

**JOVAN ĐORĐEVIĆ**

**ARHITEKTURA  
I  
ORGANIZACIJA  
RAČUNARA  
MAGISTRALA**

**BEOGRAD, 2014.**



DJM



# PREDGOVOR

Ova knjiga je napisana kao osnovni udžbenik iz arhitekture i organizacije računara i pokriva osnovne koncepte iz arhitekture i organizacije procesora, memorije, ulaza/izlaza i magistrale.

Autor

Beograd  
marta 2014.



# SADRŽAJ

<b>PREDGOVOR</b> .....	<b>1</b>
<b>SADRŽAJ</b> .....	<b>3</b>
<b>1 MAGISTRALA</b> .....	<b>1</b>
1.1 OSNOVNI POJMOVI.....	1
1.2 ARBITRACIJA.....	5
1.2.1 PROCESOR.....	5
1.2.2 ARBITRATOR.....	6
1.3 CIKLUSI NA MAGISTRALI.....	14
1.3.1 MAGISTRALE SA ATOMSKIM CIKLUSIMA.....	14
1.3.1.1 ASINHRONA MAGISTRALA.....	14
1.3.1.2 SINHRONA MAGISTRALA.....	16
1.3.2 MAGISTRALE SA PODELJENIM CIKLUSIMA.....	19
1.4 SISTEMI SA VIŠE MAGISTRALA.....	25
1.5 PRILOZI.....	27
1.5.1 VREMENSKI OBLICI SIGNALA PRI ARBITRACIJI SA PRAĆENJEM.....	27
1.5.2 ASINHRONA MAGISTRALA SA ATOMSKIM CIKLUSIMA.....	29
1.5.2.1 PROCESOR.....	30
1.5.2.2 MEMORIJA.....	37
1.5.3 SINHRONA MAGISTRALA SA ATOMSKIM CIKLUSIMA.....	39
1.5.3.1 PROCESOR.....	39
1.5.3.2 MEMORIJA.....	40
1.5.4 MAGISTRALE SA PODELJENIM CIKLUSIMA.....	41









# 1 MAGISTRALA

U ovoj glavi se razmatraju neki elementi realizacije magistrale. U okviru toga se daju osnovni pojmovi, arbitracija, ciklusi na magistrali i sistemi sa više magistrala.

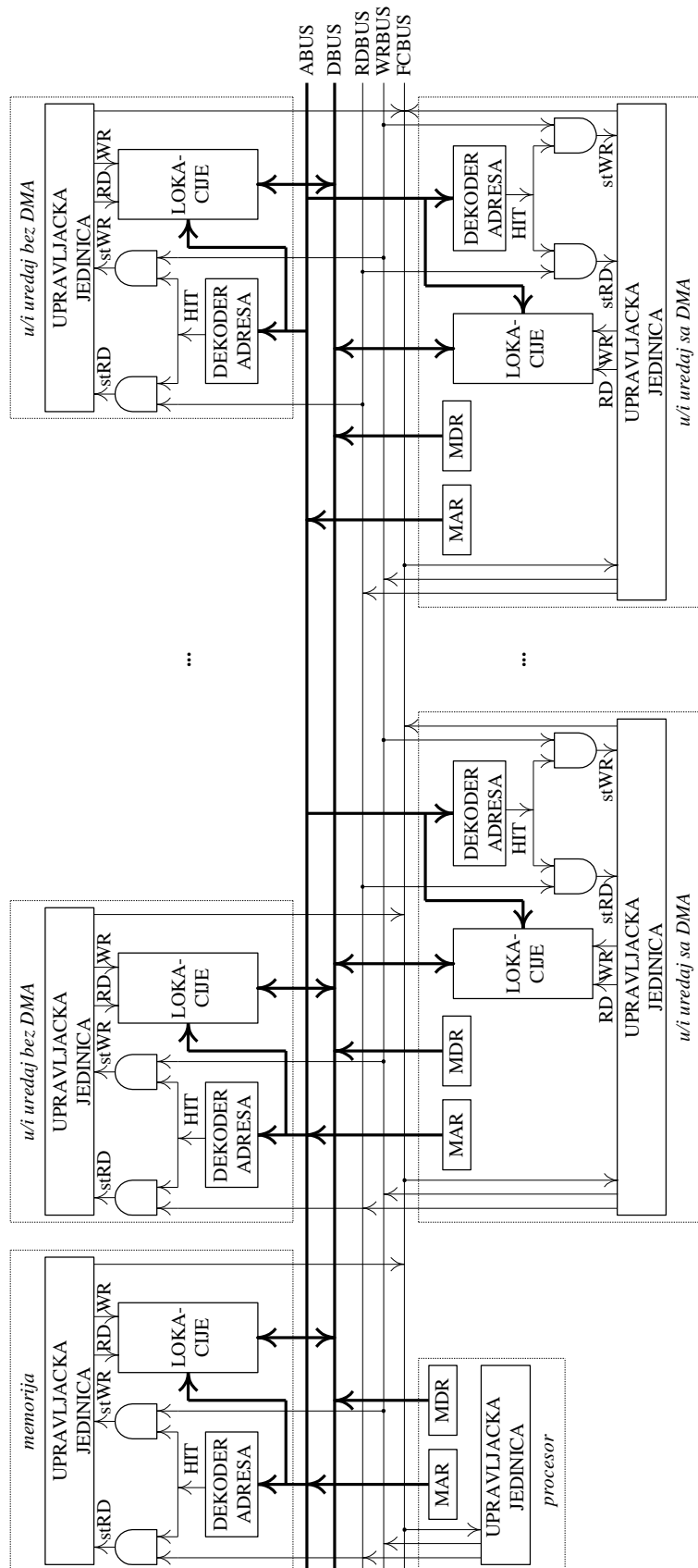
## 1.1 OSNOVNI POJMOVI

Magistrala je uređena grupa linija koja služi za povezivanje modula računarskog sistema i to procesora, memorije i ulazno/izlaznih uređaja. Preko magistrale se prenose sadržaji između registara procesora, memorijskih lokacija i registara ulazno/izlaznih uređaja. Ceo tok prenosa nekog sadržaja između dva modula, naziva se ciklus na magistrali. Modul koji započinje ciklus na magistrali, naziva se gazda (*master*), a modul sa kojim gazda realizuje ciklus, naziva se sluga (*slave*). Gazda može da bude procesor i uređaji sa direktnim pristupom memoriji. Sluga može da bude memorija i uređaji bez i sa direktnim pristupom memoriji. Procesor čita sadržaje memorijskih lokacija i upisuje sadržaje u memorijske lokacije prilikom čitanja instrukcija i operanada i upisa rezultata, kao sastavnog dela izvršavanja instrukcija kojima se vrše određena računanja. Procesor čita sadržaje registara uređaja i upisuje sadržaje u registre uređaja prilikom izvršavanja instrukcija kojima se dobija status uređaja, vrši inicijalizacija uređaja, zadaje režim rada, realizuje startovanje i zaustavljanje uređaja, i vrši prenos podataka između procesora i uređaja i obratno. Procesor prihvata brojeve ulaza u tabelu sa adresama prekidnih rutina iz registara uređaja u okviru opsluživanja zahteva za prekid. Uređaji sa direktnim pristupom memoriji čitaju sadržaje memorijskih lokacija i upisuju sadržaje u memorijske lokacije u okviru prenosa podataka iz memorije u izlazni uređaj, iz ulaznog uređaja u memoriju i iz memorije u memoriju.

Magistralu čine tri grupe linija i to adresne linije, linije podataka i upravljačke linije. Po adresnim linijama gazda šalje adresu memorijske lokacije ili registra uređaja prilikom čitanja sadržaja ili upisa sadržaja. Po linijama podataka gazda šalje sadržaj koji treba da se upiše u memorijsku lokaciju ili registar uređaja čija adresa se nalazi na adresnim linijama. Po linijama podataka sluga šalje očitani sadržaj memorijske lokacije ili registra uređaja čija adresa se nalazi na adresnim linijama. Po upravljačkim linijama gazda šalje signale kojima određuje da li treba da se realizuje čitanje ili upis sadržaja. U nekim realizacijama magistrale po upravljačkim linijama i sluga šalje signale kojima signalizira gazdi da li su čitanje ili upis sadržaja uspešno ili bezuspešno realizovani.

Slanje adrese memorijske lokacije ili registra uređaja po adresnim linijama od strane gazde prilikom čitanja ili upisa sadržaja na strani sluga naziva se adresiranje lokacija u slugi. Adresa lokacije se šalje od strane gazde po adresnim linijama svim slugama. Zadatak svih sluga je da izvrše proveru da li dati sadržaj predstavlja adresu neke od lokacije unutar datog sluga. Upravljački signal za čitanje ili upis gazda šalje svim slugama onoliko kasnije u odnosu na trenutak slanja adrese po adresnim linijama koliko je dovoljno svim slugama da utvrde da li sadržaj na adresnim linijama predstavlja adresu neke od lokacija unutar datog sluga ili ne. Ukoliko je sistem korektno realizovan samo jedan sluga bi trebalo da prepozna sadržaj na adresnim linijama kao adresu neke od svojih lokacija i da po pojavljivanju upravljačkog signala za čitanje ili upis realizuje čitanje ili upis.

Adresiranje lokacija u slugi i čitanje ili upis na strani sluga su prikazani na slici 1.



Slika 1 Adresiranje na magistrali sa memorijski preslikanim ulazno/izlaznim adresnim prostorom

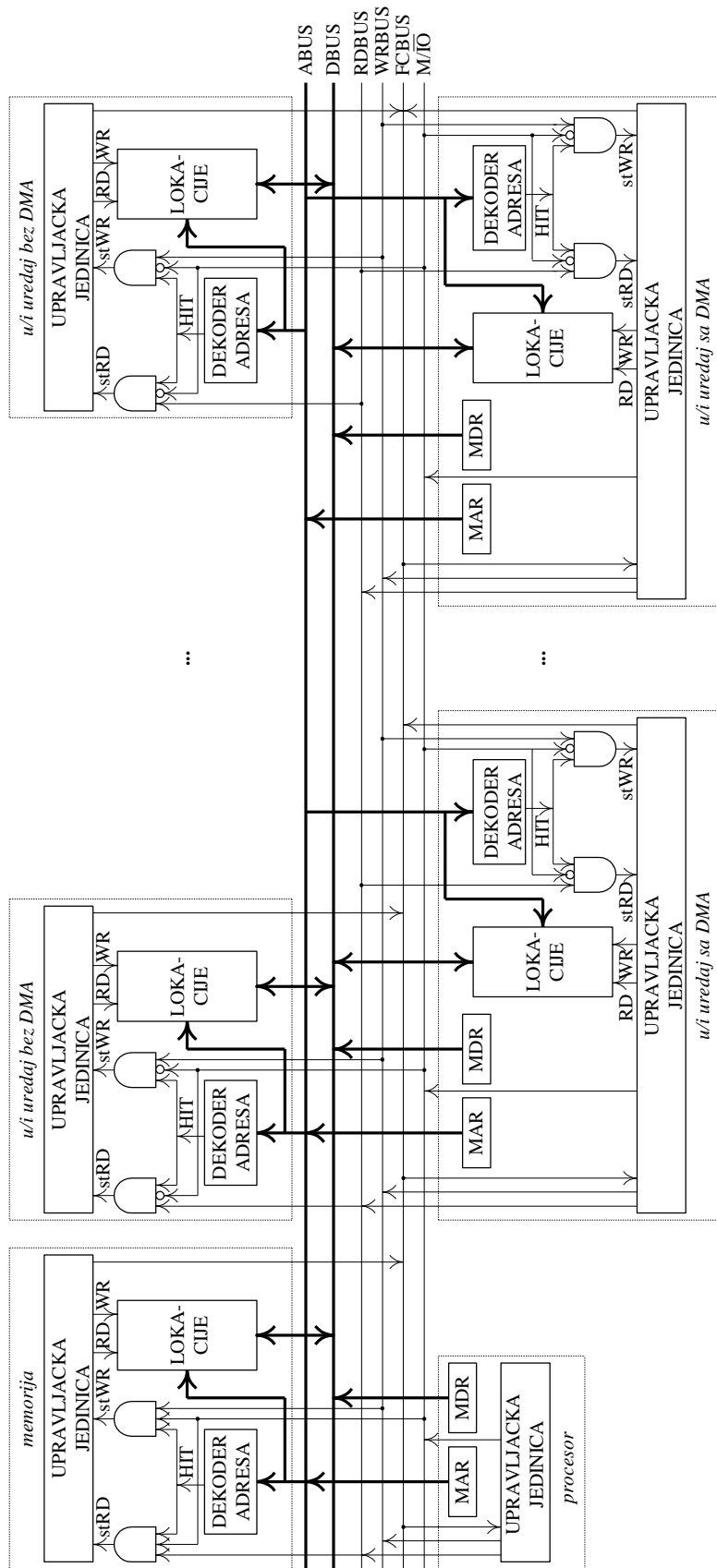
Adresu lokacije u nekom od modula gazda šalje po adresnim linijama **ABUS** magistrale i u slučaju čitanja i u slučaju upisa. Sadržaj sa adresnih linija magistrale **ABUS** primaju svi moduli i proveravaju u okviru kombinacionih mreža nazvanih *dekoder adresa* da li dati sadržaj predstavlja adresu neke od lokacija u okviru datog modula. Signal **HIT** na izlazu kombinacione mreže *dekoder adresa* ima vrednost jedan samo kod onog modula u kome se nalazi adresirana lokacija, dok u slučaju svih ostalih modula ima vrednost nula. Modul sa vrednošću jedan signala **HIT** postaje sluga sa kojim će gazda realizovati čitanje ili upis.

U slučaju čitanja upravljačka jedinica gazde generiše i šalje po upravljačkoj liniji **RDBUS** upravljački signal čitanja. Signal **RDBUS** primaju svi moduli, ali se na osnovu signala **RDBUS** samo u slugi u kome je signala **HIT** jedinica formira signal **stRD** i njime samo upravljačka jedinica sluge startuje da realizuje čitanje. Upravljačka jedinica sluge generiše signal **RD** čime se iz *lokacije* adresirane sadržajem na adresnim linijama magistrale **ABUS** čita sadržaj i šalje po linijama podataka magistrale **DBUS** svim modulima vezanim na magistralu. Istovremeno upravljačka jedinica sluge generiše i šalje po upravljačkoj liniji **FCBUS** upravljački signal završetka čitanja. Signal **FCBUS** primaju svi moduli vezani na magistralu, ali na njega reaguje samo gazda koji je generisao signal **RDBUS** tako što upisuje sadržaj sa linija **DBUS** u svoj prihvatni registar podatka **MDR** i završava čitanje.

U slučaju upisa upravljačka jedinica gazde generiše i šalje po linijama podataka magistrale **DBUS** sadržaj koji treba upisati u slugi i po upravljačkoj liniji **WRBUS** upravljački signal upisa. Sadržaj sa linija **DBUS** i signal **WRBUS** primaju svi moduli, ali se na osnovu signala **WRBUS** samo u slugi u kome je signala **HIT** jedinica formira signal **stWR** i njime samo upravljačka jedinica sluge startuje da realizuje upis. Upravljačka jedinica sluge generiše signal **WR** čime se u *lokacije* adresiranu sadržajem na adresnim linijama magistrale **ABUS** upisuje sadržaj sa linija podataka magistrale **DBUS**. Istovremeno upravljačka jedinica sluge generiše i šalje po upravljačkoj liniji **FCBUS** upravljački signal završetka upisa. Signal **FCBUS** primaju svi moduli vezani na magistralu, ali na njega reaguje samo gazda koji je generisao signal **WRBUS**, tako što završava upis.

Opseg adresa koje mogu da se koriste za adresiranje memorijskih lokacija se naziva memorijski adresni prostor, dok se opseg adresa koje mogu da se koriste za adresiranje registara uređaja naziva ulazno izlazni adresni prostor. Ukoliko se registrima uređaja pristupa istim instrukcijama kao i memorijskim lokacijama, kaže se da je ulazno/izlazni adresni prostor memorijski preslikan. Ukoliko postoje posebne instrukcije za pristup registrima uređaja i postoje posebne instrukcije za pristup memorijskim lokacijama, kaže se da su ulazno/izlazni i memorijski adresni prostori razdvojeni.

Adresiranje memorijskih lokacija i registara uređaja za slučaj kada je ulazno/izlazni adresni prostor memorijski preslikan je prikazano na slici 1. Ovde se samo vrednošću adrese generisane od strane gazde određuje da li se čitanje ili upis realizuje sa memorijskim lokacijama ili registrima uređaja. Adresiranje memorijskih lokacija i registara uređaja za slučaj kada su ulazno/izlazni adresni prostor i memorijski adresni prostor razdvojeni je prikazano na slici 2. Ovde se vrednošću posebnog signala  $M/\overline{IO}$  generisanim od strane gazde određuje da li se čitanje ili upis realizuje sa memorijskim lokacijama ili registrima uređaja. Ukoliko se usvoji da gazda postavlja signal  $M/\overline{IO}$  na vrednost jedan kada čitanje ili upis treba da se realizuje sa memorijskim lokacijama i na vrednost nula kada čitanje ili upis treba da se realizuje sa registrima uređaja, memorija treba da reaguje i realizuje čitanje ili upis kada signal  $M/\overline{IO}$  ima vrednost jedan a jedan od ulazno/izlaznih uređaja kada signal  $M/\overline{IO}$  ima vrednost nula.



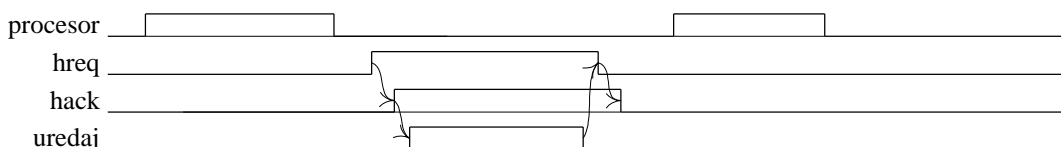
Slika 2 Adresiranje na magistrali sa razdvojenim memorijskim i ulazno/izlaznim adresnim prostorima

## 1.2 ARBITRACIJA

Arbitracija je odlučivanje o tome ko može da realizuje ciklus na magistrali. U zavisnosti od toga kako se realizuje arbitracija razlikuju se sistemi kod kojih je to funkcija procesora i sistemi kod kojih je to funkcija arbitratora.

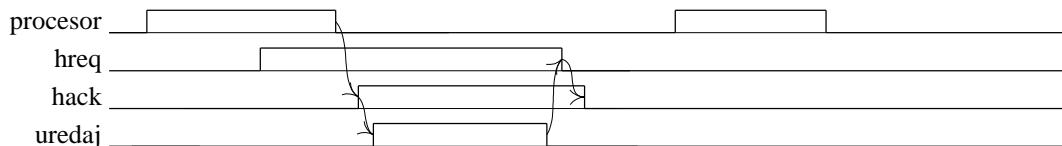
### 1.2.1 PROCESOR

Kod ovih sistema magistrala je uvek u posedu procesora, pa ulazno/izlazni uređaj koji treba da realizuje ciklus na magistrali mora najpre da uputi procesoru zahtev za korišćenje magistrale i da tek po dobijanju dozvole korišćenja magistrale od procesora realizacije ciklus na magistrali. S toga između ulazno/izlaznog uređaja i procesora postoji par linija **hreq** i **hack**. Po liniju **hreq** uređaj šalje zahtev za korišćenje magistrale, a po liniji **hack** procesor šalje uređaju dozvolu korišćenja magistrale. Uređaj postavlja signal **hreq** na aktivnu i čeka da dobije aktivnu vrednost signala **hack**. Po dobijanju aktivne vrednosti signala **hack** uređaj realizacije ciklusa na magistrali pri čemu drži aktivnu vrednost signala **hreq** sve vreme trajanja ciklusa na magistrali. Tek po završetku ciklusa na magistrali uređaj postavlja signal **hreq** na neaktivnu vrednost, na šta mu procesor, postavljanjem signala **hack** na neaktivnu vrednost, ukida dozvolu korišćenja magistrale. Karakteristične situacije koje mogu da nastanu prilikom razmene signala **hreq** i **hack** između uređaja i procesora su prikazane na slikama 3, 4 i 5.



Slika 3 Uređaj i procesor ne čekaju

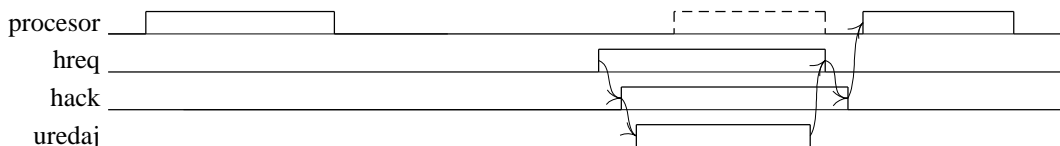
Na slici 3 je prikazana situacija u kojoj ni uređaj ni procesor ne moraju da čekaju da bi realizovali ciklus na magistrali. Na početku procesor izvesno vreme koristi magistralu (signal **procesor** je najpre aktivan pa zatim neaktivan). U trenutku kada uređaj upućuje procesoru zahtev za korišćenje magistrale (signal **hreq** aktivan) procesor ne koristi magistralu (signal **procesor** neaktivan). S toga procesor odmah daje uređaju dozvolu korišćenja magistrale (signal **hack** aktivan), pa uređaj kreće sa realizacijom ciklusa na magistrali (signal **uređaj** aktivan). Po završetku ciklusa na magistrali (signal **uređaj** neaktivan) uređaj ukida procesoru zahtev korišćenja magistrale (signal **hreq** neaktiva), pa procesor ukida uređaju dozvolu korišćenja magistrale (signal **hack** neaktivan). Kada nešto kasnije procesoru ponovo zatreba magistrala (signal **procesor** aktivan), procesor može odmah da krene sa realizacijom ciklusa na magistrali. Ovo je idealna situacija sa aspekta iskorišćenja magistrale jer uređaj i procesor ne čekaju, već kada im je magistrala potrebna odmah kreću sa realizacijom ciklusa na magistrali.



Slika 4 Uređaj čeka procesor

Na slici 4 je prikazana situacija u kojoj uređaj tek pošto sačeka da procesor završi ciklus na magistrali kreće sa realizacijom svog ciklusa na magistrali. Na početku procesor izvesno vreme koristi magistralu (signal **procesor** je aktivan). U trenutku kada uređaj upućuje procesoru zahtev za korišćenje magistrale (signal **hreq** aktivan) procesor još uvek koristi magistralu (signal **procesor** aktivan), pa procesor ne daje odmah uređaju dozvolu korišćenja

magistrale (signal **hack** neaktivan). S toga procesor tek kada završi svoj ciklus na magistrali (signal **procesor** neaktivan) daje uređaju dozvolu korišćenja magistrale (signal **hack** aktivan), pa uređaj kreće sa realizacijom ciklusa na magistrali (signal **uređaj** aktivan). Po završetku ciklusa na magistrali (signal **uređaj** neaktivan) uređaj ukida procesoru zahtev korišćenja magistrale (signal **hreq** neaktivan), pa procesor ukida uređaju dozvolu korišćenja magistrale (signal **hack** neaktivan). Kada nešto kasnije procesoru ponovo zatreba magistrala (signal **procesor** aktivan), procesor može odmah da krene sa realizacijom ciklusa na magistrali. Ovo je konfliktna situacija koja se rešava time što uređaj čeka da procesor završi započeti ciklus na magistrali.



Slika 5 Procesor čeka uređaj

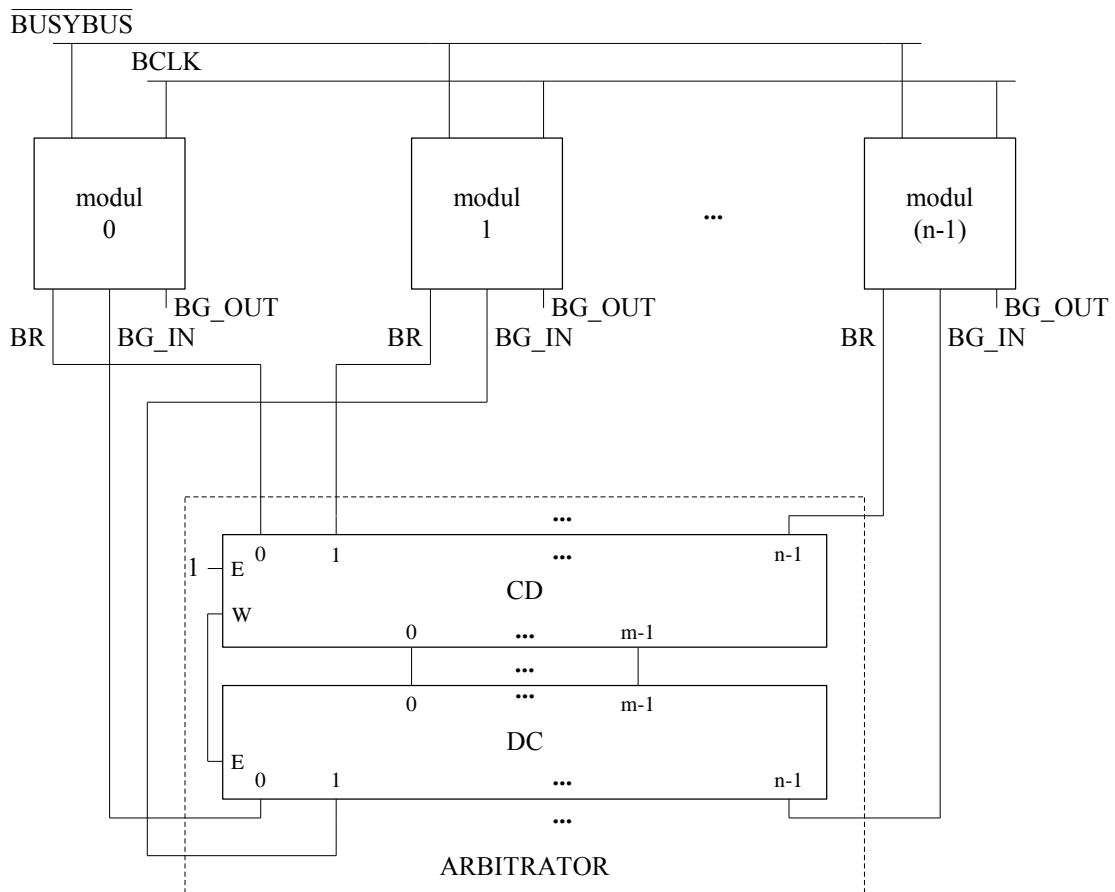
Na slici 5 je prikazana situacija u kojoj procesor tek pošto sačeka da uređaj završi ciklus na magistrali kreće sa realizacijom svog ciklusa na magistrali. Na početku procesor izvesno vreme koristi magistralu (signal **procesor** je najpre aktivan pa zatim neaktivan). U trenutku kada uređaj upućuje procesoru zahtev za korišćenje magistrale (signal **hreq** aktivan) procesor ne koristi magistralu (signal **procesor** neaktivan). S toga procesor odmah daje uređaju dozvolu korišćenja magistrale (signal **hack** aktivan), pa uređaj kreće sa realizacijom ciklusa na magistrali (signal **uređaj** aktivan). U toku trajanja ciklusa na magistrali (signal **uređaj** aktivan) procesoru je potrebna magistrala (isprekidane linije za signal **procesor** aktivan), ali ne može da je koristi sve dok uređaj ne završi tekući ciklus. Po završetku ciklusa na magistrali (signal **uređaj** neaktivan) uređaj ukida procesoru zahtev korišćenja magistrale (signal **hreq** neaktivan), pa procesor ukida uređaju dozvolu korišćenja magistrale (signal **hack** neaktivan). Tek sada procesor može da krene sa realizacijom svog sledećeg ciklusa na magistrali (signal **procesor** aktivan). Ovo je konfliktna situacija koja se rešava time što procesor čeka da uređaj završi započeti ciklus na magistrali.

## 1.2.2 ARBITRATOR

Kod ovih sistema svi moduli koji žele da realizuju neki ciklus na magistrali, a to su procesor i ulazno/izlazni uređaj sa direktnim pristupom memoriji, moraju najpre da učestvuju u arbitraciji, pa tek potom modul koji dobije dozvolu korišćenja magistrale može da realizuje ciklus na magistrali. U zavisnosti od toga kako se utvrđuje koji od modula dobija dozvolu korišćenja magistrale razlikuju se paralelna i serijska arbitracija.

U slučaju paralelne arbitracije postoji poseban uređaj koji se naziva arbitrator i koji se sastoji od koda prioriteta CD i dekodera DC (slika 6). Između modula koji žele da realizuju neki ciklus na magistrali i arbitratora postoji par linija BR i BG\_IN. Po liniju BR modul šalje zahtev za korišćenje magistrale, a po liniji BG\_IN arbitrator šalje modulu dozvolu korišćenja magistrale. Uzeto je da arbitrator ima n linija ulaza i izlaza, pri čemu je  $n=2^m$ , i da je linija 0 najvišeg a linija (n-1) najnižeg prioriteta. U zavisnosti od toga na koju od n linija arbitratora su povezane linije BR i BG\_IN modula određuje se prioritet modula. S toga je modul koji je povezan na liniju 0 najvišeg a modul koji je povezan na liniju (n-1) najnižeg prioriteta. Ukoliko zahtevi za korišćenje magistrale stignu po više linija BR istovremeno, dozvola se daju samo po jednog liniji BG\_IN i to po onoj koja odgovara liniji BR najvišeg prioriteta po kojoj je upućen zahtev za korišćenje magistrale.

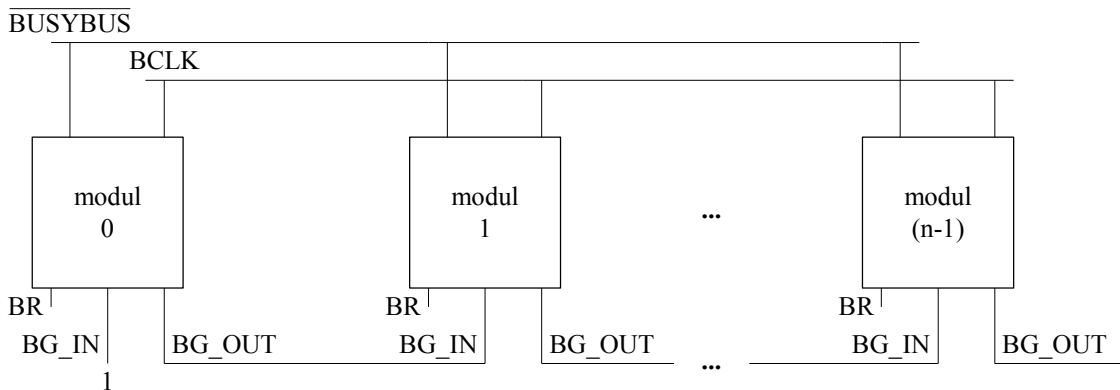




Slika 6 Paralelna arbitracija

U slučaju serijske arbitracije moduli su povezani u lanac linijama BG\_IN i BG\_OUT (slika 7). Po liniji BG\_IN koja je povezana na liniju BG\_OUT modula višeg prioriteta se dobija dozvola korišćenja magistrale od modula višeg prioriteta, a po liniji BG\_OUT koja je povezana na liniju BG\_IN modula nižeg prioriteta se prosleđuje dozvola korišćenja magistrale modulu nižeg prioriteta. U zavisnosti od toga kako su povezane linije BG\_OUT i BG\_IN modula u lancu određuje se prioritet modula. Modul koji je prvi u lancu je najvišeg, a modul koji je zadnji u lancu je najnižeg prioriteta. Na liniji BG\_IN modula prvog u lancu je 1, čime se obezbeđuje da on ima dozvolu korišćenja magistrale. Ovaj modul, kao i svaki drugi modul u lancu koji na liniji BG\_IN ima dozvolu korišćenja magistrale, vrši proveru da li on ima zahtev korišćenja magistrale. Ako da, onda će on realizovati ciklus na magistrali, a preko svoje linije BG\_OUT na liniju BG\_IN sledećeg modula u lancu poslati vrednost 0 kao indicaciju da nema prosleđivanja dozvole korišćenja magistrale. Vrednost 0 na liniji BG\_IN modula daje vrednost 0 na liniji BG\_OUT. U ovom slučaju svi preostali moduli u lancu na svojim linijama BG\_IN imaju vrednost 0 i time nemaju dozvole korišćenja magistrale. Ako ne, onda on neće realizovati ciklus na magistrali, a preko svoje linije BG\_OUT na liniju BG\_IN sledećeg modula u lancu šalje vrednost 1 kao indicaciju da on prosleđuje dozvolu korišćenja magistrale sledećem modulu u lancu. Ovaj modul se po dobijanju vrednosti 1 na svom ulazu BG\_IN ponaša na identičan nači kao prethodni modul. Ovaj modul vrši proveru da li on ima zahtev korišćenja magistrale. Ako da, onda će on realizovati ciklus na magistrali, a preko svoje linije BG\_OUT na liniju BG\_IN sledećeg modula u lancu poslati vrednost 0 kao indicaciju da nema prosleđivanja dozvole korišćenja magistrale. Ako ne, onda on neće realizovati ciklus na magistrali, a preko svoje linije BG\_OUT na liniju BG\_IN sledećeg modula u lancu šalje vrednost 1 kao indicaciju da on prosleđuje dozvolu korišćenja magistrale sledećem modulu u lancu. Na ovaj način se dozvola korišćenja magistrale prosleđuje od

modula višeg prioriteta koji nemaju zahteve korišćenja magistrale ka modulima nižeg prioriteta sve dok dozvola korišćenja magistrale ne stigne do modula koji ima zahtev korišćenja magistrale. Dati modul realizuje ciklus na magistrali, a dozvolu više ne prosleđuje.



Slika 7 Serijska arbitracija

Zahtevi korišćenja magistrale BR predstavljaju interno generisane zahteve korišćenja magistrale sinhronizovane na signal takta magistrale BCLK. Ovo se nameće kao posledica pretpostavke da svaki modul radi sa svojom internom periodom signala takta, pa interno generisani zahtevi korišćenja magistrale dobijaju vrednosti 1 i 0 u trenucima signala takta modula. Tako generisani zahtevi su vremenski nesinhronizovani pa bi, ukoliko bi se oni koristili u arbitraciji, mogli da stvore probleme i u slučaju paralelne i u slučaju serijske arbitracije. Sinhronizovanjem tih interno generisanih zahteva na signal takta magistrale BCLK obezbeđuje se da signali zahteva BR svih modula dobijaju vrednosti 1 i 0 u trenucima signala takta magistrale BCLK. Pri tome perioda signala takta magistrale BCLK mora da bude veća od propagacije kroz arbitrator u slučaju paralelne arbitracije (slika 6) i od kompletne propagacije u lancu u slučaju serijske arbitracije (slika 7).

Dok modul koji je gazda realizuje ciklus na magistrali u trajanju više perioda signala takta modula, u drugim modulima mogu da se jave zahtevi za korišćenje magistrale. S toga se paralelno sa ciklusom na magistrali odvija arbitracija i u toku trajanja ciklusa neki modul može da dobije dozvolu korišćenja magistrale. Taj modul ne sme da krene sa realizacijom svog ciklusa na magistrali dok modul koji je započeo ciklus na magistrali ne kompletira ciklus. Da bi se to osiguralo uvodi se i signal zauzeća magistrale  $\overline{\text{BUSYBUS}}$ . Vrednost 0 ovog signala označava da je magistrala zauzeta, a stanje visoke impedanse da je slobodna. Modul koji kao gazda realizuje ciklus na magistrali na početku ciklusa postavlja ovaj signal na 0, a po završetku ciklusa u stanje visoke impedanse. Modul koji dobije dozvolu korišćenja magistrale pre nego što krene sa realizacijom ciklusa na magistrali mora da proveri vrednost signala  $\overline{\text{BUSYBUS}}$ . Ukoliko signal  $\overline{\text{BUSYBUS}}$  ima vrednost 0 magistrala je zauzeta i modul koji je dobio dozvolu ne sme da krene sa realizacijom ciklusa na magistrali, već mora da sačeka da signal  $\overline{\text{BUSYBUS}}$  pređe u stanje visoke impedanse. Ukoliko je signal  $\overline{\text{BUSYBUS}}$  u stanju visoke impedanse magistrala nije zauzeta i modul koji je dobio dozvolu najpre postavlja signal  $\overline{\text{BUSYBUS}}$  na vrednost nula pa tek onda kreće sa realizacijom ciklusa na magistrali. Po završetku ciklusa na magistrali dati modul postavlja signal  $\overline{\text{BUSYBUS}}$  u stanje visoke impedanse. Time se omogućuje da modul koji je u međuvremenu dobio dozvolu korišćenja magistrale ali nije mogao da krene da realizuje ciklus na magistrali jer je magistrala bila zauzeta, sada može, postavljanjem signala  $\overline{\text{BUSYBUS}}$  na 0, da zauzme magistralu i krene sa realizacijom svog ciklusa na magistrali. Najjednostavnije rešenje je da se

signal  $\overline{\text{BUSYBUS}}$  postavlja u stanje visoke impedanse i na vrednost 0 na signal takta magistrale BCLK.

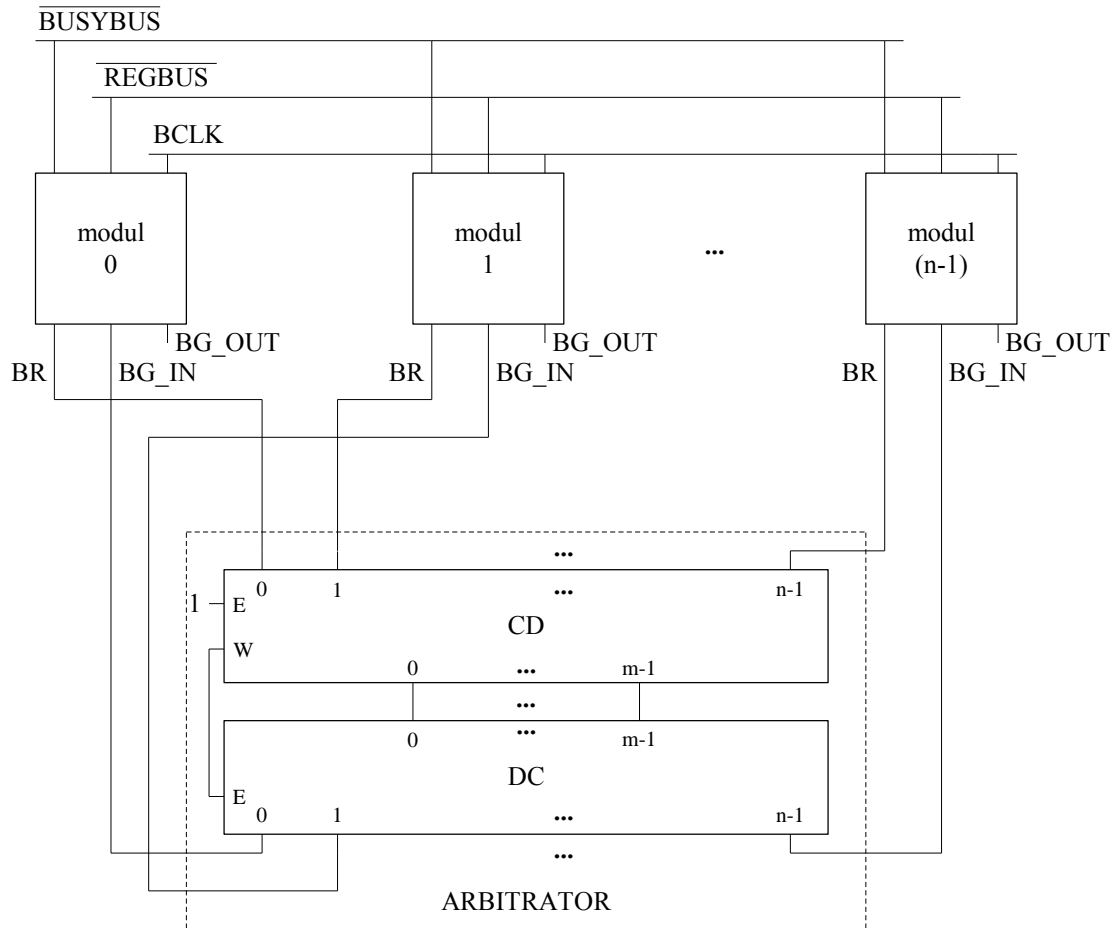
U zavisnosti od toga kada se dozvoljava modulima da učestvuju u arbitraciji razlikuju se sistemi sa praćenjem zahteva i sistemi sa pamćenjem zahteva. U slučaju sistema sa praćenjem zahteva svi moduli koji imaju zahteve korišćenja magistrale uvek učestvuju u arbitraciji. Stoga kad god je magistrala slobodna i signal  $\overline{\text{BUSYBUS}}$  u stanju visoke impedanse, modul najvišeg prioriteta koji je uputio zahtev korišćenja magistrale dobija dozvolu korišćenja magistrale. Ovim pristupom forsiraju se zahtevi od modula višeg prioriteta. Ovo ima za posledicu da zahtevi od modula nižeg prioriteta mogu veoma dugo da čekaju. Može se desiti da modul niskog prioriteta postavi zahtev i ne dobija dozvolu dok u međuvremenu nekoliko modula višeg prioriteta mogu više puta da postavljaju zahteve i dobijaju dozvole.

U slučaju sistema sa pamćenjem zahteva u određenim trenucima se za arbitraciju registruju zahtevi koji su do tog trenutka upućeni i u arbitraciji dalje učestvuju samo registrovani zahtevi. S toga kad god je magistrala slobodna i signal  $\overline{\text{BUSYBUS}}$  u stanju visoke impedanse, modul najvišeg prioriteta čiji je zahtev korišćenja magistrale u listi registrovanih zahteva dobija dozvolu korišćenja magistrale. Svaki sledeći put kada magistrala postane slobodna i signal  $\overline{\text{BUSYBUS}}$  pređe u stanje visoke impedanse, modul najvišeg prioriteta čiji je zahtev korišćenja magistrale u listi registrovanih zahteva dobija dozvolu korišćenja magistrale. U međuvremenu mogu da se pojave zahtevi i od modula višeg prioriteta od onih koji su u listi registrovanih zahteva, ali oni ne učestvuju u arbitraciji sve dok svi zahtevi koji su u listi registrovanih zahteva ne dobiju dozvole korišćenja magistrale i realizuju cikluse na magistrali. Tek kada svi zahtevi koji su u listi registrovanih zahteva dobiju dozvole korišćenja magistrale i realizuju cikluse na magistrali ponovo se za arbitraciju registruju zahtevi koji su do tog trenutka upućeni i u arbitraciji dalje učestvuju samo registrovani zahtevi. Ovim pristupom se omogućuje da i zahtevi od modula nižeg prioriteta mogu posle manjeg čekanja da dobijaju dozvole korišćenja magistrale i realizuju cikluse na magistrali. Za razliku od sistema sa praćenjem zahteva upućeni zahtevi od modula nižeg prioriteta registruju se prilikom prvog registrovanja zahteva i treba samo da sačekaju da zahtevi od modula višeg prioriteta među registrovanim zahtevima budu realizovani.

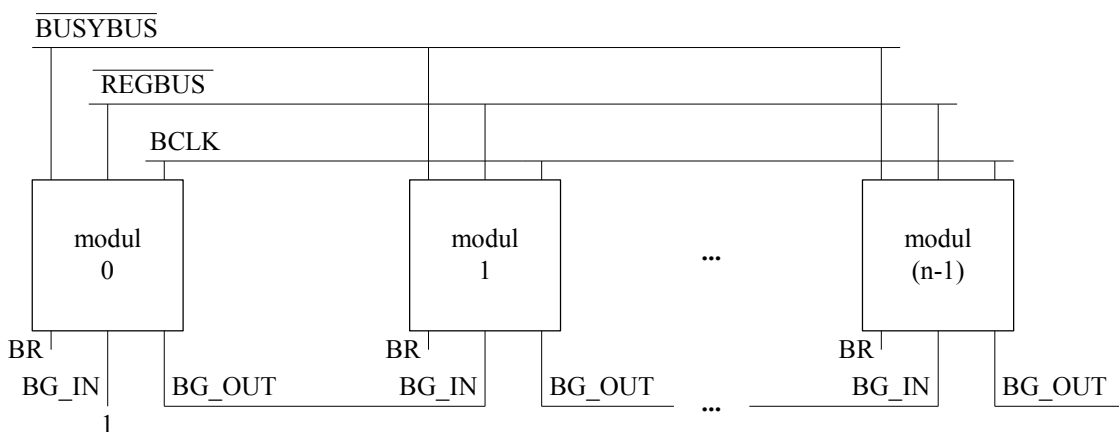
Da bi se to osiguralo uvodi se i signal registrovanih zahteva  $\overline{\text{REGBUS}}$ . Vrednost 0 ovog signala označava da ima registrovanih zahteva u listi, a stanje visoke impedanse da nema registrovanih zahteva. Modul čiji se zahtev registruje postavlja ovaj signal na 0, drži ga na 0 sve vreme dok čeka da dobije dozvolu korišćenja magistrale i dok ne realizuje ciklus na magistrali, i tek po završetku ciklusa postavlja ga u stanje visoke impedanse. Modul koji ima zahtev mora svoj zahtev prvo da registruje, pa tek posle toga može da dobije dozvolu korišćenja magistrale. Pri registraciji zahteva mora da se proverí vrednost signala  $\overline{\text{REGBUS}}$ . Ukoliko signal  $\overline{\text{REGBUS}}$  ima vrednost 0 još uvek ima registrovanih zahteva u listi i modul koji ima zahtev ne može da ga registruje, već mora da sačeka da signal  $\overline{\text{REGBUS}}$  pređe u stanje visoke impedanse. Ukoliko je signal  $\overline{\text{REGBUS}}$  u stanju visoke impedanse nema registrovanih zahteva u listi i modul koji ima zahtev može svoj zahtev da registruje tako što postavlja signal  $\overline{\text{REGBUS}}$  na vrednost nula. Modul sa registrovanim zahtevom najpre čeka da dobije dozvolu korišćenja magistrale pa tek onda kreće sa realizacijom ciklusa na magistrali. Po završetku ciklusa na magistrali dati modul postavlja signal  $\overline{\text{REGBUS}}$  u stanje visoke impedanse i time se izbacuje iz liste registrovanih zahteva. Ukoliko dati modul nije bio zadnji modul u listi registrovanih zahteva, preostali moduli drže signal  $\overline{\text{REGBUS}}$  na 0, a ako je bio zadni signal  $\overline{\text{REGBUS}}$  prelazi u stanje visoke impedanse. Tek kada signal  $\overline{\text{REGBUS}}$

pređe u stanje visoke impedanse moguća je nova registracija pristiglih zahteva. Najjednostavnije rešenje je da se signal  $\overline{\text{REGBUS}}$  postavlja u stanje visoke impedanse i na vrednost 0 na signal takta magistrale BCLK.

U sistemima sa pamćenjem zahteva paralelna i serijska arbitracija se realizuju na sličan način kao i u sistemima sa praćenjem zahteva. Razlika je samo u signalu  $\overline{\text{REGBUS}}$ .



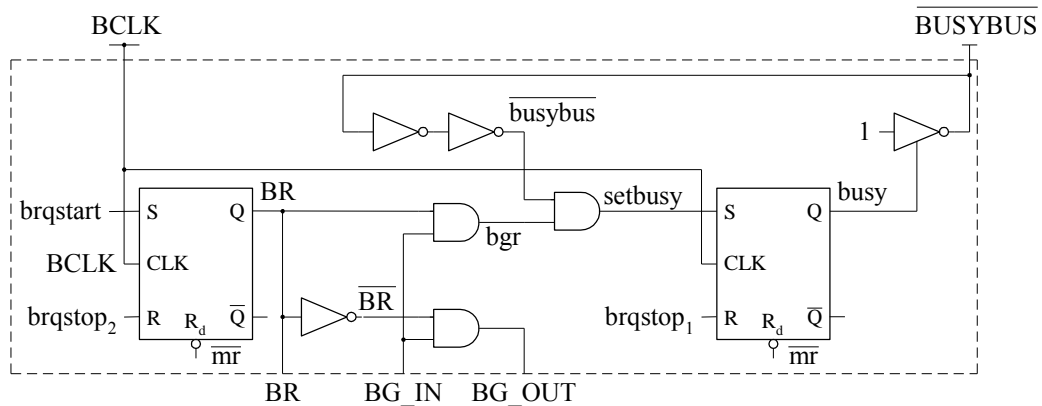
Slika 8 Paralelna arbitracija sa pamćenjem



Slika 9 Serijska arbitracija sa pamćenjem

Da bi arbitracija mogla da se realizuje svaki modul koji može da bude gazda na magistrali mora da ima određenu logike. Struktura i funkcionisanje te logike za paralelnu i serijsku arbitraciju sa praćenjem zahteva i arbitraciju sa pamćenjem zahteva se daju u daljem tekstu.

Deo modula za realizaciju paralelne i serijske arbitracije sa praćenjem zahteva je dat na slici 10.



Slika 10 Deo modula za realizaciju arbitracije sa praćenjem zahteva

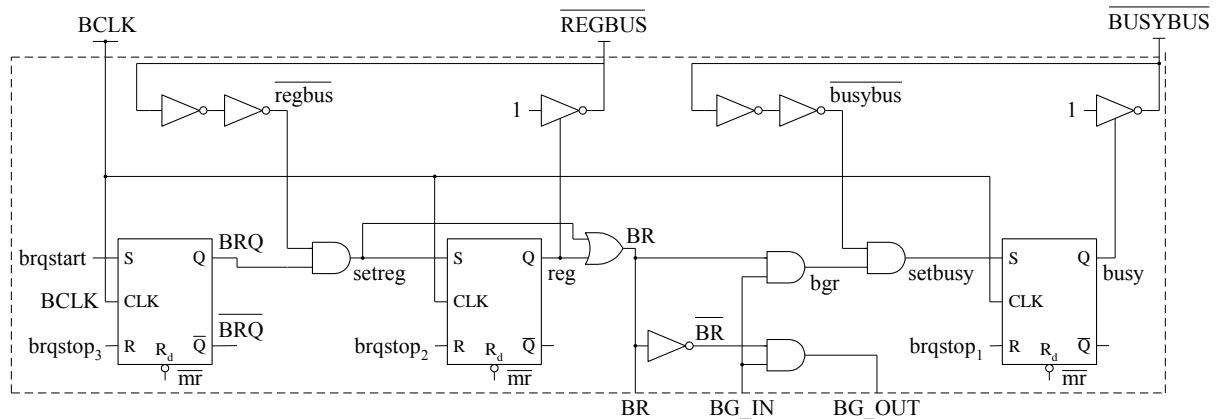
Signal **brqstart** je interno generisani zahtev korišćenja magistrale modula koji se postavlja na vrednosti 1 i 0 na signal takta modula. Signal **brqstart** ima vrednost 1 nešto duže od periode signala takta magistrale **BCLK** da bi flip-flop **BR** mogao na signal takta magistrale **BCLK** da se postavi na vrednost 1. Signal **BR** je registrovani zahtev korišćenja magistrale i predstavlja interno generisani zahtev korišćenja magistrale **brqstart** sinhronizovan na signal takta magistrale **BCLK**. U paralelnoj i serijskoj arbitraciji učestvuje registrovani zahtev korišćenja magistrale **BR**. Flip-flop **BR** ostaje postavljen na vrednost 1 najpre dok se ne dobije dozvola korišćenja magistrale i zatim dok se ne obavi kompletan ciklus na magistrali. Po završetku ciklusa na magistrali interno se generiše signal završetka korišćenja magistrale **brqstop<sub>2</sub>** koji se postavlja na vrednosti 1 i 0 na signal takta modula. Signal **brqstop<sub>2</sub>** ima vrednost 1 nešto duže od periode signala takta magistrale **BCLK** da bi flip-flop **BR** mogao na signal takta magistrale **BCLK** da se postavi na vrednost 0.

Modul kreće sa realizacijom ciklusa na magistrali kao gazda tek kada se flip-flop **busy** postavi na vrednost 1. Flip-flop **busy** se postavlja na vrednost 1 na signala takta magistrale **BCLK** onda kada signal **setbusy** ima vrednost 1. Signal **setbusy** ima vrednost 1 onda kada i signal **bgr** i signal **busybus** imaju vrednosti 1. Signal **bgr** ima vrednost 1 kada postoji registrovani zahtev korišćenja magistrale (signali **BR** ima vrednost 1) i kada je dobijena dozvola korišćenja magistrale (signal **BG\_IN** ima vrednost 1). Signal **busybus** ima vrednost 1 onda kada magistrala nije zauzeta i kada je signal **BUSYBUS** u stanju visoke impedanse (signali **busy** svih modula su 0). Signal dozvole korišćenja magistrale se dobija po liniji **BG\_IN** i u slučaju paralelne i u slučaju serijske arbitracije. U slučaju paralelne arbitracije signal dozvole se dobija od arbitratora (slika 6), a u slučaju serijske arbitracije od modula prethodnika u lancu (slika 7). U slučaju paralelne arbitracije linija sa signalom zahteva **BR** se povezuje na arbitratore (slika 6) dok se linija sa signalom **BG\_OUT** ne koristi, a u slučaju serijske arbitracije linija sa signalom zahteva **BR** se ne koristi van modula, dok se linija sa signalom **BG\_OUT** povezuje na liniju **BG\_IN** modula sledbenika u lancu (slika 7). Signal **BG\_OUT**, koji se koristi samo u serijskoj arbitraciji, postaje 1 samo kada signali **BR** i **BG\_IN** imaju vrednost 1. To se dešava kada u datom modulu ne postoji zahtev korišćenja magistrale (signal **BR** je 0) i kada je u serijskoj arbitraciji dobijena dozvola korišćenja magistrale (signal **BG\_IN** je 1). U svim ostalim slučajevima signal **BG\_OUT** je 0.

Flip-flop **BR** se u svim modulima koji učestvuju u arbitraciji postavlja na vrednosti 1 i 0 na signal takta **BCLK**. Ukoliko se u nekom modulu na i-ti signal takta **BCLK** flip-flop **BR**

postavi na 1, u datom modulu se flip-flop **busy** može postaviti na vrednost 1 na (i+1)-vi signal takta **BCLK** jedino ukoliko se u vremenskom periodu između i-tog i (i+1)-vog signala takta **BCLK** dobije dozvola korišćenja magistrale (signal **BG\_IN** ima vrednost 1) i ukoliko je zadnji gazda magistrale na neki raniji ili najkasnije na i-ti signal takta **BCLK** oslobodio magistralu (signal **busybus** postavljen na 1). Modul koji je najvišeg prioriteta među modulima koji na i-ti signal takta magistrale **BCLK** postavljaju flip-flopove **BR** na 1, dobija dozvolu korišćenja magistrale **BG\_IN**, pa pošto samo u njemu signali **bgr** i **setbusy** dobijaju vrednost 1, na (i+1)-vi signal takta **BCLK** u njemu se flip-flop **busy** postavlja na vrednost 1. Time se signal zauzeća magistrale  $\overline{\text{BUSYBUS}}$  postavlja na vrednost 0 i kreće sa realizacijom ciklusa na magistrali. Po završetku ciklusa na magistrali interno se generišu signali završetka korišćenja magistrale **brqstop<sub>2</sub>** i **brqstop<sub>1</sub>** koji se postavljaju na vrednosti 1 i 0 na signal takta modula. Signali **brqstop<sub>2</sub>** i **brqstop<sub>1</sub>** imaju vrednost 1 nešto duže od periode signala takta magistrale **BCLK** da bi flip-flopovi **busy** i **BR** mogli na signal takta magistrale **BCLK** da se postave na vrednost 0.

Deo modula za realizaciju paralelne i serijske arbitracije sa pamćenjem zahteva je dat na slici 11.



Slika 11 Deo modula za realizaciju arbitracije sa pamćenjem zahteva

Modul za realizaciju arbitracije sa pamćenjem zahteva se razlikuje u odnosu na modul sa praćenjem samo u tome da postoji flip-flop BRQ koji služi za pamćenje zahteva, da postoji signal registrovanih zahteva  $\overline{\text{REGBUS}}$  i da svaki upamćeni zahtev postane registrovani zahtev, tako što se vrednost BRQ upisuje u BR, onda kada više nema registrovanih zahteva i signal  $\overline{\text{REGBUS}}$  nema više vrednost 0.

Signal **brqstart** je interno generisani zahtev korišćenja magistrale modula koji se postavlja na vrednosti 1 i 0 na signal takta modula. Signal **brqstart** ima vrednost 1 nešto duže od periode signala takta magistrale **BCLK** da bi flip-flop BRQ mogao na signal takta magistrale **BCLK** da se postavi na vrednost 1. Signal **BRQ** je upamćen zahtev korišćenja magistrale i predstavlja interno generisani zahtev korišćenja magistrale **brqstart** sinhronizovan na signal takta magistrale **BCLK**. Međutim u paralelnoj i serijskoj arbitraciji ne učestvuje upamćen zahtev korišćenja magistrale **BRQ** već registrovani zahtev korišćenja magistrale **BR**. Da bi upamćen zahtev korišćenja magistrale postao registrovani zahtev korišćenja magistrale potrebno je da se po postavljanju flip-flopa **BRQ** na vrednost 1 i flip-flop **reg** postavi na vrednost 1 i time i **BR** postane 1. Da bi se flip-flop **reg** postavio na vrednost 1 potrebno je i da signal  $\overline{\text{regbus}}$  ima vrednost 1, pri čemu signal  $\overline{\text{regbus}}$  ima vrednost 1 jedino ukoliko nema više registrovanih zahteva korišćenja magistrale i registri **reg** u preostalim modulima imaju vrednost 0. To će se desiti na onaj signal takta **BCLK** na koji je na vrednost 0 postavljen flip-flop **reg** zadnjeg modula iz spiska modula sa registrovanim zahtevima korišćenja magistrale i dati modul postavljanjem flip-flopa **busy** na vrednost 1 krenuo sa realizacijom ciklusa na magistrali. Tek kada **BRQ** i  $\overline{\text{regbus}}$  imaju vrednost 1 signal **setreg** postaje 1 pa se na signal takta **BCLK** flip-flop **reg** postavlja na 1. Najjednostavnije je da se signal **brqstop3** generiše na osnovu signala **setreg**, pa da se flip-flop **BRQ** postavlja na vrednost 0 na isti signal takta **BCLK** na koji se flip-flop **reg** postavlja na vrednost 1. Signal **BR** se normalno generiše na osnovu vrednosti signala **reg**, a realizovan je kao ILI funkcija sa signalom **setreg** da bi dobio vrednost 1 već na onaj signal takta na koji signal **setreg** postane 1 a na na prvi sledeći signal takta na koji i flip-flop **reg** postaje 1. Od trenutka kada signa **BR** postane jedan i paralelna i serijska arbitracija se odvijaju na identičan način kao i u slučaju arbitracije sa praćenjem zahteva.

## 1.3 CIKLUSI NA MAGISTRALI

U zavisnosti od toga koliko je zauzeta magistrala prilikom ciklusa, razlikuju se magistrale sa atomskim ciklusima i magistrale sa podeljenim ciklusima.

### 1.3.1 MAGISTRALNE SA ATOMSKIM CIKLUSIMA

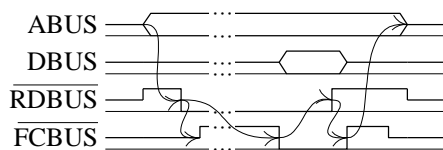
U slučaju magistrale sa atomskim ciklusima postoje ciklus čitanja, ciklus upisa i ciklus prihvatanja broja ulaza, a magistrala je zauzeta sve vreme dok se realizuje prenos podatka između gazde i sluga. U zavisnosti od toga kako se utvrđuje šta i kada gazda i sluga treba da urade prilikom realizacije ciklusa na magistrali, razlikuju se asinhrona i sinhrona magistrale.

#### 1.3.1.1 ASINHRONA MAGISTRALA

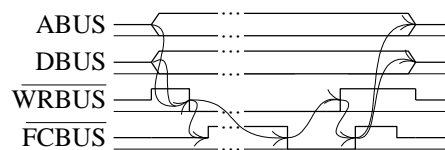
Moduli magistrale obično rade asinhrono svaki na svoj signal takta i trajanje svakog ciklusa je određeno vremenom pristupa modula sluga. Pri ciklusu čitanja, gazda šalje adresu na adresne linije **ABUS** i signalom na upravljačkoj liniji **RDBUS** startuje čitanje u slugi. Po završenom čitanju, sluga šalje očitani podatak na linije podataka **DBUS** i signalom na upravljačkoj liniji **FCBUS** gazdi signalizira da je podatak raspoloživ. Pri ciklusu upisa, gazda šalje adresu na adresne linije **ABUS**, podatak na linije podataka **DBUS** i signalom na upravljačkoj liniji **WRBUS** startuje upis u slugi. Po završenom upisu, sluga signalom na upravljačkoj liniji **FCBUS** gazdi signalizira da mu adresa i podatak nisu više potrebni. Pri ciklusu prihvatanje broja ulaza, procesor, signalom potvrde prekida **inta** na jednoj od linija, startuje čitanje sadržaja registra broja ulaza u jednom od ulazno/izlaznih uređaja. Po završenom čitanju, uređaj šalje očitani sadržaj na linije podataka **DBUS** i signalom na upravljačkoj liniji **FCBUS** procesoru signalizira da je broj ulaza raspoloživ.

Svi moduli su preko bafera sa tri stanja povezani na adresne linije, linije podataka i upravljačke linije magistrale. Pri tome na linije magistrale odgovarajuće sadržaje mogu preko bafera sa tri stanja da propušta samo modul koji je trenutno gazda i modul koji kao sluga sa njim realizuje ciklus na magistrali. Svi ostali uređaji svoje bafere sa tri stanja drže u stanju visoke impedanse i ne opterećuju linije magistrale.

Vremenski oblici signala **ABUS**, **DBUS**, **RDBUS** i **FCBUS**, koje na magistrali razmenjuju gazda i sluga prilikom realizacije ciklusa čitanja, dati su na slici 12.



Slika 12 Vremenski oblici signala za ciklus čitanja na asinhronoj magistrali



Slika 13 Vremenski oblici signala za ciklus upisa na asinhronoj magistrali

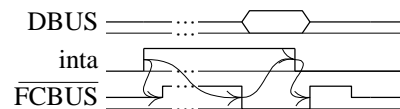
Dok ne krene sa realizacijom ciklusa čitanja gazda drži adresne linije **ABUS** i upravljačku liniju čitanja **RDBUS** u stanju visoke. To isto čini i modul koji će postati sluga sa linijama podataka **DBUS** i upravljačkom linijom **FCBUS**. Gazda kreće sa realizacijom ciklusa čitanja tako što otvara bafere sa tri stanja za adresne linije **ABUS** i upravljačku liniju čitanja **RDBUS**. Na linijama **ABUS** je adresa lokacije, a na liniji **RDBUS** vrednost 1 signala čitanja. Sadržaj sa linija **ABUS** primaju svi moduli (slika 1) i na svojim dekoderima adresa proveravaju da li dati sadržaj predstavlja adresu neke od lokacija u datom modulu. Posle vremena dovoljnog da se u svim modulima ova provera završi i samo u jednom od njih



formira vrednost 1 a u ostalima vrednosti 0 signala **HIT**, gazda postavlja liniju **RDBUS** na vrednost 0. Signal sa linije **RDBUS** primaju svi moduli (slika 1), ali na njega reaguje i za dati ciklus postaje sluga samo onaj modul u kome signal **HIT** ima vrednost 1. Sluga započinje čitanje i otvara bafere sa tri stanja za linije podataka **DBUS** i upravljačku liniju **FCBUS**. Na linijama **DBUS** je dok traje čitanje nedefinisani sadržaj, a na liniji **FCBUS** vrednost 1. Kada se u slugi završi čitanje na linijama **DBUS** se pojavljuje očitani sadržaj, a na liniji **FCBUS** vrednost 0. Na vrednost 0 signala **FCBUS** reaguje gazda i upisuje sadržaj sa linija **DBUS** u svoj prihvatni registar podatka. Po završetku upisa u prihvatni registar podatka gazda postavlja liniju **RDBUS** na vrednost 1, što je indikacija slugi da gazdi sadržaj sa linija **DBUS** nije više potreban. Na vrednost 1 signala **RDBUS** reaguje sluga tako što ukida sadržaj sa linija podataka **DBUS** i prebacuje ove linije u stanje visoke impedanse, i postavlja liniju **FCBUS** najpre na vrednost 1 a zatim i u stanje visoke impedanse. Na vrednost 1 signala **FCBUS** reaguje sluga tako što ukida sadržaj sa adresnih linija **ABUS** i prebacuje ove linije u stanje visoke impedanse. Time je ciklus čitanja kompletiran.

Vremenski oblici signala **ABUS**, **DBUS**, **WRBUS** i **FCBUS**, koje na magistrali razmenjuju gazda i sluga prilikom realizacije ciklusa upisa, dati su na slici 13. Razmena signala između gazde i sluga je veoma slična kao i za slučaj ciklusa čitanja. Jedina razlika je da sadržaj po linijama podataka **DBUS** sada šalje gazda i da umesto signala čitanja **RDBUS** gazda šalje signal upisa **WRBUS**.

Vremenski oblici signala **DBUS** i **FCBUS**, koje na magistrali razmenjuju procesor kao gazda i uređaj kao sluga, i signala potvrde **inta**, koji po posebnoj liniji koja ne pripada magistrali procesor šalje ulazno/izlaznom uređaju prilikom realizacije ciklusa prihvatanje broja ulaza, dati su na slici 14.



Slika 14 Vremenski oblici signala za ciklus prihvatanje broja ulaza na asinhronoj magistrali

Razmena signala između procesora kao gazde i nekog od uređaja kao sluga je veoma slična kao i za slučaj ciklusa čitanja. Razlika je u tome da procesor kao gazda ne koristi adresne linije **ABUS** i upravljačku liniju **RDBUS** za realizaciju ciklusa prihvatanje broja ulaza. Umesto toga postavljanjem signala **inta** na vrednost 1 i 0 procesor kao gazda započinje i završava ciklus prihvatanje broja ulaza na isti način kao što to čini postavljanjem signala **RDBUS** na vrednost 0 i 1 u slučaju ciklusa čitanja. Ulazno/izlazni uređaj na koji je linija **inta** povezana kao sluga reaguje na vrednost 1 i 0 signala **inta** na isti način na koji reaguje na vrednost 0 i 1 signala **RDBUS** u slučaju ciklusa čitanja. Sluga po linijama **DBUS** šalje broj ulaza i postavljanjem signala **FCBUS** na vrednost 0 i 1 ukazuje kada je sadržaja na linijama važeći i nevažeći.

Dobra strane asinhrono magistrale je da se signali između gazde i sluga razmenjuju na identičan način bez obzira na to koliko je vreme pristupa sluga, pa se na magistralu mogu povezivati moduli sa različitim vremenima pristupa i svakom od njih pristupiti sa njegovim vremenom pristupa. Loša strana asinhrono magistrale je da se u situacijama kada gazda i sluga rade sa posebnim signalima takta gubi vreme prilikom razmene upravljačkih signala **RDBUS**, **WRBUS** i **FCBUS**.

### 1.3.1.2 SINHRONA MAGISTRALA

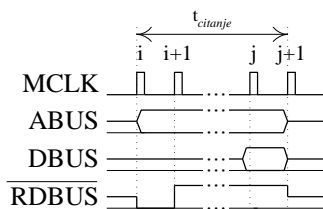
Moduli magistrale rade sinhrono na isti signal takta **MCLK**, a trajanje ciklusa može da bude fiksno ili promenljivo. Razmatra se najpre varijanta realizacije ciklusa na sinhronoj magistrali kada je trajanje ciklusa fiksno a zatim i varijanta kada je trajanje ciklusa promenljivo.

U varijanti sa sinhrono magistrale sa fiksnim trajanjem ciklusa, gazda šalje adresu na adresne linije **ABUS** i vrednošću 0 signala na upravljačkoj liniji **RDBUS** trajanja jedna perioda signala takta određuje da čitanje treba da se realizuje u slugi. U ovom slučaju je vreme čitanja fiksno, tako da gazda pretpostavlja da je po isteku tog vremena podatak raspoloživ na linijama podataka **DBUS** pa ga upisuje u svoj prihvatni registar podatka i uklanja adresu sa adresnih linija **ABUS**. Pri ciklusu upisa, gazda šalje adresu na adresne linije **ABUS**, podatak na linije podataka **DBUS** i vrednošću 0 signala na upravljačkoj liniji **WRBUS** trajanja jedna perioda signala takta određuje da upis treba da se realizuje u slugi. U ovom slučaju je vreme upisa fiksno, tako da gazda pretpostavlja da je po isteku tog vremena podatak upisan u slugi i uklanja adresu sa linija **ABUS** i podatak sa linija **DBUS**. Pri ciklusu prihvatanje broja ulaza, procesor, signalom potvrde prekida **inta** na jednoj od linija trajanja jedna perioda signala takta, određuje da čitanje sadržaja registra broja ulaza u jednom od ulazno/izlaznih uređaja kao slugi treba da se realizuje. U ovom slučaju je vreme čitanja broja ulaza fiksno, tako da gazda pretpostavlja da je po isteku tog vremena podatak raspoloživ na linijama podataka **DBUS** pa ga upisuje u svoj prihvatni registar podatka.

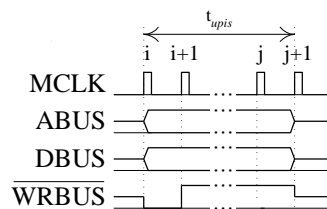
Svi moduli su kao i u slučaju asinhrono magistrale, preko bafera sa tri stanja, povezani na adresne linije, linije podataka i upravljačke linije magistrale. Pri tome, na linije magistrale odgovarajuće sadržaje mogu, preko bafera sa tri stanja, da propuštaj samo modul koji je trenutno gazda i modul koji kao sluga sa njim realizuje ciklus na magistrali. Svi ostali uređaji svoje bafere sa tri stanja drže u stanju visoke impedanse i ne opterećuju linije magistrale.

I ovde sa kao i u slučaju asinhrono magistrale adresa po linijama **ABUS** šalje svim modulima (slika 1) koji na svojim dekoderima adresa proveravaju da li dati sadržaj predstavlja adresu neke od lokacija u datom modulu. Posle vremena dovoljnog da se u svim modulima ova provera završi, a ovde je uzeto da je to vreme manje od jedne periode signala takta **MCLK**, samo u jednom od njih se formira vrednost 1 a u ostalima vrednosti 0 signala **HIT**. I ovde signal sa linije **RDBUS** primaju svi moduli (slika 1), ali na njega reaguje i za dati ciklus postaje sluga samo onaj modul u kome signal **HIT** ima vrednost 1.

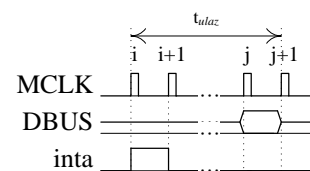
Vremenski oblici signala **ABUS**, **DBUS** i **RDBUS**, koje na magistrali razmenjuju gazda i sluga prilikom realizacije ciklusa čitanja, dati su na slici 15.



Slika 15 Vremenski oblici signala za ciklus čitanja na sinhronoj magistrali



Slika 16 Vremenski oblici signala za ciklus upisa na sinhronoj magistrali



Slika 17 Vremenski oblici signala za ciklus prihvatanje broja ulaza na sinhronoj magistrali

Dok ne krene sa realizacijom ciklusa čitanja gazda drži adresne linije **ABUS** i upravljačku liniju čitanja **RDBUS** u stanju visoke impedanse. To isto čini i modul koji će postati sluga sa

linijama podataka **DBUS**. Vreme trajanje ciklusa čitanja  $t_{\text{čitanje}}$  je fiksno i započinje na i-ti signal takta **MCLK** i završava se na (j+1)-vi signal takta **MCLK**. Između ova dva signala takta **MCLK** i gazda i sluga znaju šta ko na koji signal takta **MCLK** treba da uradi. Gazda uvek na i-ti signal takta otvara bafere sa tri stanja i pušta adresu na adresne linije **ABUS** i liniju **RDBUS** postavlja na vrednost 0, na (i+1)-vi signal takta liniju **RDBUS** postavlja na vrednost 1, na (j+1)-vi signal takta upisuje sadržaj sa linija podataka **DBUS** u svoj prihvatni registar podatka i završava ciklus na magistrali tako što adresne linije **ABUS** i upravljačku liniju čitanja **RDBUS** postavlja u stanju visoke impedanse. Svi moduli počev od i-tog signala takta primaju adresu i do pojave (i+1)-vog signala takta se u jednom od njih formira vrednost 1 signala **HIT**. Modul sa vrednošću 1 signala na **HIT** na (i+1)-vi signal takta utvrđuje da signal **RDBUS** ima vrednost 0 i kao sluga kreće sa čitanjem, najkasnije do j-tog signala takta pušta pročitani sadržaja na linije podataka **DBUS** i na (j+1)-vi signal takta završava ciklus na magistrali tako što linije podataka **DBUS** postavlja u stanju visoke impedanse.

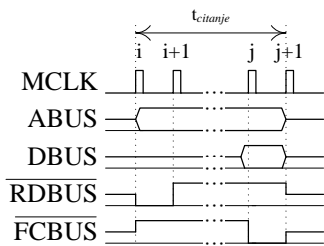
Vremenski oblici signala **ABUS**, **DBUS** i **WRBUS**, koje na magistrali razmenjuju gazda i sluga prilikom realizacije ciklusa upisa, dati su na slici 16. Razmena signala između gazde i sluge je veoma slična kao i za slučaj ciklusa čitanja. Jedina razlika je da sadržaj po linijama podataka **DBUS** sada šalje gazda i da umesto signala čitanja **RDBUS** gazda šalje signal upisa **WRBUS**.

Vremenski oblici signala **DBUS** koje na magistrali razmenjuju procesor kao gazda i uređaj kao sluga i signala potvrde **inta**, koji po posebnoj liniji koja ne pripada magistrali, procesor šalje ulazno/izlaznom uređaju prilikom realizacije ciklusa prihvatanje broja ulaza, dati su na slici 17.

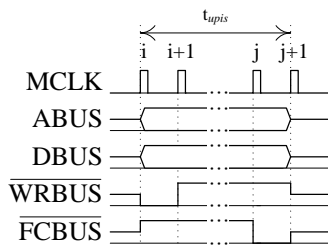
Razmena signala između procesora kao gazde i nekog od uređaja kao sluge je veoma slična kao i za slučaj ciklusa čitanja. Razlika je u tome da procesor kao gazda ne koristi adresne linije **ABUS** i upravljačku liniju **RDBUS** za realizaciju ciklusa prihvatanje broja ulaza, već samo liniju **inta**, dok sluga kao i u slučaju ciklusa čitanja šalje broj ulaza po linijama **DBUS**.

Dobra strana sinhronne magistrale je da se zbog toga što gazda i sluga rade sinhrono sa istim signalom takta i što je fiksirano šta gazda i šta sluga rade na određene signale takta, ne gubi vreme za razmenu upravljačkih signala između gazde i sluge. Loša strana sinhronne magistrale može da bude fiksno trajanje ciklusa, jer se moduli kod kojih je vreme pristupa duže od onog za koje je magistrala projektovana ne mogu dodavati, dok se sa modulima kod kojih je vreme pristupa kraće od projektovanog vremena pristupa magistrale ciklusi realizuju sa projektovanim vremenom pristupa magistrali. Ovaj problem se otklanja u varijanti sinhronne magistrale sa promenljivim trajanjem ciklusa na magistrali. Kod ove magistrale, pored linija koji postoje kao i kod sinhronne magistrale sa fiksnim trajanjem ciklusa na magistrali, postoji i linija **FCBUS** koju sluga postavlja na vrednost nula u trajanju jedna perioda signala takta **MCLK** kao indikaciju gazdi da je ciklus na magistrali završen.

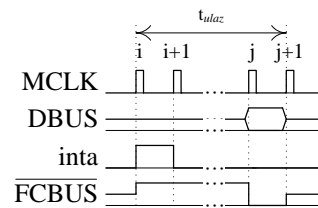
Vremenski oblici signala koje na magistrali razmenjuju gazda i sluga prilikom realizacije ciklusa čitanja, upisa i prihvatanje broja ulaza na sinhronoj magistrali sa promenljivim trajanjem ciklusa na magistrali dati su na slikama 18, 19 i 20, respektivno. Sva tri ciklusa se realizuju se realizuju na sličan način kao i u slučaju sinhronne magistrale sa fiksnim trajanjem ciklusa. Razlika je samo u tome što sluga po isteku vremena pristupa postavlja signal **FCBUS** na vrednost nula u trajanju jedna perioda signala takta **MCLK**, a gazda tek kada signal **FCBUS** postane nula završava ciklus na magistrali.



Slika 18 Vremenski oblici signala za ciklus čitanja na sinhronoj magistrali



Slika 19 Vremenski oblici signala za ciklus upisa na sinhronoj magistrali



Slika 20 Vremenski oblici signala za ciklus prihvatanje broja ulaza na sinhronoj magistrali

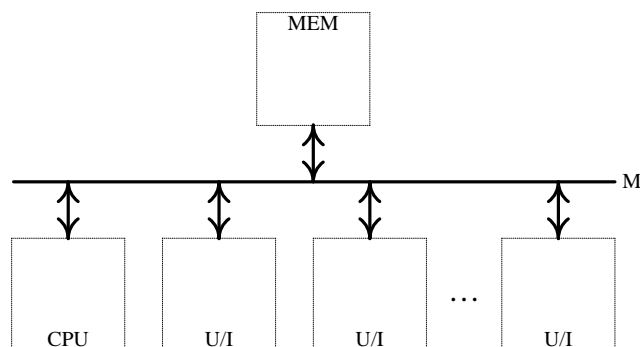
### 1.3.2 MAGISTRALNE SA PODELJENIM CIKLUSIMA

U slučaju magistrale sa podeljenim ciklusima magistrala je zauzeta samo onoliko vremena koliko je neophodno da se informacije neophodne za realizaciju čitanja ili upisa u nekom modulu kao slugi prenesu iz gazde u slugu. Dok traje samo čitanje ili upis u datom modulu magistrala je slobodna. To omogućuje da se za to vreme iskoristi magistrala da se informacije neophodne za realizaciju čitanja ili upisa u nekom modulu kao slugi prenesu iz nekog drugog gazde u slugu. Na ovaj način je moguće ostvariti da se u više modula paralelno realizuje čitanje i/ili upis. Međutim, kod ovih magistrala postoji potreba i da modul po završenom čitanju u posebnom ciklusu na magistrali u kome je sada on gazda vrati očitani podatak.

Da bi čitanje i upis moglo da se realizuju na ovakav način u memorijskim modulima koji mogu da budu sluge mora sada da postoje registri u koje će se upisivati informacije neophodne za realizaciju čitanja ili upisa. Pored toga svaki modul koji može da bude gazda mora da dobije neki broj koji će predstavljati njegov identifikator i koji će se prenositi zajedno sa drugim informacijama u poseban registar modula koji kao sluga treba da realizuje čitanje. Dati modul po završenom čitanju u posebnom ciklusu na magistrali u kome je sada on gazda koristi dobijeni identifikator da bi identifikovao modul kome treba da vrati očitani podatak. Zbog toga svaki modul mora da ima adresni registar, registar podatka i registar identifikatora.

Sami ciklusi na magistrali predstavljaju slanje relevantnih informacija iz registara gazde u registre sluge i svode se na slanje zahteva za izvršenje odgovarajuće operacije i vraćanje podatka. Stoga na magistrali sa podeljenim ciklusima mogu da se realizuju ciklus slanje zahteva za čitanje, ciklus slanje zahteva za upis, ciklus slanje zahteva za dobijanje broja ulaza i ciklus vraćanje podatka.

U dosadašnjim razmatranjima podrazumevalo se da se računarski sistem sastoji iz procesora P, određenog broja ulazno/izlaznih uređaja U/I i memorije realizovane kao jedan memorijski modul M (slika 21). Ukoliko ulazno/izlazni uređaji U/I sadrže nonDMA kontrolere (kontrolere bez direktnog pristupa memoriji), tada memoriji MEM kao gazda pristupa jedino procesor P, pa procesor može odmah da pristupi memoriji uvek kada ima zahtev da realizuje neki ciklus sa memorijom. Ukoliko neki od ulazno/izlaznih uređaja U/I sadrže DMA kontrolere (kontrolere sa direktnim pristupom memoriji), tada memoriji MEM kao gazda pristupa ne samo procesor P već i ulazno/izlazni uređaji U/I sa DMA kontrolerima, pa u situacijama kada je neki ciklus sa memorijom u toku neki od njih neće moći odmah da pristupe memoriji kada imaju zahtev da realizuju neki ciklus sa memorijom.



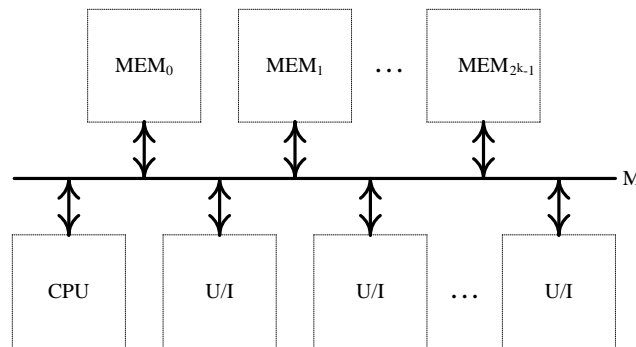
Slika 21 Računarski sistem sa memorijom MEM realizovanom kao jedan memorijski modul

U računarskim sistemima se koriste ulazno/izlazni uređaji U/I sa DMA kontrolerima sa ciljem da se poveća propusna moć celokupnog računarskog sistema. To bi trebalo da se ostvari time što DMA kontroleri preuzimaju od procesora deo poslova oko ulaza/izlaza i time rasterećuju procesor za neke druge poslove. Međutim, pune efekte korišćenja ulazno/izlaznih uređaja U/I sa DMA kontrolerima u računarskim sistemima u kojima je memorija MEM realizovana kao jedan memorijski modul (slika 21) nije moguće ostvariti. To je posledica toga što je memorija MEM realizovana kao jedan memorijski modul, pa ukoliko se desi da neki od DMA kontrolera ima zahtev da realizuje neki ciklus sa memorijom a memorija je zauzeta jer procesor ili neki od DMA kontrolera realizuje neki ciklus sa memorijom, on će morati da sačeka da se tekući ciklus sa memorijom završi. Isto važi i za procesor koji će, ukoliko ima zahtev da realizuje neki ciklus sa memorijom a memorija je zauzeta jer neki od DMA kontrolera realizuje neki ciklus sa memorijom, morati da sačeka da se tekući ciklus sa memorijom završi.

Mogući pristup za rešavanje konflikta kod istovremenih zahteva za pristup memoriji od strane procesora i DMA kontrolera je da memorija MEM ne bude realizovana kao jedan memorijski modul (slika 21), već da bude realizovana sa više memorijskih modula (slika 22). Tako memorija može da bude realizovana ne sa jednim memorijskim modulom MEM određenog kapaciteta i vremena pristupa (slika 21), već sa  $2^k$  modula  $MEM_0, MEM_1, \dots, M_{2^k-1}$  (slika 22), pri čemu je kapacitet svakog od  $2^k$  modula  $MEM_0, MEM_1, \dots, M_{2^k-1}$  (slika 22)  $2^k$  puta manji od kapaciteta modula MEM (slika 21), a vreme pristupa isto. Sada postoji mogućnost da ukoliko procesor i DMA kontroleri imaju istovremene zahteve za pristup memoriji da adrese memorijskih lokacija budu u različitim memorijskim modulima.

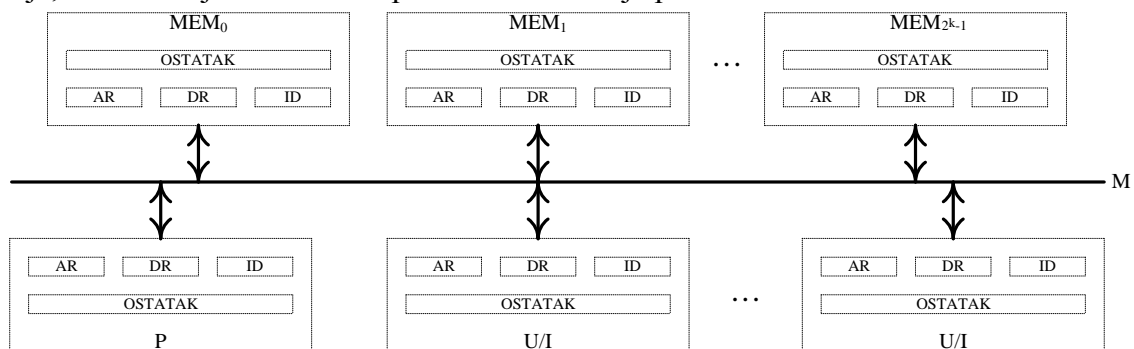
Međutim zbog toga što je magistrala realizovana sa atomskim ciklusima, ciklus na magistrali mora najpre da se obavi sa jednim memorijskim modulom, pa tek potom sa drugim memorijskim modulom. Da bi se ostvarili željeni efekti realizacijom memorije sa više modula potrebno da magistrala bude realizovan sa podeljenim ciklusima. Kod magistrale sa

podeljenim ciklusima magistrala je zauzeta samo dok se informacije potrebne za realizaciju ciklusa ne prebace iz registara gazde u registre sluge. Dok traje sam upis ili čitanje unutar modula magistrala je slobodna, pa neki drugi gazda na isti način može informacije potrebne za realizaciju ciklusa da prebaci iz svojih registara u registre sluge, ukoliko se adresa za koju se odnosi zahtev nalazi u neko drugom modulu i kome upis ili čitanje nije u toku.



Slika 22 Računarski sistem sa memorijom MEM realizovanom sa više memorijskih modula

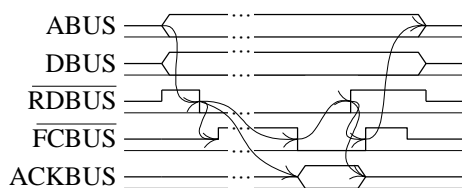
Stoga je za realizaciju upisa u neki memorijski modul potreban poseban adresni registar AR i registar podatka DR u svakom memorijskom modulu (slika 23). Prilikom slanja zahteva za upis gazda upisuje adresu i podatak u registre AR i DR, startuje operaciju upisa i oslobađa magistralu. Tek po isteku vremena pristupa memorijskog modula u memorijskom modulu može ponovo na isti način da se startuje nova operacija upisa. Međutim problem postoji sa realizacijom čitanja iz memorijskog modula. Prilikom slanja zahteva za čitanje gazda upisuje adresu u registar AR, startuje operaciju čitanja i oslobađa magistralu. Po isteku vremena pristupa memorijskog modula u registar DR memorijskom modula biće upisan pročitani podatak koji memorijski modul sada kao gazda treba u posebnom ciklusu vraćanja podatka da vrati u registar DR modula koji je prethodno poslao zahtev za čitanje. Da bi to mogao memorijski modul da uradi mora da se kod svakog slanja zahteva za čitanje u memorijski modul prenese informacija o modulu koji je poslao dati zahtev. Zato se svakom modulu koji može da bude gazda dodeljuje i smešta u registar ID jedinstveni broj koji predstavlja identifikator modula. Prilikom slanja zahteva za čitanje iz registara AR i ID gazde se u registre AR i DR sluge prenosi adresa i identifikator. Memorijski modul kada završi čitanje u posebnom ciklusu pored podatka šalje i identifikator modula koji je inicirao ovo čitanje, da bi modul koji taj identifikator prepozna kao svoj mogao da prihvati očitani podatak. Zbog toga se na magistrali sa podeljenim ciklusima realizuju tri ciklusa i to ciklu slanje zahteva za čitanje, ciklus slanje zahteva za upis i ciklus vraćanje podatka.



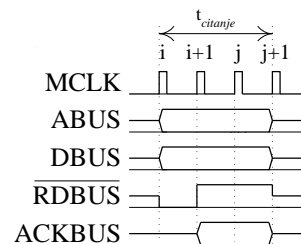
Slika 23 Računarski sistem sa memorijom MEM realizovanom sa više memorijskih modula

Magistrala može da se realizuje i kao asinhrona i kao sinhrona i u oba slučaja ima adresne linije **ABUS**, linije podataka **DBUS** i upravljačke linije **RDBUS**, **WRBUS**, **DABUS** i **ACKBUS**. U slučaju asinhronne magistrale postoji i linija **FCBUS**. Pored toga postoji i posebna linija **inta** između procesora i nekog ulazno/izlaznog uređaja koja ne pripada magistrali. Po linijama **ABUS** i **DBUS** gazda šalje slugi informacije relevantne za ciklus koji treba da se realizuje. Signalom na liniji **RDBUS** gazda određuje da se radi o ciklusu slanje zahteva za čitanje, signalom na liniji **WRBUS** o ciklusu slanje zahteva za upis, signalom na liniji **inta** o ciklusu slanje zahteva za čitanje broja ulaza i signalom na liniji **DABUS** o ciklusu vraćanje podatka. Ciklus može da bude realizovan uspešno ili bezuspešno u zavisnosti od toga da li je modul slobodan ili zauzet, na šta ukazuje aktivna i neaktivna vrednost signala **ACKBUS** koji sluga šalje gazdi. Gazda proverava vrednost signala **ACKBUS** u trenutku kada signal **FCBUS** postaje aktivan u slučaju asinhronne magistrale ili na j-ti signal takta **BCLK** u slučaju sinhronne magistrale. U slučaju da se utvrdi da je signal **ACKBUS** neaktivan ciklus nije uspešno realizovan jer je modul zauzet. Stoga se ponovo pokušava sa realizacijom datog ciklusa sve dok se ne dobije aktivna vrednost signala **ACKBUS** kao indikacija da je modul slobodan i da je ciklus uspešno realizovan.

U slučaju ciklusa slanje zahteva za čitanje gazda šalje adresu po adresnim linijama **ABUS**, identifikator po linijama podataka **DBUS** i generiše signal **RDBUS** na načina dat na slikama 24 i 25 u zavisnosti od toga da li se radi o asinhronoj ili sinhronoj magistrali.



Slika 24 Vremenski oblici signala za ciklus slanje zahteva za čitanje na asinhronoj magistrali



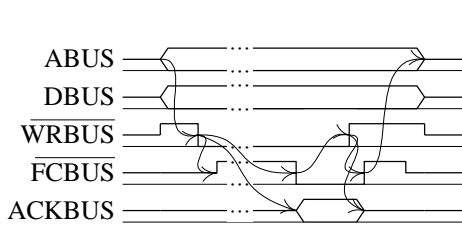
Slika 25 Vremenski oblici signala za ciklus slanje zahteva za čitanje na sinhronoj magistrali

U slučaju asinhronne magistrale (slika 24) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 12. Modul koji prepoznaje sadržaj na linijama **ABUS** kao adresu neke od svojih lokacija postaje sluga. Ukoliko je slobodan sluga koristi vrednost 0 signala **RDBUS** da najpre upiše sadržaj sa linija **ABUS** u svoj adresni registar i sadržaj sa linija **DBUS** u svoj registar identifikatora i zatim postavi signale **ACKBUS** i **FCBUS** na vrednost 1 i 0, respektivno. Na vrednost 0 signala **FCBUS** gazda proverava signal **ACKBUS** i postavlja signal **RDBUS** na vrednost 1. Ostatak ciklusa je kao na slici 12. Ukoliko nije slobodan sluga radi sve kao i kada je slobodan jedino ne upisuje sadržaje sa linija **ABUS** i **DBUS** u svoj adresni registar i registar identifikatora i signal **ACKBUS** postavlja na vrednost 0. U slučaju sinhronne magistrale (slika 25) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 15. Modul koji prepoznaje sadržaj na linijama **ABUS** kao adresu neke od svojih lokacija postaje sluga. Ukoliko je slobodan sluga koristi vrednost 0 signala **RDBUS** da na (i+1)-vi signal takta **MCLK** upiše sadržaj sa linija **ABUS** u svoj adresni registar i sadržaj sa linija **DBUS** u svoj registar identifikatora i postavi signal **ACKBUS** na vrednost 1. Na j-ti signal takta **MCLK** gazda proverava signal **ACKBUS**. Ostatak ciklusa je kao na slici 15. Ukoliko nije slobodan sluga radi sve kao i kada je slobodan

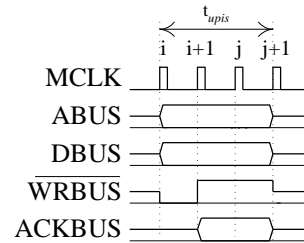


jedino ne upisuje sadržaje sa linija **ABUS** i **DBUS** u svoj adresni registar i registar identifikatora i signal **ACKBUS** postavlja na vrednost 0.

U slučaju ciklusa slanje zahteva za upis gazda šalje adresu po adresnim linijama **ABUS**, podatak po linijama podataka **DBUS** i generiše signal **WRBUS** na načina dat na slikama 26 i 27 u zavisnosti od toga da li se radi o asinhronoj ili sinhronoj magistrali.



Slika 26 Vremenski oblici signala za ciklus slanje zahteva za upis na asinhronoj magistrali

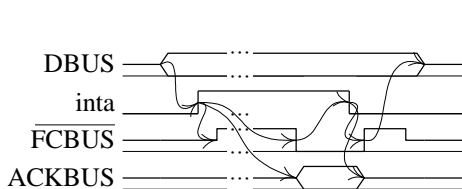


Slika 27 Vremenski oblici signala za ciklus slanje zahteva za upis na sinhronoj magistrali

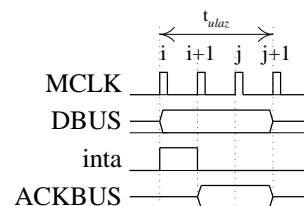
U slučaju asinhronne magistrale (slika 26) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 13. Modul koji prepoznaje sadržaj na linijama **ABUS** kao adresu neke od svojih lokacija postaje sluga. Ukoliko je slobodan sluga koristi vrednost 0 signala **WRBUS** da najpre upiše sadržaj sa linija **ABUS** u svoj adresni registar i sadržaj sa linija **DBUS** u svoj registar podatka i zatim postavi signale **ACKBUS** i **FCBUS** na vrednost 1 i 0, respektivno. Na vrednost 0 signala **FCBUS** gazda proverava signal **ACKBUS** i postavlja signal **WRBUS** na vrednost 1. Ostatak ciklusa je kao na slici 13. Ukoliko nije slobodan, sluga radi sve kao i kada je slobodan, jedino ne upisuje sadržaje sa linija **ABUS** i **DBUS** u svoj adresni registar i registar podatka i signal **ACKBUS** postavlja na vrednost 0.

U slučaju sinhronne magistrale (slika 27) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 16. Modul koji prepoznaje sadržaj na linijama **ABUS** kao adresu neke od svojih lokacija postaje sluga. Ukoliko je slobodan, sluga koristi vrednost 0 signala **WRBUS** da na (i+1)-vi signal takta **MCLK** upiše sadržaj sa linija **ABUS** u svoj adresni registar i sadržaj sa linija **DBUS** u svoj registar podatka i postavi signal **ACKBUS** na vrednost 1. Na j-ti signal takta **MCLK** gazda proverava signal **ACKBUS**. Ostatak ciklusa je kao na slici 16. Ukoliko nije slobodan, sluga radi sve kao i kada je slobodan, jedino ne upisuje sadržaje sa linija **ABUS** i **DBUS** u svoj adresni registar i registar podatka i signal **ACKBUS** postavlja na vrednost 0.

U slučaju ciklusa slanje zahteva za dobijanje broja ulaza gazda šalje identifikator po linijama podataka **DBUS** i generiše signal **inta** na načina dat na slikama 28 i 29 u zavisnosti od toga da li se radi o asinhronoj ili sinhronoj magistrali.



Slika 28 Vremenski oblici signala za ciklus slanje zahteva za dobijanje broja ulaza na asinhronoj magistrali

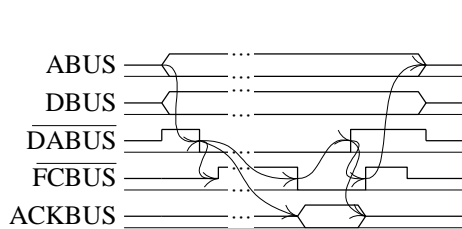


Slika 29 Vremenski oblici signala za ciklus slanje zahteva za dobijanje broja ulaza na sinhronoj magistrali

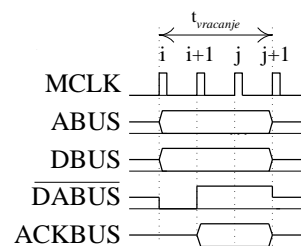
U slučaju asinhronne magistrale (slika 28) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 14. Modul koji prima signal **inta** postaje sluga. Ukoliko je slobodan, sluga koristi vrednost 1 signala **inta** da najpre upiše sadržaj sa linija **DBUS** u svoj registar identifikatora i zatim postavi signale **ACKBUS** i **FCBUS** na aktivnu vrednost 1 i 0, respektivno. Na vrednost 0 signala **FCBUS** gazda proverava signal **ACKBUS** i postavlja signal **inta** na vrednost 0. Ostatak ciklusa je kao na slici 14. Ukoliko nije slobodan sluga radi sve kao i kada je slobodan jedino ne upisuje sadržaje sa linija **DBUS** u svoj registar identifikatora i signal **ACKBUS** postavlja na vrednost 0.

U slučaju sinhronne magistrale (slika 29) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 17. Modul koji prima signal **inta** postaje sluga. Ukoliko je slobodan, sluga koristi vrednost 1 signala **inta** da na (i+1)-vi signal takta **MCLK** upiše sadržaj sa linija **DBUS** u svoj registar identifikatora i postavi signal **ACKBUS** na vrednost 1. Na j-ti signal takta **MCLK** gazda proverava signal **ACKBUS**. Ostatak ciklusa je kao na slici 17. Ukoliko nije slobodan, sluga radi sve kao i kada je slobodan, jedino ne upisuje sadržaje sa linija **DBUS** u svoj registar identifikatora i signal **ACKBUS** postavlja na vrednost 0.

U slučaju ciklusa vraćanje podatka gazda šalje identifikator po adresnim linijama **ABUS**, podatak po linijama podataka **DBUS** i generiše signal **DABUS** na načina dat na slikama 30 i 31 u zavisnosti od toga da li se radi o asinhronoj ili sinhronoj magistrali.



Slika 30 Vremenski oblici signala za ciklus vraćanje podatka na asinhronoj magistrali



Slika 31 Vremenski oblici signala za ciklus vraćanje podatka na sinhronoj magistrali

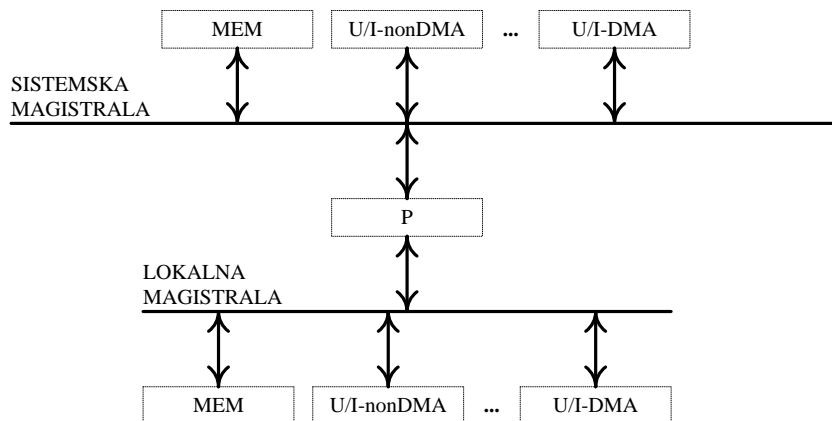
U slučaju asinhronne magistrale (slika 30) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 26. Modul koji prepoznaje sadržaj na linijama **ABUS** kao svoj identifikator postaje sluga. Ukoliko je slobodan, sluga koristi vrednost 0 signala **DABUS** da najpre upiše sadržaj sa linija **DBUS** u svoj registar podatka i zatim postavi signale **ACKBUS** i **FCBUS** na vrednost 1 i 0, respektivno. Na vrednost 0 signala **FCBUS** gazda proverava signal **ACKBUS** i postavlja signal **DABUS** na vrednost 1. Ostatak ciklusa je kao na slici 26. Ukoliko nije slobodan, sluga radi sve kao i kada je slobodan, jedino ne upisuje sadržaje sa linija **DBUS** u svoj registar podatka i signal **ACKBUS** postavlja na vrednost 0.

U slučaju sinhronne magistrale (slika 31) vremenski oblici signala su veoma slični vremenskim oblicima signala sa slike 27. Modul koji prepoznaje sadržaj na linijama **ABUS** kao svoj identifikator postaje sluga. Ukoliko je slobodan, sluga koristi vrednost 0 signala **DABUS** da na (i+1)-vi signal takta **MCLK** upiše sadržaj sa linija **DBUS** u svoj registar podatka i postavi signal **ACKBUS** na vrednost 1. Na j-ti signal takta **MCLK** gazda proverava signal **ACKBUS**. Ostatak ciklusa je kao na slici 27. Ukoliko nije slobodan, sluga radi sve kao i kada je slobodan, jedino ne upisuje sadržaje sa linija **DBUS** u svoj registar podatka i signal **ACKBUS** postavlja na vrednost 0.

## 1.4 SISTEMI SA VIŠE MAGISTRALA

Sistemi se veoma često projektuju tako da sadrže više od jedne magistrale. To se čini sa ciljem da se smanje čekanja modula koji kao gazde realizuju cikluse na magistrali i da se kombinuju dobre strane sinhronih i asinhronih magistrala.

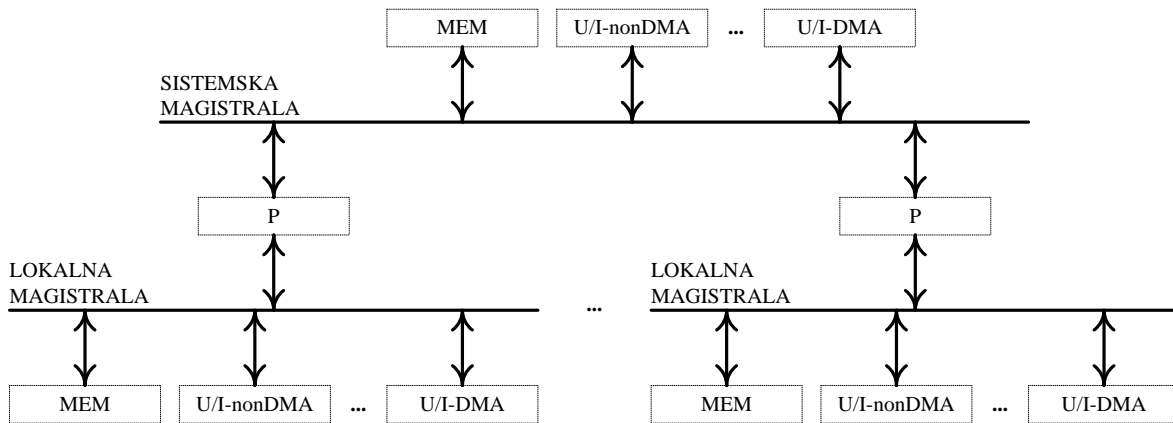
Na slici 32 je dato povezivanje procesora, memorije, u/i uređaja bez DMA i u/i uređaja sa DMA preko dve magistrale koje se nazivaju sistemska magistrala i lokalna magistrala. Jedan modul memorije i više modula u/i uređaja bez DMA i u/i uređaja sa DMA povezani su na sistemska magistralu, a drugi modul memorije i više drugih modula u/i uređaja bez DMA i u/i uređaja sa DMA povezani su na lokalnu magistralu. Procesor je vezan i na sistemska i na lokalnu magistralu. Prilikom konfigurisanja sistema adresni prostor korisnika se deli na dva dela. Adrese iz jednog dela se koriste za adresiranje lokacija modula koji su povezani na sistemska magistralu, a drugi opseg adresa za adresiranje lokacija modula koji su povezani na lokalnu magistralu. Prilikom realizaciju ciklusa na magistrali procesor, na osnovu vrednosti generisane adrese, utvrđuje da li dati ciklus treba da realizuje preko sistemske ili preko lokalne magistrale. Za realizaciju ciklusa preko sistemske i lokalne magistrale postoje posebni adresni registri, registri podataka i upravljačka logika za svaku magistralu. Povećanje propusne moći sistema se postiže time što sada postoji mogućnost da dok procesor pristupa memoriji ili u/i uređajima bez DMA ili u/i uređajima sa DMA povezanim na sistemska magistralu, u/i uređaji sa DMA povezani na lokalnu magistralu mogu da pristupaju modulu memorije povezanom na lokalnu magistralu. Na sličan način dok procesor pristupa memoriji ili u/i uređajima bez DMA ili u/i uređajima sa DMA povezanim na lokalnu magistralu, u/i uređaji sa DMA povezani na sistemska magistralu mogu da pristupaju modulu memorije povezanom na sistemska magistralu. U ovakvim sistemima se obe magistrale obično realizuju kao asinhronih magistrala.



Slika 32 Sistem sa sistemskom i lokalnom magistralom

Ovakav način povezivanja se naročito koristi u multiprocesorskim sistemima (slika 33). Svaki procesor ima svoju lokalnu magistralu, a svi procesori koriste jednu sistemska magistralu. Modulima memorije i u/i uređaja bez DMA i u/i uređaja sa DMA povezanim na neku lokalnu magistralu može da pristupa samo procesor koji je vezan na datu lokalnu magistralu, dok ostali procesori ne mogu. Modulu memorije i modulima u/i uređaja bez DMA i u/i uređaja sa DMA povezanim na sistemska magistralu mogu da pristupe svi procesori. Kod ovakvog povezivanja se očekuje da svaki procesor najčešće pristupa modulu memorije i modulima u/i uređaja bez DMA i u/i uređaja sa DMA povezanim na lokalnu magistralu, da modulu memorije povezanom na sistemska magistralu pristupa samo za potrebe sinhronizacije i komunikacije sa ostalim procesorima i da modulima u/i uređaja bez DMA i

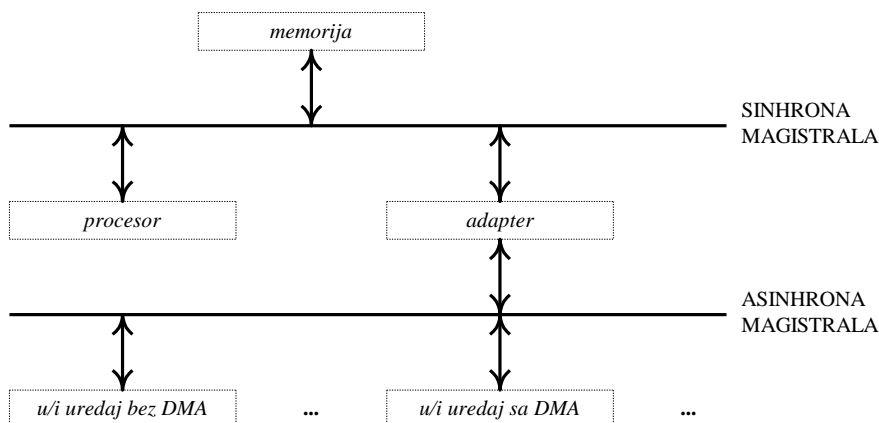
u/i uređaja sa DMA povezanim na sistemsku magistralu pristupa sporadično jer bi na sistemsku magistralu trebalo da su povezani u/i uređaji koji su skupi i koji se retko koriste.



Slika 33 Multiprocesorski sistem sa sistemskom i lokalnim magistralama

Prilikom konfigurisanja sistema adresni prostor korisnika se deli na dva dela. Adrese iz jednog dela se koriste za adresiranje lokacija modula koji su povezani na lokalnu magistralu, a drugi opseg adresa za adresiranje lokacija modula koji su povezani na sistemsku magistralu. Prilikom realizaciju ciklusa na magistrali procesor, na osnovu vrednosti generisane adrese, utvrđuje da li dati ciklus treba da realizuje preko lokalne ili preko sistemske magistrale. Opseg adresa za adresiranje lokacija modula koji su povezani na lokalnu magistralu je jedinstven za svaki procesor, dok je opseg adresa za adresiranje lokacija modula koji su povezani na sistemsku magistralu zajednički za sve procesore.

Na slici 34 je dato povezivanje procesora, memorije, u/i uređaja bez DMA i u/i uređaja sa DMA preko dve magistrale koje se nazivaju sinhrona magistrala i asinhrona magistrala. Moduli procesor i memorija povezani su na sinhronu magistralu, a moduli u/i uređaja bez DMA i u/i uređaja sa DMA povezani su na asinhronu magistralu. Modul adapter je vezan i na sinhronu i na asinhronu magistralu. Prilikom konfigurisanja sistema adresni prostor korisnika se deli na dva dela. Adrese iz jednog dela se koriste za adresiranje lokacija modula memorije koji je povezana na sinhronu magistralu, a drugi opseg adresa za adresiranje lokacija modula u/i uređaja bez DMA i u/i uređaja sa DMA koji su povezani na asinhronu magistralu. Procesor sve cikluse realizuje na sinhronoj magistrali. Pri tome na adrese iz opsega adresa dodeljenih modulu memorije reaguje modul memorije, a na adrese iz opsega adresa dodeljenih modulima u/i uređaja bez DMA i u/i uređaja sa DMA reaguje modul adapter. Ciklus sa samim u/i uređajem bez i sa DMA realizuje modul adapter. U ovakvim sistemima se sinhrona magistrala obično realizuju sa podeljenim ciklusima.

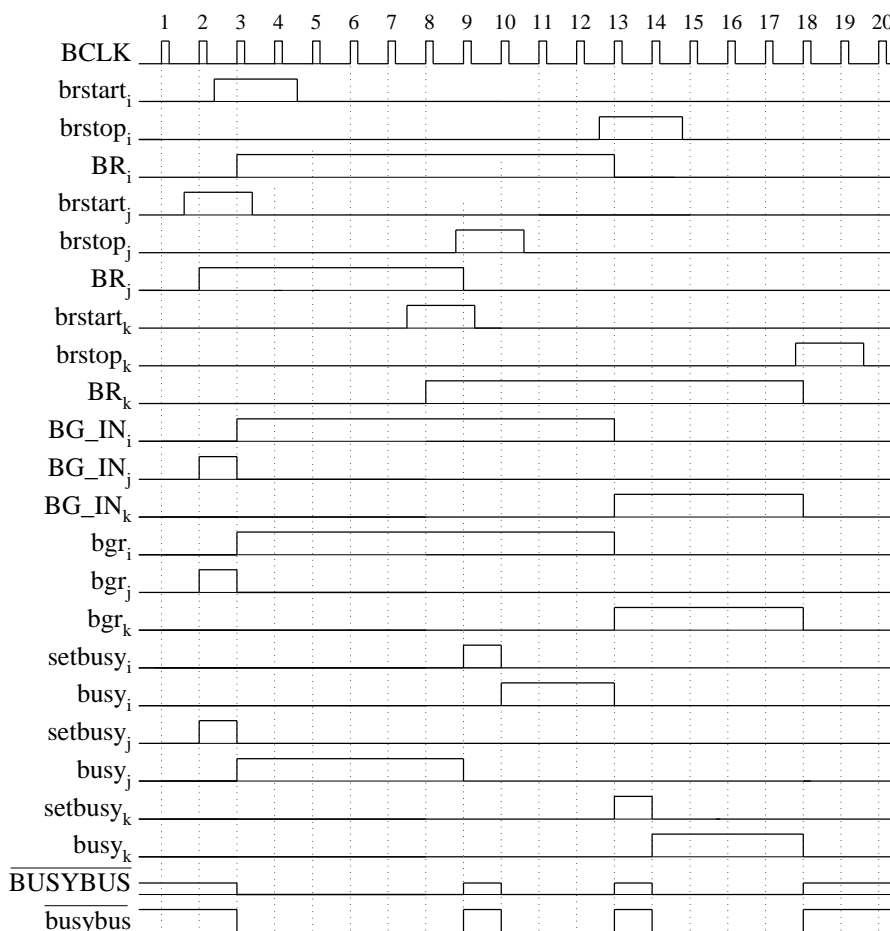


## 1.5 PRILOZI

U ovom odeljku su za neke od tema iz prethodnih odeljaka data dodatna objašnjenja.

### 1.5.1 VREMENSKI OBLICI SIGNALA PRI ARBITRACIJI SA PRAĆENJEM

Paralelna arbitracija sa praćenjem zahteva, koja se realizuje prema šemi sa slike 6 i sa modulima realizovanim kao na slici 10, je ilustrovana vremenskim oblicima signala koji se formiraju u modulima  $i$ ,  $j$  i  $k$  i razmenjuju između modula  $i$ ,  $j$  i  $k$  i arbitratora (slika 35).



Slika 35 Vremenski oblici signala pri paralelnoj arbitraciji sa praćenjem

Signali  $\mathbf{brstart}_i$ ,  $\mathbf{brstart}_j$  i  $\mathbf{brstart}_k$  su interno generisani signali zahteva, signali  $\mathbf{BR}_i$ ,  $\mathbf{BR}_j$  i  $\mathbf{BR}_k$  su registrovani signali zahteva i signali  $\mathbf{brstop}_i$ ,  $\mathbf{brstop}_j$  i  $\mathbf{brstop}_k$  su interno generisani signali završetka korišćenja magistrale u modulima  $i$ ,  $j$  i  $k$ , pri čemu modul  $i$  ima najviši, a modul  $k$  najniži prioritet. Signali  $\mathbf{BG\_IN}_i$ ,  $\mathbf{BG\_IN}_j$  i  $\mathbf{BG\_IN}_k$  su signali dozvola koje arbitrator šalje gazdama  $i$ ,  $j$  i  $k$ , pri čemu se na svaki signal takta  $\mathbf{BCLK}$  jedan od signala dozvola  $\mathbf{BG\_IN}_i$ ,  $\mathbf{BG\_IN}_j$  i  $\mathbf{BG\_IN}_k$  postavlja na vrednost 1, ukoliko postoji bar jedan registrovani zahtev  $\mathbf{BR}_i$ ,  $\mathbf{BR}_j$  i  $\mathbf{BR}_k$  sa vrednošću 1. Vrednost 1 jednog od signala  $\mathbf{bgr}_i$ ,  $\mathbf{bgr}_j$  ili  $\mathbf{bgr}_k$  ukazuje da je odgovarajući modul  $i$ ,  $j$  ili  $k$  modul najvišeg prioriteta koji u vremenskom intervalu između  $i$ -tog i  $(i+1)$ -vog signala takta  $\mathbf{BCLK}$  ima vrednosti 1 registrovanog signala zahteva  $\mathbf{BR}$  i signala dozvole korišćenja magistrale  $\mathbf{BG\_IN}$ . Dati modul će na  $(i+1)$ -vi signal takta  $\mathbf{BCLK}$  postati gazda ukoliko je u vremenskom intervalu između  $i$ -tog i  $(i+1)$ -vog signala takta  $\mathbf{BCLK}$  magistrala slobodna, na šta ukazuje vrednost 1 signala zauzeća magistrale

$\overline{\text{busybus}}$ . Vrednost 1 jednog od signala  $\text{setbusy}_i$ ,  $\text{setbusy}_j$  i  $\text{setbusy}_k$  ukazuje da su se u vremenskom intervalu između  $i$ -tog i  $(i+1)$ -vog signala takta **BCLK** u odgovarajućem modulu  $i$ ,  $j$  ili  $k$  stekli uslovi da dati modul na  $(i+1)$ -vi signal takta **BCLK** postane gazda. Vrednost 1 jednog od signala  $\text{busy}_i$ ,  $\text{busy}_j$  i  $\text{busy}_k$  ukazuje da je odgovarajući modul  $i$ ,  $j$  ili  $k$  na  $(i+1)$ -vi signal takta **BCLK** postao gazda, čime signali zauzeća magistrale  $\overline{\text{BUSYBUS}}$  i  $\text{busybus}$  dobijaju vrednost 0.

Modul koji je postao gazda zatim realizuje ciklus na magistrali i sve to vreme drži signale zauzeća magistrale  $\overline{\text{BUSYBUS}}$  i  $\text{busybus}$  na vrednosti 0. Za to vreme se u preostalim modulima na svaki signal takta **BCLK** vrši postavljanje signala registrovanih zahteva **BR**, pa se saglasno tome i menjaju vrednosti signala **BG\_IN** i **bgr**. Međutim, kako sve to vreme signal  $\text{busybus}$  imma vrednost 0, signali  $\text{setbusy}$  modula ostaju 0, pa ni jedan od tih modula na može da postane gazda. Tek kada gazda završi ciklus na magistrali i generiše signal **brstop**, na  $i$ -ti signal takta **BCLK** njegovi signali **BR** i  $\text{busy}$  postaju 0, signal  $\overline{\text{BUSYBUS}}$  prelazi u stanje visoke impedanse i signal  $\text{busybus}$  postaje 1. Onaj modul u kome u vremenskom intervalu između  $i$ -tog i  $(i+1)$ -vog signala takta **BCLK** signal  $\text{setbusy}$  postane 1, na  $(i+1)$ -vi signal takta **BCLK** postavlja svoj signal  $\text{busy}$  na 1, čime signali  $\overline{\text{BUSYBUS}}$  i  $\text{busybus}$  postaju 0. Time dati modul postaje sledeći gazda.

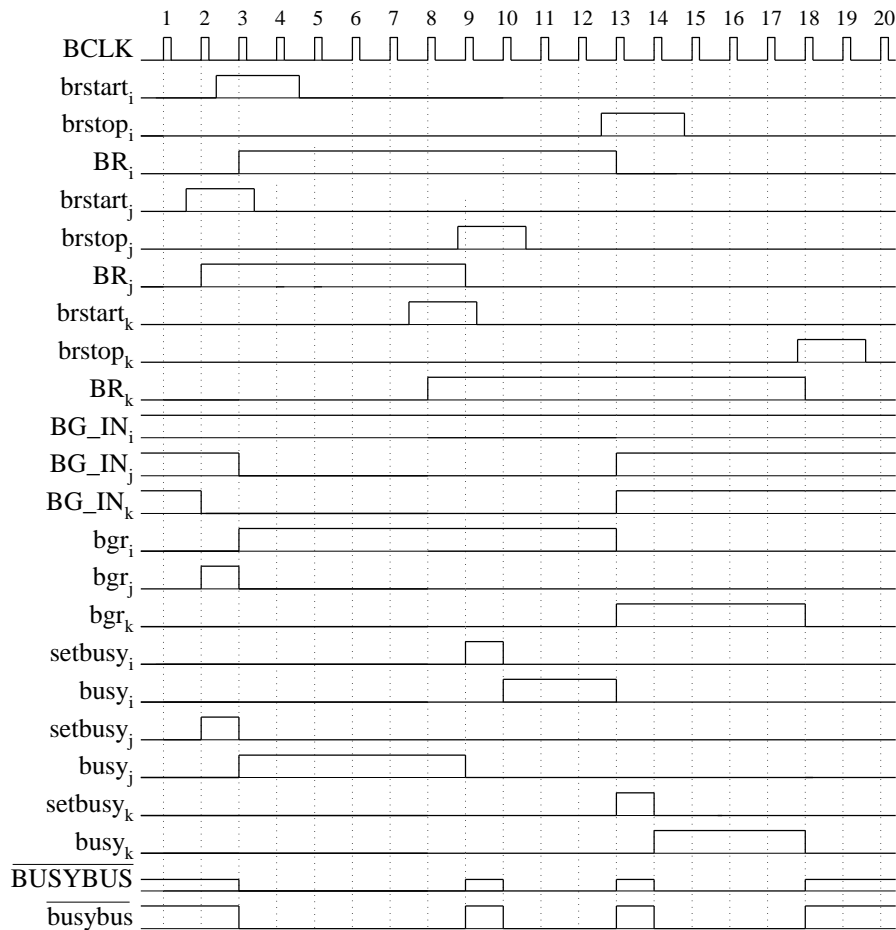
Na vremenskim oblicima signala koji se posmatraju, prva arbitracija koja je bitna se realizuje u vremenskom intervalu između signala takta 2 i 3, i u njoj učestvuje samo gazda  $j$ , jer je, na signal takta 2, samo signal  $\text{BR}_j$  postavljen na vrednost 1. S toga tada najpre signal  $\text{BG\_IN}_j$  pa zatim signal  $\text{bgr}_j$  i nakraju signal  $\text{setbusy}_j$  postaju 1, a kako tada i signal  $\text{busybus}$  ima vrednost 1, i signal  $\text{setbusy}_j$  postaje 1. Na signal takta 3, signal  $\text{busy}_j$  postaje 1, a signal  $\text{busybus}$  0. Treba uočiti da su na signal takta 3 najpre signal  $\text{BR}_i$ , zatim signal  $\text{BG\_IN}_i$  i na kraju signal  $\text{bgr}_i$  postali 1. Međutim, kako je na signal takta 3 signal  $\text{busybus}$  postao 0, signal  $\text{setbusy}_i$  ostaje 0 sve dok modul  $j$  koji je gazda ne završi ciklus na magistrali i postavi signal  $\text{busybus}$  na 1. Na signal takta 8 i signal  $\text{BR}_k$  postaje 1, ali se situacija sa signalima dozvola  $\text{BG\_IN}_i$ ,  $\text{BG\_IN}_j$  i  $\text{BG\_IN}_k$  ne menja.

Sledeća arbitracija koja je bitna se realizuje u vremenskom intervalu između signala takta 9 i 10, jer na signal takta 9 signali  $\text{BR}_j$  i  $\text{busy}_j$  postaju 0, a signal  $\text{busybus}$  1. U arbitraciji učestvuju moduli  $i$  i  $k$ , jer su tada aktivni signali  $\text{BR}_i$  i  $\text{BR}_k$ . S obzirom da je modul  $i$  višeg prioriteta od modula  $k$  signali  $\text{BG\_IN}_i$  i  $\text{bgr}_i$  su i dalje 1. Međutim, kako sada signal  $\text{busybus}$  ima vrednost 1, to i signal  $\text{setbusy}_i$  postaje 1. Stoga na signal takta 10 signal  $\text{busy}_i$  postaje 1, a signal  $\text{busybus}$  0.

Zadnja arbitracija koja je bitna se realizuje u vremenskom intervalu između signala takta 13 i 14, jer na signal takta 13 signali  $\text{BR}_i$  i  $\text{busy}_i$  postaju 0, a signal  $\text{busybus}$  1. U arbitraciji učestvuje samo modul  $k$ , jer je tada aktivan samo signal  $\text{BR}_k$ . S toga tada najpre signal  $\text{BG\_IN}_k$  pa zatim signal  $\text{bgr}_k$  i nakraju signal  $\text{setbusy}_k$  postaju 1, a kako tada i signal  $\text{busybus}$  ima vrednost 1, i signal  $\text{setbusy}_k$  postaje 1. Na signal takta 14, signal  $\text{busy}_k$  postaje 1, a signal  $\text{busybus}$  0.

Prva sledeća arbitracija bi bila bitna se realizuje posle signala takta 18, jer tada signali  $\mathbf{BR}_k$  i  $\mathbf{busy}_k$  postaju 0, a signal  $\mathbf{busybus}$  1.

Serijska arbitracija sa praćenjem zahteva, koja se realizuje prema šemi sa slike 7 i sa modulima realizovanim kao na slici 10, je ilustrovana vremenskim oblicima signala koji se formiraju u modulima  $i, j$  i  $k$  i razmenjuju između modula  $i, j$  i  $k$  (slika 36).



Slika 36 Vremenski oblici signala pri paralelnoj arbitraciji sa praćenjem

Vremenski oblici signala u slučaju serijske i paralelne arbitracije se razlikuju samo za signale  $\mathbf{BG\_IN}_i$ ,  $\mathbf{BG\_IN}_j$  i  $\mathbf{BG\_IN}_k$ . Signal  $\mathbf{BG\_IN}_i$  je sve vreme 1 jer nema zahteva od modula višeg prioriteta i modul  $i$  prosleđivanjem uvek dobija dozvolu. Do signala takta 2 i signali  $\mathbf{BG\_IN}_j$  i  $\mathbf{BG\_IN}_k$  su 1 jer do tada i signali registrovanih zahteva  $\mathbf{BR}_i$ ,  $\mathbf{BR}_j$  i  $\mathbf{BR}_k$  imaju vrednost 0. Pošto na signal takta 2 signal  $\mathbf{BR}_j$  postaje 1 modul  $j$  ne prosleđuje dalje dozvolu, pa signal  $\mathbf{BG\_IN}_k$  ima vrednost 0. Međutim na signal takta 3 signal  $\mathbf{BR}_i$  postaje 1, pa modul  $i$  ne prosleđuje dalje dozvolu modulu  $j$ , a modul  $j$  ne prosleđuje dozvolu modulu  $k$ . Stoga od signala takta 3 do signala takta 13 kada signal  $\mathbf{BR}_i$  postaje 0, signali  $\mathbf{BG\_IN}_j$  i  $\mathbf{BG\_IN}_k$  ima vrednost 0. Međutim na signal takta 13 signal  $\mathbf{BR}_i$  postaje 0, pa modul  $i$  prosleđuje dalje dozvolu modulu  $j$ . Tada i  $\mathbf{BR}_j$  ima vrednost 0, pa modul  $j$  prosleđuje dozvolu modulu  $k$ . Zbog toga od signala takta 13 i signali  $\mathbf{BR}_j$  i  $\mathbf{BR}_k$  imaju vrednost 0.

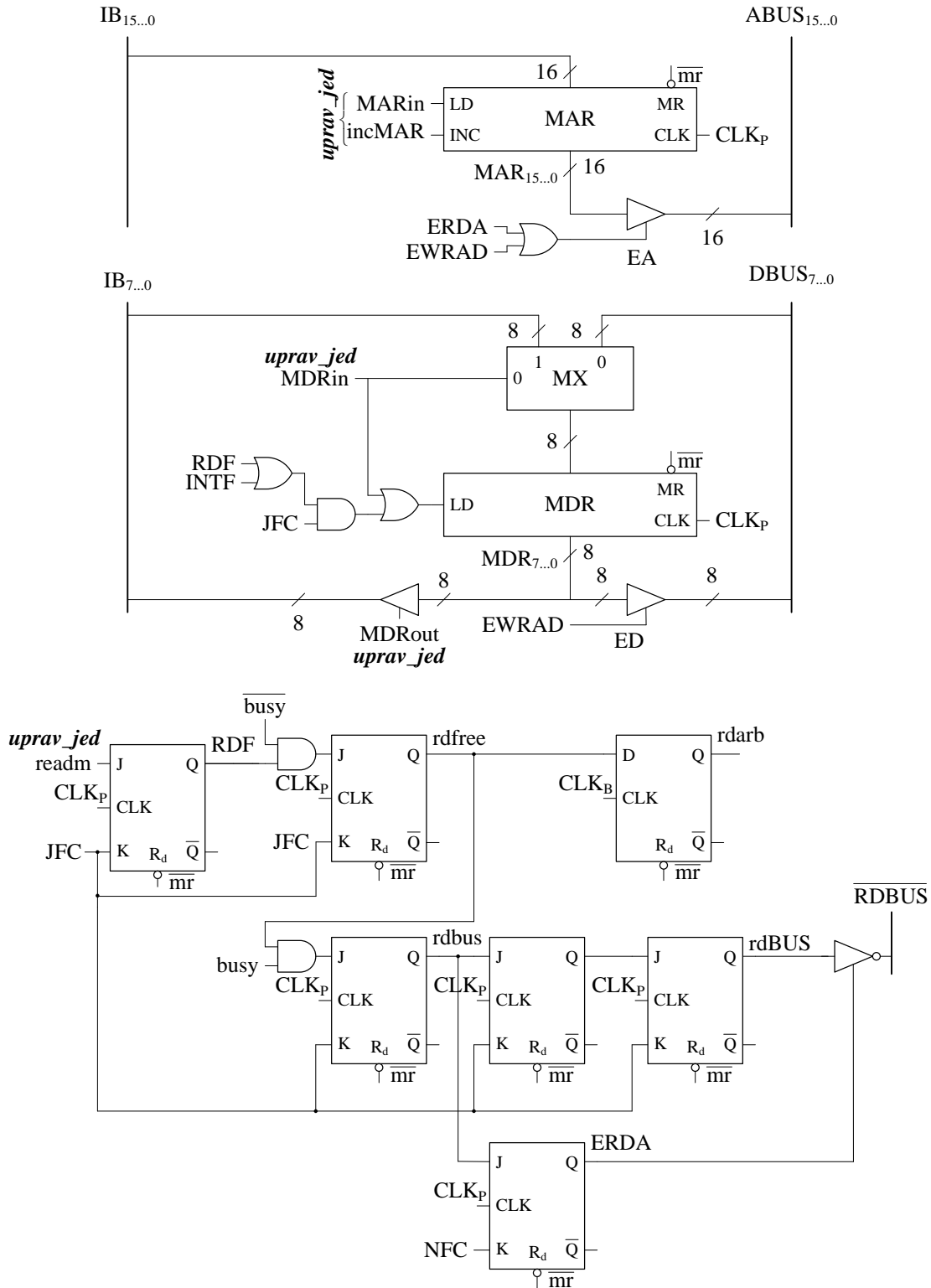
### 1.5.2 ASINHRONA MAGISTRALA SA ATOMSKIM CIKLUSIMA

U ovom odeljku se daje jedna moguća realizacija dela procesora za povezivanje procesora kao gazde i dela memorije za povezivanje memorije kao sluge na asinhronu magistralu sa atomskim ciklusima. U slučaju magistrale sa atomskim ciklusima postoje ciklus čitanja, ciklus

upisa i ciklus prihvatanja broja ulaza. Prilikom realizacije ovih ciklusa procesor kao gazda i memorija kao sluga razmenju signale čiji su vremenski oblivi dati na slikama 12, 13 i 14.

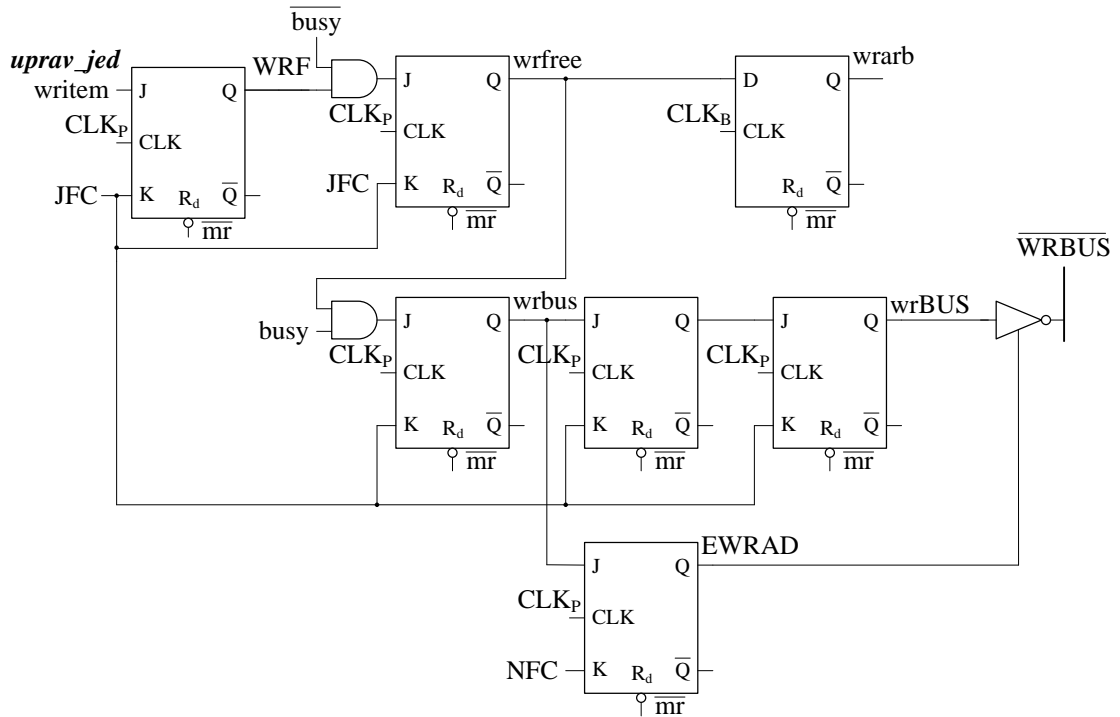
### 1.5.2.1 PROCESOR

Varijanta sa dva koraka. U *i*-tom koraku se startuje ciklus čitanja, upisa ili prihvatanje broja ulaza. *i* ostaje samo jedna perioda signala takta posebnim korakom u kome se čeka završetak ciklusa brojača koraka CNT.

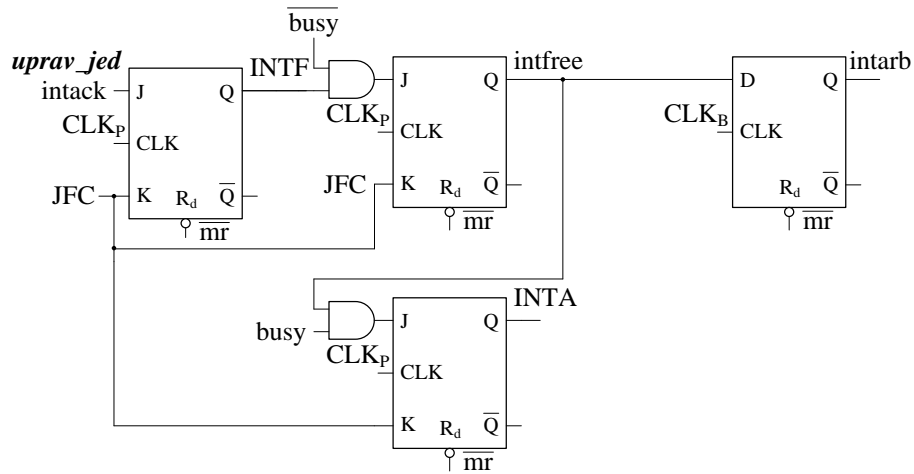


Slika Startovanje ciklusa čitanja

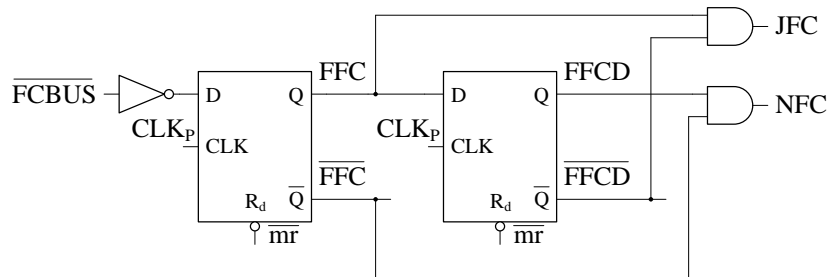




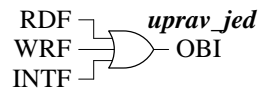
Slika Startovanje ciklusa upisa



Slika **Error! Reference source not found.** Startovanje ciklusa prihvatanje broja ulaza



Slika **Error! Reference source not found.** Završetak ciklusa



Slika **Error! Reference source not found.**b Ciklus u toku

Upravljačka jedinica cikluse čitanja, upisa i prihvatanja broja ulaza realizuje u sledećim sekvencama.

Ciklus čitanja.

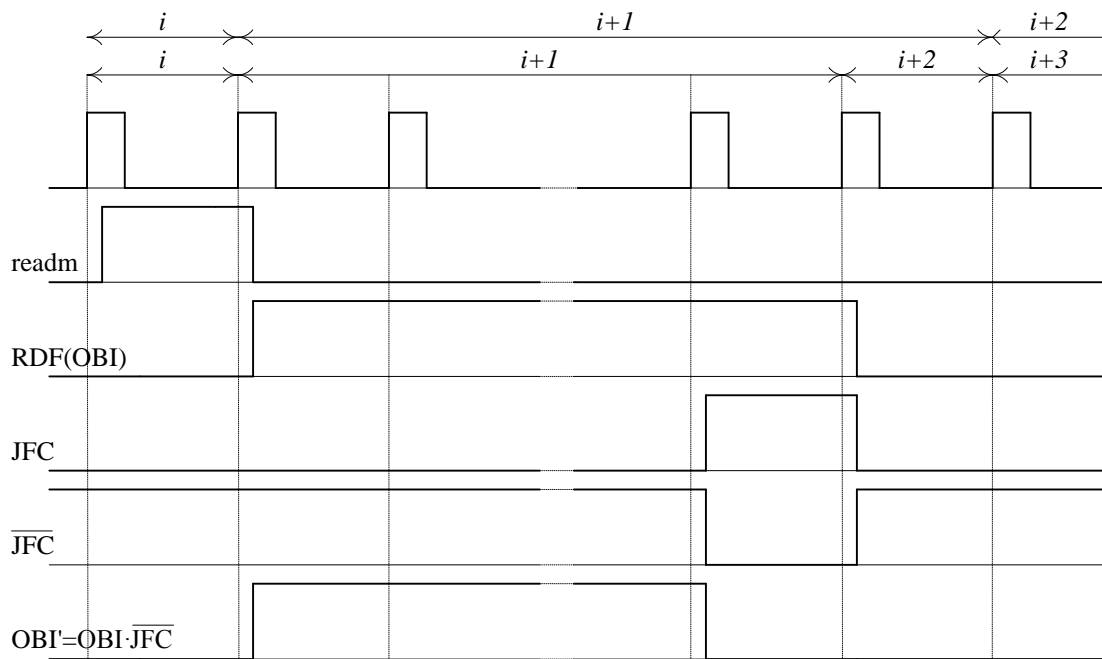
...  
 step<sub>i-1</sub> Xout, MARin,  
 step<sub>i</sub> readm,  
 step<sub>i+1</sub> *if* OBI *goto* step<sub>i+1</sub>,  
 step<sub>i+2</sub> MDRout, Yin,  
 ...

Ciklus upisa.

...  
 step<sub>i-2</sub> Xout, MARin,  
 step<sub>i-1</sub> Yout, MDRin,  
 step<sub>i</sub> writem,  
 step<sub>i+1</sub> *if* OBI *goto* step<sub>i+1</sub>,  
 step<sub>i+2</sub> ... ,  
 ...

Ciklus prihvatanje broja ulaza.

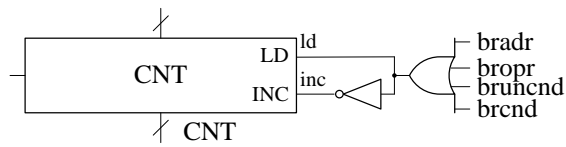
...  
 step<sub>i</sub> intack  
 step<sub>i+1</sub> *if* OBI *goto* step<sub>i+1</sub>,  
 step<sub>i+2</sub> MDRout, Yin,  
 ...



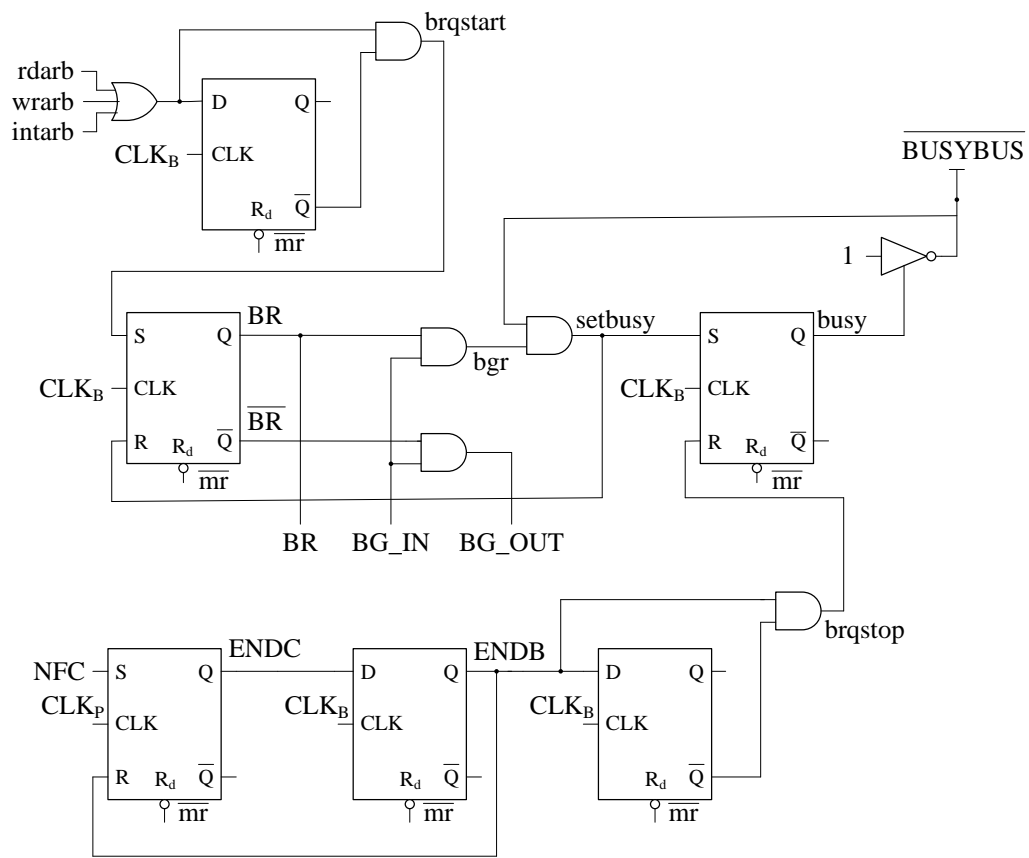
Slika Procesor sa OBI

$$\mathbf{brend} = \dots + \mathbf{brOBI} \cdot \mathbf{OBI} + \dots$$

$$\mathbf{brOBI} = \dots + T_{i+1} + \dots$$



Slika Brojač koraka CNT



Slika **Error! Reference source not found.**.c Arbitracija

Varijanta sa zaustavljanjem brojača koraka CNT.

Upravljačka jedinica cikluse čitanja, upisa i prihvatanja broja ulaza realizuje u sledećim sekvencama.

Ciklus čitanja.

...

step<sub>i-1</sub> Xout, MARin,

step<sub>i</sub> readm,

step<sub>i+1</sub> MDRout, Yin,

...

Ciklus upisa.

...

step<sub>i-2</sub> Xout, MARin,

step<sub>i-1</sub> Yout, MDRin,

step<sub>i</sub> writem,

step<sub>i+1</sub> ... ,

...

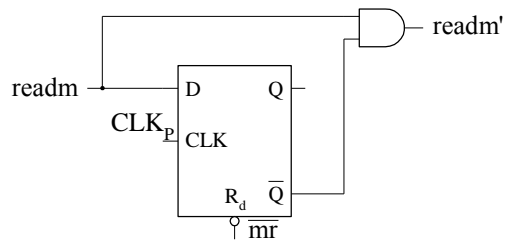
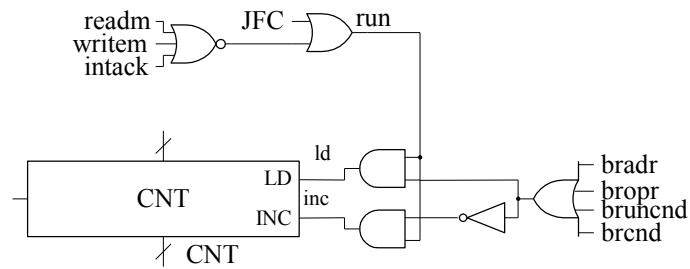
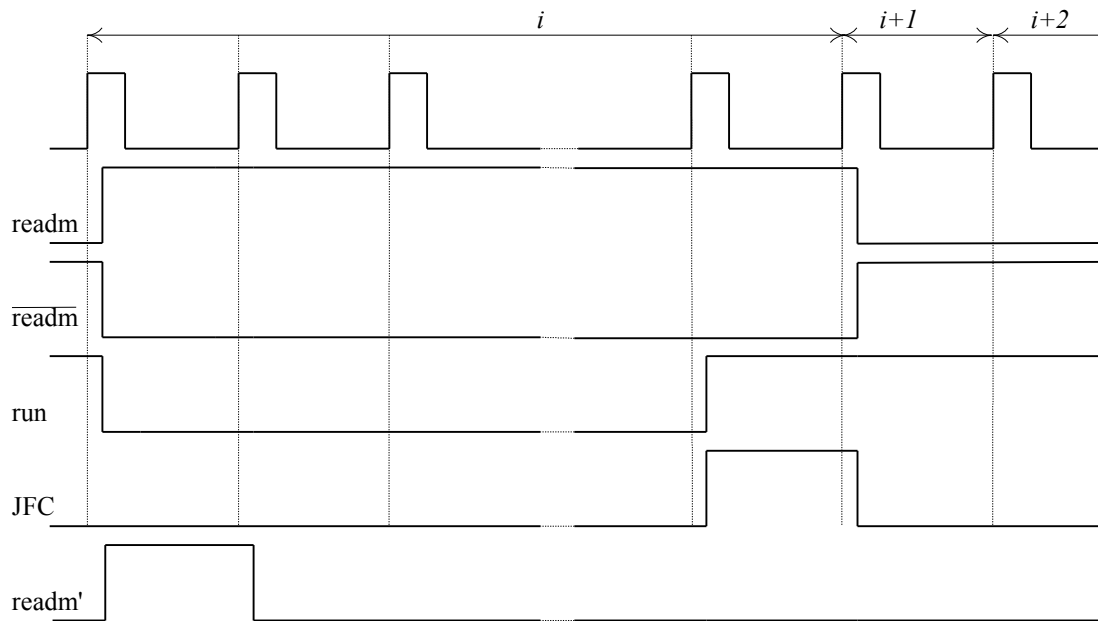
Ciklus prihvatanje broja ulaza.

...

step<sub>i</sub> intack

step<sub>i+1</sub> MDRout, Yin,

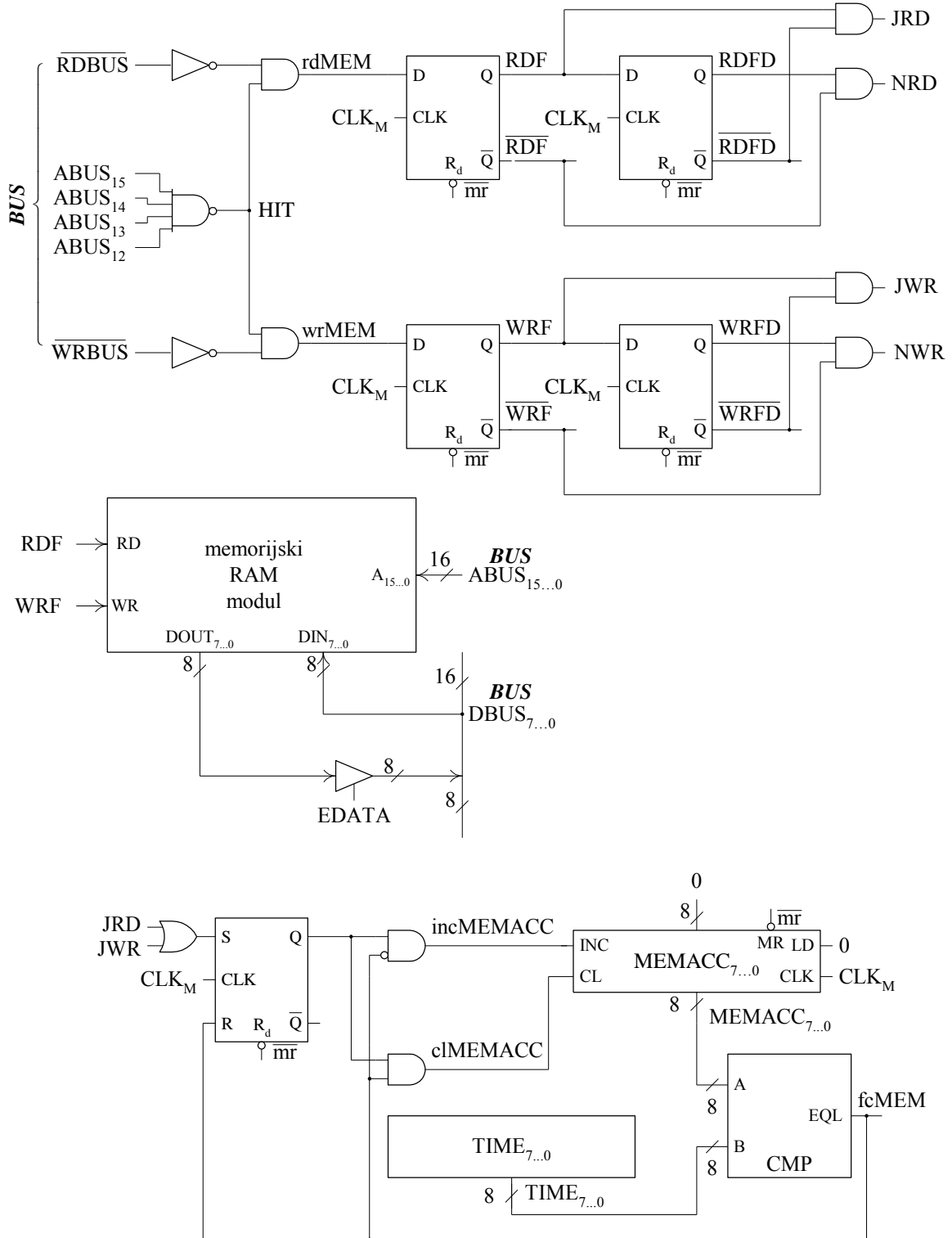
...

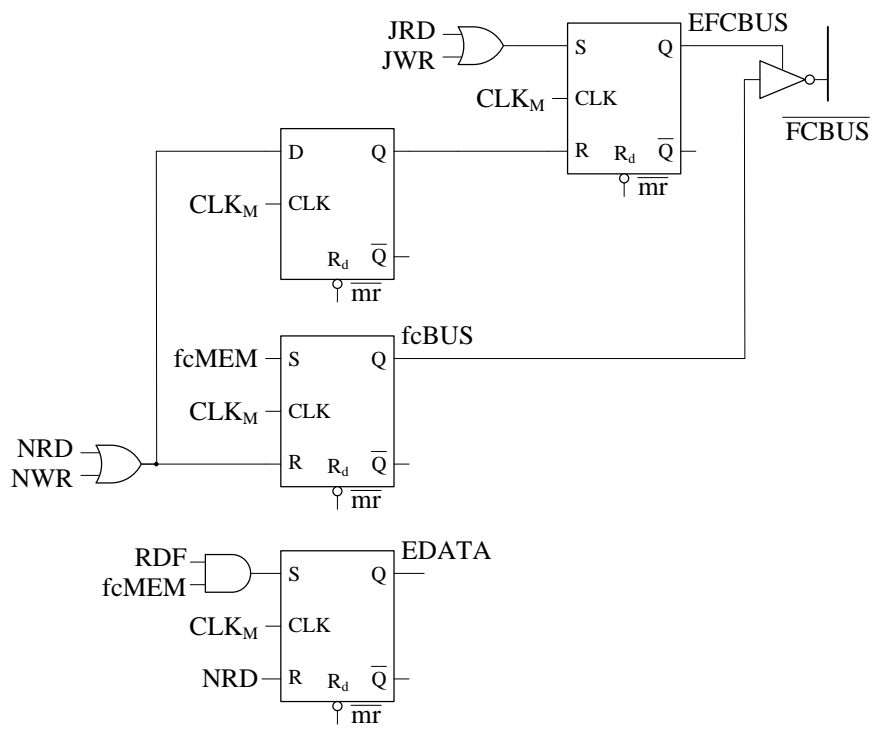


Slika **Error! Reference source not found.**d Procesor sa run

### 1.5.2.2 MEMORIJA

U ovom odeljku se razmatra realizacija ciklusa čitanja i upisa u memoriji kao slugi na asinhronoj magistrali.





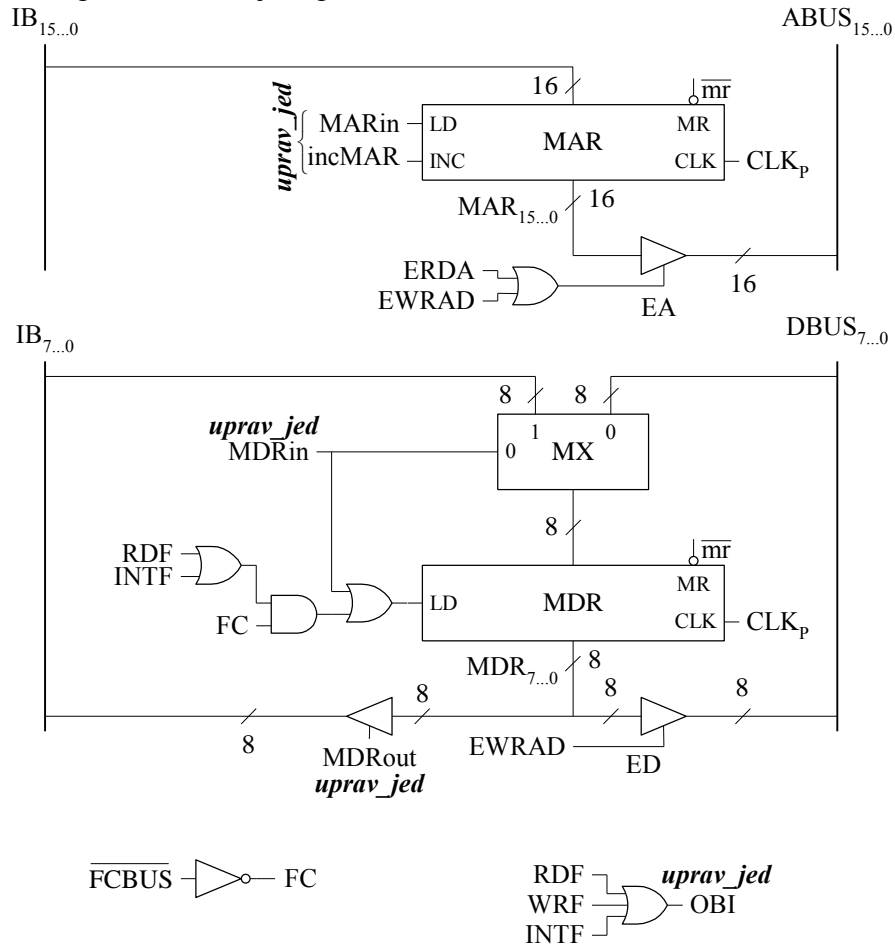


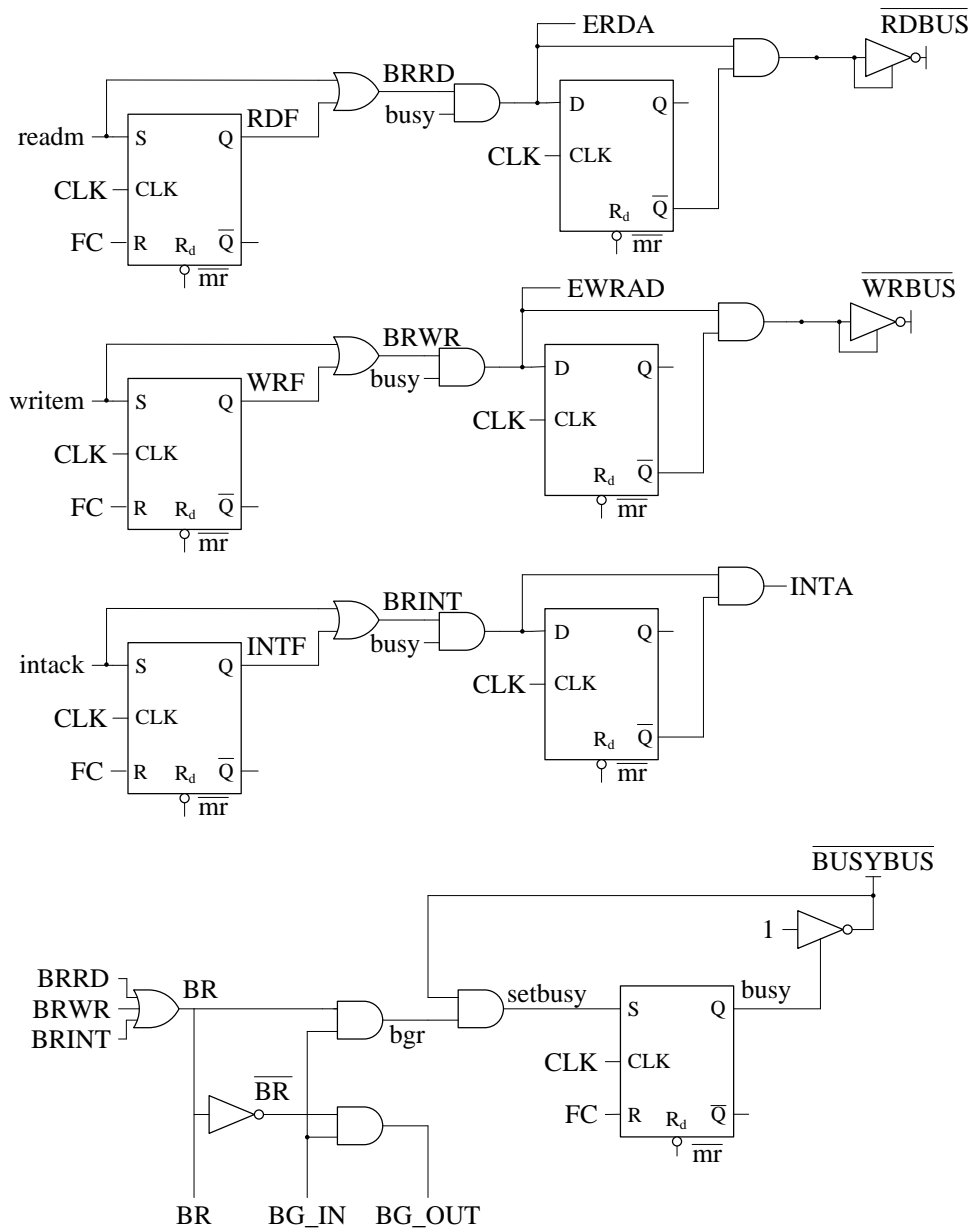
### 1.5.3 SINHRONA MAGISTRALA SA ATOMSKIM CIKLUSIMA

U ovom odeljku se daje jedna moguća realizacija dela procesora za povezivanje procesora kao gazde i dela memorije za povezivanje memorije kao sluge na sinhronu magistralu sa atomskim ciklusima.

#### 1.5.3.1 PROCESOR

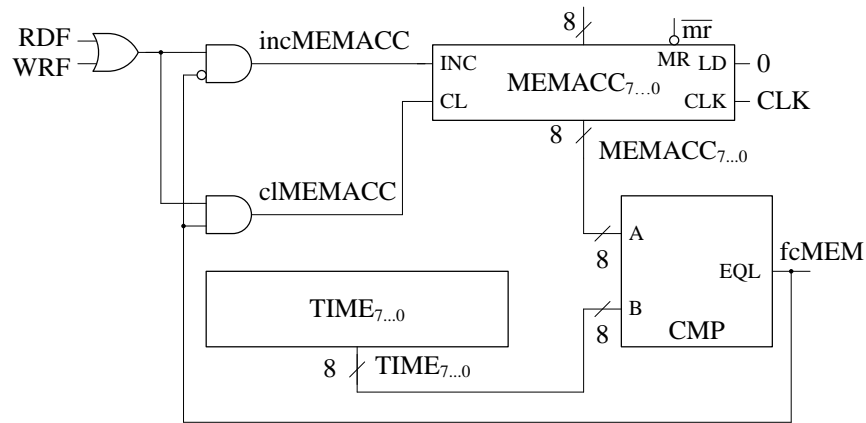
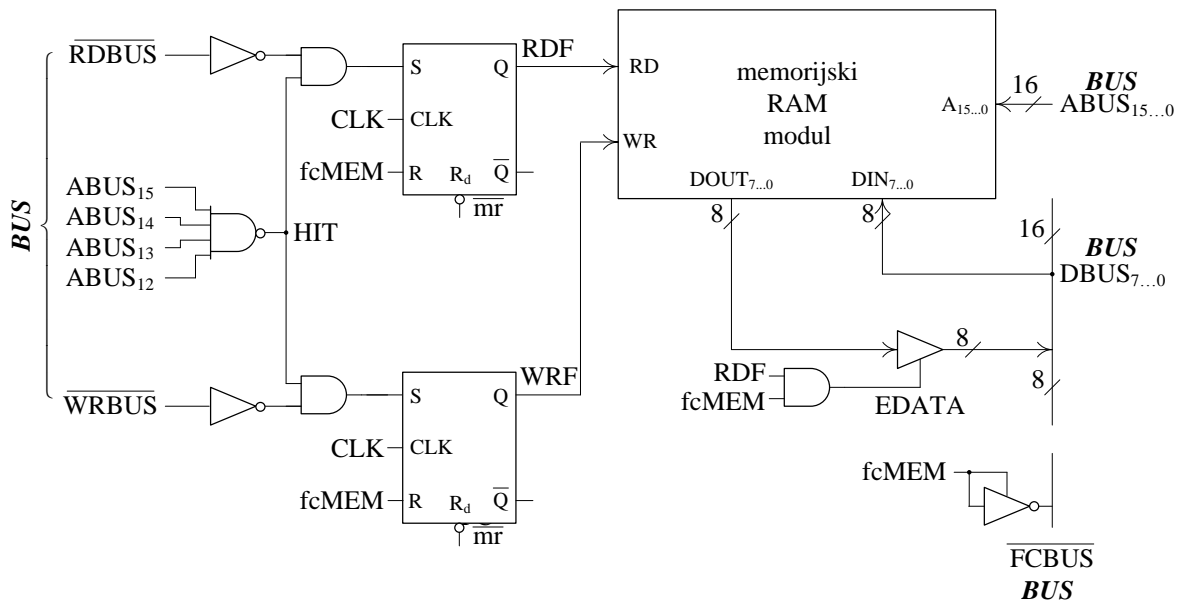
U ovom odeljku se daje jedna moguća ciklusima čitanja, upisa i prihvatanja broja ulaza u precesoru kao slugi na sinhronoj magistrali.





### 1.5.3.2 MEMORIJA

U ovom odeljku se daje jedna moguća ciklusima čitanja i upisa u memoriji kao slugi na sinhronoj magistrali.



#### 1.5.4 MAGISTRALE SA PODELJENIM CIKLUSIMA

U slučaju magistrale sa podjeljenim ciklusima postoje ciklus čitanja, ciklus upisa i ciklus prihvatanja.

