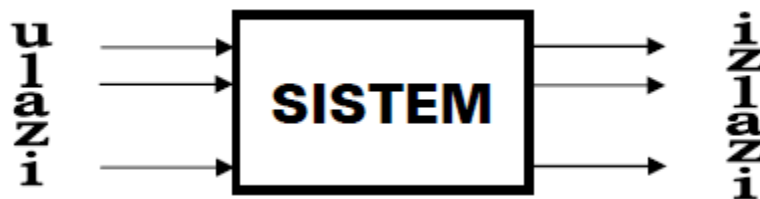


1. Sistem

Sistem je skup elemenata koji ostvaruju neku funkciju i interakciju sa okolinom. Dakle, sastoji se od skupa elemenata koji nisu samo povezani međusobno, već moraju biti povezani sa okolinom da bi sistem mogao da dobije informacije iz okoline i da bi obezbeđivao informacije za okolinu. To znači da sistem ima ulaze preko kojih dobija informacije iz okoline i ima izlaze preko kojih, nakon obrade ulaznih signala, utiče na okolinu. Skup elemenata koji ne utiče na okolinu predstavlja bezvredan sistem.



Sl. 1-1 Opšta predstava sistema

Postoje sistemi koji su nastali evolucijom i sistemi koji su nastali delovanjem čoveka. Naša svakodnevna životna iskustva su nam omogućila da razlikujemo primere prirodno nastalih i proizvedenih sistema. Jedan primer prirodno nastalog sistema je ljudsko biće u društvenom okruženju. U tom slučaju ulazi u sistem su ostvareni kroz jedno ili više čovekovih čula, kao što su čula vida, sluha, mirisa, osećanja, ili dodira. Izlazi su obezbeđeni kroz sposobnosti čoveka da govori, piše, ili pokreće delove tela. Funkcionalne sposobnosti ljudskog bića variraju kod svake individue, i prostiru se u veoma širokom spektru. Tako, na primer, profesor svojim sposobnostima treba uspešno da vodi interakciju sa studentima radi stvaranja njihovih znanja a zatim ispitivanja kako su dobro naučili. Primeri napravljenih sistema su brojni. Samo u industriji se mogu prepoznati mnogobrojni sistemi, koji imaju za cilj da stvaraju materijalna dobra.

1.1 Elektronski sistem

Danas je gotovo nemoguće zamisliti proizveden sistem bez elektronskih elemenata-komponenti. Naime veliki broj sistema, svrstan u grupu elektronskih sistema, sačinjen je od elektronskih komponentama. Za to je zaslužan razvoj mikroelektronike i tehnologije izrade elektronskih kola, učinivši ekonomski opravdanim najraznovrsnije primene elektronskih sistema u gotovo svim oblastima ljudske delatnosti. Raznovrsnost elektronskih sistema, od

jednostavnih realizovanih sa nekoliko diskretnih elemenata pa do visoko prefinjenih fleksibilnih sistema realizovanih visoko integrisanim komponentama složene funkcionalnosti, nameće potrebu sistematizovanja elektronskih sistema.

Nadalje će termin *sistem* podrazumevati da se radi o elektronskom sistemu jer su upravo oni predmet daljih razmatranja.

1.2 Tipovi elektronskih sistema

1.2.1 Podela po pririodi obrađivanih signala

Prva i najuobičajena podela elektronskih sistema je prema vrsti informacija koje obrađuje, ili kako se obično kaže prema domenu u kome izvršavaju svoju funkciju. Tako se mogu razlikovati:

- Analogni elektronski sistem,
- Digitalni elektronski sistem, i
- Analogno-digitalni elektronski sistem.

Analogni sistemi obrađuju kontinualne informacije, po čemu su bliski sa prirodnim veličinama. Iako su prvobitni elektronski sistemi razvijani kao analogni jer je tehnologijom diskretnih elektronskih komponenata praktično to bilo jedino moguće, neprestan razvoj analognih sistema je uslovljen njegovim svojstvom da kontinualnom obradom u osnovi ostvaruju rad u realnom vremenu. Tako i pored osnovnog nedostatka, osetljivosti na smetnje, danas se analogna obrada koristi za izuzetno brze pojave (na primer kada se radi o obradi visokofrekventnih signala reda GHz) i gde digitalnom sistemu treba nedopustivo mnogo vremena za obradu.

Digitalni sistem je sistem koji obrađuje diskretne informacije. Diskretni entitet koji čini ovu informaciju može predstavljati bilo šta, od prostih aritmetičkih intedžera, slova azbuke ili drugih apstraktnih simbola do vrednosti napona, pritiska ili bilo koje fizičke veličine. Digitalnom sistemu u procesiranju informacije nije bitno šta predstavljaju ovi entiteti. Dakle, *digitalni sistem kao ulaz prima digitalnu informaciju koja predstavlja brojeve, simbole ili fizičke veličine, obrađuje ovu ulaznu informaciju na neki specifičan način, i generiše digitalni izlaz.*

Postoje tri osnovna razloga što se tradicionalni analogni sistemi sve više zamenjuju digitalnim sistemima. To su veća fleksibilnost, veća pouzdanost i niža cena digitalnih sistema.

Danas najšire primenjeni digitalni sistemi su bazirani na mikroracunarima. To je zbog toga što mikroracunar uz svoju nisku cenu ima i osobinu univerzalnosti, s obzirom da izvršava samo jednu globalnu funkciju - obrađuje digitalne informacije. U velikom broju primena mikroracunara, mikroracunar je potreban za obradu informacija koje se odnose na fizičke veličine kao što su pritisak ili temperatura. S obzirom da pojave u prirodi nisu digitalne (sve dok se razmatranje ne odnosi na nivo kvantne mehanike), fizičke veličine kao vreme, temperatura ili bilo šta drugo moraju biti prevedene u digitalni oblik da bi mogle da se obrađuju računom. Uobičajeni način za ovo je da se fizička veličina koja se obrađuje najpre prevede u napon ili struju. Ovo se radi korišćenjem pretvarača - elementa koji prevodi energiju koja dolazi do njega u jednom obliku u energiju drugog oblika kao njegov izlaz. Termospreg je dobar primer temperaturnog pretvarača: on daje izlazni napon srazmeran temperaturi ambijenta u kome se nalazi. Ovaj izlazni napon postaje analog temperature elementa. S obzirom da računar radi samo sa diskretnim veličinama, koje se obično mogu pridružiti brojevima, kontinualna promenljiva koja predstavlja fizičku veličinu mora se prvo konvertovati u digitalni oblik. Znači jedan elektronski sistem sastoji se i iz analognog i iz digitalnog dela.

Analogno-digitalni sistem je naime sistem koji poseduje i analogni deo radi povezivanja sa prirodnim kontinualnim veličinama u okolini i digitalni radi obrade koja se izvršava u digitalnim uređajima, odnosno mikroracunarima i računarima.

Iako se elektronski sistemi danas baziraju na mikroprocesorima - mikroracunarima, gotovo uvek sadrže i sklopove analognih kola. To je prirodno s obzirom da u okruženju sistema egzistiraju analogne veličine, kao na primer kod sistema za upravljanje i nadzor različitih procesa. S druge strane i u sistemima koji se za ostvarivanje osnovne funkcije moraju projektovati kao analogni, mikroprocesori - mikroracunari gotovo uvek nađu mesto. U njima oni ostvaruju funkciju nadzora i upravljanja nad analognim elektronskim uređajima i elementima unutar sistema, tako da se povećava sigurnost u radu sistema i olakšava njegova primenljivost. Tako se danas sve više gubi podela na analogne i digitalne sisteme, već se govori o blokovima za obradu analognih i digitalnih informacija unutar sistema.

1.2.2 Podela po nameni

- Elektronski sistemi opšte namene (mogu se koristiti za više različitih primena)
- Elektronski sistemi specifične namene (projektovani za konkretnu primenu):
 - Sistem za prikupljanje i obradu podataka - Data Acquisition System,

- Sistemi za upravljanje, nadzor,
- Informacioni sistemi,
- Nove potklase - ekspertni, itd.

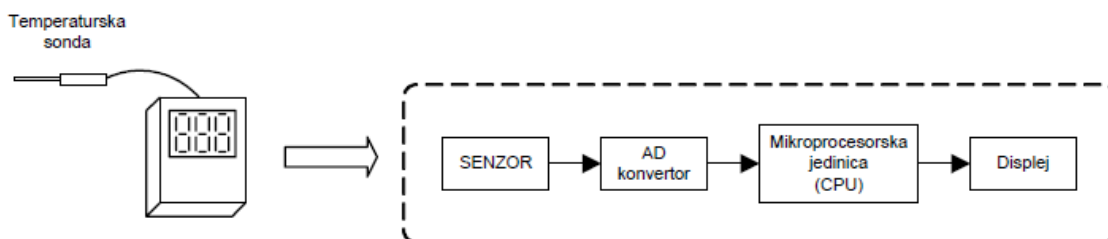
1.2.3 Podela prema vremenu potrebnom za izvršenje zadataka

- Sistemi koji ostvaruju obradu u realnom vremenu (RTS - *real-time system*)
- Sistemi koji nemaju specificirano vreme obrade

Kada se od računarskog sistema zahteva da pribavlja podatke, daje podatke, ili ima interakciju sa okolinom u unapred određenom vremenskom intervalu, za njega se kaže da je *računarski sistem za rad u realnom vremenu (real-time computer system)*. Tako gledajući, može izgledati da su svi računarski sistemi na neki način sistemi za rad u realnom vremenu. Međutim, pravi sistemi za rad u realnom vremenu su oni koji upravljaju procesima i mehanizmima kod kojih je sa stanovišta nepoželjnih trajnih havarijskih posledica neophodno u određenom vremenu ili trenutku preduzeti neku akciju - aktivnost. Drugim rečima i u najgorem slučaju računarski sistem mora da u definisanom vremenu da odgovarajući odziv, tj. zadovoljava definisano vreme odziva.

PRIMER 1.

Razmotrimo primer jednostavnog RTS-a čija je struktura prikazana na Sl. 1-2. Radi se o uređaju za merenje temperature i prikazivanje rezultata merenja na cifarskom displeju. Sistem se sastoji od: temperaturske sonde, A/D konvertora, mikroprocesora i cifarskog displeja. Rad sistema (program) je organizovan tako da se u “beskonačnoj” petlji, jedna za drugom, neprekidno obavljaju tri aktivnosti: očitavanje digitalizovana vrednost izmerene temperature, konverzija očitane brojne vrednosti u oblik pogodan za prikaz na displeju i prikazivanje rezultat na cifarskom displeju.



Sl. 1-2 Jednostavan RTS

DO FOREVER

{

1. Očitati digitalizovanu vrednost izmerene temperature (npr. 12-bitna neoznačena vrednost);
2. Konvertovati pročitano vrednost u format pogodan za prikaz na displeju (npr. 3 BCD cifre, XY.Z⁰C)
3. Postaviti novi prikaz na displej.

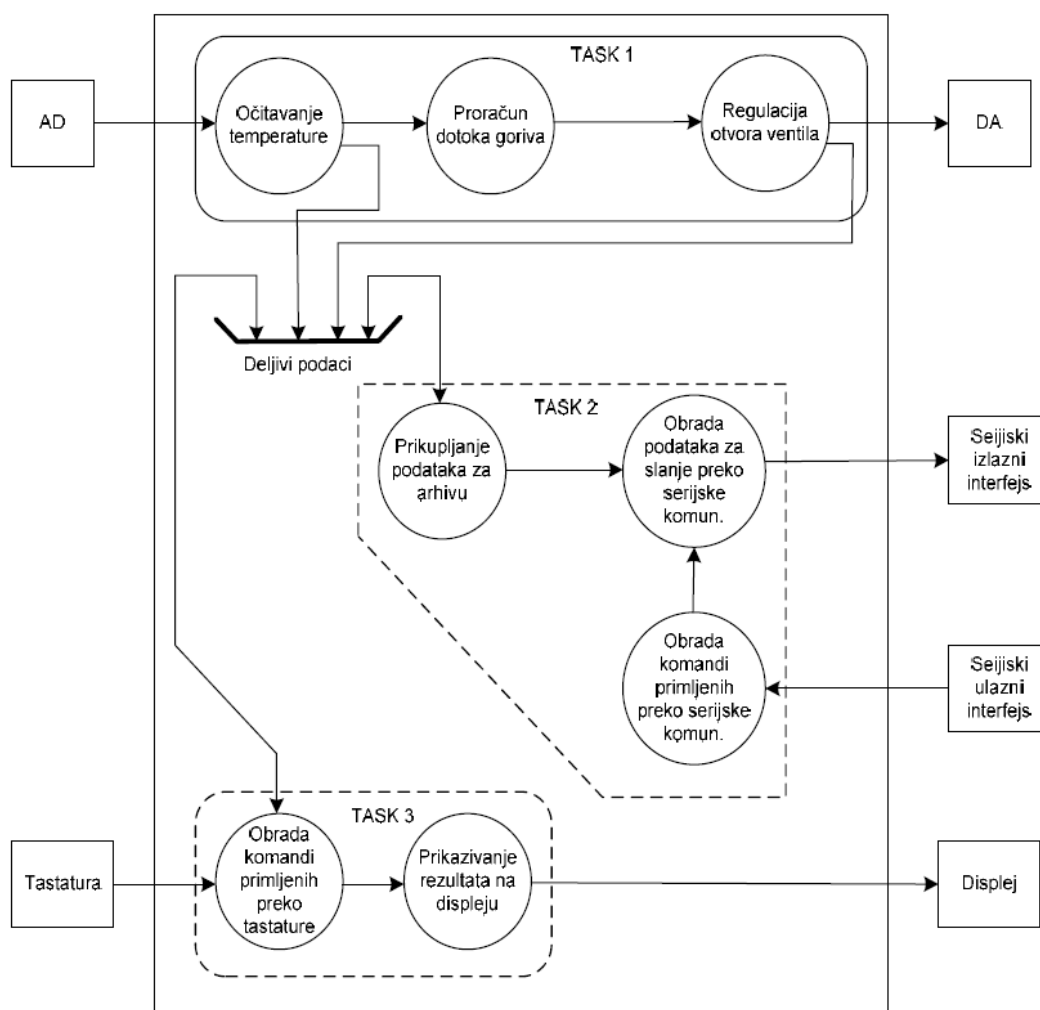
}

U ovom primeru se radi o sekvencijalnom (*single-thread*) programu, tj. program se sastoji od jedinstvene, neprekidne sekvence instrukcija koje se izvršavaju u “beskonačnoj” (ili kontinualnoj) petlji. Naravno, sekvencijalnost programa ne isključuje korišćenje potprograma i procedura. (U konkretnom primeru, programer bi mogao da koristi posebne potprograme za očitavnje AD konvertora i postavljanje novog prikaza na displej.) Kod ovog sistema ne postoje stroga vremenska ograničenja, s obzirom da se rezultat dobija brzo, što dodatno pojednostavljuje projektovanje softvera.

Međutim, iako se radi o jednostavnom sistemu (hardverski i softverski), čak i pod pretpostavkom da se za razvoj softvera koristi neki viši programski jezik (npr. C), programer mora dobro da poznaje strukturu i način funkcionisanja hardvera. tj. projektant mora biti ekspert kako za hardver tako i za softver.

PRIMER 2.

Razmotrimo, sada, jedan nešto složeniji RTS (Sl. 1-3), koji je po strukturi sličan najvećem broju RTS-a niske i srednje složenosti. Radi se o sistemu za upravljanje kotlom na tečna goriva. Temperatura se meri temperaturskim senzorom, a rezultujući analogni signal se digitalizuje uz pomoć A/D konvertora. Mikroprocesor izračunava signal greške (razliku između zadate i izmerene temperature) i na bazi toga izračunava upravljački signal koji se preko D/A konvertora vodi do ventila, a čija vrednost određuje otvor ventila, tj. protok goriva (funkcija koja odgovara zadatku TASK 1). Pored ovog glavnog zadatka, od sistema se zahteva da obavi i veći broj sekundarnih zadataka. Prvi se odnosi na interakciju sa operaterom, koji posredstvom tastature i displeja treba da ima mogućnost zadavanja temperature i uvida u sve druge informacije o radu sistema (zadatak TASK 3). Drugo, sistem treba da arhivira podatke koje se odnose na rad sistema, kao što su aktivnosti operatera, izmerene vrednosti temperature. Treće, sistem treba da ima mogućnost serijske komunikacije sa nadređenim računarom, u cilju prenosa arhiviranih podataka i upravljanja sa udaljenog mesta) - odgovara zadatku TASK 2.



Sl. 1-3 Sistem za upravljanje kotlom na tečna goriva

Mada je po hardverskoj strukturi sistem tek nešto složeniji od sistema iz prethodnog primera, softver razmatranog mikroprocesorskog sistema je značajno komplikovaniji, kako u pogledu funkcije tako i u pogledu unutrašnje organizacije. Organizacija softvera u vidu sekvencijalnog programa više nije najbolje rešenje. Problem predstavljaju zahtevi za obavljanjem većeg broja različitih zadataka, od kojih se neki iniciraju “*asinhrono*” (tačan trenutak kada se javlja potreba za izvršenjem zadatka nije poznat) pod uticajem događaja iz okruženja (kao što je npr. zadatak obrade komande zadate preko tastature ili zahtev za serijskom komunikacijom), dok se drugi startuju periodično u fiksnim, unapred poznatim vremenskim intervalima (*sinhrono*), kao što je npr. zadatak očitavanja temperature u kotlu i izračunavanje vrednosti upravljačkog signala za ventil. Pod ovim uslovima, mogu se javiti situacije kada je potrebno istovremeno (tj. *konkurentno* ili *paralelno*) izvršavati dva ili više zadataka, npr. ako se u toku izračunavanja vrednosti upravljačkog signala, javi zahtev sa

tastature. S obzirom da se radi o jednoprocorskom sistemu, u jednom trenutku može biti izvršavan samo jedan zadatak, dok svi ostali zadaci koji su spremni za izvršenje moraju da čekaju da CPU postane slobodan (tzv. *kvazi-konkurentnost*). Sistemi koji podržavaju kvazikonkurentnost su *multitasking* sistemi. Problem multitaskinga se uobičajeno rešava pomoću *prekida (interrupt-a)*, tako što se zadaci realizuju kao rutine za obradu prekida iniciranih odgovarajućim događajima (npr. pritisak na dirku tastature, prekid od tajmera, prekid od komunikacionog interfejsa).

Odlaganje izvršenja pojedinih zadataka zbog zauzetosti CPU-a može da dovede do otkaza sistema ukoliko je odziv koji se određuje zadatkom vremenski kritičan (npr. upravljački signal za ventil mora biti izračunat unutar vremena koje je specificirano algoritmom automatskog upravljanja - npr. ne sme biti duže od periode odmeravanja ulaznog signala). Međutim, nisu svi zadaci vremenski kritični, u smislu da se kod njih može tolerisati i duže kašnjenje (npr. prikaz na displej). S tim u vezi javlja se problem *planiranja izvršenja zadataka*, na način koji će obezbediti da svi zadaci zadovolje zahteve u pogledu brzine odziva. Šta više, RTS treba da **garantuje** da će krajnji rokovi vremenski-kritičnih zadataka biti ispoštovani u svim okolnostima. Uobičajeno, ovaj problem se rešava uvođenjem *prioriteta* zadataka. Naime, svakom zadatku, u skladu sa stepenom njegove kritičnosti, dodeljuje se prioritet, a uvek kada postoji više od jednog iniciranog zadataka, za izvršenje se bira zadatak najvišeg prioritetom.

Takođe, zadaci ne moraju biti nezavisni, već može postojati potreba da u toku rada sistema različiti zadaci na neki način interaguju, komuniciraju i sinhronišu. Npr. vrednost upravljačkog signala izračunata od strane TASK_1, zajedno sa izmerenom temperaturom, se arhivira od strane TASK_2. To nameće potrebu da TASK_1, na neki način dostavi podatke TASK_2 i da ga obavesti o tome. U bliskoj vezi sa interakcijom zadataka je problem *deljivih resursa*. Za komunikaciju između zadataka uobičajeno se koriste tehnike zasnovane na *prenosu poruka* ili na *deljivim promenljivim*. Za sinhronizaciju zadataka koriste se mehanizmi kao što su *signali, semafori, kritične sekcije*.

1.2.4 Podela prema fleksibilnosti

- Fiksni-ugrađeni (*embedded*)
- Opšte namene

Embedded (mikro)računarski sistem je elektronski sistem koji sadrži mikroracunar programiran za specifičnu namenu i interakciju sa fizičkim svetom. U ovom kontekstu reč

„embedded“ znači skriveno u nečemu tako da spolja nije neposredno vidljivo. Dakle, kada se kaže „embedded mikroračunarski sistem“ ima se na umu mikroračunarski sistem koji je ugrađen u neki veći uređaj ili sistem i kao takav nije neposredno vidljiv korisniku sistema. Tako, personalni računar (u obliku desktop ili laptop računara) sam po sebi nije embedded sistem (već računar opšte namene), ali može biti korišćen za konstrukciju nekih embedded sistema. S druge strane, mikrotalasna pećnica, digitalni časovnik, savremeni automobil i robot jesu embedded sistemi. Kada koristimo mikrotalasnu pećnicu nemamo doživljaj da radimo sa računarom, uprkos tome što svim funkcijama mikrotalasne pećnice (kao što su kontrola zagrevanja, tajming, osvetljenje) upravlja računar (u obliku mikrokontrolera) koji je ugrađen u ovaj uređaj. Softver koji upravlja embedded sistemom programiran je u ROM-u i nije neposredno dostupan korisniku sistema. Korisnik ne može da „instalira“ novi program niti da piše svoj program za ovakav sistem.

Za razliku od embedded sistema, računarski sistem opšte namene tipično poseduje tastaturu, disk i grafički displej i može biti programiran za različite namene (npr. obrada teksta, elektronska pošta, računovodstvene primene, sistemi baza podataka itd). Korisnik računara opšte namene ima neposredan pristup softveru, u smislu da može da bira koji će operativni sistem da instalira i koji program će da pokrene. S obzirom da računar opšte namene poseduje priključak za prenosivi disk i mrežni interfejs, novi programi mogu lako da se dodaju. Najrasprostranjeniji tip računara opšte namene je personalni računar. Kategorija moćnijih računara obuhvata radne stanice i superračunare. Radne stanice i superračunari po pravilu sadrže više procesora i mnogo veću količinu memorije u odnosu na tipičan personalni računar. Ovi moćni računar se koriste za obradu velike količine podataka (npr. bankarski sistemi) i intenzivna izračunavanja (za potrebe naučnih istraživanja). Iako oblast embedded sistema ne obuhvata računare opšte namene, mnogi osnovni principi embedded sistema se mogu primeniti na sve računarske sisteme.

1.2.5 Podela prema interakciji

- Samostalni (Stand-alone)
- Zavisni (Interaktivni sa drugim)

Samostalni sistemi su nezavisni elektronski uređaji/sistemi. Oni prihvataju digitalne i analogne ulazne signale na svojim ulaznim portovima, obavljaju kondicioniranje signala (pojačanje, filtriranje, linearizaciju, normalizaciju,...), vrše AD konverziju, procesiraju digitalne podatke, i predaju rezultate pridruženim izlaznim uređajima. Izlazni uređaji mogu

biti tipa displeji, upravljački blokovi, ili neki drajverski (pobudni) stepeni. Uređaji za zabavu kakvi su konzole za video igre, MP3 plejeri, digitalne kamere, mikrotalasne peći, veš mašine, i dr., su tipični predstavnici sistema koji pripadaju ovoj kategoriji.

1.2.6 Podela prema broju procesora

- Jednoprocesorski
- Višeprocessorski

Višeprocessorski sistemi predstavljaju mogućnost značajnog poboljšanja za veliki broj aplikacija koje imaju problem sa procesorom. Kod njih se veliki poslovi dekomponuju-razlažu na nekoliko manjih poslova koji se izvršavaju istovremeno i tako se vreme izvršavanja aplikacija smanjuje.

1.2.7 Podela prema broju procesa

- Jednoprocesni
- Višeprocessni (višezadačni)

1.2.8 Podela prema razmeštaju delova sistema

- Distribuirani (sa i bez hijerarhije)
- Nedistribuirani

Kod distribuiranih sistema (prostorno raspoređenih) računarski resursi se (fizički-memorija, procesori, diskovi i logički-procesi, datoteke) nalaze na više od jednog mesta. Distribuiranim upravljanjem povezuju se resursi i koordinira aktivnost u distribuiranom sistemu. Strategije distribuiranog upravljanja mogu biti: centralizovane, hijerarhijske ili potpuno autonomne. Cilj distribuiranja računarskih resursa je povećanje performansi sistema i/ili povećanje pouzdanosti sistema.

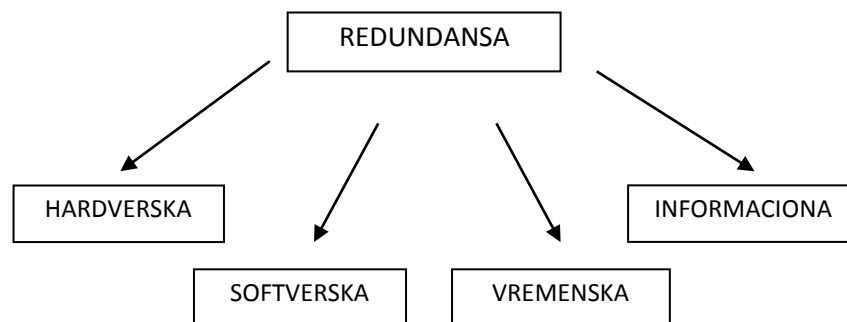
Distribuirani sistemi mogu biti:

- Snažno spregnuti - multiprocessorski sistemi
- Umereno spregnuti – lokalna računarska mreža, klijent-server arhitektura
- Slabo spregnuti – potpuno autonomni sistemi u računarskoj mreži (LAN/WAN)

1.2.9 Podela prema redundantnosti

- Neredundantni
- Redundantni

Redundansa se može definisati kao dodatni resurs preko onog koji je potreban za normalno funkcionisanje sistema. Osnovni resursi RTS-a su računarski hardver, softver, vreme i informacija (podaci), pa se prema tome može i klasifikovati redundansa kao što je i prikazano na Sl. 1-4.



Sl. 1-4 Osnovni tipovi redundanse

- Hardverska redundansa - podrazumeva postojanje dodatnog hardvera sa svrhom detektovanja i/ili tolerancije otkaza. Fizičko umnožavanje hardvera je najčešći oblik redundanse koji se danas koristi čemu značajno doprinosi i činjenica da poluprovodničke komponente postaju sve većeg stepena integracije i sve jeftinije. Prema načinu delovanja hardverske redundanse se dele na pasivnu, aktivnu i hibridnu.
- Softverska redundansa - predstavlja dodavanje softvera, iznad onog neophodnog za obavljanje zadatih funkcija, a u cilju detekcije i tolerisanja grešaka. Osnovna tehnika čiste softverske redundanse za detekciju i maskiranje otkaza u softveru je N verzija programa. Kod ove tehnike postoje najmanje tri verzije softvera (obično razvijene od tri nezavisna tima za razvoj softvera) koje rade paralelno. Ovo znači da je za svaku verziju softvera potreban poseban procesor. Nakon izvršene obrade svaka verzija softvera daje svoj rezultat, a jedan od procesora vrši poređenje i većinskim glasanjem daje konačan rezultat.

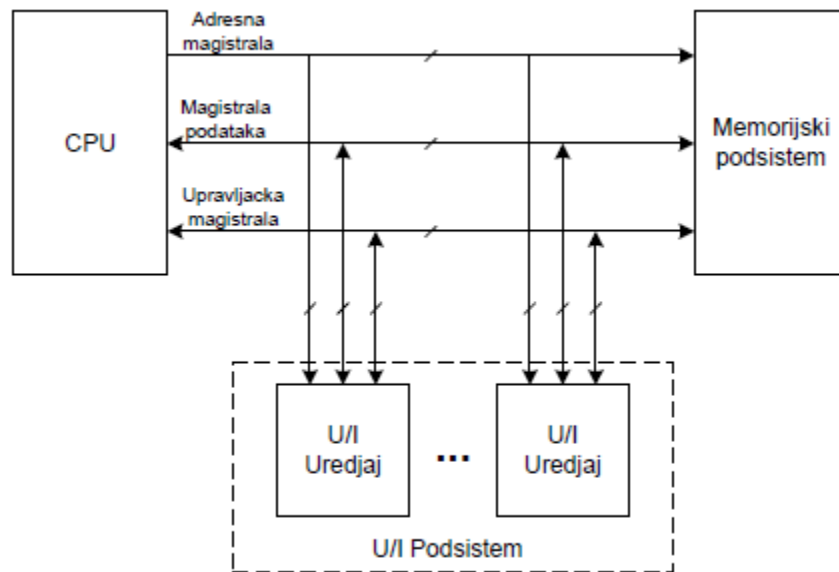
- Informaciona redundansa - predstavlja dodavanje informacija iznad zahtevanih za izvršavanje zadate funkcije. Dobri primeri informacione redundanse su kodovi za detekciju, kao i kodovi za detekciju i korekciju grešaka, koji se formiraju dodavanjem redundantnih bitova rečima, ili prevodenjem reči u neki novi oblik koji sadrži redundante informacije.
- Vremenska redundansa - podrazumeva korišćenje redundantnog vremena u radu sistema. Osnovni princip vremenske redundanse sastoji se u ponavljanju izračunavanja na način koji omogućava detekciju otkaza. Naime, ista izračunavanja izvršavaju se dva ili više puta i upoređuju rezultati u cilju određivanja eventualnog neslaganja. Ako se detektuje greška, možemo ponoviti izračunavanje da bi videli da li neslaganje ostaje, ili nestaje. Da bi se postigla pouzdanost u radu sistema neophodno je uspešno izvršiti proceduru tolerisanja otkaza, odnosno neophodno je prvo detektovati grešku koja predstavlja uzrok otkaza. U osnovi, detektori otkaza zasnovani su na korišćenju redundanse u hardveru, informaciji, softveru i/ili vremenu.

1.3 Računarski sistem

Većina savremenih računarskih sistema, počev od jednostavnih mikrokontrolera do personalnih računara i radnih stanica velike moći izračunavanja, zasnovana je na istoj bazičnoj organizaciji koju čine sledeća tri glavna podsistema (Sl. 1-5):

- Centralna procesorska jedinica (CPU – Central Processor Unit). Interno, CPU se sastoji od upravljačke jedinice, koja upravlja radom celokupnog računara i staze podataka koja sadrži registre, aritmetičko-logičku jedinicu (ALU) i druge funkcijske jedinice specijalizovane za pojedine tipove obrade podataka (matematički koprocesori, akceleratori i sl.).
- Memorijski podsistem. Sačinjen od specijalizovanih kola sa mogućnošću memorisanja informacije. U memoriji računara čuvaju se program i podaci. CPU vidi memoriju kao skup memorijskih lokacija, gde svaka lokacija, u zavisnosti od organizacije memorije, sadrži 8/16/32/64 bita. Svakoj memorijskoj lokaciji pridružena je jedinstvena adresa.
- Ulazno-izlazni podsistem. Skup specijalizovanih kola koja služe za spregu CPU sa akterima iz spoljnog sveta, kao što su tastatura, miš i displej. Ovo su integrisana kola koja, tipično, poseduju svoju upravljačku jedinicu i stazu podataka. CPU vidi U/I uređaj kao skup od nekoliko memorijskih lokacija (tzv. registara ili portova). Upisom/čitanjem u/iz ovih lokacija moguće je upravljati radom U/I uređaja, tj. inicira neke aktivnosti perifernog uređaja, postavi

uređaj u željeni režim rada, pribavi informaciju o tekućem stanju uređaja, prenese/prihvati podatke u/od uređaja.



Sl. 1-5 Organizacija računarskog sistema

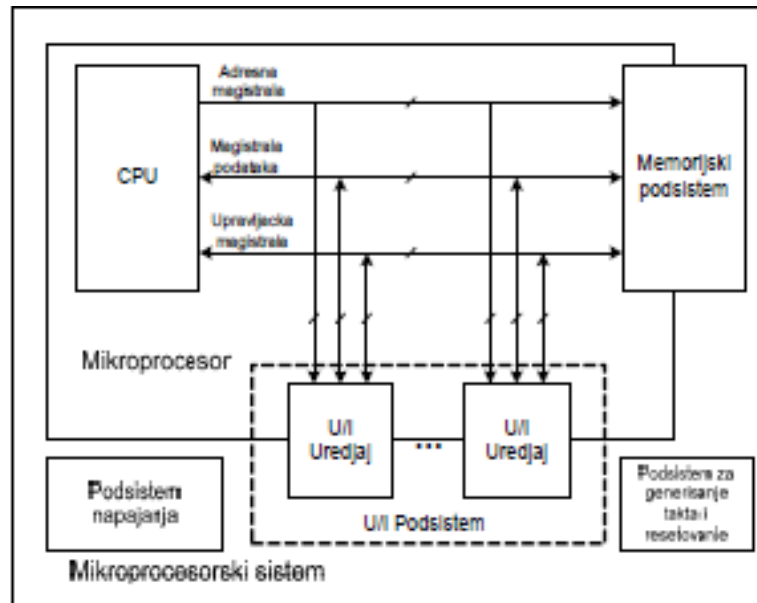
CPU je u stanju da obavlja elementarne računске operacije i razmenjuje podatke sa memorijom i specijalizovanim interfejsnim kolima za komunikaciju sa spoljnim svetom. Program se sastoji od niza instrukcija, od kojih svaka ukazuje na jednu elementarnu operaciju, a sve zajedno definišu izračunavanja koja računar treba da obavi kako bi se ostvario željeni algoritam. Program sastavlja čovek (programer) i zajedno sa ulaznim podacima smešta ih u memoriju računara. CPU pribavlja iz memorije instrukcije i podatke, i izvršava instrukcije, transformišući tako ulazne podatke u izlazne, koje, po okončanju rada, predaje okruženju. Koncept računara je nezavisan od tehnologije. Prvi računari bili su elektromehaničke naprave. Međutim, danas, računarski sistemi se realizuju isključivo u digitalnoj integrisanoj tehnologiji. Ova tehnologija omogućava realizaciju računskih mašina koje se sastoje od više miliona elementarnih sklopova (gejtova) i u stanju su da obavljaju više stotina miliona elementarnih računskih operacija u sekundi.

1.3.1 Pojam mikrorprocesora

Mikroprocesor je integrisano kolo (IC) koje u sebi sadrži CPU zajedno sa dodatnim hardverom koji omogućava direktnu spregu CPU sa drugim resursima računarskog sistema (memorija, U/I uređaji) (Sl. 1-6). Osim CPU jedinice, mikroprocesor može sadržati i izvesnu

količinu memorije i neke U/I uređaje kao što su tajmeri i brojači događaja, komunikacioni kontroleri i sl. koji se koriste kod velikog broja primena.

Mikroprocesorski sistem (MPS) je računarski sistem realizovan na bazi mikroprocesora. Pored mikroprocesora MPS sadrži podsistem napajanja i podsistem za generisanje takta i resetovanje. Takođe, u zavisnosti od zahteva konkretne primene, MPS može sadržati, u vidu zasebnih komponenti, dodatnu memoriju i specifične U/I uređaje.



Sl. 1-6 Organizacija mikroprocesorskog sistema.