1. **Pojam arhitekture računala**

Izraz „Arhitektura računala“ uveden je 60tih godina u IBM-u. Upotrebljavan je za opisivanje **programskog modela računala na razini strojnog jezika** (asemblera). S vremenom se taj pojam proširuje na **algoritme koji se upotrebljavaju u osnovnim funkcionalnim jedinicama** računala (ALU, CU, U/I, MEM.)

Ona se općenito može definirati kao **slika računarskog sustava kojom se sustav predočava programeru u strojnom jeziku**, ali i programeru koji piše prevodioce za više programske jezike kojim se naredbe u višim programskim jezicima prevode u strojne instrukcije.

1. **Hijerarhijski model računala**

Struktura računalnog sustava mora biti takva da ostvaruje sljedeće ciljeve:

* Povećanje propusnosti (eng. **Throughput**)
* Povećanje prilagodljivosti (eng. **Flexibility**)
* Povećanje pouzdanosti (eng. **Reliability**)
* Povećanje raspoloživosti (en. **Availability**)
* Ostvarivanje niže cijene sustava (eng. **Low Cost**)

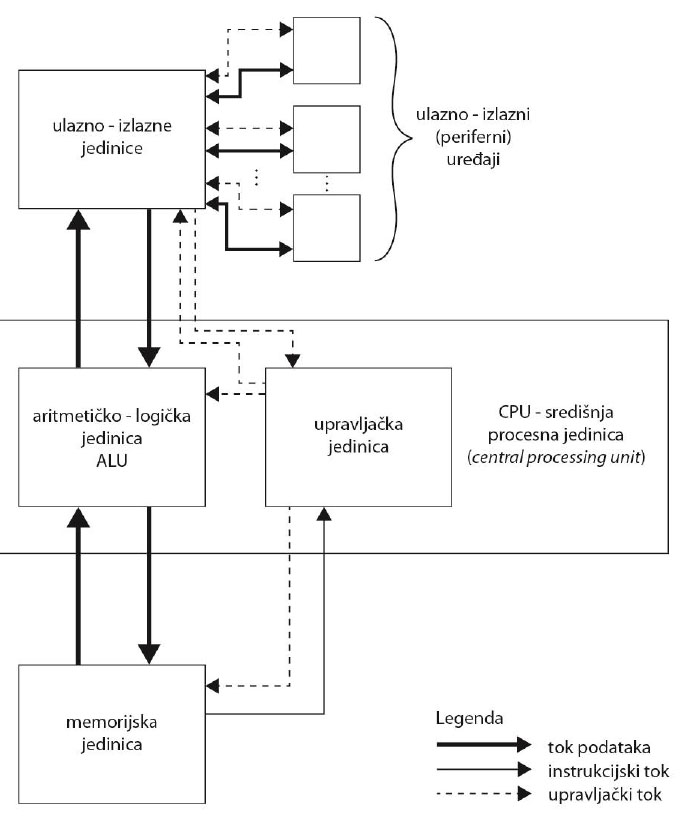
Ostvarenje nabrojanih ciljeva ostvaruje se zahvatima na **trima** sustavnim elementima arhitekture:

* Sklopovlju (**Hardware**)
* Programskoj opremi (**Software**)
* Odnos čovjek - računalo i način primjene računala (**Humanware**)

1. **Pet funkcijskih jedinica koje čine računalo**

* Aritmetička jedinica
* Upravljačka jedinica
* Memorija
* Ulazna jedinica
* Izlazna jedinica

1. **Von Neumannov model računala – nacrtati**



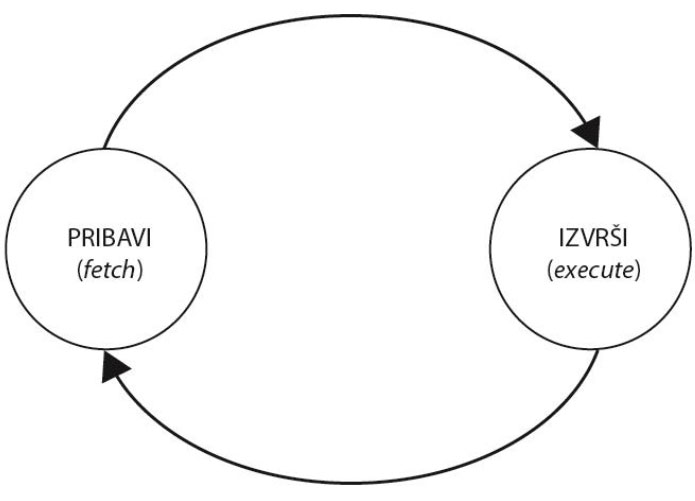
1. **Osnovna obilježja PC-a (IBM standard)**

* Računalo koristi jedan od **procesora kompatibilnih s Intelovim procesorom 80X86**
* Računalo je izgrađeno na osnovu **ISA specifikacije** (Industry Standard Arhitecture)
* Računalo koristi jednu od **sabirnica kompatibilnih s ISA ili PCI sabirnicu**, uključujući i odgovarajuće utore za proširenje.
* Računalo koristi **BIOS kompatibilan s IBM-ovim.**
* Računalo je u stanju izvršavati programe kompatibilne **s operacijskim sustavom MS-DOS i MS-Windows.**

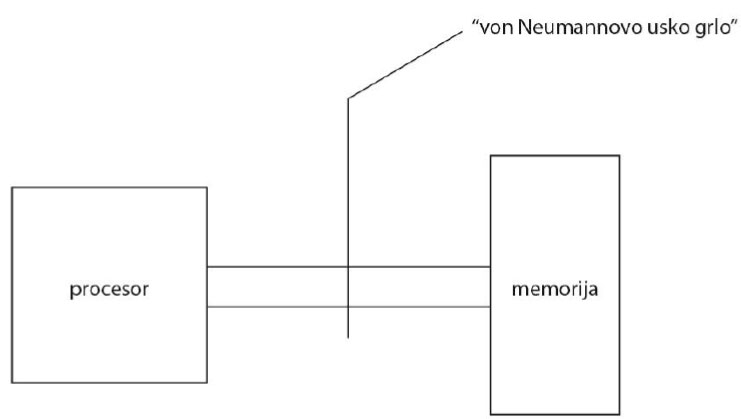
1. **Podsustavi PC-a**

* kućište s napajanjem
* matična ploča
* procesor
* memorija
* diskovni podsustav
* video podsustav
* audio podsustav
* mrežni podsustav
* osnovni ulazno izlazni uređaji
* ostalo

1. Izvođenje strojne instrukcije
2. **Dijagram stanja instrukcijskog ciklusa za von Neumannov model računala**



1. **Pojednostavnjeni model von Neumannovog računala**



1. **Osnovni moduli mikroračunala**

* mikroprocesor
* memorijski moduli
* ulazno-izlazni moduli
* moduli posebne namjene
* pomoćni sklopovi (izvori napajanja, generator signala vremenskog vođenja)

1. **Matična ploča**

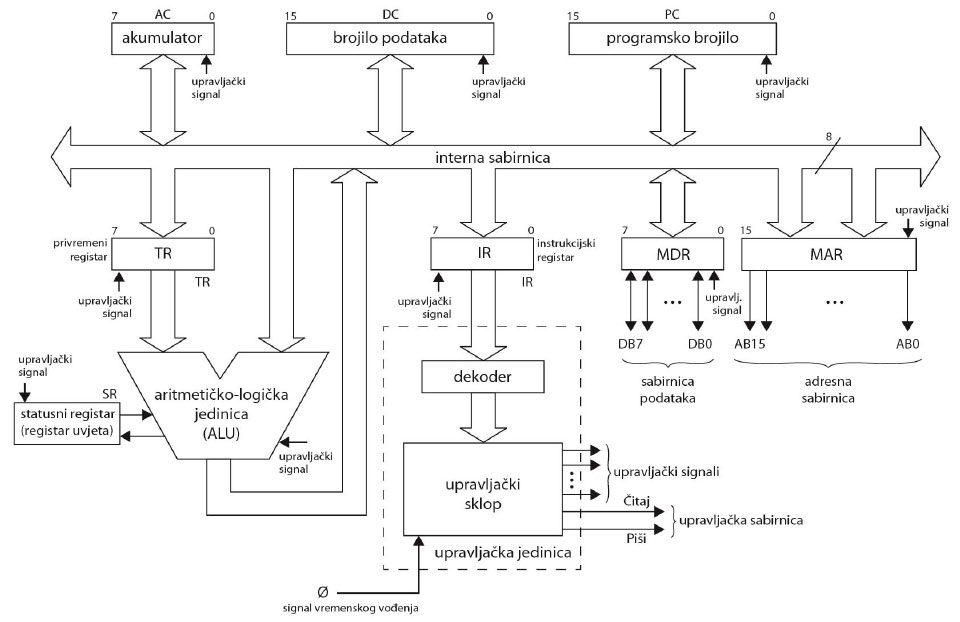
Matična ploča je osnovni element računala. Ona međusobno povezuje sve komponente osobnog računala. Na njoj se nalaze:

Procesor, chipset, BIOS, memorija, sabirnice, utori sabirnica, svi konektori potrebni za komunikaciju s periferijom. Konektori za diskovne uređaje, konektori na stražnjoj strani ploče i ostali integrirani podsustavi.

1. **Koji parametri karakteriziraju matičnu ploču**

* Vrsta sabirnica
* Arhitektura
* Chipset
* BIOS
* Organizacija memorije

1. **Pojednostavljeni model 8-bitnog CISC procesora**



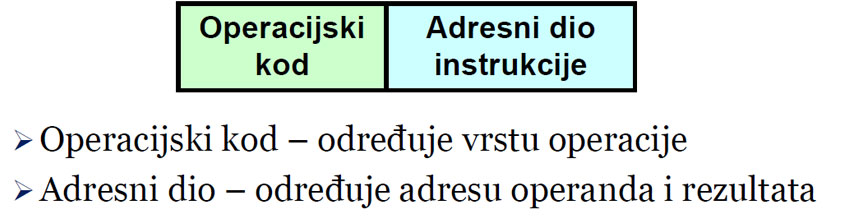
1. **Izvršavanje instrukcija**

Instrukcije se izvršavaju u dvije faze. U fazi **PRIBAVI** procesor pribavlja instrukciju (koja se sastoji od opt koda i adrese operanda ili operand). Za vrijeme faze **IZVRŠI** procesor dekodira kod i izvršava operaciju koja je zadana operacijskim kodom. Važno je još naglasiti da su instrukcije kao i podaci pohranjeni u istom obliku i istoj memoriji

1. **Elementi standardne arhitekture procesora**

* Upravljačka jedinica
* Aritmetičko-logička jedinica
* Jedan ili više akumulatora
* Registri opće namjene
* Adresni registri
* Interne sabirnice

1. **Asembler - Oblik instrukcijske riječi**



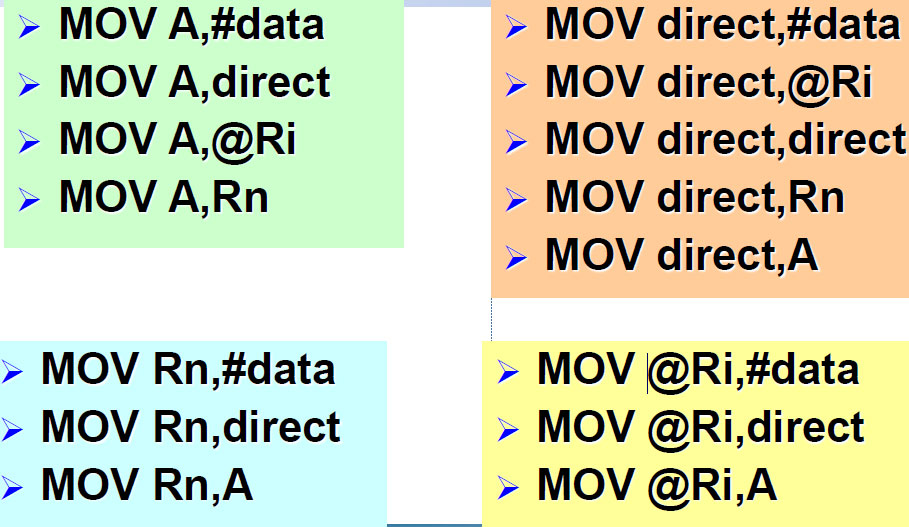
1. **Asembler – načini adresiranja**

* Neposredno adresiranje konstante; ADD A, #data ADD A, #15h
* Direktno adresiranje bajta unutarnjeg RAM-a; ADD A, direct ADD A,20h
* Indirektno adresiranje preko registara R0 i R1; ADD A,@R0
* Adresiranje općih registara; ADD A, Rn

1. **Asembler – aritmetičke instrukcije**

* instrukcije za zbrajanje i oduzimanje (ADD, ADDC, SUBB)
* instrukcije za množenje i dijeljenje (MUL, DIV)
* instrukcije za uvećanje i umanjenje sadržaja za 1 (INC, DEC)
* instrukcije za podešavanje sadržaja nakon zbrajanja pakiranih BCD brojeva (mnemonik DA)

1. **Asembler – instrukcije prijenosa podataka**



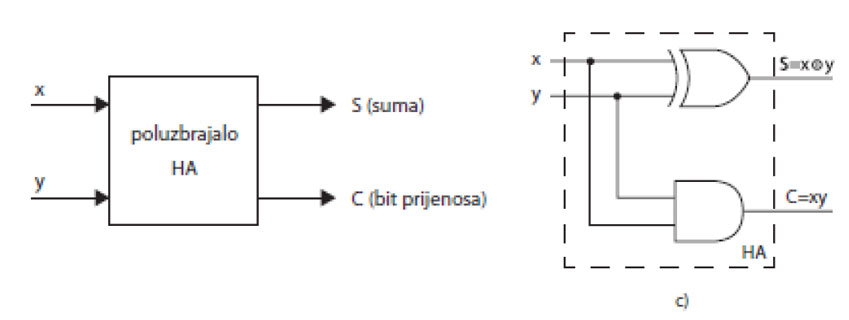
1. **Asembler – instrukcije DJNZ i CJNE**

Instrukcije DJNZ obavlja funkciju dekrementiranja i usporedbe te skoka iz petlje kada varijablu izjednačimo s nulom. Slično radi i instrukcija CJNE, gdje vršimo usporedbu između dvije varijable, i kada one postanu jednake izlazimo van petlje.

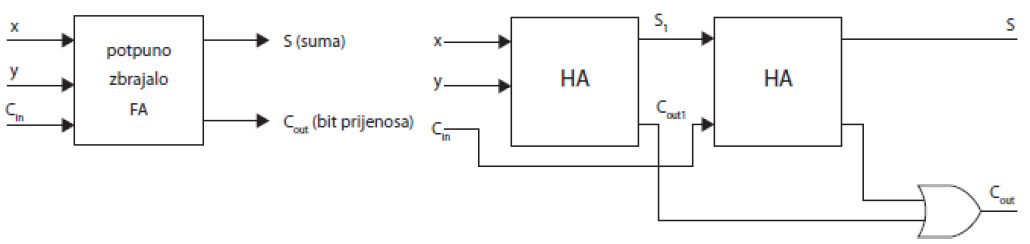
1. **Operacijski kod**

Operacijski kod (OP code) je dio instrukcijeske riječi koji određuje koju operaciju procesor treba obaviti. To je strojna riječi koju procesor razumije. Operacijski kod dobiva slovčanu kraticu koja podsjeća na djelovanje instrukcije – mnemonik. Najčešći mnemonisi su ADD, SUBB, MUL, DIV, INC.

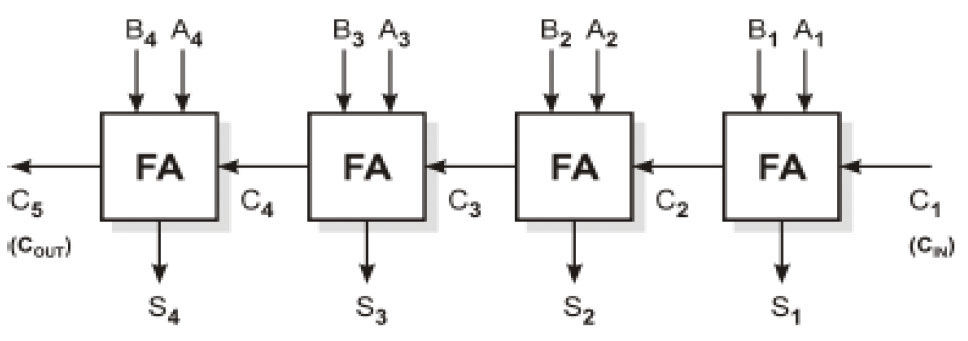
1. **Poluzbrajalo HA - nacrtati na razini modula**



1. **Potpuno zbrajalo FA - na razini modula**



1. **Paralelno zbrajalo – nacrtati**

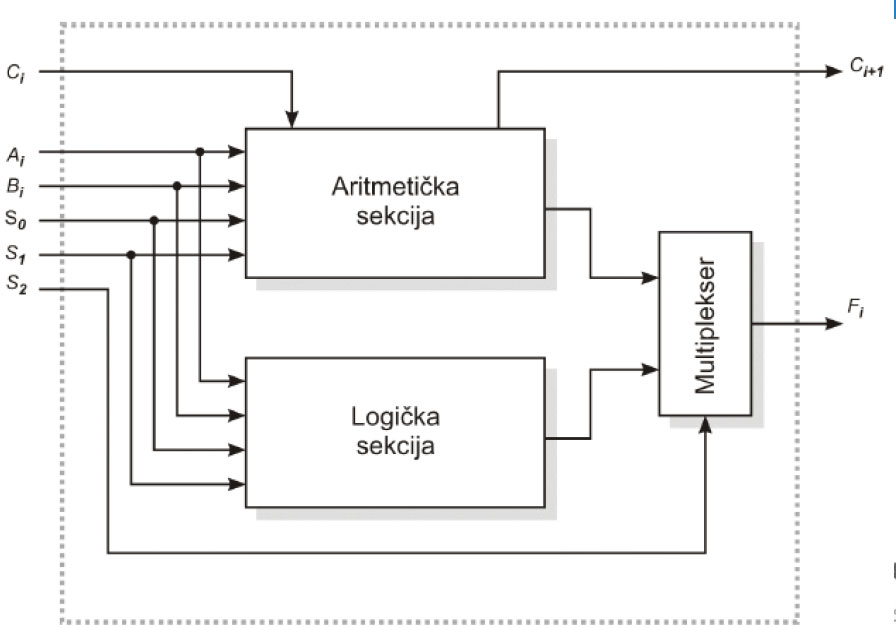


1. **ALU – funkcionalni prikaz**

Svaki stupanj ALU može se u funkcionalnom smislu prikazati kao:

* Kombinacija aritmetičke sekcije
* Kombinacija logičke sekcije

1. **i-ti stupanj ALU**



1. **Ulazi i izlazi ALU**

* Ai i Bi su ulazi za operande
* Ci je bit prijenosa prethodnog stupnja
* Fi je jednobitni rezultat aritmetičke ili logičke operacije
* Ci+1 je bit prijenosa u sljedeći stupanj
* Upravljački ulazi S0, S1 i S2 upotrebljavaju se za izbog različitih aritmetičkih i logičnih operacija

1. **Oblikovanje jednostavne ALU**

Kod oblikovanja jednostavne aritmetičko logičke jedinice koristi se sljedeći pristup:

Prvo se oblikuje sklopovlje aritmetičke sekcije neovisno o logičkoj sekciji

Zatim se određuju logičke operacije koje se mogu izvesti sklopovima iz aritmetičke sekcije  
Na kraju se izvode preinake na sklopovima kako bi se mogle izvoditi sve željene logičke operacije

1. **Upravljačka jedinica \_ četiri temeljne funkcije**

* Uspostavljanje određenog stanja tijekom svakog instrukcijskog ciklusa
* Određivanje sljedećeg stanja na temelju trenutnog stanja, stanja zastavica (flagova) u statusnom registru i stanja na ulaznim upravljačkim linijama procesora
* Pohranjivanje informacije koja opisuje tekuće stanje u kojem se procesor nalazi
* Generiranje upravljačkih signala za izmjenu podataka između procesora i drugih funkcijskih jedinica.

1. **Mikroprogramirana upravljačka jedinica**

Središnja građevna sastavnica upravljačke jedinice je upravljačka memorija (mikroprogramska memorija, eng. Control memory), u njoj su pohranjeni „Upravljački signali“ u formatu poput strojnih instrukcija u programu. U tom obliku pohranjene upravljačke signale nazivamo mikroinstrukcijama. Sljedovi takvih mikroinstrukcija čine jedan ili veći broj programa koji se nazivaju mikroprogramima.

Upravljačka memorija mikroprogramirane upravljačke jedinice pohranjuje više mikroprograma.

Svakoj strojnoj instrukciji iz instrukcijskog skupa odgovara jedan mikroprogram koji će se pobuditi na temelju: operacijskog koda strojne instrukcije, na temelju vanjskih uvjeta ili na temelju stanja zastavica statusnog regista.

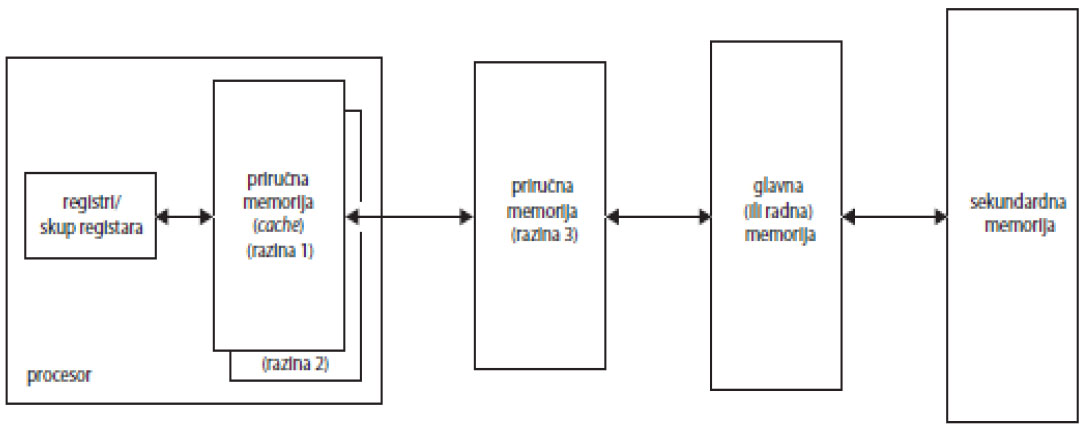
Drugim riječima mikroprogramska memorija sadržava skup mikroprograma kojima se emulira skup strojnih instrukcija procesora, možemo na to gledati kao na računalo unutar računala.

Upravljačka jedinica tijekom izvođenja mikroprograma prolazi kroz mikrofazu – pribavi i mikro-fazu izvrši.

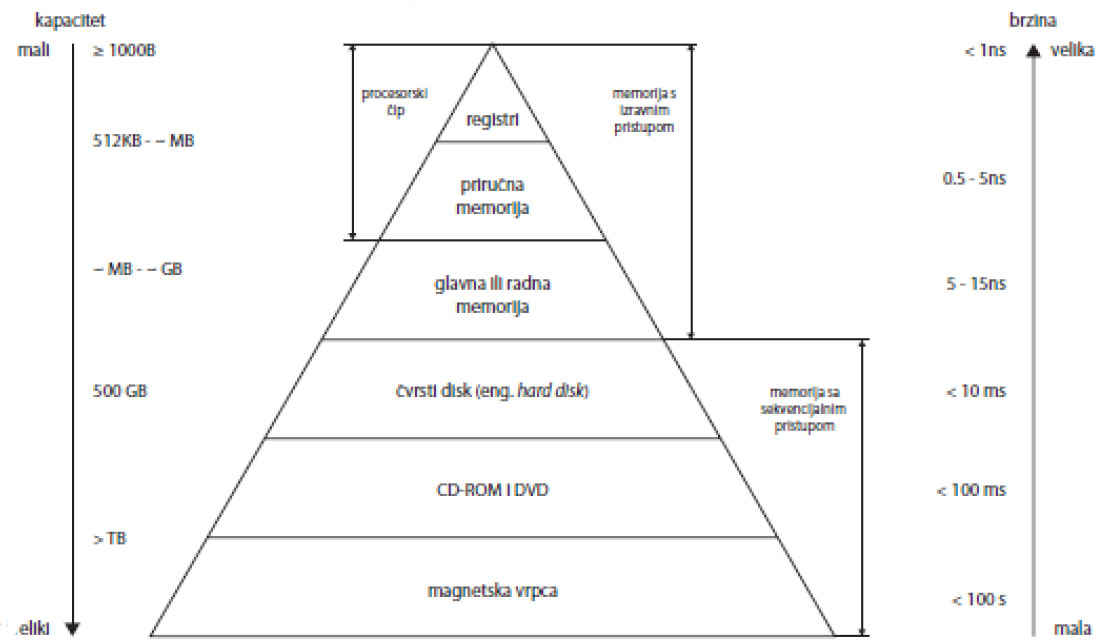
1. **Mikroprogram i makroprogram**

Element iz niza upravljačkih signala nazivamo **mikroinstrukcijom**. Niz takvih mikroinstrukcija čine **mikroprogram**. Mikroinstrukcija je uzrok jedne ili više osnovnih operacija na razini prijenosa podataka ili aktiviranja pojedinih sklopova, takva osnovna operacija nazive se **mikrooperacija**. Instrukcija pribavljena iz memorije (makroinstrukcija) uvjetuje izvođenje čitavog niza **mikroinstrukcija**. Niz makroinstrukcija čini makroprogram koji je smješten u glavnoj memoriji.

1. **Memorijska hijerarhija – nacrtati**



1. **Simbolički prikaz memorijske hijerarhije u računarskom sustavu**



1. **Sekundarna memorija karakteristike i tipovi**

Sekundarna memorija ima tri važne dobre značajke, to su vrlo veliki kapacitet, neizbrisivost i malu cijenu po pohranjenom bitu podatka. Nedostatak leži u relativno velikom vremenu pristupa.

Sekundarna memorija predstavlja proširenje glavne memorije i služi primarno za rezervnu pohranu podataka i programa.

Tipovi sekundarne memorije: magnetska diskovna memorija, optička memorija, trake

1. **Magnetska diskovna memorija – podsustavi**

Kad govorimo o podsustavima magnetske diskovne memorije, govorimo o podsustavu koji čine fizičke komponente (sklopovski) i elektronički podsustav (eng. Drive Electronics)

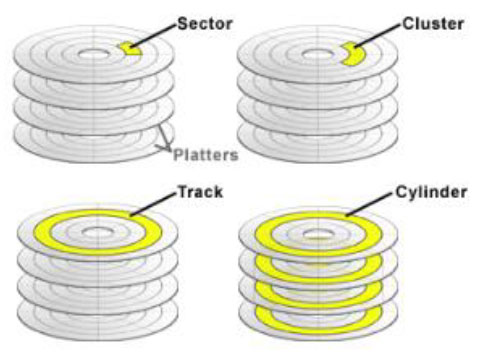
1. **Podsustav koji čine fizičke komponente**

* Glava za čitanje i pisanje
* Ručice na kojima se nalaze glave
* Jedna ili više kružnih ploča presvučenih magnetskim materijalom (diskovi)
* Elektromotor (spindle)

1. **Podsustav koji čini elektronički podsustav**

Elektronički podsustav diska sastoji se od **upravljačke jedinice diska**, **memorije**, **sklopova za zapis i čitanje podataka** te **sklopova za upravljanje** aktuatorom (ručicom) i elektromotorom.

1. **Organizacija podataka na disku - staza, cilindar, sektor, klaster**



1. **Organizacija podataka na disku – sektor**



1. **Dinamički parametri diskovne jedinice**

Dinamički parametri diskovne jedinice odnose se na vrijeme pristupa podacima tijekom operacije pisanja ili čitanje. To vrijeme određuje se na temelju triju operacija: **pozicioniranje glave** za čitanje i pisanje na odgovarajuću stazu, **pristup željenom sektoru** koji ovisi o rotacijskom kašnjenju (ovisno o brzini diska) te **prijenosu** samih podataka sa diska.

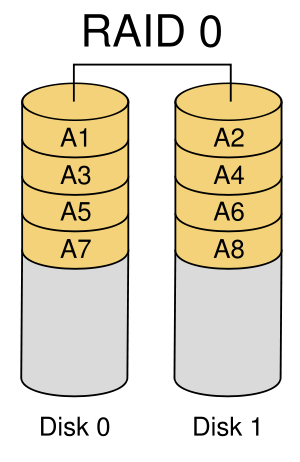
1. **Polje diskova RAID - osnovne značajke**

RAID je polje hard diskova (2 ili više) koji se prema vanjskoj logici ponašaju kao jedna cjelina. Ono može raditi na dva načina, softwerski i hardverski.

Pojavljuje se u 6 osnovnih načina izvedbe. Svaki od njih je karakterističan prema tome se diskovi uvijek prema sustavu prikazuju kao jedna logička jedinica. Podaci su raspoređeni po svim diskovima. Zalihosni diskovni kapacitet upotrebljava se za pohranu zaštitnog koda koji jamči oporavak podataka u slučaju kvara diskovne jedinice.

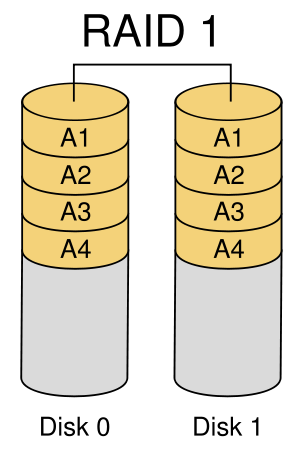
1. **Polje diskova – RAID 0**

RAID 0 nema zalihosti. U toj su izvedbi podaci porazdjieljeni na sve diskove u polju. Prednost mu je velika brzina prilikom rada,mana je u maloj sigurnosti podataka. Primjena u manjim serverima gdje je potrebna velika brzina pri radu.



1. **Polje diskova – RAID 1**

Kod RAID-a 1 zalihost je 100 %. Podaci su zrcaljeni na 2 ili više diska. Ako jedan disk ispadne iz rada, jednostavno se rabi zrcaljeni disk umjesto njega. Glavni nedostatak mu je cijena jer takva diskovna organizacija zahtjeva dva puta veći diskovni prostor.



1. **Struktura diska s jednom primarnom particijom**



1. **Master Boot Record – MBR**

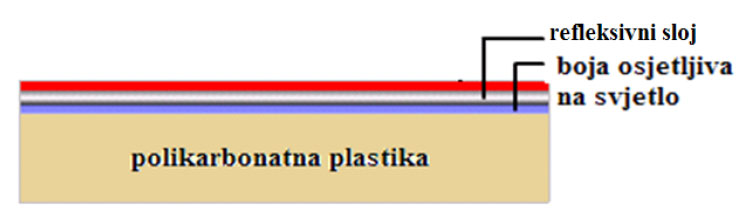
Master Boot Record je područje veličine 512 B koje se nalazi na početku prvog diska, CHS adresa mu je 001. MBR se kreira pri formatiranju particije na tvrdom disku programom FDISK. Osnovna namjena mu je pronaći i učitati operacijski sustav. Sadrži 3 glavna dijela: MBR kod, particijsku tablicu i magičnu riječ.

1. **Optičke memorije - način pohrane i čitanja podataka**

Optičke memorije su strukturirane tako da se informacija pohranjuje u binarnom obliku uzduž spiralne staze diska. Podaci se pohranjuju u obliku udubina ili izbočina, koji predstavljaju nule ili jedinice. Čitaju se pomoću lasera refleksijom svjetla od reflektirajućeg aluminijskog sloja na disku.

1. **Objasniti razliku CD-R i CD-RW**

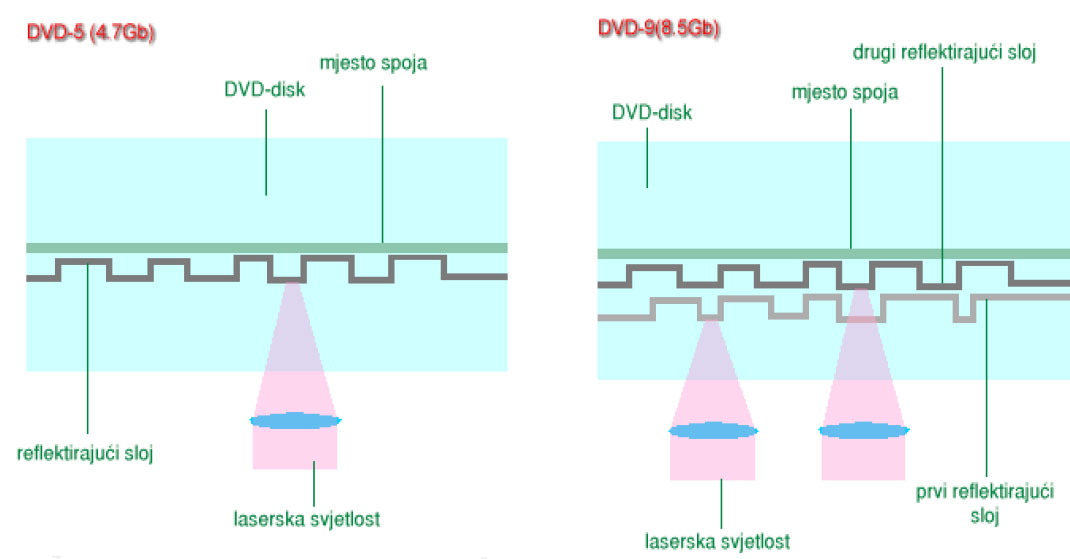
CR-R je građen od nekoliko slojeva. Nosivi dio je **polikarbonatni** disk, a na njega je nanesen sloj **organske boje osjetljive na svjetlo** (služi za snimanje podataka).

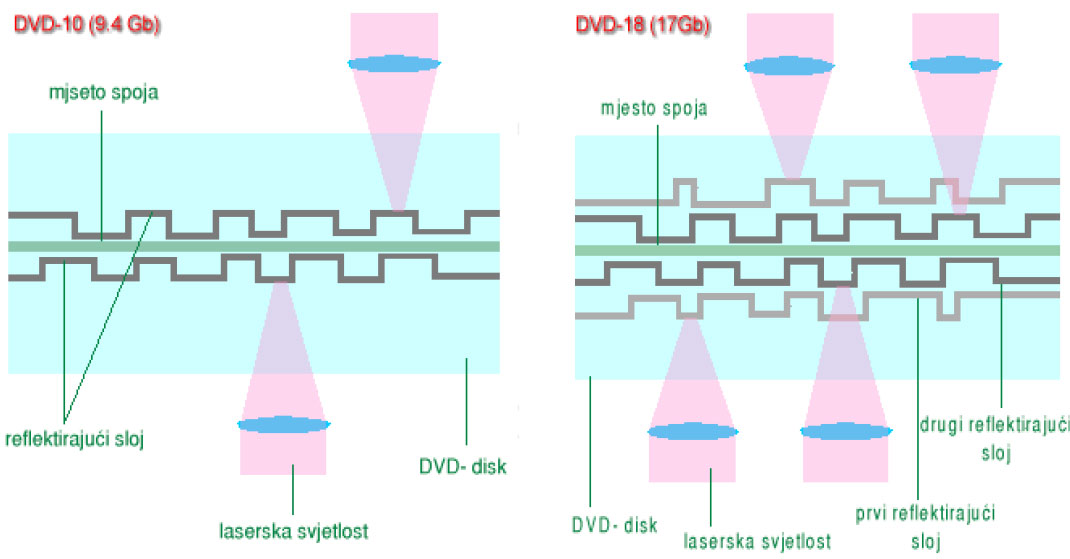


S druge strane, CD-RW je složene strukture. Pisanje i brisanje omogućava **posebni sloj** u disku koji ima svojstvo da pri zadanoj temperaturi koja dolazi iz lasera mijenja svoje stanje iz **kristalnog** u **amorfno**.



1. **DVD - kapaciteti i izvedbe**





1. **Pojam U/I sustava i U/I podsustava**

Sustav putem kojeg se omogućava komunikacija računala s vanjskim svijetom. U/I sustav mora podržavati raznolike oblike izmjene podataka između procesora i periferije.

1. **Periferni uređaji - pojam i moguće podjele**

Izmjena podataka računala s vanjskim svijetom ostvaruje se različitim perifernim uređajima koje još nazivamo i vanjski uređaji ili ulazno-izlazni uređaji. Periferni uređaji su preko ulazno-izlaznog sučelja priključeni na sabirnicu računala.

* Periferni se uređaji mogu razvrstati prema tome jesu li:
* Ulazni, izlazni ili ulazno-izlazni
* Jesu li namijenjeni komunikaciji s čovjekom ili strojem (s drugim računalom?)
* Prema brzini prijenosa podataka

1. **Osnovna izvedba ulazno-izlaznog upravljača**

Ulazno-izlazni upravljači, ovisno o funkcijama koje obavljaju, mogu biti izvedeni kao:

* Jednostavno sučelje između procesora i jednog perifernog uređaja
* U složenijim izvedbama mogu nezavisno upravljati većim brojem perifernih uređaja

U oba slučaja bitno je da podržava funkcije koje se odnose na međupohranjivanje podataka i upravljanje periferijom.

1. **Funkcije koje treba podržati ulazno-izlazni upravljač**

* Ulazno-izlazni upravljač mora upravljati vremenskim vođenjem i upravljanjem toka podataka između perifernih uređaja i internih komponenti računalnog sustava
* Komunicirati s procesorom i s perifernim uređajem
* Vršiti međupohranjivanje, tj. buffering
* Otkrivanje i ispravljanje pogrešaka, tj. debuging

1. **Načini izmjene podataka**

* Programirani ulazno-izlazni prijenos podataka
* Prekidni ulazno-izlazni prijenos podataka
* Prijenos podataka izravnim pristupom memoriji (DMA pristup)

1. **Programirani ulazno-izlazni prijenos podataka**

Podaci se izmjenjuju između procesora i ulazno-izlaznog upravljača pod **izravnim** **programskim** **upravljanjem** procesora. Procesor je taj koji izvodi program kojim izravno upravlja ulazno-izlaznim operacijama i to tako da očitava status perifernog uređaja, šalje mu naredbe i prenosi podatke.

1. **Tipovi programiranog UI prijenosa**

* Programirani bezuvjetni
* Programirani uvjetni

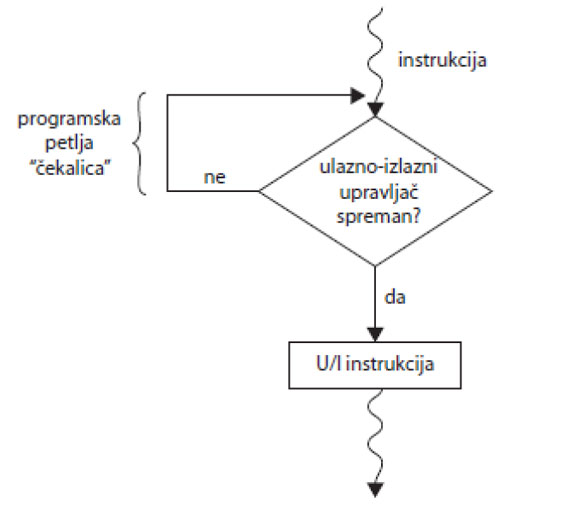
1. **Programirani uvjetni prijenos objasniti i nacrtati algoritam**

Takav pristup zahtijeva ispitivanje stanja ulazno-izlaznog upravljača, odnosno perifernog uređaja, prije nego li se obavi prijenos podataka.

Periferni uređaj je jednom ili većim brojem posebnih signalnih linija za rukovanje povezan s ulazno-izlaznim upravljačem ili točnije njegovim statusnim registrom kojem dojavljuje informaciju o spremnosti za prijenos podataka.

Ta se spremnost ogleda u promjeni stanja određenog ili određenih bitova u statusnog registru upravljača.

Ako periferni uređaj nije spreman za prihvat ili slanje podataka, programski se tok zatvara u petlju čekalicu. U petlji se uzastopce ispituje stanje statusnog registra. Tek kad je periferni uređaj spreman, izvodi se instrukcija za ulazno-izlazni prijenos podataka.



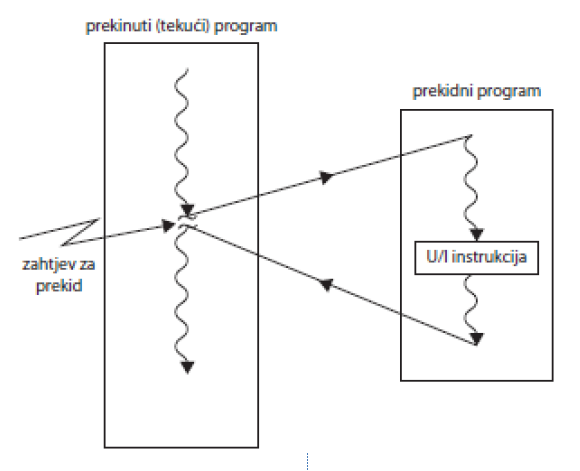
1. **Prekidni ulazno-izlazni prijenos podataka –pojam**

Procesor na zahtjev perifernog uređaja prekida izvođenje tekućeg programa i prelazi na izvođenje prekidnog programa kojim poslužuje periferni uređaj. Odnosno obavlja prijenos podataka.

Vanjski svijet, odnosno periferni uređaj koji je uzročnik ili izvod prekida upućuje procesoru zahtjev za prekid izravno posredstvom ulazno-izlaznog upravljača. Taj zahtjev procesoru stiže jednim ili većim brojem upravljačkih linija. Obično je više ulazno-izlaznih upravljača priključeno na jednu prekidnu liniju.

Postoji više ulazno-izlaznih upravljača koji mogu istodobno zahtijevati prekid. Prekidna linije je aktivna kada je u logičnom stanju 0.

1. **Shematski prikaz prekidnog prijenosa podataka**



1. **Postupak prekidnog prijenosa**

Periferni uređaj ili ulazno-izlazni upravljač aktiviraju prekidnu liniju. Procesor završava tekuću instrukciju prije nego što će odgovoriti na zahtjev za prekid. Procesor neposredno nakon završetka tekuće instrukcije ispituje postoji li zahtjev za prekid i ako postoji, utvrđuje razinu prekida te provjerava je li prekid maskiran (da li postoji prioritet). Ako procesor prihvaća prekid, on će signalnom potvrde prekidan INTACK obavijestiti ulazno-izlazne upravljače, odnosno UI uređaja o prihvaćanju prekida. Procesor sprema minimalni kontekst na stog. Puni se PC sadržajem koji predstavlja adresu prve instrukcije prekidnog programa, tj. adresu. Procesor započinje s fazom PRIBAVI i dohvaća prvu instrukciju prekidnog programa. Nakon što je prijenos podataka ostvaren, sadržaji se radnih registara obnavljaju tako da se uzimaju sa stoga, drugim riječima obnavlja se kontekst. Procesor nastavlja tamo gdje je stao.

1. **Vektorski prekid**

Vrsta prekida kod koje se uzročnik prekida izravno identificira jednoznačnim kodom koji ujedno služi za brzo određivanje početne adrese prekidnog programa. Početna adresa prekidnog programa naziva se još i prekidni vektor. Svakom od ulazno-izlaznim upravljača, priključenih na prekidnu liniju, jednoznačno je pridružen n-bitni kod, 8-bitni vektorski broj koji će ulazno-izlazni upravljač postaviti na sabirnicu podataka nakon što primi signal potvrde prekida od procesora. Procesor će unijeti sa sabirnice podataka vektorski broj i interno ga pohraniti, taj broj će upotrijebiti za generiranje početne adrese memorijskih lokacijama na kojima je pohranjena adresa prve instrukcije prekidnog programa.

1. **Ulančavanje**

U praksi broj ulazno-izlaznih upravljača premašuje broj raspoloživih parova linija IRQ. U tom se slučaju koristi shema ulančavanja signalnom linijom potvrde prekida INTACK. Ulazno-izlazni upravljač koji je najbliži procesoru ima najveći prioritet jer prvi prima signal INTACK. Ako je zahtijevao prekid ne prosljeđuje ga dalje drugim upravljačkim lancima.

1. **Izravni pristup memoriji DMA - pojam i osnovna obilježja**

Kod ovog tipa prijenosa podataka između memorije i periferije izravno ne sudjeluje procesor. Omogućuje najbrži prijenos podataka između vanjskog svijeta i memorije računarskog sustava.

Bitno je spomenuti DMA upravljač, posebni sklop koji se zbog svoje složenosti često naziva i DMA procesor.

1. **Obavljanje DMA prijenosa**

DMA prijenos obavlja se na temelju sljedeća tri glavna koraka:

* Inicijalizacija DMA upravljača
* DMA prijenosa podataka
* Završetka DMA prijenosa

1. **Načini DMA prijenosa podataka**

* Prijenos podataka krađom ciklusa
* Prijenos podataka u snopu
* Kombinacijom ovo dvoje

1. **Prijenos podataka krađom ciklusa**

Međusobno se isprepliću operacije DMA prijenosa s normalnim sabirničkim ciklusom u kojem procesor pristupa memoriji. S obzirom da procesore ne treba cijelo vrijeme sabirnicu, DMA prijenos se može obaviti kada je sabirnica slobodna. To se događa kada je procesor zauzet internom operacijom.

U tim kratkim vremenskim intervalima DMA upravljač preuzima upravljanje sabirnicom i obavlja prijenos podataka. Ovaj način prijenosa podataka je transparentan zašto što je nevidljiv procesoru i ne narušava njegove performanse.

1. **Prijenos podataka u snopu**

DMA upravljač prisvaja sabirnicu za vrijeme prijenosa cijelog bloka podataka. Brzina DMA prijenosa tada je iznimno velika i odgovara brzini koju omogućuje najslabija karika. Sporost perifernog uređaja kompenzira se tako da DMA upravlja ima spremnim većeg kapaciteta pa već spremni podati čekaju na DMA prijenos. Tijekom DMA prijenosa u snopu procesor je zaustavljen i ne napreduje s izvođenjem instrukcija.

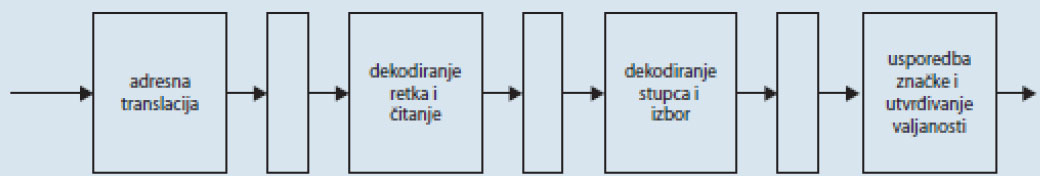
1. **Protočnost – pojam**

Protočnost ili Pipelining definiramo kao poseban oblik paralelizma. Istodobno izvođenje više operacija koji se dobiva rastavljanjem nekog složenijeg zadatka na manje pod-zadatke. Ti pod-zadaci jedan drugoga slijede određenim redom te se izvršavaju u dodijeljenim samostalnim jedinicama.

1. **Protočna struktura - pojam i nacrtati**

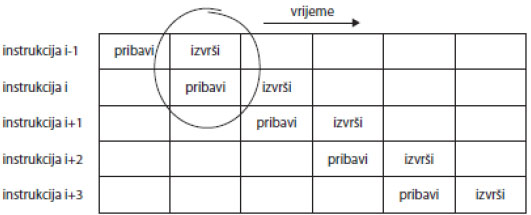
Postupak rastavljanja zadataka na manje pod-zadatke može se provesti na raznim razinama organizacije računarskog sustava.

Protočna izvedba priručne memorije sastoji se od četiri protočna segmenta. To su **adresna translacija**, **dekodiranje retka i čitanje**, **dekodiranje stupca i izbor** te **usporedba značke i utvrđivanje**.



S druge strane protočnost je vrlo važna u izvedbi upravljačke jedinice kojoj je osnovna funkcija izvođenje instrukcija. Broj identičnih zadataka koji se izvode vrlo je velik. Zadatak izvođenja instrukcija može se rastaviti na veći broj pod-zadataka. Pod-zadaci slijede u nizu i pogodni su za izvođenje u sklopovsko ostvarenim protočnim segmentima

Mi rastavljanjem takve osnovne funkcije upravljačke jedinice dobivamo instrukcijsku protočno strukturu. Protočni segmenti su potpuno samostalni u svojem radu i moguće je preklapanje.



Kada govorimo o instrukcijskoj protočnoj strukturi a RISC procesore ona se sastoji od sljedećih protočnih segmenata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IF | ID-OD | EX | ME | WB |

* Pribavljanje instrukcije (F)
* Dekodiranje instrukcije i dohvat operanada (ID-OD)
* Pristup memoriji (ME)
* Upis rezultata ili podatka dohvaćenog iz memorije (WB)

1. **Super-skalarni procesori – pojam**

Pojam super-skalarni procesori označava posebnu vrstu procesora koji se koriste tzv. skalarnim tipovima podataka – podacima jednostavne strukture kao što su, npr. Cjelobrojni (integer) ili brojevi s pomičnim zarezom (floating point). Za razliku od skalarnih tipova podataka, postoje i strukturni tipovi kao što su vektori, dvo i višedimenzionalna polja podataka, nizovi i skupovi.

1. **Osnovna značajka superskalarnih procesora**

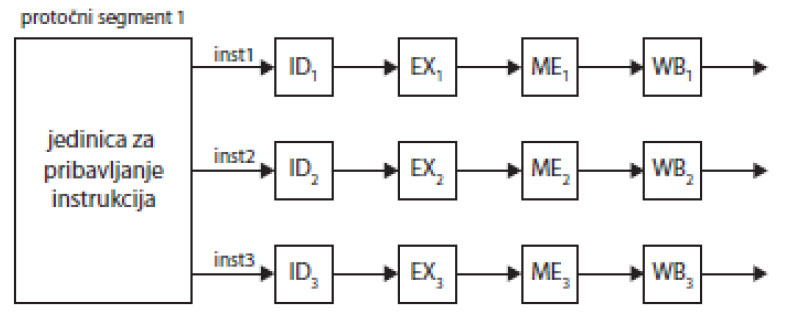
Osnovna značajka super-skalarnih procesora je da je prosječan broj perioda po instrukciji manji od jedan. To znači da se tijekom jedne periode signala vremenskog vođenja izdaje i izvršava veći broj instrukcija.

Superskalarni procesor iskorištava implicitni paralelizam koji je sadržan u slijedno napisanim programima.

Superskalarnost zahtijeva od procesora da detektira i iskoriste skriveni paralelizam na razini instrukcija.

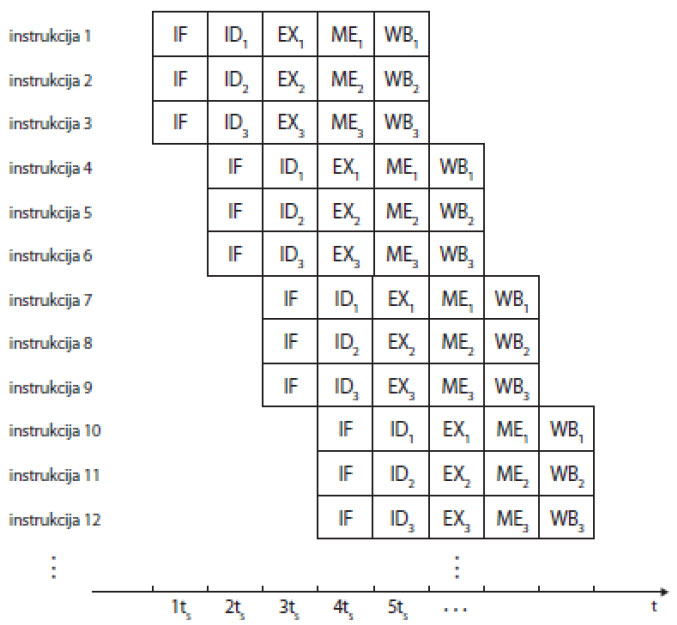
Superskalarni procesori izvedeni su od više izvršnih jedinica od kojih je svaka izvedena kao protočna. Dakle superskalarni procesor je procesor s većim brojem nezavisnih instrukcijskih protočnih struktura.

1. **Načelna organizacija superskalarnog procesora**



1. **Izvođenje slijeda instrukcija u superskalarnom procesoru**

Istodobno u svakoj periodi signala vremenskog vođenja iz tri nezavisne protočne strukture izlaze rezultati triju instrukcija.



1. **Paralelizam - dva različita konteksta**

U povijesti računarskih sustava možemo uočiti različite razine paralelizma koje se pojavljuju u procesoru. U početku oblik paralelizma temeljio se na bitovnoj razini. Očitovao se u povećanju duljine riječi procesora, od 4 bitnih, 8 bitnih, 16 bitnih pa sve do 32 bitnih i 64 bitnih procesora. Zahtjevi koji su pratili razvoj arhitekture procesora i koji su koristili paralelizam na razini bita bili su: **Povećanje izravno adresljivog prostora** (prije nego izravno povećanje performanse procesora), poboljšani načini prikaza i rukovanja **brojevima s pomičnim zarezom.**

Kada govorimo o paralelizmu, razlikuje dva različita konteksta:

* **Raspoloživi** paralelizam u programima
* **Iskorišteni** paralelizam koji se pojavljuje tijekom izvođenja programa

1. **Raspoloživi paralelizam**

Raspoloživi paralelizam sadržan je u biti samog problema i njegovom programskom rješenju. Može se govoriti o raspoloživom funkcijskom paralelizmu te o raspoloživom podatkovnom paralelizmu.

1. **Razine raspoloživog funkcijskog paralelizma**

Razlikujemo četiri razine raspoloživog funkcijskog paralelizma:

* Paralelizam na razini instrukcija
* Paralelizam na razini programskih petlji
* Paralelizam na razini procedura
* Paralelizam na razini program

1. **Iskorišteni paralelizam - pojam i razine**

Iskorišteni paralelizam jest onaj koji se pojavljuje tijekom izvođenja programa

* Paralelizam na razini instrukcija
* Paralelizam na razini dretvi
* Paralelizam na razini procesa
* Paralelizam na korisničkoj razini

1. **Osnovna značajka VLIW procesora**

To je arhitektura procesora s vrlo dugim instrukcijskim riječima (eng. Very Long Instruction word), a temelji se na tri ustaljena koncepta:

* Horizontalno mikro-programiranje
* Super-skalarna obrada
* Višestrukost funkcijskih jedinica

Osnovna značajka ove arhitekture procesora je vrlo duga instrukcijska riječ kojom se određuju višestruke operacije koje se mogu istodobno izvesti. Programer ili prevodilac (što je češći slučaj) izravno specificiraju operacije koje se mogu izvesti istodobno.

Procesor ima višestruke funkcijske jedinice koje dijele veliki zajednički skup registara. Svaka funkcijska jedinica izvedena je kao protočna jedinica. Operacije koje se izvode istodobno u funkcijskim jedinicama sinkronizirane su VLIW instrukcijama koje su duljine npr. 256, 512 ili 1024 bita

1. **Načini kombiniranja triju instrukcija u svežnju**

Instrukcijska riječ za procesor AI-64 arhitekture duljine je 128 bita i sadržava tri instrukcijske tipa RISC koje tvore tzv. 'svežanj' (eng. bundle).

Svaka od instrukcija u svežnju duljine je 41 bita pa u svežnju tri instrukcije zauzimaju 123 bita, dok se preostalih 5 bitova koriste kao predložak koji sadržava informaciju o njihovim raspoređivanju (eng. scheduling).

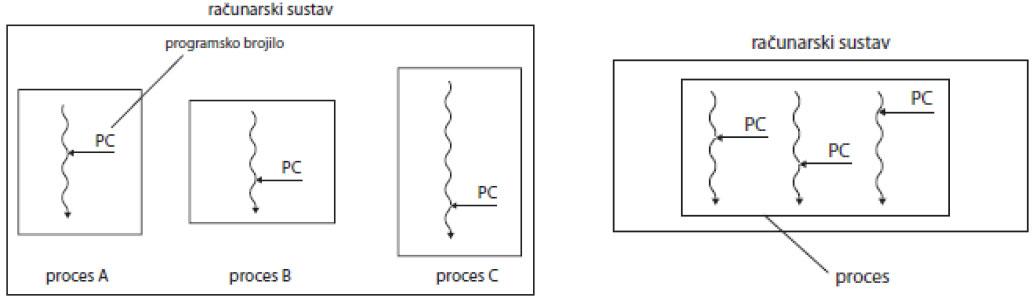
Predložak određuje tipove instrukcija u svežnju i određuje koje će se instrukcije izvesti paralelno.

* i1 || i2 || i3 - sve se tri instrukcije izvode paralelno
* i1 & i2 || i3 - prvo se izvodi instrukcija i1, a zatim se instrukcije i2i i3izvode paralelno
* i1 || i2 & i3 - prvo se instrukcije i1i i2izvode paralelno, a onda instrukcija i3
* i1 & i2 % i3 - sve se tri instrukcije u svežnju izvode slijedno.

1. **Paralelizam na razini dretvi i procesa**

Program koji se izvršava organiziran u jedan ili u više procesa. Proces se sastoji barem od jedne instrukcijske dretve. Svaki proces ima svoj virtualni procesor. U jednoprocesorskom računarskom sustavu, mora se stvarni procesor dijeliti između više procesa prijenosnom upravljanja s procesa na proces.

Samo se jedan proces izvodi u vremenu. Zbog brzog prospajanja s procesa na proces, svi procesi promatrani u vremenu napreduju.



Proces s više dretvi može se izvoditi u jednoprocesorskom računarskom sustavu tako da se procesor dodjeljuje naizmjence pojedinim dretvama.

U slučaju izvođenja više dretvi u jednoprocesorskom sustavu ne može se očekivati ubrzanje odvijanja procesa.

Potpuno iskorištenje paralelizma na razini dretvi postiže se u višeprocesorskom računarskom sustavu. Svaki procesor zadužen je za jednu ov diše raspoloživih dretvi u procesu. Višedretveni rad.

1. **SIMD računarski sustavi**

SIMD računarski sustavi iskorištavaju podatkovni paralelizam djelovanjem na vektorima ili dvo – i višedimenzionalnim poljima podataka.

Podatkovni paralelizam se u SIMD arhitekturi iskorištava zahvaljujući brojnim izvršnim jedinicama koje su sinkronizirane tako da djeluju paralelno i sve izvode istu operaciju koja je određena jednom instrukcijom

Osnovne zamisli SIMD nalazimo i u arhitekturi suvremenih procesora

Mnogi od njih imaju SIMD instrukcije koje su namijenjene poboljšanju performansi za multimedijske aplikacije

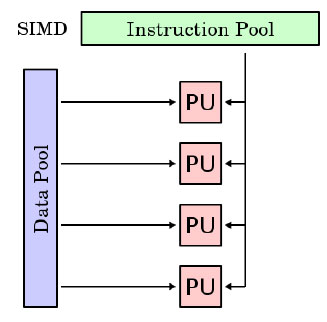
One omogućuju istodobno izvođenje jedne operacije na većem broju operanada u većem broju ALU

64-bitna ALU pretvara u dvije 32-bitne ALU ili u četiri 16-bitne ALU ili u osam 8-bitnih ALU koje djeluju istodobno

Kraći se operandi pojavljuju u multimedijskim aplikacijama

Podaci duljine 8 bita koriste za definiranje vrijednosti primarnih boja (R, G, B) slikovnih elemenata

16-bitni podaci koriste za prikaz vrijednosti zvučnog uzorka



1. **Karakteristike višeprocesorskih sustava**

* Sustav sadrži dva ili više sličnih procesora opće namjene i približno jednakih mogućnosti za obradu podataka.
* Svi procesori dijele zajedničku unutarnju memoriju .
* Svaki od procesora može imati i svoju memoriju u kojoj čuva određene podatke .
* Komunikacija između procesora se vrši preko dijeljene memorije .
* Neadekvatan odnos broja procesora i brzine dijeljene memorije može pogoršati performanse višeprocesorskog sustava.
* Svi procesori dijele isti skup U/I uređaja, bilo preko zajedničkih kanala bilo preko kanala koji su priključeni samo na pojedine procesore.
* Svi procesori u sustavu su pod kontrolom jednog istog operacijskog sustava koji je zadužen za raspoređivanje poslova, datoteka i kontrolu svih resursa.
* Višeprocesorski sustavi najčešće imaju centralnu upravljačku jedinicu.

1. **Grafički procesori - osnovna obilježja**

* Grafički su se procesori pretvorili u programibilne paralelne procesore.
* Grafički procesori i njima pridružene programske rutine ostvarene s OpenGL i DirectX definiraju različite modele grafičke obrade.
* S arhitektonskog gledišta, grafički procesor je visoko paralelni, višedretveni procesor s vrlo velikim brojem jezgri namijenjen vizualnom računanju (engl. visual computing).
* Definiran je i novi model programiranja.
* Velika procesna moć.
* Veliki stupanj paralelizma ostvaren vrlo velikim brojem procesora, odnosno jezgri.
* Podržavaju više programske jezike i programska okruženja opće namjene.
* Nudi djelotvornu primjenu grafičkih procesora na područjima izvan računalne grafike.