

DRUGI DEO - Centralni procesor

8. Struktura mikroprocesora ? - strana 31 slika 2.2

Centralni procesor (CPU) je ključni dio računarskog sistema koji izvršava aritmetičke i logičke operacije, upravlja memorijom i ulazno-izlaznim podsistemom.

Kod starijih računarskih sistema procesor je bio realizovan na jednoj ili više stampanih ploča.

U eri mikroprocesora centralni procesor se realizuje kao jedno integrisano kolo.

Na jednoj silikonskoj pločici zaštićenoj plastikom ili keramičkim kućem, smestene su sve potrebne komponente. Spoljni prikljuci se preko izvoda (pinova i terminalnih tacaka) i podnožja integrisanog kola povezuju sa ostatkom računarskog sistema na stampanoj ploči. Ovako integrisano kolo se naziva **mikroprocesor**, a ceo računarski sistem izgrađen oko njega **mikroračunar**.

9. Mašinski jezik ?, odje se može izvuc nesto i za 2 pitanje.

ISA – Instruction set Architecture - skup instrukcija i načini adresiranja potrebnih podataka-definisani.

Centralni procesor analizira binarni kod instrukcije, i izvršava niz mikrooperacija potrebnih za njenu realizaciju. Format binarnog koda je najuže vezan za sam procesor (masinu) te se stoga instrukcije nazivaju **masinske instrukcije**. Binarni program smješten u memoriji se konsekvntno naziva **masinski program**.

Masinski jezik je metod binarnog programiranja-zadavanje liste instrukcija i njima potrebnih podataka.

Masinske instrukcije, masinski program, masinski jezik (sta je sta, opisati) -> PITANJE NA ISPITU.

9.1 Format instrukcije ?

Format instrukcije definiše sintaksu masinskog jezika, određuje sledeće elemente: **tip operacije, broj i lokaciju operanda**.

Operandi mogu biti: **polazni ili odredisni** u zavisnosti od toga da li sadrže ulazne ili izlazne argumente date instrukcijom.

Kod operacije definiše operaciju koja će se izvršiti nad polaznim operandima radi generisanja rezultata koji će se upisati u odredisni registar.

Adresni slog jednoznacno označava lokaciju operanda u memoriji ili u nekom od registara.

Slika 2.3.

Broj operanada - varira uglavnom od nula do tri (nulta, unarna, binarna, ternarna instrukcija).

Adresnost je atribut instrukcije koji odgovara broju pristupa memoriji koji su potrebni radi njenog izvršavanja. Ukoliko su svi operandi implicitni ili u registrima adresnost instrukcije je nula.

Masinska instrukcija je određena kodom operacije i adresnim slogom koji određuje lokaciju operanda u registrima ili u memoriji. Tri osnovna tipa adresiranja: **registarsko, neposredno i memorijsko**.

Minimalna dužina instrukcije jednaka je jednoj procesorskoj kao i memorijskoj rijeci a može biti i duža.

11. Tipovi instrukcija i format podataka ?

TIPOVI INSTRUKCIJA:

Klasifikacija se vrši po dva kriterijuma : **1. vezan za lokaciju**, **2. za funkciju instrukcije**

Pod 1: **registarske instrukcije** (svi operandi su u registrima), **memorijske** (ako se bar jedan operand nalazi u memoriji, to zahtijeva pristup memoriji)

Pod 2: **Aritmetičke** (+, -, *, / binarno kodiranih brojeva, cjelobrojnih ili decimalnih), **logičke** (NE, I, ILI, NILI, EX-ILI, također i pomjeranje sadržaja registara-shiftovi), **instrukcije grananja** (realizuje bezuslovne i uslovne skokove, uslovno vrši provjeru nekog uslova), **instrukcije kontrole procesora i periferija**.

FORMAT PODATKA:

Format - interna predstava u racunaru, tj. način binarnog kodiranja podataka, takav da omogući i olakša njihovo čuvanje i obradu u digitalnom racunaru. Sve je u binarnoj formi. Prvo su numeričke vrijednosti: cjelobrojne, realne. Tekstualni podaci. Savremeni racunari imaju i audio i video signale.

Cijeli brojevi:

Cijeli označeni brojevi (integers) su osnova funkcionisanja digitalnih racunara.

Njihov format je određen tako da olakša izvođenje osnovnih aritmetičkih operacija, sabiranja i oduzimanja.

Cjelobrojne vrijednosti u racunaru su određene **znakom i vrijednošću**, znak je bit najveće težine (MSB). Nula u bitu znaka (S – sign) označava pozitivan broj, jedinica negativan.

Preostali biti definišu vrijednost cijelog broja, pri čemu se kod negativnih brojeva vrijednost zadaje njenim dvostrukim komplementom (oduzimanje cijelih brojeva se svodi na sabiranje). **Slika 2.6.**

Ovakav format direktno pomaže realizaciju ALU čija je osnova puni sabirac (full adder).

Slika 2.7.

Realni brojevi:

Format realnih brojeva ili brojeva u tekucem zarezu određen je standardom IEEE754.

Binarna predstava realnog broja obuhvata tri polja: **znak, eksponent i mantisa**.

Definisane su dvije varijante formata, brojevi **jednostruke i dvostruke preciznosti** koji se razlikuju samo po dužini polja eksponenta i mantise.

Znak je bit najveće težine, eksponent je pomjeren za konstantnu vr. (bias).

Mantisa je normalizovana pri čemu se vodeća jedinica ne čuva. Svaka binarna cifra mantise ima težinu 2^{-i} računato sa lijeva, gdje prvi bit ima težinu -1, sledeći -2 itd. **Slika 2.8.**

Ovakva predstavljena vr odg decimalnom broju izračunatoj prema formuli: **Knjiga.**

Redosled bajtova (endianess) – Big Endian (podrazumijeva da se u prvi bajt memorije smjesti bajt najveće težine, pa potom svi ostali), Little Endian (polazi od bajta najmanje težine).

12. Organizacija registara ?

Registri opšte namjene – registri čija je bitska dužina jednaka procesorskoj samim tim i memorijskoj rijeci. Primarno za čuvanje podataka te se nazivaju i registri podataka.

Adresni registri – neophodni ako su adrese duze u odnosu na podatke, sto je tipicno kod starijih mikroprocesora.

Programski brojac – specijalni registar koji cuva adresu tekuce i sledece instrukcije. Prije ucitavanja tekuce instrukcije ukazuje na njenu lokaciju, nakon cega se uveca i zato u toku izvrsne faze ukazuje na sledecu instrukciju.

Instrukcioni registar(Instruction Register IR) – se u fazi pribavljanja instrukcije puni sadrzajem ucitanim iz memorije sa lokacije na koju ukazuje programski brojac, sluzi za dekodiranje ucitane instrukcije.

Ukazivac steka(Stack Pointer SP) – registar koji ukazuje na trenutnu poziciju steka, segmenta memorije kojim se upravlja u maniru LIFO(Last in First Out) liste.

Registar stanja(Status Register SR) – smesta programsku rec stanja procesora koja se sastoji od **kontrolnih i indikatorskih bita**, kontrolni bit se postavlja od strane programa u cilju omogucavanja izveznih rezima rada centralnih procesora(Interrupt Enable, Kernel/User, Little/Big Endian, Memory Mode).Indikatorski biti se postavljaju automatski na osnovu nekog dogadjaja u toku izvrsenja aritmetickih i logickih operacija, zbog cega se u ALJ dodaje poseban registar. **Slika 2.13?**

13. Interna magistrala ? - strana 44 slika 2.14

Magistrala generalno oznacava jedinstveni prenosni put kojim se povezuje vise polaznih i odredisnih elemenata, u ovom slucaju registra i ALJ, izmedju njih medjusobno i izmedju registara i ALJ omogucen je dvosmjerni prenos.

Fizicka realizacija interne magistrale zasniva se na upotrebi multipleksera a cesce na koriscenju registara sa tri stanja.

Interna magistrala sa multiplekserima:

Vektorski mux sa n bita posreduje u prenosu sadrzaja iz jednog od polaznih registara do odredisnog registra, pod kontrolom upravljackih signala **Ki**. Ovakav mux stavlja se ispred svakog od registara u koji se vrši zapis, pa se stoga ovo resenje naziva **magistrala sa dodeljenim multiplekserima**. Broj ulaza muxa zavisi od broja polaznih registara.

Osnovna mana ovakvog resenja je veliki broj veza, koji je smanjen uvodjenjem **jednostruke magistrale** (usteda broja logickih kola, smanjuje broj potrebnih veza), realizovane takodje pomocu muxa. Ukoliko procesor ima N registara, ovo resenje koristi samo jedan mux sa $N-1$ ulaza a njegov izlaz formira jedinstveni put(magistralu).

Ako mozes nabubaj i sliku 2.16. (a) $6n$ ulaza $\rightarrow 9n$ logickih kola, b) $3n$ ulaza $\rightarrow 4n$ logickih kola)

Prenos sadrzaja polaznog registra na magistralu vrši se postavljanjem adekvatnog signala dozvole, dok zapis u odredisni registar kontrolisu signali za punjenje registara (**Load**).

15. Registarska memorija ? - strana 47 slika 2.19

Ekonomicnija varijanta organizacije u odnosu na registre koji medjusobno komuniciraju preko **magistrale**, predstavlja organizacija procesorskih registara u obliku **registarske memorije**. To je interna memorija malog kapaciteta i maksimalne brzine.

Registarska i operativna se razlikuju po obimu i brzini, razlikuju se i po nameni.

Operativna memorija sadrži **podatke i instrukcije**, a registarska memorija samo **operande**. Razmena informacija izmedju dvije memorije vrši se pomoću standardnih programskih instrukcija Load/Store.

Nedostatak ovakve organizacije, sa standardnom RAM memorijom, je neophodnost višestrukog pristupa memoriji radi izvršenja standardnih instrukcija, koje se kod magistralne organizacije izvršavaju u jednom taktu ciklusa.

Kod Magistralne org, komunikacija izmedju registara se obavlja direktno i u jedno taktu, izborom polaznog i odredisnog registra. Ode to nije moguće te se prenos odvija posredstvom prihvatnih registara A i B. Polazni registar se bira adresiranjem registarske memorije, i operacijom citanja sadržaj se prenosi u registre A i B. Posle obrade u ALJ, rezultat se zapisuje u registarsku memoriju koriscenjem adekvatne obrade i signala W.

Slika 2.21 (dual port)->Konacna verzija registarske memorije, opremljena adresom odredista D, i nezavisnim prikljucima A i B. -----NE MORAS OVO

17. Generički centralni procesor ? - strana 50 slika 2.23

Iako virtualan, GCP koristi sve grupe i tipove signala koji se sreću u realnim mikroprocesorima. Sustinski je vrlo blizak starijim ili skromnijim mikroprocesorima koji zbog malog broja adresnih linija podataka ne koriste tehniku mux prikljucaka.

Slika 2.23 (20, 21, 22, 23, 24 sve pise posebno sta su sta ove komponente dolje)

18. Vrste magistrala i kratko objašnjenje (adresna magistrala - ABUS, magistrala podataka - DBUS, upravljačka magistrala - CBUS) ?

Vrste magistrala: **adresna, magistrala podataka, upravljačka magistrala.**

Adresna magistrala - skup linija koje identifikuju lokacije kojima se pristupa od centralnog procesora u toku upisa ili očitavanja. Fizički je jednosmjerna, adresni prikljuci su izlazne linije sa tri stanja.

Magistrala podataka - skup linija za paralelan prenos podataka izmedju procesora i lokacije identifikovane validnom adresom, u toku upisa ili citanja. U oba smjera.

Upravljačka magistrala - naziv za sve kontrolne signale koji regulisu komunikaciju izmedju procesora i njegove okoline. Paralelni prenos podataka preko magistrale podataka.

19. Kontrola nad transakcijom (biranje i rukovanje) ?

Transakcija-izvršenje neke aktivnosti na magistrali (tipično prenos podataka).

Inicijator uvijek procesor (i u slučaju gdje DMA ne učestvuje).

Masinski ciklus-Vrijeme koje procesor potroši za jednu transakciju.

Izvršenje transakcije ima tri ciklusa: **pokretanje, kasnjenje i realizaciju.**

Zahtjev - podrazumjeva definisanje svih parametara za realizaciju prenosa, pre svega adrese uređaja i smjera.

Strobing-biranje!!! Koristi jedan signal **Strobe**, on označava da su svi ostali signali na paralelnoj magistrali postavljeni, odnosno da je **Zahtev** postavljen i da je validan, i da prenos može da se izvrši.

Handshaking-rukovanje!!! Koristi dva kontrolna signala: **Zahtjev**(request) – kojim procesor zahtijeva prenos podataka i **Odgovor**(reply, ready) – kojim prozvani uređaj označava izvršenje transfera, **Zahtjev** praktično mijenja Strobe samo što sad njegova dužina zavisi od signala **Odgovor**. Zahvaljujući ovim signalima, u toku transakcije se razlikuju četiri stanja, što omogućuje optimalnu kontrolu nad njenim izvršavanjem.

21. Rukovanje prekidima (INTREQ, INTACK, NMI)?

Sistem prekida(interrupts), je uveden kao efikasna tehnika komunikacije sa perifernim uređajima, koji omogućuje brz odziv procesora na U/I događaje uz minimalno trošenje dodatnog procesorskog vremena.

Vektor prekida – U/I uređaj preko magistrale podataka šalje svoju identifikaciju tj.vektor prekida na osnovu koje procesor može naći adresu odgovarajuće prekidne rutine. Zbog toga masinski ciklus prekida u punom (vektroskom) modelu ima faze **zahteva, prepoznavanja, prihvata**. Komunikacija U/I uređaja i procesora ostvaruje se posredstvom dva upravljačka signala,signal zahtjeva i potvrde prekida.

Dva upravljačka signala: **INTREQ (Interrupt Request)**, **INTACK(Interrupt Acknowledge)**.U mnogim procesorima postoji mehanizam tzv. nemaskirajućeg prekida **NMI(Non Maskable Interrupt)** prekid koji ne može biti programski zabranjen.

23. Upravljanje procesorom (RESET, HALT)?

Pored već navedenih postoje i druge upravljačke linije, npr. **RESET** (jedna od najvažnijih), čije aktiviranje izaziva trenutni prekid svih tekucih aktivnosti i postavljanje procesora u početno stanje. Zadatak **reset (boot) procedure** je da u različite registre centralnog procesora i upravljačke jedinice postavi odgovarajući sadržaj, početno stanje iz kog je moguće započeti rad na siguran način. Početno stanje podrazumeva brisanje svih kontrolnih registara stanja, ali i postavljanje tzv. **boot adrese**, početne adrese sa koje počinje učitavanje i izvršenje prve instrukcije. To je prva adresa koju generise procesor po uspešnoj inicijalizaciji i često se zove **reset vektor**.

Mnogi imaju i izlaznu liniju stanja.

Stanje halt/idle/sleep (više naziva) – procesor uspori takt procesora sve do pojave prekida.

25. Vremenska kontrola centralnog procesora ?

-**Ciklus takta** - elementarni interval aktivnosti obrade koja se naziva mikrooperacija. Promjene stanja centralnog procesora uzrokovane su učitavanjem nove instrukcije ili pojavom nekog spoljnog događaja, ali su sinhronizovane taktom (najčešće njegovom prednjom ivicom).

VREMENSKE FUNKCIJE:

-**Sekvenciranje mikrooperacija** zahteva razlikovanje taktova po vremenu, po redosledu pojavljivanja u odnosu na početak tekuće instrukcije. U toku sekvencijalnog izvršenja instrukcije uzastopni takti periodi međusobno se razlikuju korišćenjem vremenski pomerenih signala koje postavlja generator vremenskih funkcije.

Mikrooperacije unutar procesa traju 1 takt ali kada izađu vrijeme se produžava. Zbog toga se ovodi termin **masinski ciklus**.

INSTRUKCIONI I MASINSKI CIKLUS:

Masinski ciklusi obuhvataju: Interne operacije, memorijske operacije citanja i upisa, citanje i pisanje U/I registara, prepoznavanje prekida, preuzimanje magistrale, reset procesora.

Instrukcioni ciklus - niz vremenskih intervala u kojima se vrši prihvatanje i izvršavanje neke instrukcije.

TREĆI DEO - Memorija

26. Šta je memorija ?

Memorije su sekvencijalna kola, namjenjena cuvanju i rukovanju informacijama u okviru digitalnog sistema. U opstem slucaju organizovana je kao skup registara. Dvije operacije: citanje(read) i pisanje(write).

Klasifikacija memorije se vrši na osnovu razlicitih kriterijuma: nacin pristupa, trajnost podatak, lokaciji memorije i namjeni, njenu fizicku izvedbu.

*Prema nacinu pristupa na:

Memorije sa slucajnim pristupom (random access, RA) – koriste samo adresu lokacije za pronalazenje zeljene informacije.

Asocijativna memorija(content adressable memory, CAM) – prosledjuje se kljuc sa kojim se, u jednom ciklusu, upoređuje sadržaj svake od njenih lokacija. U slucaju poklapanja sadržaja isporučuje se njena adresa.

Klasicna adresna memorija – RAM (Upis i pisanje). To je prva sekvencijalna memorija kojom se realizuje operativna (radna) memorija racunara. Izbrisiva memorija.

*Prema trajnosti podataka, razlikuju se memorije sa **permanentnim**(zadržavaju sadržaj i ako ostanu bez napajanja, ROM) i **privremenim** pristupom(RAM),kao i memorije sa **izbrisivim(RAM)** i **neizbrisivim(ROM)** pristupom.

Klasican ROM (Samo citanje). Neizbrisiva memorija. To je kombinaciono kolo.

*Prema lokaciji i namjeni: **procesorska**(interna, skrivena), **operativna**(radna, glavna) i **spoljna**(masovna).

U odnosu na brzinu registara centralnog procesora, najbliza je interna memorija (jer je na istom integrisanom kolu) , ona je i najskuplja. U sredini je glavna, ili operativna memorija, kako po položaju tako i po performansama. Najsporija, najjeftinija ali i najvećeg kapaciteta je spoljna memorija.

Operativna memorija ključna za rad centralnog procesora jer iz nje učitava instrukcije i podatke, i u nju zapisuje rezultate obrade.

Procesorska memorija je ultra brza, skromnog kapaciteta i direktno spregnuta sa centralnim procesorom. Procesorska memorija služi za ubrzanje komunikacije između centralnog procesora i operativne memorije (skrivena memorija).

Najdalje je spoljna memorija, najsporija ali i najvećeg kapaciteta.

Spoljna se može koristiti za proširenje adresnog prostora centralnog procesora, odnosno za tzv virtualnu memoriju.

*Prema fizickom principu konstrukcije: **poluprovodnicke** (RAM, ROM, Flash), **magnetne** (diskovi, diskete, trake), **opticke** (CD, DVD), i sve cesce **Flash** memorije.

29. SRAM I DRAM ?

SRAM (Static Random-Access Memory) - najvisa brzina, umjerena gustina pakovanja na memorijskom cipu. To je izvorna poluprovodnicka memorija, koja koristi bistabilno kolo (flip-flop) kao osnovu memorijske celije. **Slika 3.12**

Uglavnom se koristi kao realizacija internih memorija u samom procesoru, interna SRAM memorija je ultra brza sa visokom cijenom, sa vremenom pristupa jednakim taktu procesora. **Slika 3.13**

DRAM (Dynamic Random-Access Memory) – Sporija od SRAM-a, dosta veci kapacitet. **Slika 3.14**

Minimizuje prostor potreban za realizaciju jedne memorijske celije jer jedan bit informacije cuva u minijaturnom kondenzatoru unutar integrisanog kola. Logicka 1- napunjen, logicka 0- prazan.

Informacije zapisane u DRAM celiju nije stabilna (staticka kao kod SRAM, zbog curenja struje preko tranzistora i praznjenja kondenzatora jedinica upisane u njega bi vremenom nestala). Glavno unapredjenje DRAM memorije je uvođenje sinhronog pristupa **SDRAM**. **Slika 3.15**. Zahvaljujuci svojoj prednosti u pogledu kapaciteta i cijene, dinamička RAM je danas osnova memorija savremenih racunarskih sistema.

MEMORIJSKI KONTROLER:

Povezivanje procesora sa svojim memorijama najcesce zahtjeva razvoj posebne spreznice jedinice koja se naziva **memorijski kontroler**. Njen zadatak je da uskladi razlicite fizicke protokole prisutne pri povezivanju procesora i memorijskih kola. Moze biti kao posebno integrisano kolo, ili kao dio samog mikroprocesora.

30. Asocijativna memorija ?

Za veoma brzu pretragu, provjera po spoljnom kljucu. U nekim primjenama neophodna je vrlo brza pretraga memorije kako bi se utvrdilo da li je sadrzaj neke o njenih lokacija istovetan spoljnom kljucu. Ako jeste, u slucaju poklapanja od interesa je adresa pronadjene lokacije. Zbog ovoga SRAM memorije se opremaju dodatnom logikom, koja moze da pretrazuje po sadrzaju (asocijativnim pristupom), u jednom taktu. Ovakva memorija se naziva adresabilnom po sadrzaju (**CAM**). Binarna i ternarna CAM (koriste samo gdje se ne mogu zamjeniti).

Binarna- kljuc i sadrzaj predstavljeni serijom nula i jedinica (bita 1/0).

Ternarna-slozenija logika koja rukuje i trecim logickim stanjem X, tj stanje „nema veze“. **Slika 3.19**

31. Rukovanje memorijom ?

Kako je operativna memorija jedinstvena ona se mora dijeliti postupkom dodele (alokacije) memorije.

UKOVANJE MEMORIJOM: oznacava postupke i tehnike dodjele(alokacije) operativne memorije koje su prisutne kao i nacin na koji krajni korisnik vidi memoriju koja mu se ustupa. Na tom planu razvijen je koncept logicke adrese, koji je zbog svojih prednosti danas dominantan.

Sa stanovista procesora i racunarskog sistema rukovanje memorijom oznacava dio hardvera koji prati i omogucuje efikasnu alokaciju memorije, koji u opstem slucaju poznat kao jedinica za rukovanje memorijom (MMU). **Slika 3.23**

MMU rjesava adresne reference prevodjenjem logicke u fizicku adresu i kontrolise pristup podacima.

OSNOVNI ZAHTJEVI:

Operativni sistem je program koji upravlja izvršenjem aplikativnih programa i deluje kao sprega izmedju korisnika i fizickih elemenata racunarskog sistema. Svoju funkciju OS ostvaruje kontrolom fizickih i logickih resursa racunarskog sistema.

Pristup procesoru i memoriji korisnici ostvaruju pomocu procesa koje pokrece OS.

Proces je osnovna izvrsna aktivnost procesora i predstavlja program u izvršenju.

Savremeni OS dele se u dvije osnovne klase: jednokorisnicki i visekorisnicki.

Efikasna dodjela radne memorije procesima je jedan od osnovnih zadataka OS.

U jednoprogramskom sistemu, memorija se dijeli na sistemski i korisnicki dio. Sistemski dio koristi operativni sistem, a drugi je u cjelosti raspoloziv aplikativnom. U visekorisnickom sistemu, ovaj drugi dio se dalje dijeli u nastojanju da se prihvati vise procesa, odnosno usluzi vise korisnika. U cilju podrške ovom zahtjevu, operativna memorija i procesor se opremaju dodatnim fizickim komponentama.

Zamena procesa (swapping) je tehnika operativnog sistema, koja eventualni nedostatak memorije rjesava tako sto na disk sklanja jedan ili vise aktivnih, ali trenutno blokiranih procesa.

33. Virtualna memorija ?

Virtualna memorija - adresiranje memorijskog prostora vece od realne operativne memorije. Osnovna ideja virtuelne memorije je da procesor generise adrese u opsegu znatno sirem u odnosu na raspolozivu fizicku memoriju, ali da svi dijelovi virtuelnog adresnog prostora ne moraju biti istovremeno prisutni u operativnoj memoriji. Onaj dio koji nije trenutno potreban cuva se u sekundarno memoriji, odnosno magnetnom disku.

Stranicna dodjela uvodi dva kljucna elementa za virtualnu memoriju: logicku adresu i podjelu memorije na neke dijelove kojima se moze nezavisno rukovati.

Osnovna tehnika u VM - **dodjela stranica na zahtjev** - stranicenje.!

Segmentacija- povezivanje skupa stranica u segmente.!

DODJELA STRANICA NA ZAHTJEV (demand paging):

Virtualizacija memorije sa stranicnom dodelom podrazumeva da neka stranica moze biti u operativnoj memoriji ili na disku. Zbog toga TPS tabela je dopunjena bitom validnosti stranice: 1 - stranica u operativnoj memoriji, 0 - na disku. DSNZ - Predstavlja dinamicku dodjelu realne memorije stranicama iz virtualnog adresnog prostora procesa.

U slucaju da je stranica u operativnoj memoriji ($V=1$), realna adresa se racuna na isti nacin kao kod klasnicnog stranicenja. Broj bloka B ucitan iz tabele TPS, je istovremeno njegova pocetna adresa, cijim sabiranjem sa odstojanjem 0 dobija realna stranica.

Ukoliko je stranica na disku ($V=0$), broj stranice B ukazuje na adresu sekundarne memorije na kojoj je smjestena stranica. Ovaj dogadjaj se naziva greskom stranice (page fault) i posredstvom prekida poziva komponentu OS zaduzenu za premestanje stranica. Njen prvi zadatak je trazenje slobodnog bloka. Ako nije moguće onda se primjenjuje algoritam za oslobadjanje. Najbolji je **LRU (Last Recently Used)** koji prati vrijeme pristupa nekoj stranici i pronalazi onu kojoj se najredje pristupa.

SEGMENTNA DODJELA:

Segmentacije je tradicionalnija tehnika, kod koje su programi bazirani na segmentima, nezavisnim blokovima memorije razlicite duzine. Segmente definise korisnik ili kompajler, shodno namjeni i svaki od njih ima svoju identifikaciju, duzinu i nezavisni adresni prostor. Segment sadrzi instrukcije ili podatke, sa kontrolisanim pristupom. Segment moze biti privatan ili dijeljen, sa dozvolom citanja, pisanja, ili izvršenja (RWE- Read, Write, Execute). Razlikovanje tipova korisnika vrsi se posredstvom tzv. nivoa privilegija i prstena zastite. Dva osnovna rezima rada: sistemski i aplikativni. Procesori Intelove x86 familije, koriste sledece segmente i registre:

- CS (Code Segment) sadrzi instrukcije,
- DS (Data Segment) cuva podatke,
- SS (Stack Segment) namijenjen steku,
- ES (Extra Segment) rezervisan za posebne namjene.

Virtuelna adresa se sastoji od broja segmenta i relativnog odstojanja unutar segmenta. Segmentacija, ova sto je do sad opisana, odgovara partitivnoj dodjeli.

SEGMENTNA DODJELA STRANICA:

Segmentacija se kombinuje sa stranicenjem sto zadrzava dobre strane oba pristupa. Kombinovana tehnika se zove **segmentna dodela stranica**, kod koje se segment stranicu, tj. dijeli na odredjeni broj stranica. Zbog toga virtualna adresa sada ima tri polja koja odredjuju segment, stranicu unutar segmenta, i odstojanje unutar stranice. Segment je i dalje u kontroli korisnika, odnosno kompajlera. Sa druge strane stranicenje unosi efikasnost i punu virtualizaciju memorije, jer sve stranice jednog segmenta ne moraju biti istovremeno pristune u radnoj memoriji.

ČETVRTI DEO - Ulazno-izlazni podsistem računarskog sistema

35. Memorijski preslikan i izolovan ulazno-izlazni podsistem ? - strana 95 slika 4.4

Povezivanje procesora sa U/I uredjajima, odnosno specijalizovanim U/I kontrolerima, vrsi se u sustini na isti nacin kao i sa memorijskim kolima. U/I sprega se ostvaruje posredstvom adresne magistrale i magistrale podataka, kao i upravljackih signala centralnog procesora.

U pogledu instrukcija i adresnog prostora koji se koriste pri pristupu U/I uredjaja tu su 2 varijante:

Prva-upotreba standardnih Load/Store instrukcija i RD/WR upravljackih signala. **To je memorijski preslikan ulaz/izlaz**, jer se U/I prolazima dodjeljuje adrese iz memorijskog adresnog prostora, isto kao i proizvoljnoj memorijskoj rijeci. Ovo ima dvije negativne posledice -smanjen raspoloziv memorijski prostor, istovremeno stvara mogucnost incidentnog pristupa U/I registrima. Otezava dekodiranje.

Druga- izolovan ulaz/izlaz-koristi posebne ulazno-izlazne instrukcije (In, Out) za komunikaciju sa U/I uredjajima, odnosno za prenos podataka izmedju jednog od registara procesora i porta U/I kontrolera. Olaksava dekodiranje u odnosu na memorijsko preslikan U/I.

38. Izvršenje prekida ?

Osnovna ideja, da se glavni tok programira kao da U/I uređaji i aktivnosti ne postoje.

Prekidna rutina - prenos kontrole ne vrši po izvršenju neke instrukcije tipa **call** nego po prijemu signala zahtjeva koji šalje periferna jedinica.

Po izvršenju prekidna rutina se instrukcijom **reti** (return from interrupt) se vraća na glavni tok.

Signali **INTREQ** (Interrupt Reguest), **INTACK** (Interrupt Acknowledge).

39. Dijeljenje sistema prekida ?-----

Sistem prekida mora odrediti koja je od perifernih jedinica postavila zahtev za prekidom, kao i to koja će od njih biti prva uslužena.

Dakle potreban je metod za određivanje izvora prekida i koji će prvi biti uslužen:

Programski metod - nevektorski sistem, jedna je zajednička prekidna rutina koja prozivkom utvrđuje izvor prekida. Prioritet je definisan u programskoj prozivci.

Hardverski metod- koristi kontroler prekida, koji prihvata zahtjeve iz više izvora, izvrši odlučivanje o prioritetu, i potom preda mikroprocesoru odgovarajući zahtjev.

40. Serijski i paralelni kontroler prekida ? - strana 101 slika 4.9 i strana

102 slika 4.11

SERIJSKI

Serijski podrazumijeva serijsku vezu uređaja koji generisu prekide.

Ako jedan ili više perifernih uređaja zahtevaju prekid, signal **INTREQ** se aktivira na ulazu procesora (stanje 0 u ovom slučaju), na šta centralni procesor, po završetku tekuće instrukcije odgovara postavljanjem signala potvrde **INTACK=1**.

Uređaj koji će biti uslužen zavisi od pozicije u lancu i na osnovu signala **PI** (Priority in) i **PO** (Priority Out). **Logika odlučivanja:**

- uređaj koji nije generisao prekid prenosi stanje **PI** na izlaz **PO** (**PO=PI**).
- u suprotnom uređaj započinje prekidni ciklus slanjem vektora adrese (**VAD**) na magistralu podataka. U toku prekidnog ciklusa **PO** ostaje neaktivan (**PO=0**).

PARALELNI

Paralelni- paralelna veza prekidnih linija omogućuje dodatne funkcije poput pojedinačne dozvole prekida i izmene njihovih prioriteta.

Jedna od prednosti paralelnog kontrolera je mogućnost pojedinačne dozvole i zabrane prekida posredstvom registra maske.

Koder prioriteta je centralna komponenta paralelnog kontrolera prekida, koja proverava prisustvo i međusobni prioritet aktivnih zahteva, i generise izlazni vektor adrese (biti **X** i **Y**).

IST(Interrupt Status) je bit stanja koji označava postojanje aktivnog zahtjeva za prekid.

Dozvola prekida **IEN (Interrupt Enable)** je upravljački signal kojim se kontrolisu svi prekidi povezani na isti prekidni kontroler.

Da bi se neki prekid realizovao neophodno je da bude dozvoljen na sva tri nivoa:

- na nivou procesora (bitom **IEN**).
- na nivou kontrolera prekida (pojedinačno i zbirnom dozvolom).
- na nivou periferne jedinice (podesavanjem upravljackog registra U/I kontrolera).

PETI DEO - Tehnike ubrzanja računarskog sistema

42. Tehnike ubrzanja uopšteno ?

Racunarski sistemi su unapredjivani:

- ***tehnoloski** (razvojem i usavršavanjem novih nacina izrade integrisanih kola u cilju postizanja sto vece brzine, pouzdanosti i stepena integracije).

- ***organizaciono** (usloznavanjem organizacije i arhitekture racunarskog sistema radi postizanja ubrzanja i/ili paralelnosti rada pri izvršenju procesa).

- ***programski** (razvojem slozenijih i efikasnijih kompajlera).

UBRZANJE MEMORIJE- problem → velika razlika u brzini izmedju procesora i DRAM operativne memorije. Dva rjesenja:

- ***memorija sa preplitanjem** (gdje se u toku svakog memorijskog ciklusa pristupa ka vise rijeci u operativnoj memoriji koje se cuvaju u razdvojenim memorijskim modulima).

- ***skrivena memorija** (mala i brza memorija koja se ubacuje izmedju procesora i operativne memorije radi smanjena vremena pristupa. Osnovna ideja ovog pristupa je da procesor, umjesto stalnog pristupa operativnoj memoriji, sto cesce cita kopiju potrebnih podataka koja se cuva u skrivenoj memoriji).

UBRZANJE PROCESORA - tehnike:

- ***vektorski procesor** (standardni skup instrukcija prosiren vektorskim).

- ***paralelna obrada** (oznacava simultano izvršenje operacija na centralnom procesoru sve u cilju ubrzanja rada). Dvije osnovne tehnike za organizaciju procesora sa paralelnim izvršenjem:

- ***umozavanje izvršnih jedinica** (koja paralelizuje izvršenje sekvencijalno učitanih instrukcija).

- ***protocna obrada** (sa osnovnom idejom o istovremenom izvršenju vise instrukcija ali u razlicitim fazama izvršenja).

UBRZANJE RACUNARA- multiprocesorski racunari koriste najmanje dva procesora, koji podjelom ukupnog opterecenja omogucuju brze izvršenje korisnicke obrade.

43. Memorija sa preplitanjem adresa ? - strana 111 slika 5.1

Uzastopne adrese odgovaraju lokacijama u susednim memorijskim modulima, sto rezultuje memorijskom mapom prikazanom na slici (ovaj donji dio). Ovakva podela susednih adresa izmedju fizickih susednih modula naziva se **preplitanje (interleaving)**.

Slika 5.1

Ovakva organizacija omogucuje istovremeni pristup svim fizickim modulima odnosno paralelni pristup memoriji. To zahtijevanje opremanje svakog RAM modula prihvatnim bafer registrom MBR, kao i :

- upis u svaki registar se radi na uobicajen nacin, iako zbog brzine moze biti realizovan preko MBR registra – odlozeni zapis

- citanje ima dvije varijante, u zavisnosti da li se promijenila *osnovna adresa*, definisana najvisim bitom adrese (A2-A15)

- ukoliko je osnovna adresa nova, citaju se svi moduli *interleaved* memorije, tacnije sve njihove lokacije sa istom internom adresom. Ovakav pristup se naziva citanje u nizu, duzina niza jednaka broju fizickih modula. Ucitani sadrzaj se cuva u MBR registrima.

-svi novi zahtjevi za citanjem su u istom nizu, odnosno sve dok se osnovna adresa ne promijeni, zadovoljavaju se prenosom sadržaja iz MBR registra. Kako su ovi registri znatno brži u odnosu na memoriju DRAM tipa, uštede u vremenu su značajne.

U praksi termin interleaved memory uvijek označava memorije sa paralelnim pristupom, koje se koriste kada je brzina pristupa imperativna.

Slika 5.2

45. Direktno preslikavanje (mapiranje) ?

Prva tehnika rukovanja je **direktno preslikavanje**, kod koje se adresa reci u skrivenoj memoriji formira izdvajanjem potrebnog broja adresnih bita najmanje težine direktno iz izvorne adrese reci u operativnu memoriju.

Ako skrivena memorija ima $S=2^k$ rijeci, operativna memorija se može podijeliti u blokove iste veličine čiji je broj jednak odnosu kapaciteta operativne i skrivene memorije. Pojedina rijec iz svakog od ovih blokova sa adresom **Aom** preslikava u skrivenu memoriju. **Asm = Aom modulo S**.

Indeks(adresa svake od memorijske rijeci u skrivenoj memoriji) i oznaka(gdje je u operativnoj memoriji njen par).

Prednost - jednostavnost postupka a i same fizičke realizacije.

Nedostatak - nagli pad procenta uspješnosti pronalazenja adrese u skrivenoj memoriji ako se desi da se dva najčešće korišćena bloka preslikavaju u istu oblast skrivene memorije.

46. Asocijativno preslikavanje ?

Asocijativno dozvoljava bilo kojoj lokaciji iz operativne memorije da bude mapirana u svaku od lokacija skrivene memorije. To znači da oznaka mora biti puna adresa (sem Byte adresnih bita). Optimalno rješenje za pretraživanje memorije oznaka postiže se asocijativnom organizacijom skrivene memorije.

Upis i citanje oznaka u memorijski blok se vrši na konvencionalan način. Dodatno je u svakom redu memorijskih ćelija pridružena logika za poredjenje upisane oznake sa odgovarajućim bitima adrese koju je generisao centralni procesor.

Potrebna je dodatna logika za izbor lokacije koja će biti obrisana nakon promasaja:

- Na slučaj (generator slučajnih brojeva),
- Zamjenom najstarijeg člana liste (FIFO),
- Zamjena člana koji najduže nije bio u upotrebi (LRU)

47. Skrivena memorija sa asocijativnim skupom ?

Realni procesori najčešće koriste **skrivena memorija sa asocijativnim skupom**, je kompromisna tehnika koja se najčešće koristi u realnim procesorima, **direktno preslikavanje sa asocijativnim skupom lokacija** – tačan naziv.

Ključna razlika u odnosu na direktno mapiranje je raspoloživost više parova oznaka/podataka za isti indeks, odnosno više „SM“ u koje može biti zapisana kopija originalne lokacije iz OM. Njihov broj se naziva dimenzijom SM.

Ista ova metoda se proširuje učitavanjem više reci istovremeno, odnosno učitavanjem niza (line). Efikasnost ovakvog rešenja zahtjeva proširenje prenosnih puteva između centralnog procesora, skrivene i operativne memorije.

49. Vektorski procesor ?

Optimizovan za sekvencijalno(uzastopno) izvršenje iste operacija ali nad nizom operanada(vektorom). $C[i] = A[i] \times B[i]$ gdje je \times operacija koja se ponavlja nad svakim elementom ulaznih vektora.

Vektorske instrukcije su složenije od standardnih. Vektorski procesori su opremljeni dodatnim komponentama koje ove instrukcije brzo i efikasno izvršavaju.

50. Superskalarni procesor ? - strana 126 slika 5.15

Oni koriste činjenicu da izvršenje instrukcije vrlo često traje duže od njenog učitavanja.

Umjesto jedne, **superskalarni računari** koriste umnožene izvršne jedinice, i paralelnu obradu ostvaruju distribucijom instrukcija i podataka ka njima. Instrukcijska jedinica zadužena je za učitavanje i pokretanje instrukcija, čije se izvršenje, pod kontrolom upravljačke jedinice, preusmerava na jednu od tri izvršne jedinice. Rezultati obrade zapisuju se u registre procesora (registarska jedinica) ili se posredstvom jedinice podataka prenose ka operativnoj memoriji. Ubrzanje rada koje se postize ovakvom organizacijom ograničeno je brojnim faktorima, i rijetko se u realnim uslovima može približiti teorijskom limitu. U većini praktičnih realizacija postoji neki stepen specijalizacije izvršnih jedinica pa se te instrukcije usmjeravaju ka jednoj od njih.

Superskalarni procesori primjenju kompleksne tehnike dinamičkog raspoređivanja instrukcija uz sve prednosti i probleme.

52. Protočni procesor, protočno učitavanje instrukcija ? - strana 133 slika 5.23

Standardni za sve namjene koje zahtijevaju visoku propusnost i performanse obrade. **Slika 5.19 Protocna obrada(pipeline processing)** – svi sekvencijalni procesi koji se izvršavaju po modulu pokretne trake gdje se dijelovi ukupnog procesa (fragmenti) realizuju na uzastopnim djelovima postrojenja (segmentima). Na nivou postrojenja, svi segmenti se izvršavaju istovremeno. Obrada neke jedinice počinje njenim ulaskom u prvi segment, realizuje se prolaskom kroz N susednih segmenata i završava napuštanjem protodne strukture. Svaki segment izvršava dio obrade a njegov rezultat je ulaz sledećem segmentu. Izlaz svakog od segmenata je i ulaz sledećeg segmenta.

Svaka operacija koja se može tako podijeliti na fragmente slične složenosti i trajanja, pogodna je za implementaciju u protocnom procesoru.

Rizici-najveći uzrok usporenja protocnih procesora, kritične situacije kada različiti segmenti istovremeno zahtijevaju pristup nekom od resurse računarskog sistema.

PROTOCNO UCITAVANJE INSTRUKCIJA:

U protocnoj organizaciji procesora, **instrukcijski bafer** je standardna struktura za protocno učitavanje instrukcija.

Nove instrukcije se učitavaju dok se tekuća izvršava, ovo se još zove baferom unapred učitanih instrukcija. Segment koji se bavi učitavanjem instrukcija obično se označava kao **IF** –

Instruction Fetch.

54. Rizik podataka i rizik grananja ?

RIZIK PODATKA nastaje kada sledeća instrukcija pokuša da koristi rezultat prethodne pre nego što je on raspoloživ.

Ovakve situacije se mogu rešiti na dva načina: **programski i hardverski.**

Programsko rjesenje se zasniva na umetanju NOP(no-operation) instrukcije izmedju kriticnih instrukcija.

Hardversko - zadrzavanje(stall), hardversko prosirenje upravljacke jedinice logikom koja vrsi detekciju operanda koji jos nije zapisan pa se umetanje NOP-a vrsi automatski.

Ukoliko negdje u procesoru, a pre svega u izolacionim registrima protodne strukture, postoji potreban podatak, ispravan ali jos nezapisan rezultat, on se moze isporuciti direktno korisniku pre njegovog regularnog zapisa u odredisni registar. Korisnik je naravno segment koji vrsi referencu na isti rezultat. Ova tehnika se oznacava kao prosledjivanje podataka (data forwarding).

RIZIK GRANANJA

(control hazard) vezan je za problem u prenosu kontrole nad izvršenjem programa odnosno pri grananju programa (uslovnom skoku).

Odloženo grananje - dodavanje dvije NOP instrukcije iza svakog grananja, nedostatak-gubljenje vremena uvijek bez obzira da li se skok desio.

Predviđanje grananja - je efikasniji metod koji u osnovnoj varijanti polazi od pretpostavke da se grananje neće desiti. Ovde se gubi vrijeme samo ako ima grananja.

Moguće su različita predviđanja skoka: **grananje neće biti nikada, grananje će biti uvijek, ili predviđanje na osnovu koda instrukcije (hoće/neće biti).**

Dinamicke metode - instrukciji grananja treba dodjeliti obeležje - status, one se zasnivaju da povecaju procenat pogodjanja, pamcenjem istorijata uslovnih grananja. Osnovna ideja jeste da instrukciji grananja treba dodjeliti obeležje - status (jedan ili vise bita) koji ce odrazavati nacin njenog izvršenja u prethodnom periodu.

Dvije osnovne tehnike:

1)Indikator grananja - skup bita (najcesce 1 ili 2) koji poput brojaca pamti prethodne ishode uslovnih grananja, na nivou procesora. Naravno ako se koristi samo jedna bit, tu se pamti samo zadnje grananje na osnovu kojeg se i pravi pretpostavka za sledece grananje.

2)Tabela istorijata grananja - ukoliko se pracenje grananja zeli ostvariti na nivou svake instrukcije treba se uvesti ova tabela (branch history table – BHT).

56. Slabo i čvrsto povezani multiprocesorski sistemi ? - strana 144 slika

5.30

Ako tehnoloska ili finansijska sredstva ne dozvoljavaju razvoj onda se koriste dva ili vise procesora.

Multiprocesorski sistemi (MPS) podrazumjevano umnozavanje i medjusobno povezivanje raspolozivih racunarskih komponenti, procesora, memorije.

Dva osnovna cilja za uvođenje MPS: poboljšanje sistemskih performansi (ubrzavanje rada na osnovu podjele posla na dijelove koji se paralelno izvrsavaju), podizanje pouzdanosti i raspolozivosti (svaka od funkcionalno identicnih racunarskih stanica je u stanju da obavi posao i da bude uvedena u operativni rad kao vodeca, dok su ostale rezervne).

Rekonfigurabilnost – postupak prilagodjavanja racunarskog sistema dogadjajima, internim ili u okruzenju.

Dva toka informacija: tok instrukcija, tok podataka.

Osnovna funkcionalna jedinica MPS je procesni element koji se cesto naziva i **cvor**.

Imamo: **slabo i cvrsto povezani MPS**.

Slabo povezani - cvorovi unutar njega međusobnu saradnju ostvaruju razmenom poruka preko komunikacione sprege. Svaki element ima svoju lokalnu memoriju, komunikacioni procesor i svoj operativni sistem.

Svrstavaju se u široku klasu distribuiranih sistema. U užem smislu računarska konfiguracija ovog tipa se naziva klasterom. **Klaster** je grupa računara spregnutih lokalnom mrežom koja se spolja vidi kao jedan računar.

Cvrsto povezani - svaki procesor ima pristup cjelokupnom memorijskom prostoru pa se nazivaju **multiprocesori sa dijeljenom memorijom**. Elementi iako mogu imati svoju lokalnu memoriju, međusobno dijele raspoložive memorijske module pod kontrolom jednog operativnog sistema. Sprega je na nivou magistrale.

Multiprocesorski sistemi moraju imati sledeće karakteristike:

- dva ili više procesora,
- svi procesori imaju pristup dijeljenoj memoriji,
- svi procesori imaju pristup U/I resursima,
- cijelim sistemom upravlja jedan operativni sistem,
- postoji saradnja između procesora na nivou hardvera i programske podrške.

58. Simetrični i asimetrični multiprocesorski sistemi ?

Simetrični multiprocesori - simetrija se mjeri uglavnom na hardverskom nivou, ali se i svaki od procesora tretira jednako od strane OS. Ona dominira kod **multicore** procesora koji su dominantni i na PC i mobilnim telefonima.

Asimetrični multiprocesor - Uloga grafičke kartice, koja samostalno ali pod kontrolom procesora određuje veliki dio aktivnosti potrebnih za prikaz video sadržaja.

Izvestan stepen asimetrije vezan je i za način pristupa deljenoj memoriji.

60. Multiprocesorski sistemi sa krosbar spregom ? - strana 148 slika 5.33

Ovo se može posmatrati kao organizacija sa više magistrala.

Krosbar matrica je potpuno odvojena od funkcionalnih jedinica i može se projektovati modularno da bi se omogućilo proširenje. Matrica se sastoji od horizontalnih i vertikalnih magistrala na čijem se preseku nalazi prekidacki element koji kontrolise prenos između njih.

Cvorni prekidac može biti u stanju:

- prespajanja (prenos poruke sa horizontalne na vertikalnu magistralu),
- propustanje (prenos na magistralu istog tipa),
- otvoreno(nema prenosa).

Prednost krosbar rješenja je u simultanom prenosu između različitih funkcionalnih modula, a osnovna mana kompleksnost fizičkog rješenja.

.

62. Koprocesori ? - strana 151 slika 5.38

Jedan od najprostijih varijanti povezivanja dva procesora se odnosi na korišćenje specijalizovanog procesora namijenjenog da rastereti osnovni procesor određenih specijalnih obrada. On se naziva koprocesor. Koprocesor je u cjelini pod kontrolom vodećeg procesora sa kojim je spojen i sa kojim deli magistralu radi korišćenja zajedničkih resursa.

Ovakav način rada posebno je pogodan za realizaciju U/I, matematičkih, grafičkih procesora i slično.

Koncept koprocesora je jedna od osnova savremenog računarstva.

Dva osnovna rezima rada:

- poluautonoman (kada vecinu svojih aktivnosti izvrsava koriscenjem resursa glavnog procesora. vodeci/prateci) i
- autonoman (pristupa centralnim, ali ima i svoje resurse).

63. Koherentnost skrivene memorije ? - straa 152 slika 5.39

Uticaj skrivenih memorija na efikasnost procesora je visok. Savremeni multiprocesori obicno koriste dva nivoa skrivene memorije gdje svako jezgro ima svoju privatnu cache memoriju prvog nivoa (L1 cache).

U slucaju da dva ili vise procesora istovremeno pristupaju promenljivoj u oper mem moze doci da neuskkladjenosti sadrzaja u skrivenim memorijama dva razlicita procesora-**koherencija**.

Slika 5.39

Ukoliko dva jezgra pristupe istoj promenljivoj X, u obje privatne skrivene memorije bice pohranjena njena pocetna vrijednost. Kada u nastavku obrade prvi procesor izmeni vrednost promenljive X, nova vrijednost bice sacuvana u njegovoj skrivenoj, ali i u zajednickoj operativnoj memoriji. Ipak, o toj promjeni drugi procesor nista ne zna, i zato zadrzava staru vrijednost promenljive X. U slucaju ponovnog pristupa, on ce regularno učitati ovu vrijednost, smatrajuci je pravom. Ovo se mora sprijeciti, a za to postoje dva rjesenja:

-Protokol invalidacije-izaziva slanje posebne komanda ka svim ostalim jezgrima da je promjenljiva na adresi X promjenjena i da su sve kopije nevazece.

-Metoda azuriranja kopije - pored adrese salje i novu vrijednost pa sve skrivene memorije cuvaju tu vrijednost.

Prvo rjesenje je prisutnije zbog manjeg opterecenja.

65. Procesori sa više jezgara i grafički procesori ?

Dva problema za dalje poboljsanje performansi: **potrosnja energija, duzina internih linija**.

Povecanje takta podize potrosnju, procesor se zagrijeva.

Drugi razlog koji ogranicava takt je kasnjenje elektricnih signala.

Rjesenje su procesori sa vise jezgara (integracija vise procesora unutar istog kucista i primjeni od ranije poznatog koncepta simetricnog multiprocesiranja). Kod njih pristup memoriji i optimizacija skrivenih memorija su kljucne za njihovu efikasnost.

Memorijski kontroler pa i memorija iza njega su direktno vezani za jedno od jezgara. Pristup ostatku samo preko drugih jezgara. Neophodna je ultra brza sprega izmedju jezgara unutar integrisanog kola zbog cijene. Svako jezgro ima svoju privatnu L1 skrivenu memoriju.

Pored hardvera, ovaj koncept imao je znacaj i na softversku industriju i opsti programski model krajnjih aplikacija. Paralelne aplikacije mogu da koriste moc svih jezgara. Konkuretno programiranje je znatno popularnije.

Osnovna funkcija OS kojom se paralelizuju programske aktivnosti je **smjena konteksta**, neophodna da bi nova programska nit postala aktivna. Smjena konteksta vezana je za stack, znaci operativnu memoriju pa je ona relativno skupa (i spora).

U namjeri da se ovaj problem ublazi razvijena je **Hyper-Threading tehnika (HT)** koja dodatnim hardverom pomaze smjenu kontekst na nivou samog procesora (jezgra)

GRAFIČKI PROCESORI: (mozda slika 5.47).

1999. pojavila se prva graficka kartica sa programskom kontrolom i sa mocnim skupom procesora posebno prilagodjenih specifcnostima video obrada. Njihovim radom upravlja

aplikacija koja se razvija u C/C++, puni u lokalnoj memoriji i odatle izvršava. Ta aplikacija ne mora biti graficka. Po prvi put graficke kartice nadmasile mogucnosti centralnog procesora. Ovakve kartice se nazivaju **grafickim procesorima**. GPU procesori brojem izvjesnih operacija nadmasuju CPU savremenike i do 20 puta.

Prva pogodnost je sama priroda grafickih operacija.

Graficke aplikacije se upisuju u kernel memoriju odakle se učitavaju instrukcije.

Učitavanje instrukcija i pristup memoriji organizovani su u okviru multiprocesorskog jezgra.

Dvije cinjenice koje uproscavaju realizaciju GPU procesora. **Graficke aplikacije nemaju suvise grananja. Sve niti u izvršenju su nezavisne.** Ako se pojavi rizik samo se data nit zaustavi a angazuju se novi procesori i nove niti.

SEDMI DEO - Arhitektura namenskih računara

66. Šta su namenski računari, kako izgledaju ? - strana 207 slika 7.1

NRS i grana racunarstva koja se bavi njima su trenutno u fokusu savremenih racunarsikih tehnologija. Termin NRS oznacava racunar ugradjen unutar drugog elektricnog ili mehanickog uredjaja. Na tom racunaru se izvršava program koji u realno vremenu kontrolise rad cijelog uredjaja, iako je nevidljiv korisniku namenskog sistema. Dakle tu su i hardver i softver ali koriste se drugacije od desktop racunara.

Prethodna slika ukazuje na razliku ove dvije kategorije NRS sistema.

RTOS – real time operativni sistem.

Zahtjevi prilikom izbora komponenti i projektovanja namenskog sistema:

- Potrebna procesna snaga i kapacitet memorije odredjuju se tako da odgovaraju minimumu koji garantuje ispunjenje vremenskih uslova vezanih za rad u realnom vremenu.

- Cijena proizvodnje, vrijeme i troskovi razvoju su veoma bitni kod jeftinih i masovnih proizvoda.

- Kvalitet softvera takodje odredjuje uspjesnost nekog proizvoda.

Ocekuje se poznavanje C jezika.

Programer NR mora poznavati hardver i principe UI sprege ka periferijama.

68. Koja tri načina postoje da prebacimo izvršni kod ?

Upisom izvrsnog koda u programski EPROM, serijskim prenosom preko UART, USB ili LAN komunikacione linije, Programer/Debugger (zahtjeva JTAG ili BDM port).

Upisom izvrsnog koda u programski EPROM, koji se potom fizicki instalira na stampanu ploču odredisnog sistema, insertovanjem u postolje. Testiranje programa se radi koriscenjem lokalnih resursa (LED dioda ili displeja, ispisima preko komunikacione linije i sl.) ali svakako bez upotrebe testnog alata (debuggera). Svaka izmjena koda zahtjeva brisanje EPROM-a i upis novog programa prije ponovne instalacije. Ovo je najelementarniji, najsporiji i najmanje komforan nacin rada.

Serijskim prenosom, preko UART, USB ili LAN komunikacione linije. Ovakav nacin rada je moguc samo ako na odredisnom sistemu vec postoji komponenta za punjenje programa (punjac, loader) koja umije da prihvati novi program, upise ga na pravo mjesto u memoriji, i potom prenese izvršenje na njega. Program se prenosi komunikacijom dva programska punjaca: jednog na host, i drugog na target strani.

Programer/Debugger – Najkomfortniji i najpouzdaniji nacin rada podrazumjeva upotrebu komponente koja se naziva programer/debuger cijim posredstvom se ostvaruje fizicka veza sa hardverskim komponentama odredisnog sistema, sto ukljucuje procesor, memoriju, i U/I

uredjaje. Ovakav nacin rada zahtjeva JTAG ili BDM port na odredisnom sistemu, odnosno hardversku podrsku za debugovanje na nivou cipa (On-Chip Debugging).

70. Šta je potrebno pre pokretanja programa ? - strana 212 slika 7.5

Pri punjenju programa, prije pokretanja, programu se dodjeljuju jos dva memorijska segmenta neophodna za rad.

Velicina memorijskog bloka rezervisanog za **Stek** definise se u fazi pripreme programa (**STACKSIZE** direktivom).

Heap je zona memorije sa dinamicom alokacijom. Funkcije malloc i free.

Sekcije se shodno tipu podataka mapiraju u neke od memorije namenskih sistema (RAM, ROM, Flash...) pomocu MEMORY i SECTIONS linkerskih direktiva.

71. Binarna kompatibilnost ?

U praksi vrlo je vazna mogunost da se neka aplikacija (program) razvijen na jednom sistemu bez ponovnog kompajliranja primeni na nekom drugom racunaru sa istim procesorom i OS.

Binarna kompatibilnost podrazumjeva postovanje niz konvencija neophodnih da bi neki program mogao biti ucitan od strane OS i izvršen na konkretnoj platformi.

Skup ovih konvencija – ABI, obuhvata pravila vezana za OS i samu fizicku platformu. Pocev od kompajlera, svi alati moraju postovati isti set ABI konvencija.

Prvi skup pravila definise format izvrsne datoteke, podjelu memorije izmedju OS i aplikacije, interakcija pri pokretanju aplikacije itd.

Zavisnost od platforme obuhvata sve elemente arhitekture procesora (skup instrukcija, tipovi podataka, Little/Big endian, adresiranja ali i:

- poravnanje podataka u memoriji,
- konvencije pozivanja funkcija,
- koriscenje registara,
- nacin formatiranja i
- koriscenje steka.

73. Šta je punjač (loader) ?

Punjac je uslužni program koji je po pravilu smesten u ROM/Flash memoriji, kako bi bio raspoloziv i po iskljucenju napajanja. Poziva ga reset procedura ako je transfer programa neophodan. Punjenje se kontrolise sa: LoadDone (promenljiva u flash memoriji), postavkom spolnog prekidaca/jumper-a ili razmenom poruka na pocetku rada.

74. Priprema programa za izvršavanje ?

Dva su osnovna nacina da se obezbedi aplikativni program NRS:

- da se unapred upise u trajnu memoriju (ROM/Flash) ili
- da se napuni u radnu memoriju (RAM).

U skladu sa time postoje dva osnovna nacina izvrsenja aplikacije:

- izvrsenje iz ROM-a, ili
- iz RAM-a nakon punjenja

Prije izvrsavanja aplikacije neophodno je u RAM memoriji formirati .data .bss i stack sekcije.

76. JTAG sprega (Hardversko testiranje)?

Sve akcije koje po zahtjevu razvojnog sistema preduzimaju **monitor ili debug-agent** završavaju se pristupom registrima centralnog procesora, memorije ili U/I uredjaja. Spoljni i nezavisni pristup internim elementima integrisanih kola je kljucna funkcionalnost hardverske sprege

koja podrzava rad savremenih razvojnih okruzenja. Ovakav nacin rada se naziva testiranjem na nivou cipa (OCD), i oznacava hardversku podrsku procesu kontrole programa i otklanjanju gresaka (debugging) ugradjenu u same procesore i njihove periferije.

Dva tipa **OCD**: standardizovan JTAG sprega i mnoga druga resenja specificna za pojedine proizvođače.

JTAG (Joint Test Action Group) je resenje predloženo 1985. Godine, a 5 godina kasnije standardizovano kao IEEE 1149.1 - standard test access port and boundary scan architecture. Ovakvo ime ukazuje namjenu:

- Standard Test Access Port kaže da se radi o standardnom portu za pristup resursima nekog digitalnog kola bez poznavanja i nezavisno od njegove interne strukture.

- Boundary Scan je arhitektura dodatne logike koja se ugradjuje u interesano kolo koja omogućuje takav pristup.

Zapravo, JTAG je Boundary-Scan prosiren OCD podrskom.

78. Veza JTAG uređaja na štampanoj ploči ?

Na stampanoj ploči nekog racunara po pravila ima više JTAG komponenti. Pristup svakoj od njih vrši se preko istog **JTAG konektora** jedinog na ploči pri čemu se svi uređaji povezuju serijski, direktnim povezivanjem signala TDO i TDI (testni signali).

Prvi uređaj u lancu je obično procesor ali to nije obavezno.

Dovodjenjem prave kontrole na JTAG konektor može se:

- kontrolisati procesor u smislu pokretanja, zaustavljanja i inspekcije rada programa,
- napuniti memorija programom ili nekim drugim sadržajem,
- pristupiti svakom od BS (Boundary scan) modula na ploči.

80. Kako izgleda grafički prikaz mikrokontrolera ? - strana 225 slika 7.19

U cilju eliminisanja uskih grla i nedostataka MIPS protodne organizacije, učinjene su dvije karakteristicne nadogradnje:

- Prva je obezbedjenje rezervnog seta registara opšte namene, pored standardnog skupa od 32 MIPS registra. Rezervni registri (shadow set) se koriste pri opsluzivanju prekida visokog prioriteta ili postavljanjem kontrolnog registra u CPO koprocesoru.

- Druga dogradnja eliminise probleme u atomskom (neprekinutom) rukovanju sa podacima, odnosno nedostatak adekvatnih MIPS instrukcija sa atomskim izvršenjem.

Slika pokazuje da su memorija i periferija organizovane u dva nivoa:

Na visem nivou: blize procesoru, nalaze se memorije, kontroleri prekida i DMA , kao i najbrze periferije. Povezane su sa procesorom pomocu Matricne magistrale(Crossbar). Bafer instrukcija sa slike je u stvari eksterna skrivena memorija za instrukcije. Flash je trajna memorija za cuvanje programa umjesto ROM-a. Pored Ethernet i USB kontrolera, smjesten je jos GPIO modul – modul brzih digitalnih U/I.

Na nizem nivou: sporije periferije i manje zahtjevne u pogledu obima podataka u prenosu, povezuje ih Klasicna magistrala(Periferijska magistrala) koja je povezana na matricnu magistralu, odje se nalaze tajmerska i kola analognih ulaza/izlaza, kao i dvije specijalizovane magistrale sa serijskim prenosom (SPI I²C).

82. Tajmeri i brojači ?(Nzm trebaju li slike 7.23 i 7.24, Tojzan ih je oznacio)

Programibilni tajmer - kada se treba izmjeriti neki vremenski interval i izazvati prekid. Osnova svakog tajmera je brojac koji broji interno ili eksterno generisane taktne impulse. Svojom namjenom se izdvaja zaštitni tajmer (WatchDog Timer, **WDT**) koji se koristi za detekciju i oporavak neispravnosti u radu racunara uzrokovanih softverskim ili hardverskim greskama. Odgovornost aplikacije je da u predviđenom vremenu reda jedne sekunde restartuje WDT tajmer cime potvrđuje da je ziva i funkcionalna. U slucaju da aplikacija upadne u beskonacnu petlju, restart ce izostati.

(NE TREBA OVO JA MSM)

84. PWM - širinski modulisani impulsi ? - strana 230 slika 7.26

U realnim aplikacijma kao izlazni upravljacki signal, cesto se koristi **PWM** (širinski modulisani impulsi, Pulse Width Modulation). Radi se o povorci impulsa iste amplitude i ucestanosti ali i promenljive sirine odnosno promenljivog odnosa signal/pauza.

Slika 7.26

Sirirna **PWM** signala varira 0-100%.

Pored perioda, **PWM** signal se zadaje preciznošću podešavanja impulsa, minimalnim korakom za koji se moze promijeniti širina impulsa. Zbog toga je vremenska rezolucija **PWM** izlaza odredjenja kolicnikom vremena perioda i broja koraka: $T_{min} = T_{period} / N_{koraka}$

Prakticna realizacija PWM izlaza nezamisliva je bez upotrebe tajmera i dodatne logike podešene zahtjevima primene.

86. ADC - Analogno digitalni konvertor ? - strana 233 slika 7.31 + šeme koje je profesor crtao na predavanjima

A/D konvertor (Analog to Digital Converter, ADC) prihvata kontinualni signal, odmjerava ga u opsegu 0 - Vref i proizvodi izlaznu cjelobrojnu vrednost srazmjernu ulaznom naponu. Jedan od osnovnih parametara **A/D konvertora** je brzina konverzije koja ogranicava ucestalost pristupa **A/D kolu**.

(Slika 7.30 mozda treba, cini mi se kao da je neko crtao)

(NE TREBA)

Najbrzi su tzv. flash konverteri sa direktnom konverzijom(istovremeno odredjuju vrijednost svake od binarnih cifara). U praksi su najcesci A/D konvertori sa uzastopnom aproksimacijom(kompromis cijene i perfomansi). Osnovna ideja SA konvertora je da se u petlji odredi svaka od cifara polazeci od bita najvece tezine (MSB). Cijeli postupak se odvija u dvije faze: prihvatanje/memorisanje ulazne vrednosti (odmeravanje) i njeno odmeravanje (A/D konverzija).

Slika 7.31

(NE TREBA)

Prijemni stepen **A/D konvertora** cine analogni multiplekseri (AMUX) i SHA (Sample and Hold Amplifier).

Prvo kolo izabira jedan od analognih ulaza i prosledjuje ga SHA kolu, koje ga odmerava i pamti u kapacitivnim celijama. Zadatak SHA je da sve vrijeme A/D konverzija obezbedi stabilnu vrijednost na ulazu komparatora.

Registar SAR cuva trenutnu odmerenu vrednost i na pocetku konverzije se brise (postavlja na 0). U nastavku se odredjuje bit po bit izlazne digitalne vrijednosti, polazeci od D0 (bita najvece tezine) prateci sledecu logiku: **(MISLIM DA OVO ISPOD NE TREBA)**

- Postavi bit tezine $2^{(n-1-i)}$ (cifru D_i) na vrijednost 1.
- Komparator upoređuje dva signala, ulaz koji se odmerava i izlaz D/A konvertora.
- Ako je generisana D/A vrijednost konvertora veca, cifra D_i se vraca na nulu, u suprotnom ostaje jedinica.
- Postupak se ponavlja sve dok se ne odredi i zadnji bit, bit najmanje tezine.
- Kraj konverzije se oznacava signalom EOC (End Of Conversion).

88. Inter-Integrated Circuit magistrala ? - strana 236 slika 7.36 i slika 7.37

I²C magistrala u celosti serijalizuje komunikaciju sa perifernim kolima jer selekciju krajnjeg uređaja radi pomocu njegove komunikacione adrese. Za realizaciju sprege su potrebna samo dva signala pa se ova magistrala cesto naziva i **dvozicnom spregom** (2 wire interface). Postoji 1 master cvor iako vise cvorova moze da se takmici za kontrolu nad magistralom.

Slika 7.36

(NE TREBA)

I²C magistrala (Slika 7.36) koristi samo dvije dvosmjerne linije, za prenos takta (SCL) i podataka (SDA). Master generise takt, dok se kontrola nad linijom podataka razmijenjuje tokom odvijanja transakcije.

Slika 7.37

(NE TREBA)

Ova slika (Slika 7.37) prikazuje opsti tok **I²C** transakcije, koja zapocinje obaranjem SDA signala dok je tak jos neaktivan. Inverznom promjenom signala se transakcija i završava te se zbog toga ove dve sekvence nazivaju start i stop bitima, koje uokviruju transakciju.

OSMI DEO - Arhitektura PC računara

89. Čipset arhitektura ? - strana 238 slika 8.1

Čipset je skup komponenti matične ploče koje upravljaju tokom podataka između procesora, memorije i U/I periferija. Čipset implementira ključne komponente U/I podsistema počev od kontrolera prekida i DMA kao i sve memorijske U/I sprege prisutne u računaru. Drugim riječima čipset definiše komunikacionu infrastrukturu preko koje se prenose svi podaci unutar računara.

Uobicajena je upotreba memorijskog i U/I kontrolera koji se nazivaju sjeverni i juzni most.

Sjeverni most (Memory Controller Hub, MCH) medjusobno povezuje najbrze komponente PC računara: procesor, memoriju, graficku karticu i zbog toga je projektovana za postizanje max performansi. Spregu sa procesorom i generatorom takta realizuje ceona magistrala (Front-Side Bus, FSB).

Prisutan je trend da se sve veci dio MCH funkcionalnosti seli ka samom procesoru, integracijom u istom kucistu za sta ima vise razloga: zavisnost od samog procesora, postizanje max brzine i propusnosti, minimizacija broja spoljnih signala.

Juzni most (IO Controller Hub, ICH) posvecen je komunikaciji sa pravim U/I periferijama, sporijim u odnosu na graficku karticu ali i dalje vrlo zahtjevnim po obimu podataka i brzini prenosa.

Na najnižem nivou LPC (Low Pin Count) magistrala objedinjuje najsporije uređaje, i zadržava opciju ISA kompatibilnosti. SuperIO je komponenta koja integriše spregu sa tastaturom, misem i stampacem kao i podršku za asinhronu serijsku komunikaciju.

Juzni most obuhvata DMA i kontroler prekida, sat realnog vremena, kontrolu napajanja, Flash/CMOS memoriju..

91. Podsystem prekida ? - strana 240 slika 8.3

U kaskadnoj konfiguraciji glavni kontroler prosledjuje zahteve pratećeg kontrolera posredstvom jednog od svojih IRQ ulaza. U tu svrhu je iskorscen treci IRQ ulaz, te je zbog toga original signal IRQ2 preusmeren na sekundarni kontroler prekida.

Slika 8.3

To znaci da je u dugom periodu PC racunar imao samo 15 U/I prekida pri cemu su mnogi od njih zauzeti standardnim periferijama(disk, tastatura, mis). U prvim implementacijama PCI magistrale uređaji na njoj su mapirani na jedan od slobodnih prekida (IRQ10 najcesce). Ovakva ogranicenja su neprihvatljiva sa pojavom multicore procesora i simetricnog multiprocesiranja te je kao zamjena **uvodena APIC (Advanced Programmable Interrupt Controller) arhitektura**. Sustina je u distribuiranoj strukturi sa lokalnim APIC modulima integrisanim sa svakim jezgrom i U/I modulima lociranim uz svaku od magistrala ili periferija povezanih na juzni most.

93. Inicijalizacija PC računara i operativnog sistema ? - strana 242 slika 8.6

Prva faza inicijalizacije PC racunara odvija se u realnom rezimu rada, pod kontrolom BIOS-a cija je ulazna tacka mapirana na adresu reset vektora. U slucaju da procesor sadrzi vise jezgara aktivira se samo jedno (bootstrap processor) koje izvorsava reset proceduru i puni i pokrece jezgro OS (kernel).

Kljucni zadatak **reset procedure** je da provjeri da li su u racunaru prisutne i funkcionalne sve neophodne komponente poput memorije, prekida, DMA, video kartice, tastature, diska i sl. Pa se naziva **POST** .(power-on self-test) procedurom. U prvom koraku pod kontrolom BIOS-a se puni i pokrece jezgro OS(kernel). Po preuzimanju kontrole kernel prevodi procesor u zasticeni rezim rada, i nastavlja sa punjenjem ostatka OS i pokretanjem korisnickih procesa .

Izbor i lociranje OS vrse se na osnovu sadrzaja prvog sektora primarnog diska PC stanice. Ovaj sektor cuva **glavni boot slog** (Master Boot Record, MBR).

U okviru MBR sloga prvih 440 bajta su rezervisani za upis **punjaca kernela**, male programske rutine koju učitava i pokrece BIOS. Boot Loader pronalazi particiju koju treba napuniti. Prateci podaci definisu pocetak i kraj particije u fizickim i logickim koordinatama kao i njenu velicinu (broj sektora).

95. Standardne PC magistrale ?

Po tipu i namjeni PC magistrale mozemo klasifikovati u cetiri kategorije:

Procesorska magistrala je na hijearhijski najvisem nivou sa namenom povezivanja procesora sa najbrzim periferijama: osnovna procesorska kola (chipset), skrivena (cache) i operativna memorija.

Lokalna U/I magistrala služi za povezivanje brzih i vremenskih kritičnih U/I uređaja poput video kartica, diska, brzih mreznih kontrolera i sl. Najpopularnije su: PCI, AGP, PCI Express. Standardna PCI magistrala (PCI jedna od ovih mag) odlikuje se prenosom podataka dužine 32 bita i radnim taktom 33 MHz. Pored unapredjenja brzine i propusnosti PCI magistrala je prva u

cjelini podržala „prikljuci i koristi“ (Plug & Play-PnP) koncept povezivanja osnovnog PC računara sa perifernim U/I karticama.

Accelerated Graphics Port (AGP) je lokalna magistrala posvećena isključivo sprezi sa video podsistemom.

Standardna U/I magistrala namenjena je povezivanju sporih U/I uređaja poput: misa, modema, zvucnih kartica i sl. Kao i za obezbeđenje kompatibilnosti sa starijim uređajima. U većini današnjih uređaja to je ISA magistrala.

Serijska U/I magistrala je u odnosu na prethodno navedene magistrale alternativan način povezivanja U/I uređaja pri čemu se izbegavaju problemi uklapanja perifernih uređaja u memorijskom i U/I adresnom prostoru računara.

97. ISA magistrala ?

ISA (Industry Standard Architecture) je najstarija PC magistrala, pristupa od pojave prvog računara ovog tipa. Raspoloživa je sa 8 i 16 bita podataka. U svojoj 16-bitnoj verziji koristi se u industrijskim komponentama.

Zbog skromnih performansi zamenjena je novim i brzim magistralama. Prisutna je u industriji, razlog je taj što nije potrebno više od njenih performansi.

Svojom izvedbom **ISA magistrala** obezbeđuje pogodan način povezivanja sa sporim perifernim uređajima, i pri tome garantuje kompatibilnost sa velikim br. računara i perifernih komponenti.

Svi signali na ISA su TTL logičkog kola osim napajanja +5V i +12V.

Pored procesora tu su signali koji omogućuju:

- kontrolu čitanja i upisa u memorijski ili U/I prostor,
- generisanje stanja čekanja,
- korišćenje sistema prekida i direktnog pristupa memoriji,
- osvežavanje memorije i detekciju greske.

98. PCI magistrala ? - strana 253 slika 8.11

Peripheral Component Interconnect (PCI) specifikacija definiše lokalnu magistralu visokih performansi koja podržava prenos podataka dugih 32 ili 64 bita preko multipleksiranih linija adresa i podataka. Osnovna namjena **PCI** je sprega procesorskog jezgra sa brzim periferijama, bilo da su integrisane na osnovnoj ploči ili povezane ugradnjom U/I kartica.

Slika 8.11

PCI magistrala je primenjena na različitim hardverskim platformama, čime je obezbeđena kompatibilnost i laka prenosivost perifernih kartica. Takođe koncept „Plug & Play“ je osnova prednost.

Slika 8.11 pokazuje prvu izvornu arhitekturu računarskog sistema sa **PCI magistralom**. Ključna komponenta je PCI most koji obezbeđuje spregu procesora, skrivene memorije i operativne memorije sa periferijama koje se priključuju preko konektora na magistrali. To znači da periferni uređaj komunicira sa **PCI magistralom** a ne procesorom direktno što dozvoljava laku zamenu procesora i kompatibilnosti PCI uređaja na računarima sa različitim CPU platformama. Izbegavaju se kasnjenja i uska grla.

PCI uređaji/kartice označene kao agent mogu biti vodeći ili pratili u zavisnosti ko pokrene razmenu podataka.

OSNOVNI SIGNALI NA PCI MAGISTRALI:

PCI magistrala koristi sopstveni interni sistem prekida.

Sustina rukovanja **PCI magistralom** podrazumeva fazu adresiranja, gdje se pored adrese postavlja i rezim rada izdvajanjem komande nad magistralom. Prenos podataka vrši se u okviru jedne ili više faza prenosa podataka. Problem: prisustvo velikog broja signala.

PCI PODRSKA KONCEPTU „PRIKLJUCI I KORISTI“:

Svaki PCI uređaj u svom adresnom segmentu čuva posebnu tabelu koja sadrži sve podatke potrebne za dodjelu sistemskih resursa i njegovo kasnije rukovanje od strane operativnog sistema.

PCI konfiguracioni prostor se sastoji od 256 bajta i podijeljen je na dva dijela. Dijelom od adrese 00h zaključno sa adresom 0Ch koji predstavlja **PCI zaglavlje**, a ostatak predstavlja **Prostor PCI uređaja**. Sva polja koja se sastoje od više bajtova su u „little endian“ obliku to jest nize adrese sadrže bajte manje težine.

100. USB ?

USB:

(USB – Universal Serial Bus), osnovna ideja Univerzalne serijske magistrale je resavanje iznijetih problema izmestanjem sporih uređaja iz racunara, tacnije izvan njegovog memorijskog i U/I prostora. Komunikacija sa njim se ostvaruje serijski, posredstvom specijalizovanog kontrolera povezanog na PCI magistralu, koji koristi samo jedan vektor prekida i jedan skup memorijskih i U/I adresa. USB magistrala je fleksibilna. Cijena koja se placa je njena unutrašnja složenost. USB je dijeljena serijska magistrala kod koje se vecina inteligentnih spreznih funkcija izvrsava na samom racunaru.

Osnovne karakteristike USB magistrale su:

- povezivanje do 127 periferijskih uređaja u viseutičnoj konfiguraciji,
- sinhrona komunikacija preko dijeljene dvozicne linije,
- brzina prenosa 1.5 Mbit/s (spori), 12Mbit/s (uređaji pune brzine), 480 Mbit/s (uređaji visoke brzine),
- automatsko prepoznavanje prikljucenja nove periferne jedinice i njen nesmetan rad,
- četiri rezima komunikacije,
- jedan tip konektora i kablova za povezivanje.

USB ARHITEKTURA:

SLIKA

Može i slike 8.19, 8.20, 8.21 i 8.28 ako hoće

Na vrhu USB stabla direktno povezanog sa PCI magistralom nalazi se glavni kontroler kao osnovna upravljacka jedinica USB magistrale.

Komunikacija sa USB uređajima odvija se na osnovu podataka koji definisu adresu USB uređaja, tip i smjer prenosa podataka, i adresu memorijskog bafera rukovaoca uređaja. Fizicka sprega ka USB uređajima odvija se posredstvom hijerarhijske strukture tzv. čvornih (hub) uređaja, čija je primarna funkcija umnozavanje pristupa USB magistrali. Korenski cvor je osnovni spojni element koji obezbedjuje komunikaciono povezivanje glavnog kontrolera sa uređajima.

Osnovna funkcija cvora je realizacija fizickog prenosa podataka.

Dodatne funkcije:

- Konverzija brzine prenosa između sporih uređaja i magistrale.
- Dozvolu i zabranu komunikacije preko priključaka.
- Prepoznavanje prikljucenja uređaja i detekciju njegove brzine prenosa.
- Kontrolu napajanja na svojim priključcima, što podrazumijeva uključivanje/isključivanje napajanja.

- Smanjivanje struje napajanja pojedinacnog uređaja.
- Odredjivanje i postavljanje statusa (stanja) priključka.

Četiri tipa USB prenosa:

- prekidni prenos, na osnovu prozivke uređaja na USB magistrali, imitira se prekid.
- masovni prenos, prenos velikog obima podataka, ali bez vremenskih uslova.
- izohroni prenos, za najzahtjevnije periferije koji zahtjevaju prenos podataka velikog obima uz vremenske uslove.
- prenos kontrolnih poruka, obuhvata komunikaciju u cilju koordinacije rada, prepoznavanja i inicijalizacije USB uređaja.

Vod - logicka konekcija, prenos podataka se desava izmedju neke odredisne tacke na uređaju, i memorijskog bafera programskog rukovaoca.

Transakcija se dalje realizuje razmenom niza paketa, predznak oznacava pocetak transakcije, a preostali paketi nose podatke ili kontrolne informacije.

FIZICKO OKRUZENJE USB MAGISTRAL:

USB specifikacija propisuje minimum mehanickih i elektricnih zahtjeva koje svako kompatibilno rjesenje mora zadovoljiti.

Mehanicki dio specifikacije definise karakteristike konektora tipa A i tipa B, lociranih na cvoru i USB uređaju.

USB kabal ima cetiri zile, dvije za napajanje, i dvije signalne (za komunikaciju). Standardna USB kabal je oklopljen, dug do 5m i moze se koristiti za sve brzine prenosa. NRZI je poznata metoda za utiskivanje takta u signal koji se prenosi, pa prenos nije potreban preko posebnih zila.

102. Video memorije (VRAM, WRAM, SGRAM, MDRAM) ? - može da pita tabelu sa brzinama strana 277 tabela 8-6

Video memorija, u kojoj se cuva svaki piksel prikaza slike, kod savremenih resenja implementirana je na samoj grafickoj kartici. Osnovni zadatak video memorije je postizanje sto veceg kapaciteta i sto brzog pristupa.

VRAM (Video RAM) – varijanta standardne DRAM memorije, opremljene sa dva pristupna porta, tako da se na njoj istovremeno moze izvorsiti i citanje i pisanje podatka. Izbjegava usko grlo uslovljeno istovremenom potrebom video procesora da upise nove video podatke, i RAMDAC kola koje upisani sadrzaj prenosi do monitora, zahtijeva mnogo rjedje osvjezavanje od DRAM – pa je zato i brza.

WRAM (Windows RAM) - unaprijedjena verzija VRAM memorije, takodje dvoportna i 25% brza. Obezbijedjena je niza cijena realizacije. Koristi se u mocnim grafickim karticama visoke rezolucije do 1600x1200. Samo ime nema nikakve veze sa Windows operativnim sistemom.

SGRAM (Synchronous Graphics RAM) – zasnovana je na SDRAM tehnologiji, ali dodatno prilagodjena izvrsenju grafickih operacija pomocu blokovskog i bitskog zapisa. Blokovski upis realizuje punjenje bloka memorije istim sadrzajem, a bitski zapis primjenom odg. maske omogucuje postavljanje jednog razreda bita, bez uticaja na ostale. Memorija je jednopristupna.

MDRAM (Multibank DRAM) – je nova vrsta Video memorije, koja se od prehodnih sustinski razlikuje po tome sto se dijeli u blokove, kojima se moze nezavisno pristupati. Blokovi su velicine 32 KB, i povezani su internom magistralom. Zato je omogucen istovremeni pristup, fleksibilnost u upotrebi raspolozive memorije pri razlicitim rezolucijama dubini boje, ista brzina pristupa nezavisno od velicine memorije. Sve to rezultuje ekonomicoscu primjene.

DEVETI DEO - Spoljne memorije

104. Prognoza otkaza ?

Industrijski standard koji se koristi za prognozu otkazivanja diskova je **S.M.A.R.T. (Self-Monitoring Analysis and Reporting Technology)**. Kada se omogući S.M.A.R.T. nadzor, na disku se stalno prate određena svojstva, na osnovu kojih se mogu predvidjeti poremećaji ili otkazivanja diska. U slučaju kritičnih problema, S.M.A.R.T. će alarmirati sistemski BIOS ili OS. Oni će potom upozoriti korisnika, koji će moći na vrijeme da napravi rezervnu kopiju podataka. Svojstva koja prati:

- broj radnih sati i broj radnih ciklusa.
- vrijeme potrebno za postizanje radne brzine okretanja.
- visina lebdenja glava
- temperatura diska i razlika u odnosu na ambijentalnu temperaturu.
- performanse pristupa i protoka podataka
- učestalnost gresaka različitih vrsta itd..

Ima i nepredvidivih kvarova a to su npr: kvarovi izazvani statičkim električnim tokom, nepravilnim rukovanjem, iznenadnim potresom, otkazom neke komponente.

106. Raid 0

Njena priroda podrazumijeva zapis isjecaka podatka (striping), na različite fizičke diskove. Pri tome nema čuvanje redundantnih podataka, pa samim tim i mogućnost oporavka podatka u slučaju greske. **Prednosti:** jednostavan i pogodan za praktičnu realizaciju, značajno se unapređuje kapacitet, ali i ukupne U/I performanse RAID 0 skupa diskova, pogotovo ako svaki od diskova posjeduje svoj kontroler. Centralni procesor ne trpi dodatno opterećenje usljed racunanja i smještanja podataka o paritetu. **Nedostatak:** zbog nedostatka redundantnih zapisa i neotpornosti na otkaz, nivo 0 i nije pravo RAID rješenje. Otkaz jednog od diskova izaziva gubljenje svih podataka, zato ne smije biti korišćeno u kritičnim aplikacijama. **Slika 9.6.**

107. Raid 1

Sustina RAID 1 postupka je paralelan zapis podatka na dva identična, međusobno nezavisna diska. U okviru RAID podistema, može biti više ovakvih udvojenih diskova. Postizanje najviših performansi zahtijeva sposobnost **disk konektora** da konkretno vrši čitanje i zapis na dva diska u paru. **Prednosti:** moguće je istovremeno čitanje sa dva diska, čime se dvostruko ubrzava pristup zapisanim podacima. Zapis ide na oba diska, što rezultira istom brzinom kao na pojedinačnom disku. Potpuno dupliranje podatka, omogućuje jednostavan oporavak podatka u slučaju otkaza, tj. prosto kopiranje na novi disk. Ovo je najjednostavnija „prava“ RAID tehnika. **Nedostatak** je neefikasno rukovanje kapacitetom diskova, da bi se izbjeglo opterećenje centralnog procesora, zahtijeva se posebna hardverska logika. Ona istovremeno omogućuje „vruću“ zamjenu pokvarenog diska. **Slika 9.7**

109. Raid 3

U svemu je sličan prehodnom rješenju, ali se umjesto Hemingovog koda racuna samo jednostavan kod parnosti, on se zapisuje na samo jedan dodatni disk, nezavisno od broja diskova podatka. U slučaju otkaza jednog od diskova, ovakav način zapisa omogućuje oporavak izgubljenih podataka. Postupak rekonstrukcije podatka je vrlo jednostavan. Ponavljanje ex-ILI operacije nad sadržajem preostalih diskova može se nadoknaditi vrijednost isječka kojeg nema. **Prednosti** ove tehnike je visoka brzina prenosa podatka i efikasno korišćenje diskova.

Nedostatak je relativne složenosti kontrolera, ukupan broj transakcija koji je zbog sinhronizacije diskova isti kao i kod pojedinačnih diskova. **Slika 9.9.**

110. Raid 4

Za svaki od blokova u vrsti, bit po bit, računa kod parnosti i zapisuje ga na poseban disk. Procedura provjere i oporavak greske je istovjetna, kao kod RAID 3. **Nedostatak** RAID 4 tehnike je vezan za zapis podataka malog obima, koji mogu da stanu na samo jedan od diskova. Tada je, zbog azuriranja koda parnosti, neophodno izvršiti citanje starog bloka (koji se zamjenjuje) i bloka na disku pariteta. Tek po učitavanju starog bita podatka i bita parnosti, kontroler može izračunati i zapisati novu vrijednost bita parnosti. Sve to zahtijeva dva citanja i jedan zapis. Kod parnosti se zapisuje paralelno sa podacima i tako se izbjegava dodatno citanje / zapis nad diskovima u RAID 4 sistemu. **Prednosti** su sposobnost izvršenja zahtjeva visoke učestanosti, brzo učitavanje podataka, i efikasno korišćenje diskova. **Nedostaci** su uglavnom vezani za relativnu sporost u zapisu podatka, što postaje još složenije u slučaju otkaza jednog od diskova, jer se tada usporava ukupna brzina prenosa podatka. **Slika 9.10**

112. Raid 6

Ova varijanta RAID rješenja je u sustini RAID nivoa 5 proširen dodatnim, nezavisno izračunatim kodom parnosti koji se zapisuje na drugi disk. Tako na RAID 6 jedinici imamo **osnovne** podatke parnosti obračunate postupkom XOR (P-parnost), ali i **dodatne** izračunate nekim drugim postupkom (Q-parnost). Zato ovo rješenje, poznato i kao RAID sa P+Q redundacijom, omogućuje regeneraciju podataka i u slučaju dvostrukog otkaza osnovnih diskova. Implementacija RAID 6 tehnike zahtijeva N+2 diskova. Jedna od varijanti implementacije podrazumijeva računanje parnosti po redovima (P-parnost) i po kolonama (Q-parnost). Ovakvo rješenje, najbolje za kritične aplikacije, istovremeno je i najsloženije. Javlja se niza brzina zapisa, uslovljena računanjem i zapisom dvostruke parnosti. **Slika 9.12**