
Specifikacija PCIw (Peripheral Component Interconnect Wannabe) magistrale

Lazar Premović
pl190091@student.etf.bg.ac.rs

April 2021.
v0.9

Rezime

PCIw je magistrala za komunikaciju sa periferijskim uredjajima u racunarskom sistemu, čiji je cilj da pored rasterećivanja glavne magistrale omogući visoku propusnost i blisku integraciju sa sistemom prekida.

Da bi ispunila te ciljeve PCIw je hibridna magistrala (operacije se izvršavaju na signal takta, čekanje na operaciju je proizvoljno) sa atomskim ciklusima i bez arbitracije (jedan uredjaj je uvek u ulozi gazde).

Ovaj dokument sadrži specifikaciju PCIw magistrale, referentnu implementaciju, razmatranja njene upotrebe u različitim situacijama i opcione dodatke specifikaciji ili potencijalne izmene u odnosu na referentnu implementaciju radi boljeg funkcionisanja u određenim slučajevima.

Sadržaj

| | | |
|----------|---|-----------|
| 1 | Specifikacija | 2 |
| 1.1 | Linije za podatke i adresiranje | 2 |
| 1.2 | Linije za signalizaciju | 3 |
| 1.3 | Ciklusi na magistrali | 4 |
| 2 | Referentna implementacija | 6 |
| 2.1 | Slanje i prijem podataka | 7 |
| 2.2 | Signalizacija | 7 |
| 2.3 | Interfejs prema uredjajima | 8 |
| 2.4 | Povezivanje sa sistemom prekida | 10 |
| 3 | Primeri upotrebe | 10 |

1 Specifikacija

PCIw je magistrala namenjena za komunikaciju izmedju procesora (ili nekog drugog uredjaja u ulozi gazde)¹ i periferijskih uredjaja².

Sve operacije na magistrali se odvijaju izmedju procesora i periferijskog uredjaja čija je adresa specifikovana adresnim linijama, tako da komunikacija izmedju dva periferijska uredjaja bez posredstva procesora nije moguća.

Zato što svaki ciklus na magistrali uključuje procesor nije potrebno imati poseban sistem arbitracije već je procesor uvek vlasnik magistrale i šalje podatke uredjajima ili proziva uredjaje da mu pošalju podatak. Ukoliko uredjaj želi da obavesti procesor da ima podatak koji je potrebno da pošalje to može uraditi posredstvom sistema prekida, upravo iz tog razloga je sama magistrala veoma usko vezana za sistem prekida.

Ciklusi na magistrali su atomski a sama magistrala je hibridna, to znači da se sami ciklusi obavljaju na signal takta ali izmedju procesorovog zahteva za izvršavanje određenog ciklusa i samog tog ciklusa može proći proizvoljan broj taktova.

Iako su ciklusi na magistrali atomski (što znači da je magistrala zauzeta svo vreme od trenutka zahteva za određen ciklus do trenutka izvršavanja tog ciklusa) moguće je prekinuti ciklus u bilo kom trenutku izmedju slanja zahteva i samog izvršavanja (što omogućava svojevrsno vremensko multipleksiranje ciklusa, ali ono nije deo specifikacije stoga se ovaj dokument neće baviti njime).

Magistrala radi u polu-dupleks režimu rada, iako je uz male modifikacije specifikacije moguće omogućiti rad u pun-dupleks režimu procenjeno je da su za veliku većinu slučajeva (komunikacija malog broja periferija sa relativno malom količinom podataka koje treba preneti u jedinici vremena) benefiti pun-dupleks režima znatno manji od dodatne cene i kompleksnosti koju on donosi sa sobom.

1.1 Linije za podatke i adresiranje

Magistrala ima jednu liniju proizvoljne širine za prenos podataka **DBUS** i jednu liniju proizvoljne širine za adresiranje uredjaja **ABUS**.

Pri povezivanju za prijem podataka moguće je prosto dovesti DBUS liniju na ulaz registra ukoliko se upis u registar vrši samo u taktu u kome se izvršava ciklus prijema.

Pri povezivanju za slanje podataka neophotno je izlaz registra dovesti na trostatički bafer koji će preneti podatke samo u taktu u kome se izvršava ciklus slanja a u suprotom će održavati stanje visoke impedanse.

Na liniju za adresiranje uredjaja jedino procesor dovodi vrednosti tako da na ABUS liniji nije potrebno nikakvo trostatičko baferisanje i uredjaji mogu slobodno da koriste vrednosti sa te linije u bilo kom trenutku.

Iako je moguće koristiti neke od bitova ABUS linije za specifikovanje ponašanja periferijskog uredjaja (npr. u koji registar treba upisati podatke) preporučuje se da se za to ipak koriste bitovi DBUS linije.

Iz ovoga se zaključuje da je podrazumevano stanje DBUS linije stanje visoke impedanse "**HiZ**" a da linija ABUS nema podrazumevano stanje već uvek adresira neki uredjaj.

¹ U daljem tekstu samo procesor

² U daljem tekstu samo uredjaj

1.2 Linije za signalizaciju

Magistrala ima 3 obavezne i 2 opcione linije za signalizaciju.

- **TX** označava zahtev za ciklus slanja podataka perifernom uređaju.
- **RX** označava zahtev za ciklus prijema podataka od perifernog uređaja.
- **FC** označava da će se ciklus obaviti na sledeću uzlaznu ivicu takta.
- **HIT** *opciono* označava da postoji uređaj na adresi specifikovanom linijom ABUS.
- **CLK** *opciono* takt magistrale.

Linija TX je obavezna linija kojom upravlja procesor i ima aktivnu vrednost u logičkoj jedinici. Aktivnom vrednošću ove linije procesor signalizira trenutno adresiranom uređaju da želi da izvrši ciklus slanja podataka perifernom uređaju. Procesor treba da drži ovu liniju aktivnom sve dok se ciklus ne izvrši tj. kada procesor detektuje aktivnu vrednost linije FC u sedećem taktu spušta liniju TX na neaktivnu vrednost. Ukoliko procesor želi da odustane od ciklusa upisa (zbog isteka vremena čekanja ili promene konteksta) može bilo kada slobodno spustiti liniju TX na neaktivnu vrednost. Kako samo procesor upravlja ovom linijom nije potrebno primenjivati trostatičko baferisanje.

Linija RX je obavezna linija kojom upravlja procesor i ima aktivnu vrednost u logičkoj jedinici. Aktivnom vrednošću ove linije procesor signalizira trenutno adresiranom uređaju da želi da izvrši ciklus prijema podataka od perifernog uređaja. Procesor može da drži ovu liniju aktivnom sve dok se ciklus ne izvrši³ (blokirajući prijem) ili može držati liniju aktivnom samo jedan takt⁴ i na osnovu vrednosti linije FC može odrediti da li je zapravo primio podatak ili ne (ovo je ekvivalentno proveriti da li postoje podaci spremni za prijem i ukoliko postoje prijemu jednog podatka). Ukoliko procesor želi da odustane od ciklusa prijema (zbog isteka vremena čekanja ili promene konteksta) može bilo kada slobodno spustiti liniju RX na neaktivnu vrednost. Kako samo procesor upravlja ovom linijom nije potrebno primenjivati trostatičko baferisanje.

Linija FC je obavezna linija kojom upravlja adresirani uređaj i ima aktivnu vrednost u logičkoj jedinici⁵. Aktivnom vrednošću ove linije adresirani uređaj signalizira procesoru da je spreman da izvrši ciklus koji je procesor zahtevao (ukoliko je procesor zahtevao ciklus slanja aktivnom vrednošću linije FC uređaj signalizira da je spreman da primi podatak a ukoliko je procesor zahtevao ciklus prijema aktivnom vrednošću linije FC uređaj signalizira da je spreman da pošalje podatak). Tek nakon što su se procesor i adresirani uređaj usaglasili (procesor je zahtevao ciklus i adresirani uređaj je signalizirao da je spreman da izvrši traženi ciklus) mogu izvršiti ciklus na magistrali, taj ciklus će se izvršiti na prvu

³ Videti Liniju TX

⁴ Kada se kaže takt misli se na takt magistrale ukoliko uređaji imaju taktove različite brzine

⁵ Kako linija FC može biti i u stanju visoke impedanse, ukoliko procesor ne razlikuje stanje visoke impedanse od stanja logičke jedinice, uz minimalne promene implementacije linija FC može biti aktivna u logičkoj nuli

sledeću uzlaznu ivicu takta i magistrala je odmah nakon toga slobodna za izvršenje sledećeg ciklusa. Kako linijom FC može upravljati bilo koji uređaj u zavisnosti od toga koji uređaj je adresiran, neophodno je koristiti trostatičko baferisanje za ovu liniju. Trostatičko baferisanje se vrši tako što samo uređaj koji je adresiran otvara svoj trostatički bafer prema liniji FC. Zahvaljujući tome dodatno je moguće detektovati ispravnost adrese koju procesor dovodi na ABUS liniju, ukoliko procesor pokuša da zahteva ciklus dok je linija FC u stanju visoke impedanse to znači da na toj adresi ne postoji uređaj i takvu grešku treba pravilno obraditi⁶

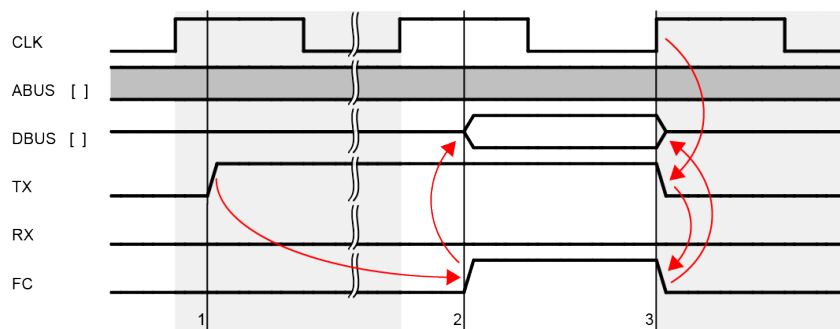
Linija HIT je opcionalna linija kojom upravlja adresirani uređaj i ima aktivnu vrednost u logičkoj nuli⁷. Svrha ove linije je da detektuje pokušaj adresiranja nepostojećeg uređaja i kao takva aktivnom vrednošću ove linije adresirani uređaj signalizira procesoru da je adresa ispravna. Ukoliko procesor pokuša da zahteva ciklus na magistrali a linija HIT nema aktivnu vrednost znači da je došlo do adresiranja nepostojećeg uređaja i takvu grešku treba pravilno obraditi. Kako linijom HIT može upravljati bilo koji uređaj u zavisnosti od toga koji uređaj je adresiran, neophodno je koristiti trostatičko baferisanje za ovu liniju. Trostatičko baferisanje se vrši tako što samo uređaj koji je adresiran otvara svoj trostatički bafer i postavlja aktivnu vrednost na liniju HIT.

Linija CLK predstavlja takt magistrale i može se izostaviti ukoliko svi uređaji na magistrali već rade na isti signal takta.

1.3 Ciklusi na magistrali

Vremenski dijagrami ciklusa na magistrali pretpostavljaju da uređaji mogu imati takt brži od takta magistrale i da je kašnjenje kombinacione logike zanemarljivo

1.3.1 Ciklus slanja podatka periferiji



Slika 1: vremenski dijagram ciklusa slanja podatka periferiji

Pri slanju podataka periferiji u koraku **1** procesor stavlja aktivnu vrednost na liniju TX i time obaveštava uređaj čija se adresa nalazi na liniji ABUS da želi da mu pošalje podatke.

Nakon proizvoljnog vremena u koraku **2** adresirani uređaj obaveštava procesor da je spreman da primi podatke postavljanjem aktivne vrednosti

⁶ Procesori bez mogućnosti da detektuju stanje visoke impedanse za ovu svrhu mogu koristiti liniju HIT

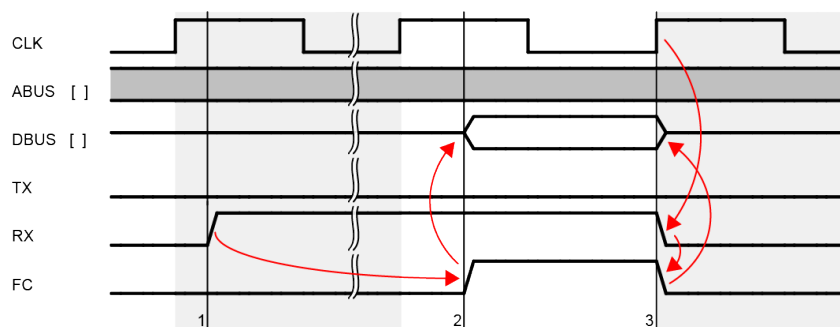
⁷ Ukoliko procesor razlikuje stanje visoke impedanse od stanja logičke jedinice, moguće je i da linija HIT ima aktivnu vrednost u logičkoj jedinici

na liniju FC, kombinacionom logikom kada detektuje aktivnu vrednost linije FC procesor otvara svoj trostatički bafer na liniji DBUS.

Nakon što su uredjaji sinhronizovani i podaci su postavljeni na liniju DBUS, potrebno je sačekati uzlaznu ivicu takta magistrale.

Na uzlaznu ivicu takta magistrale (**3**) podaci sa DBUS linije se upisuju u prihvatni registar uredjaja i na istu ivicu je zbog kašnjenja kombinacione logike moguće ukloniti aktivnu vrednost sa linije TX. Uklanjanje aktivne vrednosti TX kombinacionom logikom uklanja aktivnu vrednost sa linije FC što opet kombinacionom logikom zatvara trostatički bafer procesora na liniji DBUS, time je magistrala vraćena u početno stanje i magistrala je spremna za sledeći ciklus koji se može izvršiti već na sledeću uzlaznu ivicu takta.

1.3.2 Ciklus prijema podatka od periferije



Slika 2: vremenski dijagram ciklusa prijema podatka od periferije

Pri prijemu podataka od periferije u koraku **1** procesor stavlja aktivnu vrednost na liniju RX i time obaveštava uredjaj čija se adresa nalazi na liniji ABUS da želi da primi podatke od njega.

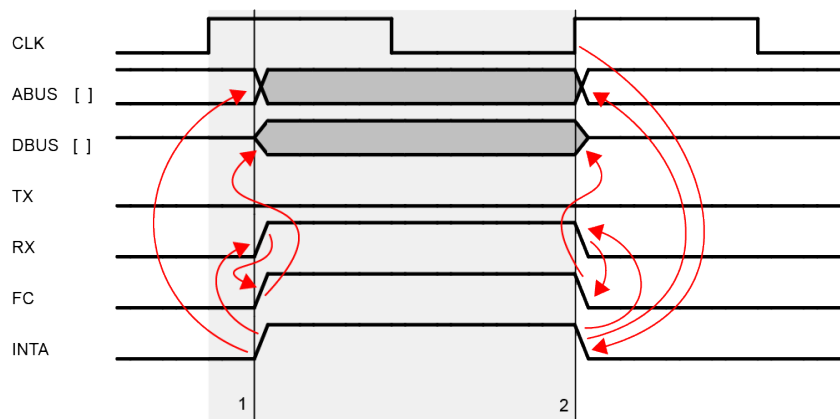
Nakon proizvoljnog vremena u koraku **2** adresirani uredjaj obaveštava procesor da je spreman da mu pošalje podatke postavljanjem aktivne vrednosti na liniju FC, kombinacionom logikom kada detektuje aktivnu vrednost linije FC uredjaj otvara svoj trostatički bafer na liniji DBUS. Nakon što su uredjaji sinhronizovani i podaci su postavljeni na liniju DBUS, potrebno je sačekati uzlaznu ivicu takta magistrale.

Na uzlaznu ivicu takta magistrale (**3**) podaci sa DBUS linije se upisuju u prihvatni registar procesora i na istu ivicu je zbog kašnjenja kombinacione logike moguće ukloniti aktivnu vrednost sa linije RX. Uklanjanje aktivne vrednosti RX kombinacionom logikom uklanja aktivnu vrednost sa linije FC što opet kombinacionom logikom zatvara trostatički bafer uredjaja na liniji DBUS, time je magistrala vraćena u početno stanje i magistrala je spremna za sledeći ciklus koji se može izvršiti već na sledeću uzlaznu ivicu takta.

1.3.3 Ciklus prijema podatka pri prekidu

Za ovaj ciklus se pretpostavlja da će uredjaj koji zahteva prekid biti spreman da pošalje podatke od trenutka kada zahteva prekid i da signal INTA traje bar do uzlazne ivice takta magistrale.

Ovaj ciklus je sličan ciklusu prijema podataka od periferije te će biti diskutovane samo razlike izmendlju dva ciklusa.



Slika 3: vremenski dijagram ciklusa slanja podatka periferiji

U koraku **1** umesto da procesor postavlja aktivnu vrednost na liniju RX ona se podstavlja kombinacionom logikom na aktivnu vrednost signala INTA, signal INTA takodje na liniju ABUS postavlja adresu uredjaja koji je zatražio prekid.

Kako je uredjaj koji je zahtevao prekid odmah spreman da pošalje podatke korak 2 ciklusa prijema podataka se dešava u koraku **1** nakon određenog kombinacionog kašnjenja, što znači da u trenutku **1** i uredjaj postavlja aktivnu vrednost na liniju FC i kombinaciona logika ubrzo nakon toga otvara trostatički bafer uredjaja ka liniji DBUS. Nakon što su uredjaji sinhronizovani i podaci su postavljeni na liniju DBUS, potrebno je sačekati uzlaznu ivicu takta magistrale.

Na uzlaznu ivicu takta magistrale (**2**) podatci sa DBUS linije se upisuju u prihvatni registar procesora i na istu ivicu je zbog kašnjenja kombinacione logike moguće ukloniti aktivnu vrednost sa linije INTA koja kombinacionom logikom uklanja aktivnu vrednost sa linije RX i na liniju ABUS vraća adresu iz registra za adresu procesora. Uklanjanje aktivne vrednosti RX kombinacionom logikom uklanja aktivnu vrednost sa linije FC što opet kombinacionom logikom zatvara trostatički bafer uredjaja na liniji DBUS, time je magistrala vraćena u početno stanje i magistrala je spremna za sledeći ciklus koji se može izvršiti već na sledeću uzlaznu ivicu takta.

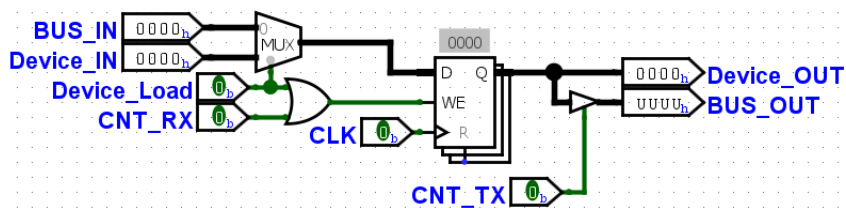
2 Referentna implementacija

Kako je implementacija magistrale samo skup linija, u ovoj sekciji biće prikazana i referentna implementacija kontrolera kojima se uredjaji i procesor povezuju na magistralu. Kontroleri sadrže svu kombinacionu logiku potrebnu za izvršavanje ciklusa na magistrali, kao i logiku koja procesoru i uredjajima olakšava upotrebu magistrale.

Referentna implementacija prepostavlja da svi uredjaji rade na isti takt kao i magistrala i da je pin **CLK** uvek povezan na taj zajednički signal takta.

2.1 Slanje i prijem podataka

Deo za slanje i prijem podataka se sastoji od jednog registra zaduženog za prijem i za slanje podataka⁸, trostatičkog bafera za povezivanje na liniju DBUS i logike za učitavanje u registar.



Slika 4: Deo za slanje i prijem podataka

Pinovi **BUS_IN** i **BUS_OUT** povezuju kontroler na DBUS liniju magistrale⁹.

Pinovima **Device_IN** i **Device_OUT** uređaji upisuju ili čitaju vrednosti iz registra za prijem i slanje podataka.

Signalom **Device_Load** se na ulaz registra kroz multiplexer dovodi signal **Device_IN** i omogućava se njegov upis u registar.

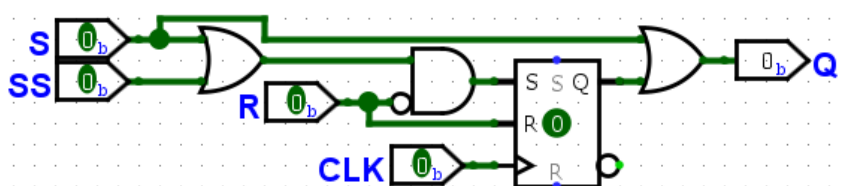
Signal **CNT_TX** otvara trostatički bafer prema DBUS liniji magistrale a signal **CNT_RX** omogućava upis u registar i propušta vrednost sa DBUS linije magistrale na ulaz registra, ovim signalima upravlja deo kontrolera zadužen za signalizaciju.

2.2 Singalizacija

Deo za signalizaciju sdrži interfejs prema uređaju koji se povezuje na magistralu, upravlja linijama za signalizaciju magistrale i upravlja delom za slanje i prijem podataka.

2.2.1 Fast set flip flop

Fast set flip flop je komponenta kojom deo za singalizaciju pamti zahteve za ciklus na magistrali.



Slika 5: Fast set flip flop

Pinovi **SS** (slow set), **R** i **Q** imaju istu funkcionalnost kao i kod standardnog RS flip fropa sa prioritetom na reset ulazu. Pin **SS** osim što se ponaša kao S ulaz flip fropa takodje postavlja izlaz na aktivnu vrednost bez obzira na stanje flip fropa.

⁸ Sistemi sa odvojenim registrima za prijem i slanje su mogući ali se oni neće razmatrati ovde jer ne donose znatno poboljšanje performansi

⁹ Ulazni i izlazni pinovi su razdvojeni zbog ne postojanja bi-direkcionog pina u alatu u kom je dizajnirana referentna implementacija, ali to ne utiče na funkcionalnost kontrolera.

2.3 Interfejs prema uredjajima

Interfejs prema uredjajima je većinski isti i za gazdu i za sluge na magistrali se sastoji od 7 ulaznih i 3 izlazne linije.

Kada kažemo da sluga zahteva ciklus na magistrali to samo znači da je spreman da izvrši taj ciklus ukoliko procesor to zahteva od njega.

Sve linije imaju aktivne vrednosti u logičkoj jedinici osim linije HIT.

- **TX** Aktivnom vrednošću ove linije u trajanju od jedne periode takta se zahteva ciklus slanja¹⁰, zahtev će biti aktivan od takta u kome se zahteva sve dok se zahtev ne ispuni ili se ne prekine.
- **SLOW_TX** Aktivnom vrednošću ove linije u trajanju od jedne periode takta se zahteva ciklus slanja¹⁰, zahtev će biti aktivan od sledeceg takta pa sve dok se zahtev ne ispuni ili se ne prekine. Ovakva linija je poželjna ukoliko je potrebno zahtevati ciklus i u istom taktu učitati vrednost u registar za slanje.
- **BURST_TX** Aktivnom vrednošću ove linije zahteva se ciklus slanja¹⁰, zahtev će biti aktivan dokle god linija ima aktivnu vrednost.
- **RX** Aktivnom vrednošću ove linije u trajanju od jedne periode takta se zahteva ciklus prijema¹¹, zahtev će biti aktivan od takta u kome se zahteva sve dok se zahtev ne ispuni ili se ne prekine.¹²
- **BURST_RX** Aktivnom vrednošću ove linije zahteva se ciklus prijema¹¹, zahtev će biti aktivan dokle god linija ima aktivnu vrednost.
- **TIMEOUT** Aktivnom vrednošću ove linije u trajanju od jedne periode takta se prekida trenutni zahtev za ciklus na magistrali.
- **FC** Izlazna linija čija aktivna vrednost označava da će se na sledeću uzlaznu ivicu izvršiti zahtevani ciklus.
- **BB** Izlazna linija čija aktivna vrednost označava da trenutno postoji zahtev za ciklus na magistrali.
- **HIT** Opciona izlazna linija koja je aktivna u nuli i čija aktivna vrednost označava da postoji uredjaj na traženoj adresi.

2.3.1 Deo za signalizaciju gazde na magistrali

Pored dva Fast set flip flop-a za pamćenje zahteva za slanje i prijem, deo za signalizaciju sadrži i kombinacionu logiku koja formira potrebne signale.

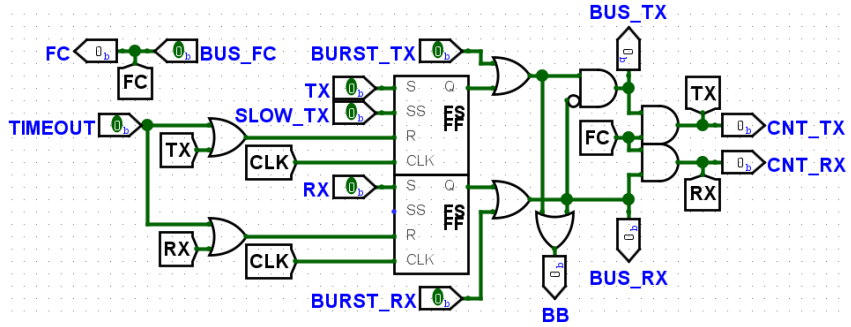
Linija FC se sa magistrale direktno prosledjuje povezanom uredjaju ali je prikazana kao deo kontrolera jer se njena vrednost koristi u logici kontrolera.

Zahtevi se brišu ili signalom timeout ili signalom koji govori delu za slanje i prijem podataka da pošalje ili primi podatak, ti signali će biti aktivni neposredno pre uzlazne ivice na koju će se preneti podatak. Signal BB (*bus busy*) je aktivan ukoliko postoji aktivan zahtev za slanje ili prijem.

¹⁰ Slanja podataka uredjaju ukoliko je u pitanju kontroler gazde ili slanja podataka procesoru tj. ciklus prijema podataka ukoliko je u pitanju kontroler periferije

¹¹ Prijema podataka od uredjaja ukoliko je u pitanju kontroler gazde ili prijema podataka od procesora tj. ciklus slanja podataka ukoliko je u pitanju kontroler periferije

¹² Linija **SLOW_RX** je izostavljena jer nema mnogo smisla ali ju je moguće dodati ukoliko je potrebna

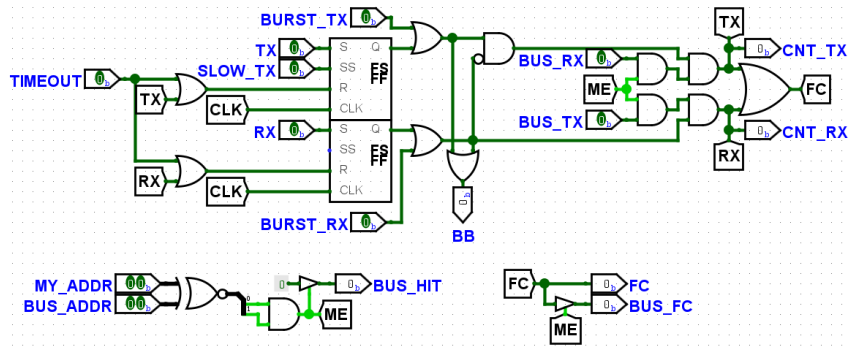


Slika 6: Deo za signalizaciju gazde na magistrali

Signali BUS_TX i BUS_RX koji se dovode direktno na signalne linije TX i RX magistrale imaju aktivne vrednosti ukoliko postoji zahtev za slanje ili prijem respektivno (ukoliko postoje zahtevi za oba ciklusa, ciklus prijema ima prioritet)¹³.

Ukoliko je adresirani uredjaj postavio aktivnu vrednost linije FC u zavisnosti od zahtevanog ciklusa delu za slanje i prijem će biti dat signal da na sledeću uzlaznu ivicu takta izvrši potrebnu operaciju.

2.3.2 Deo za signalizaciju sluge na magistrali



Slika 7: Deo za signalizaciju sluge na magistrali

Kod sluge na magistrali deo za signalizaciju pre svega formira signal ME koji ima aktivnu vrednost ukoliko su adresa uredjaja i vrednost na liniji ABUS magistrale jednake, aktivna vrednost signala me govori uredjaju da se procesor njemu obraća.

Ukoliko magistrala sadrži liniju HIT onda se ukoliko signal ME ima aktivnu vrednost kroz trostatički bafer na liniju HIT postavlja aktivna vrednost.

Slično kao kod kontrolera gazde signali za brisanje zahteva se formiraju od signala TIMEOUT i signala koji upravljaju delom za slanje i prijem podataka.

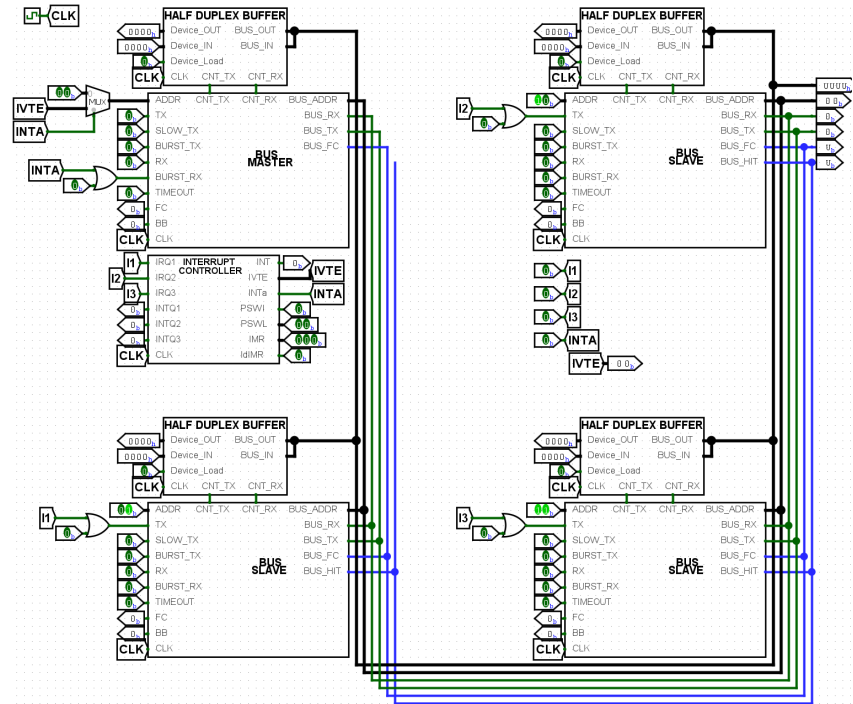
Signal BB se formira isto kao kod kontrolera gazde, isto važi i za prioritet operacija¹³.

Signali koji upravljaju delom za slanje i prijem se formiraju ukoliko su aktivni signali ME i odgovarajući par zahteva uredjaja i procesora (zahtev za slanje procesora i zahtev za prijem uredjaja i obratno).

¹³ Iako postoji prioritet zahteva, nikada ne bi trebalo istovremeno imati aktivne zahteve za oba ciklusa, pre izdavanja zahteva za ciklus potrebno je da uredjaj povezan na kontroler proveri da li je neki zahtev već na čekanju signalom BB

Ukoliko je neki od signala za upravljanje delom za slanje i prijem aktivan formira se i signal FC kojim se daje do znanja procesoru da će se na sledeću uzlaznu ivicu takta izvršiti ciklus na magistrali, signal FC se direktno prosledjuje uređaju a preko trostatičkog bafera koji je aktivan samo ukoliko je signal ME aktivan se dovodi na FC liniju magistrale.

2.4 Povezivanje sa sistemom prekida



Slika 8: Magistrala integrisana sa sistemom prekida

Pri integraciji PCIw magistrale sa sistemom prekida potrebno je voditi računa o pet stvari.

Adresa uređaja mora biti ista kao broj linije zahteva za prekid koji mu pripada.

Ukoliko uređaj koji želi da zatraži prekid želi i da pošalje podatke preko magistrale, neophodno je da pre zahtevanja prekida učita u registar za slanje podatke koje želi poslati.

Neophodno je povezati signal zahteva prekida tako da njegova aktivna vrednost takodje aktivira i TX signal tog uređaja.

Neophodno je povezati signal prihvatanja prekida tako da njegova aktivna vrednost takodje aktivira i BURST_RX signal procesora.

Neophodno je da se pri prihvatanju prekida multiplexerom propusti broj linije prekida koji se prihvata na liniju ABUS.

3 Primeri upotrebe

U ovoj sekciji su prikazani primeri nekih standardnih operacija na magistrali i kako se one realizuju kao mikrokod¹⁴. Kombinacijom ovih operacija moguće vrlo lako ostvariti kompleksnije operacije.

Kako su interfejsi za procesor i periferijske uređaje identični, dati primeri su primenljivi i za procesor i za periferijske uređaje

¹⁴ Sintaksa mikrokoda se nalazi na <http://lazar2222.github.io/ETF/Zolakod.pdf> (uskoro)

3.0.1 Slanje sa čekanjem

Za operaciju slanja podataka na magistrali koja čeka da se podaci prenesu pre nastavljanja sa izvršavanjem najlakše je koristiti signal BURST_TX i čekati da se pojavi aktivna vrednost linije FC. Pretpostavlja se da se u registrima za adresu i podatak već nalaze potrebne vrednosti.

Primer 1: Mikrokod za slanje sa čekanjem

```
//Ucitavanje vrednosti registara  
BURST_TX brif !FC then this  
//Slanje obavljeno
```

3.0.2 Slanje sa čekanjem i upisom podatka u istom taktu

Ova operacija je slična slanju sa čekanjem ali može da uštedi jedan takt ukoliko je potrebno učitati vrednost u registar za adresu i/ili u registar za podatak. Stoga se za realizaciju koristi signal SLOW_TX.

Primer 2: Mikrokod za slanje sa čekanjem i upisom podatka u istom taktu

```
Device_Load , SLOW_TX brif !FC then this  
//Slanje obavljeno
```

3.0.3 Slanje bez čekanja

Ova operacija šalje zahtev za slanje podataka i nastavlja sa izvršavanjem mikrokoda bez da čeka da se ciklus na magistrali izvrši, pre slanja zahteva potrebno je proveriti da li je magistrala zauzeta (poželjno operacijom čekanja na izvršenu operaciju). Samo slanje zahteva se obavlja postavljanjem linije TX na aktivnu vrednost u trajanju od jednog takta. Pretpostavlja se da se u registrima za adresu i podatak već nalaze potrebne vrednosti.

Ukoliko je potrebno upisati podatak u istom taktku u kom se šalje zahtev moguće je koristiti liniju SLOW_TX na isti način kao i liniju TX.

Primer 3: Mikrokod za slanje bez čekanja

```
//Ucitavanje vrednosti registara  
TX
```

3.0.4 Slanje sa upitom

Kod slanja sa upitom pokušavamo da pošaljemo podatak jedan takt i obaveštavamo uređaj da li je podatak poslat. Za slanje sa upitom koristimo signal BURST_TX a linija FC nam govori da li je slanje uspešno. Pretpostavlja se da se u registrima za adresu i podatak već nalaze potrebne vrednosti.

Primer 4: Mikrokod za slanje sa upitom

```
//Ucitavanje vrednosti registara  
BURST_TX brif !FC then NeuspesnoSlanje  
//Uspesno slanje
```

3.0.5 Prijem sa čekanjem

Za operaciju preijema podataka sa magistrale koja čeka da se podaci prenesu pre nastavljajnja sa izvršavanjem najlakše je koristiti signal BURST_RX i čekati da se pojavi aktivna vrednost linije FC. Pretpostavlja se da se u registru za adresu već nalazi potrebna vrednost.

Primer 5: Mikrokod za prijem sa čekanjem

```
//Ucitavanje vrednosti registra
BURST_RX brif !FC then this
//Prijem obavljen
```

3.0.6 Prijem bez čekanja

Ova operacija šalje zahtev za prijem podataka i nastavlja sa izvršavanjem mikrokoda bez da čeka da se ciklus na magistrali izvrši, pre slanja zahteva potrebno je proveriti da li je magistrala zauzeta (poželjno operacijom čekanja na izvršenu operaciju). Samo slanje zahteva se obavlja postavljanjem linije RX na aktivnu vrednost u trajanju od jednog takta. Pretpostavlja se da se u registriu za adresu već nalazi potrebna vrednost.

Primer 6: Mikrokod za prijem bez čekanja

```
//Ucitavanje vrednosti registra
RX
```

3.0.7 Prijem sa upitom

Kod prijema sa upitom pokušavamo da primimo podatak jedan takt i obaveštavamo uređaj da li je podatak primljen. Za prijem sa upitom koristimo signal BURST_RX a linija FC nam govori da li je prijem uspešan. Pretpostavlja se da se u registru za adresu već nalazi potrebna vrednost.

Primer 7: Mikrokod za prijem sa upitom

```
//Ucitavanje vrednosti registra
BRUST_RX brif !FC then NeuspesanPrijem
//Uspesan Prijem
```

3.0.8 Čekanje na izvršenu operaciju

Ukoliko se koriste operacije bez čekanja, pre slanja novog zahteva ili pre obrade podataka koji su zahtevani potrebno je sačekati da se zahtev na čekanju izvrši. Za proveru da li postoji zahtev na čekanju se koristi signal BB čje se performanse mogu poboljšati kombinacijom sa signalom FC.

Primer 8: Mikrokod za čekanje na izvršenu operaciju

```
brif BB && !FC then this
//Zahtev izvršen
```

3.0.9 Primer prenosa velike količine podataka

Ovde su dati primeri mikrokoda za uredjaj koji šalje i uredjaj koji prima podatke tako da se nakon režijskih troškova prenoji po jedna reč svakog takta.

Primer 9: Mikrokod uredjaja koji šalje podatke

```
zero: brif BB then this  
LDTX  
TX, if FC then LDTX brif FC then this else zero
```

Primer 10: Mikrokod uredjaja koji prima podatke

```
zero: BURST_RX brif !FC then this  
BURST_RX, LDRX brif FC then this else zero
```

Signal LDTX učitava sledeći podatak u registar za slanje a signal LDRX učitava podatak iz registra za prijem gde je potrebno.