

PRVI DEO - Osnovne funkcionalnosti

1. Šta je arhitektura, šta je organizacije, koje su razlike ?

...

2. Odakle se kreće projektovanje računara (ISA) ?

Projektovanje kreće od definisanja instrukcija koje procesor podržava, što je ugovor između softvera i hardvera, nakon definisanja ISA (Instruction Set Architecture) kreće se na projektovanje arhitekture i organizacije.

U opstem slučaju računarski sistem predstavlja skup funkcionalnih jedinica (hardware) i programskih komponenti (software) kojima se obezbeđuje konačan broj usluga grupama njegovih korisnika.

Arhitektura (architecture) - osobine vidljive programeru – osobi koja razvija program za njega.

Organizacija - parametri vidljivi projektantu računarskog sistema.

Arhitekturu stoga čini skup instrukcija procesora, skup njegovih registara (često se naziva registarska arhitektura), način predstavljanja podataka, adresne tehnike, ulazno-izlazni mehanizmi...

Organizaciju čine funkcionalne jedinice, način njihove međusobne sprege, kontrolni signali kojima se definiše izvršavanje instrukcija procesora.

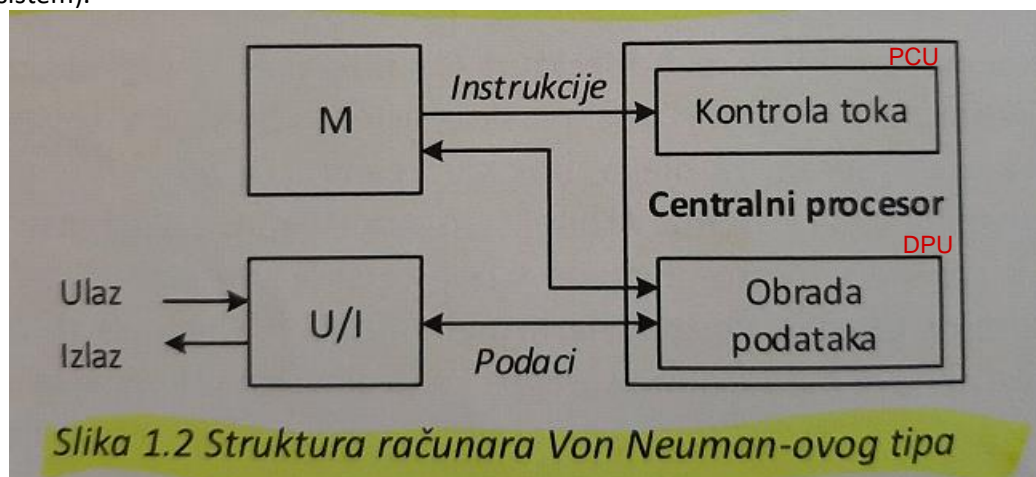
U sustini, arhitektura i organizacija računarskog sistema su nezavisne u tom smislu što na jednoj istoj arhitekturi može da postoji više različitih organizacija. Pri tome svaki računar je jedna implementacija neke arhitekture.

Programska kompatibilnost, odnosno mogućnost izvršavanja istih programa na različitim generacijama nekog procesora, je posljedica dijeljenja iste arhitekture.

3. Organizacija računara , 4. Osnovna Von Neumann arhitektura?

Sto se tiče organizacije računara, tačnije skupa osnovnih komponenti i načina njihovog sprezanja, Von Neuman je 1946. godine iznio prijedlog.

Pored centralnog procesora (ključna komponenta), tu su memorija (smještanje i čuvanje podataka u toku obrade) i U/I podsystem (razmjena informacija sa okolinom, uključujući i komunikaciju sa korisnicima računarskog sistema).



Centralni procesor ima dvije funkcionalne cjeline – PCU (Program Control Unit) - jedinica za kontrolu programskog toka (obuhvata upravljačku jedinicu ali i registre posebne namjene) i DPU (Data Processing Unit) – jedinica za obradu podataka (uključuje ALU i skup registara podatka). Spoljni podaci, oni koji se

razmjenjuju sa U/I uređajima, prolaze istu putanju kao i oni iz memorije. To znači da i oni prolaze kroz jedinicu za obradu podataka, što ograničava efikasnost samog prenosa. Da bi se ovo uklonilo, dodao se direktan pristup **memoriji (Direct Memory Access, DMA)**. Pod kontrolom CP ali bez njegovog direktnog učesca, periferni uređaj i memorija autonomno obavljaju prenos podataka. Znači uveden je **specijalni upravljački automat DMA kontroler**, koji upravlja prenosom između U/I jedinice i memorije a CP procesor se angazuje samo na početku i na kraju U/I aktivnosti.

+Slika 1.3

5. Šta je kod instrukcije i kakva adresiranja postoje ?

To je grupa bita koja opisuje datu instrukciju i njene operande. Binarna i masinska instrukcija su sinonimi sa kodom instrukcije jer svi definišu zahtev ispostavljen procesoru.

Djeli se na: **kod operacije i adresni dio**.

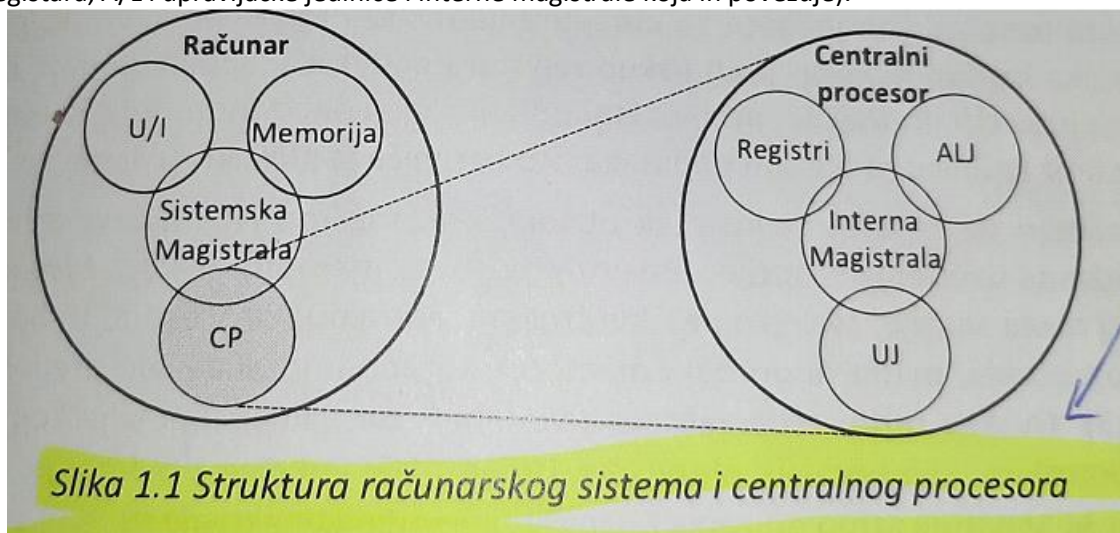
Slika : KOP | Adresni Dio

Implicitno adresiranje (lokacija operanda je zadata kodom operacije), **neposredno adresiranje operanda** (definiše njegovu vrednost koja se cita iz memorije neposredno iza koda operacije), **neposredno adresiranje memorije** (definiše adresu memorijske lokacije na kojoj se podatak nalazi).

6. Struktura centralnog procesora, navesti komponente i čemu one služe? - strana 1

slika 1.1

(Na visem nivou, osnovne komponente svakog računara su centralni procesor, memorija, U/I podsistem i sistemska magistrala koja ih spreže. Sam centralni procesor kao ključna komponenta računara, sastoji se od registara, A/L i upravljačke jedinice i interne magistrale koja ih povezuje).



Centralni procesor je ključni dio računara koji učitava i izvršava instrukcije, dakle realizuje obradu informacija. Sastoji se iz nekoliko osnovnih funkcionalnih cjelina.

Osnovne funkcionalne cjeline: skup registara, aritmeticko-logička jedinica, upravljačka jedinica, interna magistrala.

Registri: služe kao brze memorijske lokacije za smestanje vazne informacije u toku rada centralnog procesora. Registri mogu biti: registri podataka, adresni registri, registri posebne namjene (programski brojac-cuva adresu sledece instrukcije i instrukcioni registar-cuva kod instrukcije u izvršenju)

Aritmeticko-logicka jedinica (ALU): je funkcionalna jedinica namenjena za realizaciju osnovnih mikrooperacija nad sadržajem registara.

Upravljacka jedinica: usmjerava informacije iz registara na ulaze ALU, koja ih potom obradjuje.

Centralni procesor sa memorijom i U/I uređajima komunicira preko spoljne magistrale a radi realizacije prenosa informacija unutra centralnog procesora se organizuje vec pomenuta interna magistrala.

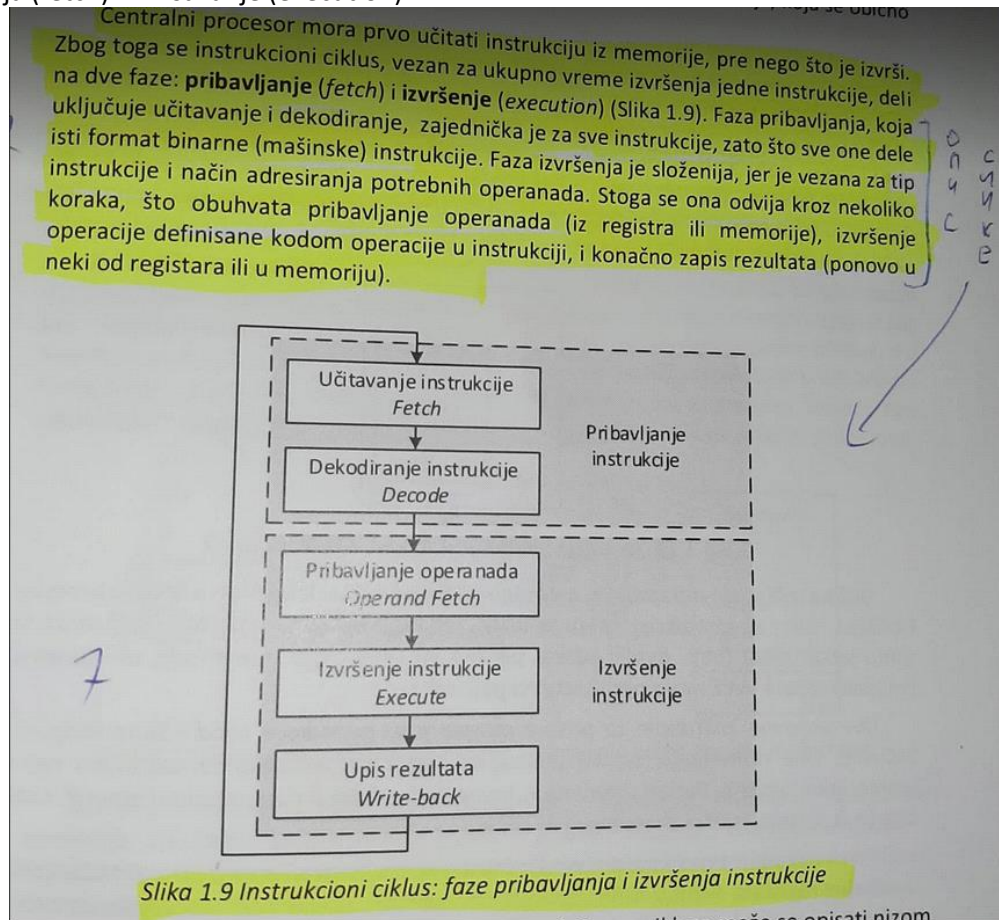
Upravljacka jedinica kontrolise funkcionalnost procesora postavljanjem upravljackih signala. Oni odredjuju koji registar ce se citati a koji puniti, koju ce mikrooperaciju ALU izvršiti. Skup upravljackih signala naziva se **kontrolna rec** procesora

(IAS MASINA, ARHITEKTURA I ORGANIZACIJA - DOPISATI)

NE TREBA.

7. Kako izgleda proces izvršenja instrukcija (fetch , execution) ?

Pribavljanja (fetch) i izvršavanje (execution)



DRUGI DEO - Centralni procesor

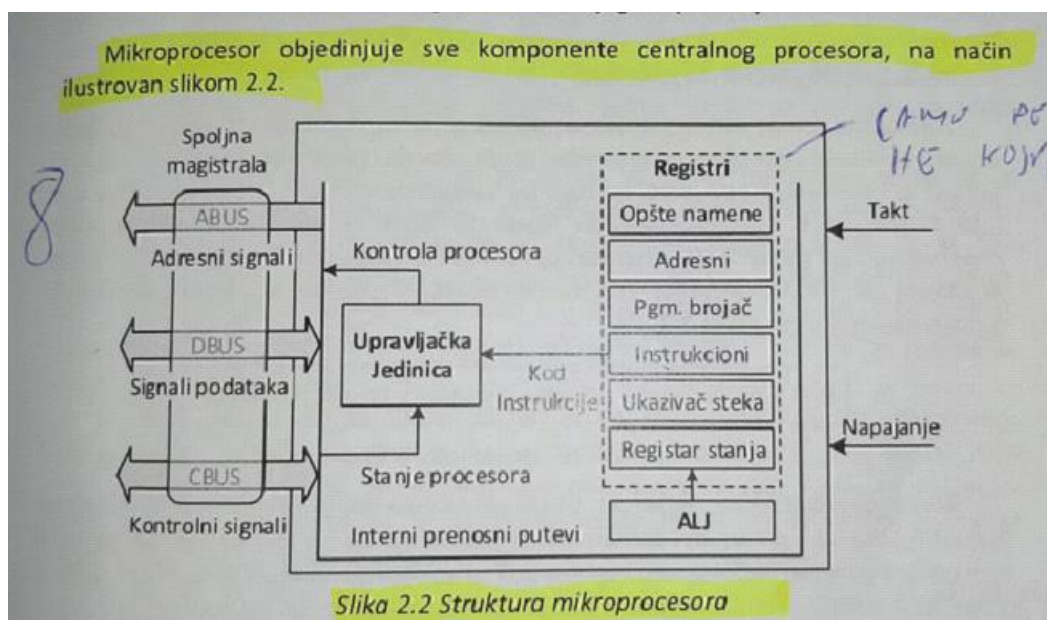
8. Struktura mikroprocesora ? - strana 31 slika 2.2

Centralni procesor (CPU) je ključni deo 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 plastičnom ili keramičkim kucistem, 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 moze izvuc nesto i za 2 pitanje. (gore)

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

(CPU) - 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 konsekvatno 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 definise operaciju koja ce se izvesti nad polaznim operandima radi generisanja rezultata koji ce se upisati u odredisni registar.

Adresni slog jednoznacno oznacava 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 odredjena kodom operacije i adresnim slogom koji odredjuje lokaciju operanda u registrima ili u memoriji. Tri osnovna tipa adresiranja: **registarsko, neposredno i memorijsko.**

Minimalna duzina instrukcije jednaka je jednoj procesorskoj kao i memorijskoj rijeci a moze biti i duza.

10. Klasifikacija računara prema skupu instrukcija (RISC I CISC) ?

CISC (Complex Instruction Set Computers), **RISC** (Reduced Instruction Set Computers)

Tabela 2.1?

CISC- su procesori sa kompleksnim kodom operacije, brojnim instrukcijama optimizovanim za programske jezike viseg nivoa i kompaktne programe.

RISC – su procesori sa jednostavnim kodom operacije, skromnog skupa instrukcija , posveceni postizanju visoke propusnosti i brzom izvršavanju instrukcija (**brzoj obradi**).

RISC – Jednostavnija upravljacka jedinica i pogodnija za primjenu tehnika za ubrzanje, pogotovo za protocnu (pipeline) organizaciju centralnog procesora sto rezultira vecom brzinom rada.

Sa druge strane CISC procesori, brzinu kompenzuju **manjim brojem fetch ciklusa. Broj registara je manji u odnosu na RISC**

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, takodje 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. nacin binarnog kodiranja podataka, takav da omoguci i olaksa njihovo cuvanje 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 oznaceni brojevi (integers) su osnova funkcionisanja digitalnih racunara. Njihov format je odredjen tako da olaksa izvodjenje osnovnih aritmetickih operacija, sabiranja i oduzimanja.
- Cjelobrojne vrijednosti u racunaru su odredjene **znakom i vrijednoscu**, znak je bit najveće težine (MSB). Nula u bitu znaka (S – sign) oznacava pozitivan broj, jedinica negativan.
- Preostali biti definisu vrijednost cijelog broja, pri čemu se kod negativnih brojeva vrijednost zadaje njenim dvostrukim komplementom (oduzimanje cijelih brojeva se svodi na sabiranje). **Slika 2.6.**

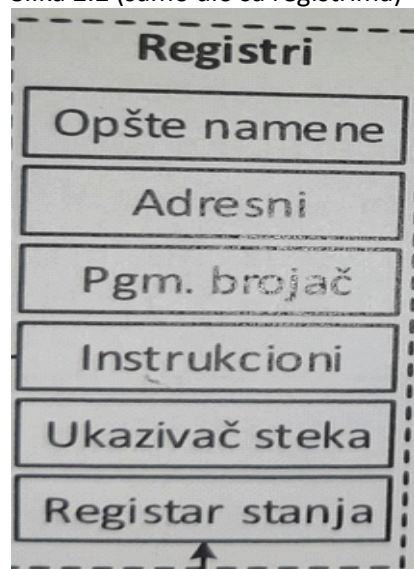
Ovakav format direktno pomaze realizaciju ALJ cija je osnova puni sabirac(full adder). **Slika 2.7.**

Realni brojevi:

- Format realnih brojeva ili brojeva u tekucem zarezu odredjen je standardom IEE754. Binarna predstava realnog broja obuhvata tri polja: **znak, eksponent i mantisu**. Definisane su dvije varijante formata, brojevi **jednostruke i dvostruke preciznosti** koji se razlikuju samo po duzini polja eksponenta i mantise.
- Znak je bit najvece tezine, eksponent je pomjeren za konstantnu vr. (bias). Mantisa je normalizovana pri cemu se vodeca jedinica ne cuva. Svaka binarna cifra mantise ima tezinu 2^{-i} racunato sa lijeva, gdje prvi bit ima tezinu -1, sledeci -2 itd. **Slika 2.8.**
- Ovakva predstavljena vr odg decimalnom broju izracunatoj prema formuli: **Knjiga**.
Redosled bajtova (endianess) – Big Endian(podrazumijeva da se u prvi bajt memorije smjesti bajt najvece tezine, pa potom svi ostali), Little Endian(polazi od bajta najmanje tezine). **Slika 2.9.**

12. Organizacija registara ?

Slika 2.2 (samo dio sa registrima)



Registri opste namjene – registri cija je bitska duzina jednaka procesorskoj samim tim i memorijskoj rijeci. Primarno za cuvanje 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 postavalja 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**.

Slika 2.14.



Interna magistrala sa multiplekserima:

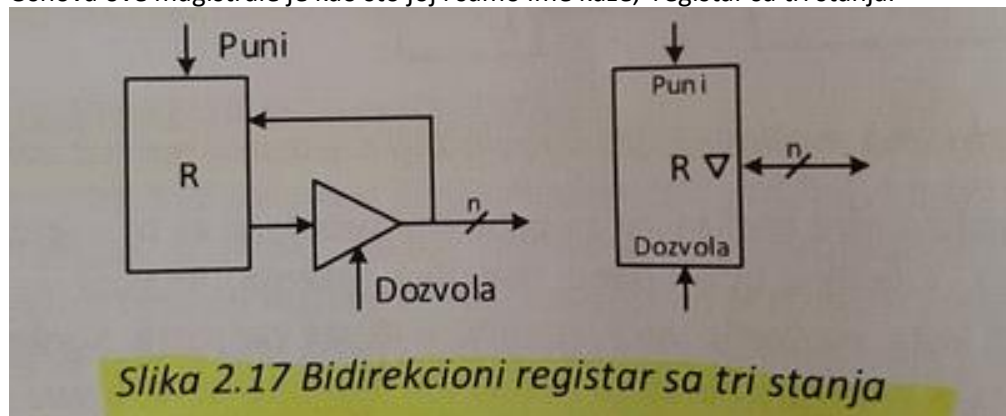
Vektorski mux sa n bita posreduje u prenosu sadržaja iz jednog od polaznih registara do odredisnog registra, pod kontrolom upravljackih signala K_i . Ovakav mux stavlja se ispred svakog od registara u koji se vrši zapis, pa se stoga ovo resenje naziva **magistrala sa dodeljenim multiplekserima**. Broj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

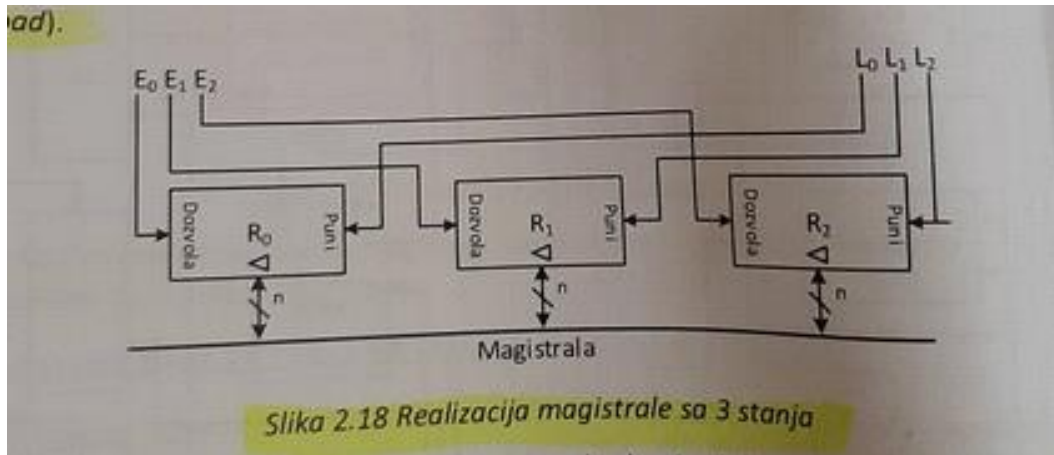
Osnovna mana ovakvog resenja je veliki broj veza, koji je smanjen uvodjenjem **jednostruke magistrale** (usteda broja logickih kola, smanjuje broj potrebnih veza), realizovane takodje pomocu muxa. Ukoliko procesor ima N registara, ovo resenje koristi samo jedan mux sa $N-1$ ulaza a njegov izlaz formira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

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

14. Interna magistrala sa tri stanja ? - strana 46 slika 2.18

Uvodjenjem ove magistrale dovodi se do **smanjenja broja linija za povezivanje** (to joj je osnovni cilj). Osnova ove magistrale je kao sto joj i samo ime kaze, **registar sa tri stanja**.





Slika 2.18 Realizacija magistrale sa 3 stanja

Trece stanje registra je postavljanje svih njegovih izlaza u stanje **visoke impedanse**, cime se prakticno izoluje u odnosu na ostatak sistema. Izlazne linije *tri-state* registra kontrolise signal **DOZVOLA(Enable)**. Prenos sadrzaja polaznog registra na magistralu vrsi se postavljanjem adekvatnog signala dozvole, dok zapis u odredisni registar kontrolisu signali za punjenje registara (**Load**).

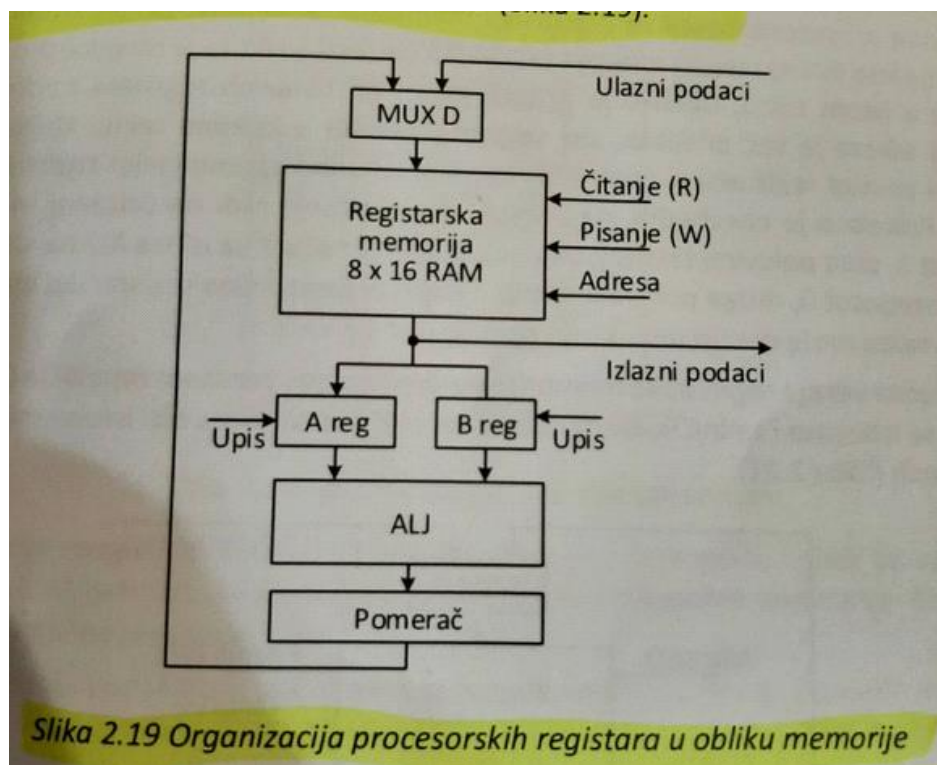
15. Registarska memorija ? - strana 47 slika 2.19

Ekonomičnija 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.

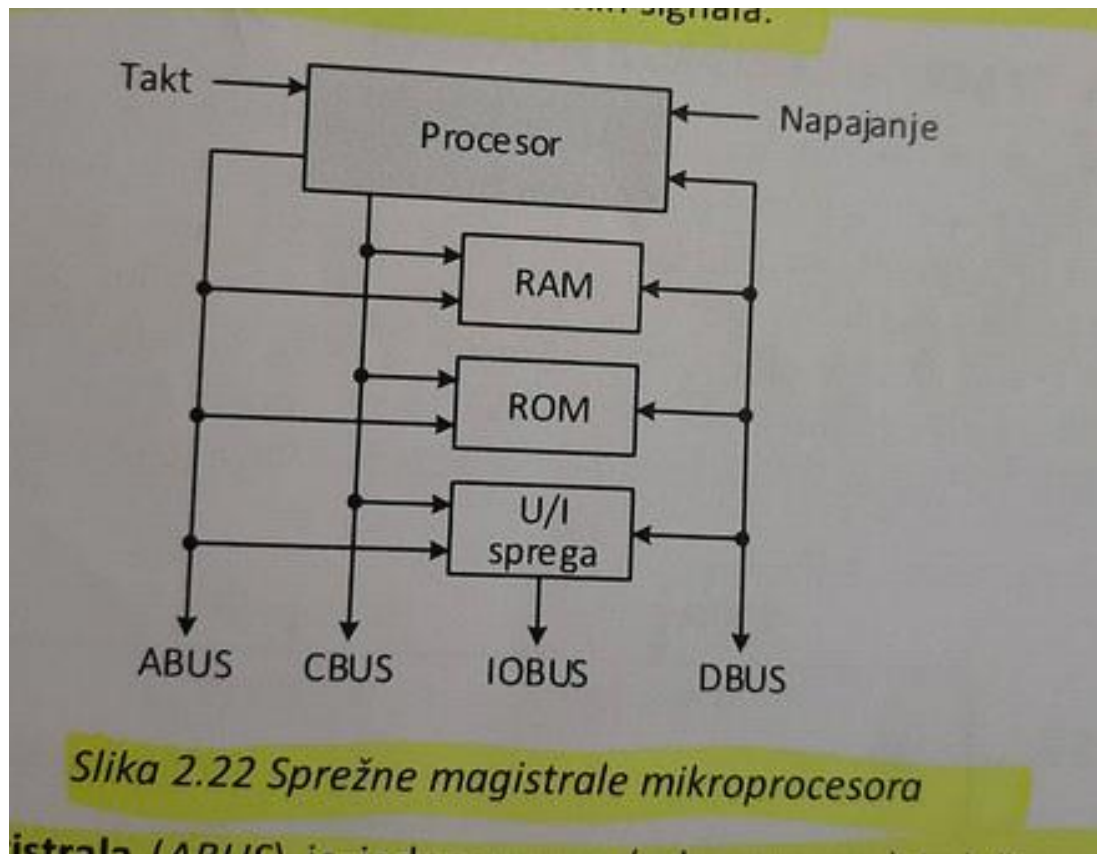
Operativna memorija sadrzi **podatke i instrukcije**, a registarska memorija samo **operande**. Razmena informacija izmedju dvije memorije vrsi se pomocu standardnih programskih instrukcija Load/Store.

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



Kod Magistralne org, komunikacija izmedju registara se obavlja direktno i u jedno taktu, izborom polaznog i odredisnog registra. Ovde 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.

Dva standarda su odje dominantna: **SPI**(Serial-Peripheral-Protocol), **I²C**(Inter-Integrated-Circuit)

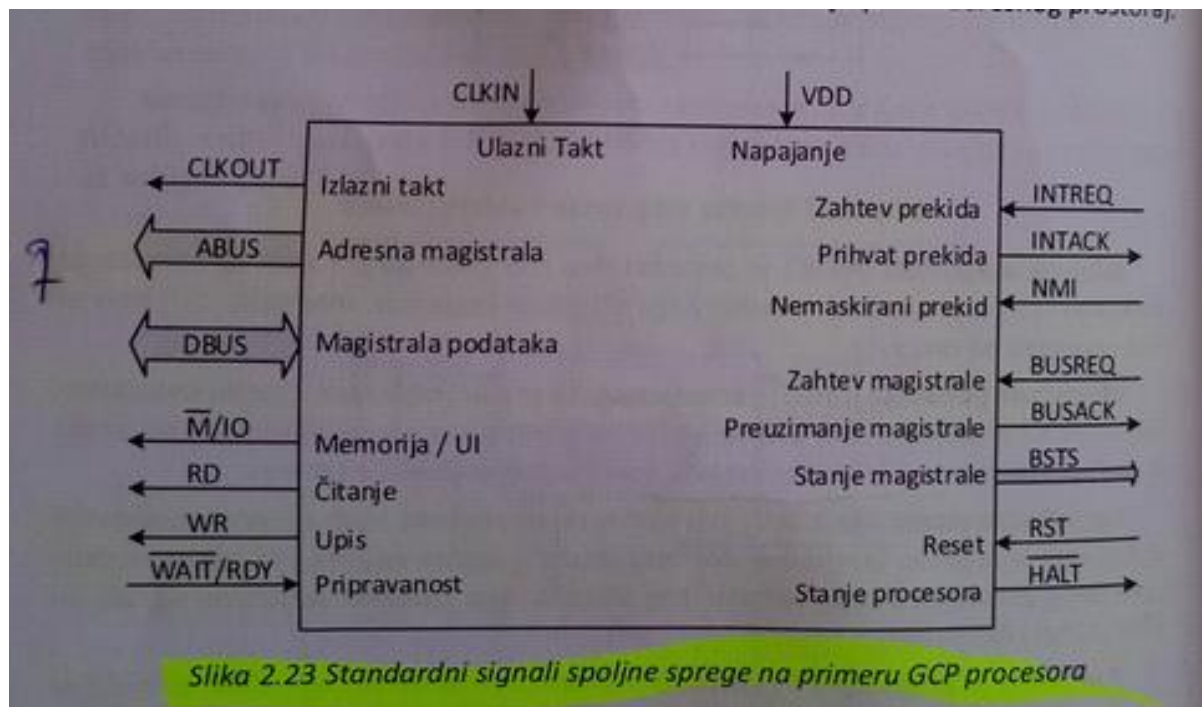


17. Generički centralni procesor ? - strana 50 slika 2.23

Iako virtualan, GCP koirsti sve grupe i tipove signala koji se sreću u realnim mikroprocesorima.

Sustinski je vrlo blizak starijim ili skromnijim mikroprocesorima koji zbog malog broja adresnih i linija podataka ne koriste tehniku mux priključaka.

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 priključci su izlazne linije sa tri stanja.

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

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

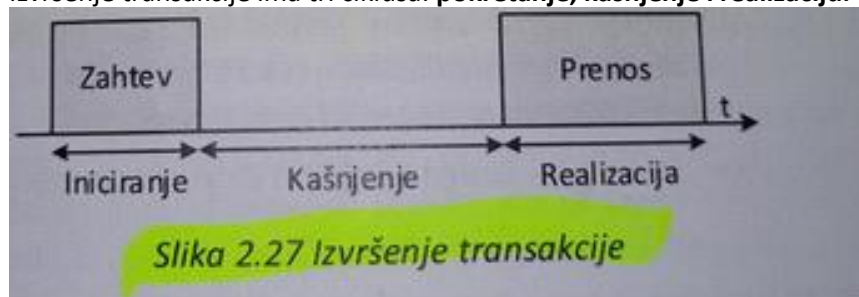
19. Kontrola nad transakcijom (biranje i rukovanje) ?

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, kašnjenje 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.

20. Deljenje transakcija i paralelni prenos podataka(M/IO, RD, WR, RDY) ?

Kod magistrala sa više potencijalnih vodećih modula.

Osnovna ideja je u tome da se vrijeme između zahtjeva i realizacije jedne transakcije iskoristi za pokretanje neke nove transakcije.

Slika 2.31

PARALELNI PRENOS PODATAKA:

Dvije osnovne operacije paralelnog prenosa preko magistrale: **citanje i pisanje**.

Skup upravljačkih signala:

-**Linija M/IO (Memory/IO)** razlikuje pristup memorijskim ili U/I lokacijama.

-**Linija RD (citanje, Read)** - procesor želi da čita memorijski ili U/I registar i da je na adresnoj magistrali već prisutna validna adresa.

-**Linija WR(upis, Write)** - ukazuje na obrnuti smjer prenosa podataka pri čemu su na adresnoj magistrali i magistrali podatka već prisutni validna adresa i podaci koje treba upisati u određeni registar.

-**Signal pripravnosti RDY(Ready)** - uvodi procesor u stanje čeka (**WAIT**) na završetak prenosa podataka.

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.

22. Preuzimanje magistrale (BUSREQ, BUSACK)?

Direktan pristup memoriji (DMA) je tehnika koja podrazumjeva da procesor kontrolu nad magistralom prepusti i perifernim uređajima. To je moguće tek kada procesor završi tekucu transakciju (masinski ciklus) na magistrali, na primjer učitavanje koda instrukcije. U koliko postoji **DMA** zahtjev tj **BUSREQ** je aktivan, procesor će osloboditi magistralu i dozvoliti **DMA** transfer aktiviranjem signala **BUSACK**.

Procesor oslobađa magistralu postavljanjem svih svojih upravljačkih izlaza u treće stanje, stanje visoke impedanse, čime omogućuje preuzimanje istih linija od strane spoljnih komponenti (**DMA kontroler, memorija, i periferni uređaji**)

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

24. Stanje magistrale (BSTS) ?

Savremeni mikroprocesori po pravilu spoljnim uredjajima dostavljaju informaciju o tome šta trenutno rade, tačnije koju transakciju trenutno izvode. To su signali stanja magistrale – **BSTS**.

koji omogućuju jednostavniju i pouzdaniju spregu sa spoljnim uredjajima.

Slika 2.37 (ne treba)

prenos podataka (čitanja ili pisanja). ... I/O indikator, a S1 za razlikovanje smer.

S2	S1	S0	Stanje magistrale (transakcija u toku)
0	0	0	Prihvatanje prekida (Interrupt Acknowledge)
0	0	1	Čitanje U/I (Read I/O)
0	1	0	Upis U/I (Write I/O)
0	1	1	Halt
1	0	0	Učitavanje instrukcije (Instruction Fetch)
1	0	1	Čitanje memorije (Read Data from Memory)
1	1	0	Upis u memoriju (Write Data to Memory)
1	1	1	Pasivna (nema aktivne transakcije)

Slika 2.37 Stanje magistrale mikroprocesora Intel 8086

BUT HAPPO

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 taktni periodi međusobno se razlikuju korišćenjem vremenski pomerenih signala koje postavlja generator vremenskih funkcije.

Mikrooperacije unutar procesa traju 1 takt ali kada izadju vrijeme se produzava. 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 vrsi 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 vrsi 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 isporucuje 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**(zadrzavaju 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 memorije).

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.

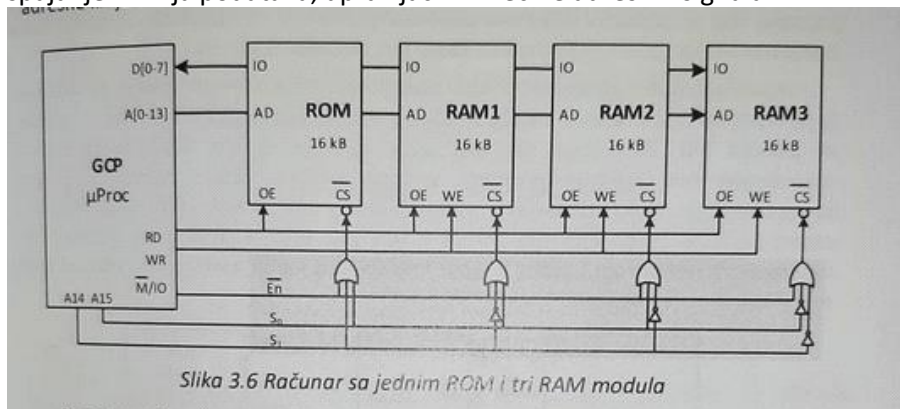
27. Linearna podela adresnog prostora računara ? VEZA PROCESORA I MEMORIJE

Memorijski podsistem cine skup memorijskih modula.

Memorijska mapa-podjela ukupnog adresnog prostora između memorijskih modula po regionima.

Jedan modul mora biti ROM mapiran nad boot vektorom, preostali najcesce RAM.

Povezivanje centralnog procesora i memorijskih kola vrši se posredstvom spoljne magistrale, direktnim spajanjem linija podataka, upravljackih i vecine adresnih signala.



Adresni prostor GCP procesora podijeljen u cetiri segmenta. Prvi segment je rezervisan za ROM, ostali za RAM. Razlikovanje memorijskih modula zahtijeva dvije adresne linije najvise tezine, ostale adresne linije su direktno vezane sa memorijskim kolima.

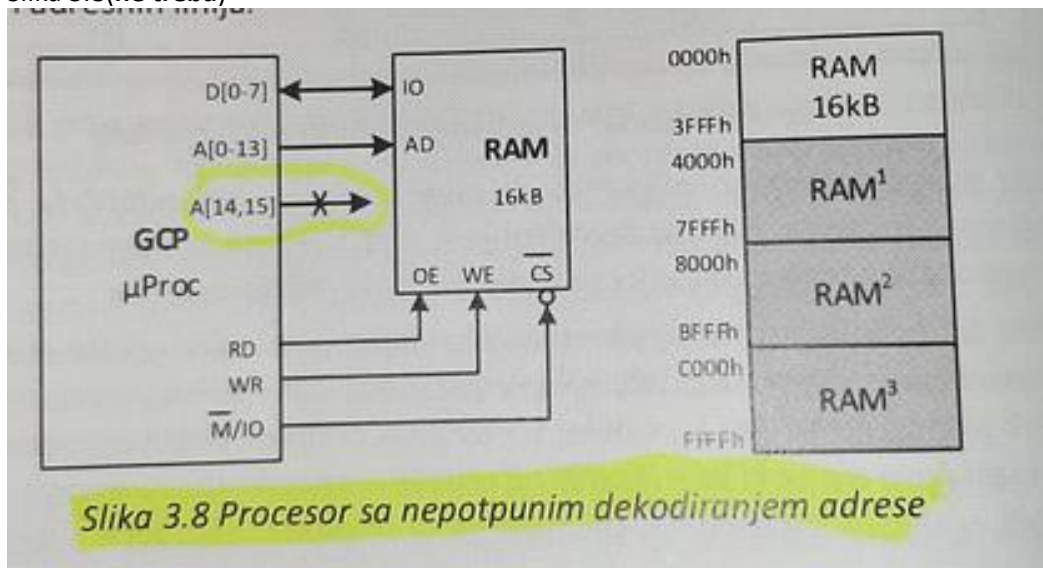
28. Puno i parcijalno dekodiranja adrese ?

Puno dekodiranje – apsolutne adrese. Koriste se sve adresne linije, i realnim modulima prekrilo ukupan adresni prostor procesora.

Nepotpuno (parcijalno) – koristi neke ali ne sve adresne linije.

U slucaju da je kapacitet instalirane memorije relativno mali, za njeno adresiranje nisu neophodni svi adresni biti. Zbog toga se neke adresne linije najvise tezine mogu ostaviti nepovezane, sto rezultuje uštedama u broju spoljnih veza i pojednostavljenjem stampe ploce. Ipak ovo izostavljanje ima i negativne posledice – preklapanje adresa i adresnih opsega. Pri nepotpunom dekodiranju svaka lokacija ima vise od jedne adrese, sto zavisi od broja nekoriscenih adresnih linija.

Slika 3.8(ne treba)



Ne koriste se dvije najvise adresne linije. Zbog toga se isti RAM modul cetiri puta pojavljuje u memorijskoj mapi: prvi put kao realna memorija, i jos tri puta kao kopija.

MEMORIJSKA MAPA PROGRAMA : 27

Kako procesor ima više memorija, različitih ne samo po adresi nego i mogućnosti pristupa (RAM ili ROM), izvršni program mora biti prilagođen memorijskoj mapi. To se sve radi u završnoj fazi povezivanja (linking) programskog koda, pomoću posebne direktive **MEMORY**. Njen sadržaj definiše ime, početnu adresu i dužinu programskih segmenata.

ORGANIZACIJA MEMORIJE SA SLUCAJNIM PRISTUPOM

Osnovna gradivna jedinica memorije je **memorijska ćelija**, memorijski element koji čuva jedan bit informacije. Kada je selektovan memorijski element će izvršiti operaciju čitanja ili upisa, zavisi šta je traženo.

Zbog velikog kapaciteta memorije, interne veze unutar memorijskog čipa moraju biti maksimalno optimizovane što vodi ka matricnoj organizaciji. Skraćivanje internih veza doprinosi uštedom prostora i ubrzanje pristupa. U slučaju vrlo velikog kapaciteta gdje bi jedna matrica takodje bila prevelika, uvodi se 3D organizacija sa više fizičkih stranica.

Slika 3.10.

29. SRAM I DRAM ?

SRAM (Static Random-Access Memory) – najviša brzina, umjerena gustina pakovanja na memorijskom čipu. To je izvorna poluprovodnička memorija, koja koristi bistabilno kolo (flip-flop) kao osnovu memorijske ćelije. **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 veći kapacitet. **Slika 3.14**

Minimizuje prostor potreban za realizaciju jedne memorijske ćelije jer jedan bit informacije čuva u minijaturnom kondenzatoru unutar integrisanog kola. Logička 1- napunjen, logička 0- prazan. Informacije zapisane u DRAM ćeliju nije stabilna (statička kao kod SRAM, zbog curenja struje preko tranzistora i praznjenja kondenzatora jedinica upisane u njega bi vremenom nestale). Glavno unapređenje DRAM memorije je uvođenje sinhronog pristupa **SDRAM**. **Slika 3.15.**

Zahvaljujući svojoj prednosti u pogledu kapaciteta i cijene, dinamička RAM je danas osnova memorija savremenih računarskih sistema.

MEMORIJSKI KONTROLER:

Povezivanje procesora sa svojim memorijama najčešće zahtjeva razvoj posebne sprezne jedinice koja se naziva **memorijski kontroler**. Njen zadatak je da uskladi različite fizičke protokole prisutne pri povezivanju procesora i memorijskih kola. Može biti kao posebno integrisano kolo, ili kao dio samog mikroprocesora.

30. Asocijativna memorija ?

Za veoma brzu pretragu, provjera po spoljnom ključu. U nekim primjenama neophodna je vrlo brza pretraga memorije kako bi se utvrdilo da li je sadržaj neke o njenih lokacija istovetan spoljnom ključu. Ako jeste, u slučaju poklapanja od interesa je adresa pronađene lokacije.

Zbog ovoga SRAM memorije se opremaju dodatnom logikom, koja može da pretražuje po sadržaju (asocijativnim pristupom), u jednom taktu. Ovakva memorija se naziva adresabilnom po sadržaju (**CAM**). Binarna i ternarna CAM (koriste samo gdje se ne mogu zamjeniti).

Binarna- ključ i sadržaj predstavljeni serijom nula i jedinica (bita 1/0).

Ternarna- složenija logika koja rukuje i trećim logičkim 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 MEMORIOM: označava postupke i tehnike dodjele (alokacije) operativne memorije koje su prisutne kao i način na koji krajni korisnik vidi memoriju koja mu se ustupa.

Na tom planu razvijen je koncept logičke adrese, koji je zbog svojih prednosti danas dominantan. Sa stanovista procesora i računarskog sistema rukovanje memorijom označava dio hardvera koji prati i omogućuje efikasnu alokaciju memorije, koji u opstem slučaju poznat kao **jedinica za rukovanje memorijom (MMU)**. **Slika 3.23**

MMU rješava adresne reference **prevodjenjem logičke u fizičku adresu** i kontrolise pristup podacima.

OSNOVNI ZAHTEVI:

Operativni sistem je program koji upravlja izvršenjem aplikativnih programa i deluje kao sprega između korisnika i fizičkih elemenata računarskog sistema. Svoju funkciju OS ostvaruje kontrolom fizičkih i logičkih resursa računarskog sistema.

Pristup procesoru i memoriji korisnici ostvaruju pomoću procesa koje pokrene OS.

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

Savremeni OS dele se u dvije osnovne klase: **jednokorisnički** i **visekorisnički**.

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

U **jednoprogramskom** sistemu, memorija se dijeli na **sistemske** i **korisnički dio**. Sistemski dio koristi operativni sistem, a drugi je u cjelosti raspoloživ aplikativnom. U **visekorisničkom** sistemu, ovaj drugi dio se dalje dijeli u nastojanju da se prihvati više procesa, odnosno usluži više korisnika. U cilju podrške ovom zahtjevu, operativna memorija i procesor se opremaju dodatnim fizičkim komponentama.

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

32. Klasifikacija tehnika rukovanja memorijom - vrste i kratko objašnjenje ?

Najjednostavnija je pojedinačna dodjela.

Prema mehanizmu dodjele memorije, multiprogramске sisteme možemo klasifikovati u dvije kategorije:

- **Multiprogramске sisteme kod kojih je uvijek cijeli adresni prostor programskog procesa prisutan u radnoj ili spoljnoj memoriji.** Memorija se dijeli na particije ili stranice – **partitivnom** i **stranicnom** dodjelom.
- **I na multiprogramске sisteme kod kojih je jedan dio adresnog prostora smješten u radnu memoriju dok je ostatak na disku. Ovo je **virtualna memorija**.**

POJEDINACNA DODJELA MEMORIJE: korisničkoj aplikaciji se dodjeljuje jedna kontinualna zona memorije koju može koristiti. **Slika 3.25**

PARTITIVNA DODJELA MEMORIJE: dijeli memoriju u particije, kontinualne zone memorije koje se dodeljuju svakom od procesa. Unutar particije nalazi se sve što je vezano za proces: programski kod, podaci i druge kontrolne informacije. Velicina particije zavisi od potrebe datog procesa, i može se dinamički mijenjati. Particija se alokira na početku, realocira u toku izvršenja i oslobadja na kraju života procesa.

Fragmentacija - negativna posledica. **Kao posledica vremenskog redosleda punjenja, zamjene i završetka procesa, u memoriji se javljaju brojne neiskoriscene zone (fragmenti), premale da se u njih napuni novi proces.** Zato OS mora povremeno pokrenuti korektivnu proceduru koja vrši sazimanje slobodnih dijelova, **defragmentaciju** memorije.

Dva rješenja, **prvo** – azuriranje i ponovno podesavanje svih adresnih osjetljivih elemenata. **drugo** - dinamička relokacija.

STRANICNA DODJELA MEMORIJE :

Memorija u logičkom adresnom prostoru je kontinualna, ali u fizičkom rasuta – stranicarenje.

U osnovi ove tehnike je podjela logičkog adresnog prostora procesa, kao i realne memorije na dijelove iste velicine – stranice i blokove. Dodjela stranice podrazumjeva pronalazjenje slobodnog bloka u realno

memoriji, na bilo kojoj lokaciji, i njeno pridruživanje datom procesu. To zahtjeva dvije tabele koje održava OS. Prva je globalna **Lista Slobodnih Blokova (LSB)** koja na nivou sistema prati alokaciju blokova procesima. **Tabela Preslikavanja Stranica (TPS)** je vezana za sam proces, i u njoj se za svaku stranicu čuva realna adresa njoj dodjeljenog bloka.

33. Virtualna memorija ?

Virtualna memorija - adresiranje memorijskog prostora veće od realne operativne memorije.

Osnovna ideja virtualne memorije je da procesor generise adrese u opsegu znatno širem u odnosu na raspoloživu fizičku memoriju, ali da svi dijelovi virtualnog adresnog prostora ne moraju biti istovremeno prisutni u operativnoj memoriji. Onaj dio koji nije trenutno potreban čuva se u sekundarno memoriji, odnosno magnetnom disku.

Stranicna dodjela uvodi dva ključna elementa za virtualnu memoriju: logicku adresu i podjelu memorije na neke dijelove kojima se može 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 može 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 dinamičku dodjelu realne memorije stranicama iz virtualnog adresnog prostora procesa.

U slučaju da je stranica u operativnoj memoriji ($V=1$), realna adresa se računa na isti način kao kod klasičnog stranicenja. Broj bloka B učitani iz tabele TPS, je istovremeno njegova početna adresa, čijim 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 smještena stranica. Ovaj događaj se naziva greskom stranice (page fault) i posredstvom prekida poziva komponentu OS zaduženu za premještanje stranica. Njen prvi zadatak je traženje slobodnog bloka. Ako nije moguće onda se primjenjuje algoritam za oslobađanje. 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 različite dužine. Segmente definiše korisnik ili kompajler, shodno namjeni i svaki od njih ima svoju identifikaciju, dužinu i nezavisni adresni prostor.

Segment sadrži instrukcije ili podatke, sa kontrolisanim pristupom. Segment može biti privatni ili dijeljen, sa dozvolom čitanja, pisanja, ili izvršenja (RWE- Read, Write, Execute).

Razlikovanje tipova korisnika vrši se posredstvom tzv. nivoa privilegija i prstena zaštite. Dva osnovna rezima rada: sistemski i aplikativni. Procesori Intelove x86 familije, koriste sledeće segmente i registre:

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

Virtualna adresa se sastoji od broja segmenta i relativnog odstojanja unutar segmenta.

Segmentacija, ova što 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 dodjela stranica**, kod koje se segment stranici, 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 pristupne u radnoj memoriji.

ČETVRTI DEO - Ulazno-izlazni podsistem računarskog sistema

34. Indirektni i direktni prenos ulazno-izlaznog podsistema ?

Indirektni prenos izmedju U/I podsistema i memorije – **procesor je posrednik**. Dva slucaja:

Prenos podataka pocinje na zahtjev procesora-ova tehnika poznata kao programirani U/I ispituje stanje svake periferne jedinice i pokrece izvršenje adekvatnih operacija. Ispitivanje stanja podrazumijeva periodičnost obraćanja perifernim jedinicama zbog čega se naziva i tehnikom prozivanja.

Prenos podataka pocinje na zahtjev U/I jedinice – Radi se o prenosu po prekidu (interrupt) rada centralnog procesora, što rezultuje znatno boljim iskoriscenjem procesorskog vremena.

Direktan prenos- **bez procesora osim na pocetku i kraju procesa**, to je direktan pristup memoriji (DMA).

SPREZNI SISTEM:

U/I kontroler upravlja tokom podatka i kontrolnim informacijama, opremljen skupom registara vidljivih procesoru i dodatnom logikom koja je orjentisana ka U/I uređaju. **Slika 4.1.**

Komunikacija sa U/I kontrolerom se odvija pomocu registara koji moze biti postavljen u 3 kategorije:
prva kategorija – **registri podatka**(komunikacija sa perifernom jedinicom) mogu biti **ulazni** ili **izlazni**.

druga kategorija – **registar stanja**(na upit procesora dostavlja stanje periferne jedinice i eventualnu potrebu za obradom)

treća kategorija – **upravljacki registar**(realizuju se komande centralnog procesora)

Slika 4.2.

Centralni proc kontrolu nad perifernim jedinicama ostvaruje posredstvom komandi koju salje U/I a to su: **upravljacke**(komanduju radom periferne jedinice), **upitne(test)**(utvrđuje se u kom je stanju periferna jedinica), **slanje podataka**(pracenje da li je prenos završen), **prijem podataka**(ako je utvrđeno postojanje podataka u U/I uređaju).

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

Povezivanje procesora sa U/I uređajima, 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 uređaja 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.

36. Programiran ulaz-izlaz ?

Kod programiranih U/I aktivnosti, program koji se zove **U/I rukovaoc (IO driver)** periodicno ispituje stanje svake U/I jedinice, ucitavanjem i analizom sadrzaja registara stanja. **(Slika 4.5) ne treba** Ukoliko je periferna jedinica spremna za prenos, procesor ce je opsluziti izvršavanjem adekvatne programske procedure. U suprotnom, procesor izvrsava druge programske aktivnosti. U oba slucaja, sve aktivnosti inicira centralni procesor.

Ovakav nacin rada sa U/I uredjajima se naziva **prozivanjem (polling)** zbog upornog periodicnog pristupa registru stanja.

37. Ulaz-izlaz na bazi prekida i klasifikacija prekida ?

Prekidi(interrupts) - pojavili se za povecanje efikasnosti rada racunarskog sistema.

Popravljaju (poboljšavaju) problem provjere stanja perifernih jedinica.

Osnovna ideja uvođenja ovog mehanizma je iniciranje prenosa podataka od strane periferne jedinice.

Prekidna rutina koja se poziva poput potprograma, izvršava samo kod nepohodan za realizaciju prenosa.

KLASIFIKACIJA PREKIDA - prema dva kriterijuma: **ko ih generise, kako se odredjuje adresa prekidne rutine.**

Po prvoj: fizicke i programske.

Po drugoj: vektorske, nevektorske.

Fizicki mogu biti:

Spoljni - izazvani od strane eksternih U/I uredjaja, to su „pravi“ prekidi. Mogu biti *maskirajuci* ili *nemaskirajuci*.

Unutrasnji (interni) – su produkt internih komponenti centralnog procesora, najcesce kao posledica nepravilnog izvršenja instrukcija. Oni se oznacavaju kao *zamke ili izuzeci (trap, exception)*,

Uzroci internih: overflow, deljenje sa nulom, nepravilna adresa, prekoracenje steka, narušavanje prava pristupa.

Programski (softverski) prekidi se izazivaju izvršenjem posebne instrukcije (int).

Prekid se izvršava slicno potprogramu, pri cemu fizicke komponente (hardver) obezbedjuje adresu grananja. Dva nacina izbora ove adrese:

Nevektorski metod podrazumeva da je adresa grananja fiksna memorijska lokacija, nakon cega je neophodna programska potraga za uredjajem koji je prekid izazvao.

Vektorski metod je resenje kod koga izvor prekida, posredstvom magistrale podataka, procesoru salje vektor prekida kao svoju identifikaciju. Na osnovu te informacije, procesor moze da odredi ili pronadje adresu prekidne rutine.

38. Izvršenje prekida ?

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

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

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

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

39. Dijeljenje sistema prekida ?-----

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

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

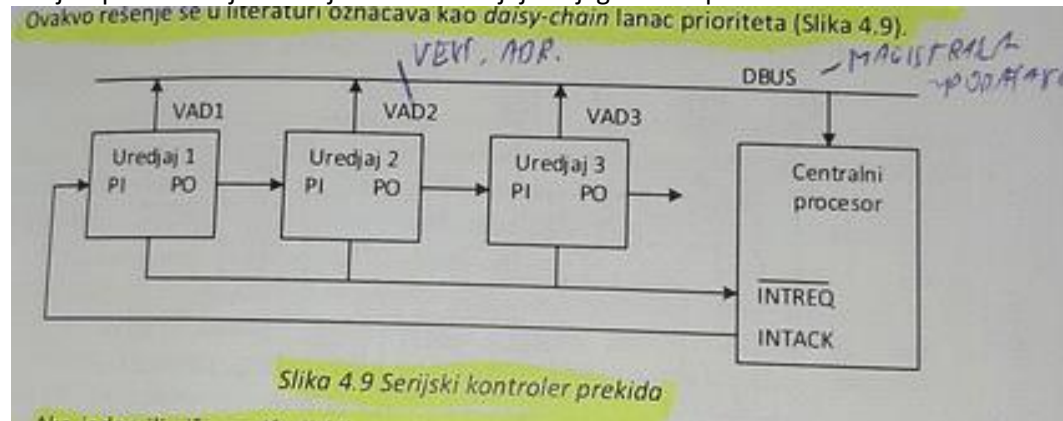
Programski metod - nevektorski sistem, jedna je zajednička prekidna rutina koja prozivkom utvrđuje izvor prekida. Prioritet je definisan u programskoj prozivci.

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.

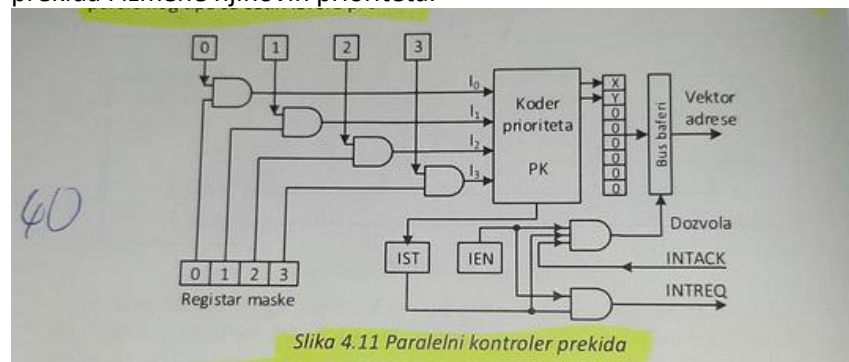
Uređaj koji će biti uslužen zavisi od pozicije u lancu i na osnovu signala PI (Priority in) i PO (Priority Out).

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 zahteva za prekid. Dozvola prekida

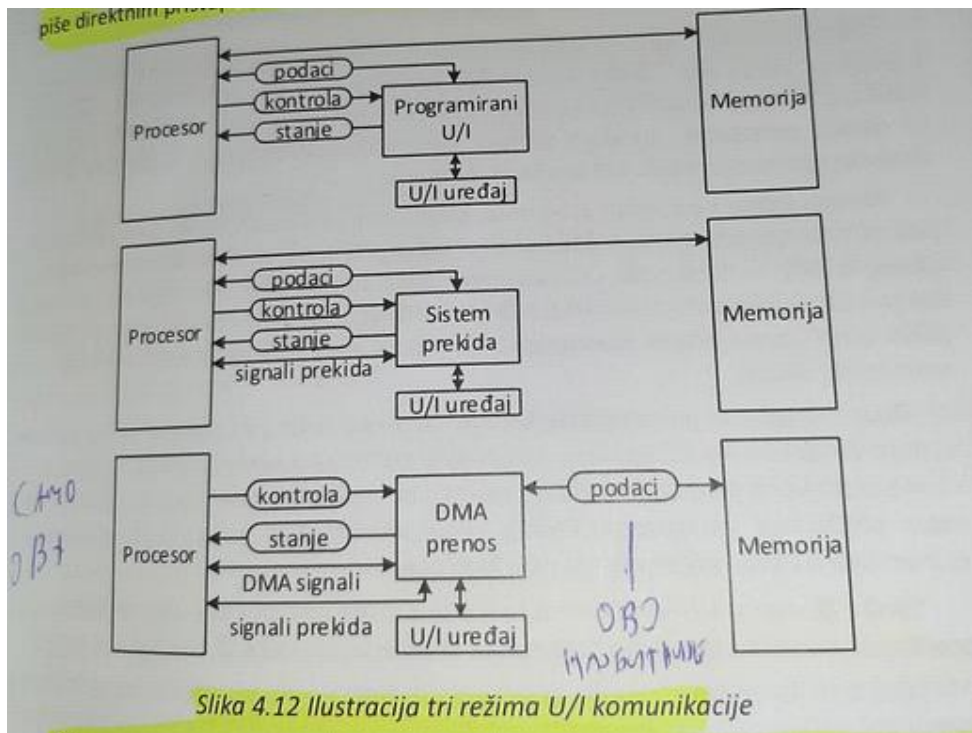
IEN (Interrupt Enable) je upravljački signal kojim se kontrolisu svi prekidi povezani na isti prekidni kontroler.

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).

41. Direktan pristup memoriji (DMA) ? - strana 103 slika 4.12 + strana 104 slika 4.13

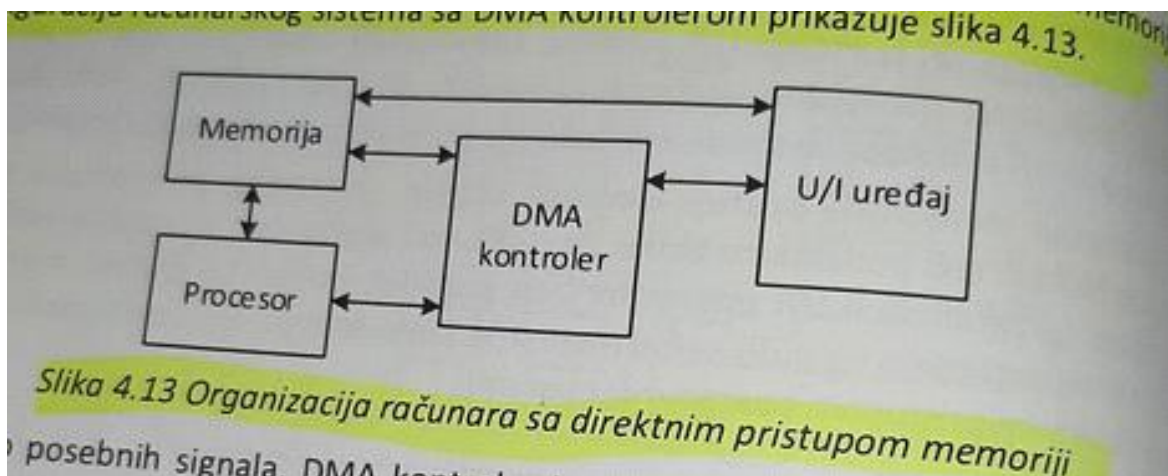
DMA prenos karakterise blokovski prenos i mnogo manje kasnjenje pri izvršavanju U/I zahtjeva. Zbog toga procesor do podataka dolazi brže i uz minimalne dodatne aktivnosti. **DMA eliminise ucesce procesora u prenosu svakog bajta ili reci podataka u bloku.** Podaci ne prolaze kroz registre centralnog procesora na putu ka/od memorije. Umjesto toga, U/I uređaj podatke čita ili piše direktnim pristupom operativnoj memoriji.



Slika 4.12 Ilustracija tri režima U/I komunikacije

DMA KONTROLER I KRADJA CIKLUSA:

DMA kontroler - njen zadatak je posredovanje izmedju perifernog uređaja i centralnog procesora, radi zahtevanja pristupa i kontrole nad spoljnom magistralom u toku razmjene U/I podataka sa operativnom memorijom.



Kradja ciklusa - sustina DMA tehnike je upravo preuzimanje memorijskih ciklusa procesora.

IZVRSENJE DMA CIKLUSA:

Sustina DMA ciklusa je preuzimanje spoljne magistrale, gdje **DMA** kontroler postavlja adresne i upravljacke signale, a U/I uređaj ili memorija linije podatka. Procesor se ipak angazuje prije pocetka i po završetku **DMA** ciklusa. U prvom koraku, procesor omogućuje **DMA** prenos zadavanjem parametara neophodnih **DMA** kontroleru a to su:

- pocetna adresa zone podatka u memoriji(start address).
- broj znakova, tj. duzina bloka koji se prenosi(byte count).
- smjer prenosa u odnosu na memoriju, citanje ili upis (transfer mode).
- komanda za pocetak (start **DMA**)

Veza **DMA** kontrolera sa svakom od perifernih jedinica koristi par linija: **zahtev**(DMA Request, DREQ) i **potvrdu**(DMA Acknowledge, DACK). Prvu liniju aktivira periferni uređaj, a drugu kontrolise **DMA** kontroler.

Koordinacija sa procesorom koristi dva sprezna signala: **zahtev za magistralu**(Bus request, BUSREQ) i **prihvat magistrale**(Bus acknowledge, BUSACK).

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 izvrsenju 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, što česce čita kopiju potrebnih podataka koja se čuva u skrivenoj memoriji.

UBRZANJE PROCESORA - tehnike:

***vektorski procesor** (standardni skup instrukcija proširen vektorskim).

***paralelna obrada** (označava 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).

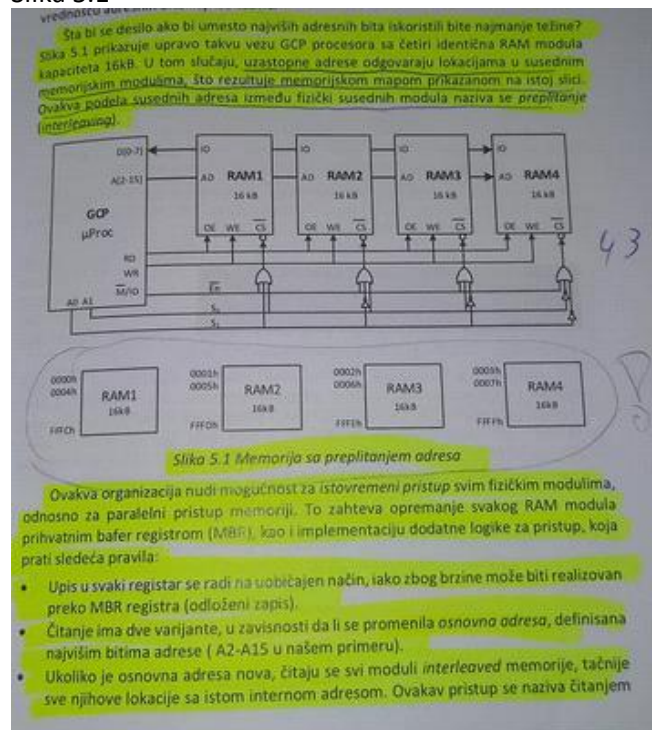
***protocna obrada** (sa osnovnom idejom o istovremenom izvršenju više instrukcija ali u različitim fazama izvršenja).

UBRZANJE RACUNARA - multiprocesorski računari koriste najmanje dva procesora, koji podjelom ukupnog opterećenja omogućuju brze izvršenje korisničke obrade.

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

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

Slika 5.1



u nizu, gde je dužina niza jednaka broju fizičkih modula. Učitani sadržaj se čuva u MBR registrima.

Svi novi zahtevi za čitanjem u istom nizu, odnosno sve dok se osnovna adresa ne promeni, zadovoljavaju se prenosom sadržaja iz MBR registara. Kako su ovi registri značajno brži u odnosu na memorije DRAM tipa, uštede u vremenu su značajne.

Ovakva organizacija omogućuje 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)

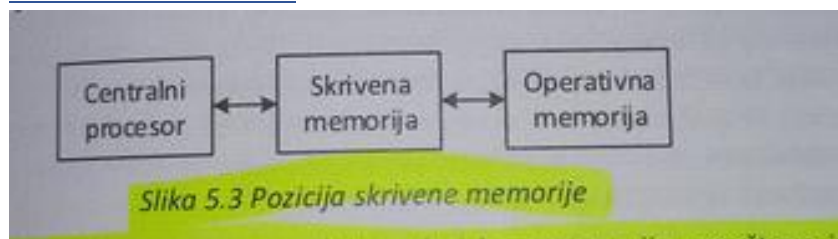
-ukoliko je osnovna adresa nova, citaju se svi moduli *interleaved* memorije, tacnije sve njihove lokacije sa istom internom adresom. Ovakav pristup se naziva citanje u nizu, duzina niza jednaka broju fizickih modula. Ucitani sadrzaj se cuva u MBR registrima.

-svi novi zahtjevi za citanjem su u istom nizu, odnosno sve dok se osnovna adresa ne promijeni, zadovoljavaju se prenosom sadrzaja iz MBR registra. Kako su ovi registri znacajno brzi u odnosu na memoriju DRAM tipa, ustele u vremenu su znacajne.

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

Slika 5.2

44. Skrivena memorija, skrivena memorija sa više nivoa ? - strana 113 slika 5.3 i strana 114 slika 5.4



SKRIVENA MEMORIJA (cache memory) je mala i brza memorija, smjestena izmedju procesora i operativne memorije. Po pravilu je u SRAM tehnologiji, jer je brza od DRAM memorije.

Skrivena memorija je na pocetku rada prazna i mora se napuniti. Cilj skrivene memorije je da srednje vrijeme pristupa memoriji vidjeno sa strane procesora priblizi vremenu pristupa skrivenoj memoriji, osnovna ideja ovog pristupa je da procesor, umjesto stalnog pristupa operativnoj memoriji, sto cesce cita kopiju potrebnih podataka koja se cuva u skrivenoj memoriji)

SKRIVENA MEMORIJA SA VISE NIVOVA:

Efikasnost skrivene memorije je uslovljena njenom brzinom. Zato je najefikasnija ako je smjestena sa procesorom u istom integrisanom kolu(ista tehnologija izrade i brzine rada). To je interna ili skrivena memorija **prvog nivoa** (Level 1, L1) ciji je kapacitet ogranicen prostorom na cipu. Spoljna skrivena memorija **nivoa dva** (Level 2, L2), smjestene na matichnoj ploci. Neki procesori koriste i **tri** ili vise nivoa(savremeni procesori – multi-core), od kojih je svaki sledeci nivo veci ali i sporiji, povecava kapacitet i sluzi primarno kao prihvatni bafer internoj memoriji.

Slika 5.4.

(skrivena memorije za rukovanje instrukcijama i podacima, nisam to ucio/pisao/podvlacio)

NE TREBA.

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 tezine direktno iz izvorne adrese reci u operativnu memoriju.

Ako skrivena memorija ima $S=2^k$ rijeci, operativna memorija se moze podijeliti u blokove iste velicine ciji 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 fizicke realizacije.

Nedostatak - nagli pad procenta uspjesnosti pronalazenja adrese u skrivenoj memoriji ako se desi da se dva najcesce koriscena 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 znaci da oznaka mora biti puna adresa (sem Byte adresnih bita).

Optimalno rjesenje za pretrazivanje memorije oznaka postize se asocijativnom organizacijom skrivene memorije.

Upis i citanje oznaka u memorijski blok se vrsi na konvencionalan nacin. Dodatno je u svakom redu memorijskih celija pridruzena logika za poredjenje upisane oznake sa odgovarajucim bitima adrese koju je generisao centralni procesor.

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

- Na slucaj (generator slucajnih brojeva),
- Zamjenom najstarijeg clana liste (FIFO),
- Zamjena clana koji najduze nije bio u upotrebi (LRU)

47. Skrivena memorija sa asocijativnim skupom ?

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

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

Ista ova metoda se prosiruje ucitavanjem vise reci istovremeno, odnosno ucitavanjem niza (line). Efikasnost ovakovog resenja zahtjeva prosirenje prenosnih puteva izmedju centralnog procesora, skrivene i operativne memorije.

48. Metodi zapisa u sistem sa skrivenom memorijom ?

Tri moguće metode:

- upis rezultata u glavnu memoriju,
- upis rezultata u skrivenu,
- upis rezultata u obje.

UPIS ZA PROLAZOM (write – trough) : je metoda koja podatak uvijek upisuje u operativnu memoriju, a u skrivenu samo ako je data lokacija vec tamo. Vrijeme pristupa OM se moze nadoknaditi tehnikom baferovanja ulaza (write buffering) cime se procesoru omogucuje nesmetan nastavak rada, koji azurnu kopiju koristi iz skrivene memorije paralelno sa zapisom podataka koji izvrsava navedena komponenta.

UPIS SA ZADRSKOM (write – back, copy – back) : u slucaju pogotka zapisuje podatak samo u skrivenu memoriju. U suprotnom ako je zapisu prethodio promasaj podatak se upisuje u operativnu memoriju. Da se ne bi gubilo vrijeme za citanje i uporedjivanje sa podacima iz OM uz svaku oznaku u SM se cuva dodatni kontrolni bit (dirty bit). Ovaj bit oznacava promenu (zaprljanost) podataka uz oznaku.

49. Vektorski procesor ?

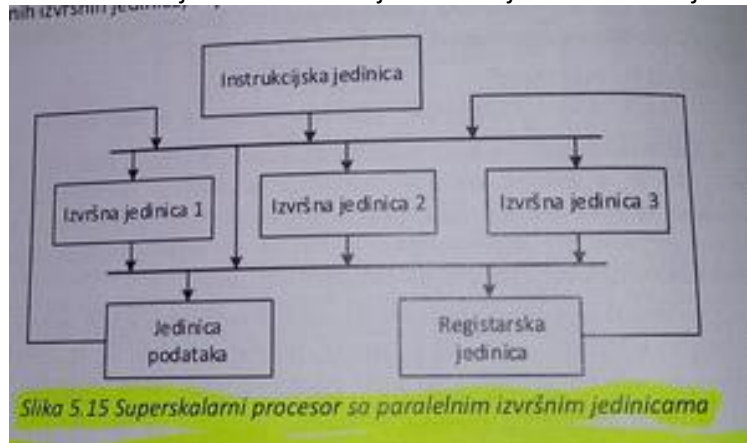
Optimizovan za sekvencijalno(uzastopno) izvršenje iste operacija ali nad nizom operanada(vektorom).

$C[i] = A[i] \times B[i]$ gdje je x 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 cinjenicu da izvršenje instrukcije vrlo cesto traje duze od njenog ucitavanja.



Umjesto jedne, **superskalarni racunari** koriste umnozene izvršne jedinice, i paralelnu obradu ostvaruju distribucijom instrukcija i podataka ka njima. Instrukcijska jedinica zaduzena je za ucitavanje i pokretanje instrukcija, cije se izvršenje, pod kontrolom upravljacke 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 ograniceno je brojnim faktorima, i rijetko se u realnim uslovima moze pribliziti teorijskom limitu. U vecini prakticnih realizacija postoji neki stepen specijalizacije izvršnih jedinica pa se te instrukcije usmjeravaju ka jednoj od njih.

Superskalarni procesori primjenju kompleksne tehnike dinamicnog rasporedjivanja instrukcija uz sve prednositi i probleme.

51. Procesor sa vrlo dugom instrukcijskom rečju ?

Procesor sa vrlo dugom instrukcijskom recju (VLIW – Very Long Instruction Word) u proces rasporedjivanja ukljucuje i programsku podrsku, kroz namjenski kompajler koji vrsi pripremu koda radi tzv. statickog rasporedjivanja instrukcija. Prilikom prevodjenja programa, dakle izvan ogranicenja realnog vremena, kompajler ispituje paralelizam izmedju instrukcija i njihovim kobinovanjem formira jednu **dugu instrukciju**, koja se pokrece u istom trenutku i paralelno izvršava. Duga instrukcija ukljucuje onoliko instrukcija, koliko **VLIW** procesor ima izvršnih jedinica. Zato je potreban format duge instrukcije koji obuhvata veci broj polja, od kojih svako definise po jednu primarnu instrukciju. Ukoliko neko od polja iz bilo kog razloga ne moze biti popunjeno korisnom instrukcijom, u njega se upisuje NOP funkcija. Dakle duge instrukcije, kao skup paralelno izvršivih osnovnih instrukcija zahtevaju veliku duzinu instrukcione rijeci, od cega i naziv ove klase procesora.

Koncept **VLIW** procesora predstavlja svojevrstu generalizaciju postupka horizontalnog mikroprogramiranja nad arhitekturom/organizacijom superskalarnih procesora.

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**

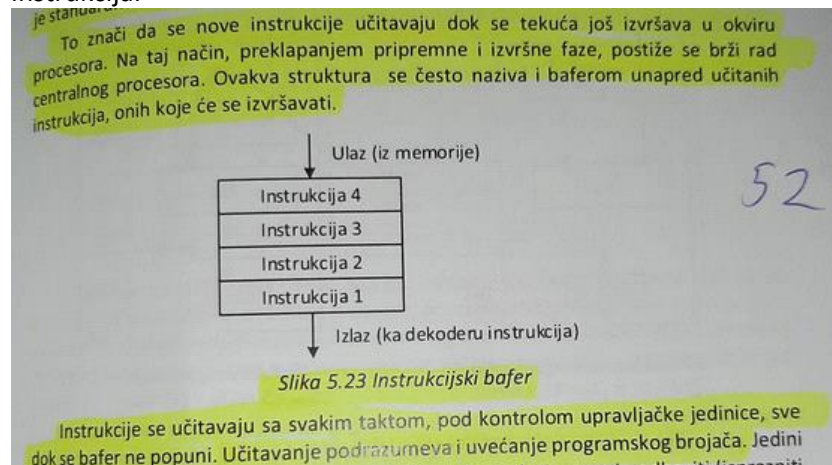
Protocna obrada (pipeline processing) – svi sekvencijalni procesi koji se izvršavaju po modulu pokretne trake gdje se dijelovi ukupnog procesa (fragmenti) realizuju na uzastopnim djelovima postrojenja (segmentima). Na nivou postrojenja, svi segmenti se izvršavaju istovremeno. Obrada neke jedinice počinje 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 podeliti 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.

PROTODNO UCITAVANJE INSTRUKCIJA:

U protodnoj organizaciji procesora, **instrukcijski bafer** je standardna struktura za protodno ucitavanje instrukcija.

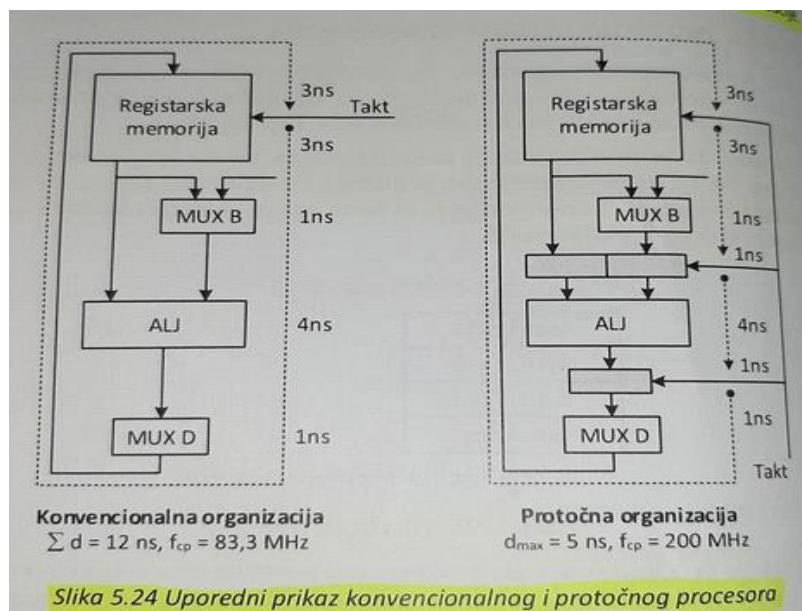


Nove instrukcije se ucitavaju dok se tekuća izvršava, ovo se još zove baferom unapred ucitanih instrukcija. Segment koji se bavi ucitavanjem instrukcija obično se označava kao **IF – Instruction Fetch**.

53. Segmentacija internih prenosnih puteva ? - strana 134 slika 5.24

Radi protodnog izvršenja instrukcija neophodno je izvršiti **segmentaciju procesora**, tačnije njegovih prenosnih puteva između registara i funkcionalnih jedinica.

Slika 5.24 – dat je uporedni prikaz konvencionalnog i protodnog procesora



Segmentacija procesora u ovom slučaju je urađena dodavanjem izolacionih registara ispred i iza aritmeticko - logicke jedinice. Tako je omogućeno izvršenje instrukcije u tri segmenta koji se konkurentno obavljaju: **pribavljanje operanda** (data operand fetch DOF), **izvršenje instrukcije** (execution, EX), i **zapis rezultata u registarsku memoriju** (write back WB). Treba primjetiti da segmentacija, sem uštede vremena zbog konkurentnog izvršenja instrukcije, omogućuje i povećanje radnog takta procesora.

54. Rizik podataka i rizik grananja ?

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

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

Programsko rešenje se zasniva na umetanju NOP(no-operation) instrukcije između kritičnih instrukcija.

Hardversko - zadržavanje(stall), hardversko proširenje upravljačke jedinice logikom koja vrši detekciju operanda koji još nije zapisan pa se umetanje NOP-a vrši automatski.

Ukoliko negdje u procesoru, a pre svega u izolacionim registrima protodne strukture, postoji potreban podatak, ispravan ali još nezapisan rezultat, on se može isporučiti direktno korisniku pre njegovog regularnog zapisa u određeni registar. Korisnik je naravno segment koji vrši referencu na isti rezultat.

Ova tehnika se označava kao prosleđivanje 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)**.

Dinamičke metode - instrukciji grananja treba dodjeliti obeležje - status, one se zasnivaju da povećaju procenat pogodjanja, pamćenjem istorijata uslovnih grananja. Osnovna ideja jeste da instrukciji grananja

treba dodjeliti obeležje - status (jedan ili više bita) koji će odrazavati način njenog izvršenja u prethodnom periodu.

Dvije osnovne tehnike:

1) Indikator grananja - skup bita (najčešće 1 ili 2) koji poput brojača 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 sledeće grananje.

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

55. Računari sa kombinovanom arhitekturom ?

Koriste prednosti CISC i RISC, nivo arhitekture-CISC, a interno organizovani kao RISC računari.

Pojavile se nove tehnike koje omogućavaju izbegavanja čekanja na izvršenje grananja:

1) **Uslovno izvršenje (predication)** – prilikom grananja, procesor izvrši obje grane, instrukcije se povezuju sa predikatom čija vrednost se čuva u specijalnom registru. Vrednost predikata je suprotna za ove grane. Tek kada uslov grananja postane poznat i predikat dobije svoju vrednost bira se odgovarajući rezultat.

2) **Sumnjivo-spekulativno učitavanje podataka (speculative loading)** unapred učitava operand iz memorije iako nije sigurno da će on biti potreban.

3) **Nagadjanje podataka (data speculation)**-pokusavaju da predvide vrednosti potrebnih podataka i nastave računanje koristeći ih.

56. Slabo i čvrsto povezani multiprocesorski sistemi ? - strana 144 slika 5.30

Ako tehnološka ili finansijska sredstva ne dozvoljavaju razvoj onda se koriste dva ili više procesora.

Multiprocesorski sistemi (MPS) podrazumevaju umnozavanje i međusobno povezivanje raspoloživih računarskih komponenti, procesora, memorije.

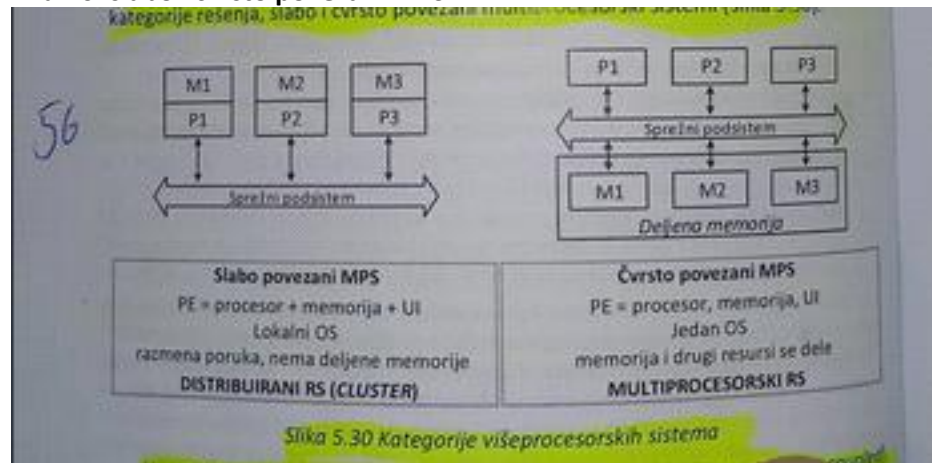
Dva osnovna cilja za uvođenje MPS: poboljšanje sistemskih performansi (ubrzavanje rada na osnovu podele posla na dijelove koji se paralelno izvršavaju), podizanje pouzdanosti i raspoloživosti (svaka od funkcionalno identičnih računarskih stanica je u stanju da obavi posao i da bude uvedena u operativni rad kao vodeća, dok su ostale rezervne).

Rekonfigurabilnost – postupak prilagođavanja računarskog sistema događajima, internim ili u okruženju.

Dva toka informacija: tok instrukcija, tok podataka.

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

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



Slabo povezani - cvorovi unutar njega medjusobnu 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 racunarska konfiguracija ovog tipa se naziva klasterom. **Klaster** je grupa racunara spregnutih lokalnom mrežom koja se spolja vidi kao jedan racunar.

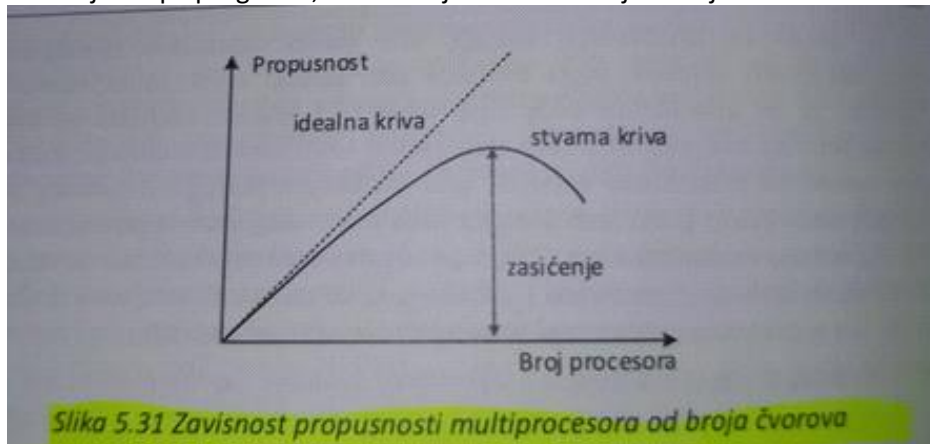
Cvrsto povezani - svaki procesor ima pristup cjelokupnom memorijskom prostoru pa se nazivaju **multiprocesori sa dijeljenom memorijom**. Elementi iako mogu imati svoju lokalnu memoriju, medjusobno dijele raspolozive memorijske module pod kontrolom jednog operativnog sistema. Sprega je na nivou magistrale.

Multiprocesorski sistemi moraju imati sledece karakteristike:

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

57. Propusna moć ? - strana 146 slika 5.31

Propusna moc multiprocesorskog sistema se definise kao recipročna vrednost vremena potrebnog za izvršenje skupa programa, ili kao broj rezultata dobijenih u jedinici vremena.



Kao što je na dijagramu prikazano u početku se povećanjem broja procesora linearno povećava i propusnost ali dalje povećanje dovodi do usporenja i odstupanja od krive.

Pouzdanost MPS se definise kao vjerovatnoća da će se traženi program izvršiti u predviđenom vremenskom intervalu.

Postoje dvije metode rješavanja problema otkaza nekog od modula (većina multiprocesorskih sistema može da nastavi rad u slučaju otkaza procesora ili nekog drugog modula).

Metode: hladne i vruće rezerve:

Kod metode hladne rezerve, modul u otkazu se zamjenjuje sa postojećim, dotle neangazovanim modulom. Moguće je ponavljanje dijela obrade u kom se javila greška.

Metoda vruće rezerve podrazumijeva da se dio ili kompletna obrada istovremeno izvršavaju na dva ili više cvorova unutar klastera, što je toku obrade moguće zamijeniti rezultatom sa ispravnog cvora.

Raspoloživost se definise kao vrijeme u kojem MPS raspoloživ korisniku, odnosno spreman za upotrebu.

Promena konfiguracije je neophodan zahtjev da bi se MPS mogao koristiti za rješavanje različitih problema i na taj način prilagoditi novim zahtjevima programa za rješavanje tih problema.

Povećana ekonomičnost se može postići na više načina jer se dijeljenjem memorijskih i U/I modula povećava njihov stepen iskoriscenja.

58. Simetrični i asimetrični multiprocesorski sistemi ?

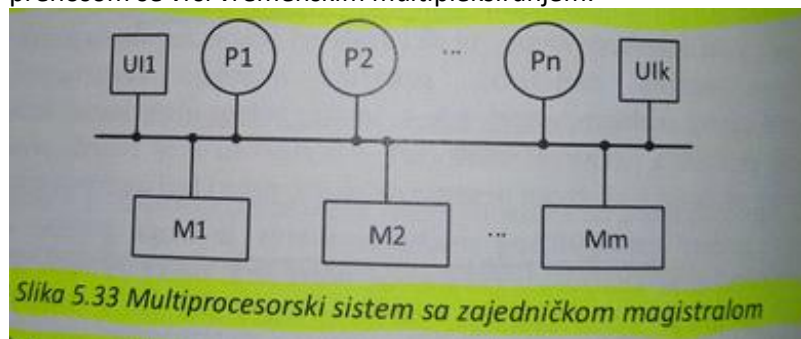
Simetrični multiprocesori - simetrija se mjeri uglavnom na hardverskom nivou, ali se i svaki od procesora tretira jednako od strane OS. Ona dominira kod **multicore** procesora koji su dominantni i na PC i mobilnim telefonima.

Asimetrični multiprocesor - Uloga graficke 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.

59. Multiprocesorski sistemi sa zajedničkom magistralom ? - strana 148 slika 5.34

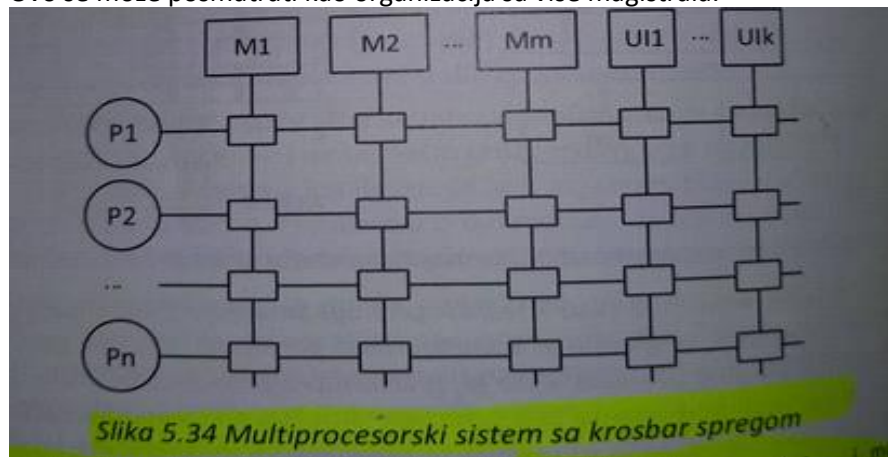
Povezivanje preko zajednicke magistrale je prvo i najjednostavnije resenje za organizaciju MPS. Osim sprege preko magistrale ne postoji direktna fizicka veza izmedju funkcionalnih jedinica a upravljanje prenosom se vrši vremenskim multipleksiranjem.



Arhitekturu sa zajednickom magistralom karakterise fleksibilnost u dodavanju ili premestanju modula, kao i niska cijena realizacije. Ipak resenje ima znacajnih nedostataka sa kasnjenjem i ukupnom efikasnoscu.

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 prekidački element koji kontrolise prenos između njih.

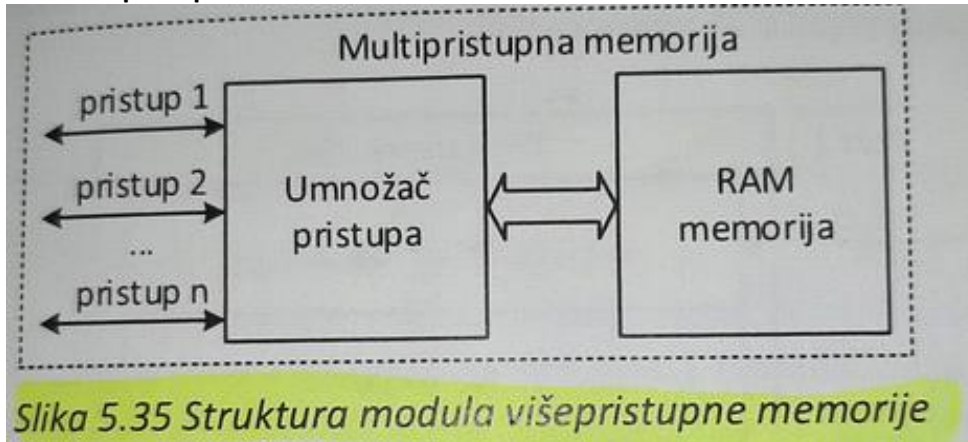
Cvorni prekidač može biti u stanju:

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

Prednost krosbar rjesenja je u simultanom prenosu izmedju razlicitih funkcionalnih modula, a osnovna mana kompleksnost fizickog rjesenja.

61. Multiprocesorski sistemi sa višepristupnom memorijom ? - strana 149 slika 5.35

Veze sa vise simultanih prenosivih puteva moze se ostvariti i pomocu vise pristupnih memorija. Moduli imaju vise od jednog pristupnog prolaza (port-a) zahvaljujuci dodatku posebne logike koja se zove **umnozac pristupa**.



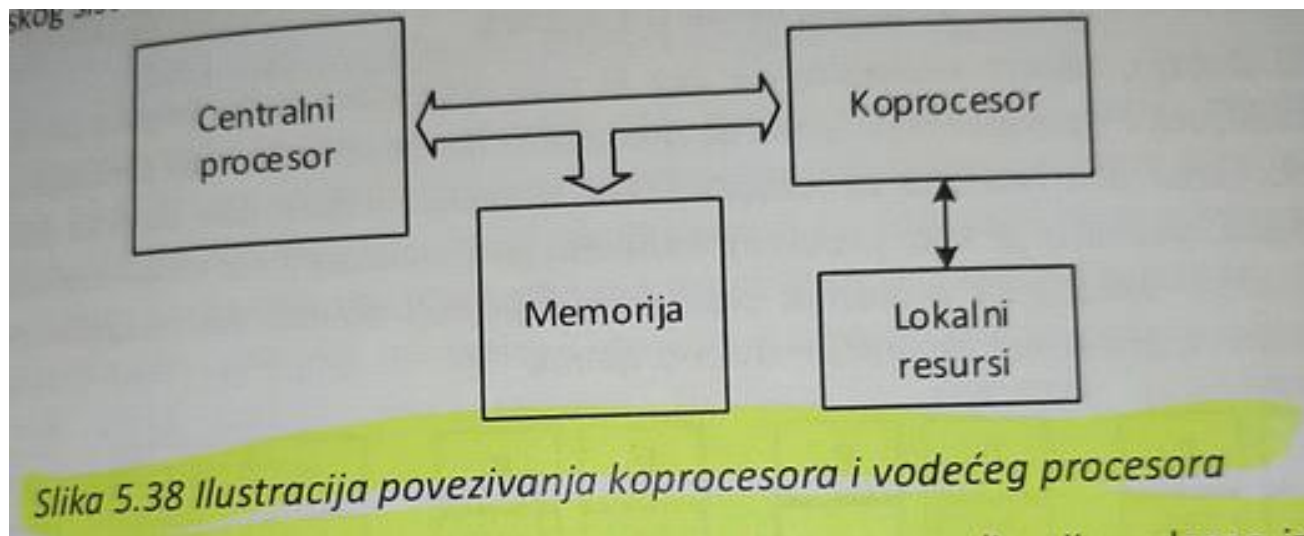
Slika 5.35 Struktura modula višepristupne memorije

Umnosac pristupa sadrzi kontrolna kola radi resavanja konflikta u uslovima kad vise procesora ili U/I jedinica trazi pristup tokom istog memorijskog ciklusa. Redosled prolaza (portova) oznacava i prioritet po kome se rjesavaju konflikti. Ovakva topologija je jeftinija od krosbar sprege, jer je manje tački gdje se rjesava konflikt. Za razliku od krosbara, kompleksnost rjesenja se sada seli na stranu memorije, ali je sustina manje vise ista.

Dvopristupne memorije ovakvog tipa (dual port memory, **DPM**) su vrlo popularne kao sredstvo za sprezanje dva procesora preko brzog kanala za prenos podataka. I u grafickim karticama se koristi verzija DRAM memorije sa dvostrukim pristupom tzv. Video RAM (VRAM) koji suprotnim stranama obezbjeđuje istovremen upis i citanje. Dvopristupna memorija ima ulogu postanskog sanducica, u koji se ostavljaju ili uzimaju poruke.

62. Koprocesori ? - strana 151 slika 5.38

Jedan od najprostijih varijanti povezivanja dva procesora se odnosi na koriscenje specijalizovanog procesora namijenjenog da rastereti osnovni procesor odredjenih specijalnih obrada. On se naziva koprocesor. Koprocesor je u cjelini pod kontrolom vodeceg procesora sa kojim je spojen i sa kojim deli magistralu radi koriscenja zajednickih resursa.



Ovakav način rada posebno je pogodan za realizaciju U/I, matematičkih, grafičkih procesora i slično. Koncept koprocesora je jedna od osnova savremenog računarstva.

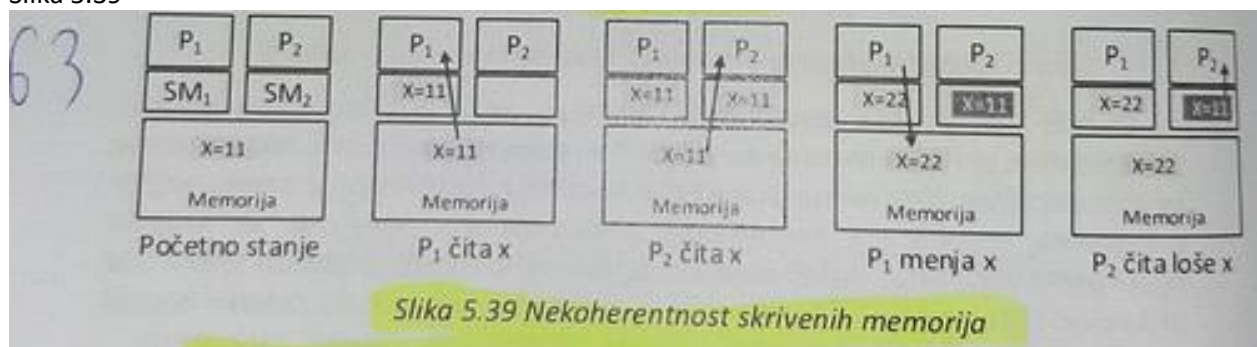
Dva osnovna rezima rada:

- poluautonoman (kada većinu svojih aktivnosti izvršava koriscenjem resursa glavnog procesora. vodeći/prateći) 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 obično koriste dva nivoa skrivene memorije gdje svako jezgro ima svoju privatnu cache memoriju prvog nivoa (L1 cache). U slučaju da dva ili više procesora istovremeno pristupaju promenljivoj u operativnoj memoriji može doći do neusklađenosti sadržaja u skrivenim memorijama dva različita procesora - **koherentnost**.

Slika 5.39



Ukoliko dva jezgra pristupe istoj promenljivoj X, u obje privatne skrivene memorije biće pohranjena njena početna vrijednost. Kada u nastavku obrade prvi procesor izmeni vrednost promenljive X, nova vrijednost biće sacuvana u njegovoj skrivenoj, ali i u zajedničkoj operativnoj memoriji. Ipak, o toj promjeni drugi procesor ništa ne zna, i zato zadržava staru vrijednost promenljive X. U slučaju ponovnog pristupa, on će regularno učitati ovu vrijednost, smatrajući je pravom. Ovo se mora spriječiti, a za to postoje dva rješenja:

- Protokol invalidacije** - izaziva slanje posebne komanda ka svim ostalim jezgrima da je promenljiva na adresi X promenjena i da su sve kopije nevazeće.

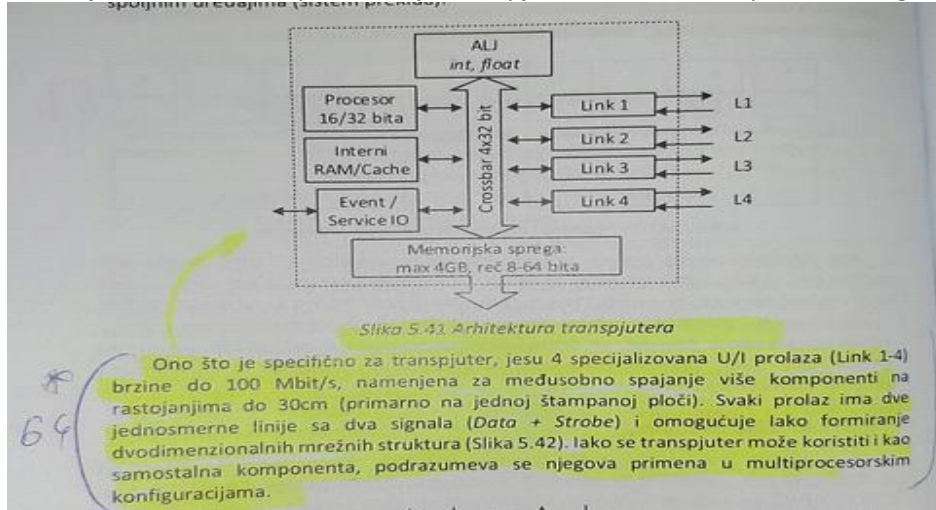
-Metoda azuriranja kopije - pored adrese salje i novu vrijednost pa sve skrivene memorije cuvaju tu vrijednost.

Prvo rjesenje je prisutnije zbog manjeg opterecenja.

64. Transpjuteri ? - strana 154 slika 5.41

Transpjuteri su razvijeni kao specijalizovane komponente posebno prilagodjene primjeni u multiprocesorskim sistemima. Struktura transpjutera je optimizovana sa ciljem postizanja visokog stepena konkurencije procesa i maksimalne funkcionalnosti.

U sustini transpjuter je mikrokontroler specijalizovane strukture, opremljen spreznim elementima namenjenim sto laksem povezivanju transpjutera unutar multiprocesorskog sistema.



Ono sto je specifcno za transpjuter jesu 4 specijalizovana U/I prolaza (link 1-4) brzine do 100 Mbit/s, namjenjena za medjusobno spajanje vise komponenti na rastojanjima do 30cm (primarno na jednoj štampanoj ploči). Svaki prolaz ima dvije jednosmerne linije, sa dva signala (Data + Strobe). Komunikacija izmedju transpjutera koristi internu memoriju sto je mnogo brze nego prenos preko spoljne memorije.

Prihvaceni su ali brzo izgubili trku sa novim i jeftinim protocnim procesorima RISC arhitekture.

65. Procesori sa više jezgara i grafički procesori ?

Dva problema za dalje poboljšanje performansi: **potrošnja energija, dužina internih linija**.

Povećanje takta podiže potrošnju, procesor se zagrijeva.

Drugi razlog koji ograničava takt je kašnjenje električnih 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 krajnih 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 ublaži razvijena je **Hyper-Threading tehnika (HT)** koja dodatnim hardverom pomaže smjenu kontekst na nivou samog procesora (jezgra)

GRAFICKI PROCESORI: (možda slika 5.47).

1999. pojavila se prva grafička kartica sa programskom kontrolom i sa moćnim skupom procesora posebno prilagođenih specifičnostima video obrada. Njihovim radom upravlja aplikacija koja se razvija u C/C++, puni u lokalnoj memoriji i odatle izvršava. Ta aplikacija ne mora biti grafička. Po prvi put grafičke kartice nadmašile mogućnosti centralnog procesora. Ovakve kartice se nazivaju **grafickim procesorima**. GPU procesori brojem izvršenih operacija nadmašuju CPU savremenike i do 20 puta.

Prva pogodnost je sama priroda grafičkih operacija.

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

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

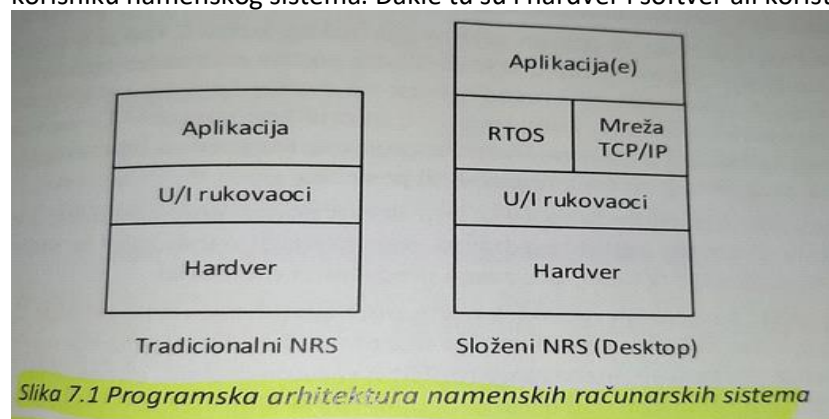
Dvije činjenice koje uprošćavaju realizaciju GPU procesora. **Grafičke aplikacije nemaju suviše grananja.**

Sve niti u izvršenju su nezavisne. Ako se pojavi rizik samo se data nit zaustavi a angažuju 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 računarstva koja se bavi njima su trenutno u fokusu savremenih računarskih tehnologija. Termin NRS označava računar ugrađen unutar drugog električnog ili mehaničkog uređaja. Na tom računaru se izvršava program koji u realno vremenu kontrolise rad cijelog uređaja, iako je nevidljiv korisniku namenskog sistema. Dakle tu su i hardver i softver ali koriste se drugačije od desktop računara.



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 određuju se tako da odgovaraju minimumu koji garantuje ispunjenje vremenskih uslova vezanih za rad u realnom vremenu.

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

- Kvalitet softvera također određuje uspjehnost nekog proizvoda.

Očekuje se poznavanje C jezika.

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

67. Koji problemi se javljaju kod namenskih računara ?

(67/68 između)(7.1 RAZVOJNO OKRUŽENJE NAMENSKIH SISTEMA)

Kako pristupiti računarskom sistemu koji nema tastaturu i monitor, a vrlo često ni LED diodu ili neki tester.

Ovde se primenjuje **cross-platform** metodologija razvoja, koja se tako zove jer se na jednom računaru (**izvorni, host**) priprema program koji će se izvršavati na drugom računaru (**određisni, target**). Razvojni računar je danas po pravilu standardna PC radna stanica, koja se po svemu razlikuje od određenog računara. Na njoj se izvršavaju kompajler, linker, debugger i drugi razvojni alati, znači na platformi različitoj od određisne. **Slika 7.2.**

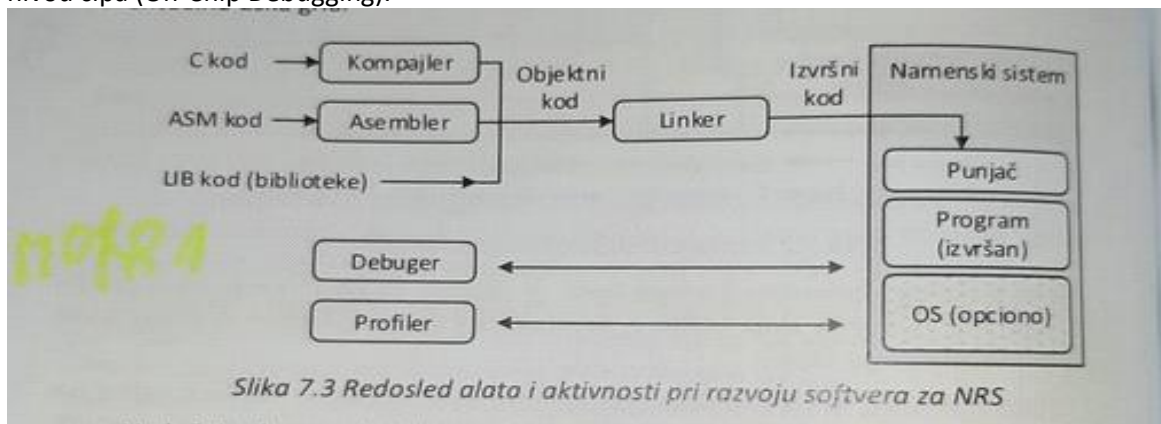
68. Koja tri načina postoje da prebacimo izvršni kod ?

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

Upisom izvršnog koda u programski EPROM, koji se potom fizički instalira na stampanu ploču određenog sistema, insertovanjem u postolje. Testiranje programa se radi korišćenjem 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 način rada.

Serijskim prenosom, preko UART, USB ili LAN komunikacione linije. Ovakav način rada je moguć samo ako na određenom sistemu već postoji komponenta za punjenje programa (punjač, loader) koja umije da prihvati novi program, upiše 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 način rada podrazumjeva upotrebu komponente koja se naziva programer/debuger čijim posredstvom se ostvaruje fizička veza sa hardverskim komponentama određenog sistema, što uključuje procesor, memoriju, i U/I uređaje. Ovakav način rada zahtjeva JTAG ili BDM port na određenom sistemu, odnosno hardversku podršku za debugovanje na nivou čipa (On-Chip Debugging).

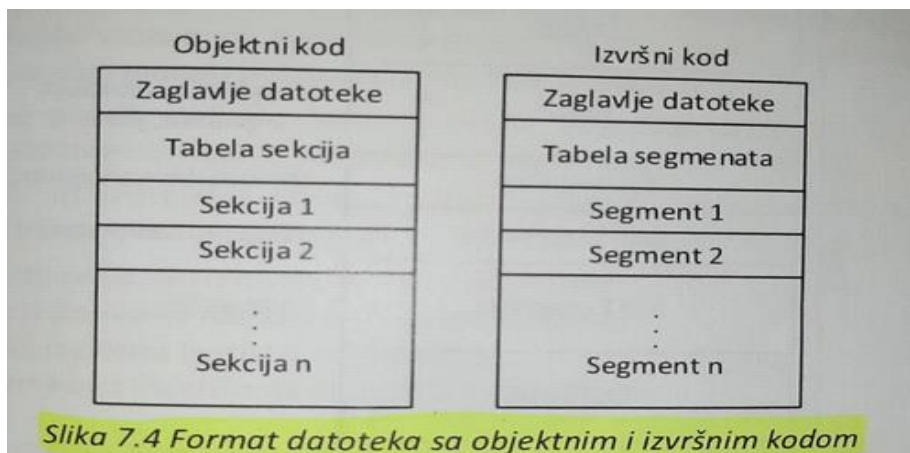


Slika 7.3 Redosled alata i aktivnosti pri razvoju softvera za NRS

69. Format binarnog koda ? - strana 211 slika 7.4

Datoteke sa objektnim i izvršnim kodom sadrže slične informacije:

- binarne instrukcije i podatke zavisne od određisne platforme,
- tabele simbola i relokacije,
- debug informacije koje koristi debugger,
- zaglavlje datoteke sa opštim informacijama poput velicine fajla binarnog koda i podataka.



Program je organizovan kao skup sekcija, grupa, sadržaja jednog tipa, koji se definišu sistemski ili po zahtjevu korisnika. Kao posebne sekcije organizovane su instrukcije, programske promenljive (podaci), tabele simbola i relokacije, debug informacije itd.

Svaka sekcija pored tipa, ima informacije o adresi punjenja i adresi izvršenja.

Programer može definisati korisničke sekcije po želji ali tri su podrazumjevana: `.text` `.data` `.bss`

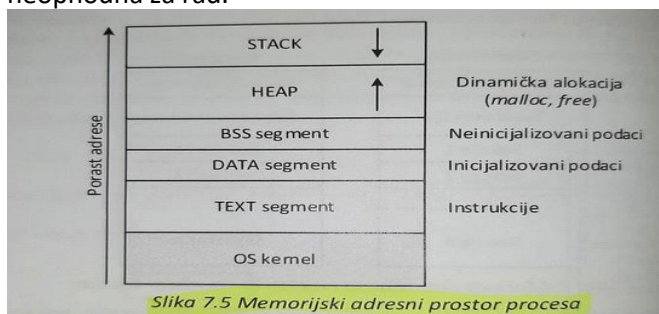
`.text` sadrži instrukcije, odnosno binarni kod koji razvija kompajler.

`.data` sadrži inicijalizovane globalne promenljive za koje pre izvršenja treba alocirati memoriju.

`.bss` (Block Started by Symbol) čuva neinicijalizovane podatke odnosno promenljive koje pre pokretanja ne moraju biti eksplicitno inicijalizovane. (u praksi obično neinicijalizovani se postavljaju na 0).

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

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



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

Heap je zona memorije sa dinamičkom alokacijom. Funkcije `malloc` i `free`.

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

71. Binarna kompatibilnost ?

U praksi vrlo je važna mogućnost da se neka aplikacija (program) razvijen na jednom sistemu bez ponovnog kompajliranja primeni na nekom drugom računaru sa istim procesorom i OS. **Binarna kompatibilnost** podrazumeva postovanje niz konvencija neophodnih da bi neki program mogao biti učitao od strane OS i izvršen na konkretnoj platformi.

Skup ovih konvencija – ABI, obuhvata pravila vezana za OS i samu fizicku platformu. Počev 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.

72. Inicijalizacija i pokretanje odredišnog sistema ?

Podizanje sistema (boot-up procedura) obuhvata tri ključne faze:

1.**Inicijalizacija hardvera u okviru reset procedure**, podrazumijeva postavljanje ispravnog rezima rada procesora i memorija (uključujući MMU ako ga ima) i U/I uređaje.

2.**Inicijalizacija softvera ima dvije grane**. Ukoliko namjenski racunar (ili desktop) koristi RTOS, neophodno je njegovo učitavanje, inicijalizacija i pokretanje. U suprotnom neophodno je punjenje programa na odredisnom sistemu.

3.**Pokretanje same aplikacije**, pod kontrolom OS ili punjaca. U oba slucaja to zahtjeva alokaciju prostora u RAM memoriji, i formiranje izvrsne slike programa.

73. Šta je punjač (loader) ?

Punjac je uslužni program koji je po pravilu smesten u ROM/Flash memoriji, kako bi bio raspoloživ i po isključenju napajanja. Poziva ga reset procedura ako je transfer programa neophodan. Punjenje se kontrolise sa: LoadDone (promenljiva u flash memoriji), postavkom spoljnog 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 izvršenja aplikacije:

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

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

75. Alati za testiranje i otklanjanje grešaka ?

NR nemaju monitor i tastaturu, te se njihovo testiranje mora izvesti na host strani, u okviru razvojnog okruženja. Dva rjesenja: **programsko i hardversko**. Prvo - isključivo specijalizovane programske komponente. Drugo - poseban hardver, dodatno ugradjen u integrisana kola.

PROGRAMSKO TESTIRANJE: MONITOR I DEBUG AGENT:

Razvojnim alatima koji se izvrsavaju na host strani neophodna je saradnja dodatnog sistemskog softvera koji se izvrsava na odredisnoj strani: monitor i debug-agent.

Monitor je prvo proširenje klasicnog punjaca koji je obogacen ugradnjom systemske podrške za inicijalizaciju namjenskog sistema i kasnije pracenje u toku rada. Monitor ima specijalizovanu korisnicku

spregu za razliku od punjaca, realizovanu u formi serijskog emulatora koja prihvata i izvršava komande host alata. Monitor omogućuje:

- citanje i pisanje memorijskih lokacija i registara odredisnog sistema,
- postavljanje i brisanje tacaka prekida,
- koracno izvršenje instrukcija,
- reset sistema i sl.

Debug-agent je u sustini isti kao monitor ali omogućuje debug funkcionalnost na nivou izvornog koda i njegove vizuelne konzole. Veci dio se obavlja na host strani gdje se posredstvom korisnicke sprege pokrecu testne (debug) akcije i prikazuju rezultati. Ključna komponenta debuggera je njegov **automat**. U odnosu na aplikaciju, debug-agent se izvršava kao nezavisan program ali mora imati pristup njenom adresnom prostoru i u toku izvršavanja.

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 uređaja. Spoljni i nezavisni pristup internim elementima integrisanih kola je ključna funkcionalnost hardverske sprege koja podržava rad savremenih razvojnih okruženja. Ovakav način rada se naziva testiranjem na nivou cipa (OCD), i označava hardversku podršku procesu kontrole programa i otklanjanju gresaka (debugging) ugrađenu u same procesore i njihove periferije.

Dva tipa **OCD**: standardizovan JTAG sprega i mnoga druga resenja specifična 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 ugrađuje u interesano kolo koja omogućuje takav pristup.

Zapravo, JTAG je Boundary-Scan proširen OCD podrškom.

77. BOUNDARY-SCAN arhitektura ? (ne znam da li trebaju slike 7.14 i 7.15, ko ce ove jade uciti)

Nova tehnika koja u osnovi ima jednostavnu ideju - na granici integrisanog kola na njegovim spoljnim priključcima treba ugraditi malu dodatnu logiku koja ce omogućiti učitavanje i postavljanje trenutne vrednosti na pinu. Logika koja se dodaje ispred svakog pina naziva se **graničnom ćelijom (Boundary cell)** pri čemu susjedne ćelije serijski povezuju u lanac.

Svaka ćelija ima dva para ulaza i izlaza:

- Signali PI i PO (Data In, Data Out)- posreduju u prenosu podataka između pina i interne logike ali i testnog okruženja.
- Signali SI i SO (Scan In, Scan Out) - se angazuju za serijski prenos podataka između neke od ćelija i spoljnih priključaka test sistema(TDI i TDO).

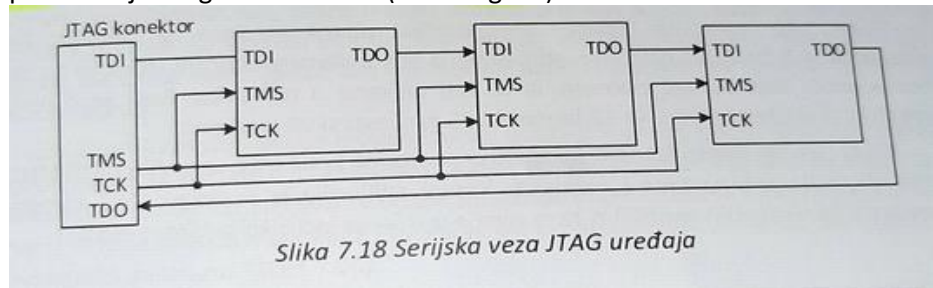
Svaka ćelija podržava četiri elementarne operacije:

- Transparent, radni režim kada BC ćelija signal sa PI prenosi na PO,
- Capture, učitavanje i pamćenje vrijednosti sa PI ulaza,
- Update, postavljanje vrijednosti na Po, samim tim na izlaz integrisanog kola,
- Shift. Serijski prenos preko SI-SO.

JTAG ARHITEKTURA (NE TREBA)

78. Veza JTAG uređaja na štampanoj ploči ?

Na štampanoj ploči nekog računara 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.

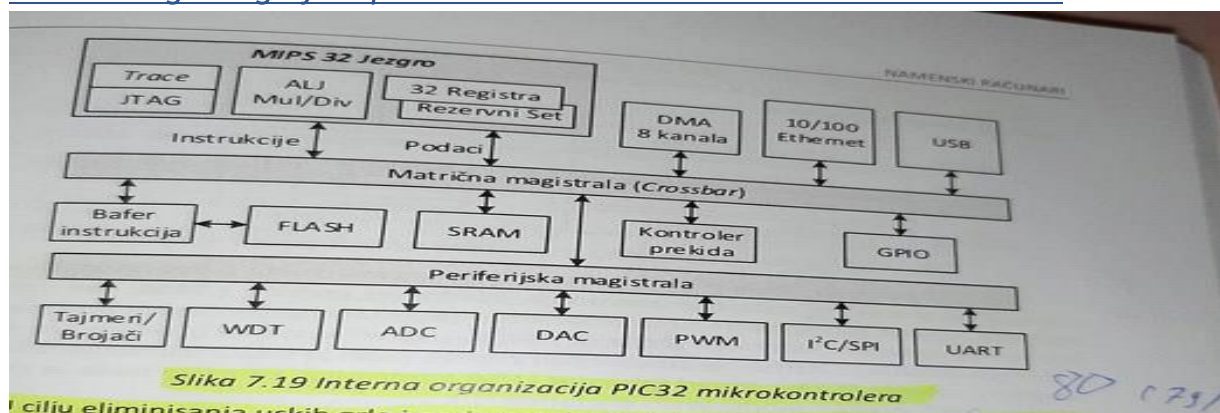
Dovodjenjem prave kontrole na JTAG konektor može se:

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

79. Namenski mikrokontroleri ?

Mikrokontroler je integrisano kolo koje objedinjuje procesor, memorije i periferije, odnosno sve komponente koje su potrebne za realizaciju i rad nekog računara relativno skromnije strukture. U slučaju namenskog računara veza sa uređajem u koji je ugrađen primarno koristi U/I signale različitog tipa. Neki signali: kontinualni (analogni), diskretni (digitalni), impulsni. Rukovanje njima zahtjeva specijalizovane U/I kontrolere tzv. programibilne periferije pomoću kojih se ostvaruje programska interakcija procesora sa okolinom, uređajem u koji je ugrađen. Komunikacione opcije : UART - asinhroni prenos podataka, Ethernet-komunikacija preko lokalne mreže.

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



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

-Prva je obezbedjenje rezervnog seta registara opste 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 i I²C).

81. GPIO - ulazni/izlazni opšte namene?

GPIO – General Purpose Input/Output

Osnovni nacin povezivanja racunara sa okruzenjem podrazumeva koriscenje skupa ulaznih i izlaznih linija kojima se očitava stanje i prenose komande. Ovaj nacin rada i cjeli podsistem naziva se paralelnim U/I. Umjesto tradicionalnih PIO (parallel IO) kola gdje se podesavanje rezima ulaz/izlaz radilo na nivou 8 bita, savremeni mikrokontroleri koriste GPIO (General Purpose IO) organizaciju gdje se svaka linija podesava individualno. Svaka GPIO ćelija sadrzi tri registra tacnije po jedan bit iz tri registra pomocu kojih se ostvaruje programski pristup U/I pinu na obodu integrisanog kola. Savremeni mikrokontroleri mogu imati i vise desetina GPIO linija sto zavisi od namene i vrste kucista integrisanog kola.

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 predvidjenom 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.

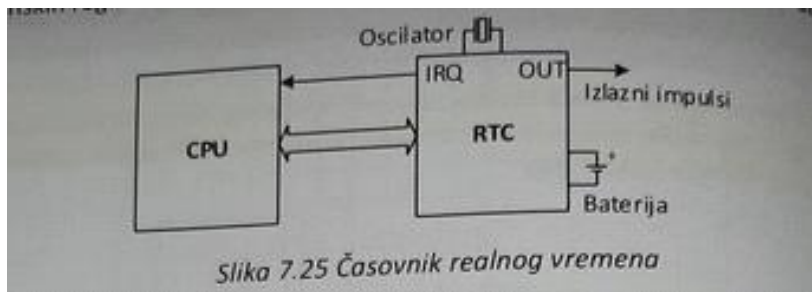
83. RTC - Časovnik realnog vremena ?

Časovnik realnog vremena (RTC-real time clock) je komponenta namenjena za pracenje trenutnog vremena i datuma. Osnovna funkcionalnost je ponovo brojanje impulsa cija ucestanost ovaj put mora biti tako izabrana da omoguci precizno mjerenje sekundi (bez gubitaka). Sekundarna ali uvijek prisutna funkcija RTC je generisanje periodicnih prekida, vremenskih prekida.

Dva parametra svakog RTC:

- Mikrotik,
- Tik

Kontrola RTC podrazumeva pristup internim vremenskim registrima koji cuvaju tekuće vreme(hh:mm:ss) i datum (dd:mm:yy). Da bi radio kako treba RTC mora imati permanentno napajanje iz baterije ili superkondenzatora.

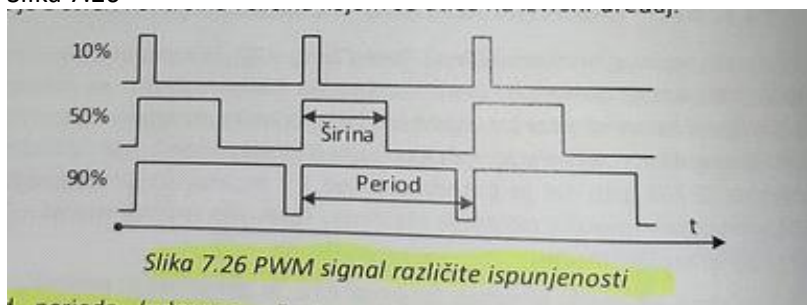


(NE TREBA OVO JA MSM)

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

U realnim aplikacijama kao izlazni upravljački signal, često se koristi **PWM** (širinski modulisani impulsi, Pulse Width Modulation). Radi se o povorci impulsa iste amplitude i učestanosti ali i promenljive širine odnosno promenljivog odnosa signal/pauza.

Slika 7.26



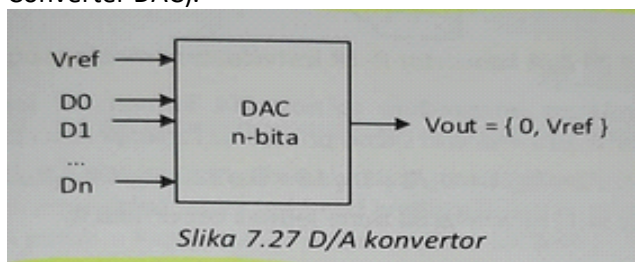
Širina **PWM** signala varira 0-100%.

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

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

85. DAC - Digitalno analogni konvertor ? - strana 231 slika 7.28 i strana 232 slika 7.29 + šeme koje je profesor crtao na predavanjima

U slučaju da je potreban pravi analogni signal mora se koristiti **D/A konvertor** (Digital to Analog Converter DAC).

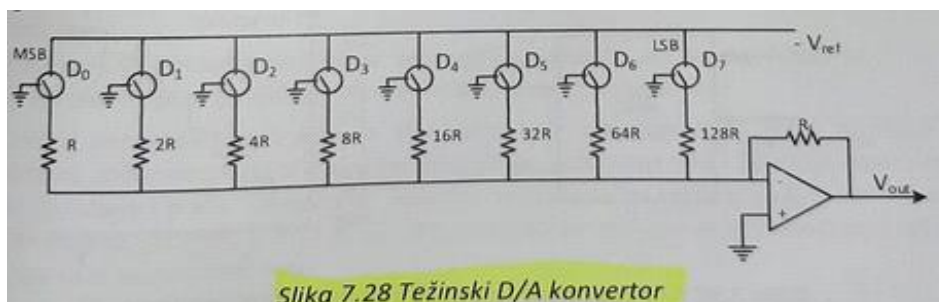


(NE TREBA)

Na osnovu ulazne n-bitne vrednosti **D/A konvertor** generise izlazni signal u opsegu 0-Vref.

Preciznost generisanog analognog signala je $V_{ref} / 2^n$. Uobicajen broj **D/A** bita u realnim aplikacijama kreće se u opsegu 8 do 12.

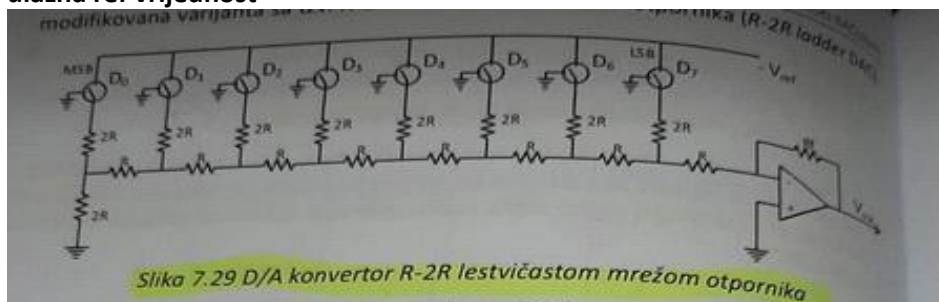
Osnovna realizacija **D/A konvertora** koristi otpornike čija je otpornost određena binarnom cifrom koja kontrolise protok struje kroz svaki od njih.



Slika 7.28 Težinski D/A konvertor

(NE TREBA)

$V_{out} = R_f/R * (D_0/1 + D_1/2 + D_2/4 + D_3/8 + D_4/16 + D_5/32 + D_6/64 + D_7/128) * V_{ref}$, (R_f , R – otpornost, V_{ref} – ulazna ref vrijednost)



Slika 7.29 D/A konvertor R-2R lestvičastom mrežom otpornika

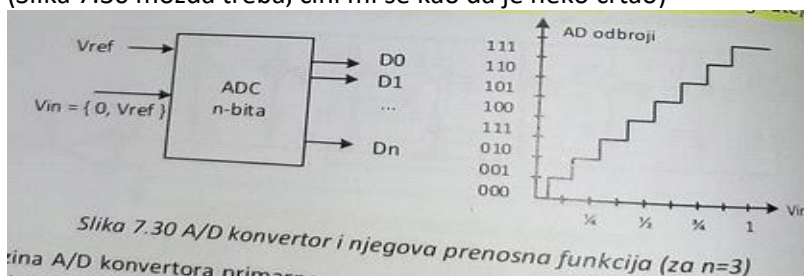
(NE TREBA)

Nedostatak prehodne konfiguracije, pogotovo u slučaju većeg broja bita, je precizna realizacija otpornika velikog opsega otpornosti. Zbog toga se najčešće koristi modifikovana varijanta sa tzv. R-2R lestvičastom mrežom otpornika (R-2R ladder DAC).

$V_{out} = R_f/R * (D_0/2 + D_1/4 + D_2/8 + D_3/16 + D_4/32 + D_5/64 + D_6/128 + D_7/256) * V_{ref}$

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 - V_{ref} i proizvodi izlaznu cjelobrojnu vrednost srazmjernu ulaznom naponu. Jedan od osnovnih parametara **A/D konvertora** je brzina konverzije koja ograničava učestalost pristupa **A/D kolu**. (Slika 7.30 možda treba, cini mi se kao da je neko crtao)

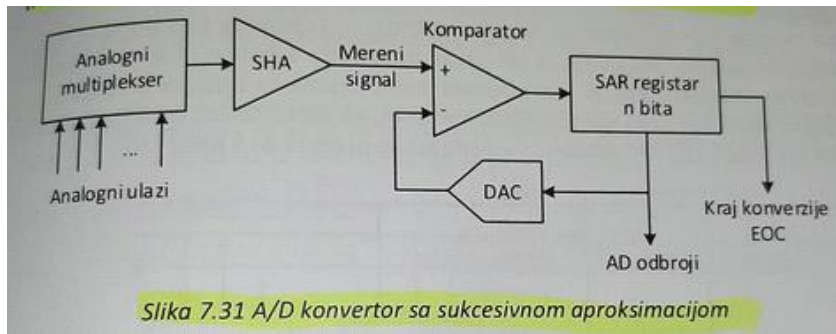


Slika 7.30 A/D konvertor i njegova prenosna funkcija (za n=3)

(NE TREBA)

Najbrži su tzv. **flash konverteri sa direktnom konverzijom** (istovremeno određuju vrijednost svake od binarnih cifara). U praksi su najčešći **A/D konvertori sa uzastopnom aproksimacijom** (kompromis cijene i performansi). Osnovna ideja **SA konvertora** je da se u petlji odredi svaka od cifara polazeći od bita najveće težine (MSB). Cijeli postupak se odvija u dvije faze: prihvati/memorisiranje 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).

87. SPI magistrala ? - strana 234 slika 7.32 ili 7.33

Ukoliko je mikrokontroler neophodno povezati sa spoljnim komponentom, nekom pomocnom memorijom, periferijom ili bilo kojim integrisanim kolom, klasican nacin podrazumijeva povezivanje preko paralelne spoljne magistrale - **SPI**. Razvijaju se serijske U/I magistrale.

SPI (Serial Peripheral Interface) je full-duplex magistrala sa sinhronom serijskom komunikacijom. Na **SPI** magistrali samo jedan uređaj moze imati status vodeceg (mastera) i to je onaj koji generise takt. **SPI** se nekad oznacava kao 4 - zicna (4-wire).

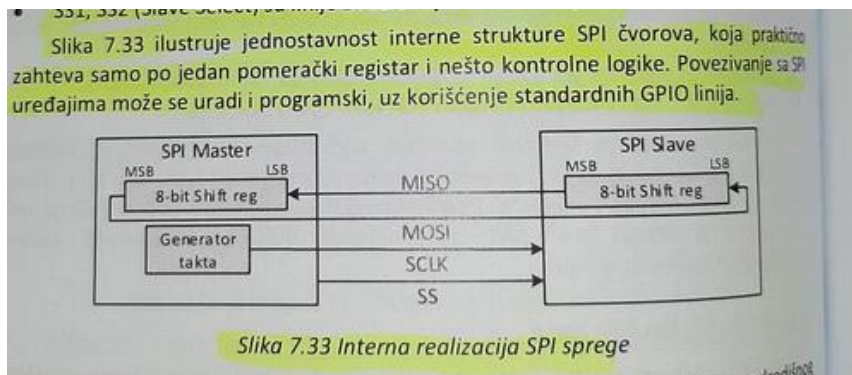
Slika 7.32



Slika 7.32 prikazuje SPI magistralu sa dve povezane periferije, a signali sa slike imaju sledeća značenja:

- MOSI (Master Out Slave In) je linija za predaju od mastera,
- MISO (Master In Slave Out) koristi slave za slanje ka masteru,
- SCK (Serial Clock) nosi zajednički komunikacioni takt,
- SS1, SS2 (Slave Select) su linije za selekciju slave uređaja.

Slika 7.33 ilustruje jednostavnost interne strukture SPI čvorova, koja praktično zahteva samo po jedan pomerački registar i nešto kontrolne logike. Povezivanje sa SPI uređajima može se uradi i programski, uz korišćenje standardnih GPIO linija.



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 pomoću njegove komunikacione adrese. Za realizaciju sprege su potrebna samo dva signala pa se ova magistrala često naziva i **dvožicom spregom** (2 wire interface). Postoji 1 master čvor iako više čvorova može da se takmiči za kontrolu nad magistralom.

Slika 7.36



(NE TREBA)

C 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 razmjenjuje tokom odvijanja transakcije.

Slika 7.37



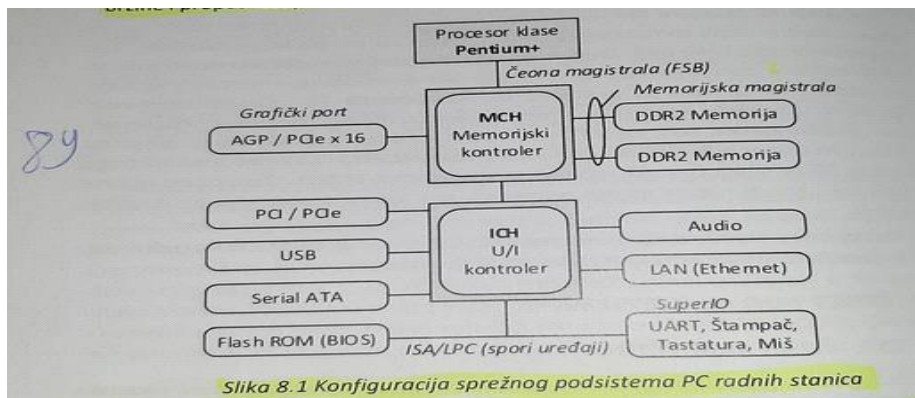
(NE TREBA)

Ova slika (Slika 7.37) prikazuje opšti tok C transakcije, koja zapocinje obaranjem SDA signala dok je tak još 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.



Slika 8.1 Konfiguracija sprežnog podsistema PC radnih stanica

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 racunara: 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..

90. Memorijska mapa ?

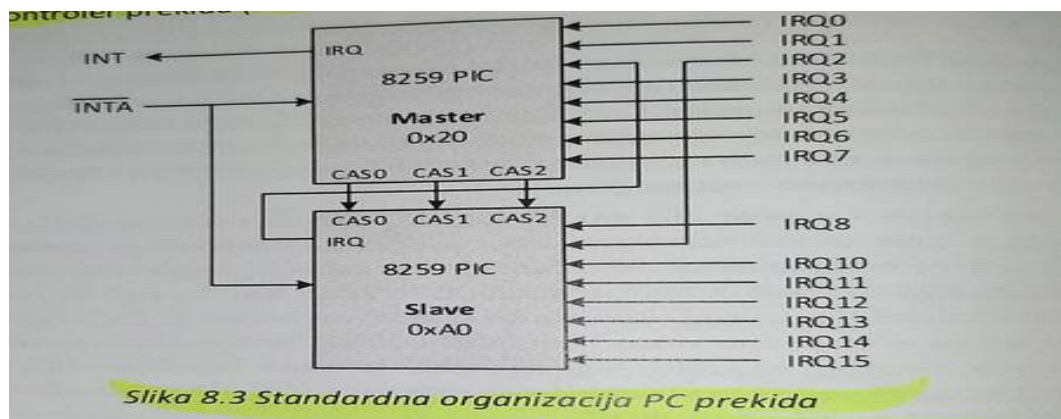
Slika 8.2 prikazuje kako mapa izgleda kod 32 – bitnih PC stanica, koje korisnicima rezervisu RAM zonu od približno 3GB. Ostatak adresnog prostora sadrži sistemske informacije, pre svega Flash Bios, PCI strukture podataka, APIC postavku, itd...

Memorijska mapa PC racunara uvijek zapocinje segmentom duzine 1MB, cija je organizacija ista kao i kod najstarijih DOS racunara. Na pocetku se nalazi 640KB korisnicke memorije, u kojoj se izvrsavao operativni sistem i aplikacije. Iznad je sistemska zona, u koju je mapirana video memorija graficke kartice, kao i sva memorija na U/I karticama. Konacno na kraju ovog segmenta se nalazi BIOS memorija, u kojoj se cuva boot procedura i podrška za inicijalizaciju radne stanice i učitavanje operativnog sistema.

91. Podsistem prekida ? - strana 240 slika 8.3

U kaskadnoj konfiguraciji glavni kontroler prosledjuje zahteve prateceg 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



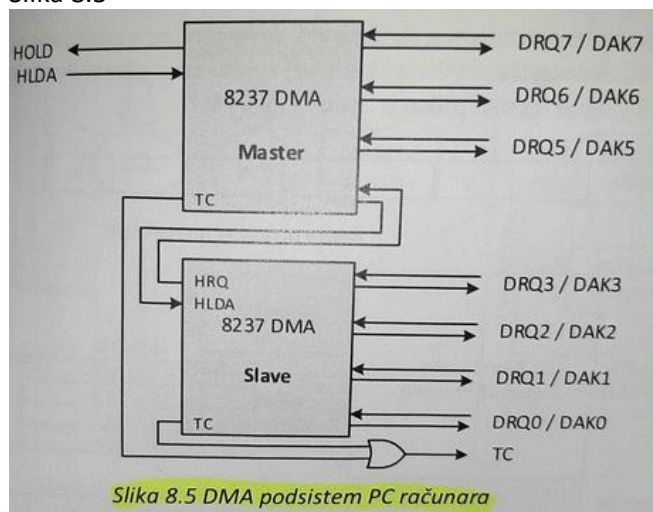
To znaci da je u dugom periodu PC racunar imao samo 15 U/I prekida pri cemu su mnogi od njih zauzeti standardnim periferijama(disk, tastatura, mis). U prvim implementacijama PCI magistrale uredjaji na njoj su mapirani na jedan od slobodnih prekida (IRQ10 najcesce).

Ovakva ogranicenja su neprihvatljiva sa pojavom multicore procesora i simetricnog multiprocesiranja te je kao zamjena **uvarena 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.

92. Podsystem DMA ? - strana 241 slika 8.5

Tradicionalni korisnici DMA prenosa su diskovi i druge masovne memorije, mrezne i vrlo brze komunikacione kartice, dakle uredaji koji rukuju velikim obimom podataka i ne trpe znacajna kasnjenja u obradi. DMA podsystem obuhvata samo 7 kanala koji su izlozeni samo ISA magistrali.

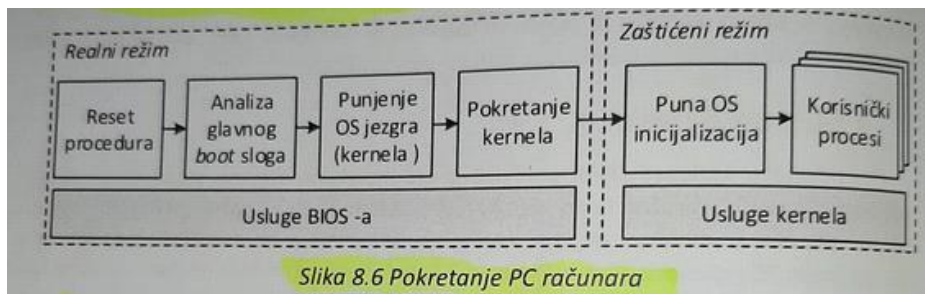
Slika 8.5



Slika prikazuje izvornu DMA strukturu koju cine dva kaskadno vezana DMA kontrolera 8237. HOLD i HLDA su signali za zahtjev i potvrdu dodele magistrale, pomocu kojih se sinhronizuje rad sa procesorom. Skup DMA kanala je znacajno skromniji u odnosu na prekid.

93. Inicijalizacija PC racunara 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 izvrsava reset proceduru i puni i pokrece jezgro OS (kernel).



Ključni 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).

94. Osobine i klasifikacija PC magistrala ? - strana 243 slika 8.8

Propusni opseg magistrale limitira maksimalnu brzinu komunikacije sa U/I jedinicima. Uredjaji koji se povezuju preko magistrale razlikuju se u pogledu kasnjenja i brzine prenosa. Uvode se specijalizovane magistrale.



(NE TREBA)

Postoje:

- procesorske,
- sistemske i
- UI magistrale.

Redosljedom, opada brzina prenosa i propusnost a raste znacaj, fleksibilnosti i prosirivosti magistrale.

Procesorka mora biti najbrza jer je posvecena komunikaciji procesora sa memorijom i grafickim video kontrolerom.

Sistemska povezuje procesor sa brzim periferijama, vitalnim za rad i perfomanske ukupnog racunarskog sistema, najcesce spoljinim memorijama i brzim mreznim kontrolerima.

Najsporija je **UI magistrala** namjenjena sprezanju sa sporijim perifernim jedinicama, raznovrsnim po tipu i funkciji.

Postoji veliki broj standarda za racunarske magistrale, ciji se izvori mogu svrstati u tri grupe:

- prva-spadaju tehnicka rjesenja odredjenog proizvodjaca,

-druga - tehnicka resenja pojedinih proizvođača koja su prilagođena i od strane nacionalnih i međunarodnih organizacija za standarde predložene kao zvaničan standard.

-Treci - kod nezavisnih proizvođača opreme, motivisane posebnim interesima i nezavisno od organizacija razvijaju i definišu standarde za svoje potrebe i stavljaju ih na raspolaganje širem krugu korisnika.

Osnovna obilježja magistrala:

-širina - broj linija preko kojih se informacije istovremeno prenose na magistrali,

-brzina - određena učestanosti radnog takta,

-propusnost - obim podataka koji se može prenijeti preko magistrale u nekom vremenu.

95. Standardne PC magistrale ?

Po tipu i namjeni PC magistrale možemo klasifikovati u četiri kategorije:

Procesorska magistrala je na hijerarhijski najvišem nivou sa namenom povezivanja procesora sa najbržim 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 mrežnih 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 unapređenja brzine i propusnosti PCI magistrala je prva u cjelini podržala „prikljuci i koristi“ (Plug & Play-PnP) koncept povezivanja osnovnog PC računara sa periferijskim 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 obezbjeđenje 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 izbjegavaju problemi uklapanja periferijskih uređaja u memorijskom i U/I adresnom prostoru računara.

96. Automatska konfiguracija PC periferija ?

Koncept automatske konfiguracije PC periferija poznat je pod nazivom „prikljuci i koristi“ (Plug & Play-PnP). Osnovni cilj PnP specifikacije je razvoj računara čija fizička i programska podrška zajednički vrše dodelu resursa i konfiguraciju periferijskih uređaja.

PnP podrazumeva automatsku detekciju i konfiguraciju hardvera i softvera što generalno nije jednostavan zadatak.

Njegovo izvršenje zahteva međusobnu uskladenost i kooperaciju svih ključnih učesnika:

-Sistemski hardver (mora biti sposoban da rukuje sa PnP),

-Periferijski hardver (svaka U/I kartica mora biti sposobna da na zahtev izvrši svoju identifikaciju),

-Sistemski BIOS (igra ključnu ulogu u izvršenju PnP funkcija),

-OS (mora biti projektovan za rad sa BIOS-om a preko njega i sa krajnjim PnP periferijama).

Većina PnP funkcija izvršava se u fazi inicijalizacije sistema po uključanju, pod kontrolom BIOS-a.

Procedura detekcije i konfiguracije PnP uređaja odvija se u više koraka:

-Formira se tabela raspoloživih resursa tj. prekida, DMA kanala i U/I adresa koje nisu rezervisane za sistemske uređaje.

-Pretragom se identifikuju svi uređaji na PCI i ISA magistrali uključujući i one koji nisu kompatibilni sa PnP standardom.

-Iz CMOS memorije učitava se ESCD tabela koja sadrži zadnju poznatu konfiguraciju.

- Upoređuje tekuće i učitane konfiguracije, detektuje se eventualno prisustvo novih fizickih komponenti.
- Ukoliko je konfiguracija sistema izmenjena neophodna je njegova rekonfiguracija.
- Iz skupa preostalih resursa dodeljuju se zahtevani resursi, o čemu se izvestava i periferni uređaj.
- Preostaje postupak azuzriranja ESCD tabele zapisom nove konfiguracije.

97. ISA magistrala ?

ISA(Industry Standard Architecture) je najstarija PC magistrala, pristuna od pojave prvog racunara ovog tipa. Raspoloziva 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 sto nije potrebno vise od njenih performansi.

Svojom izvedbom **ISA magistrala** obezbedjuje pogodan nacin povezivanja sa sporim perifernim uređajima, i pri tome garantuje kompatibilnost sa velikim br. racunara i perifernih komponenti.

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

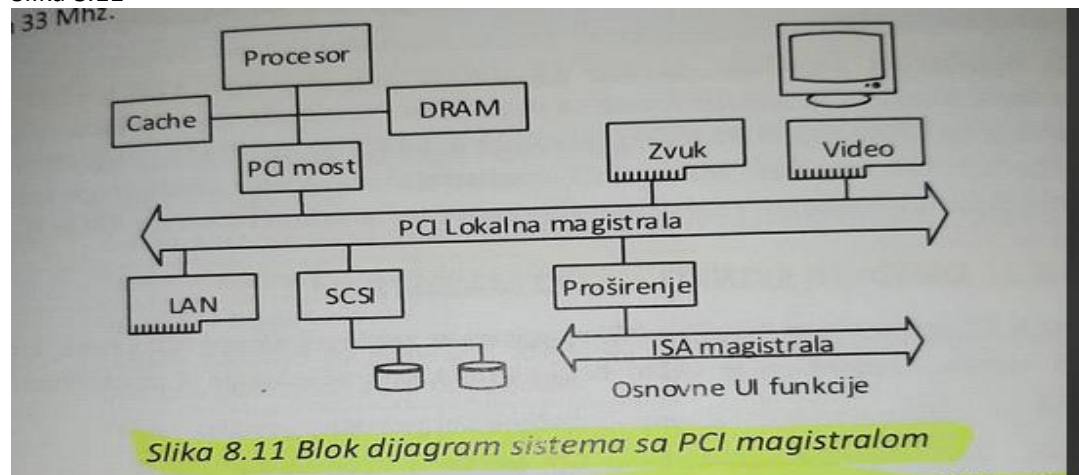
Pored procesora tu su signali koji omogucuju:

- kontrolu citanja i upisa u memorijski ili U/I prostor,
- generisanje stanja cekanja,
- koriscenje sistema prekida i direktnog pristupa memoriji,
- osvezavanje memorije i detekciju greske.

98. PCI magistrala ? - strana 253 slika 8.11

Peripheral Component Interconnect (PCI) specifikacija definise lokalnu magistralu visokih performansi koja podrzava 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



Slika 8.11 Blok dijagram sistema sa PCI magistralom

PCI magistrala je primenjena na razlicitim hardverskim platformama, cime je obezbijedjena kompatibilnost i laka prenosivost perifernih kartica. Takodje koncept „Plug & Play“ je osnova prednost.

Slika 8.11 pokazuje prvu izvornu arhitekturu racunarskog sistema sa **PCI magistralom**. Kljucna komponenta je PCI most koji obezbedjuje spregu procesora, skrivene memorije i operativne memorije sa periferijama koje se prikljucuju preko konektora na magistrali. To znaci da periferni uređaj komunicira

sa **PCI magistralom** a ne procesorom direktno što dozvoljava laku zamenu procesora i kompatibilnosti PCI uredjaja na racunarima sa razlicitim CPU platformama. Izbegavaju se kasnjenja i uska grla. PCI uredjaji/kartice oznacene kao agent mogu biti vodeci ili prateci u zavisnosti ko pokrece 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.

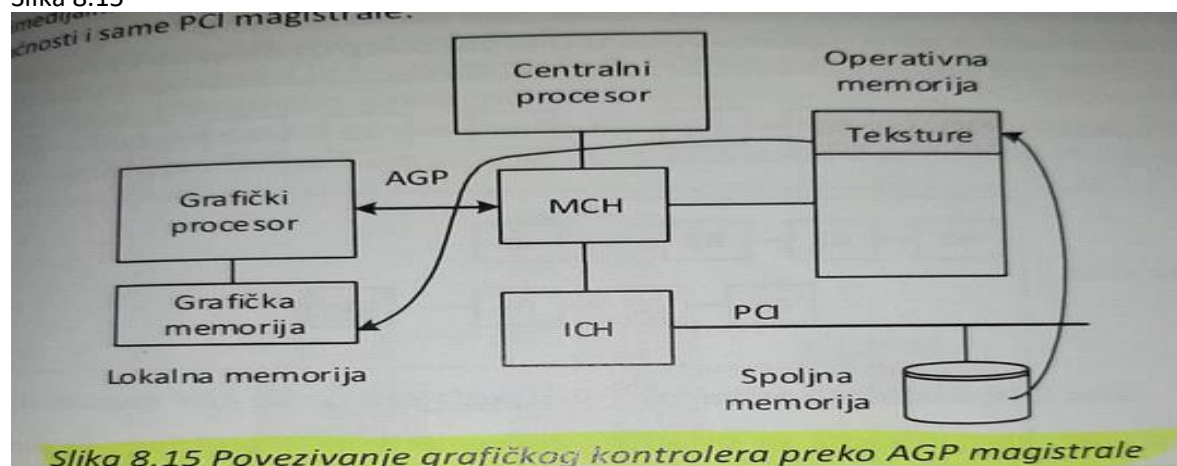
PCI PODRSKA KONCEPTU „PRIKLJUCI I KORISTI“:

Svaki PCI uredjaj u svom adresnom segmentu cuva posebnu tabelu koja sadrzi 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 zakljucno sa adresom 0Ch koji predstavlja **PCI zaglavlje**, a ostatak predstavlja **Prostor PCI uredjaja**. Sva polja koja se sastoje od vise bajtova su u „little endian“ obliku to jest nize adrese sadrže bajte manje tezine.

99. AGP magistrala ? - strana 259 slika 8.15

Slika 8.15



Slika 8.15 Povezivanje grafičkog kontrolera preko AGP magistrale

AGP magistrala nastaje kao rjesenje problema zagusenja PCI magistrale usled ogromne kolicine video podataka koje treba prenijeti u realnom vremenu, pod strogim vremenskim ogranicenjima u pogledu kasnjenja (**AGP-Accelerated Graphics Port**). AGP je razvijen u namjeri da obezbijedi sto vecu propusnost veze procesora i video kartice, potpunim uvazavanjem svih specificnosti video podsistema.

AGP magistrala je razvijana na osnovu PCI 2.1 standarda, spretni signali, AGP konektor su identicni PCI signalima i PCI konektoru. Jedna od kljucnih prednosti AGP je izolacija video podsistema u odnosu na ostatak PC racunara cime se PCI uredjajima obezbedjuje dodatni prenosni opseg.

Komunikacija sa sistemskom memorijom podrzana je sa dva posebna rezima rada.

- Protocni rezim (pipe) - serija upita se zadaje u sekvenci, bez cekanja da se prethodno izdati zahtev završi.
- Uporedno adresiranje – 8 dodatnih linija za adresiranje, cime se novi ciklus pristupa memoriji zapocinje u toku trajanja prethodnog.

100. USB ?

USB:

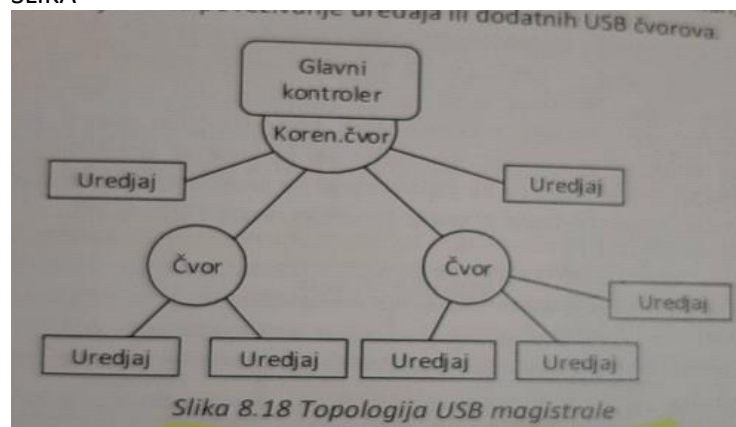
(USB – Universal Serial Bus), osnovna ideja Univerzalne serijske magistrale je resavanje iznijetih problema izmestanjem sporih uređaja iz racunara, tacnije izvan njegovog memorijskog i U/I prostora. Komunikacija sa njim se ostvaruje serijski, posredstvom specijalizovanog kontrolera povezanog na PCI magistralu, 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 dvoicne linije,
- brzina prenosa 1.5 Mbit/s (spori), 12Mbit/s (uređaji pune brzine), 480 Mbit/s (uređaji visoke brzine),
- automatsko prepoznavanje priključenja 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

hoces

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, cija 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 izmedju sporih uređaja i magistrale.
- Dozvolu i zabranu komunikacije preko prikljuca.
- Prepoznavanje priključenja uređaja i detekciju njegove brzine prenosa.
- Kontrolu napajanja na svojim priključcima, sto podrazumijeva ukljucivanje/iskljucivanje napajanja.
- Smanjivanje struje napajanja pojedinacnog uređaja.
- Odredjivanje i postavljanje statusa (stanja) prikljucka.

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 MAGISTRALE:

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.

101. PCI express magistrala ? - strana 271 slika 8.31

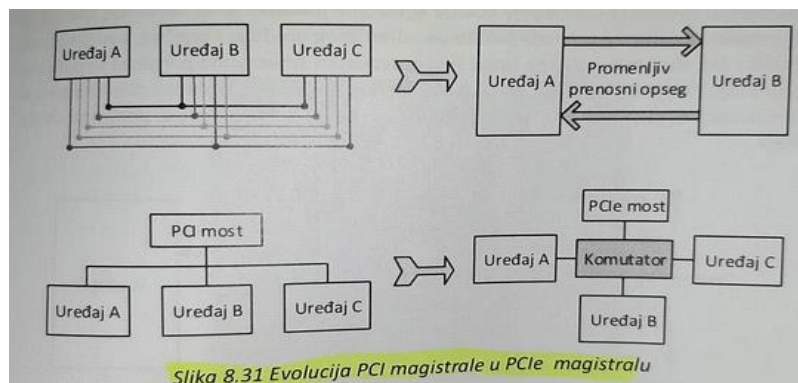
Serijalizacijom PCI magistrale, tj. zamjene visestrukih paralelnih vodova sa brzim serijskim komunikacionim vodom nastaje PCI Express (PCIe) magistrala.

Dvije razlike u odnosu na PCI:

-dijeljeni paralelni signali zamijenjeni su direktnim komunikacionim vodovima izmedju pojedinih uređaja,

-uvedena je prekidacka kontrolna logika, **komutator (switch)** koja posreduje u uspostavljanju veze izmedju PCI uređaja i procesora omogucujuci uzajamnu komunikaciju uz minimalna kasnjenja.

Slika 8.31



8.32, 8.33

To znaci da se u PCIe sistemima izmedju izvora i odredista uvek formira veza tipa tacka-tacka, pri cemu se poruke usmeravaju posredstvom komutatora na nacin karakteristican za komunikacione sisteme. Arhitektura PCIe magistrale je organizovana po nivoima. Kompatibilnost sa PCI je zadržana. -*Brzina prenosa podataka se moze povecati kombinovanjem vise osnovnih PCIe kanala - staza (lane) u zajednicki prenosni put-tok(run).

ZAKLJUČAK: PCI Express je komponenta koja je dovela do unapredjenja savremenih racunara, povecanjem brzine prenosa i uzajamnu nezavisnost izmedju komponenti.

Dodatno, implementacija PCIe standarda nije skuplja u odnosu na ta ista rjesenja i ne poskupljuje proizvodnju PC racunara.

(VIDEO PODSISTEM, GRAFICKA KARTICA?):

Video podsistem (mogucnost grafickog prikaza podataka) se sastoji od graficke kartice i monitora za prikaz slike.

Graficka kartica je komponenta koja rukuje vizuelnim izlazom racunara, odnosno onim sto se prikazuje na ekranu monitora. Moderna graficka kartica se sastoji od tri glavne komponente:

- grafickog procesora
- video memorije i
- tkz. RAMDAC (RAM to Digital/Analog Converter) kola.

Slika 8.37.

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 izvršiti 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 izvršenju 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 ekonomiconosu primjene.

103. RAMDAC ? - strana276 slika 8.37 i strana 278 slika 8.38

Zadatak **RAMDAC** kola je konverzija digitalne slike iz RAM memorije u analogni **RGB signal** koji se salje ka monitoru. Mnogo puta u sekundi **RAMDAC** cita sadrzaj video memorije i uz pomoc D/A konvertora po jednog za svaku od tri boje, digitalni zapis pretvara u tri analogna signala odredjenog naponskog nivoa. Kombinacija tri signala tačno definise boju prikazanog piksela. Brzina konvertovanja odredjuje brzinu osvjezavanja graficke kartice. Bolja **RAMDAC** kola rade na 200-250 MHz i realizovana su kao posebno kolo van grafičkog procesora.

Analogna video sprega obuhvata tri video signala (R,G,B) i signale za sinhronizaciju potrebne prilikom osvjezavanja slika na monitoru. Ovakav nacin povezivanja monitorom je inicijalno definisan standardom VGA i do danas je pod istim imenom.

(DIGITALNE VIDEO KARTICE) (NE TREBA)

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 elektricitetom, nepravilnim rukovanjem, iznenadnim potresom, otkazom neke komponente.

105. Redundantni skup nezavisnih diskova (RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5 i RAID 6) ? - sve slike za RAID

U cilju unapređenja kapaciteta, brzine i pouzdanosti rada disk jedinice, razvijen je koncept koji podrazumijeva rad više standardnih diskova prihvatljive cijene. Njihov rad objedinjuje OS, ali mnogo česće posebno projektovan kontroler diska. Ovakvo rješenje je poznato kao **RAID Disks**, osobine su:

- RAID je skup nezavisnih fizičkih diskova koje OS vidi kao jedinstveni logički disk.
- Podaci se distribuiraju na sve diskove u okviru RAID uređaja.
- Dodatni prostor diskova se koristi za čuvanje informacija o paritetu, čime se obezbeđuje mogućnost oporavka podataka u slučaju otkaza jednog ili više fizičkih diskova u okviru RAID jedinice.

RAID nudi: veći logički kapacitet, višestruko ubrzanje pristupa i već spominjanu sigurnost podataka. Uobičajena je podjela RAID rešenja u 7 nivoa. 2 i 4 nisu našli primjenu u praksi a 0 ne zadovoljava 3 uslov.

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 vrsi citanje i zapis na dva diska u paru. **Prednosti:** moguće je istovremeno citanje sa dva diska, čime se dvostruko ubrzava pristup zapisanim podacima.

Zapis ide na oba diska, što rezultira istom brzinom kao na pojedinacnom 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**

108. Raid 2

Koristi tehniku paralelnog pristupa, što znači da svi pojedinačni diskovi učestvuju u izvršenju svakog U/I zahtjeva. Diskovi su međusobno sinhronizovani, tako da su sve glave za svaki od diskova u svakom trenutku u istom položaju. Zapis podatka se vrši u isječcima, vrlo kratkim, često reda byte – a ili rijeci. Računa se Hemingov kod na odg. diskovima za detekciju(jednostrukih) i korekciju(dvostrukih) gresaka. On se čuva na dodatnim diskovima, čiji je broj proporcijalan broju diskova podatka (N-1) . **Prednosti** ove tehnike je „on-line“ korekcija gresaka, visoka brzina prenosa podataka i relativno jednostavna realizacija u odnosu na više RAID nivoe. **Nedostatak** je jer uzrokuje veliki gubitak kapaciteta instaliranih diskova, broj transakcija zbog međusobne sinhronizacije diskova je isti kao i kod pojedinačnih diskova i na kraju visoka cijena realizacije. **Slika 9.8**

109. Raid 3

U svemu je sličan prehodnom rješenju, ali se umjesto Hemingovog koda računa 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-ILL 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 greške 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**

111. Raid 5

Je vrlo sličan RAID 4, jedino se blokovi pariteta zapisuju na sve diskove, a ne na samo jedan kao kod RAID 4. Na taj način se izbjegava usko grlo vezano za disk pariteta, kojem se kod nivoa 4 pristupa pri svakom zapisu. Uobičajeno je čuvanje blokova pariteta po principu kružne distribucije (round-robin), gdje se za svakih N blokova, paritet zapisuje na preostali disk. Ovakva organizacija obezbeđuje najviše performanse u pogledu učestanosti zahtjeva za učitavanje podataka, srednje pri zapisu. Ukupna propusnost sistema je visoka, kao i efikasnost u korišćenju diskova. Sama realizacija, tačnije logika RAID 5 kontrolera, je za nijansu složenija od nivoa 4. **Slika 9.11**

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 rjesenje, poznato i kao RAID sa P+Q redundacijom, omogućuje regeneraciju podataka i u slucaju dvostrukog otkaza osnovnih diskova. Implementacija RAID 6 tehnike zahtijeva N+2 diskova. Jedna od varijanti implementacije podrazumijeva racunanje parnosti po redovima (P-parnost) i po kolonama (Q-parnost). Ovakvo rjesenje, najbolje za kritичne aplikacije, istovremeno je i najsloženije. Javlja se niza brzina zapisa, uslovljena racunanjem i zapisom dvostruke parnosti. **Slika 9.12**

Милорад Максић ПР104/2020

Мастиловић Радослав ПР106/2020