

Informatika 1

Neslužbena skripta



SADRŽAJ

1. POJAM INFORMACIJSKOG SUSTAVA	4
1.1. Definicija informacijskog sustava	4
1.2. Razlozi izgradnje IS-a	4
1.2.1. Poslovni sustav	4
1.2.2. Sustavski pristup	5
1.3. Pojmovi vezani uz informacijski sustav	5
1.3.1. Podatak	5
1.3.2. Algoritam	6
1.3.3. Informacija	6
1.4. Funkcije informacijskog sustava u poslovnom sustavu	6
1.5. Ciljevi informacijskog sustava	7
1.6. Elementi informacijskog sustava	8
1.6.1. Hardware	8
1.6.2. Software	8
1.6.3. Liveware	8
1.6.4. Orgware	9
1.6.5. Netware	10
1.6.6. Dataware	10
2. HARDWARE	12
2.1. Elementi računalnog sustava prema von Neumann-u	12
2.2. Sabirnice	13
2.3. Takt rada računala	13
2.4. Građa procesora	14
2.4.1. Aritmetičko-logička jedinica	14
2.4.2. Upravljačka jedinica	15
2.4.3. Registri	15
2.4.3.1. Akumulator	16
2.4.3.2. Buffer registri	16
2.4.3.3. Registri opće namjene	16
2.4.3.4. Programsko brojilo (PC)	16
2.4.3.5. Registar statusa	17
2.4.3.6. Indeks registar	17
2.4.3.7. Posmačni (shift) registri	17
2.4.3.8. Pokazivač adrese staka (SP)	18
3. DIGITALNO RAČUNALO	19
3.1. Bit, bajt i strojna riječ	19
3.2. Kodovi i kodiranje	20
3.2.1. Potreba za kodom sa dovoljnim brojem permutacija	20
3.2.2. Redundanca u kodu	21
3.2.3. BCD kód	21
3.2.4. ASCII i EBCDI kód	22
3.3. Osnovne računске operacije u brojevnim sustavima	23
3.4. Točke kodiranja	24
3.5. Zapisivanje brojeva u fiksnom zarezu	24
3.6. Zapisivanje brojeva u pokretnom zarezu	25
3.7. Provjera pariteta	25
3.8. Bistabili	26
3.9. Osnovni logički sklopovi	27
3.9.1. NE sklop	28
3.9.2. I sklop	28
3.9.3. ILI sklop	28
3.9.4. NI sklop	28
3.9.5. NILI sklop	28
3.9.6. EX-ILI sklop	28
3.10. Dekoderi	28
3.11. Brojila	28
3.12. Poluzbrajač (polusumator)	28
3.13. Potpuni zbrajač (potpuni sumator)	29
3.14. Registri	30
3.15. Izvođenje instrukcije	30
3.16. Organizacija procesora oko jedne sabirnice	31
3.17. Organizacija procesora oko dvije sabirnice	32
3.18. Organizacija procesora oko tri sabirnice	32
3.19. Tehnologije izrade tranzistora	32
3.19.1. Unipolarna tehnologija	33
3.19.2. Bipolarna tehnologija	33
3.20. Unutarnje memorije	34
3.20.1. Vrste unutarnjih memorija	34
3.20.1.1. ROM memorije	34
3.20.1.2. RAM memorije	35
3.20.2. Organizacija unutarnjih memorija	35
3.20.2.1. Dvodimenzionalna organizacija	36
3.20.2.2. Trodimenzionalna organizacija	36
3.20.2.3. Memorija organizirana u stack	37
3.20.3. Načini adresiranja unutarnjih memorija	38
3.20.3.1. Neposredno adresiranje	38
3.20.3.2. Direktno adresiranje	38

3.20.3.3. Indirektno adresiranje	38
3.20.3.4. Relativno adresiranje s obzirom na PC.....	39
3.20.3.5. Adresiranje registara i pomoću registara	39
3.20.3.6. Indeksirano adresiranje.....	41
3.20.3.7. Adresiranje po stranicama (straničenje).....	41
3.21. Vanjske memorije	42
3.21.1. Magnetske vanjske memorije	42
3.21.2. Optičke vanjske memorije.....	45
3.22. Virtualna memorija.....	46
3.23. Monitor	47
3.23.1. Princip rada monitora s katodnom cijevi (CRT).....	47
3.23.2. Princip rada monitora s tekućim kristalima (LCD).....	48
3.23.3. Rezolucija monitora.....	50
3.23.4. Pixel.....	50
3.23.5. Način iscrtavanja slike i frekvencija osvježavanja (CRT ekrani).....	50
3.24. Skener.....	51
3.24.1. Princip rada skenera.....	51
3.24.2. Vrste skenera	51
3.25. Pisač.....	52
3.25.1. Pisači s udarcem.....	52
3.25.1.1. Matrični pisači.....	52
3.25.2. Pisači bez udarca.....	53
3.25.2.1. Laserski pisači.....	53
3.25.2.2. Tintni pisači (ink-jet)	54
3.25.3. Rasterska točka.....	55
3.25.4. Font.....	55
3.25.4.1. Vrste fontova.....	55
3.25.4.2. RIP – pretvorba iz vektorskog formata u rasterski	56
3.26. Organizacija obrade podataka	57
3.26.1. Sekvencijalna (slijedna) obrada	57
3.26.2. Sekvencijalna (slijedna) obrada s prioritetima.....	57
3.26.3. Time-sharing obrada (obrada s podjelom vremena).....	57
3.26.4. Multiprogramska obrada (Multiprogramming, Multitasking).....	58
3.26.5. Multiprocesiranje (Multiprocessing)	58
3.26.6. Obrada u stvarnom vremenu (Real-time obrada).....	58
4. SOFTWARE	59
4.1. Sistemski software	59
4.2. Operacijski sustavi.....	59
4.2.1. Vrste operacijskih sustava	59
4.2.2. Funkcije operacijskih sustava	60
4.2.3. Obrada prekida i operacijski sustav	60
4.3. Programski jezici.....	61
4.3.1. Elementi programskih jezika.....	61
4.3.2. Generacije programskih jezika	62
4.3.3. Kompajleri i interpreteri	63
4.3.4. Blok dijagram i dijagram tijeka podataka	63
4.3.5. Metode programiranja	64
4.3.6. Osnovne programske strukture	64
4.4. DBMS	65
4.5. Uslužni (pomoćni) programi.....	65
5. STRUKTURE ZAPISA PODATAKA	66
5.1. Podatak, entitet, slog, datoteka	66
5.2. Materijalni nositelji podataka.....	66
5.2.1. Nositelji analognog zapisa	66
5.2.1.1. Mikrofilm.....	66
5.2.1.2. Papir i slika	67
5.2.2. Nositelji digitalnog zapisa	67
5.2.2.1. Magnetska vrpca.....	67
5.2.2.2. Magnetski disk.....	67
5.2.2.3. Optički disk	68
5.3. Organizacija zapisa podataka	69
5.4. Datotečni sustavi.....	69
5.4.1. FAT (File Allocation Table).....	70
5.4.2. NTFS (NT's Native File System).....	70
5.5. Defragmentacija	71
5.6. Datoteka.....	71
5.6.1. Slijedna ili sekvencijalna datoteka	72
5.6.2. Direktna ili relativna datoteka	72
5.6.3. Indeksna datoteka.....	72
5.6.4. Indeksno-sekvencijalna datoteka	73
5.7. Baze podataka	73
5.7.1. Relacijske baze	73
5.7.2. Skladišta podataka i dimenzijske baze	74

1. POJAM INFORMACIJSKOG SUSTAVA

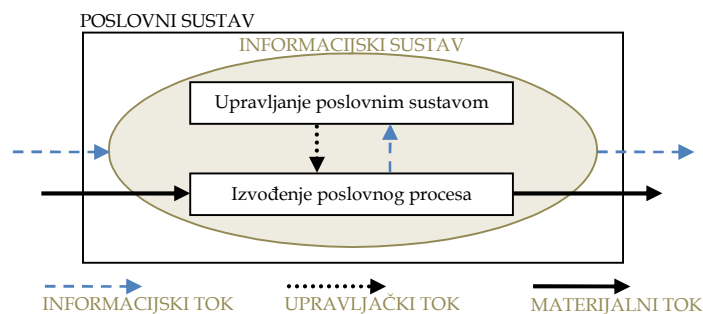
1.1. Definicija informacijskog sustava

- Efikasno obavljanje gotovo svih poslova, u bilo kojem poslovnom sustavu, popraćeno je informacijama i podacima. Nezavisno o vrsti poslovnih procesa koje podržava ili veličini organizacije u kojoj funkcionira, informacijski sustav (IS) se pojavljuje i primjenjuje kao ključni element poslovanja. Njegova je povećana uloga i važnost popraćena i rastućom primjenom informacijske tehnologije (IT).
- **Cilj je informacijskog sustava pribaviti informacije potrebne pri izvođenju poslovnog procesa i upravljanju poslovnim sustavom. Uobičajeni su dijelovi informacijskog sustava sustav za obradu transakcija, upravljački izvještajni sustav, sustav za potporu odlučivanju i sustav uredskog poslovanja.**
- Djelovanje IS-a upotpunjuje se primjenom IT-a i s njima povezanim programima, procedurama, uputama, algoritmima i znanjem kojima se IT pokreću zbog izvršenja poslovnih zadataka i ciljeva.
- Informacijski sustav (IS) je prema tome sprega i sustav materijalnih i nematerijalnih elemenata kojima se opisuje poslovna stvarnost, rješavaju poslovni zadaci i ispunjavaju poslovni ciljevi. Kao dio IS-a, čovjek/pojedinac formalizira poslovno okruženje u podatke, procedure, algoritme, informacije i znanja te usklađujući primjenu IT-a i programsku podršku, ispunjava poslovne funkcije i zadatke (dostavljanje i čuvanje podatka neophodnih za odlučivanje, održanje procesa te razvoj i neprekidnost poslovanja).
- Kao model poslovnog sustava, IS opisuje poslovni sustav, on je njegov informacijski model. Elementi poslovnog sustava (podaci, aktivnosti, funkcije, zadaci, izvršitelji, postupci itd), preslikavaju se na model informacijskog sustava gdje ih dijelimo na:
 - **model podataka** – definira podatke koji nastaju ili se koriste u poslovnom sustavu.
 - **model procesa** – definira procese ili funkcije koje se odvijaju u poslovnom sustavu.
 - **model izvršitelja** – definira one elemente koji obavljaju funkciju u poslovnom sustavu (software, hardware, liveware, orgware).

1.2. Razlozi izgradnje IS-a

1.2.1. Poslovni sustav

- Informacijski sustav djeluje unutar nekog poslovnog sustava, omogućavajući mu da komunicira unutar sebe i sa svojom okolinom. Slika prikazuje djelovanje informacijskog sustava unutar poslovnog sustava.



SLIKA 1.1 INFORMACIJSKI SUSTAV KAO DIO POSLOVNOG SUSTAVA

- U poslovni sustav ulaze i izlaze materijalni (materijal, sirovina, energija) i informacijski tokovi. Dvije su temeljne aktivnosti svakog poslovnog sustava:
 - **Izvršavanje poslovnih procesa**
 - Kad govorimo o poslovnom procesu, mislimo na osnovnu djelatnost promatranog poslovnog sustava odnosno na poslove koji se u njemu obavljaju.
 - U proizvodnom poduzeću poslovni proces sastoji se npr. od poslova proizvodnje, nabave potrebnih sirovina i energije, plasmana proizvedenih proizvoda itd.

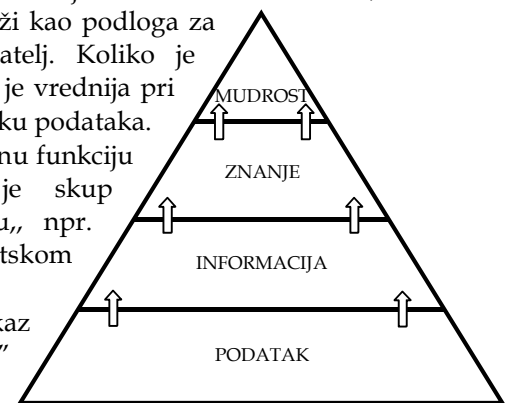
- U bankama poslovni proces obuhvaća poslove obavljanja finansijskih transakcija, kreditiranja, štednje itd.
- **Upravljanje poslovnim sustavom**
 - Svaki poslovni odnosno stvarni sustav (npr. poduzeće, ustanova, privreda, društvo) nastoji izgraditi svoj informacijski sustav koji će dati podlogu za brzo i kvalitetno odlučivanje, dakle za **transformaciju informacija u odluke**.
- Informacijska se tehnologija u mnogim poslovnim procesima koristi jer znatno podiže efikasnost odvijanja poslovnog procesa, npr. rezervacijski sustavi zrakoplovnih kompanija, poslovanje banaka...

1.2.2. Sustavski pristup

- Sustav je cjelina koja se ne može rastaviti na svoje dijelove a da se pri tome ne izgube njena osnovna svojstva.
- Sustavnim se pristupom **istražuju pojave i rješavaju problemi u njihovoj cjelokupnosti** i složenosti, obuhvaćajući sve bitne veze i odnose između dijelova sustava i između sustava i okoline. Pojave i probleme razmatraju s više gledišta stručnjaci različitih znanstvenih područja.
- Sustavni pristup ne označava spajanje samo dosad odvojenih predmeta znanstvenog proučavanja, nego i razdvojenih područja ljudskih aktivnosti. Sustav je sređena cjelina koju čine njegovi elementi povezani na određeni način, a njihovim međusobnim djelovanjem postižu se određeni ciljevi funkcioniranja sustava. Da bismo opisali neki sustav, potrebno je pobliže odrediti:
 - koji *elementi* pripadaju sustavu,
 - koje *veze* postoje između elemenata unutar sustava i između elemenata sustava i okoline i
 - kakvo je *ponašanje* (funkcioniranje) sustava
- Realni sustav sačinjen je od elemenata koji su dio konkretnog realnog objekta, a koji se opisuju kao sustav (mehanički, fizički, biološki, društveni).

1.3. Pojmovi vezani uz informacijski sustav

- Iako se često pojmovi *informacija* i *podatak* smatraju sinonimima, njihovo značenje se razlikuje: informacija odnosno obavijest proizlazi iz podataka – zapisa na nekom mediju.
- **Informacija** odnosno obavijest jest činjenica s određenim značenjem. Ona donosi novost, obavještava o nečemu, te otklanja neizvjesnost i općenito služi kao podloga za odlučivanje. Vrijednost informacije određuje sam primatelj. Koliko je upotrebljena informacija relevantnija, točnija i svježija, toliko je vrednija pri odlučivanju. Informaciju ćemo sačuvati ako je zapišemo u obliku podataka.
- **Podatak** je pojam kojim opisujemo i kvantificiramo elementarnu funkciju realnog sustava u određenom trenutku. Podatak je skup prepoznatljivih znakova zapisanih na određenom mediju,, npr. papiru (knjiga, notes), filmu (fotografija, filmski zapis), magnetskom mediju (datoteka na disku u računalu)...
- Informatički gledano, **elementarni podatak** simbolički je prikaz jednog obilježja promatranog objekta. Npr. podatak "Marko" prikazuje nam ime promatrane osobe.
- **Znanje** je uređen skup informacija nekog područja. Njime se opisuje stanje stvari, koje je ustanovljeno po nekom prihvaćenom kriteriju. Svako područje organizira u danom vremenskom trenutku svoje znanje utvrđivanjem prikladnih kategorija koncepata (elemenata znanja), njihovih međusobnih odnosa i ograničenja. Postoje različiti načini prikaza znanja. Najpoznatiji prikaz znanja je u obliku pravila (npr. u matematici aksioma, teorema itd.).



SLIKA 1.2: PODATAK, INFORMACIJA, ZNANJE I MUDROST

1.3.1. Podatak

- **Podatak je skup prepoznatljivih znakova zapisanih na određenom mediju.** Elementarni podatak je simbolički prikaz jednog obilježja promatranog objekta.

- Osim tradicionalnih strukturiranih podataka u obliku datoteka i baza podataka sve se više koriste i slabo strukturirani podaci kao što su različite vrste dokumenata te multimedijски podaci kao što su grafika, fotografije, zvuk i video.
- Danas se podaci intenzivno distribuiraju na korporacijskim mrežama i na internetu. Također se intenzivno razvijaju skladišta podataka u koja se pohranjuju podaci iz više različitih izvora podataka.

1.3.2. Algoritam

- **Algoritam je postupak ili pravilo za sustavno rješavanje određene vrste problema.** To je bilo koje dobro rješenje bilo kojeg problema. Sastoji se od opisa konačnog skupa koraka. Svaki od njih sadrži jednu ili više izjava, a svaka izjava jednu ili više operacija.
- Algoritmima su u prvo vrijeme nazivana samo pravila računanja s brojevima zapisanim u dekadnom sustavu, da bi se kasnije taj naziv počeo koristiti za pravila obavljanja raznovrsnih zadataka.
- S praktičnog gledišta, algoritmi su jasno definirani postupci za izvršavanje određenog problema. Algoritmi, raščlanjeni na uzastopne korake, prevode neki skup početnih (ulaznih) vrijednosti u skup završnih (izlaznih) vrijednosti. Početne i završne vrijednosti te eventualni međurezultati pohranjuju se u prikladne strukture podataka.
- Algoritmi se koriste za jasno utvrđivanje pravila dostizanja nekog postavljenog cilja. Uz svaki algoritam moraju biti jasno definirana početna stanja objekta na kojima se obavljaju operacije.
- Algoritam mora biti sastavljen od konačnog broja koraka koji utvrđuju slijed operacija koje treba obaviti nad objektima kako bi se dobila završna stanja objekata ili rezultati. Svaki korak opisuje se instrukcijom. Obavljanje algoritma naziva se algoritamskim procesom.
- Vrste algoritma:
 - **Specijalizirani** – mogu se primjeniti samo na pojedina početna stanja objekata.
 - **Općeniti** – mogu djelovati uz različita početna stanja objekata. Kod njih se definira klasa objekata i početnih stanja koja su dopuštena.

1.3.3. Informacija

- Poslovne informacije se dijele na dizajn proizvoda, tehničke podatke, upute menadžmenta, operativne baze podataka, operativne procese, tehničke baze podataka, znanje zaposlenih i računalni softver, a nalazimo i podjelu na poslovne ugovore, radne upute, poslovne rezultate i izvješća, baze podataka, sistemsku dokumentaciju, korisničke priručnike, operativne procedure, planove, aplikacijski i sistemski softver, razvojne i pomoćne alate itd.
- U poslovnim sustavima one predstavljaju ključni izvor poslovanja, jednu od najvećih poslovnih vrijednosti, osnovni izvor za stjecanje dohotka i pokretačku snagu za stvaranje nove vrijednosti. Informacija su bitne u svim oblicima poslovanja, a posebno su važne pri donošenju odluka, poboljšanju performansa sustava, postizanju tržišnog uspjeha i u podršci radnim procesima. Naglašena je i komunikacijska uloga informacije. U svakom slučaju, **informacije su nosioci poslovnih promjena**, instrument formaliziranja poslovne okoline i upravljanja okolišem.
- Općenito govoreći, informacije nastaju kad se niz uređenih i prepoznatljivih, ali značenjem nezanimljivih znakova kojima se predstavlja obilježje nekog objekta ili stvarnosti sredi i organizira u smislen oblik koji ima značenje, a za primatelja je novost. U poslovnim organizacijama poslovnom informacijom postaje onaj sadržaj koji je u funkciji ostvarenja poslovnih ciljeva. Poslovne informacije najčešće služe kao temelj odlučivanja, ali se koriste i za održavanje i podršku poslovnim procesima, međusobnu komunikaciju zaposlenika itd., a mogu biti i osobito zanimljive ako se moraju čuvati ili pak predstavljaju neposrednu financijsku vrijednost.

1.4. Funkcije informacijskog sustava u poslovnom sustavu

- Informacijski sustav djeluje unutar nekog poslovnog sustava omogućavajući mu da komunicira unutar sebe i sa okolinom.
- Cilj informacijskog sustava je opskrbiti poslovni sustav sa svim potrebnim informacijama, potrebnim pri izvođenju poslovnog procesa i upravljanju poslovnim sustavom.
- Mnogobrojni pozitivni rezultati nastaju primjenom IS-a. Između ostalog ističu se uloge IS-a kao: pokretača poslovanja, osnovnog sredstva privređivanja, sredstva za stjecanje prednosti i razlikovanje, sredstva za unapređenje poslovanja, osnovnog podsustava organizacije itd.

- IS-om čovjek/pojedinac formalizira poslovno okruženje u podatke, procedure, algoritme, informacije i znanja te usklađujući primjenu IT-a i programsku podršku, ispunjava poslovne funkcije i zadatke (dostavljanje i čuvanje podatka neophodnih za odlučivanje, održanje procesa te razvoj i neprekidnost poslovanja).
- Pritom se IS u velikoj mjeri oslanja na ICT (informacijsko-komunikacijsku tehnologiju) te njihovom primjenom obrađuje, prenosi, pohranjuje, dohvaća i objavljuje informacije i podatke kojima se opisuje tijek, stanja i procesi poslovnog sustava.

1.5. Ciljevi informacijskog sustava

- Informacijski sustav djeluje unutar nekog poslovnog sustava, omogućavajući mu da komunicira unutar sebe i sa svojom okolinom. Slika 1.1 prikazuje djelovanje informacijskog sustava unutar poslovnog sustava. U poslovni sustav ulaze i izlaze materijalni (materijal, sirovina, energija) i informacijski tokovi. Informacijski sustav preuzima informacije, obrađuje ih i prerađene prezentira poslovnom sustavu ili okolini. Informacijski je sustav, dakle, podsustav poslovnog sustava.
- Poslovni sustav dobiva informacije iz raznih izvora, vanjskih ili unutarnjih, a informacijski sustav ih obrađuje u nove i korisne informacije.
- Cilj je informacijskog sustava opskrbiti poslovni sustav svim njemu potrebnim informacijama, potrebnim pri:
 - **izvođenju poslovnog procesa**
 - **upravljanju poslovnim sustavom.**

IZVOĐENJE POSLOVNOG PROCESA:

- Kad govorimo o poslovnom procesu, mislimo na osnovnu djelatnost promatranog poslovnog sustava odnosno na poslove koji se u njemu obavljaju. U proizvodnom poduzeću poslovni proces sastoji se npr. od poslova proizvodnje, nabave potrebnih sirovina i energije, plasmana proizvedenih proizvoda itd. U bankama poslovni proces obuhvaća poslove obavljanja financijskih transakcija, kreditiranja, štednje itd.
- Informacijska se tehnologija u mnogim poslovnim procesima koristi jer znatno podiže efikasnost odvijanja poslovnog procesa. U nekim poslovnim procesima nemoguće je konkretno obavljati posao bez primjene informacijske tehnologije, npr. rezervacijski sustavi zrakoplovnih kompanija, poslovanje banaka itd. Upotrebom informacijske tehnologije može se izvođenje poslovnog procesa "automatizirati".
- **Automatizacija** poslovnog procesa jedan je od važnijih zadataka primjene informacijske tehnologije odnosno izgradnje informacijskog sustava temeljenog na informacijskoj tehnologiji.

UPRAVLJANJE POSLOVNIM SUSTAVOM I ODLUČIVANJE:

- Drugi zadatak informacijskog sustava jest osigurati informacije za efikasno upravljanje poslovnim sustavom. Informacija, bilo unutarnjeg ili vanjskog karaktera, predstavlja podlogu za donošenje određene poslovne odluke. Odluka je pak u funkciji ostvarivanja svrhe i cilja postojanja sustava. Pomoću odluka svjesno se upravlja sustavom (na Slici 1.1 to su upravljački tokovi).
- Prema tome, svaki poslovni, odnosno stvarni sustav (npr. poduzeće, ustanova, privreda, društvo) nastoji izgraditi svoj informacijski sustav koji će dati podlogu za brzo i kvalitetno odlučivanje, dakle za transformaciju informacija u odluke. Valja naglasiti da je odlučivanje proces koji ne traje samo u trenutku donošenja. Proces odlučivanja sastoji se od ovih koraka (Stallings i dr, 1988):
 - Prepoznavanje problema
 - Pronalaženje i ocjenjivanje opcija
 - Odabir opcije
 - Provedba
 - Ocjena
- Razmotrimo sada funkciju informacijskog sustava u pojedinim poslovnim područjima. Najvažnija poslovna područja primjene informacijskog sustava jednog poduzeća s primjerima primjene informacijske tehnologije su: proizvodnja, financije/računovodstvo, marketing, istraživanje i razvoj te ljudski resursi. Valja napomenuti da neki informacijski resursi, a to su

prvenstveno podaci, pripadaju čitavom poduzeću, a ne samo poslovnom području u kojem se obrađuju.

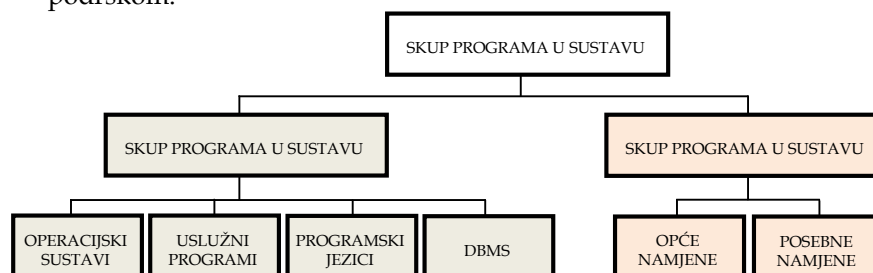
1.6. Elementi informacijskog sustava

1.6.1. Hardware

- Materijalna osnovica koju čine informacijske tehnologije, npr. računala, radne stanice, modemi, fizičke linije za komunikaciju itd.
- Hardver je najniža razina računalnog sustava. Čine je svi dijelovi od koje je načinjen računalni sustav:
 - svi mehanički dijelovi
 - magnetski, električni i elektronički sastavni dijelovi
 - naprave i uređaji (kućište, pogonski motor diska, magnetska vrpca, disk, poluvodička memorija, izvori električnog napajanja, integrirani sklopovi)
- Hardver je osnovica računalnog sustava u koju spadaju
 - **CPU** - sastavljena od elektroničkih komponenti i nema mehaničkih dijelova
 - **glavna memorija** kao osnovni uređaj za pohranu podataka koja se također sastoji od el. komponenti i ne sadrži mehaničke dijelove
 - **ulazno-izlazni uređaji** (tipkovnica, zaslon, pisači, veze, sabirnice...) i obično se sastoje od el. komponenti i različitih mehaničkih dijelova
 - **vanjska memorija** koja se također sastoji od el. dijelova i različitih mehaničkih komponenti
 - **komunikacijski uređaji**.

1.6.2. Software

- U software spada sva programska podrška koja se koristi pri radu tog sustava, to je skup svih programa koji se koriste u IS-u. Možemo ga definirati i kao onaj dio sustava koji postoji ali ne u fizičkom obliku nego u obliku informacija pohranjenih na računalu.
- Sistemski software čine programi koji promatrano zajedno predstavljaju skup programa koji služe za kontrolu i rad strojne podrške pri računalnoj obradi podataka.
- Software dijelimo na:
 - **Aplikativni** (korisnički programi) - skup korisničkih programa potrebnih za rješavanje raznih problemskih situacija koji proizlaze iz domene zadataka inf. Sustava.
 - **Sistemski** (sustavski programi) – skup strojno orijentiranih programa s funkcijom upravljanja i kontrole strojnog sustava u cilju sinhronizacije rada s aplikativnom podrškom.



SLIKA 1.3: PODJELA SOFTWARE-A

1.6.3. Liveware

- U liveware ubrajamo "živu" komponentu IS-a.
- Odnosi se na zaposlenike individualce, timove, njihovo znanje i sve osobe uključen u djelovanje IS-a
- Čovjek je osnovna komponenta IS-a jer kao njegov dio čovjek/pojedinac formalizira poslovno okružje u podatke, procedure, algoritme, informacije i znanja te usklađujući primjenu IT-a i programsku podršku, ispunjava poslovne funkcije i zadatke (dostavljanje i čuvanje podataka neophodnih za odlučivanje, održanje procesa te razvoj i neprekidnost poslovanja).
- Zaposlenici IS-a koji ulaze u liveware definiciju su npr. *korisnici* - od IS-a primaju informacije, *sistem analitičari* - veza između korisnika i IS-a, *programeri* - stvaraju programe, *program*

analitičari - spoj poslova programera i program-analitičarima u manjim tvrtkama, operatori na računalu, operatori unosa podataka, analitičari baze podataka, administratori, menadžeri...

1.6.4. Orgware

- Pod pojmom "orgware" se podrazumijeva organizacija tehničke opreme IS-a (hardware), programske opreme IS-a (software) i ljudi-izvršitelja poslova u IS-u u skladnu cjelinu.
- Definira koncepciju i organizaciju rada informacijskog sustava.
- Razlikujemo više oblika organizacije IS-a radi čim boljeg upravljanja istim:
 - **Odjel za informacijski sustav**
 - **Korisničko računalstvo**
 - **Informacijski centar**
 - **Unajmljivanje informacijskih usluga (Outsourcing)**

ODJEL ZA INFORMACIJSKI SUSTAV:

- Centralizirani oblik upravljanja IS-om.
- Na čelu je glavni informacijski menadžer sa zadacima:
 - Planiranje strategije IS-a
 - Efikasno vođenje odjela za IS
 - Briga o sigurnosti IS-a
 - Praćenje tehnoloških trendova



SLIKA 1.4: INFORMACIJSKI SUSTAV PODUZEĆA A

KORISNIČKO RAČUNALSTVO:

- Decentralizirani oblik upravljanja IS-om.
- Glavni razlog zbog kojeg se korisničko računalstvo razvilo je nemogućnost profesionalnih informatičara da odgovore na silne zahtjeve korisnika za novim programima.
- Stoga se potiče korisnika da sam pomoću neke već napravljene aplikacije konstruira program koji bi zadovoljio njegove specifične potrebe.

INFORMACIJSKI CENTAR:

- Informacijski centar (engl. Information Center), organizacijska je jedinica proizašla iz potrebe da se korisnicima na organiziran način pomogne u njihovim individualnim potrebama u vezi s obradama podataka.
- Najznačajniji zadaci informacijskog centra:
 - Izobrazba za informacijsku pismenost,
 - pomoć pri korištenju osobnih računala,

- izobrazba za korištenje pojedinih programa ili aplikacija te konzultiranje pri njihovom korištenju,
- održavanje služba pomoći korisnicima,
- ocjenjivanje, instaliranje i provjera novih programa ili aplikacija odnosno drugih proizvoda informacijske tehnologije,
- praćenje i poticanje upotrebe normi informacijske tehnologije,
- osiguranje sigurnosti sustava i podataka,
- izrada programa za povezivanje aplikacija,
- administriranje podacima,
- vođenje projekata uvođenja korisničkih aplikacija,
- osiguranje kvalitete korisničkih aplikacija,
- marketing programa ili aplikacija itd.

UNAJMLJIVANJE INFORMACIJSKIH USLUGA:

- Unajmljivanje informacijskih poslova izvan poduzeća počelo je početkom šezdesetih godina.
- Unajmljivanje ima pozitivne ali i negativne posljedice.
- Posljednjih godina mnoga poduzeća gledaju kako će informacijski servis iznajmiti izvana.
- Funkcije koje se mogu pribaviti izvan vlastitog poduzeća mogu biti vrlo jednostavne, npr. poslovi masovnog unosa podataka, ali i vrlo zahtjevne, kao što je razvoj i izgradnja kompletnog informacijskog sustava.
- Ima više razloga zašto je u posljednje vrijeme unajmljivanje informacijskih usluga u porastu:
 - osnovni razlog je u tome što se poduzeće želi u potpunosti posvetiti svom osnovnom poslu i u njemu biti vrhunsko
 - ponekad je razlog i nedostatak informacijskih stručnjaka

1.6.5. Netware

- Netware je mrežna komponenta IS-a, tj. komunikacijska infrastruktura bazirana na informacijskim tehnologijama. Jedan od primjera su računalne mreže.
- **Računalne mreže** su sustavi povezanih računala.
- U mrežnom okruženju računala razmjenjuju podatke, dijele vlastite izvore, omogućavaju komunikaciju, paralelni rad, kreiranje virtualne organizacije itd.
- Za ostvarenje računalne mreže potrebna je odgovarajuća softverska i hardverska podrška, a nazivamo je netware ili podrška za mrežu.
- Razlikuju se pojmovi:
 - arhitektura ili topologija mreže
 - organizacija mreže
 - mrežni protokoli
 - mrežna spojišta
 - LAN mreže
 - WAN mreže
 - prijenosni mediji koji su opisani kroz druge lekcije pa se ovdje neće detaljnije pojašnjavati

1.6.6. Dataware

- Dataware su svi sadržaji u informacijskom sustavu kojima se opisuju činjenice iz realnog svijeta i poslovnog sustava na koji se odnose a organizirani i oblikovani tako da budu razumljivi i da se mogu koristiti u poslovanju za donošenje odluka i ostvarivanje ciljeva i zadataka.
- Osnovno, dataware IS-a sadržan je u podacima, informacijama i znanju:
 - podatak je skup prepoznatljivih znakova zapisanih ne nekom mediju; podatak je simbolički prikaz jednog obilježja promatranog objekta.
 - informacije nastaju kad se niz uređenih i prepoznatljivih, ali značenjem nezanimljivih znakova kojima se predstavlja obilježje nekog objekta ili stvarnosti sredi i organizira u smislen oblik koji ima značenje a za primatelja je novost.
 - Za povezivanje činjenica iz stvarnog svijeta u podatke iz kojih nastaju informacije potrebno je znanje. **Znanje** je svijest i razumijevanje informacija i mogućnost njihova povezivanja u smislu podrške i obavljanja zadataka.

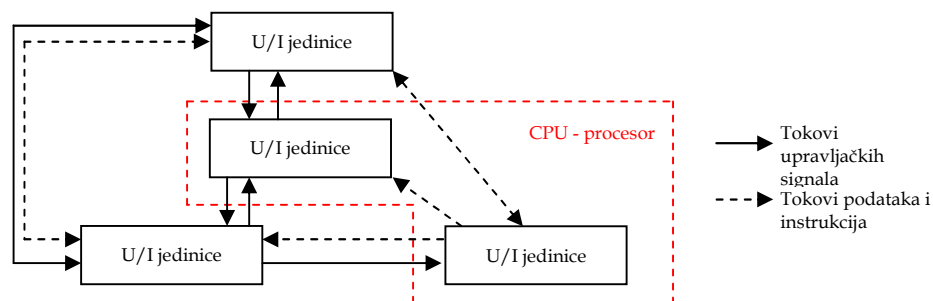
- U dataware IS-a ubrajaju se dizajn proizvoda, tehničke podaci, upute menadžmenta, operativne baze podataka, operativni procesi, tehničke baze podataka, znanje zaposlenih i računalni softver, poslovni ugovori, radne upute, poslovni rezultati i izvješća, baze podataka, sistemska dokumentacija, korisnički priručnici, operativne procedure, planovi, aplikacijski i sistemski softver, razvojni i pomoćni alati itd.

2. HARDWARE

- Hardware je materijalna osnovica informacijskog sustava koju čine informacijske tehnologije, npr. računala, radne stanice, modemi, fizičke linije za komunikaciju itd.
- Hardware je najniža razina računalnog sustava. Čine ga svi dijelovi od koje je načinjen računalni sustav:
 - svi mehanički dijelovi
 - magnetski, električni i elektronički sastavni dijelovi
 - naprave i uređaji (kućište, pogonski motor diska, magnetska vrpca, disk, poluvodička memorija, izvori električnog napajanja, integrirani sklopovi)
- Hardware je osnovica računalnog sustava u koju spadaju:
 - **CPU** - sastavljena od elektroničkih komponenti i nema mehaničkih dijelova
 - **glavna memorija** kao osnovni uređaj za pohranu podataka koja se također sastoji od el. komponenti i ne sadrži mehaničke dijelove
 - **ulazno-izlazni uređaji** (tipkovnica, zaslon, pisari, veze, sabirnice...) i obično se sastoje od el. komponenti i različitih mehaničkih dijelova
 - **vanjska memorija** koja se također sastoji od el. dijelova i različitih mehaničkih komponenti
 - **komunikacijski uređaji**.

2.1. Elementi računalnog sustava prema von Neumann-u

- Konceptija modela računala po von Neumann-u prevladava u četiri generacije računala.
- Model su 1946. godine opisali von Neumann, Burks i Goldstine, a sastoji se od 4 osnovne jedinice:
 - **Memorijske jedinice (memorija)**
 - **Upravljačke jedinice**
 - **Aritmetičko-logičke jedinice**
 - **Ulazno-izlazne jedinice**
- Jedinice prikazane u modelu su povezane podatkovnim i upravljačkim tokovima, koji su u računalima fizički realizirani u obliku sabirnica (bus).



SLIKA 2.1: VON NEUMANNOV MODEL RAČUNALA

MEMORIJA:

- Instrukcije i podaci su svedeni na numerički kod i pohranjuju se na isti način i u istom obliku u memorijsku jedinicu.
- Memorija nema obradbenih sposobnosti, a obavlja samo dvije temeljne operacije:
 - čitanje iz memorije
 - zapisivanje u memoriju.

UPRAVLJAČKA JEDINICA:

- Jedinica koja "razumije" numerički kod kojim su predstavljeni podaci i instrukcije.
- Njen zadatak je da tumači numeričke kodove i u skladu s njima generira upravljačke signale kojima upravlja izvođenjem instrukcija i koordinira radom rač. sustava.
- Zajedno sa aritmetičko-logičkom jedinicom čini cjelinu koja se naziva centralna procesorska jedinica (CPU ili procesor).

ARITMETIČKO-LOGIČKA JEDINICA:

- Jedinica u računalu koja posjeduje obradbene sposobnosti se naziva aritmetičko-logička jedinica

- Zajedno sa upravljačkom jedinicom čini cjelinu koja se naziva centralna procesorska jedinica (CPU ili procesor).

ULAZNO-IZLAZNE JEDINICE:

- Zadužene su za komunikaciju računala sa korisnikom i okolinom računalnog sustava.
- Omogućavaju unos podataka potrebnih za obradu u računalo i prikaz rezultata računalne obrade.
- Omogućavaju i prijenos podataka između računalnih sustava.

2.2. Sabirnice

- **Sabirnice su skup spojnih vodova koji povezuju sve elemente računalnog sustava u funkcionalnu cjelinu.**
- One omogućavaju komunikaciju između različitih dijelova računalnog sustava.
- Sabirnice najčešće dijelimo po dva kriterija – po smještaju i po vrsti sadržaja koji se njima prenosi.
- Podjela sabirnica *po smještaju*:
 - **Unutarnje sabirnice** – nalaze se unutar mikroprocesora i povezuju dijelove mikroprocesora u funkcionalnu cjelinu.
 - **Vanjske sabirnice** – povezuju mikroprocesor sa ostalim dijelovima računalnog sustava.
- Podjela sabirnica *po vrsti sadržaja koji se njima prenosi*:
 - **Podatkovne sabirnice** – povezuju procesor sa memorijskim i U/I sklopovima, dvosmjerne su i služe za prijenos podataka:
 - u procesor – dohvaćaju se instrukcije i operandi potrebni za izvođenje programa.
 - iz procesora – prema memoriji i/ili U/I sklopovima se radi pohrane ili prikaza šalju rezultati obrade.
 - **Adresne sabirnice** – povezuju procesor sa memorijskim i U/I sklopovima, jednosmjerne su (od procesora prema van) i služe za adresiranje memorijske ili U/I lokacije na kojoj se nalazi podatak kojeg treba prenijeti podatkovnom sabirnicom.
 - **Upravljačke sabirnice** – povezuju upravljačku jedinicu mikroprocesora sa svim ostalim dijelovima računalnog sustava i služe za prijenos upravljačkih signala kojima upravljačka jedinica upravlja i koordinira radom računala.

ŠIRINA SABIRNICE

- Pojam koji označava broj paralelnih vodova koji čine određenu sabirnicu (označava se brojem vodova ili **brojem bitova koji se mogu prenijeti u jednom ciklusu**).
- Ovisi o vrsti sabirnice i o konstrukcijskim značajkama procesora i računalnog sustava u cjelini.
- **Širina adresne sabirnice** ovisi o veličini adresnog polja mikroprocesora, tj. o količini memorije koju procesor može adresirati – ukoliko je adresno polje veličine 2^N , tada je potrebna i adresna sabirnica koja može adresirati 2^N lokacija – tj. širina takve sabirnice mora iznositi N vodova.
 - Tipične širine – 8-bitne, 16-bitne, 32-bitne, 64-bitne...
- **Širina podatkovne sabirnice** ovisi najviše o **veličini strojne riječi** kojom računalo barata, tako da je širina podatkovne sabirnice jednaka nekom višekratniku veličine strojne riječi – tj. obično je N puta šira od same strojne riječi.
 - N veći od 1 se obično uzima zato da bi se jednim ciklusom moglo prenijeti u/iz procesora više strojnih riječi istovremeno.
 - tipične širine – 8-bitne, 16-bitne, 32-bitne, 64-bitne, 128-bitne, 256-bitne...
- **Širina upravljačke sabirnice** ovisi isključivo o broju kontrolnih vodova potrebnih za upravljanje računalom.

2.3. Takt rada računala

- Da bi se rad računala mogao odvijati sinkronizirano, sve promjene stanja u njemu moraju se zbivati u točno određenim diskretnim vremenskim momentima, određenim taktnim signalima.
- Ti signali se generiraju u **generatoru taktnih signala (clock-glavni sat)** i odašilju se u generator upravljačkih signala koji je dio upravljačke jedinice, a ona opet dio CPU-a.

- Proizvode se u generatoru taktnih signala pomoću posebnog kristalnog oscilatora, da bi se dobila stabilna frekvencija.
- Takt rada računala se izražava frekvencijom kojom se izmjenjuju taktni signali.
- Oni daju "ritam" pod kojom radi procesor, ali i cijelo računalo.
- Današnja računala imaju frekvenciju rada koja se mjeri u GHz.
- Bilo koja akcija u računalu može se napraviti samo u trenutku koji je određen generatorom taktnih signala. Na taj način različite jedinice izvode svoje akcije u točno određenim trenucima u fiksno određenim vremenskim intervalima.
- Na taj način se usklađuje rad i djelovanje svih elemenata računalnog sustava pri obradi podataka

2.4. Građa procesora

- Procesor (CPU – Central Processing Unit) je osnovni dio računalnog sustava.
- On regulira, koordinira i upravlja aktivnostima svih jedinica sustava.
 - Obavlja aritmetičku i logičku obradu podataka.
 - CPU je "mozak" računala – brine se za sukcesivno pritjecanje podataka i instrukcija, te njihovo identificiranje i povezivanje s odgovarajućom programskom logikom u cilju obrade, memoriranja ili ispisivanja preko izlaznih jedinica računalnog sustava.
 - CPU ima dvije osnovne funkcije:
 - **obrada podataka, te**
 - **nadzor i usklađivanje djelovanja cjelokupnog računalnog sustava.**
- Procesor je izgrađen od sljedećih osnovnih dijelova:
 - **aritmetičko-logičke jedinice,**
 - **upravljačke jedinice, te**
 - **registara opće i posebne namjene.**



SLIKA 2.2: PROCESOR

2.4.1. Aritmetičko-logička jedinica

- Neumann, Burks i Goldstine su 1946 god. opisali koncepte na kojima se temelji većina današnjih računalnih sustava.
- Računalni sustav se sastoji od četiri osnovne funkcijske jedinice, a to su:
 - memorija
 - upravljačka jedinica
 - aritmetičko-logička jedinica
 - ulazno-izlazna jedinica.
- Memorija nema obradbenih sposobnosti te može obavljati samo dvije osnovne operacije:
 - čitanje i pisanje.
- Upravljačka jedinica dohvaća instrukcije iz memorije, tumači instrukcije te generira upravljačke signale.
- Zbog toga što memorija nema obradbenih funkcija, tj. sposobnosti, u sustavu mora postojati posebna jedinica koja služi za obradu podataka – to je ARITMETIČKO-LOGIČKA JEDINICA.

OPERACIJE ARITMETIČKO-LOGIČKE JEDINICE:

- Zajedno sa upravljačkom jedinicom tvori cjelinu, tj. centralnu procesnu jedinicu (CPU).
- Aritmetičko-logička jedinica obavlja aritmetičke i logičke operacije.
- Aritmetičke operacije:
 - zbrajanje, oduzimanje, množenje, dijeljenje, posmak (shift), rotacija
- Logičke operacije:
 - logičko I, logičko ILI, logičko NE
- Do aritmetičkih operacija se ne dolazi putem aritmetičkih već preko logičkih operacija zato što elementarni digitalni sklopovi izvršavaju logičke operacije.

GRAĐA ARITMETIČKO-LOGIČKE JEDINICE:

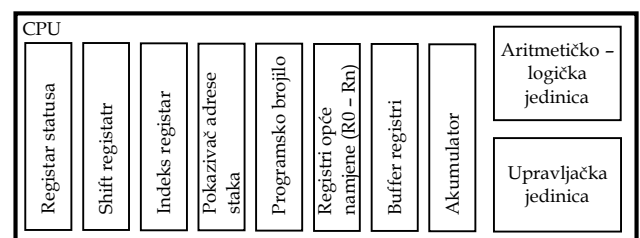
- Sastoji se od :
 - **Sklopova i**
 - **Registara**
- Sklopovi obavljaju aritmetičke i logičke operacije, a registri sudjeluju u operacijama tako što privremeno pohranjuju podatke i rezultate.

2.4.2. Upravljačka jedinica

- Upravljačka jedinica je najsloženiji dio središnjeg procesora, koji ima zadatak kontrolirati i koordinirati rad svih ostalih dijelova računalnog sustava.
- Ona određuje kad treba nešto učiniti, memorirati, izračunati, te donijeti logičku odluku u smislu što je veće, što je manje, što je ekvivalentno s nečim, što je numeričko, što je pozitivno, a što je negativno itd.
- Upravljačka jedinica obavlja svoj zadatak na osnovi instrukcija koje preuzima u slijedu iz glavne memorije te ih interpretira, i nakon toga inicira određenu akciju generiranjem slijeda **upravljačkih signala**.
- Upravljačka jedinica svaku instrukciju izvršava kroz dvije faze, i to:
 - fazu pripreme (pribavljanja) i
 - fazu izvršenja.
- Njezine najvažnije funkcije su:
 - upravljanje čitanjem odnosno pisanjem podataka u glavnoj memoriji
 - upravljanje transferom informacija između glavne memorije i aritmetičke jedinice
 - upravljanje radom aritmetičke jedinice i
 - koordinacija rada svih ostalih dijelova računalnog sustva.
- Upravljačka jedinica sadrži razne sklopovske elemente kao što su:
 - registar instrukcija,
 - programsko brojilo,
 - indeksni registar,
 - glavni sat,
 - generator upravljačkih signala,
 - vremenski krugovi,
 - vremenski programator...

2.4.3. Registri

- **Registar je dio memorije koji služi za privremenu pohranu podataka.**
- Izrađuju se od elemenata statičke memorije (bistabila i tranzistora) – zbog brzine rada.
- Najčešće su smješteni u procesoru.
- Svi podaci moraju biti smješteni u odgovarajuće registre prije nego što mogu biti obrađeni.
- Ako se dva broja trebaju pomnožiti, oba broja moraju biti u registrima, a i rezultat je također smješten u registre.
- Registar može sadržavati adresu memorijske lokacije gdje su podaci spremljeni, a ne nužno i same podatke.
- Broj registara koje procesor ima i veličina svakog (broj bitova) pomažu da se odredi snaga i brzina procesora.
- Npr. 32-bitni procesor je onaj u kojem je svaki registar širok 32 bita – stoga svaka procesorska instrukcija može manipulirati s 32 bita podataka.
- Vrste registara su sljedeće:
 - **akumulator,**
 - **buffer registri,**
 - **registri opće namjene,**
 - **programsko brojilo,**
 - **pokazivač adrese staka,**
 - **indeks registar,**
 - **posmačni (shift) registar, te**
 - **registar statusa.**
- S obzirom na funkciju koju obavljaju, postoje:
 - opći registri,
 - memorijski registri,
 - osnovni registri,
 - registri za operacije s kliznim zarezom,
 - registri instrukcija,
 - registri stanja.



SLIKA 2.3: SHEMA PROCESORA

2.4.3.1. Akumulator

- Akumulator je registar u kojem se dobivaju (akumuliraju) rezultati različitih operacija s binarnim brojevima koje je izvršila aritmetičko-logička jedinica.
- Sudjeluje u gotovo svim operacijama s podacima, a u njemu se čuvaju i parcijalni rezultati različitih operacija.
- U njemu se akumuliraju (prikupljaju, nakupljaju) različiti rezultati za vrijeme izvođenja programa.
- Podaci do akumulatora dolaze putem podatkovne sabirnice.
- S akumulatora se podaci obično prenose na buffer-registre, a sa njih na ulaze aritmetičko-logičke jedinice.

2.4.3.2. Buffer registri

- Buffer registri služe za privremenu pohranu podataka.
- Smješteni su na ulazima aritmetičko-logičke jedinice (na svakom ulazu po jedan), gdje privremeno čuvaju podatke (operande) sve dok se ne prikupe svi potrebni podaci za daljnju obradu u ALU.
- Još jedna česta primjena buffer registara je kod usklađivanja rada sklopova koji rade različitim brzinama.
- Sporiji sklopovi ne mogu prihvaćati podatke tempom kojim ih dostavljaju brži sklopovi, pa se između brzih i sporijih sklopova postavljaju buffer registri koji čuvaju podatke toliko dugo dok sporiji sklopovi ne budu spremni za prihvatanje podataka.

2.4.3.3. Registri opće namjene

- Prisutni su kod svih vrsta procesora. Neke vrste procesora imaju svega nekoliko takvih registara, dok ih druge vrste imaju i po nekoliko desetaka.
- Obično ih označavamo sa R_i gdje je $0 \leq i < N$ (N je ukupni broj registara opće namjene).
- S obzirom da su u pitanju registri opće namjene, u njima se mogu upisivati najrazličitiji podaci.
- Budući da su smješteni neposredno u procesoru, pristup do njih i njihovog sadržaja je vrlo brz i jednostavan (brži i jednostavniji nego pristup do podataka smještenih u radnoj memoriji).
- Što češćom uporabom registara opće namjene tijekom obrade (a ne radne memorije) se znatno ubrzava rad procesora.
- Duljina im ovisi o duljini strojne riječi koju upotrebljava računalo. Npr. kod 32-bitnih računala su i registri opće namjene 32-bitni.

2.4.3.4. Programsko brojilo (PC)

- PC je jedan od najvažnijih registara u procesoru i bez njega nije moguć rad računala.
- PC ima ključnu ulogu u procesu izvođenja instrukcija.
- U njemu je pohranjena *adresa lokacije u memoriji koja sadrži:*
 - **sljedeću instrukciju koju treba izvesti, ili**
 - **sljedeći operand koji je potreban za izvođenje trenutne instrukcije.**
- Adresa pohranjena u PC se preko vanjske adresne sabirnice prenosi u memoriju ROM ili RAM – adresira se određena memorijska lokacija i njen sadržaj se podatkovnom sabirnicom prenese u procesor.
- Nakon pribavljanja sadržaja sa zadane memorijske lokacije, sadržaj PC-a se uveća za 1 i pokazuje na sljedeću memorijsku lokaciju.
- "Uvećavanje za 1" treba shvatiti relativno – taj "1" može biti jedan bajt, dva bajta ili više – veličina te "jedinice" ovisi o duljini strojne riječi računala.
- Ako je strojna riječ dugačka 4 bajta, tada "uvećavanje za 1" znači povećavanje sadržaja PC-a za 4 bajta.
- "Uvećanjem za 1" dobivamo da PC uvijek pokazuje na iduću strojnu riječ.
- Iznimno se sadržaj PC-a ne mora povećati za 1, već se u PC može upisati i proizvoljna adresa koja pokazuje na bilo koju potrebnu memorijsku lokaciju – ta se osobina koristi kod izvršavanja strojnih instrukcija za skok na određenu memorijsku lokaciju.
- PC je realiziran kao brojilo – digitalni sklop koji je složenije građe od običnog registra, jer mora omogućavati inkrementiranje svog sadržaja (za iznos duljine strojne riječi).
- Osim inkrementiranja, mora omogućiti i izravan upis potrebne adrese.

2.4.3.5. Registar statusa

- Registar statusa čini cjelinu s aritmetičko-logičkom jedinicom.
- Zadatak mu je da **signalizira (ne)postojanje određenog stanja prilikom obrade podataka u ALU**.
- Registar statusa se sastoji od niza međusobno **nepovezanih** bistabila.
- Svaki bistabil predstavlja jedan bit koji ima ulogu *zastavice* kojom se signalizira određeno stanje
- Vrste zastavica se u detaljima razlikuju od procesora do procesora, ali slijedeće zastavice su tipične za većinu registara statusa:
 - **C (Carry) – prijenos**
 - Koristi se kod operacija sa šift-registrom (rotiranje i pomicanje).
 - Koristi se i kod zbrajanja – mogući prijenos "1 dalje" koji nastaje zbrajanjem dviju binarnih jedinica se signalizira postavljanjem ove zastavice u stanje "1".
 - **O (Overflow) – preljev**
 - Postavlja se u stanje "1" kad se kod aritmetičkih operacija prekorači opseg zapisivanja brojeva u nekom registru.
 - Npr. 8-bitni registar može pohraniti brojeve od 0 do 255. Ukoliko je rezultat neke operacije broj 256 ili više, tada se on ne može zapisati u 8-bitni registar, pa se takvo stanje signalizira postavljanjem zastavice "O" na vrijednost "1".
 - **N (Negative) – ili S (Sign)**
 - Stanje ove zastavice je povezano sa vrijednošću krajnjeg lijevog bita (bita predznaka broja) u aritmetici dvojnog komplementa – ako je zastavica u stanju "1" tada je rezultat negativan broj.
 - **H (Half-carry) – poluprijenos**
 - Koristi se za signalizaciju prijenosa "1 dalje" u BCD aritmetici.
 - Npr. ako radimo sa 8-bitnim riječima, tada u jednu takvu riječ stanu 2 BCD-kodirane dekadске znamenke (bitovi 0 do 3 predstavljaju desnu znamenku, a bitovi 4 do 7 predstavljaju lijevu znamenku).
 - Do poluprijenosa dolazi kad se "1 dalje" prenosi iz desne u lijevu BCD znamenku, tj. kad "1 dalje" nastane između 3. i 4. Bita.
 - U takvoj situaciji se zastavica "H" postavlja u stanje "1"
 - **Z (Zero) – Nula**
 - Postavlja se u stanje "1" kad god je rezultat neke operacije jednak nuli.
 - **P (Parity) – paritet**
 - Postavlja se u stanje "1" uvijek kad nastane greška u provjeri pariteta kod prijenosa podataka.
 - **ION (Interrupt On) – omogućavanje prekida**
 - Postavljanjem u stanje "1" se signalizira da je omogućen prekid programa.
 - Stanjem "0" se signalizira zabrana prekidanja programa

2.4.3.6. Indeks registar

- Indeks registar je registar specifične namjene, a koristi se najčešće za realizaciju indeksiranog adresiranja memorije.
- Kod takvog načina adresiranja memorije, adresa memorijske lokacije se dobije zbrajanjem sadržaja indeks registra sa nekom vrijednošću (pomakom – offset).

2.4.3.7. Posmačni (shift) registri

- Posmačni ili SHIFT registri su takvi registri kod kojih se **izlaz iz jednog bistabila prenosi na ulaz drugog**, pod uvjetom da ti bistabili čine jedan registar.
- Pri tome sljedeći bistabil u nizu preuzima stanje prethodnog.
- Smješteni su u CPU.
- Pomoću njih se mogu izraditi stack memorije, a služe i pri obavljanju nekih računskih operacija.
- Šifter (shift registar) služi za **pomicanje binarnih podataka ulijevo ili udesno**.
- Pri tome se mogu obavljati dvije različite operacije:
 - **rotiranje** (ulijevo i udesno), i
 - **pomicanje** (ulijevo i udesno)

ROTIRANJE ULIJEVO:

- Šifter i bit C (carry) registra statusa čine cjelinu.
- Pri rotiranju ulijevo svaki se bit u šifteru pomakne za jedno mjesto ulijevo, tj. na susjedno mjesto veće težinske vrijednosti.
- Bit najveće težine iz šiftera prelazi u bit C registra statusa, a njegov sadržaj prelazi u bit najmanje težine šiftera

ROTIRANJE UDESNO:

- Analogno rotiranju ulijevo, samo se svaki bit u šifteru pomiče za jedno mjesto udesno, tj. na susjedno mjesto manje težinske vrijednosti.
- Bit najmanje težine iz šiftera prelazi u bit C registra statusa, a njegov sadržaj prelazi u bit najmanje težine šiftera.

POMICANJE ULIJEVO:

- Bit najveće težine iz šiftera prelazi u bit C registra statusa, a na mjesto najmanje težine u šifteru dolazi **nula**.
- Jedan pomak sadržaja šiftera ulijevo je ekvivalentan množenju sadržaja šiftera sa 2.

POMICANJE UDESNO:

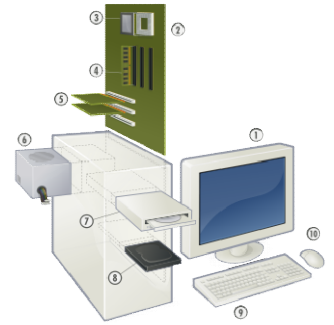
- Bit najmanje težine iz šiftera prelazi u bit C registra statusa, a na mjesto najveće težine u šifteru dolazi nula.
- Jedan pomak sadržaja šiftera udesno je ekvivalentan dijeljenju sadržaja šiftera sa 2.

2.4.3.8. Pokazivač adrese staka (SP)

- Pokazivač adresa staka (Stack Pointer – SP) je potreban kod svih računala koja koriste memoriju organiziranu u stak (stog).
- To je posebna vrsta organizacije memorije, koja je posebno korisna kod obrade prekida programa.
- Stak memorija se adresira pomoću SP registra.
- Dakle, sadržaj SP registra se, slično kao i PC registra, može priključiti na adresnu sabirnicu u svrhu adresiranja stak memorije.

3. DIGITALNO RAČUNALO

- Digitalna računala grade se od digitalnih elektroničkih elemenata koji izražavaju određene logičke funkcije. To su logički sklopovi I, ILI, NI, NILI itd., zatim bistabili, registri, brojlara, memorije, sklopovi za izvođenje aritmetičko logičkih operacija.
- Ti se sklopovi na određeni način povezuju i tako se grade digitalna računala. Način na koji se od sklopovskih komponenti sagradi digitalno računalo nazivamo **arhitekturom računala**.
- Jedan od osnovnih pokazatelja veličine i snage digitalnog računala jest veličina upotrijebljene riječi, tj. broj bitova koji se u jednom zahvatu zapisuje u memoriju i registre te s kojima se odjednom izvode različite operacije.
- Danas već zastarjela podjela računala po veličini strojne riječi navodi i mikroročunala kao jednu od kategorija:
 - **Mikroročunala**
 - Mikroročunala upotrebljavaju najčešće riječi od 4, 8, 16 bitova.
 - To su relativno kratke riječi, pa se mikroročunala svrstavaju među najmanja računala u pogledu snage za obradu podataka.
 - **Miniračunala**
 - Miniračunala su nešto veća i moćnija od mikroročunala, barem u principu, iako su moguće i iznimke.
 - Kod njih je duljina upotrijebljene riječi najčešće oko 16 bitova, ali ima miniračunala i od 12 i od 32 bita.
 - **Srednja i velika računala**
 - Srednja i velika računala obično upotrebljavaju duljinu riječi od 32 do 64 bita. Takve duljine riječi daju računalu znatno veću snagu.
 - Npr. 32 bitno računalo može u jednom zahvatu prenijeti, pohraniti ili izvršiti obradu nekog 32 bitnog podatka, dok bi 16 bitnom računalu za isti posao trebala 2 zahvata, a 8 bitnom 4 zahvata.

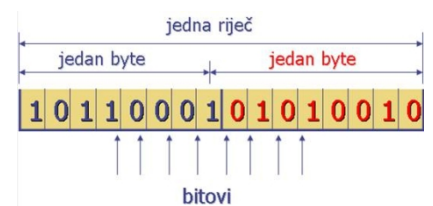


SLIKA 3.1: DIGITALNO RAČUNALO

- Veličina strojne riječi u današnje vrijeme više nije kriterij prema kojem se neko računalo može proglasiti mikroročunalom, a neko drugo računalom npr. velikim računalom. Nemoguće je pronaći jedinstveni kriterij za takvu vrstu podjele, pa je stoga ispravnije navesti neka od obilježja koja karakteriziraju suvremeno shvaćanje pojma 'mikroročunalo':
 - jezgru mikroročunala čini centralna procesna jedinica – CPU tj. mikroprocesor.
 - suvremeni mikroprocesor osim osnovnih obradbenih sklopova u sebi sadrži i priručnu (cache) memoriju za kratkotrajnu pohranu podataka, te osnovne ulazno/izlazne kontrolne sklopove.
 - primjena mikroročunala seže od kućne i osobne (personal computer – PC), pa sve do (barem djelomično) primjene u znanstvene i inženjerske svrhe (workstations – radne stanice).
 - mikroročunala nisu namijenjena opsluživanju kritičnih poslovnih sustava, kod kojih je bitna iznimna pouzdanost rada i dostupnost računalne opreme – takve poslove obavljaju specijalizirani poslužitelji (serveri) i velika (mainframe) računala, koje odlikuje daleko veći kapacitet radne i vanjske memorije, mnogo veća brzina u izvođenju ulazno/izlaznih operacija, veća kvaliteta izrade samog računala, te mogućnost zamjene većine neispravnih dijelova računala bez prekida u radu istog.

3.1. Bit, bajt i strojna riječ

- **Bit**
 - jedno binarno mjesto čini jedan bit binarnog podatka.
 - bit predstavlja najmanju jedinicu za zapis binarnih podataka u računalu.
- **Riječ**
 - više bitova međusobno se združuju u veći binarni broj te čine riječ - riječ označava cjelovit podatak i ona je standardna za određeno računalo.



SLIKA 3.2: BIT, BAJT I STROJNA RIJEČ

- duljinu riječi odabiru konstruktori pri koncipiranju računalnog sustava (npr. riječi od 8 bita, 12 bita, 16 bita, 32 bita...).
- svi podaci u računalu se zapisuju u obliku riječi.
- **Bajt**
 - riječi se sastoje od većeg broja bitova, pa ih je moguće dijeliti na manje cjeline (veće od jednog bita) koje se nazivaju bajtovi.
 - bajt se obično sastoji od 8 bitova, pa se tako 16-bitna riječ sastoji od 2 bajta, 24-bitna riječ sastoji od 3 bajta itd.
 - bajt se najčešće sastoji od 8 bitova (iako to nije pravilo), a prema tome gore prikazana riječ sastoji se od 2 bajta).

3.2. Kodovi i kodiranje

- Kako računo može neposredno razumijeti samo jezik binarnih brojeva neophodno je sve instrukcije i podatke s kojima on radi prevesti u takav oblik.
- Isto tako unutar računala različite komponente zahtijevaju određeni format i oblik zapisivanja u binarnom zapisu korištenjem određenih formata strojnih riječi.
- Pri tome se koriste odgovarajući kodovi za svaku vrstu podataka (brojčanih ili tekstualnih). Pod te kodove uvrštavamo različite simboličke kodove koji odgovaraju tim potrebama. To su:
 - tetradni kodovi
 - BCD
 - Aiken
 - Excess-3
 - Gray
 - ASCII
 - EBCDI
- Važno je da sustav kodiranja mora zadovoljiti načelo **jednoznačnosti i ekonomičnosti**. Jednoznačan je onaj kod u kojem su sve riječi različite. Ekonomičan je onaj kod koji ima najmanju redundanciju.

3.2.1. Potreba za kodom sa dovoljnim brojem permutacija

- Da bismo određenu vrstu podataka mogli u računalu kodirati određenim kodnim sustavom, dotični kodni sustav mora imati dovoljno veliku "abecedu", tj. mora sadržavati dovoljno velik broj permutacija elementarnih kodnih simbola.
- U digitalnim računalima kodni simboli su osnovne binarne znamenke – 0 i 1.
- Tako npr. "abecedu" tetradnih kodova čine sve četvero-znamenkaste (grč. tetra = četiri) permutacije binarnih 0 i 1.
- Želimo li kodirati samo dekadске brojeve, tada odabrani kodni sustav mora imati odgovarajuću binarnu kodnu permutaciju za svaku osnovnu dekadsku znamenku (0 do 9).
- Osnovnih dekadskih znamenki ima 10, pa kodni sustav mora imati barem 10 binarnih permutacija.
- Ukoliko kodiranje dekadskih znamenki izvršimo jednostavnom konverzijom dekadске znamenke u binarnu, tada je najdulja permutacija potrebna za kodiranje znamenke 9: $9_{(10)} = 1001_{(2)}$.
- Iz toga se vidi da je za uspješno kodiranje dekadске znamenke potreban binarni kôd od četiri binarne znamenke – tj. treba nam tetradni kôd.
- Želimo li kodirati cijelu abecedu govornog jezika (velika i mala slova), brojeve i specijalne znakove koje koristimo prilikom pisanja (interpunkcija, razmaci i ostali specijalni znakovi), tada tetradni kôd nije dovoljan jer omogućava samo 16 (24) permutacija – u abecedi govornog jezika imamo 30 slova (npr. velikih i onda još 30 malih slova), 10 osnovnih dekadskih znamenki te 10-ak interpunkcijskih znakova.
- Vidimo da trebamo kôd sa barem 80-tak permutacija, koje možemo prikazati sa 7 binarnih znamenki (6 nije dovoljno, jer 2^6 daje 64 permutacije, a 7 binarnih znamenki daje $2^7 = 128$ permutacija).
 - primjer – 7-bitni ASCII (American Standard Code for Information Interchange) kôd.

- 7-bitnim ASCII kodom nije moguće obuhvatiti znakove svih svjetskih (pa ni zapadnjačkih) jezika, pa postoji potreba za još duljim kodovima, npr. 8-bitnim kodom ($2^8 = 256$ permutacija) mogli bismo pokriti sva latinična pisma.
 - primjeri – prošireni 8-bitni ASCII kôd i EBCDI (Extended Binary Coded Decimal Interchange) kôd.
- Ni 256 permutacija nije dovoljno za znakove u pismima istočnjačkih jezika – zato je uveden 16-bitni ($2^{16} = 65536$ permutacija) kod koji je u stanju obuhvatiti sve znakove iz svih relevantnih svjetskih jezika.
 - primjer – Unicode.

3.2.2. Redundanca u kodu

- Redundanca (ili zalihost) u kodu je **razlika između broja mogućih i broja iskorištenih binarnih riječi (permutacija)** u nekom kodu.
- Strogo gledano, redundanca predstavlja neekonomičnost koda, tj. gubitke u kodu, ali se može korisno iskoristiti za povećanje sigurnosti u prijenosu kodiranih podataka – npr. paritetnim bitom za provjeru ispravnosti prenešenih podataka.
 - Primjer – redundanca kod tetradnih kodova:
 - Tetradni kod nudi 16 mogućih permutacija. Za kodiranje dekadskih znamenki je potrebno 10 permutacija. Iskoristivost tetradnog koda je prema tome:
 - $10/16 * 100 = 62,5\%$
 - Redundancu izražavamo u bitovima po slijedećoj formuli:
 - $R = \log_{(2)}M - \log_{(2)}K$
 - pri čemu je M maksimalni broj permutacija u kodu, a K broj iskorištenih permutacija u kodu
 - Za tetradni kod je $M=16$, a $K=10$, pa imamo slijedeću formulu:
 - $R = \log_{(2)}16 - \log_{(2)}10 = 4 - 3,32 = 0,68$ bita

3.2.3. BCD kôd

- Binarni i decimalni brojevi sustavi
 - Za razliku od decimalnog brojevnog sustava, kod binarnog sustava baza nije deset, nego broj dva.
 - U binarnom brojevnom sustavu postoje svega dvije znamenke: nula i jedan. One predstavljaju dva binarna stanja koja se u računalu pojavljuju kao impuls, odnosno stanje mirovanja ili akcije.
 - Memorija svakog računala izrađenog u binarnoj tehnici sastoji se od niza memorijskih elemenata koji predstavljaju stanje 0 ili 1.
 - Kako bi se u memoriji mogli smjestiti podaci izraženi u decimalnom brojevnom sustavu, potrebno je da se memorijski elementi grupiraju u grupe po četiri.
 - Kombinacijom vrijednosti elemenata u takvoj grupi dobivamo željeni broj od nula do devet.
 - Oznake sastavljene od četiri pozicije nazivaju se tetrade, a kombiniranjem njihovih položajnih vrijednosti može se izraziti bilo koji broj od nula do devet. Broji se na način uobičajen za binarno brojenje od 0 do 9, a zatim se prelazi na početnu vrijednost.
- Kodiranje
 - Ako željenu decimalnu vrijednost želimo prikazati binarno, tada ćemo oznakom jedan prikazati onu mjesnu vrijednost koja zbrojena s ostalim daje željenu decimalnu vrijednost, a nulom označiti nepotrebnu mjesnu vrijednost. Svakom decimalnom broju pridružena je određena mjesna vrijednost.
 - Kôd predstavlja oznaku fizičkog alfabeta koje su kombinirane prema unaprijed utvrđenim pravilima sa svrhom da vjerno izraze određene semantičke oznake i njihove kombinacije ili fizičke oznake izražene nekim drugim kodom.
 - Primjena pojedine vrste koda ovisi o nosiocu podataka i zahtjeva izvođenje operacija u kodovima. Računalu su prepoznatljivi samo znakovi izraženi pomoću dva diskretna stanja, stanje jedan ili stanje nula.
 - **Kodiranje predstavlja proces povezivanja alfabeta s drugim alfabetom**, odnosno prijelaz s jednog znaka alfabeta na odgovarajući drugi znak. Pravilo za prelazak je dan listom povezivanja, tzv. kodom. On predstavlja ključ za kodiranje.

- Upotrebom tetrađa dobivamo tetrađne kodove koji se temelje na izražavanju brojeva pomoću četiri bita.
- Četiri bita dovoljna su za izražavanje oznaka decimalnog brojevnog sustava, a da nije u potpunosti iskorištena zaliha znakova tetrađe.
- To omogućuje stvaranje tetrađnih kodova među kojima je najpoznatiji i najjednostavniji BCD tetrađni kôd.
- Uz BCD kôd koriste se i Aiken kôd, Excess 3 kôd i Gray kôd.
- BCD kôd
 - Kada se za prikaz decimalnih brojeva upotrebljavaju binarni brojevi, to se naziva **binarno kodirane dekade**, odnosno BCD kod.
 - Taj način zapisivanja brojeva stvara kompromis između binarnog i decimalnog brojnog sustava. Unutar jedne dekade brojevi se zapisuju na binarni način, a dekade se međusobno povezuju kao kod decimalnog brojnog sustava.
 - Na taj način se mogu zapisivati i relativno veliki brojevi, jer čitanje binarnih brojeva zapisanih na taj način ne zadaje velike poteškoće.
- Brojevi se u BCD kodu memoriraju u tzv. **pakiranom obliku**, što znači da se svakom tetradom iskazuje po jedna decimalna oznaka, osim sa zadnjom tetradom kojom se iskazuje predznak broja. Kada postoje neiskorištene tetrađe, one se nazivaju pseudotetrađe.
- BCD kôd nije ništa drugo nego prikaz brojeva binarnim znamenkama prema decimalnom sustavu.
- Ime tog koda je izvedeno od engleskog izraza “Binary Coded Decimal” što znači “binarno kodirani decimalni brojevi”.
- Osnova BCD koda se sastoji u tome da se svaka decimalna znamenka može zapisati sa četiri bita, odnosno četiri binarna mjesta.
- Svaki pojedinačni znak direktno se izražava pomoću jedne tetrađe, a tetradama se dodijeli decimalna vrijednost mjesta, tj. težinska vrijednost.
- Neparni decimalni brojevi prilikom BCD zapisa na kraju imaju jedan, dok parni na kraju imaju nulu.
- Broj 2960 – zapisan sa četiri tetrađe u BCD zapisu:

```
0010 1001 0110 0000
 2   9   6   0
```

BCD	Decimalno
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

TABLICA 3.3: BCD KÔD

3.2.4. ASCII i EBCDI kôd

- Svi podaci, bez obzira na to što znače, zapisuju se u digitalnim računalima u obliku binarnih brojeva. Oni se upotrebljavaju zato što je za njihov prikaz dovoljno imati elemente sa samo dva stabilna stanja. Pomoću ovih kodova se zapisuju podaci, naredbe, slova, razni znakovi i drugo.
- **Prikaz nenumeričkih veličina**
 - Spomenuli smo da se u digitalnom računalu mogu zapisivati samo binarni brojevi, no digitalna računala rade i sa slovima i drugim znakovima.
 - Računalo ne zapisuje slova i druge znakove u izvornom obliku, već se različiti simboli prikazuju i zapisuju pomoću binarnih brojeva, koji se potom tumače na određeni način.
 - Simboli se u računalu zapisuju brojevima pisanim u nekom kodu koji određuje kako se tumači binarni broj.
 - Postoje razni kodovi koji određuju kako se tumači binarni broj (npr. ASCII, EBCDI...).
 - Svaki kod ima neke prednosti na svom specifičnom području. No, svi zajedno, jer nisu međusobno usklađeni, mogu izazvati zbrku. U takvoj situaciji standardizacija se nameće kao jedini izlaz – **kao standard je definiran ASCII kod**.
- **Alfanumerički kodovi – ASCII i EBCDI**
 - Među alfanumeričkim kodovima najčešće je u upotrebi kôd poznat pod nazivom ASCII (engl. American Standard Code for Information Interchange).
 - To je sedmerobitni kôd, što daje 128 kombinacija, što je dovoljno za prikaz svih znamenaka, slova i znakova.

- Ovaj kôd se koristi u prijenosu podataka između računala i ulazno - izlaznih jedinica, te kao oblik kodiranja alfanumeričkih podataka prilikom njihove pohrane u memoriji.
- Obično se kodnoj grupi dodaje i osmi bit koji služi za kontrolu pariteta, ali neki proizvođači koriste taj bit za proširenje područja kodiranja za daljnjih 128 znakova.
- Sve informacije koje se prenose ovim kodom mogu se podijeliti u dvije grupe znakova i to:
 - **grafičke znakove** i
 - **upravljačke signale.**
- Grafički znakovi obuhvaćaju sve alfanumeričke znakove, interpunkcije i druge simboličke znakove.
- Upravljački signali s jedne strane osiguravaju upravljanje komunikacijom, a s druge strane upravljaju formatiranjem poruka.
- Uz ASCII kod vrijedno je spomenuti i osmerobitni EBCDI (engl. Extended BCD Interchange Code). Taj kôd daje ukupno 256 mogućih kombinacije što je i više negoli potrebno za znamenke, slova i znakove pa ostaje veliki broj kombinacija koje koristimo za uporabu kao upravljačke signale.
- Kao u ASCII kodu, znakovi EBCDI koda dijele se na grafičke i upravljačke i uglavnom su jednaki znakovima u ASCII kodu.
- Primjena EBCDI koda više je orijentirana na obradu podataka, a manje na komunikaciju jer u okviru 8 bitova nije sadržan paritetni bit.

3.3. Osnovne računске operacije u brojevnim sustavima

- Zbrajanje binarnih brojeva:
 - Pravila zbrajanja binarnih brojeva:

Izraz	Rezultat	Prijenos
0 + 0	0	0
0 + 1 (ili 1 + 0)	1	0
1 + 1	0	1
1 + 1 + 1	1	1

TABLICA 3.4: PRAVILA ZBRAJANJA BINARNIH BROJEVA

- *Postupak zbrajanja*
 - pretpostavimo da treba zbrojiti dva višebitna binarna broja.
 - počevši zbrajanje od pozicionog mjesta najmanje težine, tj. krajnjeg desnog bita, i idući dalje nalijevo, dobije se:

$$\begin{array}{r}
 01101 \\
 + 10010 \\
 \hline
 11111
 \end{array}$$

- *Zbrajanje kod kojeg postoji prijenos (1 dalje)*
 - nešto složenije je zbrajanje kod kojeg postoji jedan dalje, tj. prijenos jedinice u stupac veće težine neposredno lijevo.

$$\begin{array}{r}
 \text{Prijenos} \rightarrow \quad 11111 \\
 \quad \quad \quad 10101 \\
 + \quad \quad \quad 11111 \\
 \hline
 \quad \quad \quad 110100
 \end{array}$$

- u ovom slučaju prva dva bita s desna (1 i 1) daju "0 pišem, 1 dalje", taj "1 dalje" iz prvog stupca s desna prelazi u drugi stupac i zbraja se sa znamenkama drugog stupca (1 i 0).
- Oduzimanje binarnih brojeva korištenjem dvojnog komplementa:
 - Uloga dvojnog komplementa je svodenje postupka oduzimanja binarnih brojeva na postupak zbrajanja binarnih brojeva.
 - Zbog toga u aritmetičko-logičkoj jedinici procesora nije potrebno imati posebne sklopove za oduzimanje, već se u tu svrhu koriste postojeći sklopovi za zbrajanje.

Primjer:

- Oduzimanje brojeva 11011001(2) i 1010(2) (11011001 – 1010) korištenjem dvojnog komplementa
- prvo se broj koji se oduzima (1010₍₂₎) nadopuni vodećim nulama, tako da ima jednak broj znamenaka kao i umanjnik (1010₍₂₎ → 00001010₍₂₎), pa se zatim takav broj komplementira (jedinice se pretvore u nule i nule u jedinice).
- time se dobiva jedinični komplement 11110101₍₂₎.
- zatim se jediničnom komplementu pribroji binarni 1 čime se dobiva dvojni komplement 11110110₍₂₎.
- na kraju se dvojnem komplementu pribroji broj od kojeg se oduzima (11011001₍₂₎) i rezultat je razlika dvaju zadanih binarnih brojeva (11001111₍₂₎).
- u slučaju da na najvećem težinskom mjestu rezultata dođe do prijenosa jedinice, ona se u konačnom rezultatu zanemaruje (upravo se to dogodilo u zadanom primjeru).

$$\begin{array}{r}
 11011001 \\
 - \underline{00001010} \\
 \hline
 \text{?}
 \end{array}
 \quad \rightarrow \text{Jedinični komplement: } 11110101 \quad \rightarrow \quad 11110101$$

$$\begin{array}{r}
 \\
 \\
 + \\
 \\
 \hline
 11110110
 \end{array}$$

Razlika zadanih brojeva:

$$\begin{array}{r}
 11011001 \\
 + \\
 \hline
 = 1 \\
 = 11001111 \quad \text{bit preljeva} \\
 = 11001111
 \end{array}$$

3.4. Točke kodiranja

- Kad govorimo o točkama kodiranja, zapravo govorimo o mjestima (lokacijama) u računalnom sustavu na kojima se izvodi kodiranje, tj. pretvorba podataka iz jednog kodnog oblika u drugi kodni oblik.
- U računalnom sustavu imamo dvije tipične točke kodiranja (mada to nisu jedine točke kodiranja):
 - tipkovnica – memorija
 - memorija – monitor/pisač

TIPKOVNICA – MEMORIJA:

- Pritiskom neke tipke na tipkovnici događa se ovo:
- Tipka djeluje kao sklopka i njenim pritiskom se zatvara strujni krug u tipkovnici i po točno određenom vodiču (koji pripada pritisnutoj tipki) krene strujni impuls.
- Mikrokontroler u tipkovnici "dešifrira" na temelju vodiča koja je tipka bila pritisnuta i šalje u računalo odgovarajući kôd (tzv. scancode) – tu već dolazi do prvog kodiranja.
- Pristupni sklop (interface) za tipkovnicu prima scancode i u suradnji sa operativnim sustavom konvertira primljeni scancode znaka u odgovarajući kôd na temelju kojeg računalo obavlja trenutni zadatak (npr. u ASCII kôd – druga razina kodiranja) i npr. pohranjuje taj kôd u memoriju na odgovarajuću lokaciju.

MEMORIJA – MONITOR/PISAČ:

- Npr. u memoriji na nekoj lokaciji imamo pohranjen binarni podatak u kodnom sustavu koji je potreban za izvršavanje zadanog zadatka nad dotičnim podatkom – npr. pohranjen je ASCII kôd znaka 'A' (veliko slovo A).
- Taj znak treba prikazati na ekranu ili ispisati na pisač – ni ekran ni pisač (većina pisača) ne znaju raditi s podacima kodiranim u ASCII obliku, već prikazuju sadržaje koji su oblikovani kao slike sastavljene od niza točaka.
- Pristupni sklopovi monitora ili pisača, u suradnji s operativnim sustavom, "prevode" primljeni ASCII kod znaka A u "sliku" slova A koju dotični uređaji mogu reproducirati – prikazati na ekranu ili ispisati na papir.

3.5. Zapisivanje brojeva u fiksnom zarezu

- U ovom načinu zapisivanja podrazumijeva se fiksirani broj decimalnih mjesta u binarnom zapisu broja – npr. 2 dec. mjesta, 5 dec. mjesta...
- Fiksni zapis decimalnih brojeva je potpuno precizan toliko dugo dok se:

- ne dostigne maksimalna vrijednost (ovisi o broju bitova koji se koriste pri implementaciji zapisa broja u fiksnom zarezu), ili
- ne pokuša zapisati broj sa većim brojem decimalnih mjesta nego što određena implementacija zapisa u fiksnom zarezu dozvoljava – tada dolazi do zaokruživanja zadnje dopuštene decimale i odbacivanja viška decimalnih mjesta.
- Zapis u pokretnom zarezu rješava problem broja decimalnih mjesta, ali zbog načina zapisa broja u pokretnom zarezu su moguća neželjena zaokruživanja decimalnih mjesta i gubitak preciznosti zapisa.
- Zapis u fiksnom zarezu se često koristi:
 - Kod zapisivanja novčanih veličina (uvijek 2 fiksne decimale, ne tolerira se gubitak preciznosti koji može prouzročiti pokretni zarez),
 - Kod uređaja koji nemaju jedinicu za rad sa pokretnim zarezom, ili
 - U situacijama kad su potrebne bolje performanse aplikacija.
- Npr. 16-bitni zapis u fiksnom zarezu sa 4 decimalna mjesta bi mogli predstaviti na sljedeći način:

Težinske vrijednosti											Pozicija dec. točke	Težinske vrijednosti				
2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰		2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴
Cjelobrojni dio												Decimalni dio				

TABLICA 3.5: 16-BITNI ZAPIS U FIKSNOM ZAREZU SA 4 DECIMALNA MJESTA

- Pozicija decimalne točke je zamišljena, za nju se ne troši niti jedan bit u zapisu.
- Ukoliko treba prikazati predznačeni broj, tada bit najveće težine u cjelobrojnom dijelu (ovdje je to težinsko mjesto 2¹¹) poprima ulogu **predznaka**.

3.6. Zapisivanje brojeva u pokretnom zarezu

- Pri ovom načinu zapisivanja svaki se broj zapisuje u obliku mantise i eksponenta, čime se uklanja problem broja decimalnih mjesta koji postoji kod zapisa sa fiksnim zarezom.
- Eksponent pokazuje koliko se brojeva ulijevo ili udesno mora pomaknuti pozicioni zarez u mantisi da bi se dobila prava vrijednost broja.
- Primjer bilježenja broja u pokretnom zarezu može se prikazati koristeći npr. trostruku preciznost zapisivanja broja, kao što je prikazano na sljedećoj slici:



SLIKA 3.6: BILJEŽENJE BROJA U POKRETNOM ZAREZU

3.7. Provjera pariteta

- Kod obrade podataka u digitalnim računalnim sustavima prenosi se velik broj podataka između različitih jedinica za obradu. Zbog razmjene između različitih jedinica, mogu nastati greške u prijenosu.
- Da bi se takve greške otkrile, upotrebljava se (kao jedna od mogućih tehnika) provjera pariteta.
- Binarni podaci se šalju tako da postoji određeni broj jedinica u binarnom podatku.
- Da bi se mogao provjeriti paritet mora postojati određeni sklop koji radi taj posao.
- Može se dogoditi da se prilikom prijenosa promijene dva bita – tada provjera pariteta ne bi uočila grešku. No, vjerojatnost pojave greške jednog bita je 1:10⁵, a vjerojatnost da se pogrešno prenese dva bita 1:10¹⁰, te se ona u praksi zanemaruje.
- Provjera pariteta može biti:
 - PARNNA – svaka prenešena riječ mora sadržavati parni broj jedinica.
 - NEPARNA – svaka prenešena riječ mora sadržavati neparan broj jedinica.

- Konstruktori računala odlučuju koji će se oblik koristiti, a među njima nema značajne razlike.
- U većini sustava je jedan bit u podatku rezerviran za provjeru pariteta.
 - Npr. kod 8-bitnog podatka je to najznačajniji bit, a ostalih 7 bitova se koristi za prijenos stvarnih podataka.

Primjer:

- Ako se koristi parna provjera pariteta, a podatak već sadrži paran broj jedinica, onda se u paritetni bit zapisuje nula, kako bi broj jedinica ostao paran (ako je broj jedinica u podatku neparan, tada u bit pariteta zapisuje jedinica, kako bi se postigao paran broj jedinica u podatku – kao što to zahtijeva parna provjera pariteta).
- Prijamnik mora na svaki način primiti paran broj jedinica, a ako ne primi, zna se da se dogodila greška i da se prijenos treba ponoviti.

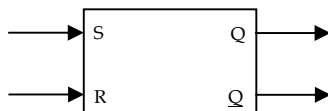
3.8. Bistabili

- Bistabili su svi oni elektronički elementi (ili elementi neke druge prirode) koji **imaju dva stabilna stanja**. Pod stabilnim stanjem podrazumjeva se ono stanje u kojem razmatrani element može ostati proizvoljno dugo i koje se može promijeniti samo nekom vanjskom intervencijom.
 - Npr. prekidač koji je otvoren ili zatvoren.
- U suvremenoj računalnoj tehnologiji najviše se upotrebljavaju bistabili ostvareni različitim elektroničkim sklopovima. Najviše se, ipak upotrebljavaju elektronički bistabili koji se u svojem elementarnom obliku sastoje od jednog para **tranzistora**.
- Daljnji elektronički element koji ima ulogu bistabila je **magnetska jezgra** koja, magnetizirana u jednom smjeru označava stanje 0, a u drugom znači stanje 1.
- Ovi elementi se najviše upotrebljavaju u računalnoj tehnologiji jer njihovo stanje može se promijeniti u vremenu manjem od jedne mikrosekunde ili mnogo brže oko nekoliko nanosekundi.
- **Brzina** im omogućava da obave velik broj operacija u vremenu daleko manjem od jedne sekunde.
- Uloge ulaza i izlaza kod bistabila
 - S logičkog gledišta, bistabil se može označiti jednim simbolom za element koji ima dva stabilna stanja, bez obzira na to kako se bistabil realizira.
 - Postoji više logičkih simbola za bistabile – takav jedan je prikazan na slici dolje.
 - Na simbolu postoje dva ulaza S i R i dva izlaza Q i \bar{Q} .
 - Kada dođe pobuda signala, odnosno stanje 1, na ulaz S, bistabil se postavi u stanje 1 i tada je na njegovu izlazu Q stanje 1, a na komplementarnom izlazu \bar{Q} stanje je nula.
 - Kada dođe pobuda na njegov ulaz R, bistabil se postavi u stanje nula, tj. na izlazu Q stanje je 0 a na komplementu \bar{Q} stanje 1.
- Moguća stanja bistabila:
 - Kada na izlazu S i R nema pobude on se nalazi u stanju izazvanom zadnjom pobudom, odnosno bistabil pamti zadnji binarni podatak.
 - Kada stanje 1 naiđe na ulaz S i na ulaz R, tada bi se bistabil trebao postaviti istovremeno i u stanje 1 i u stanje 0, što je nemoguće prema definiciji bistabila. Ne mogu se istovremeno vrata zatvarati i otvarati. Zbog toga se stanje 1 na S i R smatra nedopuštenim i ne može se upotrebljavati u normalnom radu.
 - Kada se na jednome ulazu nalazi 0, a na drugome 1, tada se mijenja stanje bistabila; stanje sa ulaza S se prenosi na izlaz Q, a stanje sa ulaza R se prenosi na izlaz \bar{Q} .
 - Zbog mogućnosti pamćenja stanja, bistabil je pogodan za zapisivanje binarnih podataka u složenijim računalnim sklopovima.
- Tablica stanja S-R bistabila:

S	R	Q	\bar{Q}
0	0	Zadržava se prijašnje stanje	
1	0	1	0
0	1	0	1
1	1	Nedozvoljeno stanje	

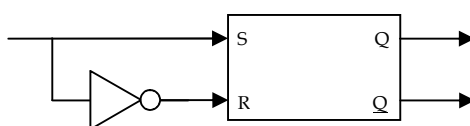
TABLICA 3.7: TABLICA STANJA S-R BISTABILA

- Logički simbol za S-R bistabil:



SLIKA 3.8: LOGI KI SIMBOL ZA SR BISTABIL

- Varijanta bistabila – **D-bistabil**
 - Prilikom povezivanja većeg broja bistabila u niz (npr. kod realizacije registara) poželjno je čim više **uštediti na broju vodova** unutar složenijih sklopova.
 - To je moguće izvesti pomoću D-bistabila, kod kojeg se štedi na jednom ulaznomvodu.
 - Ušteda se ostvaruje drugačijom realizacijom bistabila, pri čemu se koristi definicija dozvoljenih stanja bistabila.
 - Kako su praktički dozvoljena samo stanja u kojima ulazi S i R imaju suprotne vrijednosti, u sklop se dodaje NE sklop koji invertira vrijednost ulaza S i prenosi ga na ulaz R

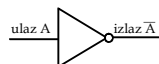


SLIKA 3.9: LOGI KI SIMBOL ZA D BISTABIL

3.9. Osnovni logički sklopovi

- Koliko god se računalo činilo složeno, njegov rad se može prikazati kombinacijama dva binarna stanja tj. može se prikazati korištenjem znamenki binarnog brojevnog sustava.
- Digitalni sklopovi koji rade pomoću znamenki binarnog brojevnog sustava nazivaju se logički sklopovi. Logički sklopovi mogu imati jedan ili više ulaza.
- Naponi na ulazima i izlazima logičkih sklopova mogu se prikazati korištenjem prije spomenutih znamenki binarnog brojevnog sustava tj. 0 i 1.
- Stanje na izlazu sklopa vezano je za ispunjenje određenih uvjeta na ulazima tj. između stanja na ulazima i stanja na izlazu postoji određena logička veza, odnosno logički sklopovi obavljaju logičke funkcije ili operacije.
- Engleski matematičar George Boole razvio je u 19. stoljeću logičku algebru koju koristimo za analizu i sintezu logičkih sklopova.
- Korištenjem metoda logičke algebre mogu se prikazati stanja logičkih sklopova jednostavnim algebarskim jednadžbama.
- Logička stanja sklopova mogu se prikazati tablicama stanja.
- Tablica stanja je prikaz svih kombinacija ulaznih binarnih veličina i tako dobivenih stanja na izlazima.
- Bitno je za napomenuti da proizvođači takvih sklopova često umjesto binarnih oznaka 0 i 1 koriste oznake L (engl. Low) i H (engl. High) tj. prikazuju naponska stanja pojedinih sklopova.
- Logički sklopovi kod kojih stanje na izlazu ovisi o trenutnom stanju na ulazu nazivaju se **kombinacijski** logički sklopovi i nemaju sposobnost pamćenja binarnih stanja.
- Za izgradnju računala potrebni su i sklopovi koji imaju mogućnost pamćenja stanja. Sklopovi kod kojih stanje na izlazu ovisi o prethodnom stanju na ulazu i o prethodnom stanju na izlazu nazivaju se **sljedni** logički sklopovi i za razliku od kombinacijskih sklopova imaju mogućnost pamćenja binarnih stanja.

3.9.1. NE sklop



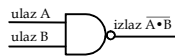
3.9.2. I sklop



3.9.3. ILI sklop



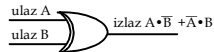
3.9.4. NI sklop



3.9.5. NILI sklop

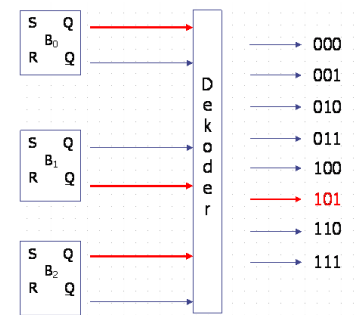


3.9.6. EX-ILI sklop



3.10. Dekoderi

- Dekoderi su sklopovi koji omogućuju, u digitalnoj tehnici, da svako stanje na registru rezultira stanjem 1 **na samo jednom jedinom izlaznomvodu** radi pokretanja neke akcije, a stanjem 0 na svim ostalim izlaznim vodovima.
- Stoga su dekoderi obično smješteni uz registre
- Upotrebom dekodera izvode se različita adresiranja u računalu, te se pokreću različite akcije kojima je ishodište odgovarajući broj na registru.
- Na slici je prikazan dekodер uz registar sa tri bita.
- Tri bita daju ukupno 8 kombinacija ($2^3 = 8$), tj. dekodер uz registar od 3 bita mora imati 8 izlaznih vodova (n -bitni registar $\rightarrow 2^n$ izlaznih vodova)
- Aktivan će biti samo jedan izlazni vod, ovisno o broju zapisanom u registru.
- U ovom slučaju je aktivni vod na skici naznačen binarnom broječanom vrijednošću iz registra.
- Npr. ako je u registru zapisan broj 5 (binarno 101), tada će biti aktivan jedino izlazni vod pored kojeg je oznaka "101".



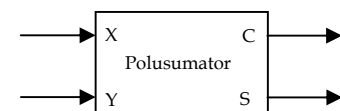
SLIKA 3.10: DEKODER SA TRI ULAZA

3.11. Brojila

- Brojilo služi, kako mu i ime kaže, za brojanje binarnih signala.
- Na ulaz u brojilo se dovode kratkotrajni impulsi, a brojilo pamti broj takvih pristiglih impulsa.
- Brojilo dobijemo tako da nekoliko S-R bistabila povežemo u seriju pomoću elementarnih logičkih sklopova.
- N bistabila u nizu čini N-bitno brojilo, koje može izbrojiti maksimalno 2^N brojeva.
- Bistabili su poredani po težinskim vrijednostima.
 - Prvi bistabil reprezentira težinsko mjesto 2^0
 - Drugi bistabil reprezentira težinsko mjesto 2^1
 - ...
 - N-ti bistabil reprezentira težinsko mjesto 2^{N-1}
- Osnovu brojenja čini mijenjanje (komplementiranje) stanja bistabila u situaciji kada bistabil niže težinske vrijednosti prelazi iz stanja 1 u stanje 0. Tada se iz bistabila niže težinske vrijednosti šalje signal koji "okida" bistabil više težinske vrijednosti i postavlja ga u stanje 1.
- Postoje brojila koja mogu samo povećavati vrijednost brojača, koja mogu samo smanjivati vrijednost brojača ili oboje.
- Brojila imaju značajnu ulogu u radu računala – jedan od najvažnijih registara u mikroprocesoru (*programsko brojilo*) je realiziran kao brojilo.

3.12. Poluzbrajač (polusumator)

- Osnovni sklop za strojno zbrajanje dvaju binarnih znamenaka je polusumator.
- Ulaze u polusumator X i Y čine dva binarna broja koja treba zbrojiti.
- Izlaz S je zbroj, a C je "jedan dalje" (engleski: carry), odnosno prijenos.
- Polusumator zbraja samo dvije binarne znamenke, no to nije dovoljno za potpuno zbrajanje.
- Naime, kako polusumator ima samo dva ulaza, ne može uzimati u obzir i eventualni "jedan dalje", koji može postojati brojnom mjestu niže težinske vrijednosti, te su za to potrebni dodatni sklopovi.

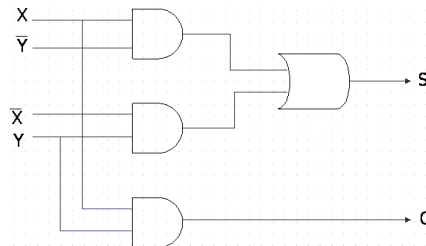


SLIKA 3.11: SIMBOL POLUSUMATORA

ULAZI		IZLAZI	
X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLICA 3.12: TABLICA STANJA POLUSUMATORA

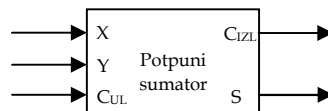
- Ako su oba ulaza 0, onda su i oba izlaza 0.
- Ako je samo jedan ulaz 0, a drugi je u stanju 1, onda je izlaz S u stanju 1, a izlaz C u stanju 0.
- Napokon, ako su oba ulaza u stanju 1, onda će na izlazu postojati samo "jedan dalje" tj. C je 1, a S je 0.



SLIKA 3.13: REALIZACIJA POLUSUMATORA POMOĆU OSNOVNIH LOGIČKIH SKLOPOVA

3.13. Potpuni zbrajač (potpuni sumator)

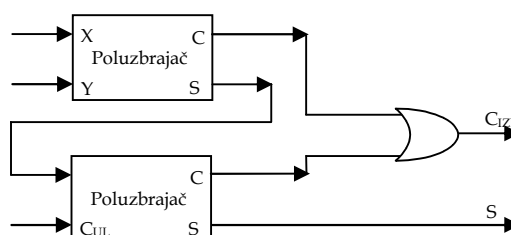
- Polusumator nije dovoljan za potpuno zbrajanje – sklop koji uzima u obzir i "1 dalje" sa mjesta niže težinske vrijednosti naziva se potpuni sumator.
- Potpuni sumator ima **tri ulaza i dva izlaza**.
 - Ulazi X i Y za binarne znamenke koje treba zbrojiti.
 - Ulaz C_{UL} za prihvaćanje signala "1 dalje" ima dva izlaza.
 - Izlaz S sa potpunim zbrojem dviju binarnih znamenaka.
 - Izlaz C_{IZ} sa izlaznim "1 dalje" koji se šalje na ulaz sumatora veće težinske vrijednosti.
- Realizira se pomoću dva povezana polusumatora.



SLIKA 3.14: SIMBOL POTPUNOG SUMATORA

ULAZI			IZLAZI	
X	Y	C_{ul}	S	C_{iz}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

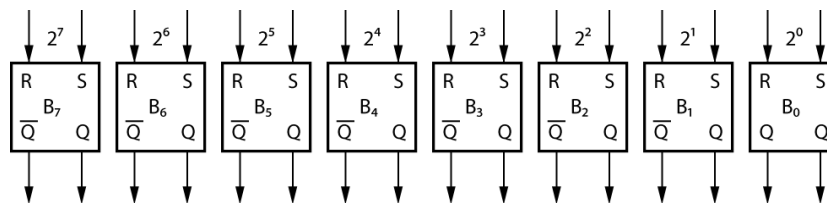
TABLICA 3.15: TABLICA STANJA POTPUNOG SUMATORA



SLIKA 3.16: REALIZACIJA POTPUNOG SUMATORA POMOĆU DVA POLUSUMATORA

3.14. Registri

- Jedan bistabil može svojim stanjima prikazati samo jedan bit podatka.
- Podaci se obično sastoje od više bitova, koji zajedno tvore jedan ili više bajtova ili riječi.
- Za zapis takvog kompletnog podatka potrebno je više bistabila u nizu na koje se može pohraniti čitav podatak – takav **niz bistabila** se zove **registar**.
- Bistabili u registru su međusobno neovisni, ali su poredani tako da svaki ima određenu težinsku (pozicionu) vrijednost – kao i binarni brojevi koji se na njih zapisuju.
- Imamo bistabil težinske vrijednosti 2^0 , pa 2^1 , ..., do 2^{n-1} gdje je n broj bistabila u registru.
- Zbog toga što računala barataju sa strojnim riječima fiksne duljine (8 bita, 12 bitova, 16 bitova, 32 bita...) i većina registara u računalu je jednake fiksne duljine kao i strojna riječ, kako bi se između registara mogli čim lakše razmjenjivati podaci.
- Podaci se između registara razmjenjuju putem sabirnica.
- Skupina registara je spojena na odgovarajuću sabirnicu (ili sabirnice).
- Izvorišni registar prenosi pomoću I-brane svoj sadržaj na sabirnicu (na poticaj odgovarajućeg signala jedino I-brane na izazima izvorišnog registra propuštaju signale na sabirnicu).
- Podaci putuju sabirnicom.
- Podatke preko I-brane prima samo odredišni registar (na poticaj odgovarajućeg signala jedino I-brane ispred ulaza odredišnog registra propuštaju signale).



SLIKA 3.17: SKICA REGISTRA SA 8 BISTABILA

3.15. Izvođenje instrukcije

- Format procesorskih instrukcija:
 - Instrukcije su u računalu predočene u obliku numeričkog (binarnog) koda. Instrukcija, predstavljena skupom bitova, sastoji se od nekoliko dijelova ili polja bitova.
 - Ta polja određuju operacije i lokacije (a u nekim slučajevima i izravno vrijednosti) operanada i rezultata.
- Oblik instrukcije opisuje se njenim formatom koji se sastoji od:
 - polja operacijskog koda,
 - nijednog, jednog ili većeg broja adresnih polja, te
 - polja za specifikaciju operanada i rezultata.
- Duljina instrukcije može varirati od, primjerice, dva do dvadeset i dva bajta.

Redni broj bitova:	0	1	2	3	4	5	6	7	8	31	
Ono što piše u instrukciji:	OPERACIJA ili INSTRUKCIJSKI KÓD				1	NAČIN ADRESIRANJA			ADRESA		

SLIKA 3.18: SHEMATSKI PRIKAZ FORMATA INSTRUKCIJE
(! ETVRTI BIT ['1'] OZNAČAVA DA SE RADI O 32-BITNOJ INSTRUKCIJI)

IZVOĐENJE INSTRUKCIJE:

- Za izvođenje pojedinih instrukcija upotrebljava se jedan ili više instrukcijskih ciklusa M1, M2, M3 i M4. Svaki instrukcijski ciklus obuhvaća određeni broj stanja, tj. memorijskih ciklusa.
- U svakom se instrukcijskom ciklusu obavljaju neke karakteristične operacije potrebne za izvođenje instrukcija. Instrukcije se, ovisno o njihovoj složenosti izvršavaju u jednom ili više memorijskih ciklusa. Ako su jednostavne, izvršavaju se u prvom memorijskom ciklusu.
- U prvom ciklusu uvijek se iz memorije čita prvi bajt instrukcije koji predstavlja operacijski kôd, pa na osnovi tog koda računalo odluči da li se instrukcija sastoji samo tog bajta ili ih ima više.
- Ako ih ima više, onda treba više puta zahvaćati podatke iz memorije, te je potrebno i više ciklusa.

- Ako se, pak, naredba sastoji od samo jednog bajta, ona se može završiti u jednom memorijskom ciklusu, što se obavlja u četvrtom ili petom stanju prvog ciklusa.
- Cjelokupan način djelovanja upravljačke jedinice tijekom izvođenja programa, ma kolikogod on složen bio, može se predočiti izmjenama dvaju osnovnih stanja za svaku instrukciju.
- Ta dva osnovna stanja su:
 - **Stanje PRIBAVI** (engl. Fetch)
 - **Stanje IZVRŠI** (engl. Execute).
- Ova se dva stanja, koja odgovaraju tijeku izvođenja instrukcija, pravilno izmjenjuju, u zavisnosti od frekvencije radnog signala, nekoliko milijuna, desetaka ili čak stotinama milijuna puta u sekundi.
- Tijekom izvođenja jedne instrukcije procesor prolazi kroz jedno stanje PRIBAVI i jedno stanje IZVRŠI.

OPIS STANJA PRIBAVI:

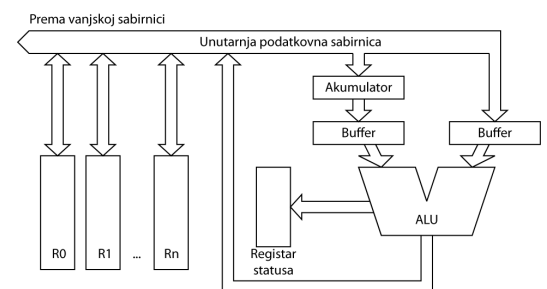
- Za stanje PRIBAVI karakteristično je da se za vrijeme njegova trajanja pribavlja (ili dohvaća) instrukcija iz memorije.
1. Iz memorije se pribavlja instrukcija. Točnije, ona se pribavlja iz memorijske lokacije na koju "pokazuje" poseban upravljački registar u procesoru. On se naziva programsko brojilo PC i sadrži adresu sljedeće instrukcije.
 2. Interno, u procesoru, povećava se sadržaj programskog brojila PC za jedan tako da na kraju stanja PRIBAVI ono (PC) pokazuje na sljedeću strojnu riječ u programu.
 3. Pribavljena instrukcija odnosno njen kod smješta se u instrukcijski registar IR. Instrukcijski se registar nalazi u procesoru i služi za pohranjivanje instrukcije koja se upravo izvodi.
 4. Upravljačka jedinica svojim sklopovima dekodira kôd instrukcije (naziva se i operacijski kôd) i na temelju ishoda dekodiranja zaključuje o kojoj je instrukciji riječ.
 5. U slučaju kad se instrukcija sastoji od većeg broja riječi pohranjenih u memoriji, na temelju dodatne informacije u operacijskom kodu ili pak u posebnom binarnom polju instrukcije koje određuje način adresiranja, upravljačka jedinica "doznaje" od koliko se riječi sastoji instrukcija i ponavlja korake 1. i 2. (stanja PRIBAVI), a dohvaćene riječi smješta u poseban registar.
 6. Stanje PRIBAVI završava onda kad je pribavljena kompletna instrukcija, tj. kad su iz memorije pribavljene sve riječi od kojih je instrukcija sastavljena.
 7. Nakon završetka stanja PRIBAVI upravljačka jedinica automatski prelazi u stanje IZVRŠI.

OPIS STANJA IZVRŠI:

- Bitna je značajka stanja IZVRŠI da se obavlja (izvršava) operacija koja je utvrđena u postupku dekodiranja. Naravno, ona se obavlja na podacima (operandima).
- I u stanju IZVRŠI može se pristupati memoriji, ali se dohvaćeni podaci iz memorije tumače kao operandi, a podatak koji se pohranjuje smatra se rezultatom.
- Nakon obavljenih operacija u stanju IZVRŠI upravljačka jedinica prelazi u stanje PRIBAVI i započinje dohvat sljedeće instrukcije iz niza instrukcija koje čine program.
- Svaka takva promjena stanja PRIBAVI-IZVRŠI odgovara potpunom izvođenju strojne instrukcije i nazva se **strojni ciklus**.

3.16. Organizacija procesora oko jedne sabirnice

- Procesor organiziran oko jedne sabirnice koristi jednu sabirnicu za dovođenje i odvođenje podataka u središnjem procesoru.
- Slika pokazuje jednu sabirnicu koja:
 - posluhuje oba buffer registra za potrebe ALU operacija,
 - odvodi podatke s ALU na vanjku sabirnicu
 - odvodi i dovodi podatke registrima.



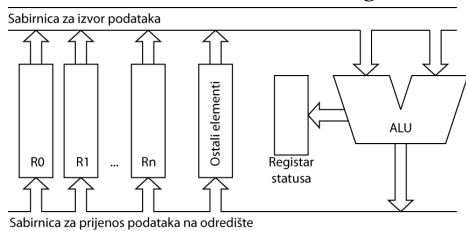
SLIKA 3.19: ORGANIZACIJA PROCESORA OKO JEDNE SABIRNICE

- U pravom procesoru, naravno, postoje tri sabirnice (upravljačka, podatkovna i adresna), no na slici je radi jednostavnosti crteža prikazana samo podatkovna.

- Za razliku od procesora organiziranih oko dvije ili više sabirnica, procesor organiziran oko jedne sabirnice može u jednom trenutku prenositi samo jedan podatak, tako da se podaci moraju prenositi vremenski multipleksirano i to u oba smjera.
- Radi toga je tako organizirani rad sporiji u odnosu na izvedbe s dvije ili tri sabirnice koji za opsluživanje različitih sklopova u procesoru imaju posebnu sabirnicu.
- Prednost takve organizacije je s druge strane u uštedi prostora jer se na procesoru utroši manji dio raspoložive površine.
- Rad s ovakvom organizacijom može se objasniti na sljedeći način (npr. ALU treba zbrojiti 2 broja i pohraniti ih u registar opće namjene ili u memoriju):
 - sabirnicom se prvo dovodi podatak s vanjske sabirnice (iz memorije) ili registra na buffer registar na desnom ulazu u ALU.
 - zatim se kroz akumulator dovodi podatak na lijevi bufer registar od ALU.
 - po obavljenoj AL operaciji podatak se istom sabirnicom provodi do registra opće namjene ili vanjskom podatkovnom sabirnicom u memoriju.

3.17. Organizacija procesora oko dvije sabirnice

- Spomenuli smo nedostatak organizacije procesora oko jedne sabirnice – u jednom času se unutar procesora može obavljati prijenos samo jednog podatka, što usporava procesor.
- Slika prikazuje moguću organizaciju procesora oko dvije sabirnice, pri čemu:
 - Jedna sabirnica služi za dovođenje podataka na ulaze ALU (izvor)
 - Druga sabirnica odvodi rezultat obrade u ALU do registra ili memorije (odredište).

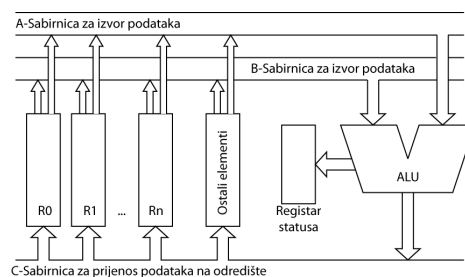


SLIKA 3.20: ORGANIZACIJA PROCESORA OKO DVIJE SABIRNICE

- Ponekad je moguće istovremeno prenositi podatke po obje sabirnice, čime se ubrzava rad sustava.
- Još uvijek nije moguće istovremeno pribaviti operande za oba ulaza ALU (pribavljaju se jedan po jedan, vremenski multipleksirano), ali se otvara mogućnost prenošenja rezultata prethodne operacije u ALU na zadano odredište **istovremeno** s pribavljanjem operanada za novu operaciju u ALU – ubrzanje rada.

3.18. Organizacija procesora oko tri sabirnice

- Već i organizacija procesora oko dvije sabirnice donosi ubrzanja u radu, a još veća poboljšanja su moguća uvođenjem treće sabirnice.
- Slika prikazuje moguću organizaciju procesora oko tri sabirnice, pri čemu:
 - Sabirnica A služi za dovođenje podataka na prvi ulaz u ALU (izvor A)
 - Sabirnica B služi za dovođenje podataka na drugi ulaz u ALU (izvor B)
 - Sabirnica C odvodi rezultat obrade u ALU do registra ili memorije (odredište).
- Poboljšanje u odnosu na organizaciju oko dvije sabirnice se očituje u mogućnosti istovremenog dovođenja obaju operanada na ulaze u ALU, a i dalje postoji mogućnost istovremenog odvođenja rezultata prethodne AL operacije na odredište.



SLIKA 3.21: ORGANIZACIJA PROCESORA OKO TRI SABIRNICE

3.19. Tehnologije izrade tranzistora

- Tranzistori su osnovni gradivni element poluvodičkih sklopova – služe za realizaciju svih elektroničkih sklopova kao njihov glavni aktivni element.
- Tranzistori se izrađuju od **poluvodičkih materijala** – materijala koji su veće električne vodljivosti od izolatora, a manje električne vodljivosti od vodiča.
- Poluvodički materijali po potrebi preuzimaju ulogu vodiča (provode struju) ili izolatora (ne provode struju).
- Najčešće korišteni poluvodički materijal je silicij.

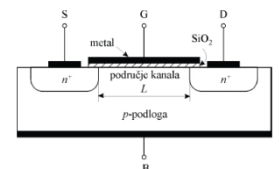
- Vodljivost poluvodiča ovisi o postojanju tzv. nosioca naboja (charge carrier) – nosioci naboja mogu biti:
 - **Elektroni** (nosioci *negativnog* naboja) – kod silicija koji je "onečišćen" materijalom sa viškom elektrona (više elektrona od čistog silicija, koji ima 4 elektrona) – osnova za n-tip poluvodiča.
 - **Protoni** (nosioci *pozitivnog* naboja) – kod silicija koji je "onečišćen" materijalom sa manjkom elektrona (manje elektrona od čistog silicija, koji ima 4 elektrona) – osnova za p-tip poluvodiča.
- Postoje dvije osnovne tehnologije za izradu tranzistora:
 - **Unipolarna (MOS) tehnologija** – koristi se samo jedna vrsta nosioca naboja.
 - **Bipolarna tehnologija** – koriste se obje vrste nosioca naboja.

3.19.1. Unipolarna tehnologija

- Kod unipolarne (MOS – Metal Oxyde Semiconductor, MOS-FET – Metal Oxyde Semiconductor - Field Effect Transistor) tehnologije se kao nosioc naboja upotrebljava samo jedna vrsta nosioca naboja – ili pozitivni ili negativni.
- MOS-FET tranzistori rade na principu djelovanja električnog polja koje se javlja protjecanjem struje između dviju elektroda.
- Električnim poljem se regulira propusnost (vodljivost) tranzistora.
- Razlikujemo tri osnovne izvedbe MOS tranzistora:
 - **NMOS**
 - **PMOS**
 - **CMOS**.

NMOS IZVEDBA TRANZISTORA:

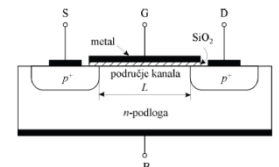
- Nositelj naboja su elektroni (negativni naboji – N tip MOS tranzistora – NMOS).
- Osnovu čini podloga (supstrat) izrađena od p-tipa poluvodiča.
- Na podlozi su izrađene dvije elektrode od n-tipa poluvodiča.
- Jedna elektroda se naziva izvor (source), a druga ponor (drain).
- Područje između njih se naziva kanal (channel).
- Na površini podloge je smješten izolator (najčešće silicijev dioksid), a sa izolatorom je povezana upravljačka elektroda (gate).
- Podešavanjem napona na upravljačkoj elektrodi se regulira električno polje, koje omogućava ili sprečava vodljivost tranzistora.



SLIKA 3.22: NMOS TRANZISTOR

PMOS IZVEDBA TRANZISTORA:

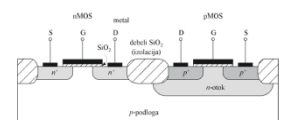
- Nositelj naboja su protoni (pozitivni naboji – P tip MOS tranzistora – PMOS).
- Za razliku od NMOS tranzistora, ovdje je supstrat izrađen od n-tipa poluvodiča, a izvor i ponor od p-tipa poluvodiča.



SLIKA 3.23: PMOS TRANZISTOR

CMOS IZVEDBA TRANZISTORA:

- CMOS = Complementary MOS.
- Za realizaciju CMOS tranzistora se koristi jedan NMOS i jedan PMOS tranzistor.
- NMOS i PMOS tranzistor su povezani zajedničkim ponorom.
- Glavna prednost (unatoč kompliciranijoj izradi) jest daleko manja potrošnja struje u odnosu na NMOS i PMOS tranzistore, pa se CMOS tranzistori koriste za izradu elektroničkih sklopova u sustavima kod kojih je bitna niska potrošnja energije.



SLIKA 3.24: CMOS TRANZISTOR

3.19.2. Bipolarna tehnologija

- Kod bipolarne tehnologije se upotrebljavaju obje vrste nosioca naboja – i pozitivni i negativni.
- Za bipolarne tranzistore su karakteristični procesi na spojištima dvaju tipova poluvodiča (n i p).
- Pod utjecajem vanjskog napona, spojište može biti propusno za nositelje naboja (vodič) ili nepropusno (izolator).

- Bipolarni tranzistori se sastoje od dva poluvodička spojišta – između dva poluvodička elementa istog tipa (bilo n-tipa, bilo p-tipa) se "u sendvič" smješta poluvodički element suprotnog tipa.
- "Lijevi" element se zove emiter (emitira nosioce naboja), "srednji" se zove baza, a "desni" element se zove kolektor (privlači nosioce naboja).
- Prema tome, ovisno o rasporedu poluvodičkih elemenata, moguće su dvije izvedbe bipolarnih tranzistora:
 - npn-tranzistor, te
 - pnp-tranzistor.

3.20. Unutarnje memorije

- Unutarnja memorija predstavlja (uz procesor) najvažniji dio računalnog sustava, bez kojeg cijeli sustav nema smisla i nije upotrebljiv.
- Njena uloga je da bilježi i (eventualno) trajno pamti binarne podatke (operande, instrukcije, međurezultate...) koji se koriste i nastaju tijekom izvođenja računalnih programa.
- Brzinom rada su u stanju parirati procesoru (za razliku od vanjskih memorija), ali su zato (opet u odnosu na vanjske memorije) manjeg kapaciteta, skuplje za izradu i (uglavnom) nisu u stanju pamtit i upisane podatke po nestanku napajanja.

3.20.1. Vrste unutarnjih memorija

- Digitalna računala obično posjeduju dvije vrste memorije – ROM i RAM memoriju.
 - ROM (Read Only Memory) – omogućava jedino čitanje pohranjenih sadržaja.
 - RAM (Random Access Memory) – omogućava i čitanje i pisanje.

3.20.1.1. ROM memorije

- ROM memorije služe za pohranu programa i podataka koji se za vrijeme rada računalnog sustava smiju samo čitati.
- Sadržaj ROM memorije određuje konstruktor računala i fiksno je definiran prilikom proizvodnje računala.
- Nakon ugradnje ROM čipova u računalni sustav, njihov se sadržaj može jedino čitati.
- Primjer: BIOS (Basic Input-Output System) ROM memorija:
 - Sadrži osnovne sistemske rutine potrebne za pokretanje računala – dakle, sadrži programe koji se moraju pokrenuti i izvršiti u fazi u kojoj operativni sustav računala još nije učitani i nije preuzeo kontrolu nad radom računala:
 - Početna dijagnostika i provjera ispravnosti hardverskih komponenti računalnog sustava (POST – Power-On Self Test).
 - Inicijalizacija hardverskih komponenti računalnog sustava.
 - Učitavanje operativnog sustava.

VRSTE ROM MEMORIJA:

- **Prave** ROM memorije – moguće ih je jedino čitati.
- **Programabilne** ROM memorije (PROM – Programmable ROM) – moguće ih je programirati od strane krajnjeg korisnika korištenjem posebnih uređaja (programatora).
 - Programiranje je moguće obaviti samo jednom – nakon toga se PROM memorije ponašaju kao prave ROM memorije, tj. moguće ih je samo čitati.
 - Osnovni memorijski element PROM memorija sadrži metalne vodove (osigurače), čijim se pregaranjem u PROM memoriju trajno upisuju binarne 0 i 1.
 - Odabirom vodova koji će biti pregoreni upravlja programator na temelju binarne slike programa kojeg treba upisati u PROM memoriju.
- **Izbrisive** ROM memorije (EPROM – Erasable PROM) – za razliku od PROM memorija, moguće ih je više puta programirati zahvaljujući mogućnosti brisanja već upisanih sadržaja.
 - S obzirom na tehniku brisanja sadržaja razlikujemo:
 - UV PROM memorije – sadržaj mem. elemenata se briše snažnim ultraljubičastim (UV) zračenjem.
 - EAPROM (Electrically Alternable PROM) ili EEPROM (Electrically Erasable PROM) – sadržaj mem. elemenata se briše električnim signalima kakvima se obavlja i čitanje sadržaja, samo u slučaju brisanja ti el. signali imaju veći napon i dulje traju.

- Bez obzira na vrstu programabilne ili izbrisive ROM memorije, postupak pisanja (i brisanja) je višestruko sporiji od postupka čitanja – iz toga je jasno vidljivo da, iako su programabilne, nisu predviđene za česte izmjene svog sadržaja.
- Usprkos trajnog pamćenja (i nakon nestanka el. napona), ovaj nedostatak ih čini nepogodnim za korištenje u ulozi memorijskog spremnika u kojeg će se pohranjivati promjenjivi sadržaji koji nastaju tijekom obrade računalnih programa – u tu svrhu se koriste RAM memorije.

3.20.1.2. RAM memorije

- RAM memorija se još naziva i radna memorija – služi kao spremnik u koji se tijekom obrade računalnih programa pohranjuju instrukcije, operandi, (među)rezultati obrade...
- Za razliku od programabilnih i izbrisivih ROM memorija, postupak upisa sadržaja nije dugotrajniji od postupka čitanja sadržaja, niti su potrebni posebni programatori za upis novih sadržaja.
- Elementarna memorijska ćelija je složenije građe nego kod ROM memorija, a s obzirom na realizaciju elementarnih memorijskih ćelija razlikujemo dvije izvedbe RAM memorije:
 - **Statičku RAM memoriju**
 - **Dinamičku RAM memoriju.**

STATIČKA RAM MEMORIJA:

- Elementarna memorijska ćelija je sastavljena od jednog **bistabila**, tj. dva unakrsno spojena tranzistora, koji pamti zadano stanje sve dok ne dođe vanjski signal koji zahtijeva promjenu stanja bistabila (0 ili 1).
- Za razliku od dinamičke RAM memorije, nije potrebno periodički osvježavati upisane podatke.
- Nisu potrebni dodatni sklopovi za osvježavanje memorije, ali je zato za realizaciju jedne elementarne ćelije potrebno više tranzistora nego kod dinamičkih RAM memorija.
- Memorije manjih kapaciteta se više isplati realizirati kao statičke memorije – relativno složeni sklop za osvježavanje naboja se ne isplati ugrađivati u malene memorije.
- Statičke RAM memorije se izrađuju i u bipolarnoj i unipolarnoj (MOS) tehnologiji.

DINAMIČKA RAM MEMORIJA:

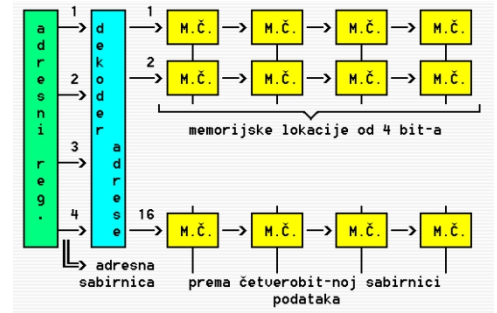
- Elementarna ćelija koja pamti jedan bit se temelji na postojanju električnog naboja na **kondenzatoru**.
- Dotični naboj se postepeno izbija, pa se, da ne bi došlo do gubitka podataka, naboj mora periodički obnavljati.
- Memorije većih kapaciteta se više isplati realizirati kao dinamičke memorije – iako je potreban relativno složen sklop za osvježavanje naboja, njegov "trošak" se kompenzira daleko manjim brojem tranzistora potrebnih za izgradnju velikog broja elementarnih mem. ćelija.
- Dinamičke RAM memorije se izrađuju u unipolarnoj (MOS) tehnologiji.

3.20.2. Organizacija unutarnjih memorija

- Pamćenje podataka u memoriji se uvijek svodi na pamćenje binarnih 0 i 1, a jedan bit podatka se pamti u elementarnoj memorijskoj ćeliji.
- Memorije moraju pamtit i bitove povezane u bajtove i riječi.
- Da bi se to ostvarilo, memorijske ćelije mogu biti povezane na različite načine – oblik povezivanja elementarnih memorijskih ćelija u bajtove i riječi nazivamo organizacijom memorije.
- Razlikujemo tri tipične organizacije memorije:
 - **Dvodimenzionalnu**
 - **Trodimenzionalnu**
 - **Memoriju organiziranu u stak.**

3.20.2.1. Dvodimenzionalna organizacija

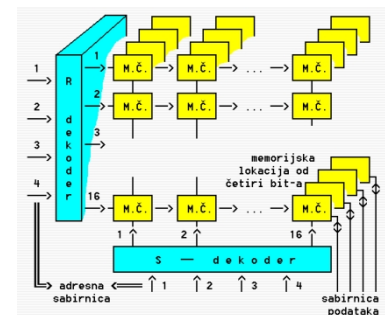
- Memorijski elementi su poredani u pravokutnu matricu, jedan do drugog i jedan ispod drugog, tako da, gledajući horizontalno tvore jednu riječ (jednu memorijsku lokaciju), a gledajući vertikalno predstavljaju skup bitova iste težine.
- Broj redaka određuje broj (kapacitet) riječi koje je moguće pohraniti u tako organiziranu memoriju.
- Broj stupaca određuje širinu riječi koju je moguće pohraniti.
- Postupak adresiranja jedne memorijske lokacije možemo opisati ovako:
 - Adresnom sabirnicom širine N bitova dolazi u memorijski sklop adresa retka (lokacije) u memoriji.
 - Ta se adresa pohranjuje u adresni registar, koji je također širok N bitova.
 - Iz adresnog registra se N-bitna adresa (binarni broj) šalje na adresni dekodler (tipa "s N na 2^N "), koji jednoznačno odredi željeni redak memorije i aktivira vod za selekciju tog retka.
 - Aktivirani redak sudjeluje u akciji čitanja ili pisanja.
 - Kod čitanja se sadržaj adresiranog retka prenosi na podatkovni registar u memoriji, a sa njega se prenosi na podatkovnu sabirnicu.
 - Kod pisanja se sadržaj podatkovnog registra (pristigao je podatkovnom sabirnicom) upisuje u adresiranu lokaciju.
 - Nedostatak ove organizacije se očituje u velikom broju vodova koji su potrebni za realizaciju memorije – za N-bitno adresno polje potrebno je 2^N selektorskih vodova, kao i adresni dekodler sa mnogo izlaza (tipa "s N na 2^N ").
 - Problem je to veći što se više povećava kapacitet memorije.



SLIKA 3.25: 2D ORGANIZACIJA MEMORIJE

3.20.2.2. Trodimenzionalna organizacija

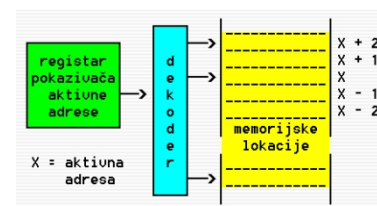
- Jedan od načina smanjivanja broja selektorskih vodova je trodimenzionalna organizacija memorije.
- To je organizacija kod koje se selekcija pojedine memorijske ćelije ne izvrši do kraja u posebnom vanjskom adresnom dekoderu, već se dio selekcije izvodi i u samoj memorijskoj ćeliji.
- Najjednostavniji način takvog adresiranja pojedinih ćelija memorije je kad se adresni registar i dekodler podijele na **dvije polovice**, jedna za horizontalno adresiranje, a druga za vertikalno.
- Svaka polovica selektira po jednu adresnu liniju u skladu s podatkom koji je dekodiran na odgovarajućoj horizontalnoj, odnosno vertikalnoj polovici adresnog dekodera.
- Adresira se samo ona memorijska ćelija koja je na presjicištu odabrane horizontalne i vertikalne linije.
- Dakle, ako se uzme polovica bitova adresnog registra za adresiranje X (vertikalnog), odnosno Y (horizontalnog) smjerova memorije, postojat će $2^{N/2}$ vodova za horizontalnu i $2^{N/2}$ vodova za vertikalnu selekciju, odnosno ukupno $2 * 2^{N/2}$ vodova.
- Npr. pri adresnom registru od 8 bitova postojat će 32 voda ($2 * 2^{8/2} = 2 * 16 = 32$). Usporedbe radi, 2D organizirana memorija s jednako velikim adresnim registrom bi zahtijevala $2^N = 2^8 = 256$ vodova.
- Nedostatak ovakve organizacije je taj da se u jednoj ravnini trodimenzionalne memorije adresira samo jedna memorijska ćelija – ona predstavlja jedan bit adresirane riječi.
- Da bi dobili više bitova u svim riječima, treba upotrebljavati onoliko ravnina memorije koliko su duge riječi koje treba pamtiti.
- Duljina riječi predstavlja treću dimenziju memorije, a bitovi jedne memorijske riječi nalaze se jedan ispod drugoga u raznim ravninama memorije.



SLIKA 3.26: 3D ORGANIZACIJA MEMORIJE

3.20.2.3. Memorija organizirana u stack

- 2D i 3D organizacija omogućavaju izravan pristup do svakog podatka – pogodne su za organizaciju RAM i ROM memorija.
- Osim takvih organizacija, postoje i organizacije kod kojih je pristup podacima omogućen samo odgovarajućim redosljedom – primjer je memorija organizirana u stak.
- Osnovne primjene stak memorije su:
 - Obrada programskih prekida
 - Pozivi potprograma
 - Rekurzivni pozivi potprograma.



SLIKA 3.27: MEMORIJA ORGANIZIRANA U STACK

PRINCIP RADA MEMORIJE ORGANIZIRANE U STAK:

- Stak memorije rade po LIFO (Last In First Out) principu – obrađuje se samo onaj podatak koji je zadnji upisan u stak, tj. podatak koji se nalazi **na vrhu** staka.
- Moguće su sljedeće operacije koje rade isključivo sa podatkom na vrhu stak memorije:
 - **Umetanje (push)** – potiskuje ostale podatke na staku prema dolje.
 - **Vađenje (pop)** – čita podatak i uklanja ga sa staka, pri čemu se ostali podaci pomiču za jedno mjesto prema gore.
 - **Čitanje** – samo čita podatak na vrhu staka, bez uklanjanja – svi ostali podaci u staku ostaju na istim položajima.
 - **Upisivanje** – samo prepisuje vrh staka novim sadržajem – svi ostali podaci u staku ostaju na istim položajima.

MOGUĆNOSTI REALIZACIJE STAK MEMORIJA:

- Stak memorija se sastoji od niza registara, od kojih je svaki jednake širine.
- Niz takvih registara poredanih jedan poverh drugog tvore stak memoriju
- Stak je moguće realizirati na 2 temeljna načina:
 - Unutar procesora, pomoću niza šift-registara
 - Unutar radne memorije, uz pomoć SP registra.
- **Realizacija pomoću šiftera:**
 - Za stak memoriju širine N bita trebamo N šift registara.
 - "Dubina" takve stak memorije ovisi o duljini šiftera – ako su šifteri M-bitni i ima ih N, tada stak može pohraniti M riječi širokih N bitova.
 - Bitovi n-tog težinskog mjesta u svakoj od M riječi su zapisani u n-tom od ukupno N šiftera.
 - m-ti bit svakog od ukupno M šiftera predstavlja jednu strojnu riječ pohranjenu u stak.
 - operacije sa stakom se svode na elementarne operacije šiftera – posmak ulijevo i udesno.
 - nedostatak: ograničena dubina zbog smještaja u procesoru.
 - prednost: brzina pristupa zbog smještaja u procesoru.
- **Realizacija pomoću SP (Stack Pointer) registra:**
 - Ova varijanta stak memorije se realizira u RAM memoriji korištenjem određenog broja memorijskih lokacija.
 - Dubina staka ovisi o broju lokacija koju ćemo odvojiti za stak – teoretski, mogli bi zauzeti i čitavu RAM memoriju.
 - Ne treba pretjerivati, jer se time smanjuje kapacitet radne memorije dostupan za ostale zadatke.
 - Kako u RAM memoriji nema šift registara, operacije nad stakom nije moguće realizirati pomicanjem podataka, već prividnim pomicanjem vrha staka.
 - Taj pomak se ostvaruje promjenom adrese koja pokazuje na vrh staka, a ta adresa je zapisana u SP registru u procesoru.
 - Pri svakoj push ili pop operaciji se poveća ili umanja sadržaj SP registra, koji na taj način uvijek pokazuje na vrh staka.
 - Adresa smještena u SP se još naziva i **aktivna adresa**.

3.20.3. Načini adresiranja unutarnjih memorija

- Do podataka zapisanih u ROM ili RAM memoriji je moguće doći na razne načine, tj. mogući su različiti oblici adresiranja memorijskih lokacija.
- Primjeri opisani u narednim podlekcijama opisuju neke od tipičnih načina na koje je moguće dohvatiti operande potrebne za izvođenje trenutne instrukcije.
- Bitno je naglasiti da je način adresiranja memorije (tj. način dohvata operanada) uvijek zadan unutar same instrukcije koja se trenutno izvršava, i to unutar operacijskog koda instrukcije.
- Upravljačka jedinica tumači operacijski kôd i na temelju njega prepoznaje zatraženi oblik adresiranja memorije, te aktivira upravljačke signale potrebne za realizaciju zatraženog oblika adresiranja.

3.20.3.1. Neposredno adresiranje

- Oblik adresiranja kod kojeg je operand potreban za izvođenje instrukcije dio same instrukcije.
- Prvi bajt instrukcije predstavlja operacijski kod (kojim je u ovom slučaju zadano neposredno (Immediate) adresiranje).
- Drugi i treći bajt (ako je operand 16-bitna riječ) instrukcije sadrže traženi operand.
- Drugi bajt je bajt manje težinske vrijednosti, a treći je veće težinske vrijednosti.
- Prednost – operand je sastavni dio instrukcije, pa je potreban samo jedan zahvat u memoriju za operandom.

Primjer:

- PC pokazuje na adresu 1BF3.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o NEPOSREDNOM adresiranju.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi operand.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

3.20.3.2. Direktno adresiranje

- Oblik adresiranja kod kojeg je adresa operanda potrebnog za izvođenje instrukcije dio same instrukcije.
- Prvi bajt instrukcije predstavlja operacijski kod (kojim je u ovom slučaju zadano direktno adresiranje).
- Drugi i treći bajt instrukcije sadrže (u ovom slučaju 16-bitnu) adresu traženog operanda.
- Drugi bajt je bajt manje težinske vrijednosti, a treći je veće težinske vrijednosti.

Primjer:

- PC pokazuje na adresu 02F1.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o DIREKTNOM adresiranju.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi manje značajan bajt prave adrese operanda. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- Sadržaj PC-a se opet poveća za 1 i PC sada pokazuje na treću adresu. Na toj adresi se nalazi više značajan bajt prave adrese operanda. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- PC tek sada pokazuje na stvarnu adresu operanda.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

3.20.3.3. Indirektno adresiranje

- Oblik adresiranja kod kojeg je adresa adrese operanda potrebnog za izvođenje instrukcije dio same instrukcije.
- Prvi bajt instrukcije predstavlja operacijski kôd (kojim je u ovom slučaju zadano indirektno adresiranje).
- Drugi i treći bajt instrukcije sadrže (u ovom slučaju 16-bitnu) adresu adrese traženog operanda.
- Drugi bajt je bajt manje težinske vrijednosti, a treći je veće težinske vrijednosti.
- Na lokaciji koja sadrži adresu adrese operanda se nalazi stvarna adresa operanda.
- Prvi bajt je bajt manje težinske vrijednosti, a drugi je veće težinske vrijednosti.

Primjer:

- PC pokazuje na adresu 00C2.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o INDIREKTNOM adresiranju.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi adresa lokacije na kojoj se nalazi manje značajan bajt adrese pokazivača. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- Sadržaj PC-a se opet poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi adresa lokacije na kojoj se nalazi više značajan bajt adrese pokazivača. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- Sada se u CPU-u spoje manje značajan i više značajan bajt iz prethodna dva koraka. Oni tvore adresu pokazivača.
- PC sada pokazuje na tu adresu. Na toj adresi se nalazi manje značajan bajt adrese operanda. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- Sadržaj PC-a se opet poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi više značajan bajt adrese operanda. Taj bajt se šalje preko podatkovne sabirnice u CPU i tamo se spremi.
- Sada se u CPU-u spoje manje značajan i više značajan bajt iz prethodna dva koraka. Oni tvore adresu operanda.
- PC tek sada pokazuje na stvarnu adresu operanda.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

3.20.3.4. Relativno adresiranje s obzirom na PC

- Oblik adresiranja kod kojeg se adresa operanda odabire u određenom adresnom području u blizini lokacije same instrukcije.
- Prvi bajt instrukcije predstavlja operacijski kod (kojim je u ovom slučaju zadano relativno adresiranje prema PC).
- Drugi bajt instrukcije sadrži vrijednost koja se interpretira kao **pomak** (zabilježen u aritmetici dvojnog komplementa) koji se pribraja adresi zapisanoj u PC.
- Pomak je u ovom slučaju 8-bitna vrijednost, zapisana u aritmetici dvojnog komplementa, što daje pomak sljedećeg raspona:.
- Pozitivni pomak od 0 do 127 – vrijednost se pribraja adresi u PC.
- Negativni pomak od -1 do 128 – vrijednost se oduzima od adrese u PC.

Primjer:

- PC pokazuje na adresu 02F5.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o RELATIVNOM adresiranju S OBZIROM NA PC.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi pomak (offset) u odnosu na PC. Taj offset se šalje preko podatkovne sabirnice u CPU i tamo se zbroji sa sadržajem PC-a.
- PC sada pokazuje na stvarnu adresu operanda.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

3.20.3.5. Adresiranje registara i pomoću registara

- Podrazumijeva se adresiranje registara opće namjene.
- Ima više vrsta adresiranja registara i pomoću registara:

ADRESIRANJE REGISTARA:

- Oblik adresiranja kod kojeg se operand preuzima direktno iz nekog od registara opće namjene.
- Ovo je **najbrži oblik adresiranja**, jer se operand već nalazi unutar procesora i nije potrebno dodatno zahvaćati u sporiju radnu memoriju.
- Operacijskim kodom je zadan oblik adresiranja, ali je zadano i koji registar opće namjene sadrži operand.
- Instrukcija koja zahtijeva adresiranje registara se sastoji od samo jednog bajta – operacijskog koda.

Primjer:

- PC pokazuje na adresu 0100.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o adresiranju REGISTARA.
- U instrukcijskom registru operacijski kôd se podijeli na dva dijela: npr. 0100 | 0101
- Prvi dio (0100) određuje vrstu adresiranja po kojoj će se adresirati memorija.
- Drugim dijelom (0101) adresira se registar u kojemu se nalazi operand.
- Dakle, operand se već nalazi u CPU-u i nije ga potrebno dohvaćati iz memorije.

INDIREKTNO REGISTARSKO ADRESIRANJE:

- Slično je indirektnom adresiranju opisanom u prethodnim lekcijama, uz sljedeće razlike:
- Adresa operanda je smještena u nekom od registara opće namjene.
- Op. kôd instrukcije određuje takav oblik adresiranja i zadaje u kojem registru opće namjene se nalazi adresa operanda.
- Dakle, adresa operanda se ne dohvaća iz radne memorije nego se čita iz registra opće namjene – takav oblik indirektnog adresiranja je brži.
- Zahvatom u memoriju se dohvaća sadržaj lokacije na koju pokazuje registar opće namjene, a koja sadrži traženi operand.

Primjer:

- PC pokazuje na adresu 003A.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o REGISTARSKOM INDIREKTNOM adresiranju.
- U instrukcijskom registru operacijski kôd se podijeli na dva dijela: npr. 0100 | 0101
- Prvi dio (0100) određuje vrstu adresiranja po kojoj će se adresirati memorija.
- Drugim dijelom (0101) adresira se registar u kojemu se nalazi adresa operanda.
- Npr. uzmimo da je u registru R3 zapisana adresa operanda.
- PC preuzima tu adresu od registra R3 i ono sada pokazuje na stvarnu adresu operanda.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

RELATIVNO REGISTARSKO ADRESIRANJE:

- Slično je relativnom adresiranju s obzirom na PC, ali sad je bazična adresa smještena u nekom od registara opće namjene.
- Prvi bajt instrukcije predstavlja operacijski kôd (kojim je u ovom slučaju zadano relativno registarsko adresiranje, kao i registar koji sadrži bazičnu adresu).
- Drugi bajt instrukcije sadrži vrijednost koja se interpretira kao pomak (zabilježen u aritmetici dvojnog komplementa) koji se pribraja adresi zapisanoj u registru opće namjene.
- Prednost u odnosu na relativno adresiranje prema PC: Operand koji se dohvaća ne mora biti smješten u blizini same instrukcije, već može biti smješten bilo gdje u memoriji – samo je prethodno potrebno u registar opće namjene upisati adresu koja (nakon što joj se pribroji ili oduzme pomak) može "dosegnuti" traženi operand.

Primjer:

- PC pokazuje na adresu 1234.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o RELATIVNOM REGISTARSKOM adresiranju.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi pomak (offset) koji treba pribrojiti sadržaju registra R0. Taj offset se šalje preko podatkovne sabirnice u CPU i tamo se zbroji sa sadržajem registra R0.
- Registar R0 sada pokazuje na stvarnu adresu operanda.
- PC preuzima tu adresu i ono također sada pokazuje na stvarnu adresu operanda.
- Operand se na kraju šalje iz memorije preko podatkovne sabirnice u CPU.

3.20.3.6. Indeksirano adresiranje

- Za adresiranje operanada se upotrebljava poseban registar u procesoru – **indeks registar**.
- Ova vrsta adresiranja je pogodna za dohvaćanje većeg broja operanada u nizu koji su smješteni u nizu uzastopnih lokacija u memoriji.
- Adresa prvog operanda u nizu se zapiše u indeks registar.
- Ta se adresa iskoristi za dohvat prvog operanda.
- Nakon dohvata prvog operanda, sadržaj indeks registra se uvećava za zadani pomak (zadan je instrukcijom u njenom drugom bajtu) i sad pokazuje na adresu sljedećeg operanda u nizu.
- Dohvaća se drugi operand.
- Ciklus se ponavlja sve dok se ne pročitaju svi potrebni operandi – tzv. autoindeksirano adresiranje.

Primjer:

- PC pokazuje na adresu 4567.
- PC se spaja na adresnu sabirnicu te u memoriji pokazuje na lokaciju s operacijskim kodom.
- Taj operacijski kôd se pošalje preko podatkovne sabirnice u CPU, gdje se spremi u instrukcijski registar te dekodira u dekoderu.
- U dekoderu sada vidimo da je riječ o INDEKSIRANOM adresiranju.
- Sadržaj PC-a se poveća za 1 i PC sada pokazuje na sljedeću adresu. Na toj adresi se nalazi pomak (offset). Taj offset se šalje preko podatkovne sabirnice u CPU.
- Unutar CPU-a se nalazi indeksni registar.
- U taj indeksni registar je prije adresiranja zapisana adresa prvog operanda.
- PC preuzima sadržaj indeksnog registra i ono sada pokazuje na adresu na kojoj se nalazi prvi operand.
- Taj prvi operand se šalje iz memorije preko podatkovne sabirnice u CPU. Prvi operand je sada spremljen u CPU.
- Sada se sadržaju indeksnog registra pribraja offset. Zbog toga, indeksni registar sada sadrži adresu drugog operanda.
- PC preuzima sadržaj indeksnog registra i ono sada pokazuje na adresu na kojoj se nalazi drugi operand.
- Taj drugi operand se šalje iz memorije preko podatkovne sabirnice u CPU. I drugi operand je sada spremljen u CPU.
- I tako dalje, sve dok se ne dohvate svi željeni operandi.

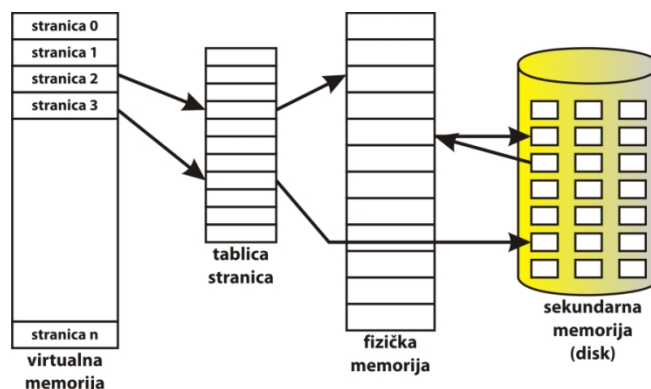
3.20.3.7. Adresiranje po stranicama (straničenje)

- Za ovu vrstu adresiranja je bitno podijeliti adresni prostor na nizove lokacija jednake duljine, koje se zovu **stranice (page)**.
- Npr. stranice su veličine 128 bajtova, tj. svaka lokacija unutar pojedine stranice se može adresirati sa 7 bitova.
- Tada bi 16-bitno adresno polje bilo podijeljeno na 512 stranica i izgledalo bi ovako:

adrese (hex.)	
0000 – 007F	0. stranica
0080 – 00FF	1. stranica
0100 – 017F	2. stranica
0180 – 01FF	3. stranica
...	...
FF00 – FF7F	510. stranica
FF80 – FFFF	511. stranica

- Instrukcija koju treba izvršiti je smještena u nekoj od stranica.
- Za dohvat njenih operanada se primjenjuje adresiranje slično relativnom adresiranju – koriste se pomaci zadani u bajtu nakon operacijskog koda instrukcije.
- Za pomak se koristi 7 bitova 8-bitne riječi i tim pomakom se može adresirati bilo koja lokacija u doseg jedne stranice (7 bitova = 128-bajtni adresni raspon).
- Krajnji lijevi, **osmi bit** pomaka određuje da li će se adresirati 0-ta stranica ili stranica na kojoj je smještena instrukcija.
 - Ako je 8. bit jednak 0, onda se adresira nulta stranica.
 - Ako je 8. biti jednak 1, onda se adresira trenutna stranica na kojoj se nalazi instrukcija.

- Nedostatak ovakvog adresiranja (uz postavke koje smo pretpostavili – 8-bitni pomak, stranice od 128 bajtova) je u tome da je **moгуće adresirati samo 2 stranice: nultu ili trenutno aktivnu**.
- Znači da i operandi za instrukciju moraju obavezno biti smješteni ili na 0-toj stranici ili na istoj stranici kao i instrukcija.
- Iako se ograničenja čine prevelikima, ovaj oblik adresiranja je iznimno značajan za današnja računala, jer se pomoću njega realizira **virtualna memorija**.
- Ovdje uočena ograničenja se rješavaju duljim strojnim riječima i adresama (tipično 32-bitnim), stranicama većeg kapaciteta (tipično oko 4 kB), kao i daleko složenijom organizacijom stranicenja (višerazinske tablice stranica...), tako da je i kroz stranicenje moguće adresirati bilo koji dio memorije.



SLIKA 3.28: PRIKAZ VIRTUALNE MEMORIJE KOJA JE VEĆA OD FIZIČKE MEMORIJE

3.21. Vanjske memorije

- Brza unutarnja radna memorija pamti podatke samo dok računalo radi – po prestanku rada (gašenjem računala) ili iznenadnim nestankom struje, podaci pohranjeni u radnoj memoriji računala se gube.
- Za potrebe trajne pohrane važnih podataka postoje vanjske memorije koje pamte na njih zapisane podatke i u trenucima kad je računalo ugašeno.
- Prednosti vanjskih memorija u odnosu na unutarnju radnu memoriju:
 - Pamte podatke i kad nisu pod el. naponom.
 - Većeg su kapaciteta.
 - Cijena po jedinici kapaciteta je niža.
- Nedostaci vanjskih memorija:
 - Višestruko su sporije od radne memorije (reda veličine 10^5 do 10^6).
 - Zbog prisustva velikog broja mehaničkih dijelova su podložnije kvarovima.

3.21.1. Magnetske vanjske memorije

- Materijalni nositelj zapisa na magnetskim vanjskim memorijama je sloj feromagnetskog materijala kojim je presvučen fizički medij.
- Manipulacijom sa elektro-magnetskim svojstvima tog materijala se ostvaruje zapis i čitanje podataka.
- Magnetske vanjske memorije se dijele na dvije osnovne podskupine:
 - **magnetske diskove**, i
 - **magnetske vrpce**.

MAGNETSKI DISKOVI:

- Njih možemo podijeliti na:
 - **čvrste diskove** (hard disk)
 - **diskete** (floppy disk).
- Građa hard diska:
 - Niz ploča od metala ili stakla je smješteno na zajedničkoj osovini oko koje sve ploče rotiraju.
 - Ploče su presvučene feromagnetskim materijalom.
 - Magnetizacijom feromagnetskog materijala se zapisuju binarni 0 i 1.

- Jedna orijentacija čestica feromag. materijala – binarna 0.
- Druga orijentacija čestica feromag. materijala – binarna 1.
- Zapisivanje i čitanje se izvodi pomoću glava za čitanje/pisanje.
- Sa svake strane svake ploče se nalazi po jedna glava za čitanje/pisanje.
- Pomicanje glava – pomoću aktuatorske ruke, koja je pokretana motorom.
- Glave nikada fizički ne dotiču površinu ploča – lebde nad pločama na mikroskopski debelom zračnom jastuku.
- Najvažniji dio glave za čitanje/pisanje predstavlja **elektromagnet**.
- Propuštanjem struje kroz zavojnicu elektromagneta se inducira magn. polje koje orijentira česticu feromagnetskog materijala u određenom smjeru – zapis binarne 0 ili 1.
- Smjer orijentacije ovisi o smjeru propuštanja struje kroz zavojnicu elektromagneta.
- Prelaskom glave preko magnetizirane čestice se u zavojnici elektromagneta inducira el. struja – čitanje binarne 0 ili 1.
- Smjer struje (pročitana binarna 0 ili binarna 1) ovisi o orijentaciji magn. polja čestice na podlozi.



SLIKA 3.29: GLAVNI ELEMENTI HARD DISKA

- Građa floppy diska:

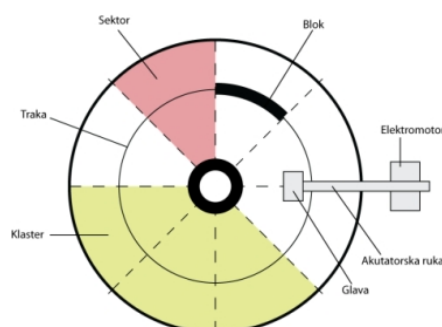
- Dijeli mnogo zajedničkih osobina sa građom hard diska, a razlike su:
- Disketa je izmjenjiv medij – u plastičnoj košuljci se nalazi **samo jedna ploča** od savitljivog materijala presvučena feromagnetskim materijalom.
- Gustoća zapisa je manja nego kod hard diska.
- Zbog samo jedne ploče i manje gustoće zapisa proizilazi da je kapacitet diskete daleko manji.
- U fiksiranom pogonu floppy diska je smješten pogonski motor koji rotira ploču (višestruko sporije nego kod hard diska), po jedna glava za čitanje/pisanje sa svake strane ploče, te aktuatorska ruka koja pokreće glavu.



SLIKA 3.30: DISKETA

- Fizička organizacija zapisa:

- Elementi fizičke organizacije zapisa:
 - **ploče**
 - **staze** – koncentrični krugovi na pločama.
 - **cilindri** – sve staze na svim pločama koje su jednako udaljene od središta rotacije ploča.
 - **sektori** – isječci staza – svaka staza je podijeljena na jednak broj sektora; sektor je najmanja adresibilna jedinica na disku.
 - **klasteri** – logička cjelina koja obuhvaća nekoliko susjednih sektora; u slučaju da se kod fizičke organizacije zapisa koriste i klasteri, tada je jedan klaster najmanja adresibilna jedinica na disku.

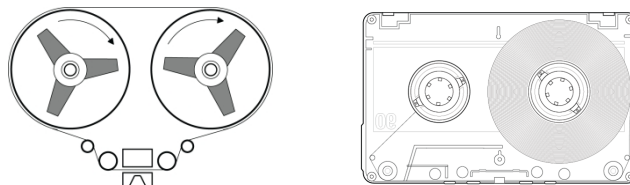


SLIKA 3.31: PRIKAZ GLAVNIH DIJELOVA MAGNETSKIH DISKOVA

- Logička organizacija zapisa:
 - Iz načina fizičke organizacije zapisa na magn. diskovima (tj. iz mogućnosti izravnog pristupa do traženog sadržaja) proizilaze slijedeće moguće logičke organizacije zapisa:
 - **SLIJEDNA ORGANIZACIJA ZAPISA**
 - Potječe sa magnetskih vrpce, kod kojih je to i jedina moguća organizacija zapisa.
 - Podaci se zapisuju u kontinuirane (slijedne, sekvencijalne) cjeline – sektor do sektora, klaster do klastera.
 - **RASUTA ORGANIZACIJA ZAPISA**
 - Omogućava dislocirano zapisivanje podataka po mediju.
 - Mogućnost iskorištenja svakog slobodnog prostora na mediju.
 - Veća fleksibilnost.
 - Obje spomenute organizacije zapisa su podržane kroz datotečni sustav.

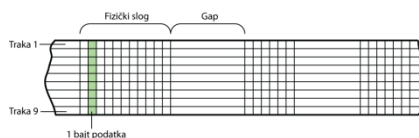
MAGNETSKE VRPCE:

- Danas se uglavnom koriste kao masovni medij za sigurnosnu pohranu podataka (backup).
- Vrpca je izrađena od savitljivog plastičnog materijala i jedne strane je presvučena sa feromagnetskim materijalom .
- Dva su osnovna oblika magnetskih vrpce:
 - na kolutovima
 - u kazetama.



SLIKA 3.32: MAGNETSKA VRPCE NA KOLUTOVIMA I U KAZETI

- Fizička organizacija zapisa kod vrpce na kolutovima:
 - Vrpca je uzdužno podijeljena na 9 traka, čime je na jednu lokaciju (poprečno po vrpce) moguće pohraniti 8 bitova podatka + 1 paritetni bit.
 - Jedna lokacija se naziva Frame.
 - Za svaku traku postoji zasebna fiksna glava za čitanje/pisanje, pri čemu se paralelo čita/zapisuje svih 9 traka.
 - Više frameova čini blok podataka (fizički slog).
 - Način zapisa je slijedeći: Fizički slog – gap – fizički slog – gap – fizički slog...



SLIKA 3.33: FIZIČKA ORGANIZACIJA ZAPISA KOD VRPCE NA KOLUTOVIMA

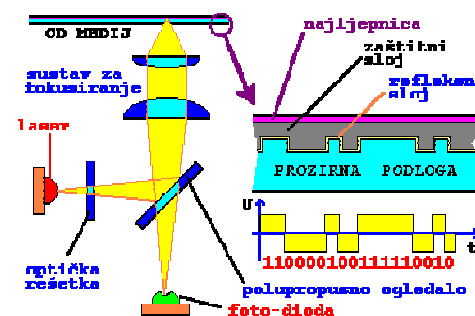
- Fizička organizacija zapisa kod vrpce u kazetama:
 - Vrpca podijeljena uzdužno na trake, ali na daleko veći broj traka – od 20 do 144.
 - Postoji samo jedna glava za čitanje/pisanje.
 - Podaci se zapisuju uzdužno niz traku.
 - Nakon što se dođe do kraja trake, mehanizam mijenja smjer vrtnje vrpce, a glava prelazi na narednu traku itd.
- Logička organizacija zapisa na magnetskim vrpce:
 - Magnetske vrpce koriste isključivo slijedni pristup do pohranjenih sadržaja.
 - Moguća je jedino slijedna logička organizacija zapisa.
 - Podaci se moraju čitati blok po blok, onim redom kako su bili i zapisani.

3.21.2. Optičke vanjske memorije

- Optičke vanjske memorije se ubrajaju u skupinu izmjenivih vanjskih memorija sa prenosivim medijima (poput disketa ili magnetskih vrpca, koje smo opisali u prethodnoj lekciji).
- Prema kriteriju da li ih se može samo čitati ili čitati i pisati, optički mediji se dijele u dvije skupine:
 - mediji koje služe samo za čitanje: CD-ROM i DVD-ROM.
 - mediji koji služe i za čitanje i za pisanje:
 - mediji za jednokratno zapisivanje: CD-R, DVD-R, DVD+R.
 - mediji višekratno zapisivanje (izbrisivi mediji): CD-RW, DVD-RW, DVD+RW, DVD-RAM.

1

- Građa CD-ROM medija:
 - Medij ima samo jednu ploču sa koje se čitaju podaci.
 - Za smještaj podataka je iskorištena samo jedna strana te ploče.
 - Materijalni nositelj je metalni reflektivni sloj na koji je tvornički utisnuta "matrica" sa udubljenjima (ili izbočenjima) i ravnim dijelovima.
 - Udubljenje (izbočenje) – binarna 0, ravnina – binarna 1.
 - Čitanje se izvodi pomoću laserske zrake koju emitira optička glava. Optička glava se sastoji od:
 - Laserske diode – izvora laserske zrake (crvene),
 - prizme, leće za fokusiranje i detektora svjetlosti.



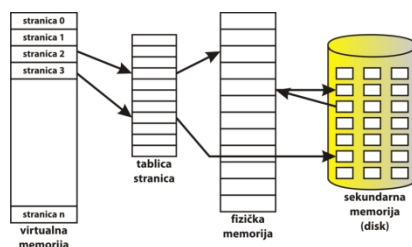
SLIKA 3.34: NAJČEŠĆE UPOTREBLJIVANA CD-ROM-A

- Fizička organizacija zapisa na CD-ROM mediju:
 - Podaci su zapisani u kontinuiranu spiralu koja počinje u središtu diska.
 - Spirala je podijeljena na blokove fiksne veličine od 2352 bajta.
 - Gustoća zapisa se povećava prema rubu medija.
- Građa DVD-ROM medija
 - Po građi je gotovo identičan CD-ROM mediju.
 - Laserska zraka kojom se obavlja čitanje je kraće valne duljine, pa je moguće čitanje "sitnijih" udubina i ravnina.
 - veća gustoća zapisa podataka na mediju iste veličine kao i CD-ROM.
- Osim veće gustoće zapisa, DVD diskovi mogu biti:
 - **dvoslojni** – na istoj strani diska jedan iznad drugoga dva sloja za zapisivanje podataka – kapacitet jedne strane DVD medija se gotovo udvostručava (8,5 GB).
 - **dvostrani** – moguće je zapisivati podatke i na drugoj strani medija – jedan medij može imati do 4 sloja za zapis (2 sa svake strane) – kapacitet medija tada iznosi oko 17 GB.
- Optički mediji koji služe za čitanje i pisanje:
 - Razlikuju se od medija koji služe samo za čitanje po vrsti materijalnog nositelja zapisa – sloj za zapisivanje nije metalni, već se sastoji od organskih boja i sličnih materijala.
 - Sloj za zapisivanje kod medija za jednokratni zapis – sloj od organske boje koja pod utjecajem snažne laserske zrake mijenja svojstva.
 - Sloj za zapisivanje kod medija za višestruki zapis – sloj od posebnog materijala koji se pod utjecajem srednje snažne laserske zrake vraća u svoje prvobitno stanje (brisanje zapisa) ili koji pod utjecajem snažne laserske zrake mijenja stanje (zapisivanje).
 - Mjesta koja su izmjenjena pod utjecajem lasera za zapisivanje slabije reflektiraju lasersku zraku prilikom čitanja, pa se po intenzitetu reflektirane zrake određuje da li je zapisana binarna 0 ili binarna 1.
- Fizička organizacija zapisa na (re)writable medijima:
 - Podaci nisu zapisani u kontinuiranu spiralu, nego u koncentrične kružnice, kao kod magnetskih diskova.
 - Staze su podijeljene na sektore.

- Logička organizacija zapisa na optičkim medijima
 - Optičke vanjske memorije spadaju u skupinu uređaja sa izravnim pristupom do pohranjenih podataka (poput magnetskih diskova).
 - Moguće su i slijedna i rasuta logička organizacija zapisa, kao i kod magnetskih diskova.

3.22. Virtualna memorija

- Svaki program prilikom izvođenja mora biti smješten u radnoj memoriji računala.
- Procesor dohvaća instrukcije i podatke iz radne memorije i pohranjuje rezultate u radnu memoriju.
- Programi su trajno pohranjeni na vanjskim memorijama (diskovima) i učitavaju se u radnu memoriju po potrebi.
- Problem nastaje kad treba **istovremeno** izvoditi veći broj programa.
- Iako današnja računala imaju relativno velike radne memorije (oko 1 GB), programi postaju sve veći i memorijski zahtjevniji.
- Fizički kapacitet radne memorije može biti nedostatan za pohranu svih programa koji se trenutno izvršavaju i njihovih radnih podataka.
- Problem se može riješiti tako da se u radnu memoriju ne sprema cijeli program i svi njegovi podaci, već samo oni njegovi dijelovi koji su trenutačno aktivni – neaktivni ostatak je i dalje pohranjen na disku.
- Posebni mehanizmi, poduprti sklopovljem suvremenih procesora, premještaju s diska u radnu memoriju one dijelove programa koji će se upravo izvoditi, a iz radne memorije na disk vraćaju dijelove koji trenutačno nisu potrebni.
- Na disku je potrebno odvojiti poseban prostor za smještaj neaktivnih dijelova programa i podataka (tzv. *swap file* ili *page file*).
- Diskovi su nekoliko redova veličine većeg kapaciteta od radne memorije (10 do 100 i više puta).
- Na taj način se kapacitet jednog manjeg fizičkog spremnika (radne memorije) može iskoristiti za potporu istovremenog izvršavanja većeg broja programa čija zajednička veličina premašuje kapacitet radne memorije.



SLIKA 3.35: PRIKAZ VIRTUALNE MEMORIJE KOJA JE VEĆA OD FIZIČKE MEMORIJE

- Virtualna memorija je sustav u kojem je radni memorijski prostor računala načinjen dijelom od brze radne memorije, a dijelom od sporih masovnih memorija.
- U trenutku kada program (proces) želi pristupiti određenoj memorijskoj lokaciji, sklopovska podrška određuje da li se ona fizički nalazi u glavnoj memoriji, a ako ne, pokreće se prijenos s vanjske memorije u radnu.
- Ne prenosi se samo sadržaj tražene lokacije, već cijela stranica u kojoj je smještena tražena lokacija.
- Zbog tih razloga cjelokupni se adresni prostor dijeli na okvire (frame), a svi programi i podaci se dijele na blokove podataka jednake veličine (stranice).
- Kad su za izvođenje operacija potrebni podaci koji se ne nalaze u radnoj memoriji tada se iz vanjske memorije (obično tvrdog diska) prebacuju (proces se naziva swapping) stanice u radnu memoriju.
- U radnoj memoriji se tako nalaze samo oni podaci koji su trenutno potrebni ili aktivni.
- Na tom se principu zasniva *multiprogramski način rada*.
- Radi toga što je samo dio procesa u radnoj memoriji, a ostali koji nisu trenutno potrebni na sekundarnoj može se znatno proširiti raspoloživi memorijski prostor, odnosno taj se prostor naziva virtualnom memorijom.

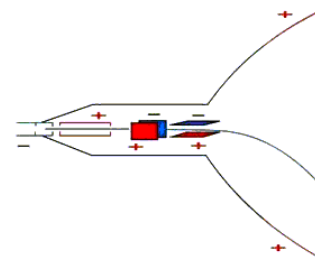
- Ovakav postupak korištenja memorije naziva se **straničenjem** (paging) i ključan je za realizaciju virtualne memorije.
- Informacije o adresama stranica u virtualnoj memoriji pohranjene su u tablice (**tabela stranica**) u kojima se vodi evidencija u kojem je okviru smještena koja stranica.
- U toj su tabeli upisani svi aktivni procesi i adrese njihovih okvira.
- Na temelju zatražene obrade pristupa se tablici stranica, a nakon izračunate stvarne adrese stranice na vanjskoj memoriji započinje proces prijenosa stranice u okvir radne memorije.
- Operacijski sustav (OS) računala upravlja razmjenu informacija između unutrašnje i vanjske memorije, dok programer umjesto stvarnih fizičkih adresa koristi virtualne – programer "vidi" adresni prostor računala kao linearni niz adresa, koji nije ograničen kapacitetom radne memorije, tj. stvoren je "privid" beskonačno velike memorije ili virtualni adresni prostor – odatle i naziv "virtualna memorija".
- Računalo u suradnji s OS-om automatski pretvara virtualne adrese u stvarne.
- Tako programer ne mora brinuti o kapacitetu glavne memorije, već koncipira program prema programskom zadatku.
- Nedostatak virtualne memorije je u tome što se troši određeno vrijeme za prebacivanje stranica iz sekundarne (vanjske) u primarnu (radnu) memoriju, opterećuju se ulazno/izlazni kanali, a sve to utječe na brzinu rada sistema.
- Unatoč tome, ovakav način korištenja memorijskog prostora je dominantan u današnjim računalima.

3.23. Monitor

- Najčešće korištena izlazna jedinica je monitor, a prikazuje sliku na ekranu.
- Prema tehnologiji koja se koristi za prikaz slike na ekranu, monitori se dijele u dvije velike skupine:
 - **monitori sa katodnom cijevi** (CRT)
 - **flat-panel monitori** (ekrani sa tekućim kristalima, plazma ekrani, OLED ekrani...).

3.23.1. Princip rada monitora s katodnom cijevi (CRT)

- Ova tehnologija izrade monitora je najstarija od svih prisutnih. Zbog dugogodišnjeg razvoja, monitori tog tipa su:
 - jeftini za proizvodnju
 - mogu prikazati vrlo kvalitetne slike u punom rasponu boja u visokim rezolucijama
 - slika je vidljiva pod bilo kojim kutem gledanja.
- Neke od glavnih mana CRT ekrana su:
 - velika potrošnja energije
 - glomaznost
 - štetna elektromagnetska zračenja
 - tehnologija prikaza slike neizbježno zamara oči
 - osjetljivost na vanjska elektromagnetska polja .
- Osnovu CRT monitora čini katodna cijev, koja služi za prikaz slike.
- Katodna cijev je izrađena u obliku "boce" koja se naglo širi prema dnu.
- U unutrašnjosti cijevi je vakuum.
- Pri vrhu katodne cijevi nalaze se tri elektronska topa, od kojih svaki emitira po jednu zraku elektrona – jednu za crvenu, jednu za plavu i jednu za zelenu boju.
- Topovi se sastoje od:
 - katode
 - izvora topline (grijač)
 - elemenata za fokusiranje zrake
- Pod utjecajem grijača, katoda počinje otpuštati elektrone.
- Zrake elektrona prolaze kroz zrakoprazan prostor u cijevi i pogađaju "dno" cijevi – tzv. ekran.

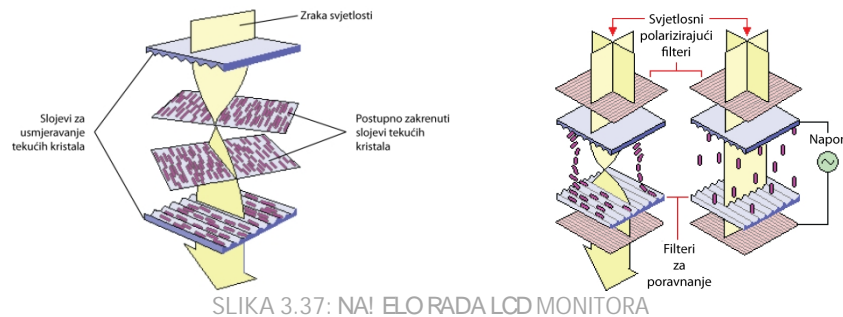


SLIKA 3.36: SKICA KATODNE CIJEVI

- Zrake se usmjeravaju na određeni dio ekrana pomoću dvaju okomito položenih izvora izvora magnetskih polja – tzv. deflektora.
- Jedan deflektor pomiče zrake smjeru gore-dolje, a drugi u smjeru lijevo-desno.
- Kombiniranim djelovanjem oba deflektora je moguće usmjeriti zrake elektrona na bilo koju točku na ekranu.
- Sa unutarnje strane površine ekrana nalazi se fosforni premaz, nanešen u obliku sitnih točaka.
- Po tri fosforne točke različitih kemijskih karakteristika čine jedan piksel na ekranu.
- Za svaki piksel postoji po jedna fosforna točka koja svijetli crveno, jedna koja svijetli plavo i jedna koja svijetli zeleno.
- Fosforne točke zasvijetle kad ih pogodi zraka elektrona.
- Intenzitet svjetlosti koju će emitirati pojedina fosforna točka ovisi o jakosti elektronske zrake koja je pogađa.
- Kombiniranjem (aditivnim miješanjem) različitih intenziteta emitirane crvene, plave i zelene svjetlosti pojedinog piksela postiže se željena boja pojedinog piksela.
- Jednaki maksimalni intenzitet crvene, plave i zelene svjetlosti daje potpuno bijelu točku.
- Odsustvo bilo kakve emitirane svjetlosti daje crnu točku.
- Međuintenziteti pojedine svjetlosti daju sve ostale boje.
- Neposredno iznad same unutarnje površine ekrana nalazi se perforirana maska (tzv. *shadow mask*) sa sitnim rupicama – po jedna za svaki piksel.
- Sve tri zrake prolaze kroz jednu rupicu na maski prije nego što pogode površinu ekrana, čime se zrake precizno fokusiraju na fosforne točke odgovarajuće boje.

3.23.2. Princip rada monitora s tekućim kristalima (LCD)

- Započeli su svoj razvoj za potrebe prijenosnih računala, gdje je bilo potrebno imati ekran čim manjih dimenzija i sa malom potrošnjom energije.
- U današnje vrijeme se LCD monitori sve više koriste i sa stolnim računalima, kao zamjena za klasične CRT monitore.
- Prednosti u odnosu na CRT monitore:
 - manje dimenzije
 - manja potrošnja energije
 - odlično fokusirana slika i uniformne boje po cijeloj površini ekrana
 - nema elektromagnetskih zračenja
 - slika manje zamara oči.
- Nedostaci u odnosu na crt monitore:
 - problem vidljivosti slike pod različitim kutevima gledanja
 - slabiji kontrast i osvjjetljenje slike
 - ograničeni raspon boja koje se mogu prikazati
 - sporije osvježavanje slike
 - postoji samo jedna "idealna", tj. radna rezolucija – na ostalim rezolucijama slika nije jednako kvalitetna
 - skuplji su za proizvodnju.
- LCD ekran se sastoji od matrice elemenata baziranih na tekućim kristalima.
- Svaki element te matrice predstavlja jednu točku na ekranu.
- Slika se stvara na temelju konstantnog bijelog pozadinskog svjetla, koje prolazi kroz matricu sa LCD elementima.
- Pojedini LCD element može propustiti ili blokirati prolazak pozadinskog osvjjetljenja, čime je pojedinu točku na ekranu moguće "upaliti" ili "ugasiti".
- Građa LCD elementa:
 - s prednje i stražnje strane LCD elemeta nalaze se polarizirajući filteri.
 - jedan propušta svjetlo u horizontalnom smjeru, a drugi u vertikalnom.
 - između polarizirajućih filtera nalaze se dva posebno obrađena sloja koji služe za poravnavanje molekula tekućih kristala.
 - slojevi su okrenuti jedan prema drugom pod kutem od 90 stupnjeva.
 - zahvaljujući tome se tekući kristali, smješteni između ta dva sloja, poravnavaju u obliku spirale (**heliksa**).
 - LCD elementi su dodatno zaštićeni gornjim i donjim pokrovnim staklom.

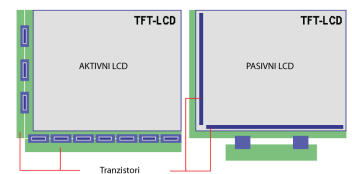


SLIKA 3.37: NA! ELO RADA LCD MONITORA

- "upaljena" točka:
 - Prvi polarizirajući filter propušta samo vertikalnu komponentu svjetlosti.
 - Prolaskom kroz tekuće kristale svjetlost se zakreće u smjeru orijentacije molekula tekućih kristala.
 - Zahvaljujući postupnom spiralnom zakretanju molekula tekućih kristala između dva sloja, i zraka svjetlosti se isto tako postupno zakreće za 90 stupnjeva.
 - Zbog toga zraka svjetla pada pod ispravnim kutem na drugi polarizator, pa zraka prolazi kroz LCD element i ta točka je "upaljena".
 - Kontrolom jačine napona se kontrolira količina propuštene svjetlosti kroz izlazni polarizator, pa neka točka može biti manje ili više "upaljena".

- "ugašena" točka:
 - Molekule tekućih kristala pod utjecajem električnog napona mijenjaju orijentaciju.
 - Iz spiralnog oblika će pod utjecajem napona prijeći u linearni oblik.
 - U tom slučaju zraka svjetlosti koja je prošla kroz prvi polarizator neće promijeniti svoju orijentaciju, pa će pod neizmjenjenim kutem pasti na drugi polarizator.
 - Kako je on okrenut okomito u odnosu na prvi polarizator, neće propustiti zraku svjetlosti i takva će točka na ekranu biti "ugašena".
 - Stupanj "izravnjanja" orijentacije molekula ovisi o veličini napona.

- Niz LCD elemenata poslaganih u obliku matrice na ploču čini osnovu LCD ekrana.
- Ovakav LCD ekran je u stanju prikazati samo jednobojnu sliku.
- Za prikaz slike u boji potrebno je grupirati po **tri** LCD elementa za svaku točku na ekranu – jedan element će propuštati samo crveni dio spektra, drugi samo zeleni, a treći samo plavi dio spektra.
- To se postiže umetanjem dodatnog filtera za boju između polarizatora.
- Slika se iscrta uključivanjem ili isključivanjem svakog pojedinog LCD elementa – ovisno o tome kako je riješen taj problem, razlikuju se dvije osnovne vrste LCD ekrana:
 - **pasivni**
 - **aktivni**.



SLIKA 3.38: AKTIVNI I PASIVNI LCD EKRAN

PASIVNI LCD EKRANI:

- LCD elementi poslagani u matricu su spojeni na vodiče.
- Za svaki redak i svaki stupac matrice postoji po jedan vodič.
- Propuštanjem struje kroz točno određeni red i stupac "gasi" se jedan LCD element.
- Propuštanje struje u vodiče se kontrolira pomoću tranzistora.
- Potreban je po jedan tranzistor za svaki red i svaki stupac.
- Slika se iscrta redak po redak.
- Pod naponom se drži vodič za prvi redak, te se istovremeno propusti napon kroz potrebne vodiče po stupcima.
- Prelaskom na idući redak, prethodni gubi napon, pa LCD elementi vrlo kratko ostaju u definiranom stanju – tj. "ugašeni" pikseli vrlo brzo postaju prozirni.
- Zbog brzine postupka iscrtaavanja, ljudsko oko ne registrira taj problem, ali:
 - Kontrast i svjetlina dobivene slike su slabi
 - Slaba je vidljivost slike pod imalo većim kutem gledanja.

AKTIVNI LCD EKRANI:

- Problem pasivnih ekrana je riješen tako da svaki LCD element ima svoj kontrolni tranzistor, koji može propuštati struju toliko dugo koliko je potrebno.
- Time je omogućena slika sa punim kontrastom boja i daleko većim kutem gledanja na ekran.

3.23.3. Rezolucija monitora

- Rezolucija (razlučivost) monitora predstavlja ukupan broj individualnih točaka koje monitor može prikazati.
- Razlučivost se izražava kao umnožak broja točaka po horizontali i vertikali ekrana.
- Tipični CRT monitori mogu prikazivati slike na većem broju razlučivosti.
- Svaki monitor ima maksimalnu razlučivost, koja ovisi o:
 - Veličini katodne cijevi – izražava u inčima kao veličina vidljive dijagonale ekrana.
 - Fizičkoj udaljenosti između fosfornih točaka – tzv. dot pitch – predstavlja udaljenost između dviju susjednih fosfornih točaka istog tipa (npr. dviju susjednih crvenih fosfornih točaka).

3.23.4. Pixel

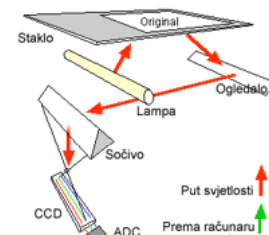
- Pixel (skraćeno od engl. picture element) je jedna točka unutar pravokutne mreže (matrice) koja predstavlja sliku koju treba reproducirati na ekranu ili ispisati na papir.
- Kao što je bit najmanja jedinica za pohranu podataka koju računalo može obrađivati, tako je i pixel najmanja jedinica koja može biti prikazana na ekranu, ispisana na papiru ili s kojom može manipulirati softver i hardver potreban za prikaz i ispis grafike i teksta.
- Fizički pixel je kod CRT monitora u boji realiziran pomoću grupe od 3 fosforne točke osnovnih boja – crvene, plave i zelene.
- Broj fizičkih pixela ovisi o veličini ekrana i o razmaku između istovjetnih elementarnih fosfornih točaka.
- O broju fizičkih pixela ovisi maksimalna razlučivost monitora.
- Taj broj ujedno predstavlja i najveći mogući broj pixela koji mogu biti vjerno prikazani na ekranu.
- Logički pixel je elementarna točka neke slike.
- Slika je definirana konačnim i nepromjenjivim brojem točaka (logičkih pixela).
- Kako monitor može raditi na različitim razlučivostima, tako se i prikaz slike mora prilagođavati razlučivosti monitora ukoliko želimo zadržati identične dimenzije slike na ekranu.
- Iz toga je očito da odnos logičkih i fizičkih pixela nije uvijek 1:1, nego je najčešće 1:N – za jedan logički pixel potrebno je više fizičkih pixela.

3.23.5. Način iscrtavanja slike i frekvencija osvježavanja (CRT ekrani)

- Zrake elektrona putuju po površini ekrana red po red, počevši od gornjeg lijevog ugla.
- Kako zrake prelaze preko pojedinih piksela, tako se (sukladno željenoj boji za pojedini piksel) pojačava/smanjuje intenzitet elektronskih zraka i svaki piksel zasvijetli odgovarajućom bojom.
- Zrake se pomiču horizontalno do kraja reda, a zatim se elektronski top brzo usmjerava na početak novog retka.
- Isti se princip primjenjuje sve do posljednjeg retka na ekranu, čime je završeno iscrtavanje cijele slike.
- Elektronski top opet usmjerava na vrh ekrana i postupak počinje iz početka.
- Frekvencija osvježavanja ekrana:
 - **Broj iscrtanih slika u sekundi** naziva se vertikalnom frekvencijom osvježavanja slike.
 - Radi što manjeg zamaranja očiju, potrebno je da vertikalna frekvencija iznosi barem 85 Hz, tj. da se slika obnavlja barem 85 puta u sekundi.

3.24. Skener

- Skeneri su ulazni uređaji koji se koriste za digitalizaciju i unos u računalo podataka pohranjenih na papirnatim ili sličnim analognim medijima.
- Princip rada svih vrsta skenera je vrlo sličan radu fotokopirnog stoja:
- Postoji više kategorija skenera, a najčešće ih se dijeli na:
 - **stolne** (flatbed) skenere
 - **ručne** (handheld) skenere
 - **protočne** skenere
 - skenere za **posebne** namjene.



SLIKA 3.39: GLAVNI DIJELOVI SKENERA

3.24.1. Princip rada skenera

- Medij kojeg treba skenirati se polaže na ravnu staklenu površinu na vrhu skenera.
- U kućištu skenera se nalazi pomični izvor usmjerene konstantne bijele svjetlosti, koji se u koracima pomiče u vertikalnom smjeru .
- Time se medij postupno obasjava liniju po liniju.
- Svjetliji dijelovi medija kojeg se skenira reflektiraju više svjetla od tamnijih dijelova medija.
- Reflektirana svjetlost se kroz sustav prizmi, zrcala i leća usmjerava na niz fotoosjetljivih senzora (tzv. glavu za skeniranje).
- Kod skenera u boji, reflektirana svjetlost se prvo propušta kroz crveni, plavi i zeleni filter, a zatim se kroz sustav leća svaka komponenta svjetlosti usmjerava na odgovarajući dio površine fotoosjetljivih senzora.
- Kod stolnih skenera se kao senzori najčešće koriste tzv. **CCD** (Charge-Coupled Device) senzori
- Uloga CCD senzora je pretvaranje svjetlosti u električni signal.
- Jači intenzitet svjetlosti stvara električni signal većeg napona.
- Analogni naponski signal se postupkom analogno-digitalne konverzije pretvara u diskretnu digitalnu vrijednost, koja predstavlja podatak o boji određene točke.

3.24.2. Vrste skenera.

STOLNI SKENERI:

- Stolni skeneri su najčešće korištena vrsta skenera prilikom rada sa osobnim računalima.
- Skenira se jedna po jedna stranica medija.
- Stranica koja se skenira je nepomična, a pomiče se glava za skeniranje.
- Mogu skenirati medije do veličine formata A3 – za veće formate se koriste protočni skeneri.

RUČNI SKENERI:

- Minijaturni skeneri – lako su prenosivi, ali daju lošije rezultate od stolnih ili protočnih skenera.
- Skenirani medij je nepomičan, a čovjek mora ručno povlačiti skener preko medija.
- Kvaliteta skena jako ovisi o mirnoći ruke i ujednačenoj brzini ručnog pomicanja skenera.
- Za skeniranje čitavog medija (npr. veličine A4) potrebno je više puta preći preko medija.
- Svakim prolazom se skenira segment ('pruga') određene širine, ovisno o širini glave skenera.
- Dolazi do problema sa preklapanjem skeniranih segmenata.

PROTOČNI SKENERI:

- Skeneri velikih dimenzija, namijenjeni za profesionalnu primjenu i skeniranje izrazito velikih medija (do formata A0 i više).
- Glava za skeniranje je stacionarna, a pomiče se medij kojeg treba skenirati.
- Medij se postavlja na bubanj koji rotira ili se pomoću sustava valjaka provlači kroz glavu za skeniranje.

POSEBNE VRSTE SKENERA:

- Tipičan primjer skenera ove kategorije je čitač bar-koda.

3.25. Pisač

- Pisači su izlazne jedinice koje zapisuju podatke na papirnate medije ili folije.
- Prema načinu na koji se stvara ispis, pisače je moguće podijeliti na dvije glavne skupine:
 - **pisače s udarcem**, i
 - **pisače bez udarca**.

3.25.1. Pisači s udarcem

- Pisači s udarcem pripadaju starijoj generaciji pisača i danas se rijetko koriste.
- Od brojnih vrsta pisača sa udarcem koji su danas gotovo potpuno nestali (pisači sa bubnjem, sa lancem, sa lepezom, sa kuglom...), još se uvijek dosta često koriste tzv. matrični pisači (Dot-Matrix printers).
- Matrični pisači se danas uglavnom koriste u POS (Point Of Sale) sustavima, tj. 'pametnim' blagajnama za ispis računa.
- Matrični pisači se još uvijek koriste i u uredskom poslovanju – postoje situacije u kojima zakonodavac propisuje obveznu izradu većeg broja kopija nekog dokumenta, a jedino matrični pisači (ili bilo koja druga vrsta pisača s udarcem), zahvaljujući načinu na koji ostvaruju otisak, omogućavaju istovremeni ispis većeg broja kopija neke stranice (mehanički udarac + korištenje indigo-papira – kao na mehaničkim pisačim strojevima).

3.25.1.1. Matrični pisači

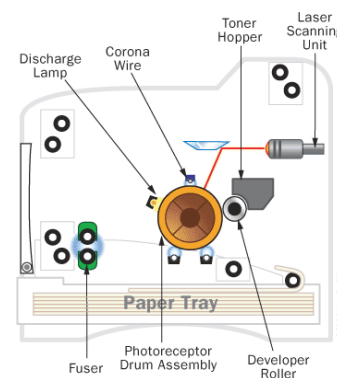
- Matrični pisač stvara ispis pomoću glave za pisanje.
- U glavi se nalazi određeni broj iglica (tipično 9 ili 24), raspoređenih u pravokutnu (kvadratnu) matricu.
- Glava se nalazi na pomičnom mehanizmu, koji pomiče glavu s lijeva na desno i natrag, omogućavajući time ispis jednog retka teksta.
- Iglice iz glave za pisanje udaraju po vrpici (ribbon) natopljenoj tintom, koja je smještena između glave i papira (slično kao kod pisačeg stroja).
- Ispis pojedinog znaka ili dijela slike se oblikuje tako da se istovremeno aktiviraju one iglice u glavi koje su potrebne za oblikovanje pojedinog slova ili dijela crteža.
- Nakon ispisa jednog retka, glava se vraća na početak reda, a valjak okomit pomiče papir za jedan redak.
- Matrični pisači mogu raditi u dva načina rada:
 - **tekstualnom**, i
 - **grafičkom**.
- Tekstualni način rada:
 - Pisač ispisuje slova, brojeve i određene unaprijed definirane specijalne simbole i znakove (npr. po ASCII tablici znakova).
 - Pisač u svojoj ROM memoriji ima u obliku bitmape definiran izgled svakog znaka, te na temelju bitmapirane definicije znaka aktivira potrebne iglice u glavi pisača, kako bi ispisao potrebni znak.
- Grafički način rada:
 - U ovom načinu rada matrični pisač za ispis ne koristi definicije znakova iz svoje ROM memorije, već točku po točku iscrtava bitmapiranu sliku koju mu šalje računalo.
- Rezolucija matričnog pisača se izražava u broju znakova (slova) po inču – **cpi** (Characters Per Inch).
- Tehnologija izrade matričnih pisača se već dulje vrijeme ne razvija i unapređuje.
- Kvaliteta tekstualnog ispisa matričnog printera je loša (u usporedbi sa laserskim ili ink-jet pisačima), a kvaliteta slikovnog izlaza je još lošija.
- Ispis je bučan i relativno spor.
- Korisne osobine matričnih pisača:
 - Mogućnost ispisa na tzv. beskonačni papir, što ne mogu ni laserski ni drugi suvremeni pisači bez udarca.
 - Istovremeni ispis više kopija iste stranice – zbog mehaničke naravi ispisa (udarci iglica), ispis je korištenjem indigo-papira moguće istovremeno zapisati na veći broj papira.

3.25.2. Pisači bez udarca

- Pisači bez udarca pripadaju novijoj generaciji pisača i danas se najčešće koriste.
- Osnovna karakteristika pisača bez udarca je da zapis na papiru ne realiziraju mehaničkim udarcima, već na razne druge načine.
- Kvaliteta ispisa teksta i slike je superiorna u odnosu na pisače sa udarcem.
- U pisače bez udarca se ubrajaju sljedeće vrste pisača:
 - laserski pisači
 - tintni (ink-jet) pisači
 - termalni pisači (kod fax uređaja)
 - solid-ink pisači
 - sublimacijski pisači...

3.25.2.1. Laserski pisači

- Način rada laserskih pisača je uvelike istovjetan načinu rada fotokopirnog stroja.
- Za razliku od matričnih pisača, laserski rade isključivo u grafičkom načinu rada – i tekst i slike se "iscrtavaju", tj. tretiraju se istovjetno – kao slike.
- Građa i način rada:
 - Zadatak laserskog pisača je da rastersku sliku prenese na papir – pomoću sljedećih osnovnih dijelova:
 - Izvora laserske zrake
 - Sustava leća i ogledala za usmjeravanje laserske zrake
 - Fotosenzitivnog bubnja
 - Tonera
 - Elektroda za nanošenje statičkog naboja na bubanj i papir
 - Niza pomoćnih valjaka za fiksiranje tonera i transport papira.
- Površina fotosenzitivnog bubnja, koji se okreće velikom brzinom, se pomoću posebne elektrode nabija negativnim nabojem.
- Pomoću stacionarne laserske zrake se rasterska slika postupno prenosi na površinu bubnja.
- Laserska zraka se preko sustava leća i rotacijskog zrcala usmjerava i fokusira na bubanj.
- Zraka se usmjerava isključivo u smjeru lijevo-desno, poprečno preko bubnja.
- Zraka pogađa samo ona mjesta na bubnju na kojima treba biti vidljiva točka.
- Mjesta koja je pogodila laserska zraka gube negativni naboj i postaju pozitivno nabijena.
- Daljnjom rotacijom bubnja, mjesta pogođena zrakom prolaze pored kazete sa tonerom.
- Toner - iznimno fini crni prah, koji ostavlja fizički trag zapisa na papiru.
- Čestice tonera su također nabijene negativno, što znači da će se prilikom prolaska bubnja toner uhvatiti samo na ona mjesta na bubnju koja su izgubila negativni naboj, a poprimila pozitivan naboj.
- Daljnjom rotacijom bubnja, on dolazi u kontakt sa listom papira, koji je prethodno pomoću posebne elektrode nabijen pozitivnim nabojem.
- Pri kontaktu se negativno nabijene čestice tonera priljepe na pozitivno nabijeni papir.
- Time je laserom definirana slika sa bubnja prenesena na papir.
- Papir sa tonerom putuje dalje prema komori sa valjcima (fusers) za fiksiranje tonera na papir.
- Elektrostatska veza pozitivno nabijenog papira i negativnih čestica tonera je preslaba da bi se toner zadržao na potrebnim mjestima.
- Pod utjecajem povišenog tlaka i temperature, toner se tali i fizički priljepljuje za papir.
- Istovremeno, bubanj nastavlja sa rotacijom, a da bi bio spreman za novi ciklus ispisa, potrebno je sa bubnja ukloniti eventualno zaostale čestice tonera .
- To se obavlja pomoću posebne elektrode (korone), koja svojim pozitivnim nabojem privlači eventualno zaostale čestice tonera.
- Nakon čišćenja, bubanj opet prolazi pored početne elektrode za negativno nabijanje i cijeli postupak se ponavlja dok se ne ispišu sve stranice.



SLIKA 3.40: GLAVNI DIJELOVI LASERSKOG PISAČA

3.25.2.2. Tintni pisači (ink-jet)

- Laserski pisači u boji su još uvijek prilično skupi, pa su za potrebe ispisa u boji razvijeni ink-jet pisači.
- Ink-jet pisači stvaraju zapis na papiru pomoću mikroskopski sitnih kapljica tinte, koje se ispaljuju iz glave za ispis.
- Kao i laserski pisači, ink-jet pisači također rade isključivo u grafičkom načinu rada, tj. isctravaju i tekst i slike kao bitmape.
- Ispis koji stvaraju ink-jet pisači mogu biti:
 - crno-bijeli – za ispis je potrebna samo crna tinta
 - u boji – za svaku osnovnu boju (cyan, magenta, yellow) potreban je po jedan spremnik za tintu.
- Princip rada:
 - Glava za ispis prelazi horizontalno preko medija za ispis (papir, folije...).
 - Na glavi se nalazi veći broj mikroskopski tankih cjevčica (tanje su od ljudske vlasi).
 - Na potrebnim mjestima će kroz određene cjevčice mehanizam u glavi za pisanje izbaciti sićušne kapljice tinte (promjera 50 do 60 mikrometara) odgovarajućih osnovnih boja.
 - Kapljice boje se upijaju u papir i miješanjem stvaraju točkicu određene boje.
 - Jednim prolazom glave po širini papira se ispisuje više redaka, jer su cijevčice u glavi posložene u više redaka i stupaca.
- Ovisno o tehnologiji izrade glave za ispis, razlikuju se dvije osnovne vrste ink-jet pisača:
 - **s termičkom glavom**, i
 - **s piezo-električnom glavom**.

INK-JET PISAČI S TERMIČKOM GLAVOM:

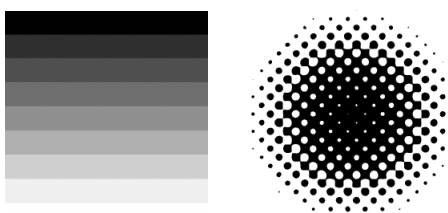
- Za izbacivanje kapljica tinte se koristi toplina.
- Grijač u vrlo kratkom intervalu snažno zagrije tintu, te prestaje sa zagrijavanjem.
- Neposredno pored grijača nastaje mjehurić tinte koji se vrlo brzo širi.
- U određenom trenutku mjehurić eksplodira i kroz cjevčicu se izbacuje kapljica tinte.
- Hlađenjem elementa za zagrijavanje, mjehurić tinte se urušava, čime se u glavi stvara zrakoprazni prostor, koji iz spremnika uvlači novu količinu tinte.
- Zbog načina rada, tinta mora biti otporna na toplinu, tj. ne smije promijeniti kemijska svojstva pod utjecajem topline.

INK-JET PISAČI S PIEZO-ELEKTRIČNOM GLAVOM:

- Ovdje se za izbacivanje kapljica ne koristi toplina, već vibracije piezo-kristala.
- Piezo-kristali vibriraju kad kroz njih prolazi struja.
- Glava za ispis se sastoji od brojnih piezo-elemenata, od kojih je svaki zadužen za izbacivanje kapljice tinte kroz jednu cjevčicu.
- Vibracije piezo-kristala se preko fleksibilne membrane prenose na tintu, čime se iz cjevčice izbacuje kapljica tinte.
- Piezo-električne glave imaju brojne prednosti u odnosu na termalne glave:
 - Omogućena je bolja kontrola nad veličinom i oblikom izbačene kapljice.
 - Sitne vibracije piezo-kristala omogućavaju izbacivanje sitnijih kapljica.
 - Ne treba zagrijavati i hladiti tintu kod svakog ciklusa ispisa.
 - Tinta ne mora biti termički otporna, čime je omogućena veća sloboda u izboru materijala za tintu, a time su moguće i kvalitetnije tinte.
- Postoje i određeni nedostaci:
 - Mehanizam je osjetljiv na eventualne mjehuriće zraka u tinti.
 - Piezo-tehnologija je u isključivom vlasništvu tvrtke Epson.
- Rezolucija pisača bez udarca:
 - Izražava se kao broj individualnih točaka po inču (**dpi** – Dots Per Inch) koje pisač može ispisati na medij.
 - Tipične rezolucije laserskih pisača se kreću od 600 do 2400 dpi .
 - Rezolucije ink-jet pisača se kreću od 600 do 4800 dpi.

3.25.3. Rasterska točka

- Svaki pisač ima vlastitu maksimalnu razlučivost, koja se izražava kao broj točaka po inču (tj. po kvadratnom inču).
- Time je određena i fizička dimenzija najmanje moguće točke koju pisač može ispisati.
- Isto tako, svaki pisač ima određen raspon boja koje može prikazati.
- Jednobojni pisači – laserski, matrični – osim boje papira, mogu prikazati samo jednu boju.
- Pisači u boji – tintni, laserski... – mogu prikazivati široki raspon boja, ali i taj raspon je konačan, a ljudsko oko je u stanju zamijetiti "nedostatak" pojedinih nijansi boja.
- Usprkos limitiranom broju boja, čak i jednobojni (crno-bijeli) pisači mogu prikazati i međutonove – različite **nijanse sivog** u rasponu od crne do bijele boje.



SLIKA 3.41: PRIKAZ NIJANSA SIVE BOJE

- Nijanse sive boje se postižu ispisom i grupiranjem većih ili manjih točaka, sastavljenih od elementarnih crnih točaka – ljudsko oko zbog svoje nesavršenosti percipira različite gustoće i veličine crnih točkica kao jednolične nijanse sive boje.
- Gušće grupirane točke daju tamniju nijansu sive, a rjeđe grupirane točke daju svijetlije nijanse sive boje.
- Osnovni element od kojeg se sastavlja slika je točka. Kako se točke tvore od najmanjih jedinica koje može ispisati ispisna jedinica, uvodi se pojam rasterskog polja odnosno rasterske linije.
- **Rastersko polje** je dio mreže ispisne jedinice unutar koje se tvori rasterska točka, dok se niz rasterskih točaka naziva **rasterska linija**.
- Gustoća rasterskih linija naziva se **linijatura** i izražava se u broju linija po inču (**lpi**) ili po centimetru (**lpcm**).
- Broj nijansi sive koje se mogu prikazati ovisi o linijaturi i rezoluciji ispisne jedinice, tj. o veličini rasterske točke.
- Rasterska točka je, dakle, krupnija od osnovne točke ispisne jedinice, a gustoća i raspored osnovnih točkica unutar rasterske točke određuju nijansu sive boje koju će percipirati ljudsko oko.
- Broj nijansi sive koje se mogu prikazati ovisi o linijaturi i rezoluciji ispisne jedinice, tj. o veličini rasterske točke.

3.25.4. Font

- Fontom nazivamo skup znakova istog pisma (npr. Courier), istog stila (npr. kosi, eng. italic), iste težine poteza (npr. masni, eng. bold), te veličine (npr. 12 točaka).
- Font i pismo nisu istovjetni pojmovi.
- **Font** se odnosi na sve znakove raspoložive u određenoj veličini, stilu i težini, dok **pismo** označava sam njihov oblik.
- Oblik, tj. dizajn pisma se još naziva i typeface – razlikujemo nekoliko vrsti pisama:
 - *Serifna* (npr. Roman, Courier) i *neserifna* (npr. Arial, Verdana) pisma (serif je krovna crta – crtica ili vitica na završecima slova).

3.25.4.1. Vrste fontova

- Fontove najčešće dijelimo prema dva kriterija:
 - širini znakova, i
 - tehnologiji izrade.

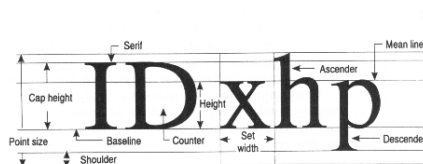
- Prema širini znakova razlikujemo:
 - **Neproporcionalne** (fiksne, nerazmjerne) fontove:
 - Svi su znakovi jednako široki – i "usko" slovo "I" i "široko" slovo "M".
 - Skup znakova je oblikovan tako da pri tiskanju svaki znak zauzima jednak prostor na papiru.
 - Npr. font Courier ("Tekst pisan fontom Courier").
 - **Proporcionalne** (razmjerne) fontove:
 - Znakovi su različite širine, zauzimaju toliko mjesta koliko im prirodno treba.
 - Skup znakova je oblikovan tako da svaki znak pri tiskanju na papiru zauzima odgovarajuću (veću ili manju) količinu prostora, npr. "usko" slovo "I" zauzima manje mjesta od "širokog" slova "M".
 - Npr. fontovi Arial, Roman, Verdana, ... ("Tekst pisan fontom Arial").

Ovo je Courier.

Ovo je Arial.

SLIKA 3.42: NEPROPORCIONALNI I PROPORCIONALNI FONT

- Prema tehnologiji izrade razlikujemo:
 - **Vektorske** (obrisne) fontove:
 - Oblik znakova je određen matematički stvorenim crtama i krivuljama umjesto uzorcima točaka.
 - Mogu se bez gubitka kvalitete dimenzionirati na bilo koju veličinu.
 - **Rasterske** (bitovne, bitmapirane) fontove:
 - Svaki je znak opisan jedinstvenim uzorkom bitova (uzorkom točaka, bitmapom).
 - Optimizirani su za određenu veličinu znakova.
 - Promjenom veličine znaka (naročito kod povećavanja) dolazi do gubitaka u kvaliteti prikaza.



SLIKA 3.43: GLAVNI DIJELOVI FONTA

3.25.4.2. RIP – pretvorba iz vektorskog formata u rasterski

- Jedan od problema prilikom prikaza ili ispisa teksta i slike na većini izlaznih uređaja je kako vektorski definiran tekst (fontovi su najčešće vektorski) ili vektorski definirane crteže pretvoriti u rasterski (bitmapirani) oblik, kakav je jedini razumljiv većini izlaznih uređaja (monitori, pisači bez udarca...).
- Postupak pretvaranja vektorskog zapisa u bitmapu naziva se **RIP** (Raster Image Processing).
- Sam postupak se izvodi pomoću RIP (Raster Image Processor) softvera.
- On kroz niz manje ili više kompleksnih koraka prevodi matematički definirane vektorske zapise u bitmape, tj. u oblik u kojem se zapis pamti kao slika definirana kao niz individualnih točkica odgovarajuće boje.
- RIP može biti prisutan u raznim "oblicima":
 - Ugrađen u pisače, kao dio softvera u ROM memoriji pisača.
 - Kao zasebna aplikacija, koja se instalira na korisnikovo računalo ili mrežni poslužitelj.
 - Kao zasebni uređaj, tj. posebno dizajnirano računalo, čija je jedina uloga rasterizacija vektorskih zapisa – zbog brzine i skupoće, primjenjivo je samo u najzahtjevnijim situacijama.

- Bitmapa:
 - slika koja će se iscrtavati na ekranu ili ispisivati na pisaču nalazi se pohranjena u memoriji
 - slika je organizirana kao pravokutna mreža točaka.
 - svaka točka u mreži predstavlja jedan piksel slike, a za zapis svakog piksela potreban je određeni broj bitova – odatle i naziv bitmapa.
 - što se više bitova upotrijebi za opisivanje pojedinog piksela, to će se više boja moći prikazati na slici.
 - za prikaz slike u punom opsegu boja (true color) potrebna su **24 bita po pikselu** (tj. 8 bitova za svaku osnovnu boju – crvenu, plavu i zelenu).

3.26. Organizacija obrade podataka

- Do sada smo opisivali kako se izvode instrukcije na strojnoj razini, kako se adresira memorija prilikom dohvaćanja operanada itd. – opisivali smo obradu podataka na nižoj (strojnoj) razini.
- Sada ćemo opisati obradu podataka na višoj razini – nećemo gledati pojedinačne instrukcije, već programe kao cjeline, te skupine programa.

3.26.1. Sekvencijalna (slijedna) obrada

- Kao što i njeno ime govori, ovaj oblik organizacije obrade podataka podrazumijeva slijedno izvođenje programa ili procesa, jednog za drugim.
- Redosljed je zadan poretkom pristizanja zahtjeva za izvođenjem pojedinog programa (procesu).
- Procesor posvećuje svoje cjelokupno vrijeme jednom procesu dok ga ne izvrši do kraja – tek tada može prijeći na izvođenje slijedećeg procesa koji čeka na izvođenje.
- Formiraju se **redovi čekanja** prema redosljedu pristizanja zahtjeva za obradom.
- Očito je da ovakav način obrade nije najefikasniji.
- Važniji programi mogu dugo čekati na obradu.

3.26.2. Sekvencijalna (slijedna) obrada s prioritetima

- To je takva obrada pri kojoj se određenim procesima dodjeljuje odgovarajuća važnost (tj. prioritet) izvođenja.
- Dodjela važnosti izvođenja (**prioritet**) izvodi se na nivou operativnog sustava.
- Proces se izvršavaju po određenom redosljedu.
- Redosljed je određen težinom ili važnošću tog procesa.
- Važnost određuje OS na temelju programirane strategije obrade.
- Nekim procesima i sam administrator sustava može dodijeliti veći ili manji prioritet izvođenja.
- Pri radu računala procesi manjeg prioriteta stavljaju se u red čekanja ako je dat zahtjev za izvođenjem procesa većeg prioriteta.
- Takav način rada koriste gotovo svi operativni sustavi i na njima je da određuju prioritete.
- Većinom procesi korisničkih aplikacija imaju manji prioritet, a procesi OS-a veći, no prema potrebi moguće su i administratorove intervencije.

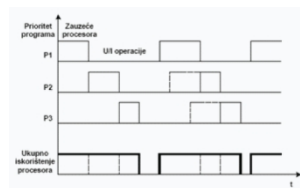
3.26.3. Time-sharing obrada (obrada s podjelom vremena)

- Obrada s podjelom vremena naziva se TIME SHARING.
- Time sharing je pojam za rad sustava u kojem se svakom programu (procesu) dodjeljuje diskretni vremenski odsječak.
- Kod obrade s podjelom vremena koriste se vrlo kratki vremenski odsječci **jednake duljine** u kojima se izvode programi.
- Takva obrada u stanju je "istovremeno" obrađivati više od jednog procesa.
- Kako su vremenski su odsječci vrlo mali korisniku čini da se istovremeno obrađuju svi pokrenuti programi.
- Taj je oblik obrade prisutan kod velikih i srednjih računala, a rijetko kod osobnih računala.
- Processorski resursi dodjeljuju se u jednakim odsječcima svim procesima koji čekaju na obradu.
- Kada jednom procesu istekne vrijeme predviđeno vremenskim odsječkom obrada se prebacuje na drugi proces i tako redom.

- Procesi koji čekaju na izvršenje nalaze se u redu čekanja koji se opslužuje po FIFO redosljedu (First In First Out), tj. po redosljedu pristizanja u red.

3.26.4. Multiprogramska obrada (Multiprogramming, Multitasking)

- Ova vrsta organizacije obrade podatka je karakteristična za višezadačne i višekorisničke operacijske sustave, koji su u stanju prividno izvršavati više procesa istovremeno.
- Procesor za svaki proces odvaja odsječak svog vremena (slično time-sharing obradi, ali ovdje odsječci **nisu jednakog trajanja**) i izvršava ga unutar njegovog odsječka vremena.
- Istjekom zadanog vremenskog odsječka prelazi se na obradu nekog drugog procesa itd.
- Opet se uočava sličnost s time-sharing obradom, ali kod multiprogramminga redosljed opsluživanja procesa nije strogo zadan redosljedom pristizanja zahtjeva za obradom.
- Moguće je dodjeljivanje različitih prioriteta procesima.
- Procesi sa višim prioritetom mogu prekidati izvođenje procesa sa nižim prioritetima – korištenjem mehanizma programskih prekida.



SLIKA 3.44: DIJAGRAM ZAUZEĆA PROCESORA PRI MULTIPROGRAMSKOJ OBRADI

3.26.5. Multiprocesiranje (Multiprocessing)

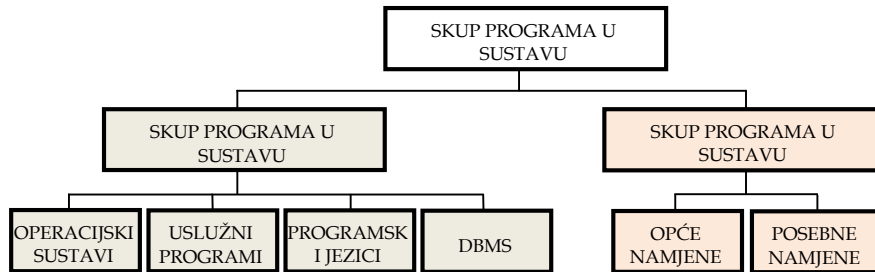
- Ova organizacija obrade podataka je karakteristična za računala koja posjeduju **više procesora**.
- Takva računala mogu stvarno obrađivati više procesa istovremeno (koliko procesora, toliko stvarnih istovremeno izvršavanih procesa), za razliku od jednoprocorskih sustava koji, zahvaljujući isključivo brzini rada, samo stvaraju privid istovremenog izvođenja većeg broja procesa (multiprogramming, time sharing).

3.26.6. Obrada u stvarnom vremenu (Real-time obrada)

- Obrada u realnom vremenu je potpuno drugačija koncepcija od svih do sada opisanih.
- Karakteristična je za računala koja se koriste u industrijskom okruženju i složenim tehnološkim sustavima, gdje služe za upravljanje proizvodnjom ili općenito za upravljanje radom takvih složenih sustava.
- Za obrade podataka u takvim okruženjima je karakteristično postojanje vremenskih ograničenja unutar kojih obrada mora biti izvedena – dakle obrada mora biti izvedena u zadanom vremenskom okviru.
- Ovisno o "osjetljivosti" sustava na probijanje vremenskih ograničenja razlikujemo:
 - "**Tvrdu**" real time obradu – vremenski kritični sustavi, kod kojih se akcije apsolutno moraju izvesti u zadanim vremenskim ograničenjima.
 - npr. *autopilot u avionu* – svako zakašnjenje u obradi može biti fatalno za sigurnost leta.
 - "**Meku**" real time obradu – vremenska ograničenja postoje, ali probijanje rokova nije kritično za funkcioniranje sustava.
 - npr. *sustav za praćenje robe pomoću bar-koda na tekućoj vrpici* – u slučaju zastoja na vrpici, sustav će usporeno raditi, ali još uvijek ostaje funkcionalan.
- Za takvu vrstu obrade potrebna su i posebna računala (prilagođena poštivanju strogih vremenskih ograničenja), ali i posebni operacijski sustavi, također prilagođeni strogom poštivanju traženih vremenskih ograničenja.

4. SOFTWARE

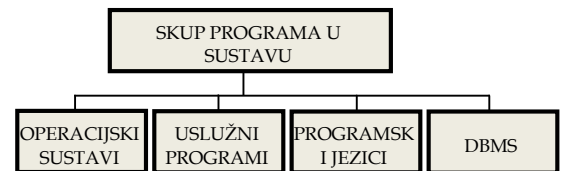
- Kao što smo već naveli u 1. poglavlju, programsku potporu radu računala (software) čine svi računalni programi koji se koriste pri radu tog sustava. Računalne komponente (sklopovlje) tvore funkcionalnu cjelinu zahvaljujući postojanju programske potpore. Podjela programske potpore prikazana je na sljedećoj slici.



SLIKA 4.1: PODJELA SOFTWARE-A

4.1. Sistemski software

- Čini ga skup strojno orijentiranih programa s funkcijom upravljanja i kontrole računalnog sustava u cilju sinkronizacije rada s aplikativnom podrškom. Tu spadaju:
 - operacijski sustavi** – koordiniraju rad računala
 - programi prevoditelji** – služe za prevođenje programa pisanih u simboličkim jezicima u strojni jezik.
 - kompilatori
 - interpreteri
 - simulatori
 - emulatori
 - generatori.
 - uslužni (pomoćni) programi** (*utility software*) – nalaze se između sistemskog i aplikativnog softvera te daju podršku u nekim operacijama vezanim za obradu podataka.
 - sustavi za upravljanje bazama podataka (DBMS)**– obavljaju sve operacije nad podatcima u bazi, te nadgledaju podatke i vode brigu o zaštiti baze.



SLIKA 4.2: SISTEMSKI SOFTWARE

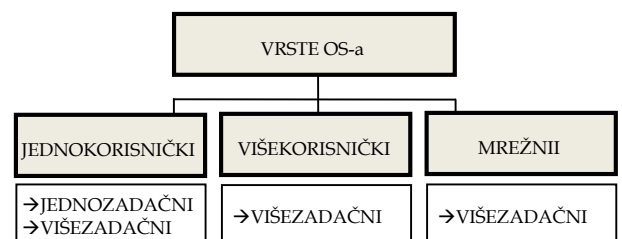
4.2. Operacijski sustavi



- Skup programa koji omogućuje izvođenje operacija računala naziva se operacijskim sustavom. Zadaća mu je da objedini sve komponente u jednu skladnu cjelinu i kao takvom upravlja svim njenim dijelovima. Isto tako skriva od korisnika mnoge nevažne detalje izvođenja operacija. Još jedan važan zadatak operacijskog sustava jest djelotvorno iskorištavanje svih dijelova računala. Sve operacije operacijskog sustava može pokrenuti čovjek preko korisničkog sučelja ili program preko sučelja primjenskog programa.
- Neki od operacijskih sustava današnjice:
 - Windows 95, 98, Millenium (Me), 2000, XP, 2003 Server. Vista itd.
 - Unix, Linux (verzije: Red Hat, Mandrake, SuSe itd.), IBM OS/2, Mac OS itd.

4.2.1. Vrste operacijskih sustava

- Osnovni zadatak operacijskog sustava jeste stvaranje uvjeta za izvođenje korisničke programske potpore. Ovisno o konfiguraciji i zadacima pojedinih računalnih sustava, operacijske sustave dijelimo na tri osnovne skupine što se vidi na slici.
- Vrste operacijskih sustava:
 - Mrežni
 - Višekorisnički
 - Jednokorisnički
 - Jednozadaćni
 - Višezadaćni.



SLIKA 4.3: VRSTE OPERACIJSKIH SUSTAVA

- **JEDNOKORISNIČKI** – svi računalni resursi stavljaju se na raspolaganje samo jednom korisniku. Jednozadaćni operacijski sustavi mogu nadzirati rad samo jedne aplikacije, dok je pokretanje više aplikacija onemogućeno (primjer: MS DOS). Višezadaćni operacijski sustavi mogu pokrenuti više aplikacija istovremeno uz korištenje jednog procesora, te omogućuju prelazak iz jedne u drugu aplikaciju (multi-tasking). Primjer takvog operacijskog sustava je MS Windows 95 pa nadalje.
- **VIŠEKORISNIČKI** – upravlja radom računala koja sva rade pod jednim operacijskim sustavom. Budući da se radi o većem broju računala koja istovremeno mogu raditi na različitim aplikacijama, takav je sustav višezadaćni sustav.
- **MREŽNI** – javljaju se u slučaju mrežne komunikacije među računalima sa različitim operacijskim sustavima. Oni se uvode radi raspodjele resursa, poboljšanju na komunikaciji između računala, povećanja pouzdanosti itd.

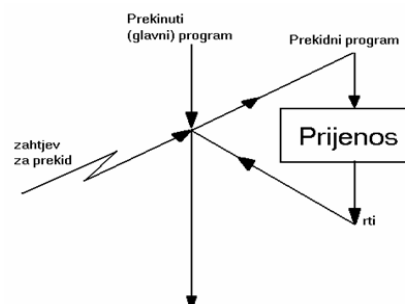
4.2.2. Funkcije operacijskih sustava

- S obzirom na vrste i namjenu operacijskih sustava, razlikujemo nekoliko funkcija operacijskog sustava. Neke opće funkcije najkorištenijih operacijskih sustava jesu:
 - *upravljanje sustavom ulazno izlaznih jedinica* – omogućuje skup modula koji moraju izvršavati nadzor i evidenciju nad jedinicama, određivanje načina pridjevanja jedinica pojedinim obradama, fizičko povezivanje određenih tijekova obrade podataka sa određenim perifernim jedinicama, fizičko oslobađanje perifernih jedinica itd.
 - *upravljanje višeprocorskim radom* – postoje osnovne vrste modula koji se brinu i kontroliraju zadatke obrade, te nadziru status i rad procesora. Njihovom se interakcijom ostvaruje primarni zadatak pridjeljivanja procesora, odnosno njegovih dijelova pojedinim zadacima obrade.
 - *upravljanje memorijom* – odražava koncepciju korištenja glavne memorije računala u koju se pohranjuju određene informacije i instrukcije od strane procesora. Ovi moduli brinu o zauzetosti pojedinih dijelova glavne memorije, utvrđuju način alokacije dijelova memorije i brisanja odnosno prijenosa podataka iz tih dijelova
 - *upravljanje aritmetičko-logičkom jedinicom*
 - *upravljanje sabirničkim sustavom, tijekom podataka*
 - *upravljanje korištenjem vremena izvođenja pojedinog procesa*
 - *obrada prekida, itd.*

4.2.3. Obrada prekida i operacijski sustav

- Jedna od osobina računala, odnosno operacijskih sustava jeste da omoguće da na zahtjev neke vanjske jedinice nastane prekid glavnog programa, te da se prijeđe na izvođenje posebnog programa za tu vanjsku jedinicu. Time računalo obavlja neku servisnu rutinu za vanjsku jedinicu koja je prekid zahtijevala, a nakon toga nastavlja izvođenje prekinutog programa.
- Kad računalo prihvati zahtjev za prekidom programa, ono mora najprije završiti instrukciju glavnog programa u toku koje je stigao zahtjev za prekid toga programa. Nakon toga mora sačuvati sve podatke potrebne za nastavak prekinutog programa. To znači da mora sačuvati adresu slijedeće instrukcije prekinutog programa, sadržaj akumulatora i drugih registara, itd. da bi se prekinuti program, nakon izvođenja neke servisne rutine zbog koje je prekinut, dalje mogao izvoditi kao da se ništa nije dogodilo. Drugi zadatak koji računalo mora obaviti jest identifikacija vanjske jedinice koja zahtijeva prekid programa. Ukoliko postoji više jedinica koje su tražile prekid programa, moraju se odrediti prioriteta jedinica.
- Osim **jednostrukih** prekida, postoje i **višestruki** prekidi. Kod jednostrukih prekida, servisna rutina koja je prekinula glavni program ne može se prekinuti, te ukoliko postoji zahtjev za prekidom programa od jedinice višeg prioriteta od one koja se izvodi, ta jedinica mora čekati da se servisna rutina koja je u tijeku završi. Kod višestrukih prekida servisne rutine koje se izvode mogu biti prekinute od servisnih rutina višeg prioriteta.
- Primjer: Povezivanje vanjskih jedinica s CPU – Sinkronizacija prijenosa – Prekidni U/I prijenos.
 - Na zahtjev vanjskog uređaja prekida se izvođenje tekućeg programa i prelazi se na izvođenje prekidnog programa.
 - Prekidni program poslužuje ulazno – izlaznu napravu.
 - U njegovom sastavu se nalaze i instrukcije kojima se obavljaju ulazne ili izlazne operacije.

- Nakon obavljenog zadatka procesor nastavlja izvođenje prekinutog programa.
- Uobičajeni slijed događaja tijekom prekidnog prijenosa:
 - Ulazno – izlazni uređaj generira zahtjev za prekid i šalje ga procesoru.
 - Na kraju izvođenja trenutne instrukcije procesor odgovara signalom potvrde i onemogućava nove prekide iste razine.
 - Procesor pohranjuje tzv. **minimalni kontekst** sa sadržajem PC-a, registra statusa i registara opće namjene.
 - Sadržaj se obično pohranjuje u memoriji (na **stogu**) - ovo je potrebno da bi se osigurao pravilan povratak iz prekidnog u prekinuti program i normalan njegov nastavak.
 - Upravljanje se prenosi na prekidni program u kojem se nalazi i programska rutina za ulazno - izlazne jedinice.
 - Nakon izvođenja programske rutine za posluživanje ulazno-izlazne jedinice, upravljanje se prenosi na izvođenje prekinutog programa.
 - To se obavlja zadnjom instrukcijom prekidnog programa, tipično **rft** (Return From Interrupt) instrukcijom.
 - Izvođenjem te instrukcije obnavljaju se sadržaji programskog brojila PC, registra stanja i registara opće namjene (uzimanjem tih sadržaja s vrha stoga).



SLIKA 4.4: OBRADA PREKIDA

4.3. Programski jezici

- Osnovni princip rada računala jest izvođenje niza instrukcija koje su unaprijed pripremljene i pohranjene u radnom spremniku računala. Niz takvih instrukcija naziva se programom. Da bi se izradio program potrebni su programski jezici, metode programiranja, te metode analize realnih sustava.
- Budući da računalo "razumije" samo strojni jezik tj. strojnu instrukciju, razvoj programa je u početku bio vrlo težak i kompleksan zadatak. Kako bi se olakšalo pisanje programa, vrlo rano su uvedeni viši programski jezici (high level languages). U tim se jezicima upotrebljavaju instrukcije koje zamjenjuju više strojnih instrukcija i zapisuju se na način razumljiv čovjeku. Kako bi se takve instrukcije prevele u jezik razumljiv računalu, dakle strojni jezik, moraju postojati programi prevodioci ili kompajleri. Danas je u upotrebi mnogo programskih jezika. Neki od najvažnijih programskih jezika su: C, C++, Delphi, Visual Basic, te nešto stariji poput COBOL-a, Pascal-a, BASIC-a i dr.
- **Strukturalno** se **programiranje** temelji na pokušaju prilagođavanja programiranja strukturi prirodnog jezika. Osnovna ideja je u izbacivanju GO TO naredbe, jer se smatra da se njome kompliciraju programska rješenja. Kod strukturalnog programiranja, program se logički dekomponira, te se hijerarhijski umeće jedna struktura unutar druge prema kriteriju odzgo prema dolje.
- Posljednjih se godina uvodi tzv. **objektno orijentirano programiranje** čija je osnovna zamisao objedinjavanje strukture podataka i funkcija koje nad njima djeluju u jedan objekt. Svaki objekt opisan je svojim stanjem i svojim mogućim ponašanjem. U takvom se okruženju funkcije nazivaju metodama. Tu ubrajamo programske jezike C++ i Javu.

4.3.1. Elementi programskih jezika

- Programski jezik čine slijedeći elementi:
 - **Ključne riječi (naredbe)**
 - **Pravila pisanja naredbi**
 - **Editor programskog koda**
 - **Program prevoditelj.**

KLJUČNE RIJEČI JEZIKA:

- Još ih nazivamo i naredbama.
- Svaka naredba izvršava nekakav elementarni zadatak na nivou programskog jezika.
- Kombiniranjem ključnih riječi u veće cjeline nastaje izvorni kod programa, tj. program napisan u određenom programskom jeziku.
- Pravila pisanja naredbi:

- Spomenuto je da program nastaje povezivanjem većeg broja naredbi nekog programskog jezika u veće cjeline. No, naredbe se ne mogu i ne smiju nizati bilo kakvim redoslijedom, već logički ispravnim redoslijedom.
- Osim toga, pojedinačne naredbe moraju biti zadane na pravilan način – u skladu sa pravilima (sintaksom) pisanja naredbi za dotični programski jezik.
- Npr. možda kod pisanja naredbi treba pripaziti na upotrebu velikih i malih slova.
- Npr. neke naredbe zahtijevaju argumente (parametre) za ispravan rad, a neke ne – treba pripaziti da se navedu svi potrebni argumenti, i to u odgovarajućem redoslijedu.

EDITOR PROGRAMSKOG KODA:

- Programski kôd moramo napisati, tj. moramo imati nekakvo razvojno programsko okruženje unutar kojeg pišemo ključne riječi programskog jezika prema zadanim pravilima pisanja naredbi.
- Izvorni kod programa možemo pisati i u najobičnijem editoru teksta, a možemo ga pisati i unutar sofisticiranih razvojnih okruženja (**IDE** – Integrated Development Environment), koja, pored mogućnosti uređivanja teksta, u jedno okruženje integriraju i:
 - isticanje ključnih riječi programskog jezika
 - pomoć za prevođenje i/ili izvršavanje programa
 - pomoć za ispravljanje grešaka u programu (debugging) itd.

PROGRAM PREVODITELJ:

- Program napisan u programskom jeziku (izvorni kôd) je razumljiv čovjeku, ali ne i računalu koje će ga morati izvršavati.
- Kako računalu razumije jedino binarni jezik nula i jedinica, izvorni kôd programa se mora prevesti u računalu razumljiv strojni oblik (niz 0 i 1).
- S obzirom na vrstu i namjenu programskog jezika, prevođenje se može izvršiti kao kompiliranje (kompajliranje) ili kao interpretiranje.
- U tu svrhu se koriste programi prevoditelji – **kompajleri ili interpreteri**.
- Svaki programski jezik mora sadržavati i odgovarajući program-prevoditelj (bilo kompajler, bilo interpreter – rijetko koji programski jezik podržava oba načina prevođenja).

4.3.2. Generacije programskih jezika

- **Prva generacija.** Za komunikaciju sa prvom generacijom računala programeri su morali pisati programe u strojnom jeziku - sa oznakama 0 i 1 binarnog koda. Krajnji korisnik, koji je želio aplikaciju, morao je raditi sa specijaliziranim programerima koji su mogli shvatiti, misliti i raditi direktno na strojnom jeziku pojedinog računala. Programiranje sa 0 i 1 reduciralo je sve iskaze, kao npr: dodaj, oduzmi i podijeli u serije nula i jedinica, čineći tako programiranje sporim i napornim procesom.
- **Druga generacija.** Tipični predstavnik nižih programskih jezika je Assembler. On predstavlja grupu tzv. simboličkih jezika, koji su barem djelomično prilagođeni čovjeku, za razliku od strojnog jezika u kome, kao što smo rekli, računalu u stvari obavlja svoje operacije. Asembler i asemblerski jezici imaju osobinu da se njihove instrukcije (operacijski kod i operandi) pišu simbolima koje nazivamo mnemonici. U principu, odnos između takvih mnemoničkih instrukcija i strojnih instrukcija (a to znači i operacija u računalu) je 1:1. To znači da se ovako napisane instrukcije koje su bliske samoj strojnoj instrukciji veoma brzo izvode u računalu jer je postupak prevođenja u strojni kod direktan. Loša strana asembliranja je, međutim, u tome što zahtijeva izuzetno mnogo pisanja (instrukcija) od strane programera, a uz to zahtijeva i veoma dobro poznavanje svih hardverskih karakteristika onih jedinica računalnog sistema koje koristimo. S obzirom da se konfiguracije i karakteristike sklopovlja računalnih sustava međusobno razlikuju, to znači da asembler jednog programa nije kompatibilan (direktno upotrebljiv) na drugom kompjuterskom sistemu.
- **Treća generacija.** Ovi jezici su bitno olakšali rad čovjeka i njegovu komunikaciju sa računalom budući da su instrukcije vrlo slične govornom engleskom jeziku ili standardnoj matematičkoj notaciji. Potrebno je prevođenje naredaba u tom jeziku u više elementarnih naredbi u mašinskom jeziku (odnos između takvih naredbi i strojnih instrukcija je 1:više). Prevođenje se obavlja pomoću kompajlera ili interpretera. Takvi se jezici ponekad nazivaju i problemski orijentirani jezici, budući da je svaki od njih dizajniran i posebno pogodan za neku vrstu problema. Najznačajniji jezici ove generacije su:

- FORTRAN (FORmula TRANslation Systems) je jedan od najstarijih viših programskih jezika. Nastao je 1955. godine.
- COBOL (Common Business Oriented Language) je nastao 1960. godine u SAD i do danas je razvijeno više njegovih verzija.
- ALGOL (ALGOrithmic Language) predstavlja algoritamski jezik koji je kreiran na bazi Fortrana sa usavršenim procedurama i osobinama.
- BASIC (Beginners All-purpose Symbolic Instruction Code) je programski jezik koji se veoma lako i brzo uči. Razvijao se u SAD-u 1970. godine i izuzetno je pogodan za interaktivni rad i rad na mikroracionalima.
- PASCAL je viši programski jezik koji je nastao kombinacijom najboljih osobina Algola, Basic-a i programskog jezika PL/1. Predstavlja strukturni programski jezik, koji zahtijeva bolja programerska znanja, ali omogućava bolju preglednost programske strukture i jednostavnije ispravljanje eventualnih grešaka.
- LISP i PROLOG su široko korišteni u umjetnoj inteligenciji i posebno kao jezici za kreiranje ekspertnih sustava.
- **Četvrta generacija.** Tzv. šablonski pristup programiranju, koji karakterizira proceduralne jezike treće generacije, predstavljao je kočnicu procesu još masovnije informatizacije (od početka 90-tih godina), a što je rezultiralo novom generacijom jezika. Ova generacija na predstavlja samo efikasne programske jezike u klasičnim smislu riječi jezik, već praktično integrirane skupove jezika i visokosofisticiranih alata. Kombinacija navedenih pristupa dovela je do sljedećih jezika četvrte generacije: Query-By-Example (QBE), APL, Multiplan, Lotus 1-2-3, Ideal, Focus. Jezici četvrte generacije omogućavaju krajnjem korisniku da dođe do potrebnih informacija samostalnim korištenjem raspoložive baze podataka (dakle, bez pomoći profesionalnog programera). U novije vrijeme se kategoriziraju na slijedeći način: jezici upita, generatori izveštaja, grafički jezici, generatori aplikacija i programski jezici vrlo visoke razine. Razvoj jezika četvrte generacije doveo je do boljih rješenja u pogledu mogućnosti preuzimanja podataka između različitih aplikacija i efikasne obrade podataka u cjelini. Također, primjenom odgovarajućih mrežnih protokola stvorena je osnova za rješavanje problema "vezivanja" softvera za određene tipove računala, tako da je korištenje jezika četvrte generacije otvorilo nove perspektive razvoju informacijskih tehnologija upravo u mrežnom okruženju.

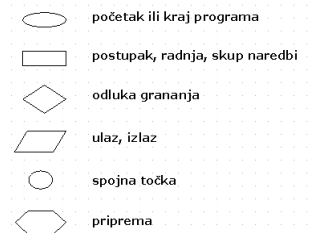
4.3.3. Kompajleri i interpreteri

- Kad se program pripremi, programer ga obično napiše u nekom programskom jeziku. Kad je program gotov, on predstavlja tzv. izvorni kôd (source code) programa. Prije nego će program biti izveden, mora se prevesti iz izvornog koda u strojni kôd korištenjem kompajlera ili interpretera.
- **Interpreter** je program koji prevodi napisani program u strojni kôd **liniju po liniju**, kako se program izvodi. Interpreter se nalazi u radnoj memoriji računala zajedno sa izvornim kodom programa kojeg izvodi. Većina interpretera ima mogućnost otklanjanja otkrivenih programskih pogrešaka. Isto tako, interpreter može dati izvještaj o grešci pomoću dijagnostičke poruke koja ukazuje na stvarni problem. U tom slučaju programer izravno ispravlja grešku, te program testira bez većih problema.
- **Kompajler** je program koji prevodi izvorne kodove u strojni jezik. Prema tome, učinak kompajlera je samo prevođenje. On ne izvodi program. Neki kompajleri zaustavljaju prevođenje ako naiđu na grešku, dok drugi nastavljaju do sljedeće greške. Programer mora sam pronaći grešku u izvornom kodu, ispraviti je i ponovno pokrenuti kompajler. Neki kompajleri prevode cijele programe, dok drugi prevode samo dijelove.

4.3.4. Blok dijagram i dijagram tijeka podataka

- **Blok dijagram:**
 - Svaki simbol u blok-dijagramu pokazuje jednostavnu proceduru i povezan je linijama poveznica koje određuju tijek obrade. Program je napisan počevši sa simbolom početka, prikazuje tijek procedura, a završava kad dođe do simbola označenog sa STOP.

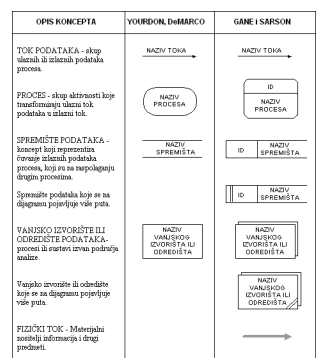
- Blok-dijagram sadrži simbole specifičnog značenja. Evo nekih najvažnijih:
 - početak ili kraj programa
 - postupak, radnja, skup naredbi
 - odluka grananja
 - spojna točka
 - ulaz, izlaz
 - priprema.



SLIKA 4.5: SIMBOLI KOD BLOK DIJAGRAMA

- **Dijagram toka podataka:**

- Dijagram toka podataka prikazuje tokove podataka u informacijski sustav i izvan njega. Sastoji se od:
 - ulaznih i izlaznih tokova podataka (zaslona, dokumenata) koje sustav dobiva ili daje u okruženje.
 - vanjskih objekata (organizacija, ljudi, drugih sustava) koji šalju prema ili primaju tokove podataka od sustava.
 - procesa sustava (programa, aktivnosti, postupaka, podsustava) koji transformiraju ulazno tokove podataka u izlazne.
 - spremišta podataka (baze podataka, kartoteke) u kojima se čuvaju podaci potrebni za izvršenje procesa ili dobiveni kao rezultat rada procesa.
- DTP NIJE dijagram toka programa i ne sadrži opise programske logike.
- Grafički prikazi (simboli različiti kod različitih autora, primjerice prema DeMarco, Yourdon, ili po Gane i Sarson):
- Tok podataka predstavlja se vektorom ili usmjerenim lukom.
- Proces (funkcija) predstavlja se zaobljenim pravokutnikom, elipsom i sl. Neki autori koriste pravokutnik.
- Spremište (skladište) podataka predstavlja se s dvije paralelne crte.
- Vanjski sustav (izvoriste ili odredište, granični entitet) predstavlja se pravokutnikom ili pravokutnicima u nizu poput karata. Neki autori ovdje koriste elipse.



SLIKA 4.5: DIJAGRAM TOKA PODATAKA

4.3.5. Metode programiranja

- Dva pristupa izgradnji aplikacije jesu **odozgo prema dolje** (Top-Down Design) i **odozdo prema gore** (Bottom - Up Design). Temelj tih metodoloških postupaka je činjenica da se svaki organizacijski sustav može podijeliti na konačan niz cjelina koje su međusobno nezavisne. Takve cjeline nazivamo modulima. Svaki modul pripada nekoj hijerarhijskoj razini. Pripadnost razini definiran je ishodištem opisivanja.
- Ukoliko je ishodišni modul, modul ulaznih sadržaja, tada je primijenjena metoda odozgo prema dolje, a ako je ishodišni modul, modul izlaznih sadržaja, primjenjuje se metoda odozdo prema gore.
- U izradi aplikacije treba odabrati, osim načina pristupa izgradnji, između slijedeća dva metodološka smjera:
 - **Metoda glavnog programera**, gdje svaki projekt raščlanjujemo na segmente te se svaki segment programira ponaosob od strane drugih programera čiji rad koordinira vodeći programer (razvija i ključne dijelove aplikacije).
 - **Strukturno programiranje**, koje se temelji na pokušaju prilagođavanja strukture programiranja strukturi prirodnog jezika. Problem se logički dekomponira, te se hijerarhijski umeće jedna struktura unutar druge prema kriteriju odozgo prema dolje.

4.3.6. Osnovne programske strukture

- Tehnika koja se obično koristi za organiziranje programa na papiru jeste **pseudokod**. To je program napisan u linijskoj formi koristeći engleske riječi i fraze za popis procedura koje će biti unesene u program.
- Osnovne logičke strukture koje možemo opisati pseudokodom jesu **sekvenca (slijed)**, **selekcija (odabir)**, **iteracija (ponavljanje)**, **te skokovi**.

- **Sekvenca (slijed)** uključuje niz naredbi koje se izvršavaju jedna za drugom.

Primjer: naredba 1
naredba 2
naredba 3

- **Selekcija (odabir)** podrazumijeva izvršavanje određenog bloka naredbi ovisno u ispunjenom uvjetu.

Primjer: IF uvjet THEN (ukoliko je zadovoljen uvjet, izvršava blok naredbi 1)
blok naredbi 1

IF uvjet THEN (ukoliko je zadovoljen uvjet, izvršava se blok 1, a ako nije izvršava se blok 2)
blok naredbi 1
ELSE
blok naredbi 2

SWITCH izraz (ovisno o vrijednosti izraza, izvršava se određen blok naredbi i to samo jedan blok)
CASE vrijednost 0
blok naredbi 1
CASE vrijednost 1
blok naredbi 2
CASE vrijednost 2
blok naredbi 3

- **Iteracija (ponavljanje)** izvršava određen blok naredbi tako dugo dok je uvjet ispunjen.

Primjer: Petlja sa izlazom na vrhu – ako uvjet nije ispunjen, ne izvršava se nijedna naredba.
WHILE uvjet DO (petlja prvo ispituje uvjet, a onda izvršava blok naredbi ovisno o ispunjenju
uvjeta)
blok naredbi

Petlja sa izlazom na dnu – izvršava se barem jedna naredba neovisno o ispunjenju uvjeta.
DO blok naredbi (petlja izvršava blok naredbi, a onda ispituje ispunjenje uvjeta)
WHILE uvjet

- **Skokovi** podrazumijevaju prijenos kontrole tijekom izvođenja programa na određenu liniju programa. Predstavnik jeste GO TO naredba koja se sve češće izbacuje (zbog strukturnog programiranja) kako se ne bi narušavao tijek izvođenja programa.

4.4. DBMS

- Sustav za upravljanje bazom podataka (Database Management System, DBMS) je programski sustav koji predstavlja **sučelje** između podataka pohranjenih u bazi podataka i korisnika koji im pristupaju. Baza podataka je skup podataka smještenih u različitim datotekama sa minimalnom redundancijom bez obzira na fizički smještaj na mediju.
- Funkcije Sustava za upravljanje bazom podataka jesu:
 - Projektiranje baze podataka (formiranje datotečnog i podatkovnog sadržaja tj. rječnika podataka).
 - Obavljanje različitih operacija sa podacima u bazi (čitanje, unos, promjena itd.).
 - Zaštita podataka (zaštita integriteta podataka, zaštita od neovlaštenog korištenja, kontrola istovremenog pristupa podacima, te obnavljanje baze podataka).
 - Nadgledanje i kontrola korištenja podataka.

4.5. Uslužni (pomoćni) programi

- Uslužna programska potpora (utility software) jest skup različitih programa kojima je namjena da korisniku olakšaju održavanje računalnog sustava. Tu spadaju i programi koji ubrzavaju rad na pojedinom radnom mjestu. Svaki operacijski sustav ima nekoliko takvih programa već ugrađenih. Npr. MS Paint, Disk defragmenter, Backup, Zip i sl.
- Uslužnu programsku potporu mogu sačinjavati i programi koji ne dolaze standardno sa operacijskim sustavom, već se instaliraju naknadno, a namjena im je da se korisnik njima služi kao uslužnim alatom, a ne da ih koristi kao podlogu za razvoj drugih programa. Takvi programi mogu biti svi kompresijski alati, antivirusni alati, alati za održavanje sustava (Norton Utilities) itd.

5. STRUKTURE ZAPISA PODATAKA

5.1. Podatak, entitet, slog, datoteka

- **Podatak** je bilo kakva forma u kojoj je fizički zabilježen neki događaj, zapažanje ili činjenica. Možemo ga definirati i kao skup prepoznatljivih znakova zapisanih na određenom mediju. Sa informatičke strane, podatak je simbolički prikaz obilježja promatranog objekta.
 - **Polje ili element podatka** jest osnovna i najmanja jedinica podatka od koje se prave druge strukture podataka.
 - Podaci se obično svrstavaju u grupe kojima se dodjeljuje ime.
 - Prema medijskoj vrsti podaci mogu biti znakovni, slikovni i zvukovni. Isto tako podaci se mogu komprimirati (sa i bez gubitka sadržaja), te se mogu kriptirati.
- **Entitet** se definira kao objekt promatranja o kojem organiziramo podatke. Entiteti imaju razne osobine, a mi promatramo sve ili samo one koje nas zanimaju.
- **Slog** odnosno zapis je element koji sadrži tj. utvrđuje obilježja jednog entiteta. U pravilu slog datoteke odgovara jednom retku u tablici.
 - Slog se sastoji od **polja**.
 - Svako polje sadrži podatke jednog obilježja (atributa) entiteta (npr. naziv kupca, šifra kupca i sl.).
 - Atribut pomoću kojeg možemo jednoznačno identificirati entitet zove se **ključni atribut ili ključ**.
 - Postoje dvije vrste slogova:
 - **Fizički slog** određen je načinom na koji su podaci smješteni u memoriju na nositelju podataka. Njime se opisuje način na koji se podaci memoriraju
 - **Logički slog** određuje korisnik informacijskog sustava u ovisnosti o funkcioniranju tog sustava. Ta je organizacija usmjerena prema realnom sustavu i njegovim objektima.
- **Datoteka** je skup istovrsnih podataka odnosno slogova, obuhvaćenih zajedničkim kriterijem i smještenih u memoriji računala.
- **Dimenzija datoteke** je određena brojem slogova u datoteci.
 - Jedan slog u datoteci sadrži podatke o jednom entitetu. Dakle, datoteka sadrži podatke o istovrsnim entitetima.
 - Slogovi, odnosno podaci u datoteci mogu biti sortirani i nesortirani. Način upisivanja slogova u datoteku ovisi o njejoj organizaciji.

5.2. Materijalni nositelji podataka

- S obzirom na tehniku zapisa podataka, možemo ih podijeliti na:
 - **analogne** (papir, bušene kartice, mikro film...) - zapis se vrši mehaničkim ili svjetlosnim djelovanjem na medij.
 - **magnetske** (tvrđi i meki disk, magn. traka, bubanj...) - zapis se vrši magnetskim djelovanjem na medij.
 - **optičke** (CD-ROM, CD-RW...) - zapis se vrši djelovanjem energetske zrake svjetlosti (laser) na medij.
- Generalna podjela jest na nosioce analognog i digitalnog zapisa, te ćemo takvu podjelu u nastavku i proučavati.

5.2.1. Nositelji analognog zapisa

Nositelji analognog zapisa su mikrofilm, papir i slika.

5.2.1.1. Mikrofilm

- uloga:
 - papir kao nositelj sadržaja ima malu otpornost na mehanička oštećenja.
 - nastaje radi potrebe čuvanja velikog broja analognih (papirnatih) podataka.
 - mikrofilm je nositelj analognog zapisa kojeg dobijemo fotografiranjem originalnog predloška određenim faktorom smanjenja (1:18, 24, 48, 92, 96, 700).
 - mikrofilm može izdržati i više od 100 god.

- formati:
 - sadržaj se prenosi na medij koji može biti:
 - u kolutu (do 2500 stranica).
 - 16 mm svitak za snimanje poslovne dokumentacije do formata A3.
 - 35 mm film pretežito za snimanje poslovno-tehničke dokumentacije do formata A0.
 - na kazeti.
 - u košuljicama (105×148 mm = 70 str. A4).
 - list ili mikrofiš (325 str. A4) - grupiran logički sadržaj pospremljen na formatu A6 (oko 64 stranice najmanje).
 - microplan film (on je zajedno sa microfishem nestandardni format mikofilma).

5.2.1.2. Papir i slika

5.2.2. Nositelji digitalnog zapisa

Nositelji digitalnog zapisa su magnetska vrpca, magnetski disk te optički disk.

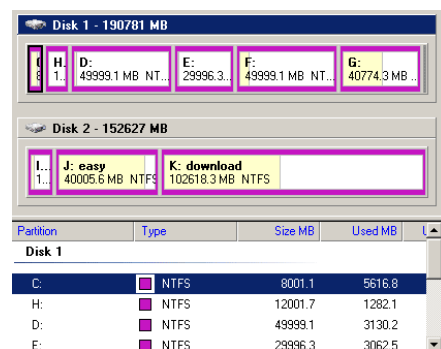
5.2.2.1. Magnetska vrpca

- najučinkovitiji medij za izradu sigurnosne kopije (backup).
- magnetska traka koja se premotava s jednog na drugi kolut.
- pohrana i organizacija podataka slična je ostalim magnetskim medijima.
- osnovni standardi:
 - **QIC standard**
 - Quarter-Inch Cartridge Drive
 - dva su standardna tipa QIC data cartridges
 - DC standard za data cartridge veličine 4x6x5/8-inch (veličina kazete)
 - MC standard veličine 3 1/4x2 1/2x3/5-inch (mini cartridge)
 - QIC-standard koristi magnetski medij (ferric oxide) i zapisuje podatke slično čvrstom disku.
 - **DAT Tape Drive standard**
 - jeftiniji od QIC i Travan traka ali skuplja je jedinica trake
 - koriste se za poslužitelje i skuplja osobna računala
 - DAT uređaji koriste dva standarda:
 - Digital Data Storage (DDS)
 - Data/DAT
 - tehnologija zapisa je digitalna i slična je snimanju na CD-e.
 - **8mm Tape Drive**
 - proizvođač Exabyte
 - kapacitet do 20G (40G s hardverskom kompresijom)
 - 8mm videotape cartridges.
 - 6M/sec prijenos Exabyte 8mm tape backup drive-a, (10M/min za DAT drive).
 - **DLT (Digital Linear Tape)**
 - Digital Linear Tape (DLT) Digital Equipment Corporation
 - kapaciteti od 35-70G (kompresirano)
 - 5-10M/sec i veći prijenos, 8mm traka
 - vrlo stalni i otporni medij
 - minimalno očekivani vije trajanja je 15,000 sati pod najtežim temperaturnim i klimatskim uvjetima
 - očekivani životni vijek od 500,000 ciklusa
 - primarno se koriste za backup mrežnih poslužitelja.

5.2.2.2. Magnetski disk

- komponente:
 - Head actuator - postavlja glave iznad zadane adrese na disku
 - Spindle motor - vrti ploče diska oko osi (5400 - 10000 RPM)
 - višestruke ploče (disks) na istoj osovinu (najčešće aluminij kao osnova) prekrivene mag. osjetljivim materijalom

- ploče se ne mogu micati
- 2 1/2-inch - prijenosna računala
- 3 1/2-inch - desktop računala
- glave za čitanje/pisanje.
- osnovne operacije su slične kao i kod mekog (floppy) diska
- glave za čitanje i pisanje se ne mogu kretati individualno kao ni diskovi
- gustoća traka je veća (> 3000 TPI)
- zapis je identičan kao i na magnetskoj disketi ili traci - promjenom polariteta uzrokovanog promjenom napona mijenja se i polaritet magnetskog polja na disku
- osobine : Interface type (IDE (Parallel ATA, Serial ATA) / SCSI), prosječno vrijeme pristupa (average seek time), kapacitet, brzina prijenosa (transfer rate), priručna memorija (cache), brzina vrtnje (RPM)
- **particije (partition)**
 - logički dio tvrdog diska koji određuje korisnik
 - particija je niz ili grupa clustera kojoj s pridjeljuje logičko ime (C, D...)
 - particija ima početni i završni sektor, broj sektora između završne i početne pozicije čine kapacitet
 - razlozi kreiranja particija:
 - mogućnost uporabe različitih OS-a na istom disku
 - ukoliko OS ne podržava veličinu cijelog diska mogu se se kirati manji dijelovi koji se posebno formatiraju
 - smanjuje se veličina klastera što utiče na iskorištenost; Ako podatak ne zauzme cijeli alocirani klaster tada preostala količina ostane neupotrebljiva pa može doći do znatnog gubitka kapaciteta. Kod diskova manje veličine veličina klastera je manja pa je optimalnije korištenje memorije
 - lakša organizacija podataka na velikim diskovima



SLIKA 5.1: ORGANIZACIJA DVAJU DISKOVA PRIMJENOM PARTICIJA

5.2.2.3. Optički disk

- osnovno:
 - 1978, Philips i Sony Corporations ujedinjuju se na razvoju audio CD
 - 1982, nastaje poznati 4.72-inch format (120mm u promjeru, 15mm rupa u sredini i debljina od 1.2mm)
 - pohranjuje 682M podataka, približno oko 333,000 stranica teksta ili 74 minuta HI-FI audio podataka (1 sec = 75 blokova, 1 blok = 2,048 bytes / 681,984,000 bytes/)
 - računalni CD disk je sličan audio CD-u, brži je nego FD ali i sporiji nego HD
- tip medija:
 - write once Read Many standard - podaci se ne mogu mijenjati
 - pohranjuju samo binarne znakove (0 i 1)
- struktura medija:
 - dimenzije : 120mm (4.75 inches) single-sided disk
 - polikarbonatni film, metallic film (aluminijски film), aluminijski film je mjesto od kuda se čitaju podaci
 - disk je podijeljen u spiralne trake s razmakom od 1.6 mikrona između krugova s gustoćom traka od 16,000 po inchu
 - udubine (pits) i ravnine (lands) dugi su od 0.9 do 3.3 mikrona
 - traka počinje od unutrašnje strane diska i završava 5mm prije kraja ruba diska (duga je oko 3 milje/650 MB)
 - podaci se čitaju u bytes (24 u jednom okviru-frame)
 - okviri čine blokove (1 blok = 98 okvira, ukupno 270.000 blokova)
 - jedan blok (1 blok = 2,352 bytes) sadrži:
 - Synchronizing bits, Identification bits, Error Correcting Code information i 2,048 bytes za podatke

- blokovi se čitaju brzinom od 75/sec pa je standardna brzina 153,600 bytes/sec ili 150K/sec.
- pisanje:
 - CD-ROM tehnologija koristi lasersko svjetlo za paljenje (oštećivanje) mikroskopskih rupica u mediju
 - podaci su pohranjeni primjenom tehnike Constant Linear Velocity (CLV)-promjenljiva brzina vrtnje i stalna brzina čitanja
 - snimanje se vrši u jednoj neprekinutoj liniji (spirali) od središta prema rubu diska
- čitanje:
 - pomoću nisko naponske laserske zrake
 - čitanje se izvodi tehnikom Constant Angular Velocity (CAV) - stalna brzina vrtnje i promjenljiva brzina čitanja
 - prijemnik signala prima slabo reflektiranu (raspršenu) ili jače reflektiranu (suženu) zraku ovisno o refleksiji površine na koju pada laserski snop
 - jaka refleksija označuje nepostojanje udubine (land), slabija refleksija označuje postojanje udubine (pits)
 - nakon dijela za sinkronizaciju slijedi dio koji ukazuje na strukturu diska traženje podataka (table of contents)> podaci se traže duž spirale tako dugo dok adresa podatka ne prođe lasersku zraku
- kapacitet i trajnost zapisa:
 - 1 GB, ali je iskorišten kapacitet do 640 MB radi potrebe stvaranja redundancije
 - trajnost je nekoliko desetaka godina ovisno o uvjetima čuvanja
- dijelovi:
 - laserska dioda - emitira zraku male snage
 - servo motor - pozicionira zraku na točnu traku pomičući zrcalo
 - povratna zraka fokusira se na lećama ispod ploče diska i šalju prema beam splitteru
 - beam splitteru usmjerava povratno (reflektirano) svjetlo na leću za fokusiranje
 - photo detector pretvara svjetlosne signale u električne signale a dobiveni se impulsi dekodiraju i šalju središnjoj jedinici

5.3. Organizacija zapisa podataka

- Organizacija zapisa podataka nam govori o tome kako su podaci zapisani na nekom mediju za pohranu podataka.
- Tako postoje **slijedna i rasuta** organizacija zapisa podataka.

SLIJEDNA ORGANIZACIJA ZAPISA:

- Potječe sa magnetskih vrpce, kod kojih je to i jedina moguća organizacija zapisa.
- Podaci se zapisuju u kontinuirane (slijedne, sekvencijalne) cjeline – sektor do sektora, klaster do klastera.

RASUTA ORGANIZACIJA ZAPISA:

- Omogućava dislocirano zapisivanje podataka po mediju.
- Mogućnost iskorištenja svakog slobodnog prostora na mediju.
- Veća fleksibilnost.
- Obje spomenute organizacije zapisa su podržane kroz datotečni sustav.

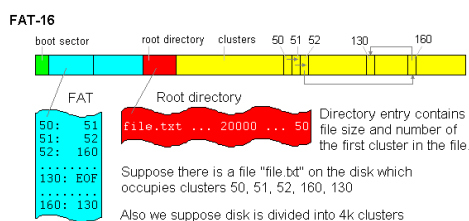
5.4. Datotečni sustavi

- Datotečni sustav je dio operacijskog sustava a služi upravljanju permanentnim podatkovnim objektima tj. objektima čije vrijednosti traju dulje nego procesi koji ih kreiraju i koriste.
- Ti se permanentni podaci čuvaju u datotekama na sekundarnim jedinicama poput tvrdih diskova
- Datoteke su organizirane u direktorije.
- Dakle, svaki datotečni sustav se sastoji od:
 - datoteka,
 - direktorija odnosno tablica, te
 - procedura za manipuliranje datotekama.

- Datoteke se mogu razvrstati na tipove koji se mogu označiti sufiksom uz ime datoteke kao npr. .bin, .exe, .doc i sl.
- Interpretacija sadržaja datoteke je u nadležnosti primjenskih programa.
- Datotečni sustav provodi operacije koje kreiraju ili brišu datoteku, otvaraju datoteku prema njezinom imenu, čitaju sljedeći objekt iz otvorene datoteke, zapisuju objekt u otvorenu datoteku ili zatvaraju datoteku.
- Realizacija datotečnog sustava može se promatrati kroz dva sloja:
 - **Prvi ili niži sloj** obuhvaća operacije neposrednog pristupa do podataka na mediju na kojem su zapisani. On nije primarno vezan uz rad s podacima datoteka, nego uz organizaciju njihova zapisa na mediju. Naziva se još i fizički sloj, jer obavlja fizičku organizaciju datotečnog sustava.
 - **Drugi ili viši sloj** nadograđuje se na niži sloj. On provodi generičke operacije kreiranja, brisanja, otvaranja i zatvaranja datoteka te postavljanja, čitanja i zapisivanja podataka u otvorene datoteke te organizaciju datoteka u direktorije.

5.4.1. FAT (File Allocation Table)

- Potječe iz vremena MS-DOS operacijskih sustava.
- Naziv dolazi od odvajanja zapisa o osobinama datoteke od samih podataka datoteke. Pristup do podataka datoteke polazi od naziva datoteke pomoću kojeg se pronalazi odgovarajuće zaglavlje tražene datoteke. Time se dolazi do informacije o mjestu zapisa početnog bloka podataka datoteke. Podaci datoteke nisu zapisani u kontinuiranim blokovima zbog fragmentacije podataka koja je uvjetovana dinamikom rada. O tome više riječi kasnije.
- FAT datotečni sustav svaku logičku disk jedinicu dijeli na četiri dijela:
 - BOOT – područje na jedinici koje nosi informaciju o samoj jedinici te programski kôd pomoću kojeg se može ostvariti postupak učitavanja operacijskog sustava,
 - FAT – tablica alokacije datoteke – tu su upisani lanci alokacije za svaku datoteku jedinice. Uvijek postoje dvije identične kopije FAT područja.
 - ROOT- ishodišni direktorij koji predstavlja bazu za grananje direktorija, a služi i za upis podataka o datotekama koje su smještene u tom direktoriju,
 - FILES – područje za smještaj podataka (datoteka i direktorija). Podaci se zapisuju u alokacijske jedinice pod nazivom klasteri. Klaster čini jedan ili više uzastopnih logičkih sektora jedinice (moraju biti potencija broja 2!).



SLIKA 5.2: FAT PRIMJER

- Informacija o podjeli jedinice na BOOT, FAT, ROOT i FILES nalazi se u području BOOT. To se područje još naziva i boot sektor jer u pravilu zauzima samo jedan sektor.
- Postoje više vrsta FAT datotečnog sustava ovisno o veličini klastera. Tako razlikujemo FAT12, FAT16 i FAT32 datotečne sustave. Ako je za opis klastera potrebno 16 bitova onda se radi o FAT16 sustavu itd. Analogno, FAT16 sustav može adresirati $2^{16} = 65536$ klastera.

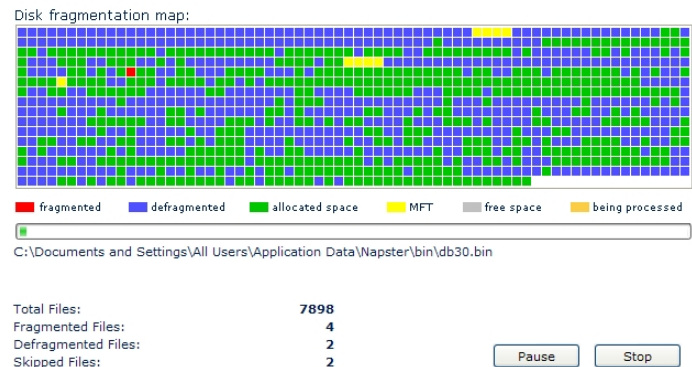
5.4.2. NTFS (NT's Native File System)

- Javlja se sa MS-Windows NT operacijskim sustavom i predstavlja alternativu FAT datotečnom sustavu.
- Sadrži datoteku koja se naziva MFT (Master File table, glavna tablica datoteka).
- Svaka datoteka, uključujući i MFT, ima u toj datoteci jedan zapis koji je opisnik datoteka.
- Jedan MFT zapis ima veličinu jedne skupine tj. veličinu klastera. Male datoteke smještaju se unutar MFT zapisa.
- Za veće datoteke u MFT se pohranjuju **metapodaci** koji opisuju karakteristike datoteke i direktorija i njihov smještaj na disku.

- Smještaj datoteke se određuje pomoću Virtualnog rednog broja skupine (Virtual Cluster Number, VCN), a pronalaze se uzastopne skupine sektora i dodjeljuju datoteci.
- NTFS se referira na disk uporabom Logičkog broja skupine (Logical Cluster Number, LCN). LCN je redni broj skupine od početka do kraja diskovnog prostora predviđenog za smještaj datoteka. Fizička adresa sektora na disku može se izračunati na temelju faktora skupine i početne adrese volumena.

5.5. Defragmentacija

- Defragmentacija je proces zapisivanja dijelova datoteke u niz uzastopnih sektora na tvrdom disku u cilju povećanja brzine pristupa podacima.
- Prilikom ažuriranja podataka odnosno datoteka, datotečni sustav najčešće sprema podatke na prvo slobodno mjesto(sektor) na disku s obzirom na položaj glave za čitanje/pisanje. U pravilo taj sektor nije ni blizu sektora na kojem se nalazi ostatak podataka iz datoteke. Iz tog razloga kad su podaci fragmentirani, datotečni sustav mora pretraživati površinu diska svaki put kad se datoteka otvori da nađe sve njezine djelove.
- Fragmentirana datoteka – datoteka čiji su dijelovi razbacani(rasuti) po površini diska.
- Fragmentacija se događa kad se datoteke brišu s diska ili kad se dodaju nove. To usporava pristup disku i u pravilu degradira performanse čitavog sustava. Iz tog je razloga disk potrebno defragmentirati različitim alatima poput Disk Defragmentera koji dolazi kao alat Microsoft Windows operacijskih sustava ili pak Norton Speed Disk alata.



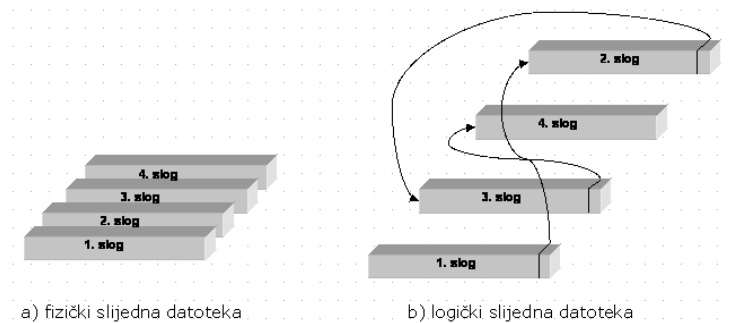
SLIKA 5.3: PRIMJER DEFRAGMENTACIJE DISKA

5.6. Datoteka

- Datoteka je skup istovrsnih podataka (slogova) obuhvaćenih nekim zajedničkim kriterijem i smještenih u memoriji računala.
- Atribut ili skup atributa koji jednoznačno identificiraju svaki slog u datoteci naziva se ključem.
- Način upisivanja slogova u datoteku ovisi o njejoj organizaciji. S obzirom na fizičku organizaciju slogova u datoteci razlikujemo:
 - **slijednu (sekvencijalnu) datoteku,**
 - **direktnu (relativnu) datoteku, te**
 - **indeksnu datoteku.**
- Više riječi o tome u narednim poglavljima.
- Danas korisnik niti ne zna način na koji su datoteke organizirane jer ga suvremeni operacijski sustavi oslobađaju većine tih briga.
- S obzirom na strukturu razlikujemo:
 - **Linearne strukture.** Svaki element skupa ima samo jedan element koji mu prethodi i samo jedan koji iza njega slijedi. Takav poredak odgovara listi.
 - **Hijerarhijske strukture.** Jedan element povezan je samo s jednim elementom više razine i sa više elemenata niže razine. Elemente koji nisu povezani ni sa jednim elementom niže razine zovemo listovi.
 - **Mrežne strukture.** Elementi mogu biti povezani sa više elemenata više i niže razine.

5.6.1. Slijedna ili sekvencijalna datoteka

- Slogovi se upisuju i memoriraju jedan iza drugoga. Oni su u memoriji fizički ili logički poredani jedan za drugim.
- Zbog te povezanosti razlikujemo **fizičku i logičku organizaciju** slijedne datoteke.
- Fizička organizacija podrazumijeva i fizičko povezivanje slogova. Dakle, tu su slogovi fizički slijedno povezani, odnosno smješteni su na susjednim adresama.
- Logička organizacija omogućuje slogovima da ne budu fizički smješteni jedan iza drugoga, no tada se povezanost ostvaruje pomoću pokazivača (pointera).



SLIKA 5.4: DVA TIPA SLIJEDNE DATOTEKE

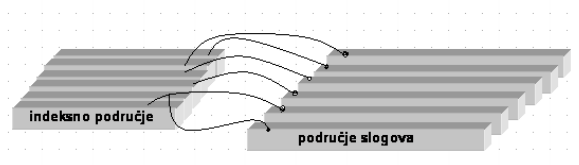
- Dobre osobine sekvencijalnog organiziranja:
 - jednostavnost rukovanja podacima,
 - u redoslijednoj obradi pristup je brz,
 - magnetske vrpce služe kao dobri medij za pohranu takvih datoteka
 - vrlo dobro iskorištenje memorije.
- Loše osobine sekvencijalnog organiziranja:
 - nemogućnost prekoredne, direktne obrade podataka,
 - podatke je potrebno uvijek na neki način sortirati prije kreiranja datoteke,
 - ažuriranje podataka zahtijeva sveobuhvatno ispitivanje.

5.6.2. Direktna ili relativna datoteka

- Slogovi se pohranjuju na relativnim adresama, počevši od adrese 1 do n. Slog s ključem 1 memorira se na relativnoj adresi 1, slog s ključem 2 na relativnoj adresi 2 itd.
- Stvari u praksi nisu toliko jednostavne, te je teško naći organizaciju u kojoj vrijedi ključ=adresa.
- Najčešće se koristi neki algoritam transformacije ključa u adresu sloga. Ukoliko znamo ključ sloga i algoritam njegove transformacije u adresu, istom slogu možemo direktno pristupiti, bez prethodnog pretraživanja preko ostalih slogova u datoteci.
- Direktna organizacija se isto kao i indeksno-sekvencijalna upotrebljava kod tvrdih diskova, mada je moguća i kod ostalih memorija.

5.6.3. Indeksna datoteka

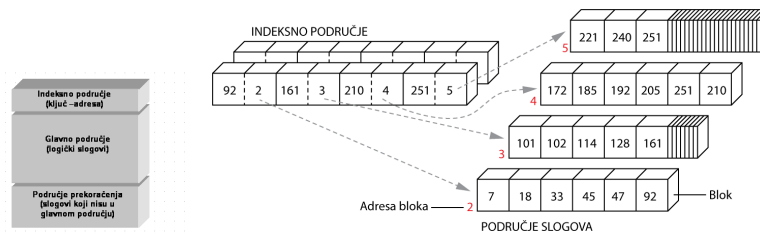
- Takva se datoteka sastoji od **indeksnog područja i područja slogova**.
- Indeksno područje sadrži indeks ili kazalo. Uz svaki ključ u kazalu nalazi se adresa njegovog sloga u području podataka.
- Pronalaženje slogova se odvija od indeksa pronalaženjem adrese sloga, a nastavlja se u području slogova pronalaženjem sloga na dobivenoj adresi.
- Dakle, prvo se mora pronaći indeks, a onda slog. Iz tog razloga indeksno se područje najčešće organizira kao direktna datoteka u kojoj je pojedinom indeksu moguće brzo pristupiti.



SLIKA 5.5: INDEKSNA DATOTEKA

5.6.4. Indeksno-sekvencijalna datoteka

- Predstavlja **kompromis** između sekvencijalne i rasute organizacije podataka.
- Istovremeno omogućuje pojedinačnu obradu slogova jer ima karakteristike indeksne datoteke, te skupnu obradu svih ili veće skupine slogova jer ima karakteristike slijedne datoteke.
- U toj je datoteci područje podataka organizirano kao slijedna datoteka koja se sastoji od blokova u koje je moguće smjestiti više slogova. Svaki blok dobiva svoju indeksnu vrijednost u indeksu. Do bloka se pristupa direktno, a do sloga u bloku slijedno.



SLIKA 5.6: INDEKSNO-SEKVENCIJALNA DATOTEKA

- Indeksno područje je zasebni, fizički odvojen skup uređenih parova (ključ – adresa).
- Glavno područje (područje podataka) služi za smještaj logičkih slogova. Slogovi su uređeni i upisani po redoslijedu svojih ključeva. Područje prekoračenja jeste memorijsko područje u kojem se nalaze slogovi koji, prilikom ažuriranja datoteke, nisu mogli biti smješteni u glavnom području datoteke.

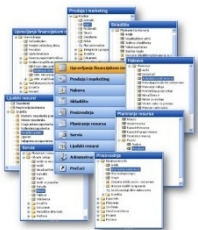
5.7. Baze podataka

- Baza podataka je skup ustrojjenih, logički povezanih zapisa, podataka ili datoteka, pohranjen u memoriji računala sa mogućnošću pretraživanja. Zapisi s podacima su u nekoj međusobnoj vezi.
- Isto tako možemo reći i da je baza podataka skup međusobno ovisnih podataka, spremljenih bez redundancije (zalihosti), koji služe jednoj ili više aplikacija na optimalan način, gdje su podaci neovisni o programima kojima se obrađuju i gdje postoji kontrolirani pristup podacima.
- Cilj uvođenja baze podataka je ubrzanje računalnih aplikacija, smanjenje troškova njihova održavanja, te opskrbljivanje krajnjih korisnika podacima potrebnim da bi svoj posao obavili što je moguće efikasnije.
- Da bi baza podataka uopće bila funkcionalna mora postojati **kreator** baze podataka, zatim **administrator** baze podataka, te **krajnji korisnik** koji najčešće bazu samo pretražuje.
- Ovisno o vrsti i namjeni podataka u bazi podataka kao i načinima korištenja podataka razlikujemo:
 - **Baze formatiranih podataka.** Najčešće se koriste u poslovnim primjenama.

Ovisno o načinu izgradnje, razlikujemo baze podataka građene po hijerarhijskom ili mrežnom modelu, te suvremene baze podataka građene po relacijskom, objektnom ili dimenzijskom modelu.

- **Baze neformatiranih podataka.** Sadrže različite tekstualne ili multimedijalne podatke. Pronalaženje dokumenata obavlja se navođenjem ključnih riječi za koje očekujemo da se nalaze u dokumentu.

- **Baze znanja.** Sadrže znanje prikazano u različitim oblicima, npr. U obliku pravila, semantičkih mreža, okvira ili scenarija.



SLIKA 5.7: BAZA PODATAKA

5.7.1. Relacijske baze

- Podaci se u relacijskoj bazi podataka nalaze u tablicama (relacijama). Načelo relacijske baze podataka jeste da korisnik vidi bazu kao **skup tablica** podataka, a rezultat svake operacije nad bazom podataka je također tablica podataka.
- Definiciju jedne relacije nazivamo **relacijskom shemom**. Ona se sastoji od:
 - Naziva relacije
 - popisa atributa (obilježja) relacije.
- Redove u relaciji nazivamo n-torke, a stupce atributima.

- Osnovne značajke relacije:
 - ne postoje dva jednaka stupca, kao ni dva jednaka retka, i
 - redosljed stupaca i redova nije bitan.
- Atribut, odnosno skup atributa pomoću kojih jednoznačno identificiramo svaki redak naziva se ključem. **Ključ** koji se sastoji od minimalnog broja elemenata pomoću kojih jednoznačno identificiramo svaki redak naziva se primarnim ključem (u shemi ga označavamo podcrtavanjem).
- Strani ili vanjski ključ jeste atribut ili skup atributa koji služi za međusobno povezivanje relacija. On uvijek pokazuje na primarni ključ neke druge relacije i taj se odnos naziva referencijalnim integritetom.
- Da bismo stvorili relacijsku bazu podataka potrebno je izraditi:
 - Konceptualni opis podataka:
 - Ustanoviti objekte poslovnog sustava.
 - Odrediti attribute čije ćemo vrijednosti bilježiti, te ustanoviti međusobno povezanost objekata.
 - Logički opis podataka:
 - Utvrđiti slogove i polja.
 - Odrediti raspored podataka u tablicama.
 - Fizički opis podataka:
 - Raspored podataka na fizičkom mediju.
 - Pristup do podataka.
- Skup operacija koje se provode nad relacijama odnosno tablicama nazivamo **relacijskom algebrom**.
 - Operacija selekcije* predstavlja selekciju n-torki jedne relacije koje udovoljavaju zadanom kriteriju.
 - Operacija projekcije* predstavlja selekciju stupaca jedne relacije.
 - Operacija spajanja* od dvije relacije formira novu relaciju spajanjem n-torki prve relacije s n-torkama druge relacije koje imaju istu vrijednost atributa spajanja.
- Za rad s bazom podataka koriste se tzv. **upitni jezici**. Najčešće korišten jezik za rad s bazom podatka je SQL.

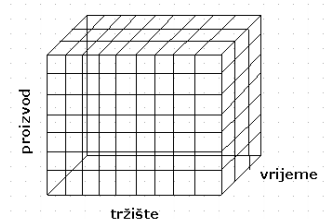
STORE			PRODUCT		
Store_Idx	City	Region	Product_Idx	Description	Brand
1	San Jose	West	1	Mountain Dew	MT Dew
2	Chicago	East	2	Top Shelf	MT Dew
3	Chicago	East	2	Essence of Freshley	Paradee Inc.
4	Los Angeles	West	3	Happy Cola	MT Dew
5	San Francisco	West	4	Pop Fusion	MT Dew
6	Phoenix	East	5	Top Shelf	MT Dew
6	Phoenix	East	7	Front Load 2L Glass	Paradee Inc.
6	Phoenix	East	8	Happy Cola	Big Soda

SALES_FACT					
Store_Idx	Product_Idx	Sales	Cost	Profit	
1	1	230	115	115	
2	2	557	281	276	
3	2	770	376	394	
4	3	472	184	288	
5	4	1100	430	670	
6	5	1421	631	790	

SLIKA 5.8: PRIMJER RELACIJSKE BAZE PODATAKA

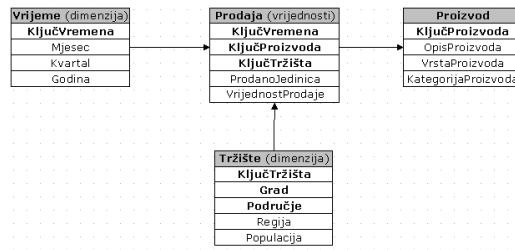
5.7.2. Skladišta podataka i dimenzijske baze

- Skladišta podataka
 - Skladištenje podataka je novi trend u obradi podataka, a usmjereno je na pravovremeno dobivanje relevantnih informacija potrebnih u postupcima **odlučivanja**.
 - Za pohranu podataka koristi se dimenzijska struktura podataka ugrađena kroz dimenzijsku bazu podataka.
 - Potreba za skladištenjem podataka se javlja zbog turbulentne okoline u kojoj se poduzeća tj. organizacije nalaze kako bi ostala konkurentna na tržištu.
 - Skladištenje podataka se vezuje uz aktivno pronalaženje i nuđenje informacija menadžeru, koristeći postupke **analitičke obrade, kopanja (rudarenja) podataka i otkrivanja znanja**.
 - Da bi se omogućilo pravovremeno i točno pružanje informacija, potrebno je izgraditi **sustav za potporu odlučivanju (DSS)**.
 - Skladište podataka je skup integriranih i obogaćenih podataka organizacije na temelju kojih se izgrađuje sustav za potporu odlučivanju.
 - Skladište podataka je za razliku od baze podataka usmjereno na dobivanje informacija potrebnih za donošenje odluka. Skladište je najčešće subjektivan, integriran, nepromjenljiv, te vremenski orijentiran skup podataka organiziranih tako da posluže pri odlučivanju.
- Dimenzijske baze podataka
 - Dimenzijska struktura podataka omogućuje vizualizaciju podataka. Tako neku pojavu možemo promatrati kroz tri dimenzije, kako je prikazano na slici, i to npr. kroz proizvod, tržište i vrijeme.
- Dimenzija ili perspektiva određena je pojavom kroz koju se prati posao.



SLIKA 5.9: DIMENZIJSKA BAZA PODATAKA

- Elementi dimenzije su pozicije ili članovi.
- Pozicije mogu biti uređene hijerarhijski, pa se u okviru jedne dimenzije pojava može pratiti detaljizirano (analitički) i agregirano (sintetički). Postupak silaženja u područje detaljnih podataka naziva se još i "svrdlanje".
- Dimenzijski podaci ugrađuju se kroz **zvjezdastu shemu**.
- Vrijednosna tablica dominantna je tablica u zvjezdastoj strukturi. Ona povezuje sve dimenzijske tablice.



SLIKA 5.10: ZVJEZDASTA SHEMA

- Postupci nad dimenzijskom bazom podataka:
 - **Rotacija** – promjena orijentacije na ekranu ili papiru
 - **Pivotiranje** – isticanje važnijih dimenzija u prvi plan, a ostalih u drugi plan tj. pozadinu.
 - **Selekcija** – selektiranje pozicija jedne dimenzije.
 - **Isijecanje** – uzimanje jednog isječka odnosno selektiranje jedne pozicije jedne dimenzije.
 - **Detaljiziranje i agregiranje** po hijerarhiji pozicija jedne dimenzije.
 - **Analitička obrada podataka** – omogućava korisniku da s lakoćom obavlja analitičke upite, odnosno obrade poput modeliranja i proračunavanja, analize vremenskih serija itd.

KRAJ.