, i to pomoću izlaza registra IRR (*Interrupt Request Register*), koji ima 4 bita u kojima se pamte zahtevi za spoljašnji maskirajući prekid po linijama IRQ0...IRQ3, respektivno, i izlaza registra PSW (odnosno bita ovog registra koji su potrebni).

b) **(20p)** Napisati mikroprogram za ovaj procesor, sa fazom izvršavanja samo za binarne jednoadresne aritmetičke i logičke instrukcije, uključujući instrukciju CMP, a predvideti postojanje ostalih. Pretpostaviti da se za izvršavanje binarnih jednoadresnih aritmetičkih i logičkih instrukcija koristi zajednički mikrokod i da postoji kombinaciona mreža, koja za aktivnu vrednost upravljačkog signala *ALUop* generiše odgovarajuće signale u ALU jedinici na osnovu kôda operacije instrukcije koja se izvršava. Kôd treba da bude prilagođen mikroprogramskoj upravljačkoj jedinici, pri čemu se u jednoj mikroinstrukciji nalaze i polje sa upravljačkim signalima i polja koja definišu uslovni skok u mikroprogramu. Treba pisati i mikroprogram za obradu maskirajućih prekida. Dohvatanje eventualne druge reči instrukcije obavlja se u fazi izvršavanja odgovarajućih instrukcija.

Pretpostaviti da postoji kombinaciona mreža koja povezuje izlaze registra X_i ($i = 0 \dots 15$) sa internom magistralom M1 kada je aktivan upravljački signal *REGout* i signal *regsel*, odnosno povezuje izlaze registra X_i ($i = 0 \dots 15$) sa internom magistralom M2 kada je aktivan upravljački signal *REGout2* i signal *regsel*. Pretpostaviti da postoji kombinaciona mreža koja povezuje internu magistralu M1 sa ulazima registra X_i ($i = 0 \dots 15$) kada je aktivan upravljački signal *REGin* i signal *regsel*, odnosno povezuje internu magistralu M2 sa ulazima registra X_i ($i = 0 \dots 15$) kada je aktivan upravljački signal *REGin2* i signal *regsel*.

Pretpostaviti da se prilikom obrade spoljašnjih maskirajućih prekida pomoću upravljačkog signala *intack* na magistrali započinje ciklus dohvatanja broja ulaza u vektor tabeli.

c) **(5p)** Napisati na assembleru ovog procesora program koji sabira elemente niza koji počinje na adresi 100h, i rezultat smešta na adresu 98h. Dužina niza je na lokaciji 99h, čiji sadržaj nije potrebno sačuvati.

Napomena: Ispit traje 120 minuta. Nije dozvoljena upotreba literature.