

Kolokvijum iz Arhitekture i organizacije računara 2

Opis arhitekture i organizacije procesora

Procesor je troadresni i ima 16 registara opšte namene, R0 do R15, svi su 16-bitni. Postoje i registri PSW i SP sa uobičajenim značenjem. Memorijske adrese su širine 16 bita, širina magistrale podataka je 16 bita, a adresiranje je na nivou 16-bitnih reči. Procesor operiše samo sa 16-bitnim celobrojnim veličinama (u daljem tekstu *reč* označava 16-bitnu veličinu). Ulazno/izlazni adresni prostor je memorijski preslikan. Vreme odziva memorije je neodređeno, magistrala je asinhrona.

Postoje spoljašnji maskirajući prekidi, za koje zahtevi dolaze po linijama IRQ0 do IRQ3 procesora, pri čemu IRQ0 ima najviši prioritet. Prekidni mehanizam je vektorisan, a periferijama se mogu dodeliti proizvoljni ulazi u vektor tabeli. Vektor tabela počinje od adrese na koju pokazuje registar IVTP. Pretpostaviti da postoji kombinaciona mreža koja na internu magistralu postavlja vrednost ulaza odgovarajućeg prekida kada se generiše upravljački signal IVTEout. Pri prekidu se na steku čuvaju PC i PSW tim redom i maskirajući prekidi se onemogućavaju brisanjem bita I u registru PSW. Stek raste prema nižim adresama, a SP ukazuje na poslednju zauzetu lokaciju.

| | | | | | | | |
|----------------------|----|------|---|------|---|------|---|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| OpCode=(0100...1111) | | REG1 | | REG2 | | REG3 | |

Slika 1. Format troadresnih instrukcija

| | | | | | | | | |
|-------------|----|-----|----|---|------|---|------|---|
| 15 | 12 | 11 | 10 | 8 | 7 | 4 | 3 | 0 |
| OpCode=0000 | | L/S | AM | | REG2 | | REG3 | |

Slika 2. Format dvoadresnih instrukcija

Postoje sledeće grupe instrukcija: troadresne instrukcije (aritmetičke, logičke itd.), dvoadresne instrukcije za prenos podataka (LOAD i STORE), jednoadresne instrukcije (CLR, INC, DEC, PUSH, POP itd.), 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 informacije o načinu adresiranja operanada. Samo kod instrukcija prenosa podataka (LOAD i STORE) postoji druga reč u kojoj je adresa ili neposredni operand.

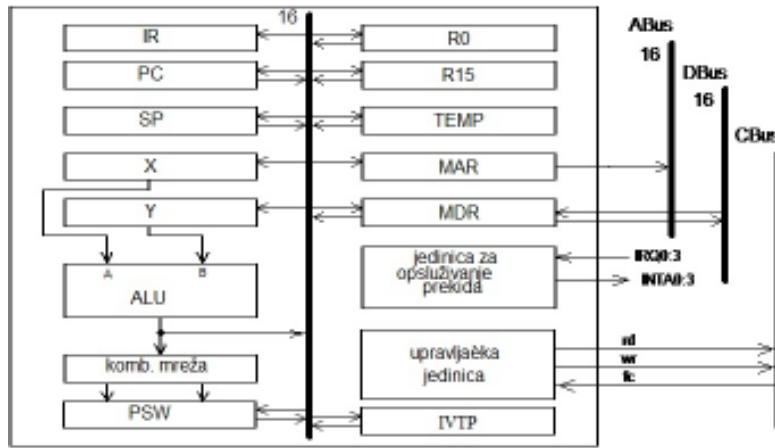
Kod svih troadresnih i jednoadresnih instrukcija (osim dvoadresnih LOAD i STORE), operandi i odredište su isključivo u registrima R0 do R15. Troadresne instrukcije imaju sledeći format (slika 1): polje *OpCode* sadrži kôd operacije, polja *REG1* i *REG2* kodove registara (od 0 do 15) u kojima su prvi i drugi operand, a polje *REG3* kôd registra u koji se smešta rezultat.

Kod dvoadresnih instrukcija za prenos podataka (LOAD i STORE), odredište za LOAD i izvorište za STORE su isključivo registri R0 do R15. Ovaj operand biće nazivan prvim operandom. Drugi operand je neposredni podatak u drugoj reči instrukcije (samo za LOAD), u nekom od registara R0 do R15, ili u memoriji. Samo ove instrukcije operišu podacima u memoriji. Ove instrukcije imaju sledeći format prve reči (slika 2): polje *OpCode=0000* označava da se radi o dvoadresnim instrukcijama, polje *L/S* određuje da li se radi o LOAD ili STORE instrukciji, polje *AM* sadrži kôd načina adresiranja drugog operanda, polje *REG2* sadrži kôd registra u kome je drugi operand kod registarskog direktnog adresiranja, odnosno adresa drugog operanda kod registarskog indirektnog adresiranja, a polje *REG3* kôd registra u kome je prvi operand koji je uvek u registru (odredište za LOAD i izvorište za STORE). Postoji pet načina adresiranja drugog operanda, kao što je prikazano na slici 3.

| AM | Značenje | Primer u assembleru |
|-----|------------------------|---------------------|
| 000 | Neposredno adresiranje | LOAD R0, #1234h |
| 010 | Registarsko direktno | STORE R1, R3 |
| 100 | Memorijsko direktno | STORE R2, 0100h |
| 101 | Memorijsko indirektno | LOAD R1, [0100h] |
| 011 | Registarsko indirektno | LOAD R1, [R2] |

Slika 3. Načini adresiranja

Organizacija procesora data je na slici 4. ALU ima, pored ostalih, i kontrolne ulaze *incA* i *decA* za inkrementiranje i dekrementiranje vrednosti na A ulazu. Mogu se koristiti sve potrebne ostale instrukcije u programiranju, sa odgovarajućim mnemonicima.



Slika 4. Organizacija procesora

Zadatak:

a) (5p) Nacrtati strukturnu šemu mreže koja povezuje izlaze registra R_i ($i = 0 \dots 15$) sa internom magistralom M kada je aktivan upravljački signal REGout i jedan od signala regsel1, regsel2 i regsel3, koji služe za selekciju registra pomoću polja REG1, REG2 i REG3 instrukcijske reči, respektivno. Nacrtati strukturnu šemu mreže koja povezuje internu magistralu M sa ulazima registra R_i ($i = 0 \dots 15$) kada je aktivan upravljački signal REGin i jedan od signala regsel1, regsel2 i regsel3, koji služe za selekciju registra pomoću polja REG1, REG2 i REG3 instrukcijske reči, respektivno.

b) (20p) Napisati mikroprogram za ovaj procesor, sa fazom izvršavanja samo za instrukciju LOAD (za sve načine adresiranja) i troadresne aritmetičke i logičke instrukcije, a predvideti postojanje ostalih. Pretpostaviti da se za izvršavanje troadresnih 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 mikronaredbi nalaze i polje sa upravljačkim signalima i polja koja definišu uslovni skok u mikroprogramu. Treba pisati mikroprogram za obradu prekida. Dohvatanje eventualne druge reči instrukcije treba da bude u fazi izvršavanja instrukcije.

c) (5p) Napisati na assembleru ovog procesora program koji uvećava svaki element niza reči počev od adrese 200h za vrednost registra R4. Niz je dugačak onoliko koliko pokazuje sadržaj registra R3. Registri R1 i R2 su slobodni za korišćenje.

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