1. 26. SISTEM PREKIDANJA. (engl. interrupt) omogućava da se izvršavanje tekućeg progr prekine zbog nekih događaja koji su nezavisni od programa. nakon izvršavanja takvih programa nastavlja se sa izvršavanjem prekinutog programa. kod računara koji poseduju sistem prekidanja svaki događaj koji je nezavisan od programa, a koji može imati uticaja na proces obrade, izaziva generisanje signala (signala prekida) koji procesoru daje informaciju o nastupanju događaja. signal dovodi do aktiviranja prekidnog pr ograma, nasuprot prekinutom programu koji se izvršavao do pojave signala prekida. istovremeno se moze pojaviti i više signala prekida pa su zato razvijeni sistemi prekidanja sa prioritetima- gde se pravi redosled u obradi signala prekida (poželjno je da korisnik može po želji da menja taj redosled). sistem prekidanja treba da omogući: organizaciju ulaska u prekidni program,izbor,

po određenim prioritetima, odgovarajućeg signala prekida koji će se obrađivati u datom trenutku,organizaciju povratka u prekinuti program i programsku (softversku) izmenu prioriteta.

2.27. ULAZNO IZLAZNI PODSISTEM. U/I podsistem se sastoji od skupa hardverskih i softverskih komponenti i namenjen je da obezbedi komunikaciju između spoljnjeg sveta (podsistem periferijskih jedinica) i centralnog dela računara (procesor, operativna memorija). pomocu njega se realizuju u i operacije na neki od načina:a) programirani prenos podataka sa pj-a gde procesor upravlja kompletnim prenosom, izvršavajući tako seriju u/i operacija. b)upravljanje prenosom podataka korišćenjem sistema prekidanja gde procesor stupa u interakciju sa okruženjem na osnovu signala prekida koje generišu pj-e. na ovaj se način poboljšava programirani ulaz/izlaz jer za prepoznavanje pj koja je pripremila podatke nije potrebno prozivanje pj-a.

3.28. U/I PROCESORI (U/I KANALI). u/i procesor (u/i kanal) obavlja sledeće funkcije: definisa nje oblasti memorije. to podrazumeva određivanje početne adrese

i veličine oblasti koja se koristi prilikom izvršavanja u/i operacija i koja se zadaje u standardnim informacionim jedinicama-bit, bajt, reč. ta dva podatka, formiraju upravljačku reč u/i kanala, koja za kanal ima istu ulogu koju za procesor igra instrukcija. u/i procesor mora da poseduje kumulativni brojač tekućih adresa, kao i brojač propuštenih podataka. o izvršavanju u/i operacija kanal obaveštava procesor tako što mu šalje dve vrste signala prekida :

programsko - upravljački signali prekida i signali prekida koji obaveštavaju procesor o završetku u/i operacije. programsko-upravljački signali prekida se planiraju programom i mogu se pojaviti u bilo kojoj etapi izvršavanja u/i operacije . signali prekida za obaveštavanje procesora o završetku u/i operacije mogu :da obaveste procesor o normalnom završetku operacije ili da, obaveste procesor da je došlo do nekih grešaka. kod računara koji poseduju u/i procesore upravljanje u/i operacijama se vrši pomoću sledećih informacija:u/i instrukcije, upravljačke reči u/i procesora, instrukcije periferijskih jedinica i kodovi (indikatori) stanja. upravljačku reč kanala zahvata iz memorije (koja može biti sopstvena ili pak rezrvisani deo operativne memorije) u/i procesor, dekodira je i izvršava.

4. 29. VRSTE U/I PROCESORA razlikuju se dva osnovna tipa u/i procesora: multipleksni i selektorski. multipleksni u/i procesori omogućuju »istovremeno« opsluživanje nekoliko periferijskih jedinica koje se nalaze u paralelnom radu. zajednički elementi kanala su ustvari registri (radni registri) i kombinacione logičke mreže koji omogućavaju izmenu tekućih parametara u/i operacija kao i razmenu podataka sa operativnom memorijom i pj-ama. selektorski kanal (u/i procesor). ovakav kanal je namenjen za privilegovano opsluživanje samo jedne pj-e. selektorski kanal bira upravljačku reč samo jednom na početku u/i operacije i odgovarajuće parametre zadržava u svojim registrima sve do izvršenja u/i operacije. kako ovaj kanal u datom intervalu opslužuje samo jednu pj-u može se reći da on ima samo jedan podkanal. selektorski kanal je namenjen opsluživanju brzih periferijskih jedinica (magnetske trake, magnetni diskovi i sl.) jer ne gubi vreme na pamćenje parametara u/i operacije, niti njihovo zahvatanje iz memorije.

5. 30. INTERFEJS. računarski sistem sadrži sledeće osnovne jedinice: procesore, module operativne memorije,ulazno-izlazne kanale,jedinice za upravljanje pj-ama (kontroleri pj),pj-e.ove jedinice su povezane unificiranim sistemom veza- sprežnim mrežama koje se još zovu i interfejsi. interfejs podleže određenoj standardizaciji, koja se odnosi na: format poruka koje se prenose preko interfejsa, naredbe koje interfejs prenosi, šeme veza interfejsa,algoritme funkcionisanja interfejsa, a i upravljačke signale koje u periodu veze između sebe izmenjuju jedinice računarskog sistema. interfejs predstavlja sveukupnost linija za predaju informacija, unificiranih elektronskih kola koja upravljaju prenosom signala po linijama, kao i algoritama za upravljanje razmenom poruka. interfejs karakterišu: vreme predaje poruka, izoblićenja prilikom predaje poruka, zahtev za strogo definisanim upravljačkim signalima.sreću se sledeći tipovi interfejsa:interfejs operativne memorije.interfejs procesor- kanal. ulazno – izlazni interfejs.

interfejs periferijske jedinice.

6. 31. ORGANIZACIJA LINIJA PREDAJE. linije interfejsa se mogu organizovati kao: sistem individualnih linija, kombinovani sistemi individualnih i zajedničkih linija

i

sistem zajedničkih linija. individualne linije su najpouzdanije jer kvarovi u ovim linijama ne utiču na rad drugih jedinica. one se koriste obično za ostvarivanje veza između računara i merno-regulacione opreme (merne tačke, izvršni organi i sl.). sistem sa individualnim i zajedničkim linijama .ovde se razmena vrši po zajedničkim linijama b , dok se upravljanje razmenom poruka vrši po individualnim linijama ai

i

ci.sistem zajedničkih linija je manje pouzdan od individualnih linija, ali je ekonomičniji. zajedničke linije su moguće u onim slučajevima kada razmena poruka između pojedinih jedinica podleže određenim pravilima. u interfejsu periferijskih jedinica koriste se zajedničke linije.

7. 32. PRIMERI INTERFEJSA. grupa linija interfejsa koja čini logičku grupu zove se magistrala. razlikuju se : ulazne magistrale, koje omogućavaju prenos poruka od procesora i / ili kanala do operativne memorije,

izlazne magistrale, koje služe za prenos poruka od operativne memorije do procesora odnosno kanala.pri čitanju sadržaja memorijske lokacije procesor putem ulazne magistrale šalje adresu memorijske ćelije, dok se preko izlazne magistrale sadržaj memorijske lokacije prenosi u procesor ili

u/i kanal .kada se radi o upisivanju sadržaja u memorijsku lokaciju onda procesor koristi ulaznu magistralu da pošalje adresu memorijske lokacije, a zatim preko iste magistrale šalje i odgovarajući sadržaj za tu lokaciju.u pogledu broja magistrala razlikuju se: jednostruki interfejs i višestruki interfejs.

8. 33. ZDRUŽENI INTERFEJS (MAGISTRALA, SABIRNICA). združeni interfejs (magistrala, sabirnica, engl. bus). sad jedan isti skup linija povezuje procesor, operativnu memoriju i pj-e. za sve razmene poruka između pojedinih jedinica koristi se skup standardizovanih signala i sve vrste razmena se vrše u sistemu podele vremena po linijama zajedničkog interfejsa.u svakom trenutku je jedna od dve jedinice koje su priključene na magistralu rukovodeća (komandna), dok je druga izvršna. svaka od priključenih jedinica može biti rukovodeća, izuzev modula operativne memorije koji može biti samo izvršna jedinica. uloge pojedinih jedinica

se u toku procesa rada neprekidno menjaju. linije veza u sklopu sabirnice se mogu podeliti na:linije namenjene za predaju poruka (podaci, adrese, upravljačke informacije), linije za odabiranje rukovodeće jedinice. za predaju poruka se koriste sledeće linije:

linije podataka, adresne linije, skup upravljačkih linija.

9. 34. PERIFERIJSKE JEDINICE. pj-e su podeljene u tri grupe: koje služe samo za ulaz podataka, kao što su:uređaji sa tastaturom, optički čitači dokumenata, ulaz jce za grafičke podatke, ulazne jce za obuhvat analognih signala. jedinice koje služe samo za izlaz podataka,jce koje služe i za ulaz i za izlaz podataka (masovne ili sekundarne memorije).postoje dva osnovna načina unošenja podataka u op mem računara: unose se direktno i podaci se prvo zapišu na medijum nosač podataka ,onda se preko ulazne jce unose u operativnu mem računara. načini zapisivanja podataka su : mehaničko, magnetsko, zapisivanje putem markiranja, zapisivanje nemehaničkim štampanjem, fotografsko zapisivanje, i aurealno zapisivanje. za unos podataka u rač tastature su još i danas nezaobilazne. pritiskom tastera na tastaturi generiše se binarna reč koja kodira informaciju pridruženu tom tasteru. najčešće su tastature koje generišu binarne reči kodirane u ascii kodu. za prikazivanje poruka se koriste video pokazivači i video ekrani. kod video ekrana se najčešće koristi katodna cev (tv ekran). slika se sastoji od određenog broja linija , a svaku liniju sačinjava određeni skup tačaka.

10. 35. ULAZ/ IZLAZ ANALOGNIH SIGNALA. neke od oblasti primene obrade analognih signala su:analiza govora,instrumentacija (merni sistemi),-analiza mernih signala u različitim oblastima, grafičko prikazivanje rezultata obrade, automatsko upravljanje, i sl. upotreba digitalnih signala za obradu analognih signala rezultira sa sledećim:ulaz signala u računar je moguć samo u digitalnom obliku, pa se u računar unose samo uzorci analognog signala,zbog korišćenja binarnih reči konačne dužine, postoji konačan broj kodnih reči za predstavljanje određenih analognih vrednosti, odnosno nekoj određenoj vrednosti analogne veličine dodeljuje se kodna reč koja kodira analognu vrednost koja najbliže odgovara datoj analognoj vrednosti, što dovodi do tzv. greške kvantizacije .

11. 36. POBOLJSANJA ARHITEKTURE VON – NEUMANN – A. osnovni model računarskog sistema (naziva se i von neumann-ov model) i poseduje sledeće osnovne funkcionalne delove:centralni procesor,memorijski podsistem i ulazno/ izlazni podsistem.kod ovog modela se sekvencijalno izvršavaju instrukcije koje su , zajedno sa podacima, smeštene u memorijskom podsistemu. radni režim se s
astoji od ponavljanja dve osnovne faze (faze pripreme – kada procesor zahvata instrukciju iz memorije i faze izvršenja). postoje sledeće tri osnovne arhitekture: arhitektura kod koje se razmena upravljačkih signala vrši preko zajedničke magistrale, arhitektura gde se razmena upravljačkih signala vrši preko centralnog procesora i arhitektura kod koje se razmena upravljačkih signala vrši preko memorije.

12. 37. RAČUNARSKI SISTEMI SA PARALELNOM OBRADOM. računarski sistemi sa paralelnom obradom se mogu podeliti na:čvrsto spregnute sisteme i slabo spregnute (distribuirane) sisteme. pri tome se pod stepenom sprege podrazumeva relativan odnos između količine lokalno obrađenih podataka i količine razmenjenih podataka.smatra

se da su dve jedinice čvrsto spregnute ukoliko su količine lokalno obrađenih i razmenjenih podataka približno iste. čvrsto spregnuti računarski sistemi

sastoje se od više procesorskih jedinica koje međusobno dele iste resurse (npr. memoriju, u/i podsistem i sl.). jedna od mogućih klasifikacija čvrsto spregnutih računarskih sistema obuhvata:sisd sisteme, simd sisteme,misd sisteme i mimd sisteme.sisd sisteme (single instruction-single data stream – jedan tok instrukcija i jedan tok podataka) simd sisteme (single instruction- multiple data stream – jedan tok instrukcija i više tokova podataka) sačinjava jedna

upravljačka jedinica, više procesora, više memorijskih podsistema i sprežnih mreža koje to sve povezuju. slabo spregnuti (distribuirani) računarski sistemi formiraju se od dva ili više autonomnih računarskih sistema koji su međusobno povezani komunikacionim mrežama za prenos podataka.

13. 38. MULTIPROCESORSKI RAČUNARSKI SISTEMI.

multiprocesorske arhitekture poseduju dva ili više procesora približno istih mogućnosti koji mogu da pristupe zajedničkoj memoriji i zajedničkim u/i jedinicama. rad procesora koordinira sistemski softver korišćenjem zajedničke memorije.razmenom upravljačkih informacija i razmenom podataka obavlja se koordinacija rada procesora. u logičkom pogledu postoje dva načina raspodele upravljačkih zadataka: vertikalna (hijerarhijska) raspodela – gde su procesori na nižem hijerarhijskom nivou podređeni procesorima na višem nivou hijerarhije, i -horizontalna (ravnopravna) raspodela – gde su svi procesori ravnopravni. razmena podataka između procesora može biti: -indirektna – kada postoji jedna centralizovana jedinica (to je obično zajednička memorija) preko koje se razmenjuju svi podaci, i-direktna – ostvaruje se preko magistrala koje spajaju procesore.prema načinu povezivanja procesora razlikuju se sledeće tri osnovne klase multiprocesorskih sistema:-sistemi sa zajedničkom magistralom, sistemi sa poprečnom matricom (engl. cross-bar),sistemi sa multipristupnom memorijom.

14. 39. DISTRIBUIRANI RAČUNARSKI SISTEMI.

distribuirani računarski sistemi podrazumevaju da se jedan posao (zadatak) izvršava na više računara paralelno. to se postiže sa pojavom lokalnih računarskih mreža. najčešće korišćeni distribuirani sistemi u dosadašnjem periodu su: hijerarhijski model. na najnižem hijerarhijskom nivou koriste se sistemi za akviziciju (prikupljanje) mernih podataka i lokalni kontroleri u realnom vremenu. na sledećem nivou se vrši analiza prikupljenih podataka i proverava radno stanje sistema kojim se vrši upravljanje. na najvišem hijerarhijskom nivou vrši se upravljanje objektnim sistemom u celini; federativni model – može se predstaviti kao jedan centralizovani računarski sistem (engl. host computer) i više distribuiranih miniračunarskih sistema u

odnosu nadređeni/ podređeni sistemi (engl. master slave). homogeni model – pretpostavlja da se distribuirani računarski sistem sastoji od skupa računarskih sistema približno istih karakteristika. obrade.

model toka podataka-ovde se obrade vrše tako što su izlazni rezultati dobijeni na jednom procesoru ulazni podaci za naredni procesor, tako da se ostvaruje tok podataka kroz sistem.

15. 40. TEHNOLOGIJE U IZRADI MIKROPROCESORA. mikroprocesor nastaje tako što se ceo procesor realizuje na jednom poluprovodničkom uzorku (čipu). jedna od mogućih klasifikacija p-a, zavisno od tipa poluprovodničke tehnologije, je da se izdvajaju dva glavna razreda i to: mikroprocesori realizovani u tehnologiji mos, i mikroprocesori realizovani u bipolarnoj tehnologiji. većina današnjih standardnih p-a je napravljena u tehnologiji mos (metal oxide somiconductor)- zbog jednostavnije izrade, lakše integracije mnoštva komponenti, kao i manje potrošnje. najvažniji podrazredi mos tehnologije su: p-kanalni mos (pmos), n-kanalni mos (nmos), komplemen
tarni mos (cmos). prednost bipolarne tehnologije je u brzini, ali se nedostaci ogledaju u visokoj potrošnji, kao i niskom nivou gustoće integracije.

16. 41. OPIS KOMPONENTI MIKRO. RAČ. SISTEMA. obično se koriste tri načina za opis komponenti koje ulaze u sastav mikroračunarskog sistema (mikroprocesor i prateće komponente-prateća kola, kontroleri, pomoćni procesori, memorijske jedinice, izlazno/ulazne jedinice,...): opis ulaznih i izlaznih
signala, opis mikroinstrukcija i opis signala u vremenskom domenu. kada se neka komponenta opisuje potrebno je da se navedu:

naziv signala koji se prenosi preko izvoda, funkcija signala, broj izvoda (nožice) preko koga se signal prenosi, da li je signal ulazni, izlazni ili ulazno/izlazni, aktivni logički nivo signala, ako on aktivira neki događaj, električne osobine: naponski nivo, da li se može izvod za koji je vezan prevesti u stanje visoke impedanse, šta treba uraditi ako se signal ne koristi,

vremenske osobine:zahtevano trajanje signala, trenutak generisanja ili prihvatanja signala u odnosu na neki referentni vremenski trenutak

17. 42. UNUTRAŠNJA ORGANIZACIJA MIKROPROCESORA. akumulator (a) je osnovni i glavni registar kod svakog računara, pa i kod mikroprocesora. aritmetičko-logička jedinica

(alj) izvršava aritmetičke i logičke operacije .statusni registri (ili indikatorski registri, engl. program status word-psw) čuvaju posebne informacije o stanju mikroprocesora. registri opšte namene , kao i kod procesora, služe za ubrzavanje operacija u alj mikroprocesora, jer smanjuju potrebu za stalnim pristupanjem operativnoj memoriji.

prelazak upravljačke jedinice iz jednog u drugo stanje obavlja se pod dejstvom sinhronizacionog signala ili signala takta (clock). naredno stanje nije uvek jednoznačno određeno već može da zavisi od vrednosti izabranog indikatorskog bita.

18. 43. MODEL MIKROPROCESORA. mikroprocesor obavlja sledeće zadatke:-upravlja svim delovima mikroprocesora kao i ostalim jedinicama mikroračunarskog sistema,

daje podršku izvršavanju instrukcija koje su smeštene u memoriji, koja se ogleda, pre svega u određivanju redosleda i interpretaciji instrukcija,izvršavanje operacija koje su definisane ovim instrukcijama, pruža podršku drugim mehanizmima kao što je prekid i slično.

prenos podataka između delova mikroprocesora se odvija preko zajedničkog snopa linija – interne magistrale. time se podržava izvršavanje različitih mikrooperacija za prenos podataka sa spoljnjeg snopa linija za podatke, kroz bafere do svih registara: programskog brojača,

instrukcijskog i indikatorskog registra i registara opšte namene. adresa koja se prenosi kroz bafere bira se multiplekserom.svaka od navedenih jedinica ima svoje signale koji. neophodno je da upravljačka jedinica definiše signale za upravljanje :baferima, multiplekserom za izbor adrese, spoljnjim jedinicama i prihvatnim registrom.

19. 44. MIKRORAČUNARSKI SISTEMI.

mikroračunarski sistemi predstavljaju računarske sisteme izgrađene na bazi mikroprocesora i ostalih mikromodula koji su svi (izuzev ram memorije) realizovani na po jednom čipu. osnovna ideja u formiranju mikroračunarskog sistema jeste da se sve njegove jedinice povežu preko tri snopa linija (magistrala): za podatke, adrese, i upravljačke signale. kod povezivanja komponenti u mikroračunarskom sistemu treba voditi računa o sledećem: svaka komponenta mora imati jedinstvenu i jednoznačno određenu adresu, i u jednom trenutku samo jedna jedinica može da vrši operaciju upisa na snop linije. neke ulazne i izlazne jedinice

ne mogu se direktno povezati sa mikroprocesorom zbog toga što ne postoji fizičko i logičko prilagođenje signala. da bi se to povezivanje ostvarilo proizvode se specijalizovana kola – kontroleri , koja imaju ulogu posrednika između ulazno/izlaznih jedinica i mikroprocesora. mikromodul pic (od engl. programable

interrupt controler- programabilni kontroler prekida) omogućava proširenje priključenja broja izvora prekida , kao i određivanje prioriteta signalima prekida. dma je kontroler za direktan memorijski pristup.rom – stalna memorija koja sadrži sistemske programe koji se najčešće koriste. ram – promenljiva memorija sa slučajnim pristupom koja se koristi kao operativna memorija .

45. MEMORIJE KOD MIKROPROCESORA.

programi (instrukcije), podaci , rezultati i međurezultati se zapisuju u memoriju i pozivaju iz memorije. bistabili (elementi sa dva stabilna stanja) su osnovni elementi za konstrukciju mem. niz bistabila je registar (obično 8, 16, 32), gde se zapisuju razni podaci. memorije se kod mikroračunara mogu podeliti u dva razreda : 1) sa sekvencijalnim pristupom- vreme pristupa zavisi od mesta gde je podatak sacuvan. 2) sa direktnim pristupom- vreme pristupa je uvek isto, bez obzira na lokaciju podatka.

ovde spadaju upisno-ispisne memorije (ram) i ispisne memorije (rom). broj binarnih cifara koje mogu da se upišu u jednu ćeliju predstavlja kapacitet ćelije. ćeliji se pristupa na osnovu adrese memorijske ćelije. svaki od registara u mem mora imati adresu pomoću koje se ta lokacija može

adresirati radi upisa ili čitanja podataka. da bi to bilo moguće moraju postojati još dva registra koji sa memorijom čine celinu. to su: adresni registar u kome se privremeno memorišu binarno kodirane adrese; i registar podataka u koji se prenose podaci koji se žele upisati u memoriju ili pročitati iz nje.

46. ROM MEMORIJE. memoriju rom odlikuju neizbrisivost i nedestruktibilnost-što znači da se rom pojavljuje kao memorijsko polje čiji je sadržaj jednom upisan i ne može se promeniti pod uticajem mikroprocesora. u rom je upisan određeni sadržaj već u vreme izrade memorijskog čipa. memorija ima oblik matrice. na njenim mestima ukrštanja se nalaze, ili ne nalaze, diode ili tranzistori. za rom memorije velikog kapaciteta se koriste mos fet tranzistori u tehnici lsi i vlsi integrisanih kola. u grupi ispisnih memorija se nalaze:rom, prom – programabilne rom memorije i epprom i eeprom- izbrisive prom memorije. prom je tip ispisne memorije koju može da isprogramira sam korisnik uz pomoć uređaja za programiranje prom-ova. postoje dva tipa programabilnih rom-ova:-bipolarna polja dioda i -bipolarna tranzistorska konfiguracija. eprom (engl. errasable prom) je memorija koju može programirati korisnik uz pomoć eprom programatora, ali se njen sadržaj može izbrisati, a zatim ponovo isprogramirati. eprom se briše osvetljavanjem čipa sa određenim ultraviolentnim zrakom (uveprom), ili pod dejstvom električnog polja (eeprom,

engl. elecrticaly errasable prom).kod eeprom memorije je moguće i delimično brisanje sadržaja.

47. RAM MEMORIJE. ove se memorije proizvode sa statičkim ćelijama (statičke ram memorije) i sa dinamičkim memorijskim ćelijama (dinamičke ram memorije).kod statičkog ram-a osnovna memorijska čelija u matrici je flip-flop čiji je sadržaj statičan, tj. veličina signala memorijskog podatka se ne menja tokom memorisanja. u dinamičkim ram memorijama informacije se memorišu u obliku električnog naboja koji se vremenom smanjuje pa je sadržaj takvog memorijskog elememta potrebno osvežavati (refresh) periodično. flip-flopovi omogućuju nedestruktivno, višestruko očitavanje, dok je čitanje dinamičke ram memorije destruktivno- po njenom očitavanju sadržaj se gubi, pa ga je potrebno ponovo upisivati. ram memorije mogu imati:-linearno adresiranje (ili adresiranje sa jednodimenzionalnom adresom) ili -koincidentno adresiranje (ili adresiranje sa dvodimenzionalnom adresom). tehnologije koje se koriste za izradu poluprovodničkih ram memorija su :-bipolarne, i -mos.

48. U/I MEDJUSKLOP KOD MIKROPROCESORA. zadatak u/i međusklopa je da obezbedi hardversku i softversku kompatibilnost dva mikrač ili mikrač i perifernih uređaja. mikroprocesor je procesor izrađen u lsi tehnologiji. on je deo mikroračunara, koga ćine još i:operativna (radna) memorija (ram,rom,prom), u/i (ulazno/izlazni) međusklopovi, pomoćni sklopovi (generator takta,...), sabirnice i monitorski progr. mikrač sistem se dobija kada se tu jos dodaju : periferni uređaji (prikazna jedinica, disk jedinica, štampači...) i progr podrška (operativni sistem, korisnički programi,...). u/i međusklop čine sledeće komponente: konektori, kabl, elektronski sklopovi. u/i međuskl mogu biti: paralelni u/i međusklopovi, serijski u/i međuskl i međuskl za posebne namene. serijski prenos podataka može biti: asinhroni i sinhroni .asinhroni prenos ne podrazmeva prenos taktnih impulsa linijom. svaki bajt se stavlja u »okvir« koga čine start i stop bit. sinhroni prenos zahteva liniju sa taktnim impulsima. paralelni u/i međuskl ima ulogu da poveže rač sa bajtno orijentisanim spoljnjim sklopovima. obično sadrži dvoje ili više vrata,upravljačke registre i upravljačku logiku. međureg podataka, koji se nalazi između interne sabirnice mikrač i interne sabirnice međuskl, poseduje 3 moguća stanja. kada rač ne saobraća sa međuskl, međureg je u stanju visoke impedanse. serijski u/i međuskl ima dve osn uloge: da pretvara paralelne oblike inf u serijske i obrnuto, i da obezbedi unošenje dodatnih bitova za prenos (start bit, bit pariteta, stop bit, bitovi za sinhronizaciju pri sinhronom prenosu).

49. PRINCIP RADA MIKROPROCESORA.

osnovni zadatak p-a je da izvršava instrukcije koje definišu operacije koje treba izvršiti nad podacima da bi se rešio neki problem. između p-a i memorije postoji stalna razmena informacija (instrukcije, adrese i podaci-operandi i rezultati). ako treba izvršiti neku operaciju nad nekim operandom, on se iz memorije prenosi u p gde se izvršava operacija, a dobijeni rezultat se prenosi u memoriju. mehanizam za određivanje adresa instrukcija koristi dva registra: programski brojač (pc) i instrukcijski registar (ir). registar pc realizuje se u obliku binarnog brojača sa paralelnim upisom ulazne adrese i upisom nule. uvodi se konvencija da je jedan od operanada uvek smešten u određenom registru i da se rezultat upisuje u isti taj registar. ovaj registar se naziva akumulator (a) i on je spregnut sa alj. mikrooperacije u pripremnoj fazi su identične za sve instrukcije i one su : ir m(pc)

-

prenos koda operacije iz memorijske lokacije sa adresom koja je u programskom brojaču pc u instrukcijski registar ir. pc pc+1

-

povećanje sadržaja programskog brojača za 1. instrukcije se predstavljaju simboličkim oznakama. primer. move

r1 , 88

; prenos operanda 88 iz memorije u registar r1, shl

r1

; pomeranje r1 za 1 mesto ulevo.

50. SINHRONIZACIJA MIKROPROC. SA DRUGIM KOMPONENTAMA.

prilagođavanje unutrašnjih signala p-a signalima drugih komponenti nije dovoljno izvršiti samo po fizičkim i logičkim nivoima, već je neohodno da se obezbedi sinhronizacija između jedinica koje razmenjuju podatke- a to obavlja upravljačka jedinica p-a. ovde će se posmatrati dva problema koji su s tim u vezi i to: a) -sinhronizacija po brzini prenosa i b)-sinhronizacija po pravu pristupa snopovima linija mikroračunarskog sistema.a) najjednostavnija situacija u toku prenosa podataka je kada su sve komponente mikroračunarskog sistema međusobno kompatibilne, b) često različite komponente mogu da koriste snopove linija za razmenu podataka sa drugim komponentama. otuda je neophodno da p može da prepusti upravljanje sa snopovima linija nekoj drugoj komponenti mikroračunarskog sistema. to se postiže korišćenjem dva dodatna signala p-a: -signal zahteva za pristup snopovima linija (engl. bus request- busrq) i -signal dozvole pristupa snopovima linija (engl. bus acknowledge- busa). komponenta koja ima pravo pristupa može da: -definiše signale na adresnim linijama,-definiše izlazne upravljačke signale, -prihvata i interpretira ulazne upravljačke signale.

51. STEK.

ako se u mikrač sist realizuje mem kod koje se podaci čitaju redosledom koji je suprotan sa redosledom upisa onda to omogućava meh koji se zove stek. radi se principu lifo (last in first out) što znači da podatak koji je poslednji upisan,a prvi se čita iz memorije. podatak koji je prvi upisan fizički je smešten u početnu lokaciju stek mem koji se naziva dno stek mem,a poslednji je smešten u lokaciji sa nazivom vrh stek mem. tokom rada p-a podaci se dinamički upisuju u stek i čitaju iz njega, tako da ta lista u steku stalno menja svoju dužinu. upravljanje stekom omogućava poseban registar- pokazivač steka ili sp. ovaj registar uvek sadrži adresu prve slobodne lokacije u stek mem, a to je lokacija koja se nalazi iza vrha steka. prilikom inicijalizacije mikrač sist adresa dna steka se upiše u sp kao početna vrednost i time se definiše zona stek mem. oznaka push znaci upis sadrzaj registra na stek a pop znaci uzimanje sa steka.

52. POZIV POTPROGRAMA I POVRATAK U GLAVNI PROGRAM.

za poziv potprograma, kao i povratak u glavni program, koristi se mehanizam steka. glavni problem kod pozivanja potprograma je da se zapamti adresa povratka, odnosno adresa instrukcije glavnog programa koja treba da se izvrši kada se kontrola izvršenja vrati iz potprograma u glavni program. problem se uz pomoć steka rešava lako: kod poziva potprograma adresa povratka se upiše u stek, a kod povratka se ta adresa uzima iz steka. tako se realizuju složeni pozivi potprograma, pozivi drugih potprograma iz pozvanog potprograma, pa i rekurzivni pozivi (mogućnost da potprogram poziva samog sebe). poziv potprograma se izvršava instrukcijom: call adresa potprograma koja se u memoriji smešta u dve (ili tri) memorijske lokacije. povratak iz potprograma u glavni program obavlja se naredbom

return . da bi mikroračunarski sistem pravilno radio potrebno je da se naredbe koje se odnose na rad sa stekom koriste u paru:

push – pop

i

call –return.

53. MEHANIZAM PREKIDA

omogućava da mp, pod dejstvom spoljnjeg događaja, prekine izvršavanje tekućeg progr i pređe na izvršavanje progr za obradu tog događaja. u opštem slučaju, prekid se koristi kada p izvršava dva zadatka: 1) sa nižim prioritetom koji se obavlja kontinualno, i 2) koji ima viši prioritet i obrađuje se povremeno, pod dejstvom nekog spoljnjeg događaja.zato postoje dva progr- tekući ili prekinuti progr, i drugi, tzv. prekidni progr ili progr za obradu prekida , namenjen zadatku sa višim prioritetom. spoljnji događaj izaziva nastanak signala prekida(interrupt) koji dovodi do prekida izvršavanja progr nižeg prioriteta. kod meh prek se koriste 2 signala: ulazni int javlja da se desio događaj vezan za prekid, i izlazni inta (interrupt acknowledge – prekid prihvaćen). početna adresa prekidnog progr je, ako postoji samo jedan prekidni prog, fiksna i unapred poznata. ako postoji više jca,koristi se tzv. vektorski prekid za određivanje početne adrese prekidnog progr. povratak u tekući progr vrsi instrukcija reti. svaka pj koja učestvuje u lancu prekida poseduje jedan stepen iz tog podsistema. stepen za lančanje prekida poseduje sledeće signale: ulazni - za dozvolu generisanja signala prekida int. ieo - izlazni signal, int-izlazni signal za prekid,

isp

- interni signal prekida.. osnovna ideja u lančanju prekida je da se pj koje generišu prekid povežu po opadajućem redosledu prioriteta tako da jedinica višeg prioriteta može prekinuti prog za obradu prekida jedinice nižeg prioriteta, dok obrnuto nije moguće.

54. ZABRANA PREKIDA

u nekim slučajevima je neophodno da se spreči prelazak p-a u stanje prekida (da bi u tom trenutku sistem radio korektno). zabranu prekida omogućavaju ili kolo i r-s flip-flop sa sledećim ulazno-izlaznim signalima: int – spoljnji ulazni signal prekida,isp – interni signal prekida koji vodi ka upravljačkoj jedinici, m – izlaz iz flip.flopa koji kontroliše prolazak signala

int do upravljačke jedinice, set – ulaz flip-flopa koji prevodi izlaz m u stanje logičke 1, reset – ulaz flip-flopa koji prevodi izlaz m u stanje logičke 0, di – signal za zabranu prekida, ei – signal za dozvolu prekida. prolazak signala prekida int kroz ili kolo zavisi od logičkog stanja signala m: m = 0 , signal int prolazi kroz kolo, odnosno, ako je int na aktivnom nivou, onda je isp aktivan, što znači da je prekid dozvoljen; m = 1, signal

int ne prolazi kroz

ili kolo, odnosno

isp = 1 bez obzira na logičko stanje signala

int, što znači da je prekid zabranjen. stanje flip-flopa se kontroliše softverski, instrukcijama di i ei

koje generišu pozitivne impulse na ulazima flip-flopa koje prevode u stanje logičke 0 ili 1.

56. klasifikacija prekida

prekidi se obično dele na spoljnje, unutrašnje i softverske. spoljnji prekidi se generišu preko spoljnjeg signala prekida koji nastaje na osnovu nekog događaja koji se dogodio izvan p-a. sličan mu je i unutrašnji prekid – kao posledica nekog događaja unutar p-a koji ima prioritet u obradi u odnosu na tekući progr. softverski prekid se uvodi u cilju povišenja stepena zaštite u mikroračunarskom sistemu koji treba

korektno da radi i kada korisnik uradi nešto pogrešno. uvode se dva moguća generalisana stanja p-a: korisničko i privilegovano. u prvom se ne mogu izvršiti sve instrukcije koje p poseduje i ne može se pristupiti svim resursima mikroračunarskog sistema. u korisničkom stanju se izvršava samo podskup od akcija koje sistem može da obavi, dok akcije koje su korisniku nedostupne obavljaju rutine koje su pod kontrolom operativnog sistema. problem kod ovog pristupa je što su korisniku često potrebne akcije koje su zabranjene u korisničkom stanju

. to se rešava tako što se određena rutina operativnog

sistema aktivira signalom prekida. naime, korisnik može da radi samo u korisničkom stanju, a rutine koje su mu potrebne mogu se izvršiti samo kao odziv na signal prekida. da bi korisnik

mogao da samostalno inicira izvršavanje neke rutine u privilegovanom režimu rada, mora da ima mogućnost da simulira prekid i to je osnovni razlog za uvođenje softverskog prekida- kako bi se omogućio prelazak iz korisničkog u privilegovani režim rada. softverski prekid se realizuje kao jedna posebna instrukcija koja se eksplicitno navodi na željenom mestu u programu, a odziv p-a na ovu instrukciju je identičan odzivu na signal prekida.

57. PODRŠKA TESTIRANJU PROGRAMA

 p-i poseduju mehanizme koji obezbeđuju hardversko- softversku podršku otkrivanju i otklanjanju grešaka a to su: izvršavanje programa korak po korak, tačke prekida i trag programa. izvršavanje programa korak po korak se sastoji u tome da p izvrši jednu instrukciju programa koji se testira, a zatim se aktivira poseban program koji kroz interakciju sa korisnikom omogućava uvid u sadržaje registara p-a i memorije. korisnik može da menja sadržaje registara i memorijskih lokacija. ovaj metod je pogodan kod manjih progr. tačke prekida su pogodne za testiranje i otklanjanje grešaka u programima koji imaju petlje, a u kojima se koraci u petlji ponavljaju veći broj puta. tačka prekida obezbeđuje da p bez zaustavljanja izvrši instrukcije koje se nalaze pre tačke prekida i da se zaustavi u tački prekida, gde se aktivira progr koji omogućava korisniku iste usluge kao za mehanizam »korak po korak«. trag progr je pogodan za testiranje i otklanjanje grešaka u progr koji obezbeđuju rad mikroračunarskog sistema u realnom vremenu. trag progr je hardversko-softverski mehanizam koji omogućava da se neke informacije koje su bitne za testiranje progr smeštaju u poseban fajl u toku izvršavanja progr koji se testira. kada se završi progr, korisnik proverava sadržaj fajla i tako prati kako je program izvršavan. ovako se program nesmetano izvršava, a informacije vezane za izvršavanje progr se čuvaju za naknadnu analizu. navedeni postupci za otkrivanje greški u programu se zasnivaju na dva mehanizma:-unutrašnji prekid za trag programa (engl. trace trap) i -prekidna tačka. unutrašnji prekid se aktivira po završetku instrukcije. ovaj se prekid može softverski maskirati, odnosno dozvoliti ili zabraniti. bit t (od trap) za dozvolu unutrašnjeg prekida za trag progr obično je u sastavu indikatorskog registra. ako je unutrašnji prekid za trag programa dozvoljen, završetak instrukcije generiše unutrašnji prekid, kojim se inicira izvršenje posebnog programa. ovaj program (engl. debugger) u interaktivnom radu omogućava korisniku da na ekranu vidi sadržaje svih registara i memorijskih lokacija i da menja ove sadržaje. prekidna tačka se ne može postaviti ako se program koji se testira nalazi u memoriji tipa rom, jer ova memorija ne dozvoljava izmenu sadržaja memorijskih lokacija na način koji se primenjuje za prekidnu tačku. takođe, prekidna tačka može da se postavi samo na kod operacije

neke instrukcije, a ne može da se postavi na lokaciju u kojoj je smešten operand ili neka druga informacija.

58. KONTROLER ZA PARALELNI ULAZ/IZLAZ

kod realizacije mikrorač sistema pogodno je da se koriste kontroleri- specijalizovane jedinice koje se spajaju sa p-om preko međusobno kompatibilnih signala. kontroler za paralelni ulaz/izlaz podržava razmenu memorijskih reči između p-a i neke pj-e.neposredno sprezanje p-a i pj-e zahteva da p proverava prenos podataka, spremnost jedinice za prijem ili predavanje podataka i sl. i zbog toga to nije pogodno. a dobro je što kontroler preuzima sve upravljačke funkcije vezane za pj-u. ovaj je kontroler , po pravilu, programabilan, što znači da u njemu postoje upravljački registri u koje p upisuje upravljačke informacije kako bi se izabrali odgovarajući radni parametri koji su specifični za određenu pj-u. upravljačke informacije se, najčešće, odnose na: smer prenosa podataka, način sinhronizacije sa pj-om, mogućnost generisanja prekida i sl. kod izlaza se podaci prenose iz ulazno/izlaznog registra prema pj, a kod ulaza se podaci prenose iz pj-e u ulazno/izlazno kontroler. sinhronizacija kontrolera sa pj-om jedinicom se vrši signalima ready i strobe koji za primer izlaza podataka (prenosa od p-a ka pj-i) imaju sledeća značenja: ready = 1,

na linijama za prenos podataka između kontrolera i pj-e se nalazi stari podatak (koji je pj već primila i obradila); ready =0 , na linijama za prenos podataka između kontrolera i pj-e nalazi novi podatak (koji pj nije obradila); strobe = 0,

pj nije spremna da prihvati novi podatak; strobe = 1,

pj je spremna da prihvati novi podatak. postupak sinhronizacije pomoću navedena dva signala (koji se naziva i rukovanje) odvija se na sledeći način: izlaz podatka preko kontrolera prema pj-i započinje operacijom upisa kojom p upisuje izlazni podatak u ulazno/izlazni registar kontrolera. dekodiranjem adresnih signala generiše se signal cs0 kojim se obavlja to upisivanje. taj podatak je novi podatak koji treba preneti pj-i.

od tog trenutka p obavlja neke druge operacije i više ne učestvuje u sinhronizaciji sa pj-om. podatak koji se do tog trenutka nalazio na linijama podataka između kontrolera i pj-e je stari (već obrađeni) podatak. izlazne linije ulazno/izlaznog registra vezane na izlazne linije za

podatke kontrolera tako da će se upisani izlazni podatak odmah naći na izlaznim linijama. kada p upiše novi podatak u ulazno/ izlazni registar, upravljačka jedinica kontrolera prevede signal ready na jedinicu i tako obaveštava pj-u da je na njenim linijama za podatke novi podatak. pj detektuje ovo i prevodi signal strobe na nulu sto znaci da je započela (ali još nije završila) operaciju prijema i obrade novog podatka. završetak ove obrade pj javlja prevođenjem signala strobe na jedinicu i od tog trenutka podatak na linijama se naziva stari podatak. kontroler onda prevodi signal ready na 0 čime se završava ciklus izlaza podatka, svi signali su dovedeni na početno stanje pa treba javiti p-u da upiše sledeći podatak. to se vrsi tako sto kontroler generiše signal prekida int.

59. KONTROLER ZA DIREKTAN PRISTUP MEMORIJI

osnovna ideja mehanizma za direktan pristup memoriji (dma) je da se prenos podataka između kontrolera i mem obavlja bez posredovanja p-a. mehanizam dma zasniva se na primeni posebnog kontrolera koji omogućava neposrednu razmenu podataka između pj i memorije.

signali za spregu sa pj-om imaju značenja: dmarq

je signal kojim pj obaveštava kontroler da je spremna za prenos sledećeg podatka.ako je

dmarq = 1 onda pj ne zahteva dma prenos, a za dmarq

= 0 pj zahteva dma prenos. signal kojim kontroler javlja pocetak ciklusa je dmaa. ako je

dmaa = 1, onda nije u toku dma prenos, dok je za

dmaa = 0 u toku.

dmarw signalom kontroler obaveštava o smeru prenosa podataka.

ako je

dmarw =1 onda se vrši prenos podataka od pj-e u memoriju, a ako je

dmarw = 0 prenos je u

suprotnom smeru.

dma kontroler poseduje registre u koje p upisuje inf kao: količina podataka koje treba preneti, početnu adresu u koju se upisuju podaci (kod ulaza) ili iz koje se čitaju podaci (kod izlaza) i sl. u tipičnom korišćenju mehanizma dma prenosa izdvajaju se tri intervala vremena:1) interval pre početaka 2)interval za vreme dma prenosa. 3)interval posle dma prenosa. p preduzima akcije vezane za završetak dma prenosa koje mogu biti: operacije nad ulaznim podacima, priprema novih izlaznih podataka, obaveštavanje korisnika da je prenos završen, inicijalizacija novog dma prenosa i sl.

60. FUNKCIONISANJE RAČUNARSKIH SISTEMA

da bi se poboljšavale karakteristike rač sistema neophodno je da se rešavaju problemi kao što su: promenljiva struktura, obezbeđenje paralelnog rada pj jedne u odnosu na drugu,i u odnosu na centralni procesor,unifikacija programiranja u/i operacija, obezbeđenja reakcija računarskog sistema na različite situacije. ovi se problemi rešavaju decentralizacijom upravljanja, unifikacijom sprežnih mreža, razvojem sistema prekidanja kao i razvojem sistemskog softvera –skupa progr kojim se upravlja resursima računarskog sistema. promenljivost

strukture raču sistema podrazumeva da repertoar operacija, povezivanje jedinica računarskog sistema kao i logika upravljanja moraju biti takvi da se sistem lako prilagođava zahtevima korisnika. to može biti obezbeđeno ako postoje: standardni formati poruka i upravljačkih signala koje razmenjuju jedinice računarskog sistema, standardizovani format u/i naredbi za sve pj-e. u/i naredba za procesor predstavlja naredbu za predaju poruke i ne vodi računa o fizičkoj prirodi pj-e koje se razlikuju samo po pridruženom broju (logičkom ili fizičkom), zajedničke magistrale za sve (ili samo neke) pj-e. da bi se ostvario paralelni rad procesora i pj-e neophodno je da u/i

naredba samo inicira rad pj-e da bi procesor potom nastavio sa obradom programa. takođe je

neophodno da procesor može da inicira rad sledeće pj-e, ne sačekavši da već inicirana pj-a završi rad.

pj-a, od trenutka iniciranja njenog rada, samostalno nastavlja izvršavanje u/i operacije stupajući u vezu sa procesorom samo na kratko – ako se za to ukaže potreba.