

Kolokvijum iz Arhitekture i organizacije računara 2

Posmatra se dvoadresni procesor koji ima 32 registra opšte namene, R0 do R31 i svi su 16-bitni. Registar R0 je hardverski ožičen na nulu, a registar R1 na jedan (drugim rečima, čitanje iz ovih registara uvek daje 0, odnosno 1, a upis se ignoriše). Postoji registar PSW 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). Vreme odziva memorije je neodređeno, magistrala je asinhrona.

Stek raste prema višim adresama, a SP pokazuje na prvu slobodnu lokaciju.

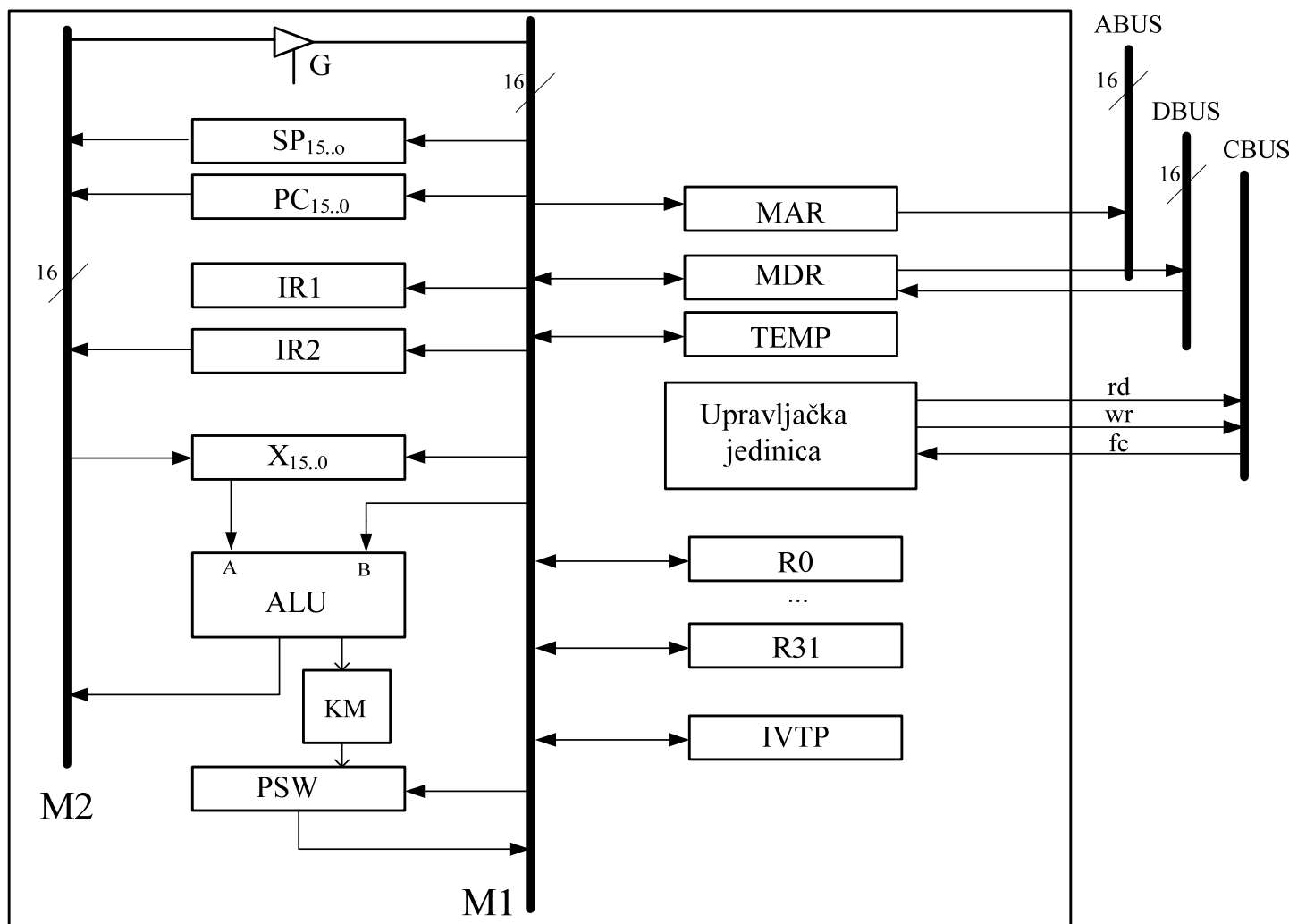
Instrukcije su dužine jedne ili dve memorijske reči. Format prve instrukcijske reči prikazan je na slici. Dvoadresne instrukcije za prvi operand koriste isključivo registarsko direktno adresiranje (polje *Reg1* određuje korišćeni registar koji je ujedno i odredište, osim za STORE), a način adresiranja drugog operanda zadaje poljem *AM* (00 – neposredno adresiranje, 01 - registarsko direktno, 10 – registarsko indirektno sa pomerajem). Polje *Reg2* određuje registar koji se koristi za adresiranje drugog operanda u registarskim adresiranjima. Jednoadresne instrukcije (instrukcija CALL i instrukcije skoka) koriste samo drugi operand (polje *Reg1* se ignoriše).

15	12	11	10	9	5	4	0
OPCODE				AM2	Reg1		Reg2

Procesor poseduje sledeće instrukcije: ADD, SUB, CMP, OR, XOR, STORE, MOV, JMP, JN, JNZ, JV, JC, RTI, CALL, RET sa kodovima 01-0Fh, respektivno i uobičajenim značenjem. Preostali kodovi se ne koriste. Jednoadresne instrukcije skoka i instrukcija CALL ignorišu prvi operand, a instrukcije RTI i RET su bezadresne i ignorišu odgovarajuće adresne bite prve instrukcijske reči. Ostale instrukcije su dvoadresne i mogu imati operande u svim dozvoljenim načinima adresiranja. Instrukcija CMP samo postavlja indikatore, a rezultat oduzimanja operanada nigde ne upisuje. Instrukcija MOV upisuje drugi operand u prvi, a instrukcija STORE obrnuto.

Procesor poseduje samo jednu liniju za spoljašnji nemaskirajući prekid. Procesor ovaj prekid vidi kao signal NMI. Postoji i jedan tip internog procesorskog prekida (prekid usled ilegalnog koda operacije) koji je većeg prioriteta od spoljašnjeg prekida. Ovim prekidima redom su dodeljeni ulazi 0 i 1 u tabeli prekidnih rutina na koju ukazuje registar IVTP. Pri prekidu se na steku čuvaju PC i PSW tim redom. Prekid usled nedozvoljenog načina adresiranja se ignoriše.

Principijelna šema organizacije procesora data je na sledećoj slici. ALU jedinica **nema** ulazne signale za inkrementiranje i dekrementiranje, već samo aluADD, aluSUB, aluOR, aluXOR i transferA.



- (5p) Nacrtati kompletnu strukturnu šemu mreže koja će u odgovarajućim trenucima generisati odgovarajući signal internog procesorskog prekida, tako da se ovaj prekid može obraditi čim se utvrdi njegovo postojanje.
- (4p) Nacrtati kompletnu strukturnu šemu mreže koja će na osnovu adresnih polja efikasno vršiti selekciju odgovarajućeg registra za čitanje/upis, na osnovu signala **Regsel1**, **Regsel2**, **Regin** i **Regout**.
- (20p) Napisati kompletan mikroprogram za ovaj procesor sa fazom izvršavanja samo za sledeće instrukcije: **ADD**, **SUB**, **AND**, **OR**, **XOR**, **CMP**. Predvideti postojanje ostalih naredbi. Za instrukcije koje se identično realizuju pisati zajednički mikroprogram, a ne za svaku posebno. Sve instrukcijske reči potrebno je pročitati pre faze izvršavanja. Pretpostaviti postojanje kombinacione mreže koja će u odgovarajućem trenutku generisati signal  $I_I$  ukoliko postoji više od jedne instrukcijske reči.  
**Potrebno je** pisati mikroprogram za obradu svih prekida. Obradu internog procesorskog prekida izvršiti što je pre moguće. Pretpostaviti da će se u odgovarajućoj prekidnoj rutini (koja se poziva kada se desi interni prekid) vrši korekcija odgovarajuće instrukcije koja je izazvala interni prekid, tako da se može ponovo nastaviti sa prekinutom instrukcijom (a ne sa narednom). Mikroprogram treba da bude prilagođen mikroprogramskoj upravljačkoj jedinici sa spajanjem operacionih i upravljačkih koraka i horizontalnim mikroprogramiranjem. **Potrebno je da traženi deo mikroprograma bude što kraći.** Određivanje broja ulaza u tabeli prekidnih rutina rešiti bilo hardverski bilo u mikroprogramu. U slučaju hardverskog rešenja potrebno je nacrtati potreban hardver.
- (4p) Napisati što efikasniji program na assembleru ovog procesora koji će da inkrementira sve elemente niza koji je smešten u memoriji počev od adrese 300h, a dužina niza je data u registru R2.

**Kolokvijum traje 120 minuta.**

**Na kolokvijumu se može osvojiti maksimalno 30 poena.**