

1 RAČUNARSKI SISTEM

Sistem sa preklapljenim memorijskim modulima sastoji se od sistemske magistrale, arbitratora, memorijskih modula i uređaja za pristup memoriji. U ovoj glavi daju se struktura edukacionog računarskog sistema, ciklusi na sistemskoj magistrali i postupak arbitracije zahteva za korišćenje sistemske magistrale, kao i osnovne karakteristike modula edukacionog računarskog sistema.

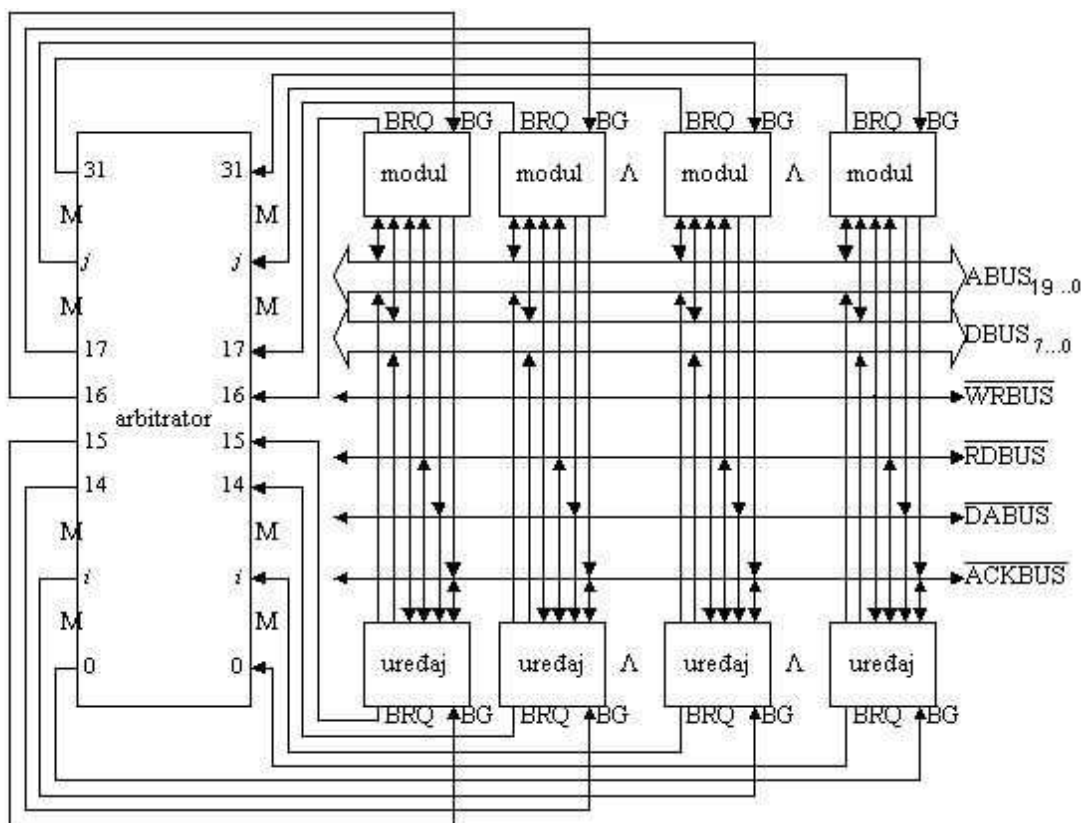
1.1 Struktura edukacionog računarskog sistema

Edukacioni računarski sistem (slika 3.1) sastoji se od sistemske magistrale, arbitratora, maksimalno 16 memorijskih modula i maksimalno 16 uređaja za pristup memoriji.

Uređaji za pristup memoriji mogu biti uređaji sa pojedinačnim pristupom memoriji i uređaji sa blokovskim pristupom memoriji. Moguća je bilo koja kombinacija uređaja i sa pojedinačnim i sa blokovskim pristupom memoriji.

Kapacitet jednog memorijskog modula je 2^{16} bajta a adresibilna jedinica je bajt. Način raspoređivanja adresa po memorijskim modulima se određuje posebnim mikroprekidačima sa kojih se signali razvode po svim modulima. Postoje i posebni mikroprekidači na samim modulima čijim se postavljanjem dodeljuje broj modula u opsegu od 0 do 15.

Svi moduli edukacionog računarskog sistema rade sinhrono na isti signal takta.



Slika 3.1 Struktura edukacionog računarskog sistema

1.2 Sistemska magistrala

Uređaji i memorijski moduli su povezani sistemskom magistralom. Magistralu čini 20 adresnih linija **ABUS_{19...0}**, 8 linija podataka **DBUS_{7...0}**, i četiri upravljačke linije **WRBUS**, **RDBUS**, **DABUS** i **ACKBUS**. Na magistrali se realizuju ciklusi: *zahtev za upis bajta podatka*, *zahtev za čitanje bajta podatka* i *vraćanje bajta podatka*. Uređaj ili memorijski modul koji inicira ciklus na magistrali naziva se master, a memorijski modul ili uređaj sa kojim se realizuje ciklus naziva se slave.

Arbitrator određuje koji od uređaja ili memorijskih modula može da bude sledeći master magistrale.

Kod ciklusa *zahtev za upis bajta podatka* master na adresne linije magistrale **ABUS_{19...0}** postavlja adresu memorijske lokacije, na linije podataka magistrale **DBUS_{7...0}** postavlja bajt podatka koji treba upisati i generiše aktivnu vrednost upravljačkog signala **WRBUS**.

Kod ciklusa *zahtev za čitanje bajta podatka* master na adresne linije magistrale **ABUS_{19...0}** postavlja adresu memorijske lokacije, na linije podataka magistrale **DBUS_{7...0}** postavlja svoj identifikator i generiše aktivnu vrednost upravljačkog signala **RDBUS**.

Kod ciklusa *vraćanje bajta podatka* master na adresne linije magistrale **ABUS_{7...0}** postavlja identifikator uređaja kome se vraća podatak, na linije podataka magistrale **DBUS_{7...0}** postavlja očitani bajt podatka i generiše aktivnu vrednost upravljačkog signala **DABUS**.

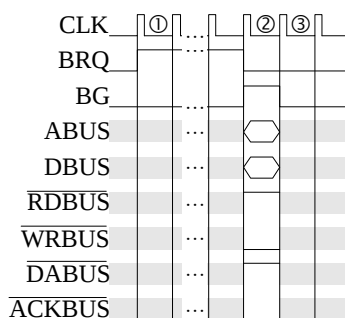
U sva tri slučaja slave šalje master-u aktivnu vrednost upravljačkog signala **ACKBUS** ukoliko je slobodan da realizuje dati ciklus, odnosno neaktivnu vrednost ukoliko nije slobodan da realizuje dati ciklus. U slučaju kada je slobodan da realizuje dati ciklus, *slave* upisuje sadržaje sa adresnih linija i linija podataka magistrale u svoje prihvatne registre. S obzirom da se ciklus na magistrali svodi na prenos odgovarajućih sadržaja iz registara master-a u registre slave-a, trajanje sva tri ciklusa je fiksno i isto. Stoga je usvojena sinhrona realizacija upravljanja na magistrali uz trajanje ciklusa od jedne periode signala takta.

Svi uređaji i memorijski moduli povezani su sa arbitratorom magistrale linijom **BRQ** preko koje upućuju zahtev i linijom **BG** preko koje dobijaju dozvolu za izlazak na magistralu. Signali zahteva i dozvola se, takođe, generišu sinhrono sa zajedničkim signalom takta računarskog sistema. Signal dozvole **BG** se daje samo za jednu periodu signala takta pa uređaj ili memorijski modul koji u nekoj periodu signala takta dobije signal dozvole **BG**, u datoj periodu realizuje jedan od tri moguća ciklusa na magistrali i ukida signal zahteva **BRQ**.

1.2.1 Ciklusi na sistemskoj magistrali

1.2.1.1 Ciklus slanje zahteva za upis bajta podatka

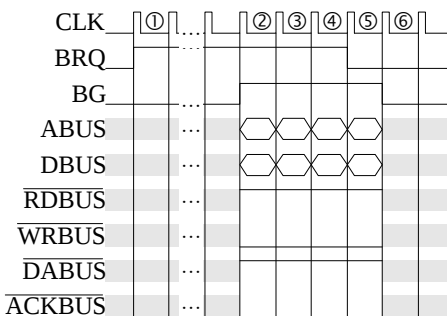
Ciklus *zahtev za upis bajta podatka* koristi uređaj sa pojedinačnim pristupom kada treba da izvrši operaciju upisa bajta podatka u memoriju. Tada on postaje master na magistrali, a slave je memorijski modul koji je adresiran. Vremenski oblici signala koje tom prilikom razmenjuje master sa arbitratorom i slave-om su prikazani na slici 3.2.



Slika 3.2 Ciklus zahtev za upis bajta podatka za slučaj uređaja sa pojedinačnim pristupom memoriji

Uređaj sa pojedinačnim pristupom koji želi da postane master na magistrali postavlja signal zahteva **BRQ** na aktivnu vrednost u periodu ①. Arbitrator posle određenog broja preioda signala takta, koji zavisi od broja uređaja koji žele da izađu na magistralu i njihovih međusobnih prioriteta, u periodu ② daje datom uređaju dozvolu postavljanjem signala **BG** na aktivnu vrednost. Uređaj koji je dobijanjem dozvole postao master postavlja adresu memorijske lokacije u koju treba upisati bajt podatka na adresne linije magistrale **ABUS**_{19...0}, bajt podatka na linije podataka magistrale **DBUS**_{7...0} i generiše aktivnu vrednost upravljačkog signala **WRBUS**. U toku periode ② memorijski moduli dekoduju sadržaj sa adresnih linija magistrale i jedan od modula tu adresu prepoznaje kao svoju. Uzeto je da je dati memorijski modul slobodan, pa će ciklus zahtev za upis bajta podatka biti uspešno realizovan. Stoga memorijski modul kao slave postavlja signal potvrde **ACKBUS** na aktivnu vrednost, što je indikacija master-u da će njegov zahtev za upis bajta podatka biti uspešno realizovan, pa on ukida signal zahteva **BRQ**. Na signal takta kojim se prelazi iz periode ② u periodu ③ memorijski modul kao slave upisuje sadržaj sa adresnih linija i linija podataka magistrale u svoje prihvatne registre. Na taj isti signal takta arbitrator master-u iz periode ② ukida aktivnu vrednost signala dozvole **BG** i, ukoliko je u periodu ② bilo zahteva za korišćenje magistrale od drugih uređaja ili memorijskih modula, jednom od njih daje dozvolu postavljanjem njegovog signala **BG** na aktivnu vrednost u periodu ③.

Ciklus zahtev za upis bajta podatka koristi i uređaj sa blokovskim pristupom kada treba da izvrši operaciju *upis bloka podataka u memoriju*. Tada on postaje master na magistrali, a slave-ovi u ovim ciklusima su memorijski moduli koji su adresirani. Vremenski oblici signala koje tom prilikom razmenjuje master sa arbitratorom i slave-ovima prikazani su na slici 3.3.

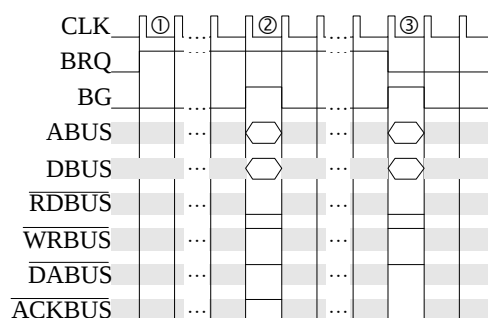


Slika 3.3 Ciklus zahtev za upis bajta podatka za slučaj uređaja sa blokovskim pristupom memoriji

Uzeto je da uređaj sa blokovskim pristupom memoriji ima blok podataka veličine četiri bajta. Za svaki bajt bloka mora da se realizuje, i to uspešno, poseban ciklus na magistrali zahtev za upis bajta podatka. Uređaj postavlja signal zahteva **BRQ** u periodi ①, a dozvole dobija u periodama ②, ③, ④ i ⑤. U svakoj od ove četiri periode realizuje se, i to uspešno, poseban ciklus zahtev za upis bajta podatka na identičan način kao i u slučaju uređaja sa pojedinačnim pristupom. U periodi ⑤ uređaj sa blokovskim pristupom memoriji ukida signal zahteva **BRQ** jer aktivnom vrednošću signala potvrde **ACKBUS** dobija potvrdu da je uspešno prihvaćen zahtev za upis i poslednjeg, četvrtog bajta bloka.

1.2.1.2 Ciklus slanje zahteva za čitanje bajta podatka

Ciklus zahtev za čitanje bajta podatka koristi uređaj sa pojedinačnim pristupom memoriji kada treba da izvrši operaciju čitanje bajta podatka iz memorije. Tada on postaje master na magistrali, a slave je memorijski modul koji je adresiran. Vremenski oblici signala koje tom prilikom razmenjuje master sa arbitratorom i slave-om prikazani su na slici 3.4.



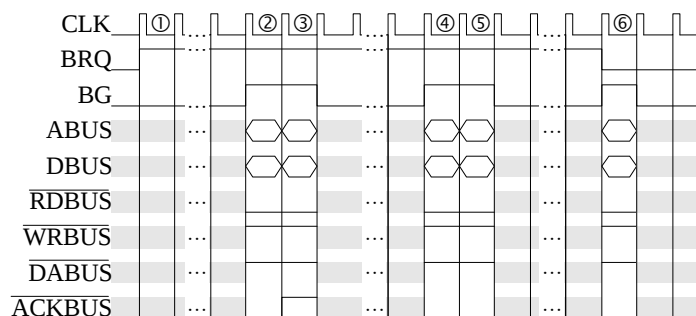
Slika 3.4 Ciklus slanje zahteva za čitanje bajta podatka u slučaju uređaja sa pojedinačnim pristupom memoriji

Uređaj sa pojedinačnim pristupom memoriji postavlja signal zahteva **BRQ** u periodi ① a ukida ga tek u periodi ③ kada uspešno realizuje ciklus na magistrali slanje zahteva za čitanje bajta podatka. Uređaj dobija signal dozvole **BG** u periodama ② i ③. Tada on kao master postavlja adresu memorijske lokacije sa koje treba očitati bajt podatka na adresne linije magistrale **ABUS_{23...0}**, svoj identifikator na linije podataka magistrale **DBUS_{7...0}** i generiše aktivnu vrednost upravljačkog signala **RDBUS**. Uzeto je da adresirani memorijski modul nije slobodan u periodi ②, da ne prihvata zahtev za čitanje bajta podatka i da stoga šalje uređaju neaktivnu vrednost upravljačkog signala potvrde **ACKBUS**. Zbog toga uređaj, iako u periodi ② dobija signal potvrde **BG**, ne ukida signal zahteva **BRQ**. S druge strane uzeto je da je u periodi ③ adresirani memorijski modul slobodan, da prihvata zahtev za čitanje bajta podatka i da, stoga, uređaju šalje aktivnu vrednost upravljačkog signala potvrde **ACKBUS**. Zbog toga uređaj u periodi ③ ukida signal zahteva **BRQ**.

Uređaj sa pojedinačnim pristupom memoriji ciklusom na magistrali zahtev za čitanje bajta podatka je realizovao samo prvi deo operacije čitanje bajta podatka. Drugi deo ove operacije se realizuje ciklusom na magistrali vraćanje bajta podatka.

Ciklus slanje zahteva za čitanje bajta podatka koristi i uređaj sa blokovskim pristupom memoriji kada treba da izvrši operaciju čitanja bloka podataka. Tada postaje master, a

slave-ovi u ovim ciklusima su memorijski moduli koji su adresirani. Vremenski oblici signala koje tom prilikom razmenjuje master sa arbitratorom i slave-ovima su prikazani na slici 3.5.



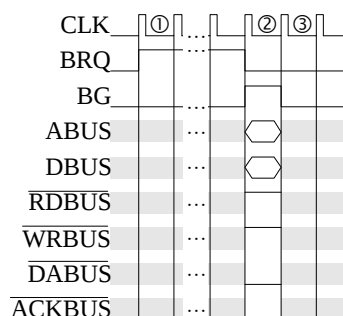
Slika 3.5 Ciklus slanje zahteva za čitanje bajta podatka u slučaju uređaja sa blokovskim pristupom memoriji

Uzeto je da uređaj sa blokovskim pristupom memoriji treba da očita blok podataka veličine četiri bajta. Za svaki bajt bloka mora da se realizuje, i to uspešno, poseban ciklus na magistrali zahtev za čitanje bajta podatka. Uređaj postavlja signal zahteva **BRQ** u periodu ①, a signal dozvole **BG** dobija u periodama ②, ③, ④, ⑤ i ⑥. U svakoj od ovih perioda realizuje se poseban ciklus zahtev za čitanje bajta podatka na identičan način kao i u slučaju uređaja sa pojedinačnim pristupom. U periodama ②, ④, ⑤ i ⑥ adresirani memorijski moduli su slobodni, ciklusi se uspešno realizuju, pa tada moduli šalju uređaju aktivne vrednosti upravljačkog signala potvrde **ACKBUS**. U periodu ③ adresirani memorijski modul je zauzet, ciklus se ne realizuje uspešno, pa tada memorijski modul šalje uređaju neaktivnu vrednost upravljačkog signala potvrde **ACKBUS**. U periodu ⑥ uređaj sa blokovskim pristupom memoriji ukida signal zahteva **BRQ** jer, aktivnom vrednošću upravljačkog signala potvrde **ACKBUS**, dobija indicaciju da će uspešno biti prihvaćen zahtev za čitanje i poslednjeg, četvrtog bajta bloka.

Uređaj sa blokovskim pristupom memoriji ciklusima na magistrali zahtev za čitanje bajta podatka je realizovao samo prvi deo operacije čitanje bajta podatka. Drugi deo se realizuje ciklusima na magistrali vraćanje bajta podatka, tako što se svaki očitani bajt podatka vraća posebnim ciklusom vraćanje bajta podatka.

1.2.1.3 Ciklus vraćanje bajta podatka

Ciklus vraćanje bajta podatka iz memorije koristi memorijski modul kada izvrši operaciju čitanja iz memorije. Memorijski modul je u ovom slučaju master ciklusa, a slave je uređaj koji je ciklusom na magistrali zahtev za čitanje bajta podatka inicirao operaciju čitanja iz datog memorijskog modula. Vremenski oblici signala koje tom prilikom razmenjuje master sa arbitratorom i slave-om su prikazani na slici 3.6.

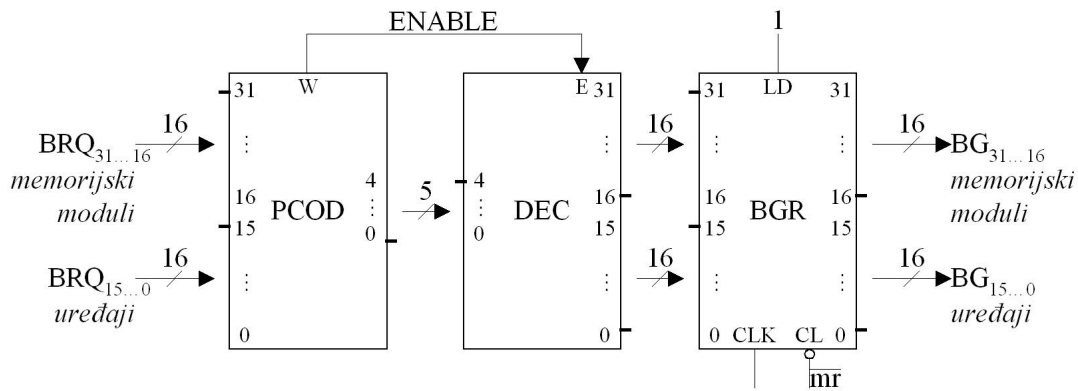


Slika 3.6 Ciklus vraćanje bajta podatka

Memorijski modul postavlja signal zahteva za korišćenje magistrale **BRQ** u periodu ①, a arbitrator mu daje dozvolu signalom **BG** u periodu ②. Memorijski modul tada kao master postavlja na adresne linije magistrale **ABUS**_{7...0} identifikator uređaja kome se vraća očitani bajt podatka. Taj identifikator uređaja memorijski modul je dobio od samog uređaja u okviru ciklusa zahtev za čitanje bajta podatka. Memorijski modul postavlja očitani bajt podatka na linije podataka magistrale **DBUS**_{7...0} i postavlja na aktivnu vrednost upravljački signal **DABUS**. Uređaj koji utvrdi da se na adresnim linijama nalazi vrednost njegovog identifikatora šalje modulu aktivnu vrednost upravljačkog signala **ACKBUS** i na signal takta kojim se prelazi iz periode ② u periodu ③ upisuje bajt podatka u svoj prihvatni registar. Aktivna vrednost upravljačkog signala potvrde **ACKBUS** je indikacija memorijskom modulu da će ciklus vraćanje bajta podatka biti uspešno realizovan pa on u periodu ② ukida signal zahteva **BRQ**.

1.2.2 Arbitracija zahteva za korišćenje sistemske magistrale

U računarskom sistemu se koristi paralelni arbitrator (slika 3.7) koji rešava zahteve za pristup magistrali od strane uređaja i memorijskih modula. Pošto je predviđeno da računar može da ima do 16 memorijskih modula i do 16 uređaja sa pojedinačnim ili blokovskim pristupom koristi se 32-bitni arbitrator. Po linijama **BRQ**_{31...16} dolaze u arbitrator signali zahteva od strane memorijskih modula, a po linijama **BRQ**_{15...0} od uređaja. Prioriteti su fiksni, pa je linija **BRQ**₃₁ najvišeg, a linija **BRQ**₀ najnižeg prioriteta. Po linijama **BG**_{31...16} odlaze iz arbitratora signali dozvola **BG** ka memorijskim modulima, a po linijama **BG**_{15...0} ka uređajima. Ovakvim načinom povezivanja signala zahteva **BRQ** i dozvola **BG** između arbitratora sa jedne i memorijskih modula i uređaja sa druge strane obezbeđeno je da zahtevi koji dolaze iz memorijskih modula imaju viši prioritet od zahteva koji dolaze iz uređaja. Time se obezbeđuje da memorijski moduli čim imaju očitani bajt podatka, dobijaju magistralu, vraćaju bajt podatka i postaju slobodni da prime novi zahtev za čitanje ili upis.



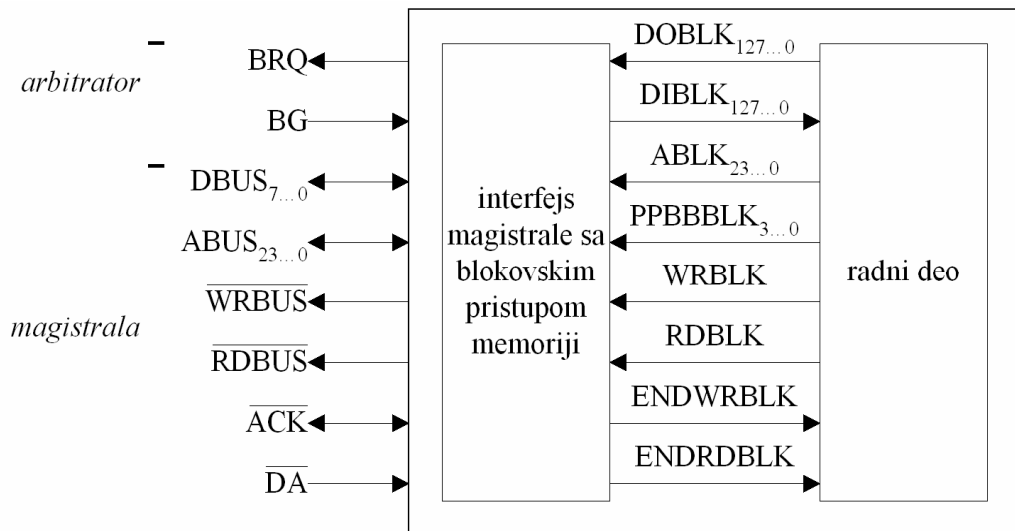
Slika 3.7 Arbitrator

Osnovni delovi arbitratora su koder prioriteta PCOD (Priority Coder), dekodek DEC i registar dozvola BGR (Bus Grant Register). Na ulaze koder prioriteta dovode se prispeli zahtevi za magistralu $BRQ_{31...0}$. Na svojim izlazima koder prioriteta PCOD daje broj one ulazne linije po kojoj je stigao zahtev najvišeg prioriteta. Dekoder DEC dekoduje taj broj i na svojim izlazima daje aktivnu vrednost signala samo na liniji koja odgovara tom broju. Signal **ENABLE** na izlazu **W** koder prioriteta PCOD ima aktivnu vrednost ako je barem jedan od signala $BRQ_{31...0}$ aktivan. Vrednosti sa izlaza dekodekera DEC se na signal takta upisuju u registar BGR. Time se formiraju vrednosti signala dozvola $BG_{31...0}$. Te vrednosti ostaju do sledećeg signala takta kada se, na osnovu arbitracije realizovane u periodi signala takta između ta dva signala takta, u registar BGR upisuju nove vrednosti i time formiraju nove vrednosti signala dozvola $BG_{31...0}$.

1.3 Uređaj sa blokovskim pristupom memoriji

Uređaj sa blokovskim pristupom memoriji se sastoji iz dva dela (slika 3.8):

- radni deo i
- interfejs magistrale sa blokovskim pristupom memoriji.



Slika 3.8 Uređaj sa blokovskim pristupom memoriji

Radni deo je osnovni deo uređaja sa blokovskim pristupom memoriji čija se struktura i funkcionalnost razlikuju od uređaja do uređaja. Radni deo može da se obrati interfejsu magistrale sa blokovskim pristupom memoriji sa zahtevom da realizuje sledeće dve operacije sa memorijom:

- upis bloka podataka i
- čitanje bloka podataka.

Interfejs magistrale sa blokovskim pristupom memoriji ove operacije sa memorijom realizuje koristeći ranije opisane cikluse na magistrali.

1.3.1 Radni deo

Kod operacije upisa radni deo šalje interfejsu 20-bitnu adresu po linijama **ABLK_{19...0}**, poziciju poslednjeg bajta u bloku po linijama **PPBBBLK_{3...0}**, blok podataka za upis maksimalne veličine 16 bajtova po linijama **DOBLK_{127...0}** i generiše aktivnu vrednost signala **WRBLK**. Operacija upisa je završena kada interfejs pošalje radnom delu aktivnu vrednost signala **ENDWRBLK**.

Kod operacije čitanja radni deo šalje interfejsu 20-bitnu adresu po linijama **ABLK_{19...0}**, poziciju poslednjeg bajta u bloku po linijama **PPBBBLK_{3...0}** i generiše aktivnu vrednost signala **RDBLK**. Očitani blok podataka se vraća po linijama **DIBLK_{127...0}**. Operacija čitanja je završena i na linijama **DIBLK_{127...0}** je važeći blok podataka kada interfejs pošalje radnom delu aktivnu vrednost signala **ENDRDBLK**.

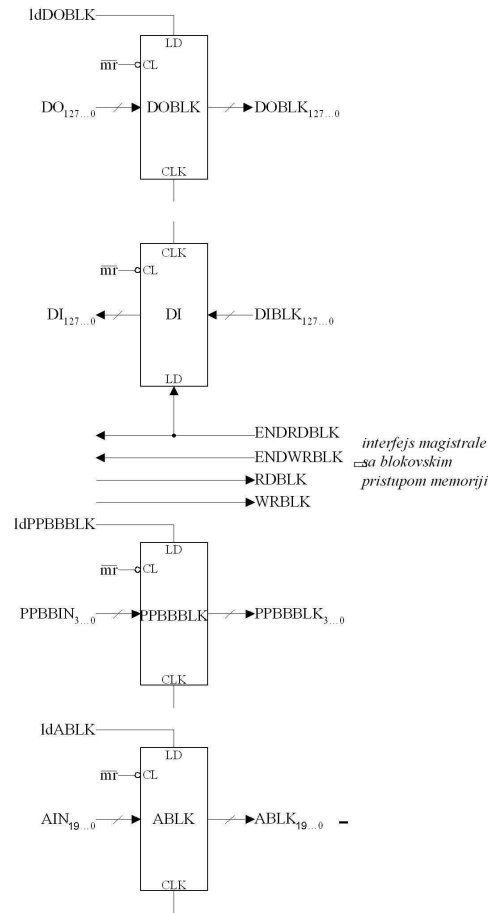
Deo radnog dela za povezivanje sa interfejsom magistrale sa blokovskim pristupom memoriji prikazan je na slici 3.9.

Registar ABLK služi za čuvanje ili adrese lokacije memorije od koja treba upisati blok podataka u slučaju operacije upisa ili adrese lokacije memorije od koje treba očitati blok podataka u slučaju operacije čitanja. Pretpostavlja se da je radni deo pre obraćanja interfejsu već upisao adresu u registar ABLK, tako što je generisao aktivnu vrednost signala **ldABLK** i na signal takta **CLK** izvršio upis sadržaja sa linija **AIN_{19...0}**.

Registar PPBBBLK služi za čuvanje pozicije poslednjeg bajta u bloku. Pretpostavlja se da je radni deo pre obraćanja interfejsu već upisao vrednost pozicije poslednjeg bajta u bloku u registar PPBBBLK, tako što je generisao aktivnu vrednost signala **ldPPBBBLK** i na signal takta **CLK** izvršio upis sadržaja sa linija **PPBBIN_{3...0}**.

Registar DOBLK služi za čuvanje bloka podataka koje treba upisati kod operacije upisa.. Pretpostavlja se da je radni deo pre obraćanja interfejsu već upisao blok podataka u registar DOBLK, tako što je generisao aktivnu vrednost signala **ldDOBLK** i na signal takta **CLK** izvršio upis sadržaja sa linija **DO_{127...0}**.

Registar DI služi za čuvanje bloka podataka koji su očitani kod operacije čitanja. Aktivna vrednost signala **ENDRDBLK** se koristi da se na signal takta **CLK** izvrši upis očitnog bloka podataka sa linija **DIBLK_{127...0}** u registar DI.



Slika 3.9 Deo radnog dela za povezivanje sa interfejsom magistrale sa blokovskim pristupom memoriji

Upravljački signal **WRBLK** se koristi da radni deo aktivnom vrednošću ovog signala trajanja jedne periode signala takta **CLK** signalizira interfejsu da se na linijama **ABLK_{19...0}** nalazi adresa lokacije počev od koje treba upisati blok podataka, da se na linijama **DOBLK_{127...0}** nalazi blok podataka, da se na linijama **PPBBBLK_{3...0}** nalazi pozicija poslednjeg bajta u bloku, i da interfejs treba da krene sa operacijom upisa.

Upravljački signal **RDBLK** se koristi da radni deo aktivnom vrednošću ovog signala trajanja jedne periode signala takta **CLK** signalizira interfejsu da se na linijama **ABLK_{19...0}** nalazi adresa lokacije počev od koje treba očitati blok podataka, da se na linijama **PPBBBLK_{3...0}** nalazi pozicija poslednjeg bajta u bloku i da interfejs treba da krene sa operacijom čitanja.

Upravljački signali **ENDWRBLK** i **ENDRDBLK** se koriste da interfejs aktivnom vrednošću ovog signala trajanja jedne periode signala takta **CLK** signalizira radnom delu da je operacija upisa u memoriju ili čitanja iz memorije završena. U slučaju operacije čitanja iz memorije aktivnom vrednošću signala **ENDRDBLK** se signalizira i da se na linijama **DIBLK_{127...0}** nalazi očitani blok podataka i da aktivnom vrednošću signala **ENDRDBLK** treba na signal takta **CLK** ovaj sadržaj upisati u registar **DI**.

1.3.2 Interfejs magistrale sa blokovskim pristupom memoriji

Interfejs magistrale sa blokovskim pristupom memoriji dobija od radnog dela zahteve da realizuje upis bloka podataka u memoriju i čitanje bloka podataka iz memorije.

Operacija upisa bloka podataka se realizuje korišćenjem ciklusa na magistrali *slanje zahteva za upis bajta podatka*. Interfejs najpre na liniji **BRQ** generiše aktivnu vrednost kojom arbitratoru upućuje zahtev za dodeljivanje magistrale. Aktivnu vrednost signala **BRQ** interfejs drži dok ne prenese, i to uspešno, zahteve za upis svih bajtova bloka. U onim periodama signala takta u kojima od arbitratora dobija aktivnu vrednost signala dozvole **BG** interfejs šalje zahtev za upis bajta bloka tako što po linijama podataka magistrale **DBUS_{7...0}** šalje bajt podatka, po adresnim linijama magistrale **ABUS_{19...0}** adresu bajta i generiše aktivnu vrednost upravljačkog signala **WRBUS**. U toj istoj periodi vrši i proveru da li je od *slave*-a dobio aktivnu vrednost signala potvrde **ACKBUS**. Ukoliko jeste, u prvoj sledećoj periodi signala takta kada dobije aktivnu vrednost signala dozvole **BG** šalje zahtev za upis sledećeg bajta bloka. Ukoliko nije ponavlja zahtev za upis istog bajta. Ovo se ponavlja dok se uspešno ne prenesu zahtevi za upis svih bajtova bloka. Tada se aktivnom vrednošću signala **ENDWRBLK** radnom delu signalizira da je ciklus *slanje zahteva za upis bajta podatka* za poslednji bajt bloka uspešno realizovan, pa da je time i operacija upisa bloka podataka završena.

Operacija *čitanje bloka podataka* se realizuje iz dva dela. U prvom delu se realizuje onoliko ciklusa na magistrali *slanje zahteva za čitanje bajta podatka* kolika je veličina bloka u bajtovima. U drugom delu se realizuje isto toliko ciklusa na magistrali *vraćanje bajta podatka*. Postupak slanja zahteva arbitratoru za dodeljivanje magistrale i generisanje ciklusa na magistrali *slanje zahteva za čitanje bajta podatka* se realizuje na isti način kao i za slučaj operacije *upis bloka podataka*. Jedino ovde u onim periodama signala takta u kojima od arbitratora dobija aktivnu vrednost signala dozvole **BG** interfejs generiše zahtev za čitanje bajta podatka tako što po adresnim linijama magistrale **ABUS_{19...0}** šalje adresu bajta, po linijama podataka magistrale **DBUS_{7...0}** šalje svoj identifikator i generiše aktivnu vrednost signala **RDBUS**. Kod identifikatora bitovi 7 do 4 predstavljaju broj uređaja, a bitovi 3 do 0 poziciju bajta u bloku. Po uspešno realizovanom ciklusu na magistrali *slanje zahteva za čitanje bajta podatka* za poslednji bajt bloka prelazi se na realizaciju drugog dela operacije čitanje bloka podataka. U okviru njega posebnim ciklusima na magistrali *vraćanje bajta podatka* mora da se vrati svaki bajt bloka čije je čitanje u prvom delu operacije čitanja bloka podataka inicirano ciklusima *slanje zahteva za čitanje bajta podatka*. Kod ciklusa *vraćanje bajta podatka* interfejs je *slave*. Kod ovog ciklusa podatak koji se vraća nalazi se na linijama **DBUS_{7...0}**, identifikator uređaja na linijama **ABUS_{7...0}** i na upravljačkoj liniji **DABUS** je aktivna vrednost. Interfejs prihvata podatak ako broj uređaja identifikatora, određen bitovima 7 do 4 identifikatora, odgovara broju datog uređaja. Pozicija unutar prihvatnog registra podatka u koju se upisuje bajt podatka data je pozicijom bajta u bloku identifikatora, određenom bitovima 3 do 0 identifikatora. Tek pošto se prime svi očitani bajtovi bloka interfejs magistrale sa blokovskim pristupom aktivnom vrednošću signala **ENDRDBLK** radnom delu signalizira da je operacija čitanja bloka podataka završena.

Interfejs magistrale sa blokovskim pristupom memoriji se sastoji iz operacione jedinice i upravljačke jedinice.

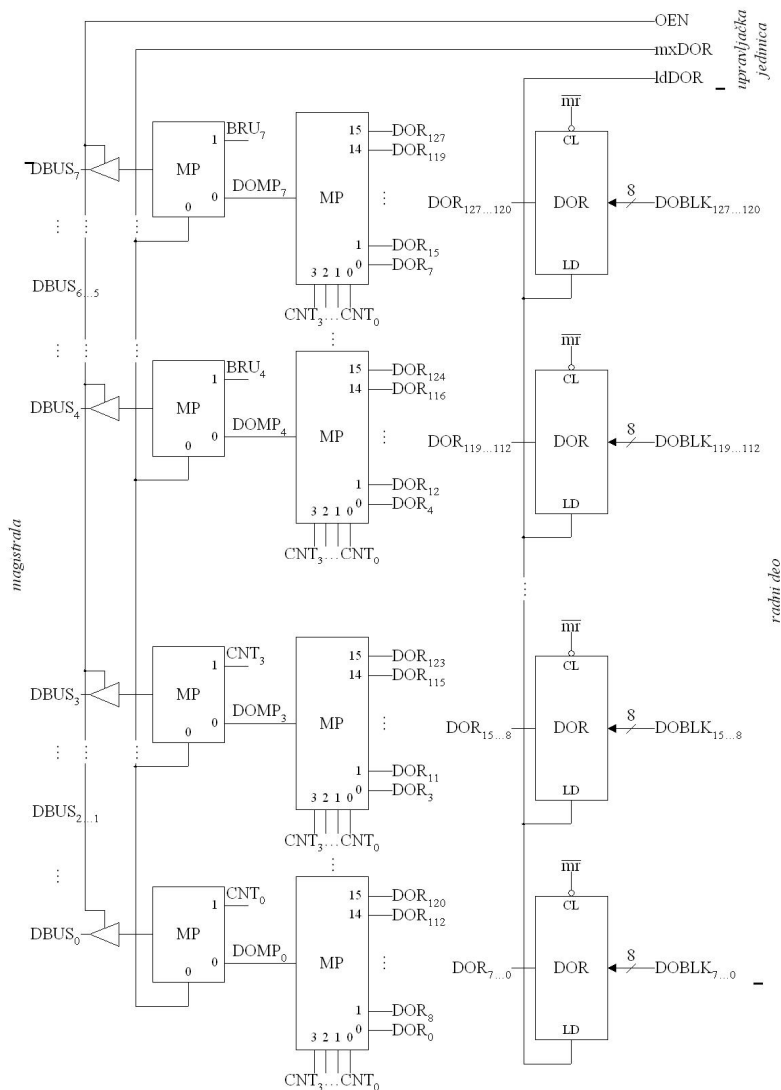
1.3.2.1 Operaciona jedinica

Operaciona jedinica se sastoji iz sledećih delova:

- prihvatnog registra bloka podataka za upis
- prihvatnog registra bloka očitanih podataka i
- adresnog registra

1.3.2.1.1 Prihvatni register bloka podataka za upis

Ovaj deo se sastoji od registra DOR, 16-ulaznih multipleksera MP16 i dvoulaznih multipleksera MP2.



Slika 3.10 Prihvatni registar bloka podataka za upis

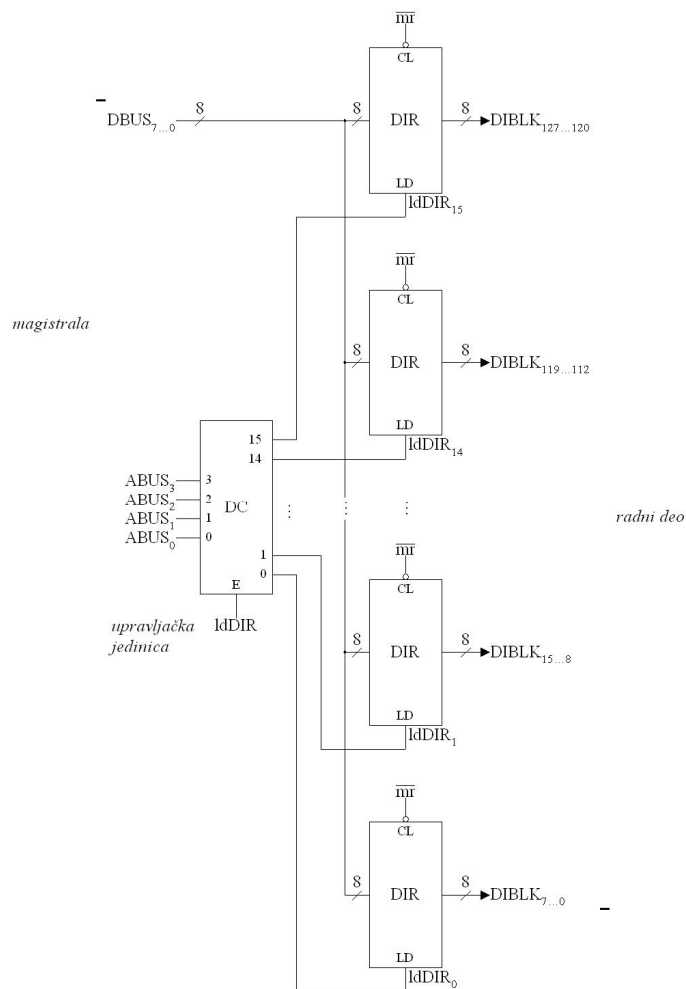
Registar DOR (*Data Output Register*) veličine 128 razreda služi za čuvanje bloka podataka koji treba upisati u slučaju operacije upisa.

Multiplekseri MP16 kojih ima 8 služe za selekciju jednog od 16 bajtova bloka podataka iz registra DOR koji treba poslati na linije podataka magistrale kod operacije upisa.

Multiplekseri MP2 kojih ima 8 služe da na linije podataka magistrale propuste selektovani bajt bloka kod operacije upisa i identifikator uređaja koji se sastoji iz broja uređaja i adrese bajta u bloku, u slučaju operacije čitanja.

1.3.2.1.2 Prihvatni registar bloka očitanih podataka

Ovaj deo se sastoji od registra DIR i dekodera sa 16 izlaza.



Slika 3.11 Prihvatni registar bloka očitanih podataka

Registar DIR (*Data Input Register*) veličine 128 razreda služi za čuvanje bloka očitanih podataka. Registar je tako organizovan da se u stvari sastoji od 16 registara veličine 8 razreda. Stoga je moguće u njega upisati posebno svaki očitani bajt bloka.

1.3.2.1.3 Adresni registar

Ovaj deo se sastoji od registara AR, PPBB, CNT i CNTRET, mikroprekidača BRU_{7...4} i komparatora CMP1, CMP2 i CMP3.



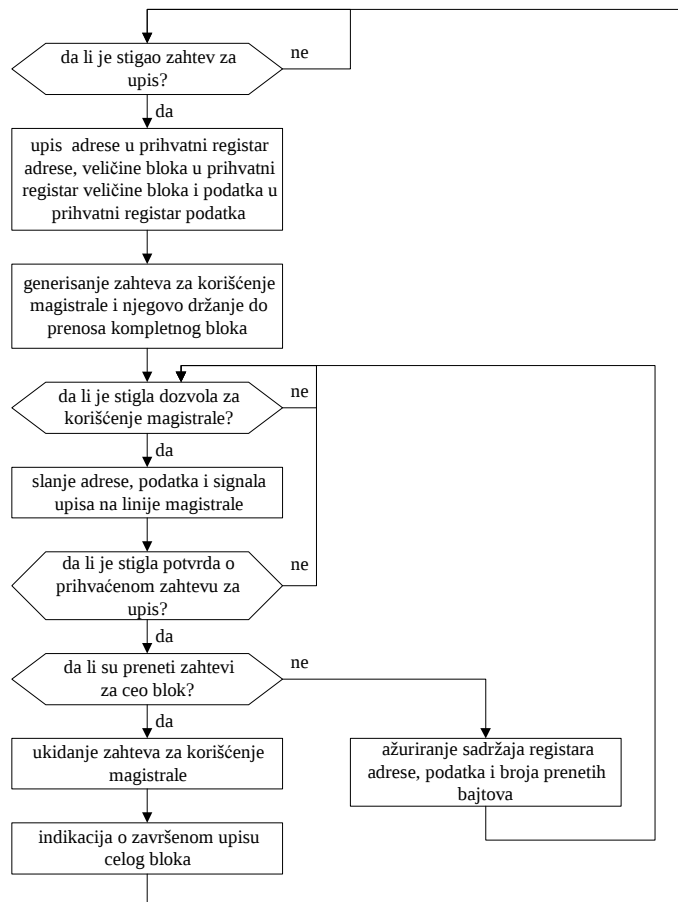
Komparator CMP2 služi za utvrđivanje da li se pri ciklusima na magistrali *slanje zahteva za upis bajta podatka* i *slanje zahteva za čitanje bajta podatka* došlo do poslednjeg bajta u bloku. Radi toga upoređuje se sadržaj registara PPBB, koji sadrži poziciju poslednjeg bajta u

bloku, i CNT, koji sadrži poziciju bajta u bloku za koji treba realizovati ciklus čitanja ili upisa. U slučaju saglasnosti signal **ENDBLK** dobija aktivnu vrednost.

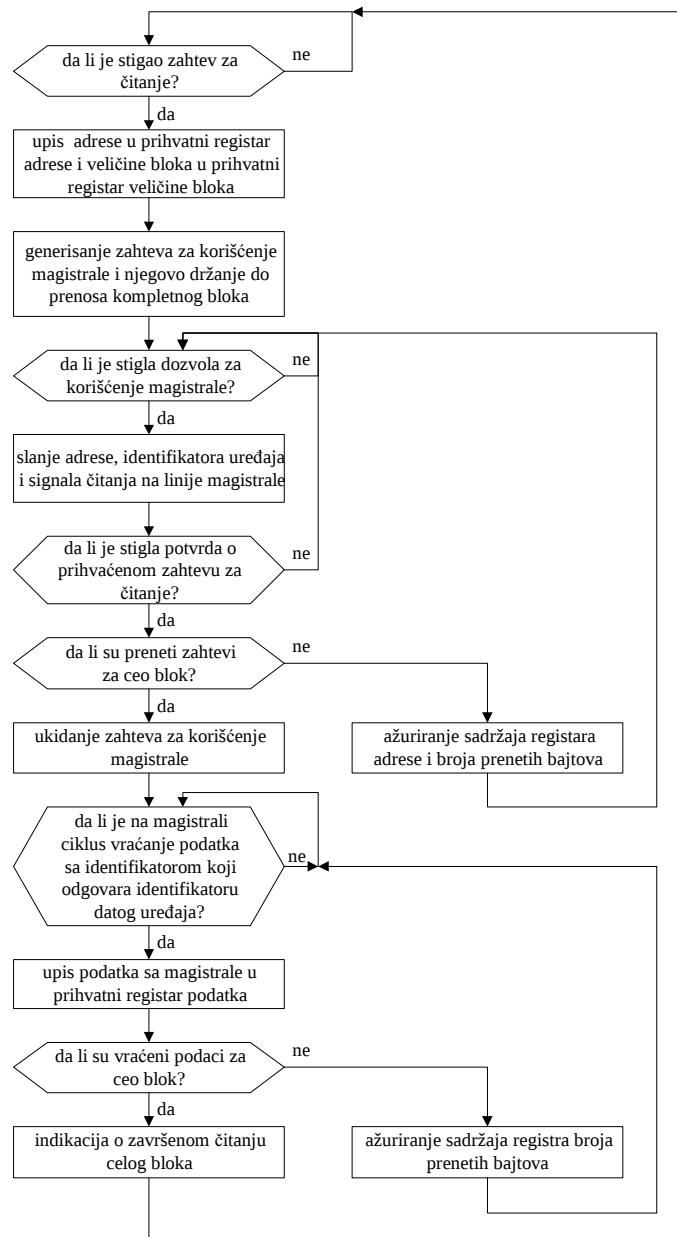
Komparator CMP3 služi za utvrđivanje da li se pri ciklusima *vraćanje bajta podatka* došlo do poslednjeg bajta u bloku. Radi toga upoređuje se sadržaj registara PPBB, koji sadrži poziciju poslednjeg bajta u bloku, i CNTRET, koji sadrži poziciju bajta u bloku za koji treba realizovati ciklus *vraćanje bajta podatka*. U slučaju saglasnosti signal **ENDBLK** dobija aktivnu vrednost.

1.3.2.2 Upravljačka jedinica

Dijagram toka operacija za operaciju upisa dat je na slici 3.13, a za operaciju čitanja na slici 3.14.



Slika 3.13 Dijagram toka operacija za operaciju upisa bloka podataka

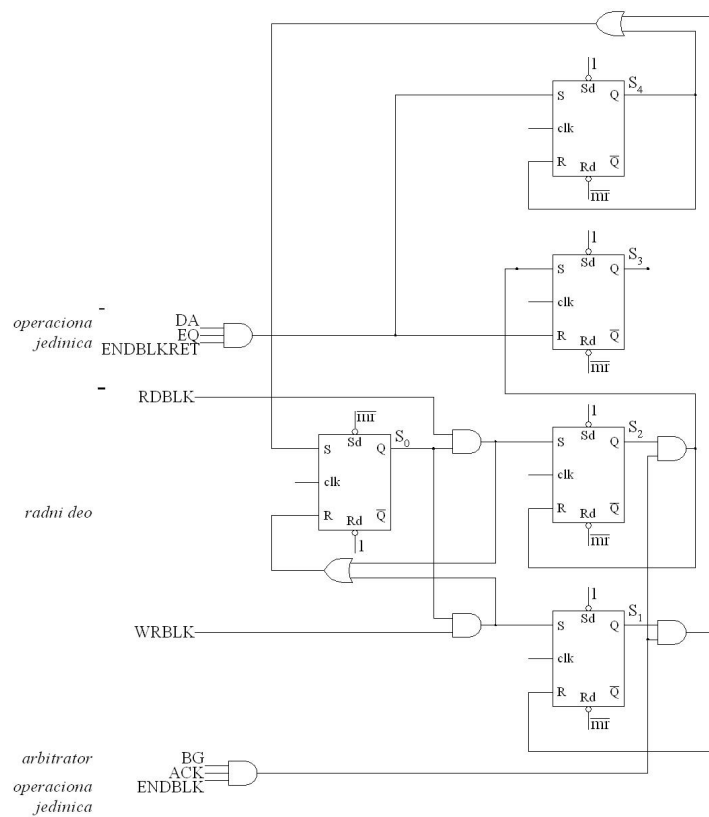


Slika 3.14 Dijagram toka operacija za operaciju čitanja bloka podataka

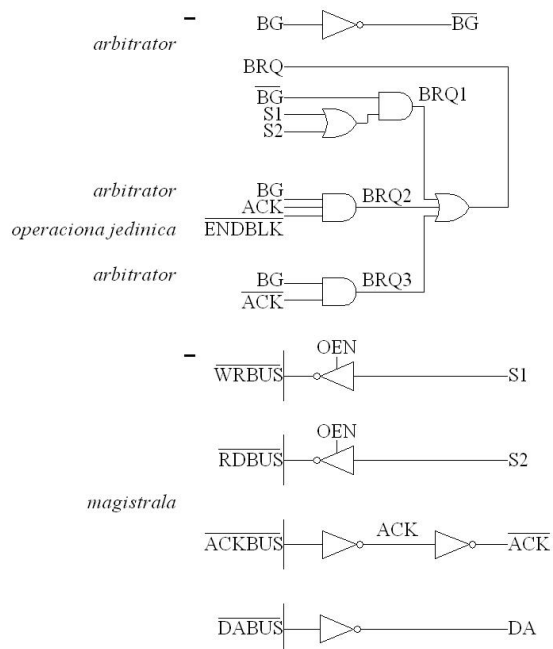
Na osnovu dijagrama toka operacija i strukture operacione jedinice, realizovana je upravljačka jedinica.

Upravljačka jedinica se sastoji iz tri dela:

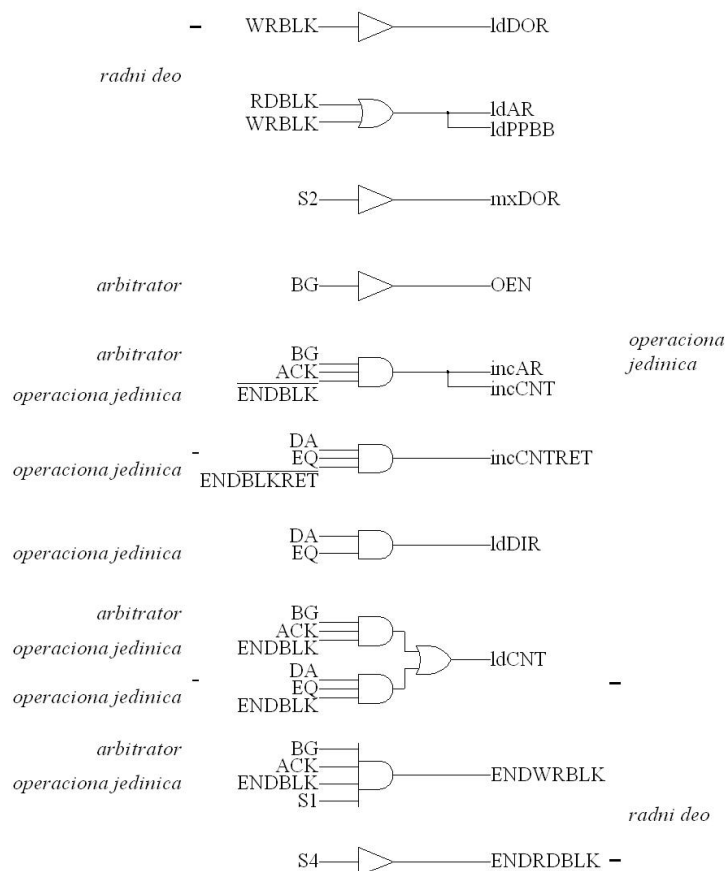
- elementi za kašnjenje (slika 3.15)
- povezivanje sa arbitratorom i upravljačkom magistralom (slika 3.16) i
- generisanje upravljačkih signala (slika 3.17).



Slika 3.15 Elementi za kašnjenje



Slika 3.16 Povezivanje sa arbitratorom i upravljačkom magistralom



Slika 3.17 Generisanje upravljačkih signala

Deo *elementi za kašnjenje* je realizovan na principu "šetajuće jedinice". Sastoji se od pet flip-flopova S_0 , S_1 , S_2 , S_3 i S_4 koji predstavljaju pet stanja u kojima se može naći upravljačka jedinica. Prema principu "šetajuće jedinice" jedinica u datom trenutku može da se nađe samo u jednom flip-flopu dok su u ostalima nule. Na početku se upravljačka jedinica nalazi u stanju S_0 , pa je jedinica u flip-flopu S_0 .

Kod operacije upisa upravljačka jedinica prelazi iz stanja S_0 u S_1 u kome generiše cikluse zahtev za upis bajta podatka za sve bajtove bloka i vraća se u stanje S_0 . Stoga jedinica šeta iz flip-flopa S_0 u S_1 i vraća se u flip-flop S_0 .

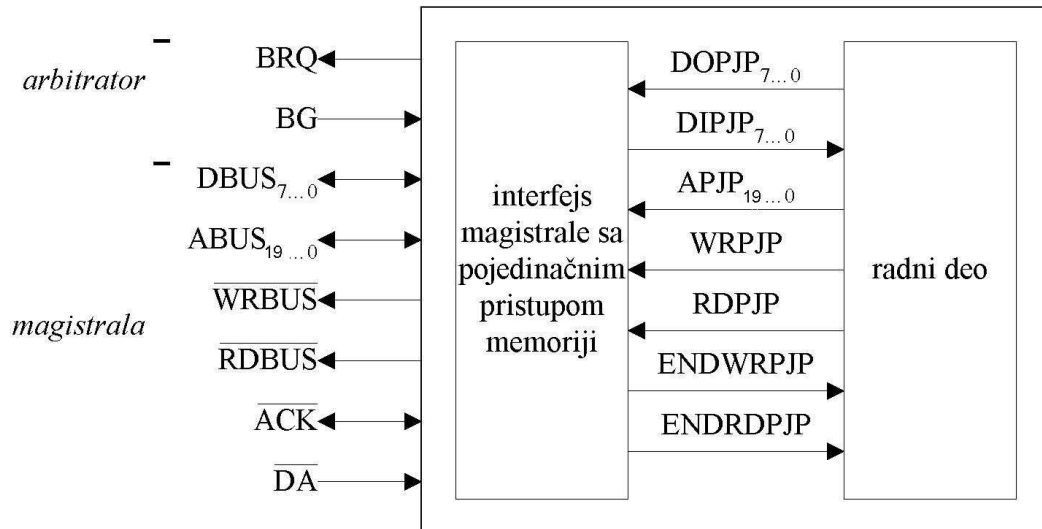
Kod operacije čitanja, najpre se na magistrali realizuju ciklusi čitanja prelaskom iz stanja S_0 u S_2 . Potom upravljačka jedinica prelazi u stanje S_3 u kome se ciklusima *vraćanje bajta podatka* vraćaju i smeštaju u prihvatni registar očitani bajtovi bloka. Po prijemu svih bajtova bloka upravljačka jedinica prelazi u stanje S_4 radi slanja indikacije radnom delu o završenoj operaciji čitanja. Na kraju upravljačka jedinica se vraća u stanje S_0 . Stoga jedinica sada šeta iz flip-flopa S_0 u S_2 , iz S_2 u S_3 , iz S_3 u S_4 i na kraju se iz S_4 vraća u flip-flop S_0 .

Jedinica prelazi iz flip-flopa S_0 u S_1 pri pojavi signala za upis **WRBLK**. Jedinica prelazi iz flip-flopa S_0 u S_2 pri pojavi signala za čitanje **RDBLK**. Jedinica prelazi iz flip-flopa S_2 u S_3 po uspešno poslatom zahtevu za čitanje i poslednjeg bajta bloka. Ovo se omogućava aktivnim vrednostima signala **BG**, **ACK** i **ENDBLK**. Jedinica prelazi iz flip-flopa S_3 u S_4 ako su vraćeni svi očitani bajtovi bloka što se indicira aktivnim vrednostima signala **DA**, **EQ** i **ENDBLKRET**. Jedinica se vraća iz flip-flopa S_1 u S_0 po uspešno poslatom zahtevu za upis i poslednjeg bajta bloka. To se indicira aktivnim vrednostima signala **BG**, **ACK** i **ENDBLK**. Jedinica prelazi iz flip-flopa S_4 u S_0 na prvi signal takta **CLK** pošto je u flip-flop S_4 upisana.

1.4 Uređaj sa pojedinačnim pristupom memoriji

Uređaj sa pojedinačnim pristupom memoriji sastoji se iz dva dela (slika 3.18):

- radni deo i
- interfejs magistrale sa pojedinačnim pristupom memoriji.



Slika 3.18 Uređaj sa pojedinačnim pristupom memoriji

Radni deo je osnovni deo uređaja sa pojedinačnim pristupom memoriji. Njegova struktura i funkcionalnost se razlikuju od uređaja do uređaja. Radni deo može da se obrati interfejsu magistrale sa pojedinačnim pristupom memoriji sa zahtevom da realizuje sledeće dve operacije sa memorijom:

- upis bajta podatka i
- čitanje bajta podatka.

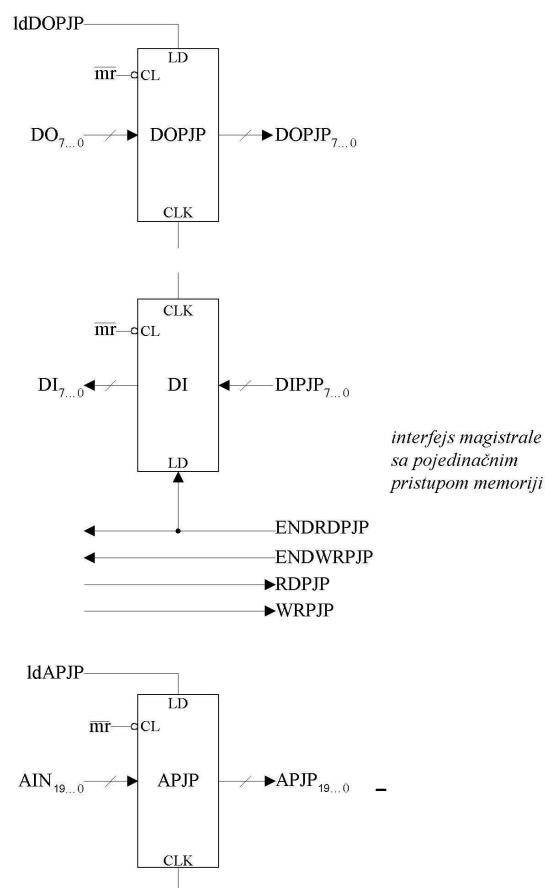
Interfejs magistrale sa blokovskim pristupom memoriji ove operacije sa memorijom realizuje koristeći ranije opisane cikluse na magistrali.

1.4.1 Radni deo

Kod operacije upisa radni deo šalje interfejsu 20-bitnu adresu po linijama **APJP**_{19...0}, podatak za upis po linijama **DOPJP**_{7...0} i generiše aktivnu vrednost signala **WRPJP**. Operacija upisa je završena kada interfejs pošalje radnom delu aktivnu vrednost signala **ENDWRPJP**.

Kod operacije čitanja radni deo šalje interfejsu 20-bitnu adresu po linijama **APJP**_{19...0} i generiše aktivnu vrednost signala **RDPJP**. Očitani podatak se vraća po linijama **DIPJP**_{7...0}. Operacija čitanja je završena i na linijama **DIPJP**_{7...0} je važeći podatak onda kada interfejs pošalje radnom delu aktivnu vrednost signala **ENDRDPJP**.

Deo radnog dela za povezivanje sa interfejsom magistrale sa pojedinačnim pristupom memoriji prikazan je na slici 3.19.



Slika 3.19 Deo radnog dela za povezivanje sa interfejsom magistrale sa pojedinačnim pristupom memoriji

Registar APJP služi za čuvanje ili adrese lokacije memorije u koju treba upisati podatak u slučaju operacije upisa ili adrese lokacije memorije sa koje treba očitati podatak u slučaju operacije čitanja. Izlazne linije ovog registra se vode kao 20 adresnih linija $APJP_{19...0}$ u interfejs. Pretpostavlja se da je radni deo pre obraćanja interfejsu već upisao adresu u registar APJP, tako što je generisao aktivnu vrednost signala $ldAPJP$ i na signal takta CLK izvršio upis sadržaja sa linija $AIN_{19...0}$.

Registar DOPJP služi za čuvanje podatka koji treba upisati kod operacije upisa. Izlazne linije ovog registra se vode kao 8 linija podataka $DOPJP_{7...0}$ u interfejs. Pretpostavlja se da je radni deo pre obraćanja interfejsu već upisao podatak u registar DOPJP, tako što je generisao aktivnu vrednost signala $ldDOPJP$ i na signal takta CLK izvršio upis sadržaja sa linija $DO_{7...0}$.

Registar DI služi za čuvanje podatka koji je očitao kod operacije čitanja. Očitani podatak se iz interfejsa vodi po linijama $DIPJP_{7...0}$ na ulaze registra DI. Aktivna vrednost signala $ENDRDPJP$ se koristi da se na signal takta CLK izvrši upis očitanoog podatka sa linija $DIPJP_{7...0}$ u registar DI.

Upravljački signal $WRPJP$ se koristi da radni deo aktivnom vrednošću ovog signala trajanja jedne periode signala takta CLK signalizira interfejsu da se na linijama $APJP_{19...0}$ nalazi adresa lokacije u koju treba upisati podatak, da se na linijama $DOPJP_{7...0}$ nalazi podatak i da interfejs treba da krene sa operacijom upisa.

Upravljački signal $RDPJP$ se koristi da radni deo aktivnom vrednošću ovog signala trajanja jedne periode signala takta CLK signalizira interfejsu da se na linijama $APJP_{19...0}$

nalazi adresa lokacije sa koje treba očitati podatak i da interfejs treba da krene sa operacijom čitanja.

Upravljački signali **ENDWRPJP** i **ENDRDPJP** se koriste da interfejs aktivnom vrednošću ovog signala trajanja jedne periode signala takta **CLK** signalizira radnom delu da je operacija upisa u memoriju ili čitanja iz memorije završena. U slučaju operacije čitanja iz memorije aktivnom vrednošću signala **ENDRDPJP** se signalizira i da se na linijama **DIPJP_{7...0}** nalazi očitani podatak i da aktivnom vrednošću signala **ENDRDPJP** treba na signal takta **CLK** ovaj sadržaj upisati u registar DI.

1.4.2 Interfejs magistrale sa pojedinačnim pristupom memoriji

Interfejs magistrale sa pojedinačnim pristupom memoriji dobija od radnog dela zahteve da realizuje upis bajta podatka u memoriju i čitanje bajta podatka iz memorije.

Operacija upis bajta podatka se realizuje korišćenjem ciklusa na magistrali *zahtev za upis bajta podatka*. Interfejs najpre na liniji **BRQ** generiše aktivnu vrednost kojom arbitratoru upućuje zahtev za dodeljivanje magistrale. Aktivnu vrednost signala **BRQ** interfejs drži dok uspešno ne prenese zahtev za upis bajta podatka. U onoj periodu signala takta u kojoj od arbitratora dobije aktivnu vrednost signala dozvole **BG** interfejs generiše zahtev za upis bajta podatka tako što po linijama podataka magistrale **DBUS_{7...0}** šalje bajt podatka, po adresnim linijama magistrale **ABUS_{19...0}** adresu bajta i generiše aktivnu vrednost upravljačkog signala **WRBUS**. U toj istoj periodu vrši i proveru da li je od *slave*-a dobio aktivnu vrednost signala potvrde **ACKBUS**. Ukoliko nije, ponavlja zahtev za upis istog bajta podatka, i to sve dotle dok se uspešno ne prenese. Ukoliko jeste, tada se aktivnom vrednošću signala **ENDWRPJP** radnom delu signalizira da je zahtev za upis bajta podatka uspešno prenet, pa da je time i operacija upisa bajta podatka završena.

Operacija čitanje bajta podatka se realizuje iz dva dela. U prvom delu se na magistrali realizuje ciklus *zahtev za čitanje bajta podatka*. U drugom delu se na magistrali realizuje ciklus *vraćanje bajta podatka*. Postupak slanja zahteva arbitratoru za dodeljivanje magistrale i generisanja ciklusa *zahtev za čitanje bajta podatka* se realizuju na sličan način kao i za ciklus *zahtev za upis bajta podatka*. Razlika je jedino da ovde u onoj periodu signala takta u kojoj od arbitratora dobije aktivnu vrednost signala dozvole **BG** interfejs generiše zahtev za čitanje bajta podatka tako što po adresnim linijama magistrale **ABUS_{19...0}** šalje adresu bajta, po linijama podataka magistrale **DBUS_{7...0}** šalje identifikator i generiše aktivnu vrednost upravljačkog signala **RDBUS**. Po završetku ciklusa *zahtev za čitanje bajta podatka* interfejs magistrale sa pojedinačnim pristupom memoriji prelazi na čekanje ciklusa *vraćanje bajta podatka*. Kod ciklusa *vraćanje bajta podatka* interfejs je *slave*. Kod ovog ciklusa podatak koji se vraća nalazi se na linijama podataka magistrale **DBUS_{7...0}**, identifikator uređaja na adresnim linijama magistrale **ABUS_{7...0}** i na upravljačkoj liniji **DABUS** je aktivna vrednost. Interfejs prihvata podatak ako broj uređaja identifikatora određen bitovima 7 do 4 identifikatora odgovara broju datog uređaja. Tek pošto se primi očitani bajt podatka interfejs magistrale sa pojedinačnim pristupom aktivnom vrednošću signala **ENDRDPJP** radnom delu signalizira da je operacija čitanja bajta podatka završena.

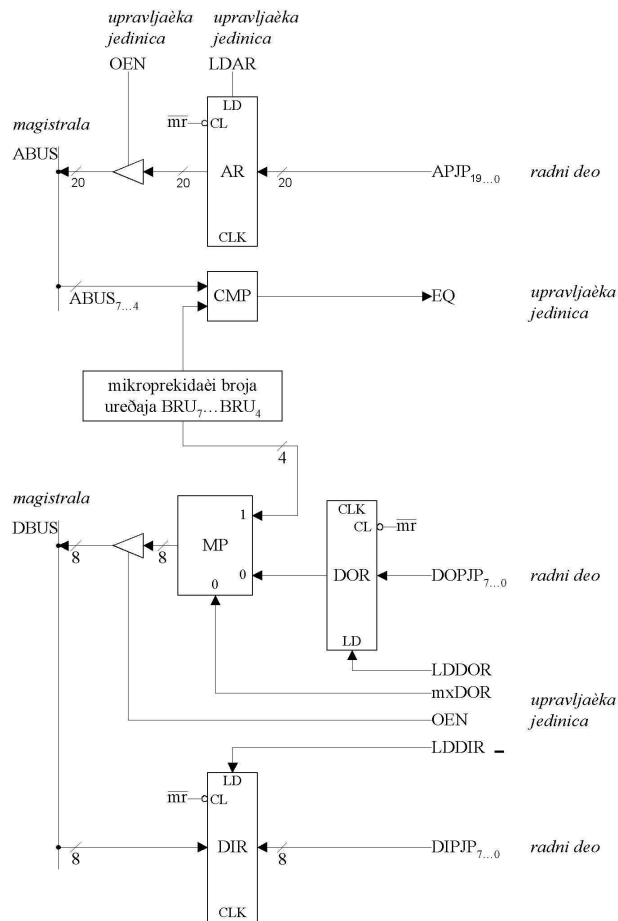
Interfejs magistrale sa pojedinačnim pristupom memoriji se sastoji iz dva dela:

- operacione jedinice i
- upravljačke jedinice.

1.4.2.1 Operaciona jedinica

Operaciona jedinica se sastoji iz sledećih delova (slika 3.20):

- mikroprekidači broja uređaja,
- prihvatni registar podatka za upis DOR i multiplekser MP,
- prihvatni registar očitnog podataka DIR,
- adresni registar AR i
- komparator CMP.



Slika 3.20 Struktura operacione jedinice

Mikroprekidači $BRU_{7...4}$ (*Broj Uređaja*) služe za postavljanje vrednosti broja uređaja. Mikroprekidači $BRU_{7...4}$ formiraju bitove 7 do 4 identifikatora uređaja. Bitovi 3 do 0 identifikatora uređaja se ne koriste kod uređaja sa pojedinačnim pristupom memoriji, pa njihove vrednosti nisu bitne.

Registar DOR (*Data Output Register*) veličine 8 razreda služi za čuvanje bajta podatka u slučaju operacije upisa i identifikatora uređaja za slučaj operacije čitanja. Na ulaze registra DOR dovode se izlazi multipleksera MP. Multiplekseri MP, kojih ima 8, služe da propuste bajt podatka kod operacije upisa i identifikator uređaja u slučaju operacije čitanja. Selekcija jednog od ova dva sadržaja se realizuje signalom $m̄xDOR$. Na ulaze multipleksera dovode se bajt podatka po linijama $DOPJP_{7...0}$ i identifikator uređaja po linijama $BRU_{7...4}$. Selektovani podatak se upisuje u registar DOR pri pojavi signala takta CLK ukoliko je aktivan signal

ldDOR. Sadržaj registra DOR se pušta na linije podataka magistrale aktivnom vrednošću signala **OEN**.

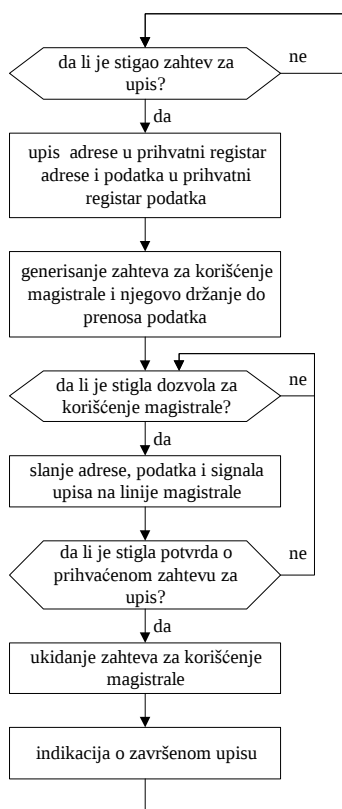
Registar DIR (*Data Input Register*) veličine 8 razreda služi za čuvanje očitano bajta podatka. Na ulaze registra dovodi se sadržaj sa linija podataka magistrale **DBUS_{7...0}**. Pri aktivnoj vrednosti signala **ldDIR** i pojavi signala takta **CLK** sadržaj sa linija **DBUS_{7...0}** se upisuje u registar DIR. Sadržaj registra DIR se po linijama **DIPJP_{7...0}** vodi u radni deo.

Registar AR (*Address Register*) služi za čuvanje adrese u koju treba upisati bajt podatka u slučaju operacije upisa ili očitati bajt podatka u slučaju operacije čitanja. Na ulaze registra AR dovodi se iz radnog dela 20-bitna adresa bajta podatka. Ova adresa se upisuje u registar AR pri pojavi signala takta **CLK** ukoliko je aktivan signal **ldAR**. Sadržaj registra AR se pušta na adresne linije magistrale aktivnom vrednošću signala **OEN**.

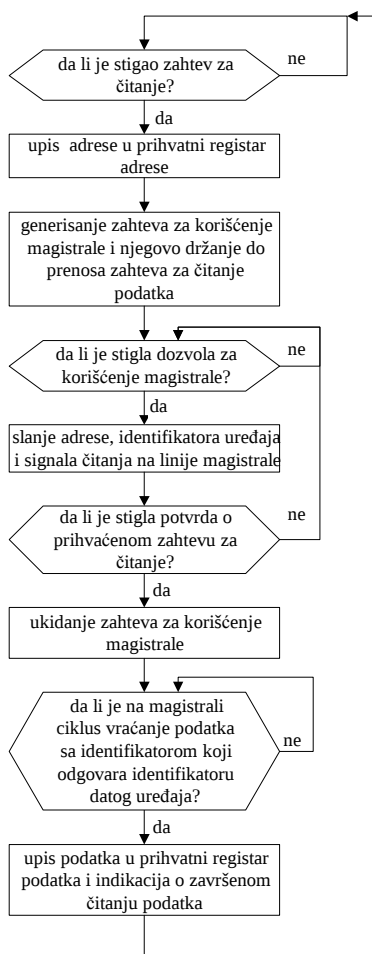
Komparator CMP služi za utvrđivanje da li se ciklus na magistrali *vraćanje bajta podatka* odnosi na dati uređaj. Radi toga upoređuje se sadržaj mikroprekidača broja uređaja BRU_{7...4} sa sadržajem na adresnim linijama magistrale **ABUS_{7...4}**, koje predstavljaju deo broj uređaja identifikatora. U slučaju saglasnosti signal **EQ** dobija aktivnu vrednost.

1.4.2.2 Upravljačka jedinica

Dijagram toka operacija za operaciju upisa dat je na slici 3.21, a za operaciju čitanja na slici 3.22.



Slika 21 Dijagram toka operacija za operaciju upisa bajta podatka



Slika 3.21 Dijagram toka operacija za operaciju čitanja bajta podatka

Na osnovu dijagrama toka operacija i strukture operacione jedinice, realizovana je upravljačka jedinica.

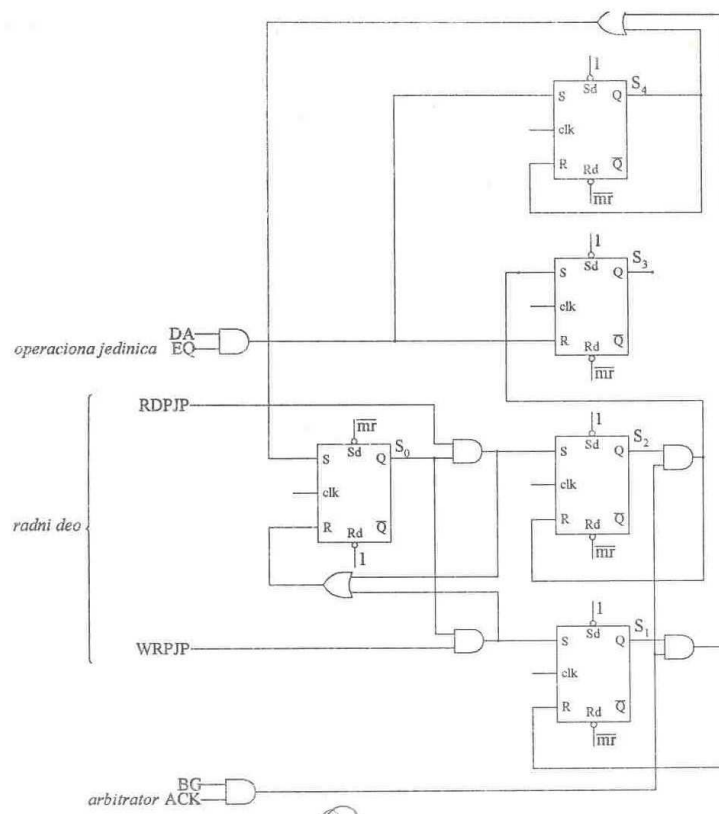
Upravljačka jedinica se sastoji iz dva dela:

- elementi za kašnjenje (slika 3.22) i
- povezivanje sa arbitratorom, upravljačkom magistralom i generisanje upravljačkih signala (slika 3.23).

Deo *elementi za kašnjenje* je realizovan na principu "šetajuće jedinice". Sastoji se od pet flip-flopova S_0 , S_1 , S_2 , S_3 i S_4 koji predstavljaju pet stanja u kojima se može naći upravljačka jedinica. Prema principu "šetajuće jedinice" jedinica u datom trenutku može da se nađe samo u jednom flip-flopu dok su u ostalima nule. Na početku se upravljačka jedinica nalazi u stanju S_0 , pa je jedinica u flip-flopu S_0 .

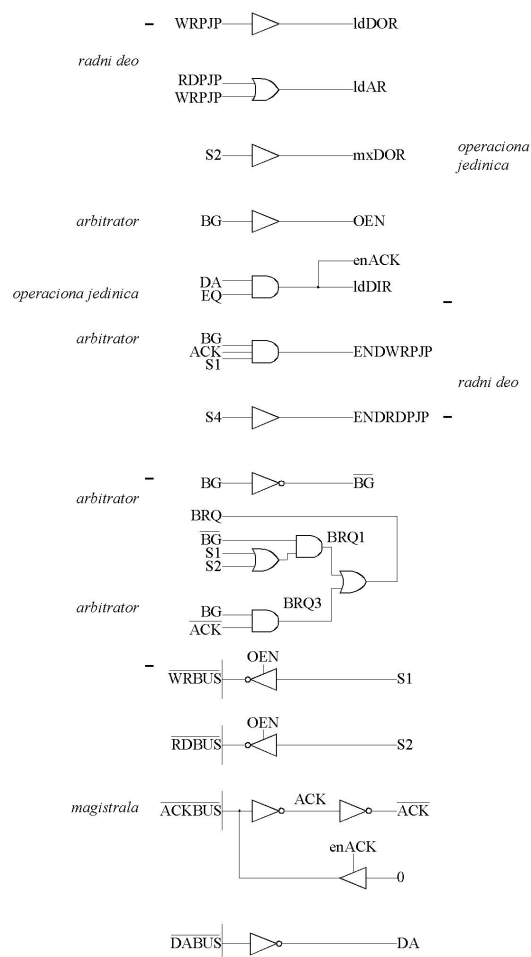
Kod operacije upisa upravljačka jedinica prelazi iz stanja S_0 u S_1 i vraća se u stanje S_0 . Stoga jedinica šeta iz flip-flopa S_0 u S_1 i vraća se u flip-flop S_0 .

Kod operacije čitanja upravljačka jedinica, najpre, prelazi iz stanja S_0 u stanje S_2 u kome generiše ciklus *zahtev za čitanje bajta podatka*. Zatim upravljačka jedinica prelazi iz stanja S_2 u stanje S_3 u kome se ciklusom *vraćanje bajta podatka* vraća i smešta u prihvatni registar očitani bajt podatka. Potom se prelazi na korak S_4 radi slanja indikacije radnom delu o završenoj operaciji čitanja bajta podatka i na kraju se vraća u stanje S_0 . Stoga jedinica sada šeta iz flip-flopa S_0 u S_2 , iz S_2 u S_3 , iz S_3 u S_4 i na kraju vraća u flip-flop S_0 .



Slika 3.22 Elementi za kašnjenje

Jedinica prelazi iz flip-flopa S_0 u S_1 pri pojavi signala za upis **WRPJP**. Jedinica prelazi iz flip-flopa S_0 u S_2 pri pojavi signala za čitanje **RDPJP**. Jedinica prelazi iz flip-flopa S_2 u S_3 po uspešno poslatom zahtevu za čitanje bajta podatka. Jedinica prelazi iz flip-flopa S_3 u S_4 ako je vraćen očitani bajt podatka što se indicira aktivnim vrednostima signala **DA** i **EQ**. Jedinica se vraća iz flip-flopa S_1 u S_0 po uspešno poslatom zahtevu za upis bajta podatka što se indicira aktivnim vrednostima signala **BG** i **ACK**. Jedinica prelazi iz flip-flopa S_4 u S_0 na prvi signal takta **CLK** pošto je u flip-flop S_4 upisana.



Slika 3.23 Povezivanje sa arbitratorom, upravljačkom magistralom i generisanje upravljačkih signala

1.5 Memorijski modul

Memorijski modul može da realizuje dve vrste operacija:

- upis bajta podatka i
- čitanje bajta podatka.

Za realizaciju ove dve operacije na magistrali se koriste sledeći ciklusi:

- zahtev za upis bajta podatka
- zahtev za čitanje bajta podatka i
- vraćanje bajta podatka.

Memorijski modul se sastoji iz dva dela:

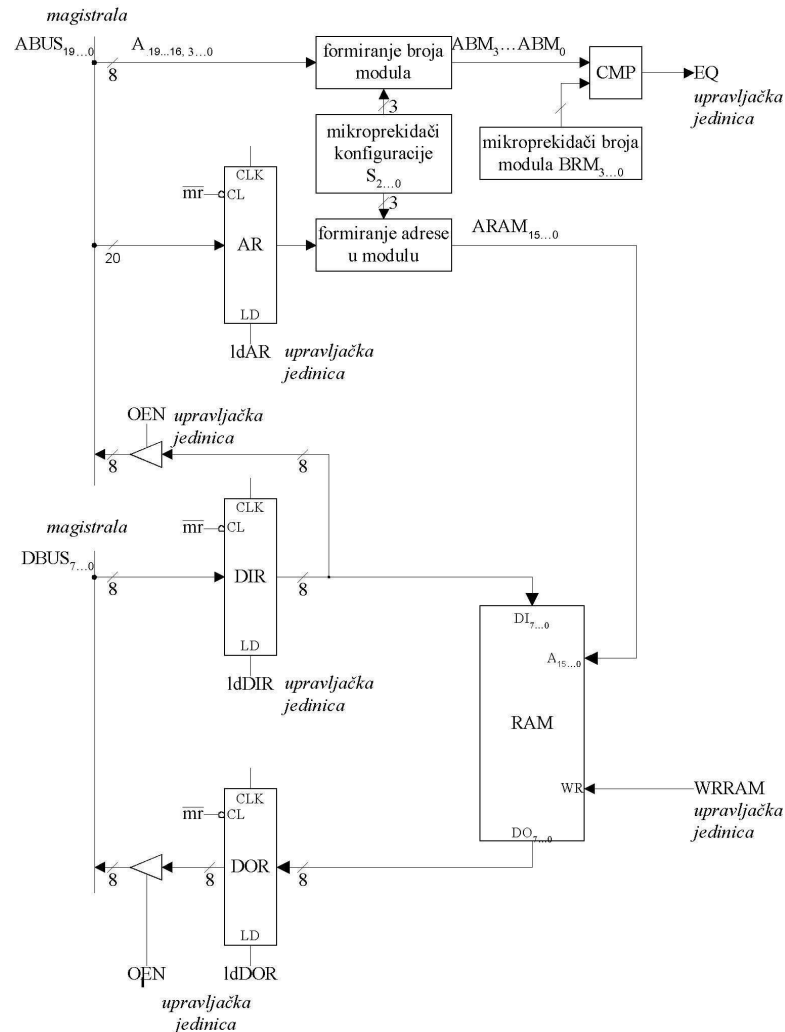
- operacione jedinice i
- upravljačke jedinice.

1.5.1 Operaciona jedinica

Operaciona jedinica se sastoji iz sledećih delova (slika 3.24):

- mikroprekidači broja modula $BRM_{3...0}$,

- prihvatni registar očitnog podatka DOR,
- prihvatni registar podataka za upis ili identifikatora DIR,
- adresni registar AR,
- mikroprekidači konfiguracije modula $S_{2...0}$ i blokovi formiranje broja modula i formiranje adrese u modulu,
- komparator CMP,
- memorija RAM i
- brojač vremena pristupa RAMACC.



Slika 3.24 Struktura operacione jedinice

Mikroprekidači broja modula $BRM_{3...0}$ služe za postavljanje vrednosti broja modula. Vrednosti broja modula memorije koje se mogu postavljati su od 0 do 15. Pretpostavlja se da su vrednosti za svaki od 16 modula korektno postavljene i da svaki memorijski modul ima jedinstveni broj.

Registar DOR (*Data Output Register*) veličine 8 razreda služi za čuvanje bajta podatka očitnog u memoriji RAM u okviru operacije *čitavanje bajta podatka*. Na ulaze registra DOR dovodi se iz memorije RAM podatak po linijama **DORAM_{7...0}**. Podatak se upisuje u registar DOR pri pojavi signala takta **CLK** ukoliko je aktivan signal **IdDOR**. Sadržaj registra DOR se

pušta na linije podataka magistrale **DBUS_{7...0}** aktivnom vrednošću signala **OEN** u okviru ciklusa na magistrali *vraćanje bajta podatka*.

Registar DIR (*Data Input Register*) veličine 8 razreda služi za čuvanje bajta podatka ili identifikatora. Na ulaze registra DIR dovodi se sadržaj sa linija podataka magistrale **DBUS_{7...0}**. Na linijama podataka magistrale **DBUS_{7...0}** nalazi se podatak u slučaju ciklusa na magistrali zahtev za upis bajta podatka. Na linijama podataka magistrale **DBUS_{7...0}** nalazi se identifikator u slučaju ciklusa *zahtev za čitanje bajta podatka*. Pri aktivnoj vrednosti signala **ldDIR** i pojavi signala takta **CLK** sadržaj sa linija **DBUS_{7...0}** se upisuje u registar DIR. Sadržaj registra DIR se po linijama **DIRAM_{7...0}** vodi u memoriju RAM u koju se upisuje u okviru operacije upisa bajta podatka. Sadržaj registra DIR se vodi i na adresne linije magistrale **ABUS_{7...0}** na koje se pušta aktivnom vrednošću signala **OEN**. Ovo se radi u slučaju ciklusa na magistrali *vraćanje bajta podatka* i predstavlja slanje identifikatora uređaja kome se vraća očitani podatak.

Registar AR (*Address Register*) služi za čuvanje adrese u koju treba upisati bajt podatka u slučaju operacije upisa ili očitati bajt podatka u slučaju operacije čitanja. Na ulaze registra AR dovodi se sa adresnih linija magistrale **ABUS_{19...0}** 20-bitna adresa bajta podatka. Na adresnim linijama magistrale **ABUS_{19...0}** nalazi se adresa bajta podatka u slučaju ciklusa na magistrali zahtev za čitanje ili upis bajta podataka ili bloka podataka. Ova adresa se upisuje u registar AR pri pojavi signala takta **CLK** ukoliko je aktivan signal **ldAR**. Sadržaj registra AR se vodi u blok formiranje adrese u modulu.

Mikroprekidači konfiguracije modula **S_{2...0}** služe za određivanje načina raspoređivanja memorijskih adresa po memorijskim modulima. Usvojena je mogućnost formiranja pet konfiguracija memorijskih modula (slika 3.25). U zavisnosti od formirane konfiguracije modula od 20-bitne adrese formira se četvorobitni broj modula i 16-bitna adresa u modulu. U bloku *formiranje broja modula* na osnovu postavljene vrednosti mikroprekidača **S_{2...0}** od signala **ABUS_{19...16}** i **ABUS_{3...0}** sa adresnih linija magistrale formira se adresirani broj modula **ABM_{3...0}** (slika 3.26). Ovi signali se vode na ulaze komparatora CMP. U bloku *formiranje adrese u modulu* na osnovu postavljene vrednosti mikroprekidača **S_{2...0}**, od signala **AR_{19...0}** sa izlaza registra AR formira se adresa bajta u modulu **ARAM_{15...0}** (slika 3.27). Ova adresa se po linijama **ARAM_{15...0}** vodi u memoriju RAM.

konfiguracija	S ₂ S ₁ S ₀	objašnjenje	broj modula	adresa u modulu
0	000	susedne adrese u istom modulu	A ₁₉ A ₁₈ A ₁₇ A ₁₆	A ₁₅ A ₁₄ ... A ₀
1	001	kombinacija susedne adrese u istom i susednim modulima	A ₁₉ A ₁₈ A ₁₇ A ₀	A ₁₆ A ₁₅ ... A ₁
2	010	kombinacija susedne adrese u istom i susednim modulima	A ₁₉ A ₁₈ A ₁ A ₀	A ₁₇ A ₁₆ ... A ₂
3	011	kombinacija susedne adrese u istom i susednim modulima	A ₁₉ A ₂ A ₁ A ₀	A ₁₈ A ₁₇ ... A ₃
4	1xx	susedne adrese u susednim modulima	A ₃ A ₂ A ₁ A ₀	A ₁₉ A ₁₈ ... A ₄

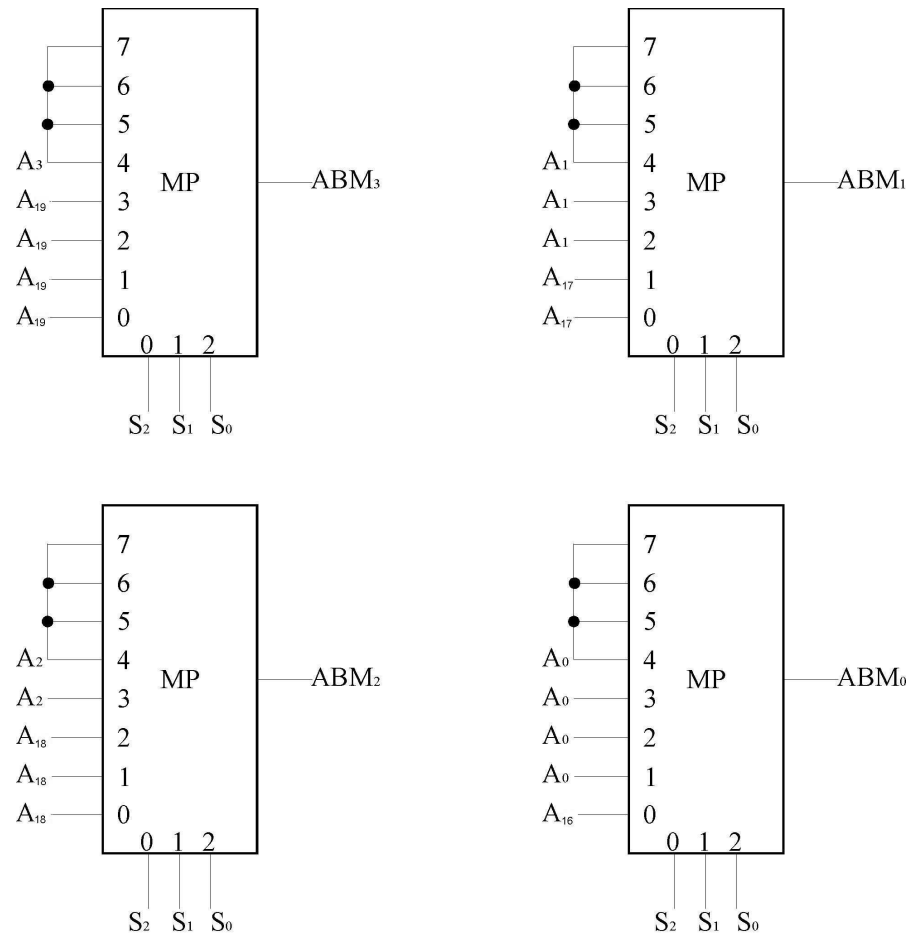
Slika 3.25 Formiranje konfiguracije modula

Komparator CMP služi za utvrđivanje da li se ciklus na magistrali u kome je memorijski modul *slave* odnosi na dati memorijski modul. Radi toga upoređuje se sadržaj mikroprekidača broja modula **BRM_{3...0}** sa sadržajem na linijama **ABM_{3...0}**, koje predstavljaju adresirani broj modula. U slučaju saglasnosti signal **EQ** dobija aktivnu vrednost.

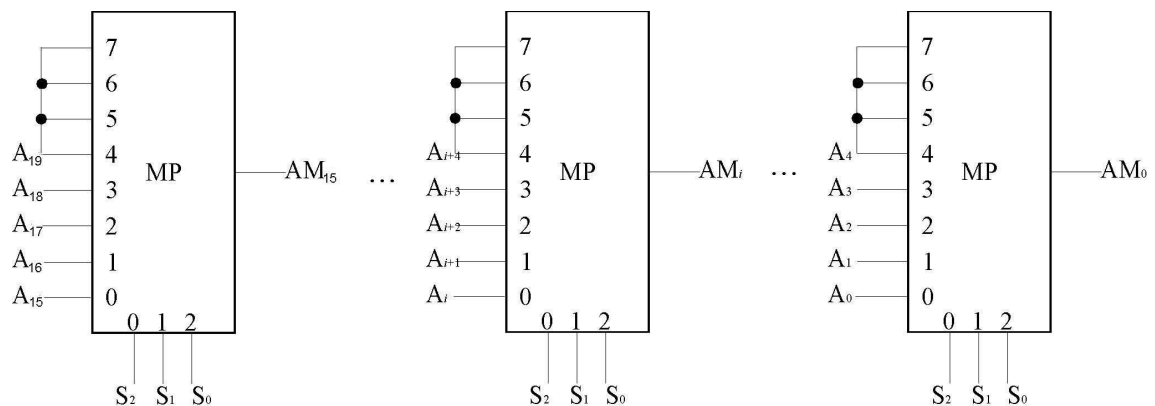
Memorija RAM služi za čuvanje podataka. Njen kapacitet je 2¹⁶ bajta. Sa memorijom RAM mogu se realizovati operacije upis bajta podatka i čitanje bajta podatka. U slučaju operacije upisa određenim sadržajem po linijama **ARAM_{15...0}** upisuje se sadržaj sa linija

DIRAM_{7...0} ukoliko je aktivna vrednost na liniji **WRRAM**. U slučaju operacije čitanja sa adrese određene sadržajem na linijama **ARAM_{15...0}** dobija se očitani podatak na linijama **DORAM_{7...0}** ukoliko je neaktivna vrednost na liniji **WRRAM**.

Brojač vremena pristupa RAMACC služi za generisanje aktivne vrednosti signala **ENDRAM** po završenoj operaciji upisa ili čitanja. Uzeto je da je vreme pristupa četiri periode signala takta **CLK**. Brojanje se omogućava aktivnom vrednošću ili signala **WRRAM** u slučaju operacije upisa ili aktivnom vrednošću signala **RDRAM** u slučaju operacije čitanja.



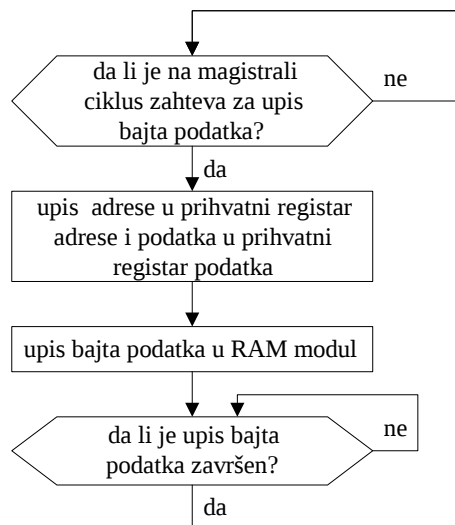
Slika 3.26 Formiranje broja modula



Slika 3.27 Formiranje adrese u modulu

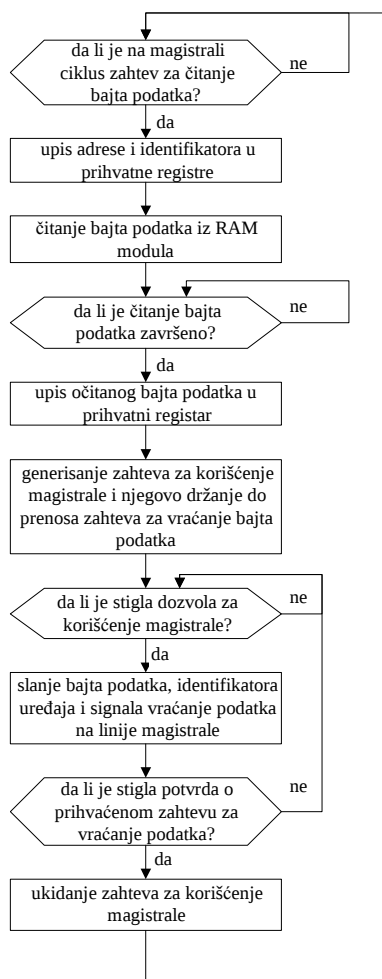
1.5.2 Upravljačka jedinica

Dijagram toka operacija za operaciju upisa dat je na slici 3.28 a za operaciju čitanja na slici 3.29.



Slika 3.28 Dijagram toka operacija za operaciju upisa bajta podatka

U slučaju operacije upisa bajta podatka upravljačka jedinica čeka da se na magistrali pojavi ciklus *zahtev za upis bajta podatka* namenjen datom memorijskom modulu. Pri pojavi ovog ciklusa adresa podatka se upisuje u prihvatni registar adrese AR i podatak u prihvatni registar podatka DIR. Potom se inicira operacija upisa bajta podatka u RAM modul. Iza toga se prelazi na čekanje indikacije da je upis bajta podatka završen. Po dobijanju ove indikacije vraća se na početak.



Slika 3.29 Dijagram toka operacija za operaciju čitanja bajta podatka

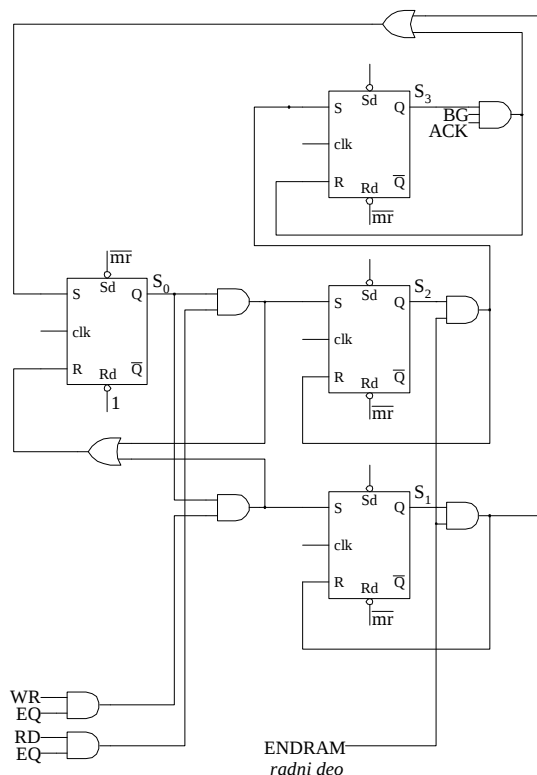
U slučaju operacije čitanja bajta podatka, upravljačka jedinica čeka da se na magistrali pojavi ciklus zahteva za čitanje bajta podatka namenjen datom memorijskom modulu. Pri pojavi ovog ciklusa adresa podatka se upisuje u prihvatni registar adrese AR, a identifikator uređaja u prihvatni registar podatka DIR. Potom se inicira operacija čitanja bajta podatka iz RAM modula i prelazi na čekanje indikacije da je čitanje bajta podatka završeno. Po dobijanju ove indikacije očitani podatak se iz radnog dela upisuje u prihvatni registar podatka DOR. Potom se generiše zahtev za korišćenje magistrale. Ovaj zahtev se drži sve dok se uspešno ne prihvati zahtev za *vraćanje bajta podatka*. Iza toga se prelazi na čekanje dozvole za korišćenje magistrale. Po dobijanju dozvole generiše se kompletan ciklus *vraćanje bajta podatka* uz slanje identifikatora uređaja i čeka potvrda da je uređaj bio slobodan i da je prihvatio zahtev za *vraćanje bajta podatka*. Ukoliko se takva potvrda ne dobije, ponavlja se već opisani postupak čekanja dozvole za korišćenje magistrale i generisanja ciklusa *vraćanje bajta podatka*. Kada se potvrda dobije ukida se zahtev za korišćenje magistrale i vraća na početak.

Na osnovu dijagrama toka operacija i strukture operacione jedinice, realizovana je upravljačka jedinica.

Upravljačka jedinica se sastoji iz dva dela:

- elementi za kašnjenje (slika 3.30) i
- povezivanje sa arbitratorom, upravljačkom magistralom i generisanje upravljačkih signala (slika 3.31).

Deo *elementi za kašnjenje* je realizovan na principu "šetajuće jedinice". Sastoji se od četiri flip-flopa S_0 , S_1 , S_2 i S_3 koji predstavljaju četiri stanja u kojima se može naći upravljačka jedinica. Prema principu "šetajuće jedinice" jedinica u datom trenutku može da se nađe samo u jednom flip-flopu dok su u ostalima nule. Na početku se upravljačka jedinica nalazi u stanju S_0 , pa je jedinica u flip-flopu S_0 .

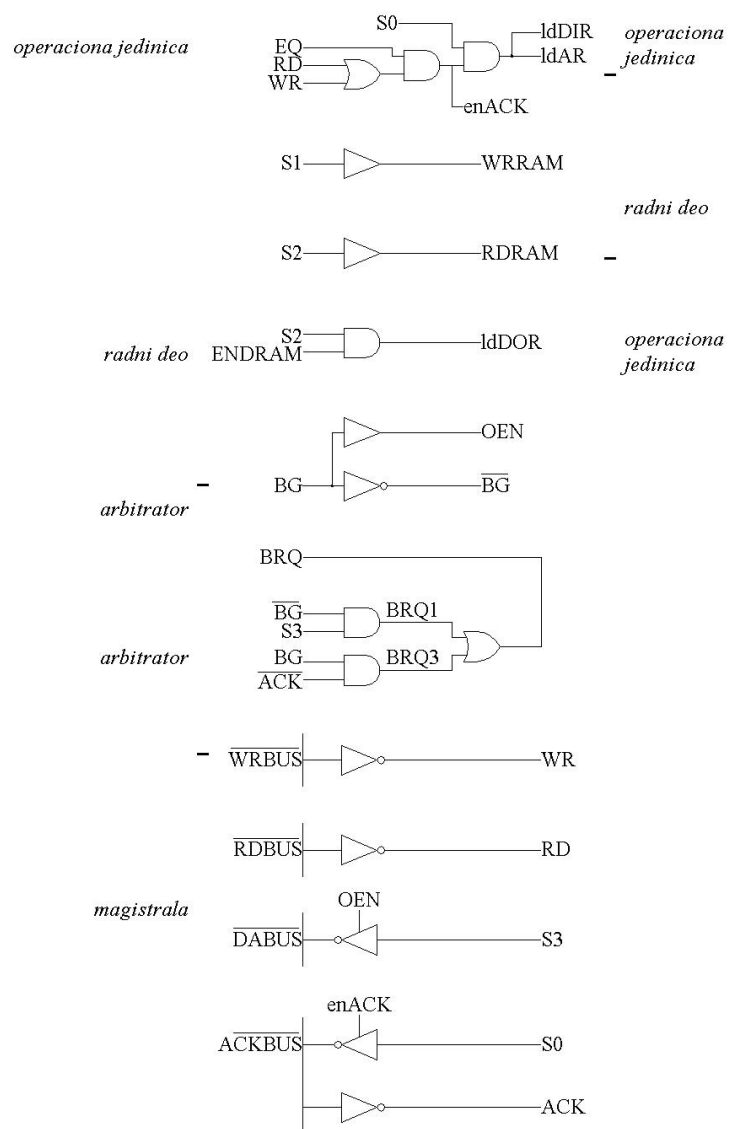


Slika 3.30 Elementi za kašnjenje

Kod operacije upisa upravljačka jedinica prelazi iz stanja S_0 u S_1 i vraća se u stanje S_0 . Stoga jedinica šeta iz flip-flopa S_0 u S_1 i vraća se u flip-flop S_0 .

Kod operacije čitanja, najpre se na magistrali realizuje ciklus čitanja prelaskom iz stanja S_0 u S_2 . Stoga jedinica sada šeta iz flip-flopa S_0 u S_2 . U stanju S_2 se čita bajt podatka iz memorije RAM. Pošto se očitani bajt podatka smesti u prihvatni registar podatka u stanju S_2 , prelazi se na stanje S_3 . Stoga jedinica sada šeta iz flip-flopa S_2 u S_3 . U stanju S_3 se na magistrali realizuje ciklus *vraćanje bajta podatka* i potom vraća na stanje S_0 . Stoga jedinica sada šeta iz flip-flopa S_3 u S_0 .

Jedinica prelazi iz flip-flopa S_0 u S_1 pri pojavi signala za upis **WR** i aktivnoj vrednosti signala saglasnosti **EQ**. Jedinica prelazi iz flip-flopa S_0 u S_2 pri pojavi signala za čitanje **RD** i aktivnoj vrednosti signala saglasnosti **EQ**. Jedinica prelazi iz flip-flopa S_2 u S_3 ako je očitani bajt podatka što se indicira aktivnom vrednošću signala **ENDRAM**. Jedinica se vraća iz flip-flopa S_1 u S_0 po uspešno realizovanoj operaciji upisa bajta podatka. Jedinica se vraća iz flip-flopa S_3 u S_0 po uspešno generisanom ciklusu na magistrali *vraćanje bajta podatka*. To se indicira aktivnim vrednostima signala **BG** i **ACK**.



Slika 3.31 Povezivanje sa arbitratorom, upravljačkom magistralom i generisanje upravljačkih signala