

4.3 ORGANIZACIJA PROCESORA

Registar naredbe – cuva glavnu rec

Pomocni registar – cuva dodatnu rec

Programski brojac – broji dobavljene reci

Aritmeticko-logicka jedinica – obavlja operacije

Vezne linije – povezuju delove procesora

Upravljacka jedinica – upravlja stanjima prekidaca iz veznih linija

Registri posebne namene: registar naredbe, pomocni registar, programski brojac, status registar I registri iz aritmeticko-logicke I upravljacke jedinice

Registri 1. I 2. podatka su posebni registri gde se smestaju podaci pre transformisanja

Registar konstante – sadrzi konstantu 1 kojom se uvecava programski brojac

Sklopovi saberi, oduzmi, I, ili, ne, levo, desno – ostvaruju logicke funkcije pojedinih naredbi

P bit ulazi (zbog dvostruke preciznosti) – izlaze N M P V

Izvršavanje **masinske naredbe** se sastoji iz faze dobavljanja I faze obavljanja

Faza dobavljanja – dobavlja se obavezna rec I smesta u registar naredbe, uvecava se programski brojac

Faza obavljanja – dobavlja se dodatna rec ako postoji I smesta u pomocni registar I izvršava se naredba

4.4 UPRAVLJANJE PROCESOROM

Upravljanje ima **ciklican** karakter – u svakom ciklusu se obavi jedan elementarni deo naredbe

Obavljanje jednog **elementarnog** dela naredbe – obavljanje jedne mikro-naredbe te naredbe

Vise **mikro-naredbi** – **mirko-program**

Faza dobavljanja – mikro-program dobavljanja, a **faza obavljanja** – mikro-program obavljanja

Ulazna adresa – adresa ulazne naredbe (ona mora doci u programski brojac)

Inicijalni mikro-program – puni ulaznu adresu u programski brojac (aktivira mikro-prog dobavljanja)

4.5 UPRAVLJACKA JEDINICA

Zadatak **upravljacke jedinice** – pre obavljanja mikro-naredbi dovodi prekidace u potrebno stanje I zadrzi ih u tom stanju dok se naredba ne obavi

Upravljacki registar – svaki bit je jedan prekidacki argument; odredjuje koja se mikro-naredba obavlja u ciklusu I predstavlja masinski oblik obavljane mikro-naredbe

Dobavljanje mikro-naredbe – prvi poluciklus, **obavljanje mikro-naredbe** – drugi poluciklus

T – takt; omogucuje razlikovanje poluciklusa dobavljanja od poluciklusa obavljanja (1 – dobavljanje, 0 – obavljanje)

Pravilna periodicnost izmene vrednosti T uvodi **pojam vremena**

Mikro-programska memorija – cuva masinske oblike mikro-naredbi; njenu velicinu odredjuje broj bita u masinskim oblicima; broj lokacija zavisi od ukupnog broja mikro-naredbi u svim mikro-programima (128 za KONCEPT; **7 bita** za lokaciju)

Registar sekvence – da bi naredna naredba bila raspoloziva u poluciklusu dobavljanja, ona se prebacuje iz upravljackog registra u registar sekvence u poluciklusu obavljanja I to neposredno nakon poluciklusa dobavljanja u toku kog je dospela u upravljacki registar

Adresa inicijalnog mikro-programa – nalazi se u registru sekvence pre pocetka rada procesora

R – 0: procesor ne radi; **1:** procesor radi

R – 1: T se menja; **R – 0:** T – 0

Dekoder naredbe – transformise kod tipa naredbe u ulaznu adresu mikro-programa obavljanja

Poslednja mikro-naredba svakog mikro-programa obavljanja kao narednu adresu koristi ulaznu adresu mikro-programa dobavljanja

Sadržaj lokacija mikro-programске memorije je read-only

Prekidacke funkcije – sklop upravljacke jedinice koji implementira sve prekidacke funkcije (one menjaju stanja prekidaca)

U upravljacku jedinicu ulazi snop sa 16 linija koji polazi od registra naredbe; njegove 4 najznacajnije linije prenose kod tipa naredbe do dekodera naredbe. Adresa lokacije mikro-programске memorije dolazi ili iz dekodera ili iz registra sekvence; u registar sekvence u poluciklusu obavljanja se moze prebaciti sadrzaj 7 najmanje znacajnih bita upravljackog registra

R&T – kada je vrednost 1 u poluciklusu dobavljanja se vrsi citanje lokacije mikro-programске memorije

U ciklusu dobavljanja svake mikro-naredbe, upravljacki registar se puni sadrzajem lokacije mikro-programске memorije

4.6 UPRAVLJANJE PREKIDACIMA UPRAVLJACKE JEDINICE

Zajednicki argumenti prekidackih funkcija – R i T; od njih zavisi stanje prekidaca P55 koji puni registar sekvence

P55 – R&~T

P53 – zatvoren iza mikro-programa dobavljanja

P54 – zatvoren u svim poluciklusima dobavljanja

4.7 UPRAVLJANJE PREKIDACIMA VAN UPRAVLJACKE JEDINICE

Nista bitno

4.8 MASINSKI OBLICI MIKRO-PROGRAMA

Nista bitno

4.9 RAZMATRANJE RADA PROCESORA

Pre pocetka se sve anulira

Puni se upravljacki registar sadrzajem sa memorijske lokacije 0000000 (na taj nacin u upravljacki registar dospe masinski oblik jedine mikro-naredbe inicijalnog mikro-programa)

U upravljacki registar se smesti sadrzaj druge memorijske lokacije

Time se za obavljanje spremi prva mikro-naredba mikro-programa dobavljanja

Prvo izvorsavanje ovog mikro-programa – prebacivanje sadrzaja memorijske lokacije (adresa je u programskom brojacu) u registar naredbe

5.1 ORGANIZACIJA RACUNARA KONCEPT

Racunar koji ima samo procesor I memoriju je **neupotrebljiv**; mora se vrsiti izmena vrednosti spoljasnjim uticajem kako bi postao upotrebljiv

5.2 ULAZNI I IZLAZNI UREDJAJI

Ulazni – tastatura; upisuje se u **registar podataka** (kontrolera) tastature

Kontroler tastature – posreduje izmedju tastature I procesora

Registar stanja – omogucava razlikovanje kada se u registru podataka nalazi nepreuzet a kada vec preuzet kod znaka

Procesor samo **cita** iz registara tastature

Za **kontroler tastature** se podrazumeva da 8bitni kod generisan u tastaturi odlazi u 8 najmanje znacajnih bita **registra podataka** tastature

Izlazni – ekran; osvetljene ili neosvetljene tacke

Znakovna pozicija – pravougaona matrica tacaka koja sluzi za prikazivanje jednog znaka

Iz **koda znaka** se odredjuje koje tacke ce biti osvetljene

Kursor – poseban znak znakovne pozicije (kursor se replace-uje odredjenim znakom pri ispisu I pomera horizontalno u desno na sledecu poziciju)

Upravljacki znakovi - \n \r

Kontroler ekrana – posreduje izmedju ekrana I procesora

Registar podataka – kod znaka koji je poslednji upucen na prikazivanje

Registar stanja – pokazuje kada je prikazan znak ciji je kod upisan u registar podataka

Za **kontroler ekrana** se podrazumeva da se 8bitni kod koji ide ka ekranu, preuzima iz 8 najmanje znacajnih bita **registra podatka ekrana**.

Echo (eho) - prikaz znakova na ekranu; **editiranje** – unos I ispravljanje pogresno unetih znakova (**interaktivni rad**)

Terminal – tastatura + ekran

Interpretiranje komande – prepoznavanje vrste komande, izdvajanje argumenata I izvršavanje

Interpreter komandi – interpretira komande (jebemti zivot); prikazuje **prompt** kojim saopstava da je spreman da preuzme komandu

Drajver terminala – preuzima I prikazuje znakove; ima 2 potprograma I 2 operacije (**ulaznu I izlaznu**); ulazna preuzima znakove sa tastature a izlazna prikazuje znakove na ekranu

Ulazni bafer – memorijska zona u koju se odlazu preuzeti znakovi

Radno cekanje (busy waiting, polling) – proverava da li je ispunjen uslov neophodan za preuzimanje znaka (kasnije proverava ispunjenost uslova I kod rada sa memorijom)

Izlazni bafer – memorijska zona u koju se odlazu znakovi za ispis

Interpreter komandi + Drajver terminala = BIOS

Interaktivni nivo BIOSa – korisnik direktno koristi komande interpretera komandi

Programski nivo BIOSa – iz korisnickih programa se pozivaju operacije drajvera terminala

5.3 VRSTE MEMORIJE

RAM – poluprovodnicke; pisanje I citanje; gube sadrzaj pri nestanku napajanja

ROM - poluprovodnicke; citanje; ne gube sadrzaj pri nestanku napajanja

Disk – magnetne; pisanje I citanje; memorija u obliku koncentricnih kruzni staza koje glava diska opisuje; glava se pomera; staze su podeljene u sektore iste velicine; jedinica citanja I pisanja je blok

Mana RAMa – nepostojanost I mali kapacitet

Mana Diska – brzina

Radna memorija (poluprovodnicke) – brzina je priblizna brzini procesora; mali kapacitet

Masovna memorija (magnetne) – sporija od procesora ali je postojana I ima veliki kapacitet

Kontroler diska – posreduje izmedju diska I procesora

Drajver diska – brine da pristup registrima kontrolera diska bude u skladu sa pravilima njihove upotrebe; pojednostavljuje koriscenje; zadatak mu je da redni broj bloka pretvori u broj staze I broj sektora; sadrzi 2 potprograma (ulaz – citanje, izlaz – pisanje)

Prosireni interpreter komandi – uvode se dve nove naredbe (pored (p)isi, (c)itaj I (i)zvrši) koje sluze za prenos bloka (n)a disk I prenos bloka (s)a diska

Prosireni interpreter komandi + drajver diska + drajver terminala = BIOS

Procesor + radna memorija + terminal + masovna memorija + BIOS = KONCEPT

5.4 KODOVI ZA OTKRIVANJE I POPRAVKU GRESAKA

Bit provere – zajedno sa bitovima podataka obrazuje **kodnu rec**

Otkrivanje pogresnih bita se zasniva na proveru da li za bite podataka I bit prove vazi zadano pravilo

Bit parnosti – postavlja se tako da kodna rec sadrzi paran broj jedinica

Hamingova udaljenost – Broj razlicitih bita izmedju bilo koje dve kodne reci (ako je udaljenost n+1 onda je moguće otkriti do n pogresnih bita)

5.5 OPERATIVNI SISTEM

Datoteka – sadržaj u blokovima na masovnoj memoriji; pridružuje joj se ime

Proces – angazovanje procesora na izvršavanju programa

Modul za rukovanje datotekama – brine o tome u kojim blokovima je sadržaj datoteke I koliko je sadržaj dugacak; sadrži potprograme poput stvaranja I uništavanja datoteke, preuzimanja I izmene njenog sadržaja

Deskriptor datoteke – obuhvata **atribute datoteke** (velicina, redni brojevi blokova, vreme nastanka); modul za rukovanje datotekama rukuje deskriptorima datoteke

Modul za rukovanje procesima – vodi brigu o tome u kojim lokacijama radne memorije su masinski oblici naredbi programa; sadrži potprograme koji podržavaju operacije za rukovanje procesima poput stvaranja I uništenja procesa

Slika procesa – predstavlja masinski oblik izvršavanog programa kao I vrednosti promenljivih I sadržaj steka koji se menjaju u toku procesa

Deskriptor procesa – obuhvata **atribute procesa** (broj lokacija radne memorije za smestanje slike procesa, evidencija lokacija u koje je slika smestena); modul za rukovanje procesima rukuje deskriptorima procesa

Modul za rukovanje radnom memorijom – rukuje slobodnim memorijskim lokacijama; sadrži potprograme za zauzimanje I oslobađanje lokacija radne memorije

Izvršne datoteke – datoteke u kojima se nalaze masinski oblici programa od kojih nastaju slike procesa; sadrže **inicijalne slike procesa** u kojima se nalaze masinski oblici programa, ulazne adrese I dužine, podaci o promenljivima

Drajveri obrazuju **modul za rukovanje kontrolerima**

Modul za rukovanje (procesima + datotekama + radnom memorijom + kontrolerima) = Operativni sistem

Operativni sistem – objedinjuje raznorodne delove računara u skladnu celinu I sakriva od korisnika sve o funkcionisanju računara što nije bitno za njegovo korišćenje

Interaktivni nivo korišćenja operativnog sistema – korisnici koriste komande interpretera komandi operativnog sistema

Programski nivo korišćenja operativnog sistema – korisnički programi pozivaju operacije pojedinih modula operativnih sistema (**sistemske pozivi**)

Sistemske programi – **editor** (izmena teksta datoteka), **prevodilac** (asembler I kompajler; **izvorna** datoteka u **objektu** datoteku), **makro preprocesor**, **linker** (stvara izvršnu datoteku koja sadrži inicijalnu sliku procesa), **loader** (učestvuje u stvaranju slike procesa koja nastaje na osnovu inicijalne slike), **dibager** (omogućuje nadgledanje izvršavanja korisničkih programa)

Bootstrap loader (inicijalni loader) – služi za pokretanje operativnog sistema; pokreće se uz pomoć biosa; za njegov izvršni oblik se rezervise nulti blok (boot block); omogućuje odabir operativnog sistema za pokretanje

5.6 PREKLJUCIVANJE

Viseprocetni režim rada – bolji je od radnog čekanja; procesor se **preključuje** sa procesa čija aktivnost je postala zavisna od desavanja događaja van procesa na proces za čiju aktivnost jedino nedostaje procesor; samo jedan proces je u stanju **aktivan**, dok su ostali u stanju **čeka** ili **spreman**

Sistemske procese – njegov zadatak je da angazuje procesor kada nema drugih istovremeno postojećih procesa

Ulazom-izlazom vodjeno preključivanje – nastaje kada se na primer za nastavak očekuje unos sa tastature terminala

Modul za rukovanje procesorom – u njemu se nalazi operacija preključivanja

Registri opšte namene su jedini deljeni resurs

Registarski bafer – prostor u kome se smesta sadržaj registara opšte namene za taj proces

5.7 PREKIDI

Prekid – izaziva ga desavanje vanjskog događaja kao što je pritisak na tastaturu, prikazivanje znaka na ekranu ili kraj prenosa jednog bajta između diska i kontrolera; postoji više vrsta kao što su prekid tastature, prekid ekrana, prekid diska; svakoj vrsti pridružen je jedan obradivac prekida

Obradivac prekida – pokreće se u okviru prekida; ulazna adresa se naziva **vektor**; izvršavanje ovog potprograma se naziva obrada prekida

Broj vektora – jednoznačan broj koji kodira svaku vrstu prekida

Najava prekida – obavestava procesor o prekidu

Potvrda prekida – posebna linija koja služi kada procesor zahteva od kontrolera broj vektora

Tabela prekida – procesor koristi broj vektora kao indeks tabele prekida čiji elementi sadrže vektore obradivaca prekida; tako procesor dobija vektor obradivaca prekida; tabela se nalazi odmah posle BIOSa na prvim nekoliko mesta memorije

Bit prekida – peti bit status registra; pokazuje o(ne)mogućnost obrade prekida

Mikro-program prekida – sadrži precizan opis aktivnosti procesora u slučaju prekida; njegova ulazna adresa se nalazi u **registru prekida** upravljačke jedinice čiji je sadržaj stalan

Serijsko povezivanje – dodela prioriteta kontrolera se zasniva na serijskom povezivanju (kada dodje do istovremenih zatrazenja prekida)

Registar broja vektora – čuva broj vektora u svakom kontroleru

Prioritet procesa – koristi se za razlikovanje prioriteta procesa; podrazumeva se da je aktivnost najprioritetnijeg procesa najhitnija

Prekidi izazivaju podelu drajvera na **gornji** i **donji** deo; između njih se nalaze baferi namenjeni za razmenu podataka između njih

Gornji deo – sastoji se od potprograma koji podržavaju operacije namenjene slojevima postavljenim iznad sloja sa drajverima; ovakve operacije omogućuju znakovnu komunikaciju između korisnika i računara ili prenos blokova na disk i sa diska; prilikom izvršavanja ovih operacija dolazi do zaustavljanja aktivnosti procesa i do preključivanja ako nisu ispunjeni uslovi za nastavljanje pomenutih aktivnosti

Donji deo – sastoji se od obradivaca prekida okrenutih kontrolerima koji reaguju na desavanje vanjskih događaja; obrade prekide omogućuju nastavljanje prethodno zaustavljenih aktivnosti procesa i dovode do preključivanja procesora na neki od ovih procesa ako je on prioritetniji od prekinutog procesa

Background procesi – preuzimaju podatke od obradivaca prekida radi nastavljanja njihove obrade

Unos znakova unapred – ovo omogućuje ulazni bafer jer prispeli znakovi ostaju sve dok ih ne preuzme neki od procesa; ako je ulazni bafer prazan aktivnost ovih procesa se zaustavlja do njegovog punjenja

Aktivnosti procesora i kontrolera su **asinhroni**

Aktivnosti kontrolera i uređaja kojeg kontroler opslužuje su **sinhroni**

Handshaking – koristi se za sinhronizaciju kontrolera i uređaja

5.8 SABIRNICA

Sabirnica (magistrala)(bus) – vezna linija koja spaja dva dela računara

Direktan memorijski pristup – direktan kontakt kontrolera diska i radne memorije

DMA kontroler – kontroler diska koji je osposobljen za direktan memorijski pristup

Registri DMA kontrolera – **registar broja staze, registar broja sektora, registar broja bajta, registar adrese, registar stanja, registar podataka**

5.9 VISEKORISNICKI RAD

Sistemska vreme – izražava se kao broj prekida sata

Visekorisnicki rad – vise korisnika istovremeno koristi racunar; neophodan je sat (digitalni sklop sastavljen od kristalnog oscilatora koji generise impulse I brojaca tih impulsa)

Logicki adresni prostor – omogucuje zastitu medjusobnog ometanja korisnika I razdvajanje slika raznih procesa; sastoji se od **logickih** adresa koje se pretvaraju u **fizicke adrese**; slike procesa se I dalje nalaze u obicnom ili **fizickom adresnom prostoru**

Granicna adresa – najveca logicka adresa

Pretvaranje logicke adrese u fizicku se svodi na sabiranje logicke adrese sa **baznom adresom**

Granicni I bazni registar – cuvaju granicnu I baznu adresu

Komparator – implementira logicku funkciju V

Sabirac – sabira logicku I baznu adresu

Granicni registar + komparator + bazni registar + sabirac = Pretvaranje logicke u fizicku adresu

MMU – sklop koji pretvara logicku adresu u fizicku

Izuzetak (exception) – javlja se kada vazi V; ispravnost logicke adrese je tada neuspesna

Mikro-program izuzetka – pokretanjem njega procesor reaguje na izuzetak

Registar broja vektora – nalazi se u procesoru; procesor ga koristi da samostalno obezbedi broj vektora za izuzetak

Obradjivac izuzetka – njegova adresa se preuzima iz tabele prekida I smesta u programski brojac tokom izvršavanja mikro-programa izuzetka

Obrada izuzetka – njen zadatak je da zaustavi nedozvoljenu aktivnost procesa uz registrovanje mesta njene pojave

Privilegovani rezim rada (sudo) – naredbe kao sto su izmena sadrzaja granicnog I baznog registra

Korisnicki prostor – deo fizickog adresnog prostora u koji se preslikava logicki adresni prostor (neprivilegovani rad procesora)

Sistemske prostor – isto kao iznad samo za privilegovani rad procesora

Sistemske potprogrami obrazuju **sistemske biblioteku**; u sistemskim potprogramima se sakrivaju sistemske pozivi

6.1 EDITOR

Komandni I znakovni rezim (kao kod vima) – komandni: svaki unos su komande; znakovni: svaki unos je tekstualan I menja tekst

6.2 ASSEMBLER

Leksicka analiza – prepoznavanje ispravnih reci

Sintaksna analiza (parsiranje) – prepoznavanje ispravnih recenica odnosno redosleda reci po pravilima prog jezika

Leksicka (sintaksna) greska – pojava neocekivanog znaka; sledi oporavak od greske u vidu pronalazenja prvog znaka od kog se moze nastaviti asembliranje

Kod naredbi se preuzima iz **tabele naredbi**

Tabela labela – sadrzi labele I njihove adrese; dopunjava se kad naidje na **definiciju labele**

Problem kada se labela koristi pre njenog definisanja dolazi kod skokova koji se nazivaju

referenciranje unapred

Zbog ovoga se asembliranje organizuje u **dva prolaza** (u prvom se analizira tekst I popunjava tabela, u drugom se analizira jos jednom I generisu se masinski oblici naredbi I direktive)

Brojac lokacija – inicijalizuje se na nulu I sadrzi adresu pocetka prve naredbe ili direktive; uvecava se nakon nailaska na svaku sledecu

Objektna sekvenca – masinski oblik programa nastao u toku drugog prolaza asemblera; sacinjavaju je masinske naredbe

Semanticke greske – ponovno definisanje vec postojece labele; koriscenje nedefinisane labele

Ulazna tacka objektna sekvence – adresa ulazne masinske naredbe od koje zapocinje izvršavanje

6.3 MAKRO PRETPROCESOR

Prepoznaje makro definicije, pozive I uslovne direktive

Ime makro definicije se smesta u **tabelu makro imena**, njeno telo u **tabelu makro tela**, argumenti u **tabelu argumenata**

Semanticke greske koje se javljaju: poziv nedefinisiranog makroa, ponovno definisanje postojećeg makroa, neslaganje broja argumenata sa brojem parametra definicije

6.4 LINKER

Zadatak linkera je da zasebne objektno sekvence programa I potprograma **linkuje (poveže)** u jednu **izvrsnu sekvencu** programa radi stvaranja inicijalne slike procesa I njenog smestanja u izvrsnu datoteku

Relokacija – može da stvori probleme kada masinske naredbe iz pomerenih objektnih sekvenci sadrže u svojim dodatnim recima adrese, jer su te **apsolutne adrese** tačne samo pod pretpostavkom da svaka objektna sekvenca počinje od nulte logičke adrese

Problem relokacije linker može da reši korekcijom apsolutnih adresa uvećavanjem apsolutnih adresa za **konstantu relokacije**

Ovakva korekcija naziva se **statička relokacija**

Asembler mora da generise **tabelu relokacije** sa logičkim adresama lokacija objektno sekvence koje sadrže apsolutne adrese

Ovaj problem se može rešiti I uvođenjem **relativne adrese** iz koje se može odrediti apsolutna adresa

Spoljasnje reference – javljaju se kada se iz programa ili potprograma poziva zasebno asembliran program I tada se **referencira spoljasnja labela** koja nije definisana u programu ili potprogramu pozivaocu

Da bi linker mogao da koriguje adrese, assembler formira **tabelu nedefinisanih labela**

Da bi linker mogao da reši problem spoljasnjih referenci, assembler mu posredstvom objektno datoteke prosledjuje **tabelu ulaznih labela**

U toku rada linker preuzima objektno sekvence I sve tabele koje su obuhvacene linkovanjem I tada odredi adrese početaka sekvenci na osnovu njihovih dužina I formira **tabelu objektnih sekvenci**

Da bi linker rešio problem spoljasnjih referenci, sve tabele ulaznih labela se spajaju u jednu **tabelu spoljasnjih referenci**

Linker radi u 2 prolaza; prvi prolaz je namenjen za formiranje tabele objektnih sekvenci, relokaciju I formiranje tabele spoljasnjih labela; drugi prolaz je posvećen rešavanju problema relokacije I spoljasnjih referenci I stvaranju izvrsne sekvence

6.5 LOADER

Loader obavlja punjenje programa koje obuhvata prepisivanje njegove izvrsne sekvence iz izvrsne datoteke u radnu memoriju radi formiranja slike procesa

Postupak pretvaranja logičkih adresa u fizičke se naziva **dinamička relokacija**

6.6 DIBAGER

Oslanja se na **koracni režim rada** procesora; procesor se u ovom režimu rada nalazi kada poseban **bit traga** status registra sadrži vrednost 1

Za dibagiranje je zgodno da postoji I posebna **naredba zamke** čije izvršavanje se svodi na izvršavanje mikro-programa izuzetka

Umesto privremenih izmena izvrsnog oblika programa, dibager može da smesti adresu odabrane naredbe u neki od posebnih **dibagerskih registara**