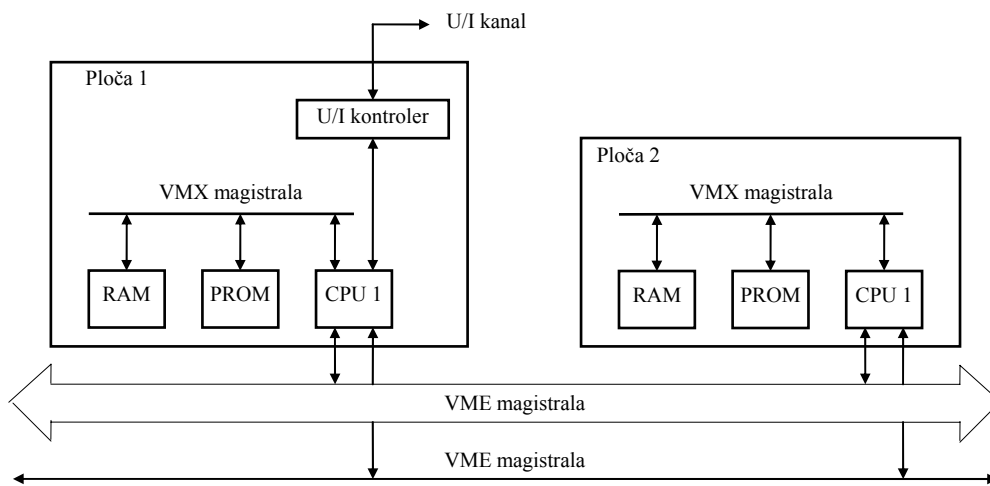


## 8. MAGISTRALE

Magistrala se koristi za povezivanje dva ili većeg broja sistemskih elemenata. Ona predstavlja skup linija (veza). Skup veza se deli od strane sistemskih elemenata, ali se takođe i koristi od strane sistemskih elemenata za međusobnu komunikaciju. Pojam magistrala obično ukazuje da su veze paralelne, pri čemu se duž istog puta prenosi po nekoliko signala. Globalno posmatrano, *sistem magistrala* predstavlja skup većeg broja magistrala koje se koriste za povezivanje različitih sistemskih elemenata u okviru računarskog sistema. Kao što je prikazano na slici 8.1, sistem magistrala čine magistrale koje su hijerarhijski organizovane na sledećim, različitim nivoima:



Sl. 8.1.

- **Nivo ploča** - na najnižem nivou je komponentno-orijentisana magistrala definisana od strane CPU-a i perifernih čipova. Obično se ova magistrala zove *lokalna magistrala*.
- **Zadnja ploča** - obezbeđuje komunikaciju između elemenata sistema, na nivou ploča. Zadnja ploča je obično realizovana kao štampana ploča sa većim brojem identičnih konektora koji su paralelno povezani. Svaka ploča se postavlja u svoj konektor i na taj način povezuje sa ostalim pločama. Zadnja ploča se često zove i *majka ploča* (*motherboard*). Definicija signala na zadnjoj ploči obično se ne vezuje za određeni tip CPU-a kao što je to slučaj sa lokalnom magistralom. Na tržištu danas postoji veći broj standardizovanih magistrala zadnje ploče, od kojih su poznatije VME, Multibus I i II, Nubus, Fastbus, Futurebus i dr.
- **Nivo interfejsa** - ovim tipovima magistrala ostvaruje se komunikacioni put između U/I uređaja (diskovi, štampači, i dr.) i ostatka sistema. Ove magistrale obično povezuju nezavisne sisteme i predviđene su za rad na većim rastojanjima u odnosu na nivo zadnje ploče. Tipični predstavnici su SCCI (*Small Computer System Interconnect*) magistrala, GPIB (*General Purpose Interface Bus*) poznata kao IEEE 488, i dr.

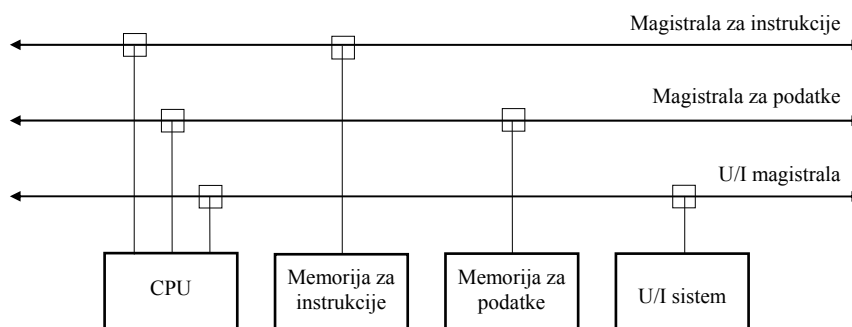
Sa ciljem da se poboljšaju performanse sistema u sistem se ugrađuju dodatne magistrale. Kada se zadnja ploča koristi za pribavljanje instrukcija iz memorije, a takođe i za prenos podataka ka/iz U/I ploča, performanse sistema su često ograničene od strane propusnosti magistrale, tj. maksimalnom brzinom sa kojom se mogu prenositi podaci po magistrali. Jedan od načina da se poboljšaju performanse je da se koriste "privatne", ili lokalne, magistrale. Tipične lokalne memorijske magistrale su iLBX (koristi se kod Multibus), VMX (koristi se kod VME), i dr. Alternativno, posebne U/I magistrale, kao što su Intelova Multichannel, se mogu koristiti da rasterete zadnju ploču od U/I aktivnosti. Poseban tip specijalizovanih magistrala se obično koristi za interprocesorski prenos poruka. Ovakve magistrale se obično realizuju kao serijske i omogućavaju prenos poruka između procesorskih modula kod multiprocesorskih sistema. Tipični predstavnici su iSSB (Multibus II) i VMS (VME).

Najveći broj magistrala se može podeliti na sledeće tri sekcije (magistrale):

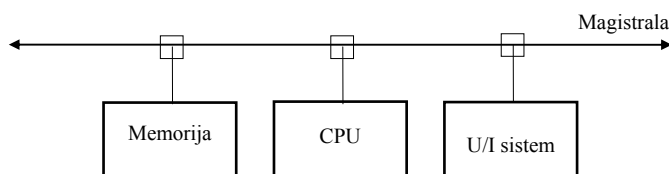
- adresna magistrala - specificira jednu memorijsku lokaciju ili U/I port koji komunicira sa CPU-om,
- magistrala za podatke - koristi se za prenos podataka,
- upravljačka magistrala - upravljački signali pomoću kojih se sinhronizuje prenos podataka. Kod ove magistrale postoje posebne linije pomoću kojih se reguliše prioritet prekida i dodela deljivih resursa na korišćenje, kao što je magistrala.

### 8.1. Podela magistrala u zavisnosti od namene

Magistrale se mogu podeliti na *namenske* i *nenamenske*. Namenske magistrale se (slika 8.2), kao što i samo ime ukazuje, koriste za obavljanje specifičnih funkcija. Osnovna njihova prednost je izrazito visoka propusnost, a nedostatak je njihov veliki broj. Kada je u sistemu instaliran veći broj namenskih magistrala, prenos se može istovremeno ostvariti po svim magistralama. Zbog zahteva za ugradnju velikog broja konektora namenske magistrale se ne koriste kod realizacije manjih računarskih sistema. Nenamenske magistrale, zovu se takođe i deljive magistrale, se koriste da obave veći broj funkcija. Na slici 8.3 CPU je povezan sa većim brojem jedinica preko jedinstvene deljive magistrale. Kod deljivih magistrala kapacitet brzine prenosa je manji, ali postoji i potreba za ugradnjom arbitražnog mehanizma pomoću koga se ostvaruje zaštita od konflikata u slučaju kada dva, ili veći broj potencijalnih korisnika istovremeno zahteva dodelu magistrale.



Sl. 8.2.



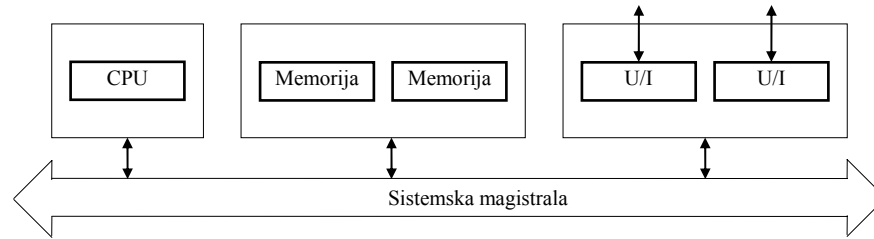
Sl. 8.3.

#### 8.1.1. Klasifikacija deljivih magistrala

Deljive magistrale možemo svrstati u zavisnosti od toga kako je izvršena podela funkcija sistema po pločama. Kriterijum podele može biti tip resursa ili funkcija koja se obavlja.

#### Podela magistrala u zavisnosti od tipa resursa

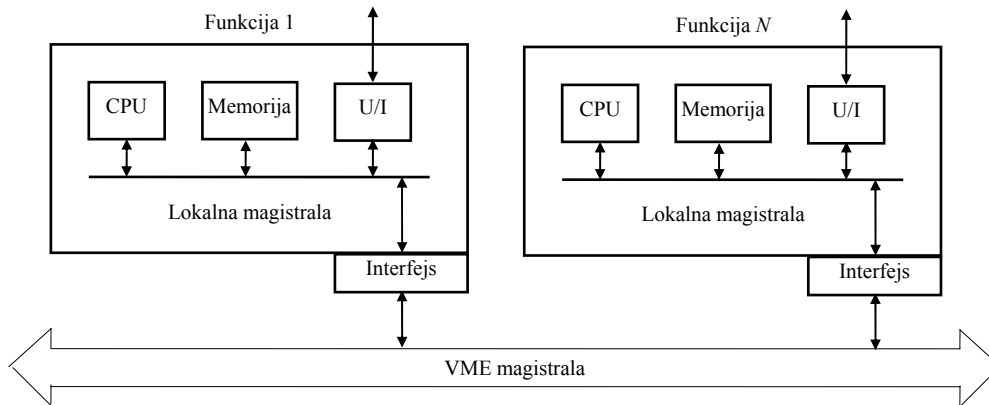
Resursi istog tipa, recimo memorije, pakuju se u jedinstvenu celinu i povezuju na magistralu. Kompletan mali računarski sistem se formira povezivanjem svih tipova resursa, CPU-memorija-U/I magistrala, kao što je prikazano na slici 8.4.



Sl. 8.4.

### Podela magistrala u zavisnosti od funkcije

U ovom slučaju vrši se povezivanje ploča koje obavljaju polunezavisne funkcije (slika 8.5). Za svaku ploču se pretpostavlja da ima dovoljnu lokalnu procesnu moć, memoriju i U/I, tj. da predstavlja jedinstveni računar na ploči.



Sl. 8.5.

#### 8.1.2. Terminologija magistrale

Ploče koje se povezuju na magistralu mogu biti tipa:

- a) **gospodar** - može inicirati ciklus na magistrali
- b) **sluga** - odaziva se gospodaru

Neki od uređaja mogu biti i gospodar i sluga, ali ne istovremno. Svi uređaji koji rade kao gospodari magistrale zovu se potencijalni gospodari magistrale. S obzirom da je u jednom trenutku magistrala dodeljena samo jednom gospodaru, koga zovemo tekući gospodar magistrale, potrebno je ugraditi arbitražni mehanizam kojim se odlučuje koji će gospodar magistrale u narednom trenutku dobiti pravo upravljanja nad magistralom. Kada prenos podataka počne, ploča koja predaje podatak se zove izvor, a ploča koja prima podatak odredište.

Kompletna sekvenca, od trenutka kada se magistrala zahteva do trenutka završetka prenosa podataka zove se *transakcija na magistrali*. Da bi se obavio prenos po magistrali, neophodno je da se obave sledeće operacije:

- zahtev
- arbitraža
- adresiranje
- prenos podatka
- detekcija i signalizacija greške.

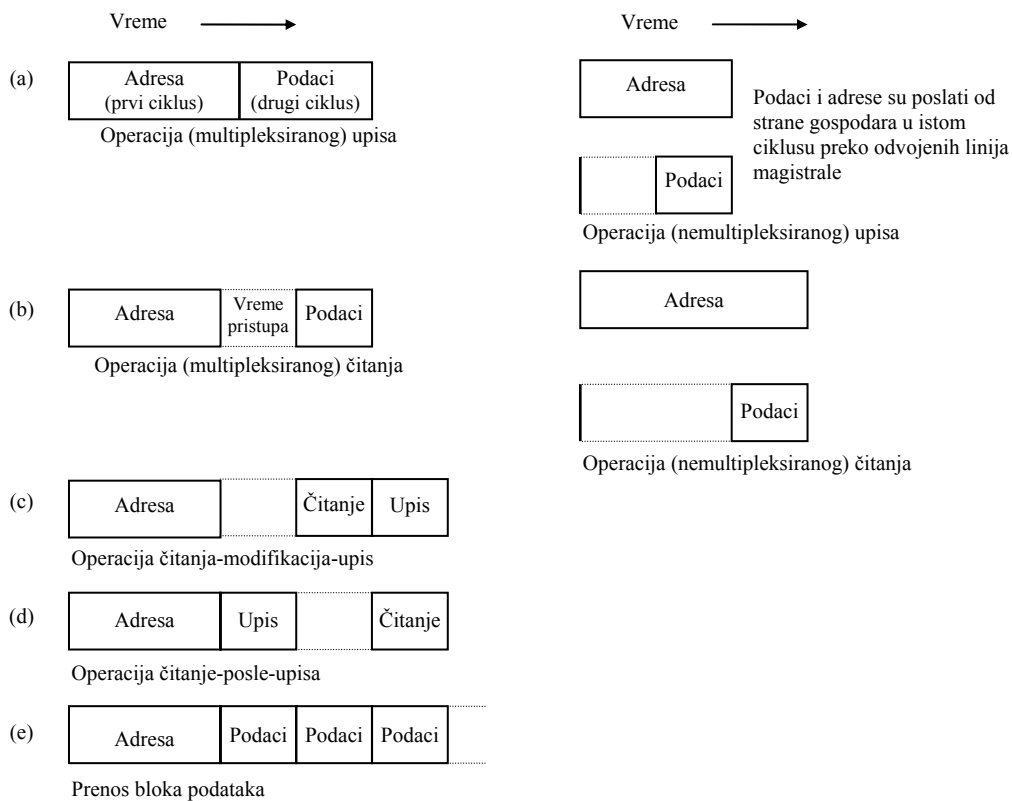
### 8.2. Prenos podataka

Kao što je prikazano na slici 8.6, moguće je ostvariti sledeće tipove prenosa podataka:

- Operacija upis (slika 8.6a) - gospodar magistrale predaje adresu u prvom ciklusu, a zatim podatak u narednom ciklusu. Sluga mora da prepozna adresu pre nego što prihvati podatak. Na levoj strani slike 8.6a prikazan je

vremenski multipleksirani rad, a na desnoj strani nemultipleksirani rad operacije upis. Kod multipleksiranog rada iste linije se u prvom ciklusu koriste za prenos adresa, a u drugom za prenos podataka. Kod nemultipleksiranog rada za prenos podataka i adresa se koriste posebne linije.

- Operacija čitanja (slika 8.6b) - gospodar predaje adresu i čeka da podatak bude dostupan od strane sluge. Vreme čekanja se često zove vreme pristupa.
- Operacija čitanje-modifikacija-upis (slika 8.6c) - čine je tri ciklusa: adresni, čitanje podataka i upis podataka. Celokupna operacija je nedeljiva, sa ciljem da se u toku modifikacije elemenata podataka zabrani pristup tom podatku od strane drugog gospodara magistrale.
- Operacija čitanje-posle-upisa (slika 8.6d) - čini je nedeljiva sekvenca upis-čitanje. Operacija čitanja se obavlja radi provere.
- Operacija prenos bloka podataka (slika 8.6e) - nakon jednog adresnog ciklusa sledi "n" ciklusa podataka. Prvi podatak se prenosi na specificiranu adresu, a naredni na sukcesivne adrese.
- Razbijeni prenos podataka - u toku operacije čitanja vreme pristupa informaciji može biti dugo. Umesto da se magistrala zauzme celo vreme, ona se oslobađa kako bi se obavile druge transakcije. Kasnije se prenos završava. Ova tehnika se često zove paketna komutacija (*message* ili *packet switching*).



Sl. 8.6.

### 8.2.1. Adresiranje

Nakon što je gospodar stekao pravo upravljanja nad magistralom on treba da uspostavi kontakt sa jednim ili većim brojem sluga sa ciljem da se ostvari prenos podataka. Ovo se zove *adresiranje*, a sam postupak čine dva koraka: adresiranje ploče i adresiranje elemenata na toj ploči.

Obično se isti mehanizam adresiranja koristi u oba slučaja. Svakoj ploči se dodeljuje blok adresa, tj. MS adresnim bitovima specificira se ploča, a LS adresnim bitovima iste sdrese specificira se elementa podataka na ploči.

U daljem tekstu ukažimo na način specificacije adrese sluge.

#### Specificacija adrese sluge

Adresa sluge specificira se adresnim linijama na magistrali. Adresa sluge obično odgovara adresi ploče, i ova je jedinstvena. U globalu razlikujemo dve metode za dodelu adrese pločama:

- **Logičko adresiranje** - zove se takođe i lokaciono-nezavisno. Svakoj ploči se dodeljuje jedinstvena adresa ili grupa adresa. Dodela adresa se ostvaruje ručnim postavljanjem prekidača ili kratkospojnika na samoj ploči. Adrese se postavljaju proizvoljno i ne zavise od tipa ploče i njene pozicije na zadnjoj ploči. Da bi se ostvarilo logičko adresiranje na nivou svake ploče, treba ugraditi dodatni hardver. Standardne magistrale kao što su VME i Unibus koriste ovaj način adresiranja.
- **Geografsko adresiranje** - zove se takođe i lokaciono-zavisno. Svaka ploča se adresira na osnovu njene fizičke lokacije, tj. mesta gde je ploča postavljena na zadnjoj ploči. Ova lokacija se zove *broj slot*a i do nje se direktno dovode signali za selekciju te ploče.

### Broj sluga koji učestvuju u transakciji na magistrali

Često se u toku rada javlja potreba da jedan gospodar komunicira sa većim brojem sluga. U tom cilju koriste se *broadcall* i *broadcast* operacije.

**Broadcall operacija** (selektivna emisija) uslovljava da sve selektovane ploče tipa sluga postave svoje podatke na magistralu. Nakon toga vrši se zbirna AND ili OR operacija podataka selektovanih sluga. Ova operacija ima ograničeno delovanje, a tipično se koristi za identifikaciju izvora prekida. Na primer, svakom potencijalnom izvoru prekida dodeljuje se jedna bit pozicija na magistrali. Ovom bit pozicijom na jedinstveni način se identifikuje sluga. Ako je magistrala podataka 32-bitna, moguće je identifikovati do 32 izvora prekida.

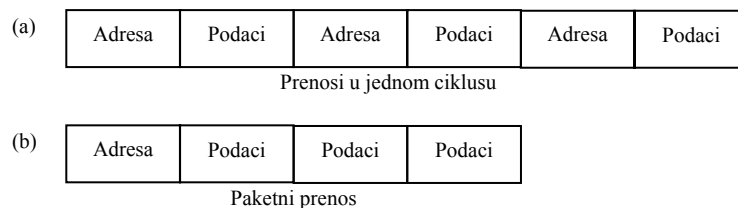
**Broadcast operacija** (emisija svima) se uglavnom koristi za održavanje konzistentnosti podataka. Njeno korišćenje ima puni smisla kod multiprocesorskih sistema kada je na nivou svakog procesora instalirana keš memorija, tj. postoji distribucija keša a shodno tome i potreba za simultano ažuriranje. Reset operacija, slučaj kada se svaka ploča postavlja u svoje inicijalno stanje, je takođe oblik broadcast operacije.

*Broadcast* i *Broadcall* operacije obično se iniciraju preko specijalnih adresa (adresa 0 ili FF), ili aktiviranjem specijalne upravljačke linije pri čemu se adresne linije koriste za selekciju podskupa, ili svih sluga.

#### 8.2.2. Prenos bloka podataka

Elementi bloka podataka se obično smeštaju u uzastopnim memorijskim lokacijama i, kao što je prikazano na slici 8.7, oni se mogu preneti na dva načina. U prvom slučaju (slika 8.7a), kažemo da se radi o prenosu na nivou jednog ciklusa, tj. posle svake generisane adrese vrši se prenos po jednog podatka.

U drugom slučaju (slika 8.7b), podaci se prenose u paketima (*burst*). U ovom slučaju magistrala je osposobljena za ovakav tip prenosa. Na početku prenosa predaje se inicijalna adresa nakon čega sledi blok podataka. Dužina bloka može biti fiksna, 1, 2, 4, 8, ili 16 reči ili promenljiva.



Sl. 8.7.

Ozbiljan nedostatak paketnog prenosa ogleda se u sledećem. Kada prenos počne, a poruka je dugačka, gospodar neće osloboditi magistralu sve dok se prenos ne završi. Sa stanovišta obrade može se desiti da se propusti analiza nekog bitnog ili prioritetnijeg događaja. Da bi se ovaj problem uspešno rešio, mora da postoji mehanizam *istiskivanja*. To znači da se tekućem gospodar magistrale signalizira kada neki potencijalni gospodar višeg nivoa prioriteta zahteva dodelu magistrale. Nakon prihvatanja zahteva tekući gospodar magistrale prekida prenos, oslobađa magistralu, a kasnije završava sa započetim prenosom.

#### 8.2.3. Sinhronizacija kod prenosa

Kada dve jedinice sistema, kao što su to procesor i memorija, razmenjuju podatke, sistemsko rešenje treba da osigura da će u toku prenosa biti ostvarena sinhronizacija u radu obe jedinice. Jedna tipična sekvenca događaja, kada izvor upisuje informaciju u odredište, ima sledeći oblik:

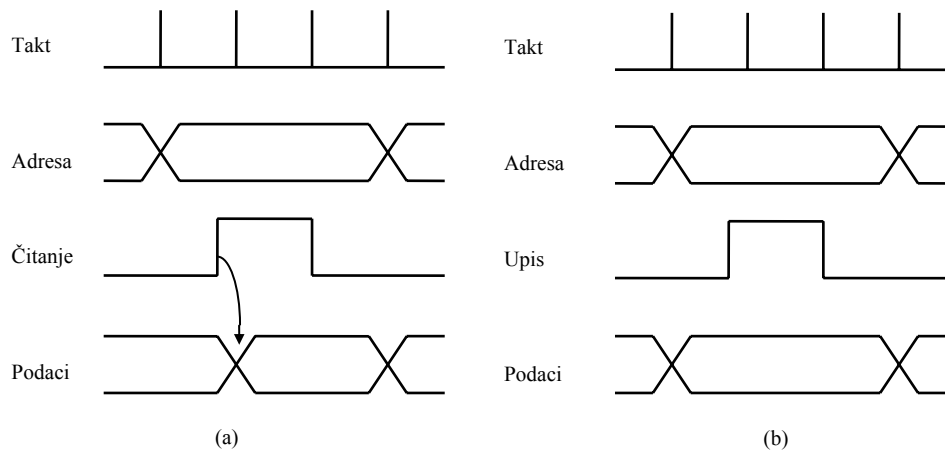
Izvor	Odredište
(1) Posavlja važeću informaciju na magistralu.	(2) Informacija koju treba primiti je stabilna.
(3) Signalizira da je informacija stabilna.	(4) Saznaje da je informacija stabilna, prihvata informaciju.
(6) Saznaje da je informacija prihvaćena.	(5) Generiše signal potvrde kojim se ukazuje da je informacija prihvaćena.
(7) Ukida informaciju.	
(8) Izdaje signal potvrde kojim se ukazuje da je informacija ukinuta.	(9) Saznaje da je informacija ukinuta.
	(10) Završava se ciklus potvrde prijema informacije
(11) Počinje naredni prenos.	

Analizirana sekvenca se može realizovati koristeći se jednim od sledeća tri rešenja vremenskog vođenja događaja:

- **sinhrono** - svi događaji se dešavaju u fiksnim vremenskim intervalima.
  - **asinhrono** - signali u sekvenci se generišu u proizvoljnim vremenskim intervalima.
  - **polusinhrono** - signali u sekvenci se generišu u fiksnim trenucima u odnosu na taktni signal.
- Saglasno ovome magistrale delimo na sinhronne, asinhronne i polusinhronne.

### Sinhronne magistrale

Kod sinhronih magistrala prenos podataka je kontrolisan globalnim taktom koji se generiše od strane zajedničkog oscilatora. Opšta sekvenca događaja kod sinhronih magistrala prikazana je na slici 8.8.



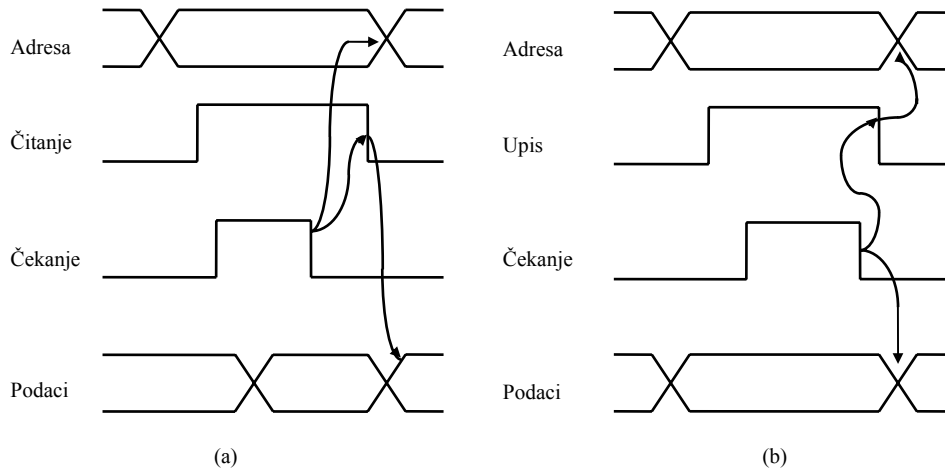
Sl. 8.8.

Na slici 8.8a opisana je sekvenca događaj kod operacije čitanje, a na slici 8.8b sekvenca događaja kod operacije upis.

I pored toga što se prenos podataka kod ovih magistrala obavlja veoma brzo, osnovni nedostatak je taj što sve komponente, kod ispravnog prenosa, mora da rade sa brzinom koja je diktirana od strane glavnog takta.

### Asinhronne magistrale

Kod ovih magistrala sinhronizacija prenosa nije diktirana glavnim taktom. Metod asinhronog prenosa podataka se više primenjuje kod magistrala mikroracunarskih sistema. Opšta sekvenca prenosa prikazana je na slici 8.9.



Sl. 8.9.

Slika 8.9a se odnosi na operaciju čitanje, a slika 8.9.b na operaciju upis.

Obično asinhronne magistrale zahtevaju "signal priznavanja" (potvrde) koga, da bi se rad magistralnog ciklusa okončao generiše adresirani uređaj. Brzi uređaji potvrđuju priznavanje odmah nakon dekodiranja adrese, dok spori uređaji unose čekanje sve dok ne budu spremni da završe ciklus. U oba slučaja efekat je isti i svodi se na generisanje WAIT ili READY signala. Suština prenosa sastoji se u tome što u toku svakog ciklusa svaki adresirani uređaj mora generisati signal priznavanja. Drugim rečima, to ne treba da bude isključivi zadatak samo sporih uređaja. Asinhronne magistrale omogućavaju da signal priznavanja bude asinhron u odnosu na sistemski takt, tako da se problem sinhronizovanog prenosa podataka rešava od strane mikroprocesora ili dodatno ugrađenog hardvera.

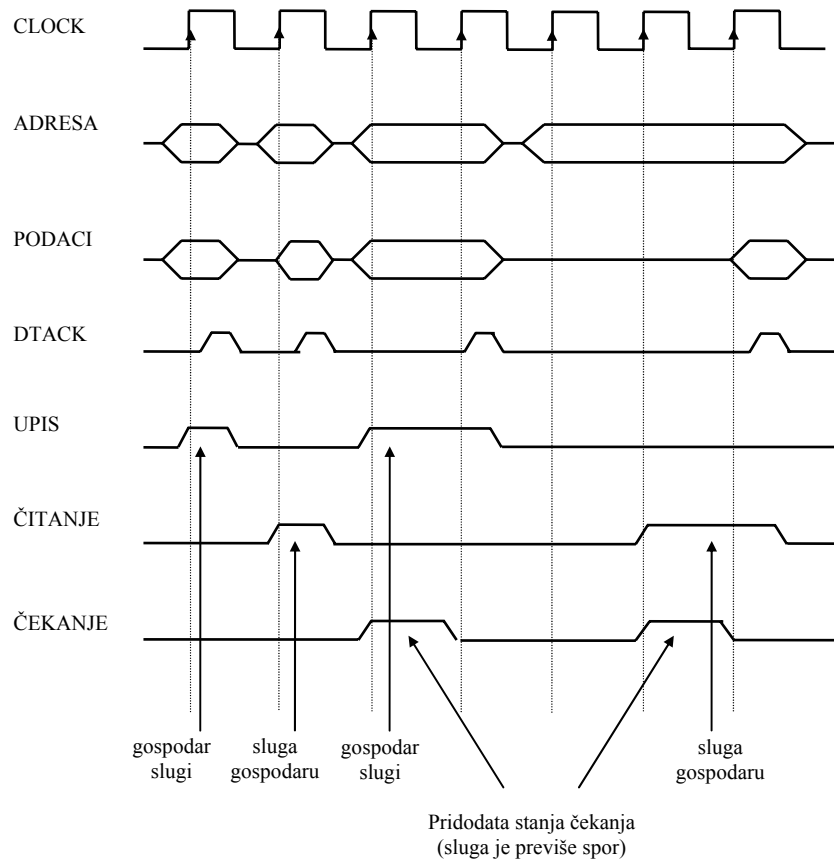
Jedna od prednosti ovog pristupa je povećana sistemska sigurnost. Naime, ako se pristupa nepostojećoj memorijskoj lokaciji ili U/I portu, sistem se automatski zaustavlja, pošto će on zauvek čekati na priznavanje. Da bi se rešio problem beskonačnog zaustavljanja, u sistem se ugrađuje posebna jedinica nazvana pas-čuvar (watchdog) čiji je prvenstveni zadatak da uslovi prelaz na izvršenje rutine za obradu greške kada dođe do zastoja.

Asinhrona magistrala se može implementirati kod mikroprocesora uz pomoć signala READY ili  $\overline{\text{WAIT}}$  i ugradnjom dodatne logike. READY ulaz mikroprocesora se uobičajeno nalazi u stanju "nisko", a postavlja se na "visoko" kada se od strane adresiranog uređaja primi signal priznavanja. Ovaj princip rada se ponekad zove *normalno-nespreman*. Kod sistema tipa *normalno-spreman*, periferno ili memorijsko interfejs kolo mora da invertuje READY kako bi se ubacila stanja čekanja. Signal READY se mora invertovati u okviru fiksnog vremenskog intervala, a ako se ne invertuje magistralni ciklus produžava bez stanja čekanja. Shodno prethodnom, *normalno-nespreman* asinhrona magistrala može tolerisati rad sa sporijim uređajima.

Mikroprocesor Motorola MC68000 direktno podržava rad asinhronne magistrale, kod koga  $\overline{\text{DTACK}}$  (*data transfer acknowledge*) ulaz zamenjuje liniju READY/ $\overline{\text{WAIT}}$  koja se koristi kod najvećeg broja mikroprocesora.

### Polusinhronne magistrale

Kod polusinhronih magistrala postoje upravljački signali čije se promene dešavaju u trenucima određenim od strane fiksne taktne pobude. Vremenski interval između upravljačkih signala može biti promenljiv ali samo u umnošcima taktne pobude. Signal fiksne taktne pobude može biti generisan od strane glavnog sistemskog oscilatora ili od strane tekućeg gospodara magistrale.



Sl. 8.10.

Veliki broj današnjih mikroprocesora koristi polusinhroni pristup za upravljanje prenosom podataka na magistrali. Mikroprocesor obično ne očekuje odziv od memorije ili perifernog uređaja. Ipak "wait" ulaz CPU-a treba da bude potvrđen ako memorija ili periferni sklop treba da produže magistralni ciklus. Kako se takt mikroprocesora ne može zaustaviti, produžava se ciklus magistrale. Mikroprocesori Z80, 8085, 8086 i drugi koriste ovakav princip kod sinhronizacije u prenosu podataka.

Prednost polusinhronog prenosa ogleda se u manjoj osetljivosti na smetnje, jer se samo signali smetnji koji su na upravljačkim linijama u toku usponske ili opadajuće ivice takta mogu pogrešno interpretirati. Nedostatak ovog prenosa je taj što se vreme odziva zaokružuje na umnožak taktnog perioda.

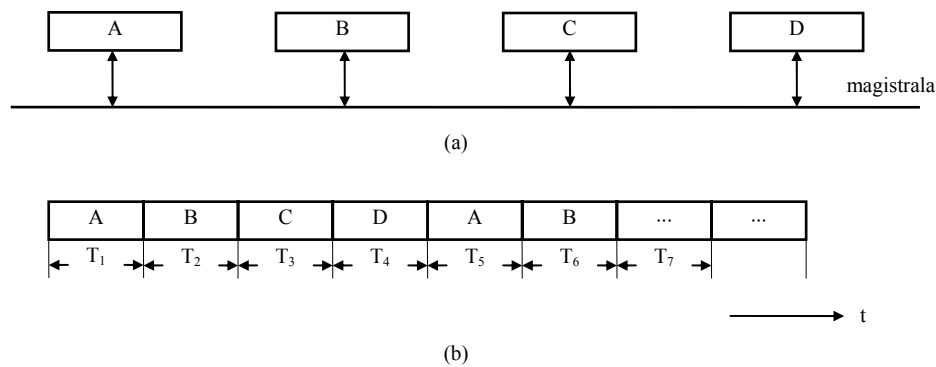
### 8.3. Arbitraža na magistrali

Obično se na magistralu priključuje veći broj potencijalnih gospodara, ali se ona u datom trenutku može dodeliti samo jednom. Arbitražni mehanizam magistrale garantuje da će se dodela ostvariti bez konflikta. Naime, u datom trenutku samo će jedan gospodar upravljati magistralom, a ostali se takmiče da bi dobili pravo upravljanja nad njom. Arbitraža se može izvesti kao statička ili dinamička.

#### 8.3.1. Statička arbitraža na magistrali

Kod ove arbitraže raspored transakcija na magistrali između potencijalnih gospodara vrši se po unapred određenom redosledu. Obično, kao što je prikazano na slici 8.11, redosled dodeljivanja je kružni.





Sl. 8.11.

Bilo kakav prenos, sinhroni, asinhroni ili polusinhroni, može se koristiti za prenos podataka, ali se u najvećem broju slučajeva radi o sinhronom prenosu, jer on svakom potencijalnom gospodarstvu garantuje određeni broj transakcija po jedinici vremena. Prednost ovakvog pristupa je jednostavnost izvođenja, tj. ugradnja jednostavnog hardvera uz veliku propusnost u toku prenosa podataka. Nedostatak ove dodele je što ona ne uzima u obzir realne potrebe za prenosom. Drugim rečima, magistrala se dodeljuje onom gospodarstvu koji u tom trenutku nema potrebe za njom, tj. on okupira magistralu NOP operacijama.

### 8.3.2. Dinamička dodela magistrale

Ovaj tip dodele obezbeđuje da se vlasnik magistrale određuje dinamičkim putem, po zahtevu. Dodela i oslobađanje magistrale ostvaruje se određenom politikom. U daljem tekstu analiziraćemo detaljnije ove politike.

#### Politike dodele magistrale

Kada potencijalni gospodar želi da obavi transakciju na magistrali, on izdaje zahtev za korišćenjem. Slični zahtevi se mogu javiti i od strane drugih potencijalnih gospodarstva. Kada tekući gospodar oslobodi upravljanje nad magistralom, on mora da odluči kom će potencijalnom gospodarstvu predati upravljanje. Da bi se to odredilo mora da postoji politika dodele koja može biti:

- **Zasnovana na prioritetu** - svakom potencijalnom gospodarstvu je dodeljen fiksni prioritet. Magistrala se dodeljuje gospodarstvu sa najvišim prioritetom.
- **Zasnovana na nepristrasnosti** - u situacijama kada potencijalni gospodarstva imaju isti prioritet, svakom gospodarstvu koji je izdao zahtev za dodelu se mora garantovati dodela magistrale pre nego što se bilo kom drugom gospodarstvu po drugi put ona ponovo dodeli.
- **Kombinovana** - politike zasnovane na prioritetu i nepristrasnosti se mogu kombinovati. Obično se zahtevima najvišeg prioriteta dodeljuje politika zasnovana na prioritetu, a zahtevima najnižeg prioriteta politika zasnovana na nepristrasnosti.

Multiprocesorski sistemi standardno koriste kombinovanu politiku.

#### Politike oslobađanja magistrale

Za oslobađanje magistrale koriste se sledeće politike:

- **Oslobađanje po zahtevu** - tekući gospodar magistrale ima pristup magistrali sve dok se ne generiše drugi zahtev, tj. on zadržava pravo upravljanja nad magistralom i pored toga što ne koristi magistralu.
- **Oslobađanje nakon obavljene transakcije** - nakon obavljene transakcije gospodar oslobađa magistralu.
- **Istiskivanje** - gospodar koji ima viši prioritet u odnosu na tekući, nakon izdavanja zahteva uslovljava da gospodar za nižim prioritetom preda upravljanje nad magistralom i pored toga što nije završio sa prenosom.

### 8.3.3. Hardverski mehanizmi za arbitražu na magistrali

Hardver za arbitražu može se realizovati kao centralizovani ili distribuirani.

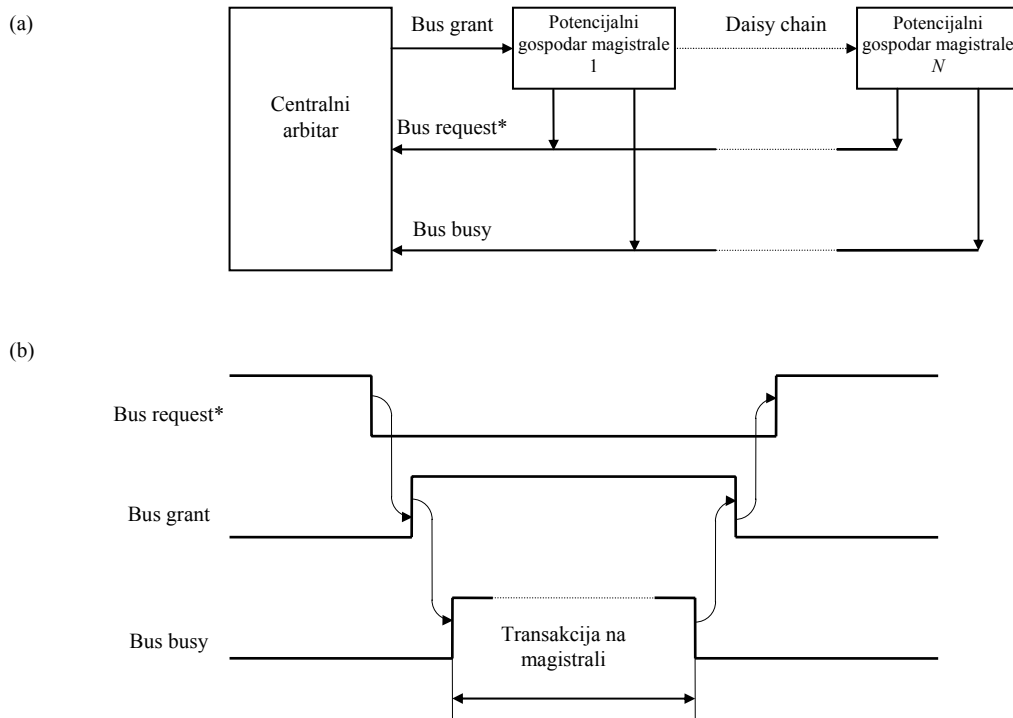
### Centralizovana arbitraža

Kod centralizovane arbitraže hardver je koncentrisan na jednom mestu, a može biti lociran u jednom od modula koji se povezuju na magistralu ili izveden kao poseban hardver koji se zove arbitar magistrale.

Gospodar koji zahteva dodelu magistrale predaje zahtev za dodelu centralnom arbitru. Arbitar odlučuje, ako postoji veći broj zahteva, kom će gospodaru dodeliti magistralu na osnovu politike dodele. Hardverski mehanizmi koji se koriste za dodelu i zahvatanje magistrale mogu se podeliti na sledeće grupe.

### Deljivi zahtev i lančano zahvatanje

Princip ovog metoda prikazan je na slici 8.12.

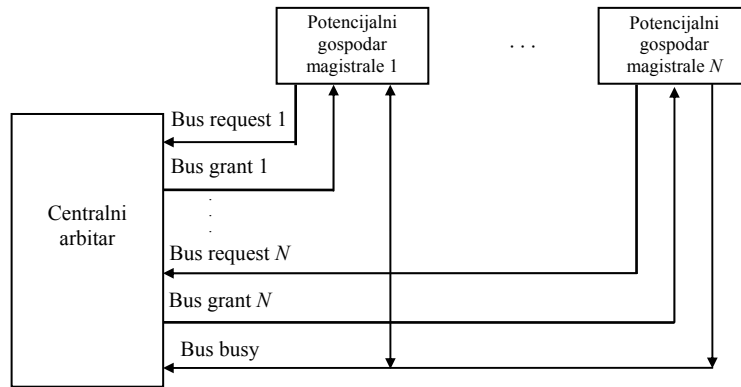


Sl. 8.12.

Svaki potencijalni gospodar magistrale izdaje zahtev za dodelu magistrale preko linije "BUS REQUEST\*". Zahtev je aktivan kao "nisko", a svi zahtevi su povezani žičanom ILI logikom. Kada centralni arbitar primi zahtev, on predaje signal "BUS GRANT" potencijalnom gospodaru magistrale označenom sa 1. Linija "BUS GRANT" povezuje sve potencijalne gospodare u lanac (*daisy chain*), tako da gospodar 1 prosleđuje signal gospodaru 2, itd. Politika dodele se zasniva na prioritetu. Ovaj gospodar koji je fizički bliži centralnom arbitru ima viši prioritet. Ako potencijalni gospodar magistrale ne zahteva dodelu on predaje signal dodele narednom gospodaru u lancu sve dok se ne naiđe na gospodara koji je izdao zahtev. Gospodar koji je izdao zahtev ne prenosi dalje signal dodele kroz lanac (prekida lanac) i aktivira liniju "BUS BUSY" na visoko čime ukazuje da je zahvatio magistralu. Kada tekući gospodar magistrale završi sa prenosom, postavlja liniju "BUS BUSY" na nisko, "BUS GRANT" je već postavljena na "nisko" od strane arbitra i naredni ciklus arbitraže može da počne. Prednost ove šeme je jednostavnost izvođenja, a nedostatak što je prioritet dodele određen fizičkom pozicijom. Naime, može se desiti da gospodar najnižeg prioriteta ne dobije magistralu na dodelu ako se zahtevi za dodelu od strane gospodara sa višim prioritetom često izdaju, tj. kaže se da će on "umreti od gladi".

### Nezavisni zahtevi i zahvatanja

Princip ovog metoda prikazan je na slici 8.13.



Sl. 8.13.

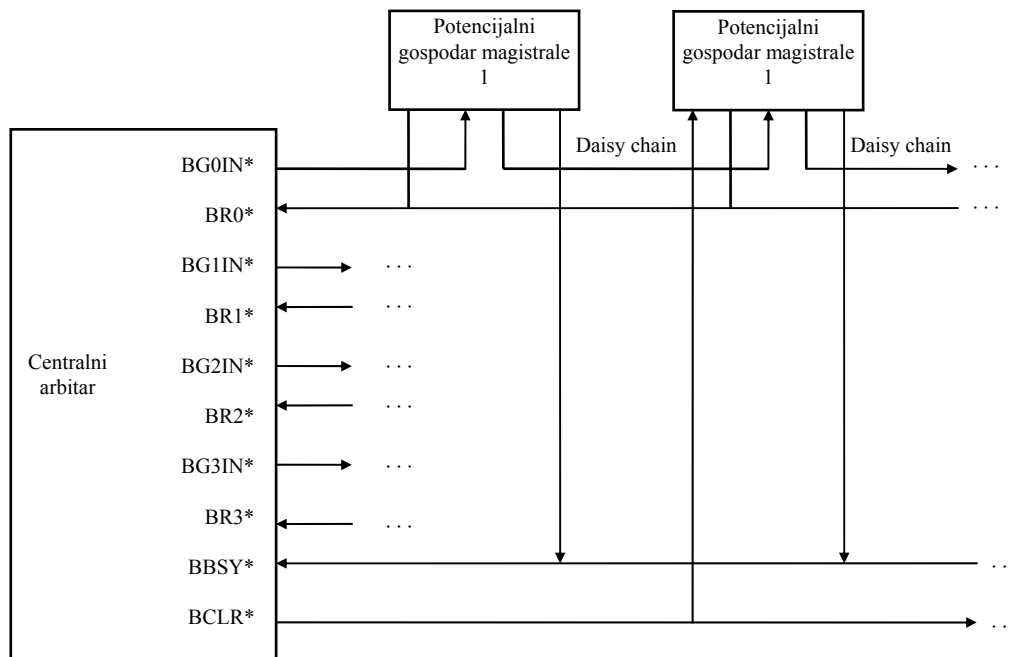
Svakom potencijalnom gospodaru magistrale se dodeljuje posebna linija za izdavanje zahteva i posebna linija za zahvatanje preko kojih se povezuje sa centralnim arbitrom. Kada potencijalni gospodar želi dodelu on aktivira svoju liniju "BUS REQUESTn". Arbitar odabira potencijalnog gospodara i aktivira odgovarajuću liniju "BUS GRANTn". Odabrani gospodar deaktivira liniju "BUS REQUEST" i aktivira liniju "BUS BUSY" ukazujući drugim gospodarima da je on taj koji trenutno koristi magistralu. Kada tekući gospodar želi da oslobodi magistralu on deaktivira liniju "BUS BUSY", nakon čega arbitar može dodeliti magistralu drugom gospodaru ako je neki od njih izdao zahtev.

Politika dodele magistrale može biti različita (kružna, sa prioritetom i dr.) i zavisi od realizacije centralnog arbitra.

Prednost ove šeme je kratko vreme arbitraže, a nedostatak je veliki broj linija za povezivanje potencijalnih gospodara magistrale i centralnog arbitra.

### Kombinovani mehanizam

Principijska šema realizacije kombinovanog mehanizma arbitraže prikazana je na slici 8.14.



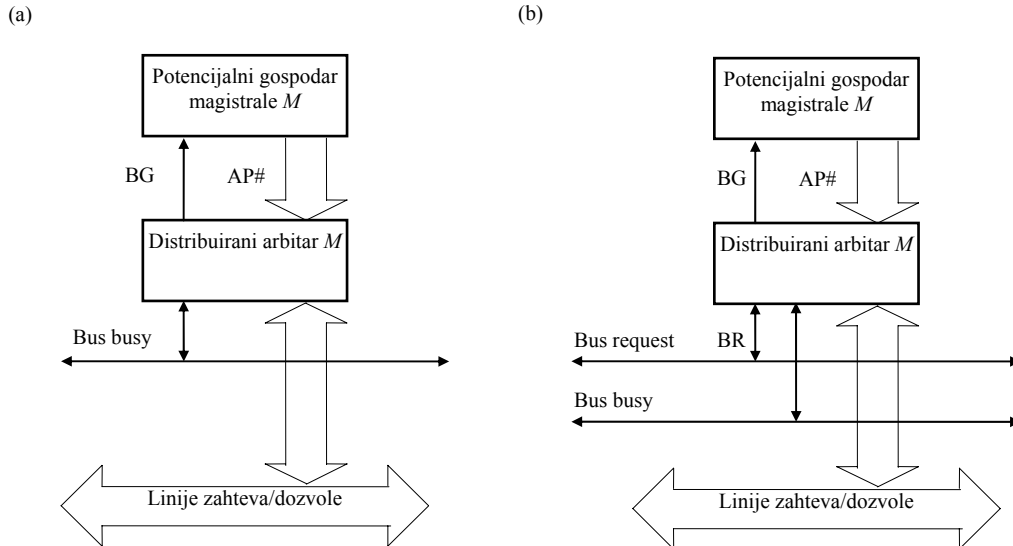
Sl. 8.14.

Sistem se zasniva na korišćenju prethodna dva arbitražna mehanizma.

### Distribuirana arbitraža

Kod ove šeme, hardver za arbitražu je raspodeljen po potencijalnim gospodarima magistrale. Ovakvo rešenje se koristi kod multiprocesorskih sistema koji rade u čvrstoj sprezi.

Princip rada je prikazan na slici 8.15.



Sl. 8.15.

Potencijalni gospodar magistrale koji zahteva upravljanje izdaje zahtev aktiviranjem linija #AP prema distribuiranom arbitru. Arbitar predaje #AP zahtev na deljive linije "REQUEST/GRANT". Svi ostali gospodarima koji izdaju zahtev za dodelu takođe predaju svoj zahtev preko svojih #AP linija. Treba istaći da se svakom #AP zahtevu dodeljuje po jedna linija na zajedničkim linijama "REQUEST/GRANT". Na ovaj način se formira zbirni #AP zahtev. Nakon ovoga, svaki distribuirani arbitar poredi svoj zahtev sa zbirnim. Ako je njegov #AP zahtev niži od zbirnog to znači da je njegov prioritet niži, tj. magistrala se dodeljuje onom gospodaru čiji je prioritet najviši.

### 8.4. Mehanizmi prekida

Prekidom se ostvaruje mehanizam kojim hardverski generisani signali uslovljavaju promenu programskog toka. Svaka CPU ima svoj sopstveni metod za implementaciju detalja koji su vezi sa obradom prekida (strukturu prekida), ali postoji nekoliko osobina koje su zajedničke za sve. Prekidi se prihvataju samo između instrukcija, tj CPU ne može da prekine rad na sredini izvršenja tekuće instrukcije (sa izuzetkom kada se javi signal greška na magistrali, ako je dostupan).

Ulazi za signalizaciju prekida mogu biti osetljivi-na-nivo (*level sensitive*) ili da vrše okidanje na ivicu (*edge-triggered*). Kod najvećeg broja CPU-ova ulazi kojima se signalizira zahtev za prekid su osetljivi-na-nivo, što znači da signal prekida mora biti aktivan sve dok se prekid ne prihvati, a ako ne, zahtev se ignoriše. Kod ulaza za prekid koji se prihvataju na ivicu, okidanje može biti na prednjoj ili zadnjoj ivici signala kojim se inicira prekid. Zahtev za prekid se pamti (lečuje) u trenutku kada se detektuje ivica. Kada se prekid prihvati, lečovani zahtev za prekid se briše. Obično se zahtevi za prekid prihvataju preko spoljnog kontrolera prekida, koji omogućava izbor okidanja na prednju ili zadnju ivicu impulsa.

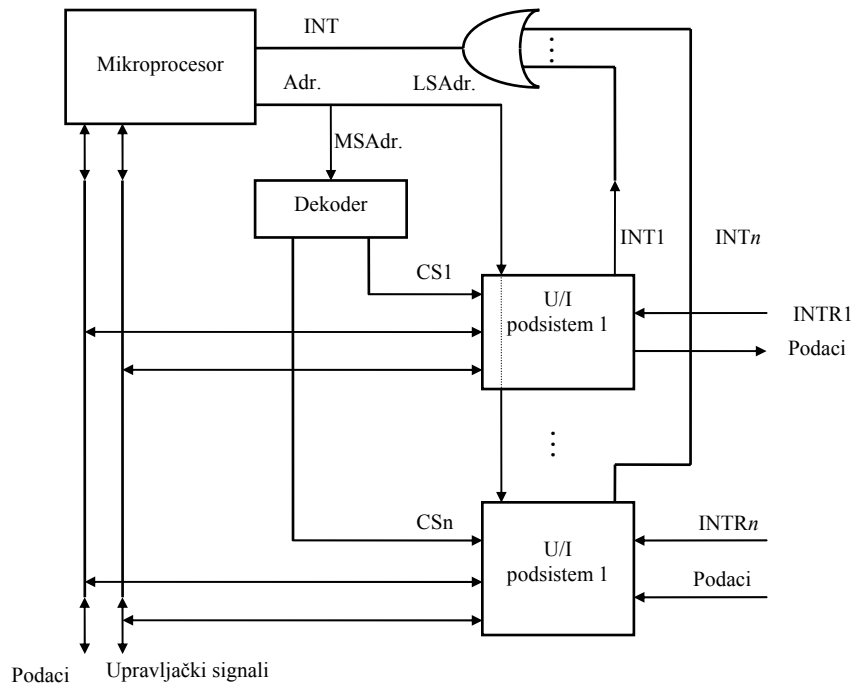
Sistemske elemente čiji rad može biti prekinut imaju ugrađen određeni hardver kojim se vrši prihvatanje zahteva za prekid. Mehanizam prekida, koji se obično realizuje na prioritetoj osnovi, može se organizovati preko namenske magistrale, ili vremenski raspodeljene magistrale pri čemu se uzima da su resursi uvek raspoloživi.

### 8.4.1. Namenska magistrala za prekid

Kod ovakvog tipa magistrale koriste se posebne linije pomoću kojih se izdaje zahtev za opsluživanje prekida. Obradu zahteva za prekid vrši poseban hardver. Tri načina prihvatanja su:

#### Metod kružne analize

Kod ovog načina (slika 8.16) svaki podsistem koji može generisati zahtev za prekid povezan je na jedinstvenu liniju kojom se izdaje zahtev. Kada procesor prihvati zahtev za prekid prelazi na izvršenje prekidne rutine. U toku prekidne rutine metodom kružne analize, tj. softverski, određuje se najpre inicijator prekida. Nakon identifikacije, prelazi se na opsluživanje uređaja koji je izdao zahtev za prekid.

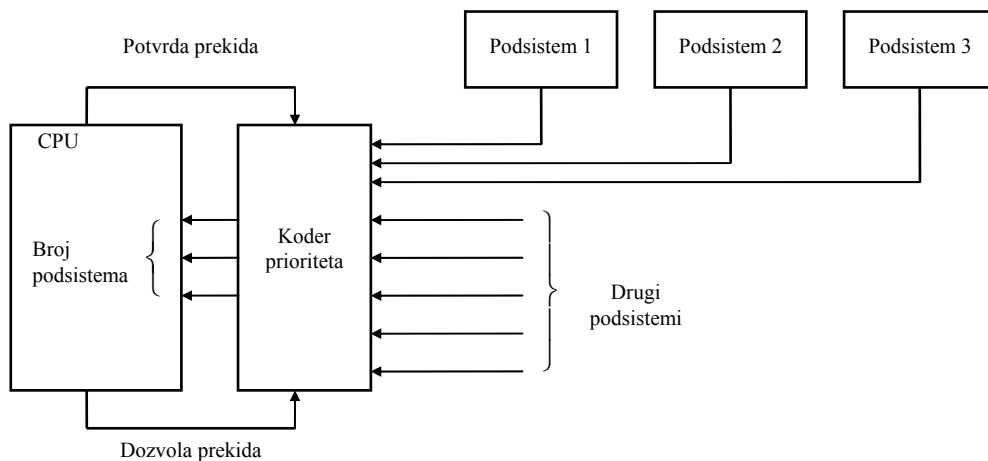


Sl. 8.16.

Prioritet opsluživanja određen je softverski i može se dinamički menjati. Prednost ove šeme ogleda se u jednostavnosti izvođenja (jednostavan hardver), a nedostatak u relativno sporom odzivu.

#### Metod sa većim brojem linija

Opšti oblik ove šeme prikazan je na slici 8.17.

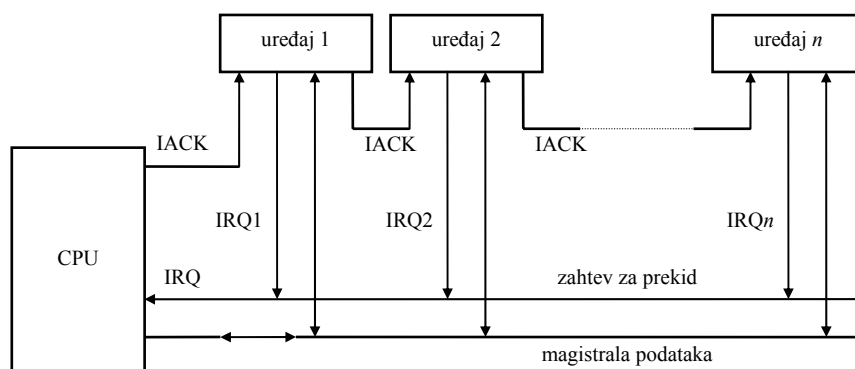


Sl. 8.17.

U ovom slučaju po jedna linija za izdavanje zahteva se dodeljuje svakom podsistemu. Ove linije su ulazi u logiku kojom se kodiraju prioriteta. Koder prioriteta je instaliran na CPU-ovoj ploči. Na svojim izlazima koder prioriteta generiše broj tekuće aktivnog ulaza sa najvišim prioriteta. Kada je samo jedna linija aktivna, njen broj se generiše na izlazu koder prioriteta. Ako je aktivan veći broj linija tada se na izlazu generiše broj koji odgovara liniji sa najvišim prioriteta. Kada je zahtev za prekid aktivan, prema CPU-u generiše se signal za prekid koga prati broj na izlazu koder prioriteta ( $I_0I_1I_2\text{ć}$ ). CPU koristi ovaj broj da bi formirao adresu rutine za obradu prekida. Na ovaj način metod kružne analize je zamenjen direktnim prelaskom na rutinu za obradu prekida. Koder prioriteta se obično realizuje kao integrisano programabilno kolo koje ima mogućnost maskiranja ulaza, generisanja zahteva za prekid sa višim nivoom u odnosu na tekući i dr.

### Metod vektorskog prekida

Principska blok šema ove metode prikazana je na slici 8.18.



Sl. 8.18.

Svi izlazi  $IRQ_1$  do  $IRQ_n$  su žičano vezani na jedinstvenu liniju  $IRQ$ . Signal  $IACK$ , priznavanje zahteva za prekid, se prostire kroz lanac, tj. iskorišćen je metod lančanja. U ovom slučaju se od podsistema koji je izdao zahtev za prekid očekuje da preda vektor broja pomoću koga se vrši njegova identifikacija. Umesto da se vektor broja generiše u okviru logike za prekid koja se instalira na CPU-ovoj ploči kao što je prikazano na slici 8.17, u ovom slučaju taj broj se predaje CPU-u od strane prekidnog podsistema, preko magistrale podataka kao deo sekvence kojom se vrši obrada tog zahteva.

#### 8.4.2. Magistrale za prekid sa vremenskom raspodelom

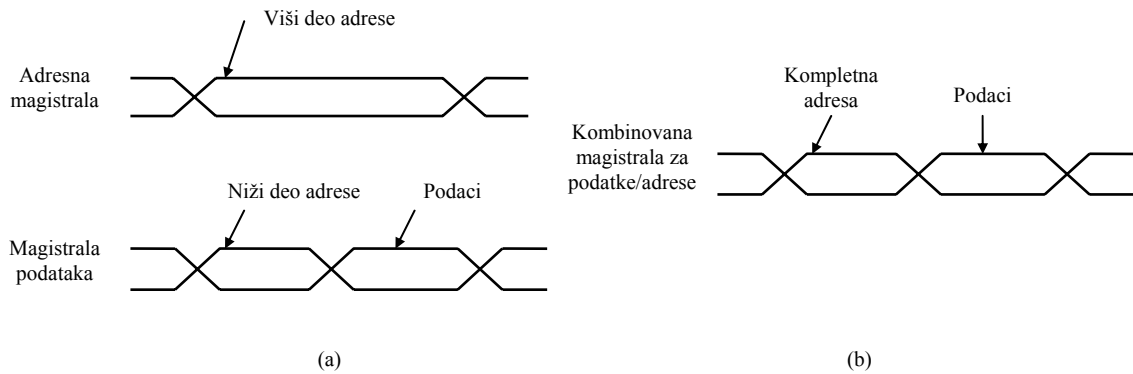
Kod ovih magistrala zahtev za prekid se izdaje kada se po magistrali pošalje poruka određene dužine. Kako u ovom slučaju ne postoji namenska magistrala za prekid, ove prekide zovemo *virtuelni prekidi*. Poruka obično sadrži: određenu adresu, izvornu adresu i druge informacije. To znači da bilo koji modul može da prekine rad drugog modula. Prenos podataka po magistrali reguliše se arbitražnom logikom, tj. nije potrebno ugrađivati dodatni hardver. Novina kod ovog pristupa je što svaki uređaj treba da ima svoju adresu.

### 8.5. Multipleksiranje podataka na magistrali

Često se u cilju smanjivanja broja linija na magistrali vrši vremensko multipleksiranje, tj. neki od signala se prenose preko istih linija na magistrali u različitim trenucima. Neka od standardnih rešenja multipleksiranja su:

- multipleksiranje adresnih linija i linija za podatke,
- multipleksiranje prekidnih linija preko adresnih linija ili linija za podatke (virtuelni prekidi),
- multipleksiranje linija za arbitražu preko linija za adrese ili podatke.

Obično se multipleksiranje adresnih linija i linija za podatke vrši na način prikazan na slici 8.19.



Sl. 8.19.

Princip multipleksiranja linija za arbitražu preko adresnih linija ili linija za podatke prikazan je na slici 8.20.



Sl. 8.20.

Svaku transakciju na magistrali čine tri ciklusa:

- ciklus komande/adresa - kod identifikuje tip komande koji se predaje, nakon toga sledi adresa sluge. Moguće komande su: čitanje, upis, prekid emisija svima i dr.
- arbitraža - koristi vreme potrebno modulu za dekodiranje adrese kako bi se rešio problem arbitraže magistrale za narednu transakciju.
- ciklus podataka - vrši se prenos podataka.

## 8.6. Detekcija greške i signalizacija

Ako se u toku transakcije na magistrali javi greška, ona treba da se detektuje. Greške koje se javljaju mogu biti sledećeg tipa:

- **Adresna greška** - adresira se nepostojeća lokacija, ili se upisuje u oblast za koju je dozvoljena samo operacija čitanje i dr. Obično se za detkciju ovakve greške koristi poseban hardver (kod sinhronih magistrala generiše se signal greška na magistrali - BUS ERROR). Obrada ovakvog tipa grešaka vrši se od strane rutine za obradu greške koja je deo operativnog sistema.
- **Greška u toku prenosa podataka** - ovaj tip greške se javlja usled smetnji, preslušavanja na vodovima ili nekorektnog rada bloka za napajanje. Obično se za detekciju greške koristi parnost. Ako se greška detektuje u najvećem broju slučajeva ciklus se ponavlja.
- **Arbitražna greška** - specifična je kod rada multiprocesorskih sistema. Mehanizam za otkrivanje greške koristi tehniku isteka vremena (*time out*) nakon čega se ukida zahtev za dodelu magistrale, a zatim izdaje novi zahtev. Ako se i posle drugog pokušaja magistrala ne dodeli, preuzimaju se odgovarajuće akcije od strane lokalnog operativnog sistema gospodara koji je izdao zahtev za dodelu magistrale.

### **8.7. Ostali signali na magistrali**

Pored signala za prenos adresa, podataka, arbitražnih signala i signala prekida, magistralom se prenose i drugi tipovi signala kao što su:

- napajanje ( $\pm V$  i masa),
- inicijalizacija sistema i reset,
- sistemski takt, i dr.