

JOVAN ĐORĐEVIĆ

**ARHITEKTURA
I
ORGANIZACIJA
RAČUNARA**

**GLAVA I
UVOD**

BEOGRAD, 2005.

DJM

PREDGOVOR

Ova knjiga je napisana kao osnovni udžbenik iz arhitekture i organizacije računara i pokriva osnovne koncepte iz arhitekture i organizacije procesora, memorije, ulaza/izlaza i magistrale.

Sistemi.

Autor

Beograd

avgusta 2005.

SADRŽAJ

| | |
|--|----------|
| PREDGOVOR | 1 |
| SADRŽAJ | 3 |
| 1 UVOD | 7 |
| 1.1 STRUKTURA RAČUNARA..... | 7 |
| 1.1.1 <i>Memorija</i> | 7 |
| 1.1.2 <i>Procesor</i> | 9 |
| 1.1.3 <i>Ulazno/izlazni uređaji</i> | 12 |
| 1.1.4 <i>Magistrala</i> | 15 |
| 1.2 FAZE IZVRŠAVANJE INSTRUKCIJE..... | 16 |
| 1.3 ARHITEKTURA I ORGANIZACIJA PROCESORA | 17 |

1 UVOD

U ovoj glavi se razmatraju neki elementi koncepta rada i strukture računara. U okviru toga se, najpre, definišu tipovi informacija u računaru, i to instrukcije i operandi, daju funkcije modula računara, i to memorije, procesora, ulazno/izlaznih uređaja i magistrale, i prikazuje struktura računara. Potom se definišu faze izvršavanja instrukcije. Na kraju se definišu elementi arhitekture i organizacije računara.

1.1 STRUKTURA RAČUNARA

Računari su elektronski uređaji u kojima se rešavanje određenih problema realizuje izvršavanjem određenog skupa aritmetičkih, logičkih i pomeračkih operacija. Operacije koje se u računaru izvršavaju se predstavljaju pomoću binarnih reči koje se nazivaju instrukcije, komande ili naredbe. Skup operacija koje računar može da izvršava je takav da bilo koji problem koji treba da se rešava u računaru može da se razloži na uređeni niz instrukcija računara koji se naziva program. Podaci nad kojima se operacije izvršavaju se, takođe, predstavljaju pomoću binarnih reči koje se nazivaju operandi.

1.1.1 Memorija

Za skladištenje oba tipa binarnih reči koristi se modul računara koji se naziva memorija.

Binarne reči koje predstavljaju instrukcije imaju strukturu datu na slici 1.

| | | | |
|----|------|------|-----|
| OC | SRC1 | SRC2 | DST |
|----|------|------|-----|

Slika 1 Struktura instrukcije

Grupa bitova označena sa OC specificira operaciju koju treba izvršiti. Grupe bitova označene sa SRC1 i SRC2 predstavljaju adrese lokacija u memoriji sa kojih treba očitati oprande i nad njima izvršiti operaciju. Grupa bitova označena sa DST predstavlja adresu lokacije u memoriji u koju treba smestiti rezultat. Ove grupe bitova se još nazivaju: OC – polje koda operacije (operation code), SRC1 i SRC2 – polja dva izvorišna (source) operanda i DST – polje odredišnog (destination) operanda. Polja SRC1, SRC2 i DST se nazivaju i adresna polja instrukcije. Binarne reči koje predstavljaju instrukcije se mogu interpretirati i na druge načine u situacijama kada adrese nekih operanada nisu date eksplicitno odgovarajućim adresnim poljima instrukcije, već su njihove adrese implicitno određene.

Binarne reči koje predstavljaju operande imaju različitu strukturu u zavisnosti od usvojenog načina interpretiranja bitova. Tako, na primer, ako se binarna reč dužine n bitova, u kojoj su bitovi označeni sa $a_{n-1}a_{n-2}...a_1a_0$, interpretira kao celobrojna veličina bez znaka, onda ona predstavlja podatak A čija se vrednost izračunava pomoću izraza

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Uz takav način interpretiranja bitova binarne reči, predstavljaju se celobrojne veličine bez znaka u opsegu 0 do 2^n-1 . Međutim, ako se ista binarna reč intrpretira kao celobrojna veličina sa znakom u drugom komplementu, onda ona predstavlja podatak A čija se vrednost izračunava pomoću izraza

$$A = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

Uz ovakav način interpretiranja bitova binarne reči, predstavljaju se celobrojne vrednosti sa znakom u opsegu -2^{n-1} do $2^{n-1}-1$. Binarne reči koje predstavljaju operande se mogu interpretirati i na druge načine kao, na primer, celobrojne veličine sa znakom predstavljene kao znak i veličina, celobrojne veličine sa znakom u prvom komplementu, veličine u pokretnom zarezu, binarno kodirani decimalni brojevi, niz alfanumeričkih znakova itd.

Neka je problem koji treba rešiti u računaru dat izrazom

$$Y=(A-B)/[C+(D \cdot E)]$$

Da bi se ovaj problem rešio u računaru potrebno ga je razložiti na uređeni niz instrukcija kojima se realizuju operacije koje računar može da izvršava. U ovom slučaju to su operacije sabiranja, oduzimanja, množenja i deljenja. Time se dobija sledeći program:

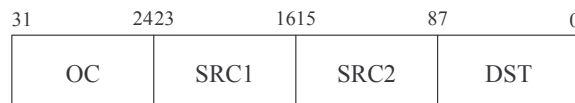
```

SUB  A,  B,  Y
MUL  D,  E,  T
ADD  T,  C,  T
DIV  Y,  T,  Y

```

U instrukcijama programa su kodovi operacija sabiranja, oduzimanja, množenja i deljenja označeni sa ADD, SUB, MUL i DIV, a adrese memorijskih lokacija izvorišnih i odredišnih operanada sa A, B, C, D, E, Y i T. Izvršavanjem ovog programa u računaru se, najpre, čitaju sadržaji memorijskih lokacija sa adresa A i B, nad njima realizuje operacija oduzimanja i rezultat smešta u memorijsku lokaciju na adresi Y. Zatim se čitaju sadržaji memorijskih lokacija sa adresa D i E, nad njima realizuje operacija množenja i rezultat smešta u pomoćnu memorijsku lokaciju na adresi T. Potom se čitaju sadržaji memorijskih lokacija sa adresa T i C, nad njima realizuje operacija sabiranja i rezultat smešta u memorijsku lokaciju na adresi T. Na kraju se čitaju sadržaji memorijskih lokacija sa adresa T i C, nad njima realizuje operacija deljenja i rezultat smešta u memorijsku lokaciju na adresi Y. Program je tako napisan da sadržaji memorijskih lokacija na adresama A, B, C, D i E ostaju neizmenjeni na kraju izvršavanja ovog programa. Zbog toga je i bilo potrebno da se koristi pomoćna memorijska lokacija na adresi T za privremeno smeštanje rezultata operacija množenja i sabiranja.

Memorija u koju se smeštaju instrukcije i operandi ima, na primer, 256 lokacija, pa se za njihovo adresiranje koristi 8 bitova. Instrukcije su smeštene počev od adrese 10 heksadecimalno, a podaci počev od adrese 30 heksadecimalno. Instrukcije su predstavljene binarnim rečima kao na slici 2.



Slika 2 Format instrukcije

Bitovi 31 do 24 predstavljaju polje koda operacije (OC), dok bitovi 23 do 16, 15 do 8 i 7 do 0 predstavljaju polja dva izvorišna (SRC1 i SRC2) i odredišnog (DST) operanda. Dužina instrukcije je 32 bita. Operandi su, radi jednostavnosti, predstavljeni kao 32 bitne celobrojne veličine bez znaka i njihove vrednosti A se interpretiraju prema izrazu

$$A = \sum_{i=0}^{31} 2^i a_i$$

Kako su dužine i instrukcija i operandi 32 bita, uzeto je i da je širina memorijske reči 32 bita.

Kodovi operacija dodeljeni operacijama sabiranja (ADD), oduzimanja (SUB), množenja (MUL) i deljenja (DIV) predstavljeni u heksadecimalnom obliku su: 01 za sabiranje, 02 za oduzimanje, 03 za množenje i 04 za deljenje. Adrese memorijskih lokacija predstavljene u heksadecimalnom obliku dodeljene operandima su 30 za A, 31 za B, 32 za C, 33 za D, 34 za E, 35 za Y i 36 za T. Sadržaji memorijskih lokacija na adresama 30 do 36, predstavljeni u heksadecimalnom obliku, su 00000009, 00000003, 00000002, 00000004, 00000001, 00000000 i 00000000.

Na osnovu toga delovi memorije sa instrukcijama i operandima izgledaju kao na slici 3.

| | | | | |
|-----|----|----|----|----|
| ... | | | | |
| 10 | 02 | 30 | 31 | 35 |
| 11 | 03 | 33 | 34 | 36 |
| 12 | 01 | 36 | 32 | 36 |
| 13 | 04 | 35 | 36 | 35 |
| ... | | | | |
| 30 | 00 | 00 | 00 | 09 |
| 31 | 00 | 00 | 00 | 03 |
| 32 | 00 | 00 | 00 | 02 |
| 33 | 00 | 00 | 00 | 04 |
| 34 | 00 | 00 | 00 | 01 |
| 35 | 00 | 00 | 00 | 00 |
| 36 | 00 | 00 | 00 | 00 |
| ... | | | | |

Slika 3 Deo memorije sa instrukcijama i operandima na početku

1.1.2 Procesor

Za izvršavanje ovog programa potreban je modul računara koji se naziva procesor. On mora, najpre, da čita binarnu reč sa adrese 10 i da je interpretira kao instrukciju saglasno formatu instrukcije sa slike 2. Zatim treba iz instrukcije da koristi polje adrese prvog izvorišnog operanda SRC1 i sa adrese 30 pročita binarnu reč, koju treba da interpretira kao operand predstavljen kao celobrojna veličina bez znaka sa vrednošću 9. Potom, iz instrukcije treba da koristi polje adrese drugog izvorišnog operanda SRC2 i sa adrese 31 pročita binarnu reč koju treba da interpretira ka operand predstavljen kao celobrojna veličina bez znaka sa vrednošću 3. Pošto su oba operanda sada raspoloživa iz instrukcije treba da koristi polje koda operacije OC i na osnovu vrednosti 02 ovog polja izvrši operaciju oduzimanja vrednosti 3 od 00000009 i rezultat 00000006 upiše u memorijsku lokaciju na adresi 35 određenoj poljem adrese odredišnog operanda DST. Sadržaj dela memorije u kome se nalaze binarne reči koje predstavljaju podatke je dat na slici 4. Procesor, zatim, mora da za binarnu reč sa adrese 11 ponovi sve aktivnosti kao i za binarnu reč sa adrese 10. Procesor ovu binarnu reč interpretira kao instrukciju množenja, pa najpre sa adresa 33 i 34 čita binarne reči koje interpretira kao celobrojne veličine bez znaka sa vrednostima 00000004 i 00000001, zatim realizuje njihovo množenje i rezultat 00000004 upisuje u memorijsku lokaciju na adresi 36. Sadržaj dela memorije u kome se nalaze binarne reči koje predstavljaju podatke je dat na slici 5.

| | | | | |
|-----|----|----|----|----|
| ... | | | | |
| 30 | 00 | 00 | 00 | 09 |

| | | | | |
|-----|----|----|----|----|
| 31 | 00 | 00 | 00 | 03 |
| 32 | 00 | 00 | 00 | 02 |
| 33 | 00 | 00 | 00 | 04 |
| 34 | 00 | 00 | 00 | 01 |
| 35 | 00 | 00 | 00 | 06 |
| 36 | 00 | 00 | 00 | 00 |
| ... | | | | |

Slika 4 Deo memorije sa instrukcijama i operandima posle instrukcije SUB

| | | | | |
|-----|----|----|----|----|
| ... | | | | |
| 30 | 00 | 00 | 00 | 09 |
| 31 | 00 | 00 | 00 | 03 |
| 32 | 00 | 00 | 00 | 02 |
| 33 | 00 | 00 | 00 | 04 |
| 34 | 00 | 00 | 00 | 01 |
| 35 | 00 | 00 | 00 | 06 |
| 36 | 00 | 00 | 00 | 04 |
| ... | | | | |

Slika 5 Deo memorije sa instrukcijama i operandima posle instrukcije MUL

Ove aktivnosti procesor ponavlja i za binarnu reč sa adrese 12. Procesor ovu binarnu reč interpretira kao instrukciju sabiranja, pa najpre sa adresa 36 i 32 čita binarne reči koje interpretira kao celobrojne veličine bez znaka sa vrednostima 00000004 i 00000002, zatim realizuje njihovo sabiranje i rezultat 00000006 upisuje u memorijsku lokaciju na adresi 36. Sadržaj dela memorije u kome se nalaze binarne reči koje predstavljaju podatke je dat na slici 6.

| | | | | |
|-----|----|----|----|----|
| ... | | | | |
| 30 | 00 | 00 | 00 | 09 |
| 31 | 00 | 00 | 00 | 03 |
| 32 | 00 | 00 | 00 | 02 |
| 33 | 00 | 00 | 00 | 04 |
| 34 | 00 | 00 | 00 | 01 |
| 35 | 00 | 00 | 00 | 06 |
| 36 | 00 | 00 | 00 | 06 |
| ... | | | | |

Slika 6 Deo memorije sa instrukcijama i operandima posle instrukcije ADD

Za slučaj razmatranog programa, procesor ponavlja ove aktivnosti i za binarnu reč sa adrese 13. Procesor ovu binarnu reč interpretira kao instrukciju deljenja, pa najpre sa adresa 35 i 36 čita binarne reči koje interpretira kao celobrojne veličine bez znaka sa vrednostima 00000006 i 00000006, zatim realizuje njihovo deljenje i rezultat 00000001 upisuje u memorijsku lokaciju na adresi 35. Sadržaj dela memorije u kome se nalaze binarne reči koje predstavljaju podatke je dat na slici 7.

```

...
30    00 00 00 09
31    00 00 00 03
32    00 00 00 02
33    00 00 00 04
34    00 00 00 01
35    00 00 00 01
36    00 00 00 06
...

```

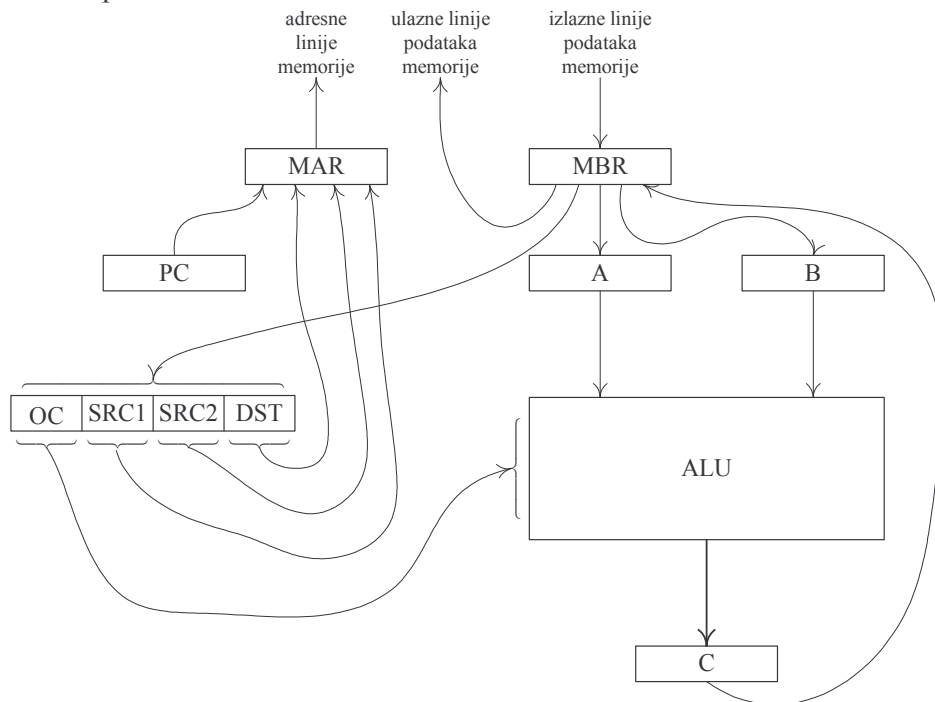
Slika 7 Deo memorije sa instrukcijama i operandima posle instrukcije DIV

Rezultat problema sračunavanja izraza

$$Y=(A-B)/[C+(D\cdot E)]$$

ako vrednosti promenljivih A, B, C, D i E iznose 9, 3, 2, 4, i 1, dobijen izvršavanjem programa sa slike 3, je 00000001 i nalazi se u memorijskoj lokaciji na adresi 35.

Ovakav rad procesora se može ostvariti sa strukturom na slici 8.



Slika 8 Struktura procesora

Na slici 8 je sa PC označen brojački registar koji se naziva programski brojač. Njegov sadržaja se koristi kao adresa memorijske lokacije sa koje treba čitati binarnu reč koja se interpretira kao instrukcija. S obzirom da su binarne reči koje se interpretiraju kao instrukcije smeštene jedna iza druge u memorijskim lokacijama i da stoga treba da se čitaj sekvencijalno, sadržaj programskog brojača PC se, najpre, koristi kao adresa memorijske lokacije sa koje se čita binarn reč, pa se, zatim, njegov sadržaj inkrementira.

Sa MAR je označen adresni registar memorije. U registar MAR se smešta sadržaj koji predstavlja adresu memorijske lokacije sa koje treba pročitati ili u koju treba upisati binarnu reč. Sadržaj registra MAR se vodi na adresne linije memorije.

Sa MBR je označen prihvatni registar podatka memorije. U registar MBR se smešta sadržaj koji je pročitao iz memorijske lokacije i koji dolazi sa izlaznih linija podataka

memorije. U registar MBR se smešta sadržaj koji treba upisati u memorijsku lokaciju. Sadržaj registra MBR se vodi na ulazne linije podataka memorije.

Sa IR je označen prihvatni registar instrukcije. U registar IR se smešta sadržaj registra MBR pročitana sa memorijske lokacije čija je adresa određena sadržajem registra PC. Binarnu reč u registru IR treba interpretirati saglasno formatu instrukcije (slika 1). Grupe bitova polja SRC1 i SRC2 treba voditi u registar MAR i koristiti kao adrese memorijskih lokacija sa kojih pročitane binarne reči treba interpretirati kao dva izvorišna operanda. Prvu pročitane binarnu reč treba iz registra MBR smestiti u registar A, a drugu u registar B. Registri A i B predstavljaju prihvatne registre izvorišnih operanada. Sadržaji registara A i B su vezani na ulazne linije podataka aritmetičko logičke jedinice ALU, dok je sadržaj grupe bitova polja OC registra IR vezan na upravljačke linije ALU. NA izlazima ALU se pojavljuje sadržaj koji predstavlja rezultat operacije specificirane poljem OC instrukcije nad operandima čije su adrese u memoriji specificirane poljima SRC1 i SRC2 instrukcije. Ovaj sadržaj se upisuje u registar C, koji predstavlja prihvatni registar rezultata. Sadržaj registra C se vodi u registar MBR, dok se grupa bitova DST registra IR vodi u registar MAR. Sadržaj registra MAR se koristi kao adresa memorijske lokacije u koju se upisuje sadržaj registra MBR. Time se u memorijsku lokaciju specificiranu poljem DST instrukcije upisuje rezultat operacije specificirane poljem OC instrukcije nad operandima čije su adrese u memoriji specificirane poljima SRC1 i SRC2 instrukcije.

1.1.3 Ulazno/izlazni uređaji

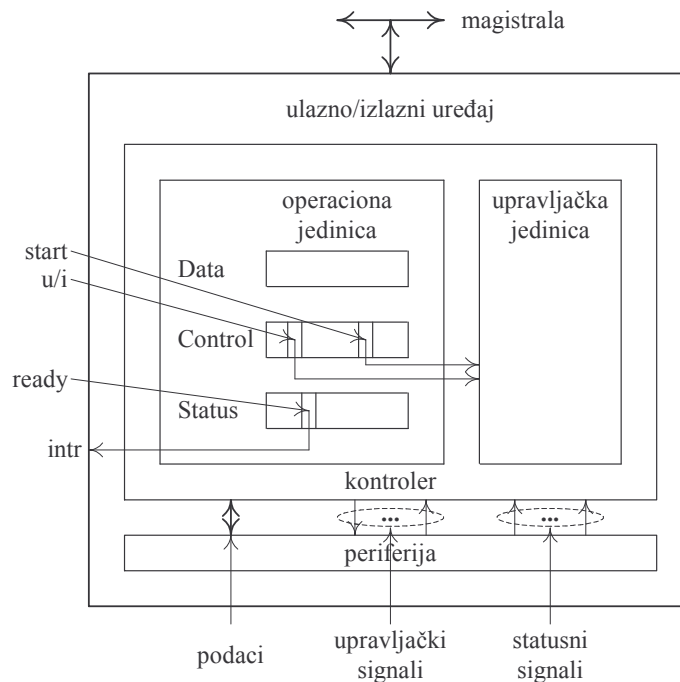
Za unos binarnih reči u delove memorije sa instrukcijama i operandima, koriste se ulazni uređaji. Oni omogućavaju da se na različite, korisnicima računara pogodne, načine specificiraju instrukcije i operandi i prenesu u odgovarajuće delove memorije kao binarne reči. Binarnе reči koje predstavljaju rezultate rešavanog problema se pomoću izlaznih uređaja prenose iz odgovarajućih delova memorije i korisnicima računara prezentiraju u nekoj pogodnoj formi. Struktura ulazno/izlaznog uređaja je data na slici 9.

Ulazno/izlazni uređaj se sastoji od ulazno/izlazne periferije i kontrolera. Periferija se realizuje kao standardni uređaj, na primer optički čitač, laserski štampač, itd., koji mogu da se prebacuju sa jednog računara na drugi računar. Periferije imaju linije za prenos podataka u formi specifičnoj za datu periferiju i upravljačke i statusne linije. Ove linije se koriste da se po protokolu definisanom za datu periferiju podaci prenose od periferije ili ka periferiji.

Za svaku periferiju određenog računara postoji poseban kontroler. On preko magostrale fizički povezuje datu periferiju sa memorijom i procesorom kao modulima računara i omogućuje programsku kontrolu prenosa podataka od periferije ili ka periferiji.

Fizičko povezivanje ulazne periferije preko magistrale uključuje uzimanje podataka iz periferije saglasno protokolu rada sa periferijom i njihovo prosleđivanje preko magistrale memoriji ili procesoru kao modulima računara saglasno protokolu razmene podataka na magistrali. Fizičko povezivanje izlazne periferije preko magistrale uključuje uzimanje podataka iz memoriji ili procesoru kao modulima računara saglasno protokolu razmene podataka na magistrali i njihovo slanje u periferiju saglasno protokolu rada sa periferijom.

Programska kontrola rada ulazno/izlazne periferije uključuje mogućnost da se izvršavanjem programa startuje i zaustavlja rad kontrolera, zatim dobijaju informacije o tome da li je podatak prenet iz ulazne periferije u kontroler ili da li je podatak prenet iz kontrolera u izlaznu periferiju, u na kraju izvrši prenos podatka iz kontrolera ulazne periferije u memoriju ili procesor ili iz memorije ili procesora u izlaznu periferiju. Radi toga u kontroleru postoje tri registra kojima se pristupa izvršavanjem instrukcija procesora. To su upravljački registar (Control), registar podatka (Data) i statusni registar (Status).



Slika 9 Struktura ulazno/izlaznog uređaja

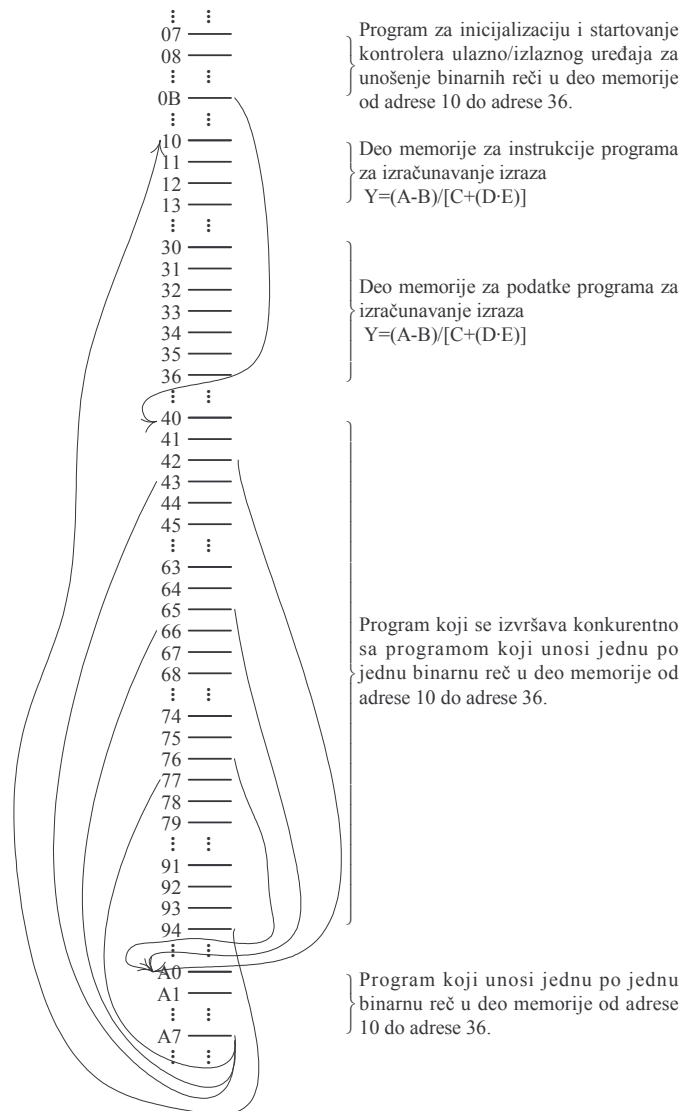
Upravljački registar ima, između ostalog, i razred Start. Aktivna vrednost ovog razreda aktivira upravljačku jedinicu kontrolera da prenosi podatke iz ulazne periferije u kontroler ili iz kontrolera u izlaznu periferiju, dok neaktivna vrednost zaustavlja upravljačku jedinicu.

Registar podatka služi da u njega upravljačke jedinica kontrolera smesti podatak prenet iz ulazne periferije. Statusni registar ima, između ostalog, razred Ready. Aktivna vrednost ovog razreda je indikacija da se u registru podatka nalazi podatak prenet iz ulazne periferije i koristi se za generisanje signala prekida intr koji kontroler šalje procesoru. Signal prekida treba u procesoru da izazove prekid u izvršavanju tekućeg programa i prelazak na izvršavanje programa koji se naziva program prekida. U okviru ovog programa treba da se prenese sadržaj registra podatka u odgovarajuće memorijsku lokaciju ili procesor i potom produži sa izvršavanjem prekinutog programa.

Registar podatka služi i da iz njega upravljačka jedinica kontrolera prenese podatak u ulaznu periferiju. Aktivna vrednost razreda Ready statusnog registra je indikacija da je podatak iz registra podatka prenet u izlazne periferiju i koristi se za generisanje signala prekida intr koji kontroler šalje procesoru. Signal prekida treba u procesoru da izazove prekid u izvršavanju tekućeg programa i prelazak na izvršavanje programa koji se naziva program prekida. U okviru ovog programa treba da se prenese sadržaj odgovarajuće memorijske lokacije ili procesora u registar podatka i potom produži sa izvršavanjem prekinutog programa.

Unošenje binarnih reči koje predstavljaju instrukcije i podatke programa sa slike 3 u deo memorije od adrese 10 do adrese 36, može se organizovati kao na slici 10. Najpre se programom od adrese 07 do adrese 0B inicijalizuje i startuje kontroler da se iz ulazno/izlaznog uređaja čitaju binarne reči. Potom se prelazi na izvršavanje programa od adrese 40 do adrese 94. Za vreme izvršavanja ovog programa kontroler prenosi jednu po jednu binarnu reč iz periferije u registar podatka i posle svake prenete binarne reči generiše prekid. Po prijemu svakog signala prekida procesor prekida izvršavanje ovog programa i prelazi na izvršavanje programa od adrese A0 do adrese A7. Ovim programom se prenosi jedna po jedna binarna reč iz registra podatka u memoriju redom od adrese 10 do adrese 36.

Posle svake prenete binarne reči produžava se sa izvršavanjem prekinutog programa. Uzeto je da je prvi prekid kontroler generisao za vreme izvršavanja instrukcije sa adrese 42, pa se posle ove instrukcije prelazi na program od adrese A0 u okviru koga se prenosi binarna reč iz registra podatka kontrolera u memoriju. Zadnjom instrukcijom ovog programa sa adrese A7 prelazi se na instrukciju sa adrese 43 prekinutog programa. Sledeći prekid kontroler generiše za vreme izvršavanja instrukcije sa adrese 65, pa se posle ove instrukcije prelazi na program od adrese A0 u okviru koga se prenosi sledeća binarna reč iz registra podatka kontrolera u prvu sledeću memorijsku lokaciju posle one u koju je uneta prethodna binarna reč. Zadnjom instrukcijom ovog programa sa adrese A7 prelazi se sada na instrukciju sa adrese 66 prekinutog programa. Slično se dešava i sa prekidom koji dolazi i za vreme izvršavanja instrukcije sa adrese 76, pri čemu se instrukcijom sa adrese A7 prelazi na instrukciju sa adrese 77 prekinutog programa. Ovo traje sve dok se programom od adrese A0 do adrese A7 ne prenese i zadnja binarna reč. Tada se u okviru ovog programa i zaustavlja kontroler.



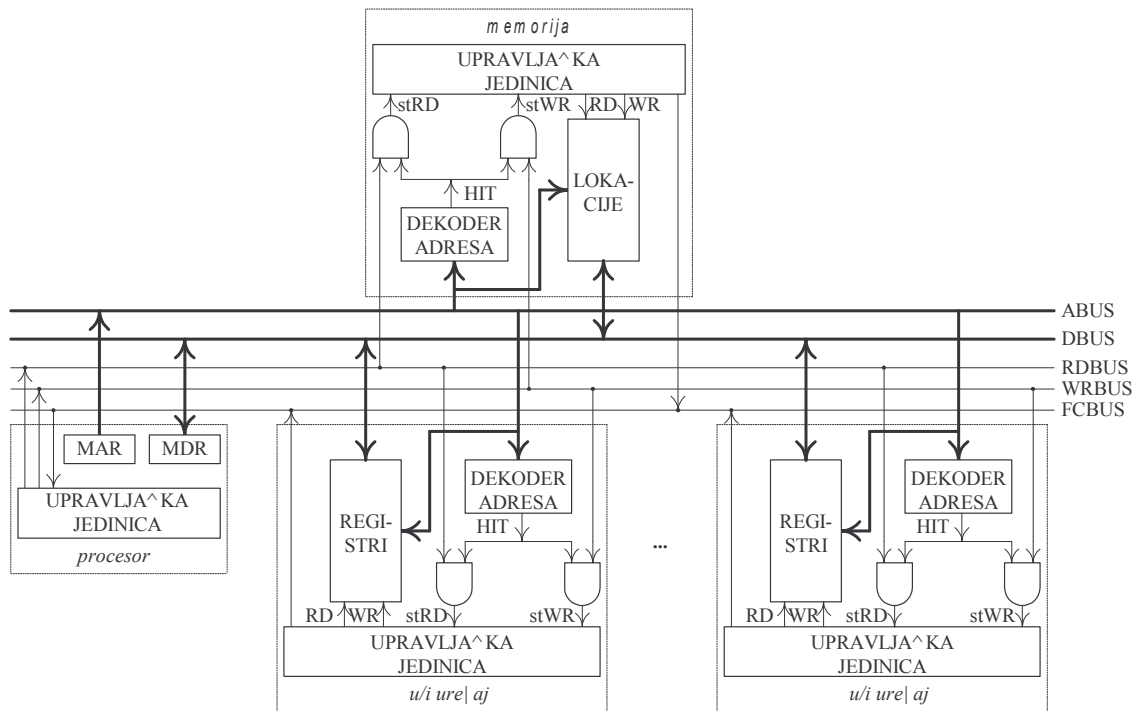
Slika 10 Struktura programa za unošenje binarnih reči

Ovde je pretpostavljeno da izvršavanje programa od adrese 40 do adrese 94 traje duže od vremena potrebnog da se izvršavanjem programa od adrese A0 do adrese A7, i to za svaku binarnu reč, prenesu sve binarne reči u deo memorije od adrese 10 do adrese 36. Stoga kada

se instrukcijom sa adrese 94 pređe na instrukciju sa adrese 10, u delu memorije od adrese 10 do adrese 13 biće binarne reči koje će se interpretirati kao instrukcije programa za izračunavanje izraza, a u delu memorije od adrese 30 do adrese 36 binarne reči koje će se interpretirati kao podaci za program od adrese 10 do adrese 13.

1.1.4 Magistrala

Magistrala je uređena grupa linija koja služi za povezivanje modula računarskog sistema i to procesora, memorije i ulazno/izlaznih uređaja (slika 11). Preko magistrale se prenose sadržaji između registara procesora, memorijskih lokacija i registara ulazno/izlaznih uređaja. Ceo tok prenosa nekog sadržaja između dva modula, naziva se ciklus na magistrali. Modul koji započinje ciklus na magistrali, naziva se gazda (*master*), a modul sa kojim gazda realizuje ciklus, naziva se sluga (*slave*). Gazda može da bude procesor, dok sluga može da bude memorija i ulazno/izlazni uređaji. Procesor čita sadržaje memorijskih lokacija i upisuje sadržaje u memorijske lokacije prilikom čitanja instrukcija i operanada i upisa rezultata, kao sastavnog dela izvršavanja instrukcija nekog programa. Procesor čita sadržaje registara uređaja i upisuje sadržaje u registre uređaja prilikom izvršavanja instrukcija kojima se dobija status uređaja, vrši inicijalizacija uređaja, zadaje režim rada, realizuje startovanje i zaustavljanje uređaja, i vrši prenos podataka između memorije i uređaja i obratno.

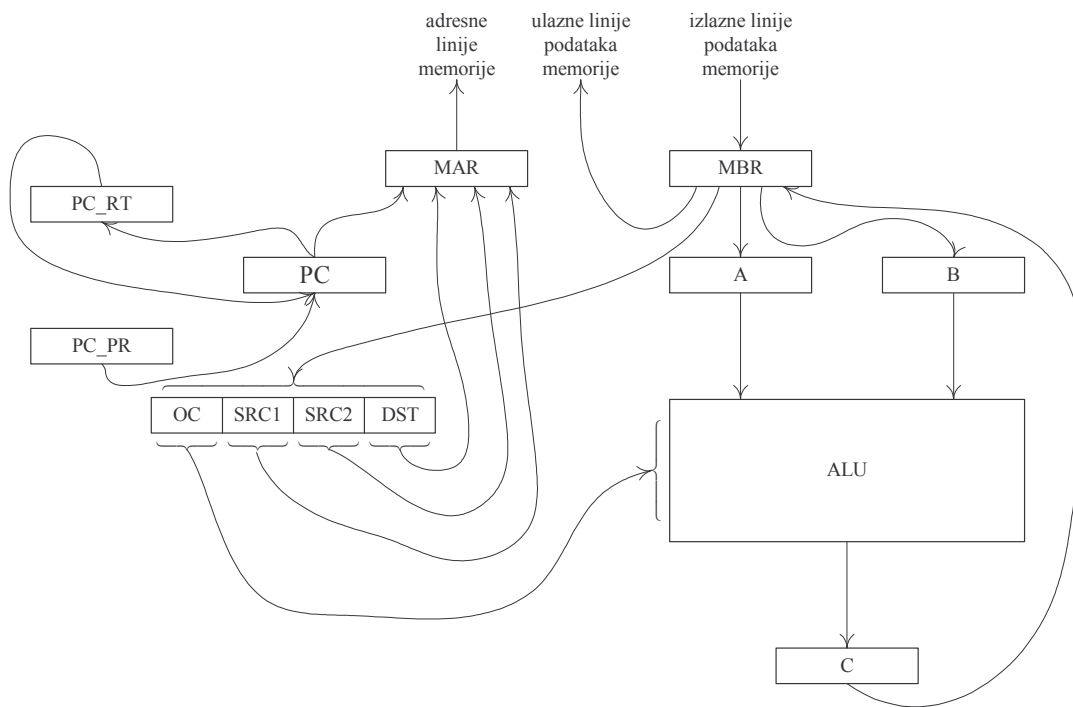


Slika 11 Adresiranje na magistrali sa memorijski preslikanim ulazno/izlaznim adresnim prostorom

Magistralu čine tri grupe linija i to adresne linije (ABUS), linije podataka (DBUS) i upravljačke linije (RDBUS, WRBUS i FCBUS). Po adresnim linijama gazda šalje adresu memorijske lokacije ili registra uređaja prilikom čitanja sadržaja ili upisa sadržaja. Po linijama podataka gazda šalje sadržaj koji treba da se upiše u memorijsku lokaciju ili registar uređaja čija adresa se nalazi na adresnim linijama. Po linijama podataka sluga šalje očitani sadržaj memorijske lokacije ili registra uređaja čija adresa se nalazi na adresnim linijama. Po upravljačkim linijama gazda šalje signale kojima određuje da li treba da se realizuje čitanje (RDBUS) ili upis (WRBUS) sadržaja. U nekim realizacijama magistrale po upravljačkoj liniji (FCBUS) i sluga šalje signal kojim signalizira gazdi da su čitanje ili upis sadržaja realizovani.

1.2 FAZE IZVRŠAVANJE INSTRUKCIJE

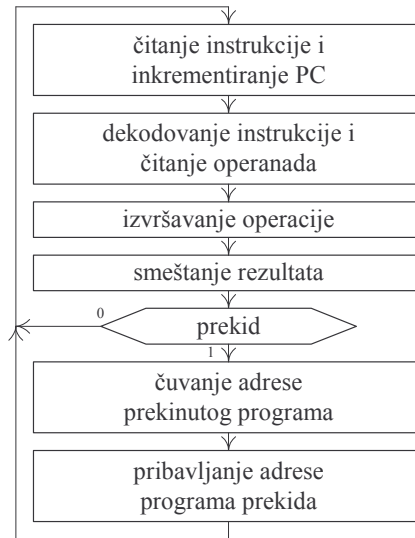
Da bi se omogućilo reagovanje u procesoru na signal prekida i skok na program prekida, strukturu procesora sa slike 8 treba dopuniti sa dva registra i to PC_RT i PC_PR, čime se dolazi do strukture procesora sa slike 12. Registar PC_RT treba da čuva adresu povratka iz programa prekida u prekinuti program, dok registar PC_PR treba da sadrži adresu programa prekida. Na kraju izvršavanja instrukcije treba, ukoliko je tokom njenog izvršavanja generisan prekid, prebaciti sadržaj registra PC u registar PC_RT i u registar PC upisati sadržaj registra PC_PR. Time se obezbeđuje da kada se sada sadržaj registra PC koristi kao adresa sa koje će se čitati instrukcija, to bude prva instrukcija programa prekida. Time se prelazi na program prekida. Pored toga, na kraju programa prekida mora uvek da bude posebna instrukcija koja prebacuje sadržaj registra PC_RT u registar PC. Time se obezbeđuje da kada se sada sadržaj registra PC koristi kao adresa sa koje će se čitati instrukcija, to bude instrukcija prekinutog programa koja bi se izvršavala da nije bilo prekida i prelaska na program prekida.



Slika 12 Struktura procesora sa reakcijom na prekid

Na osnovu svega iznetog izvršavanje instrukcije se sastoji iz sledećih faza: čitanje instrukcije i inkrementiranje registra PC, dekodovanje instrukcije i čitanje operandi, izvršavanje operacije, smeštanje rezultata, čuvanje adrese prekinutog programa i pribavljanje adrese programa prekida (slika 13). Kroz faze čuvanje adrese prekinutog programa i pribavljanje adrese programa prekida se prolazi samo ukoliko je tokom prolaska kroz prethodne faze generisan prekid.

Tokom izvršavanja faze čitanje instrukcije i inkrementiranje registra PC najpre se sadržaj registra PC prebacuje u registar MAR i vrši inkrementiranje sadržaja registra PC, zatim se sadržaj memorijske lokacije čija je adresa određena sadržajem registra MAR upisuje u registar MBR i na kraju sadržaj registra MBR upisuje u registar IR.



Slika 13 Dijagram toka izvršavanja instrukcije

Tokom izvršavanja faze dekodovanje instrukcije i čitanje operandata najpre se grupa bitova iz polja SRC1 registra IR prebacuje u registar MAR, zatim se sadržaj memorijske lokacije čija je adresa određena sadržajem registra MAR upisuje u registar MBR i na kraju sadržaj registra MBR upisuje u registar A. Ovo se ponavlja tako što se grupa bitova iz polja SRC2 registra IR prebacuje u registar MAR, zatim se sadržaj memorijske lokacije čija je adresa određena sadržajem registra MAR upisuje u registar MBR i na kraju sadržaj registra MBR upisuje u registar B.

Tokom izvršavanja faze izvršavanje instrukcije se nad sadržajima registara A i B u aritmetičko logičkoj jedinici ALU realizuje operacija određena grupom bitova iz polja OC registra IR i rezultat upisuje u registar C.

Tokom izvršavanja faze smeštanje rezultata najpre se grupa bitova iz polja DST registra IR prebacuje u registar MAR, zatim sadržaj registra C prebacuje u registar MBR i na kraju sadržaj registra MBR upisuje u memorijsku lokaciju čija je adresa određena sadržajem registra MAR.

Ukoliko tokom prethodnih faza nije generisan prekid prelazi se na fazu čitanje instrukcije i inkrementiranje registra PC. Time se završava izvršavanje tekuće instrukcije i prelazi na izvršavanje prve sledeće instrukcije u programu. Ukoliko je tokom prethodnih faza generisan prekid prelazi se na faze čuvanje adrese prekinutog programa i pribavljanje adrese programa prekida.

Tokom izvršavanja faze čuvanje adrese prekinutog programa, sadržaj registra PC se prebacuje u registar PC_RT.

Tokom izvršavanja faze pribavljanje adrese programa prekida, sadržaj registra PC_PR se prebacuje u registar PC i prelazi se na fazu čitanje instrukcije i inkrementiranje registra PC. Time se završava izvršavanje tekuće instrukcije i prelazi na izvršavanje prve instrukcije programa prekida.

1.3 ARHITEKTURA I ORGANIZACIJA RAČUNARA

Kod razmatranja računarskih sistema koriste se pojmovi arhitekture računara i organizacije računara. Pod arhitekturom računara se podrazumeva sve ono što treba da se zna o računaru da bi za njega mogli da se napišu programi koji će se uspešno izvršavati i uvek davati

identične rezultate bez obzira na to kako je računar realizovan. Pod organizacijom računara se podrazumevaju različiti načini realizacije računara.

Elementi arhitekture računara su programski dostupni registri i memorijske lokacije, tipovi podataka, formati instrukcija, načini adresiranja, skup instrukcija i mehanizam prekida.

Elementi organizacije računara su moguće tehnike realizacije modula računara, kao na primer realizacije operacionih jedinica sa direktnim vezama, jednom ili više internih magistrala, realizacije upravljačkih jedinica nekom od ožičenih tehnika, mikroprogramskih tehnika ili tehnikom preklapanja različitih faza većeg broja instrukcija, realizacije magistrala određenih širina, korišćenje ili ne korišćenje keš memorija, različite realizacije keš memorija itd.