

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Štajdohar

PRIMERJAVA STANDARDOV USB 2.0 IN USB 3.0

DIPLOMSKO DELO NA VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentor: pred. mag. Igor Škraba

Ljubljana, 2011

Št. naloge: 00536/2010

Datum: 03.12.2010



Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGOR ŠTAJDOHAR**

Naslov: **PRIMERJAVA STANDARDOV USB 2.0 IN USB 3.0**
COMPARISON OF USB 2.0 AND USB 3.0 STANDARDS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

V nalogi opišite arhitekturo povezave USB 2.0, protokol, vrste transakcij in konfiguriranje naprav USB. Preglejte spremembe, ki jih na fizični, povezovalni in protokolni plasti prinaša standard USB 3.0 in opišite kako je dosežena kompatibilnost s standardom USB 2.0.

Mentor:

pred. mag. Igor Škraba

Dekan:

prof. dr. Nikolaj Zimic



ZAHVALA

Že dolgo pred vpisom na fakulteto me je najbolj zanimala arhitektura in strojni del računalništva. Zato sem se tudi odločil za temo, ki pokriva to področje. Med študijem sem spoznal, da pred. Igor Škraba predava predmete, ki me najbolj zanimajo. Zato sem bil mnenja, da bo pred. Igor Škraba najbolj primeren mentor za moje diplomsko delo, in nisem se motil. Ob priložnosti se bi mu rad zahvalil za literaturo, dobro vodenje in nasvete, ki so mi pomagali pri izdelavi diplomskega dela.

Zahvalil bi se rad tudi staršem, ki so mi finančno omogočili brezskrbni študij in me tekom njega tudi spodbujali. Zahvaljujem se tudi sošolcem, ki so mi nesebično pomagali v času študija.

POVZETEK

Danes je povezljivost med napravami ena ključnih zahtev na katere morajo misliti razvijalci elektronskih naprav. V diplomski nalogi je opisano eno najbolj popularnih vodil, to je standard USB. Cilj diplomske naloge je opisati bistvene razlike med standardoma USB 2.0 in USB 3.0 in združljivost med standardoma. Diplomsko sem razdelil na dva dela. Sprva sem opisal USB 2.0 in nato predstavil novosti, ki jih prinaša USB 3.0. Na kratko pa je predstavljen tudi USB OTG, ki je dodatek k standardu USB 2.0.

V uvodu diplome sem opisal razloge, ki so botrovali, da so proizvajalci začeli razvijati USB in kratek kronološki pregled standarda. Za tem so predstavljene glavne komponente, ki sestavljajo USB kot celoto. V obeh delih diplome sem začel z pregledom komunikacije na nizkonivojskem delu. Nato sem predstavil posamezne pakete, ki so značilni pri komunikaciji USB. Večji del diplome pa opisuje protokolni nivo. Ta opisuje pravila in način komunikacije glede na napravo, ki je priključena na vodilo. Opisan je tudi konfiguracijski postopek, ki se izvede ob priklopu naprave USB na vodilo.

Ključne besede: univerzalno serijsko vodilo, transakcije USB, full-speed, high-speed, super-speed

ABSTRACT

Connectivity between broad range of devices is today one of the key components on which developers of electronic devices have to focus. In my diploma thesis I will present the USB (Universal serial bus), today being most widely used PC bus. The main goal is to present and point out the differences between USB 2.0 and USB 3.0, on the other hand however, I also point out the compatibility between both USB standards. Paper is divided into two parts, first I have focused on describing USB 2.0 standard followed with the presentation of the key innovations used in USB 3.0 standard. Being one of the important aspects of USB 2.0 standard USB On the go (OTG) technology, is presented as a short reference.

Introduction is focused on the reasons for the birth of USB technology and also features the chronological overview of the USB standard. Following introduction, are presented the key components that are forming USB as a standard. Both parts of my paper are dedicated to low-level communication overview. For the major part of my paper I have focused on protocol layer of USB standard. For communication on that level individual packages are characteristic. This is the component that describes the rules and ways of communication for a USB device plugged into a bus. I have also put some thought into a configuration process, when the USB device gets plugged into the bus.

Keywords: universal serial bus, USB transactions, full-speed, high-speed, super-speed

ZAHVALA	I
POVZETEK	II
1 UVOD.....	1
1.1 KRONOLOŠKI PREGLED TEHNOLOGIJE USB.....	1
2 PREGLED ARHITETKTURE USB	3
2.1 USB GOSTITELJSKI KRMILNIK.....	4
2.2 NAPRAVA USB	4
2.3 POVEZOVALNI KABLI	5
3 FIZIČNA PLAST USB	7
3.1 LOGIČNI NIVOJI IN PREPOZNAVANJE HITROSTI VODILA	7
3.2 KODIRANJE PODATKOV IN GRADNJA PAKETA	8
3.3 NAPAJANJE IN SUSPENDIRANJE NAPRAV	9
4 MODEL PODATKOVNEGA TOKA.....	11
4.1 KONČNE TOČKE V NAPRAVI.....	12
4.2 LOGIČNE POVEZAVE MED GOSTITELJEM IN NAPRAVO.....	13
5 PROTOKOL USB	15
5.1 ZGRADBA PAKETA USB.....	15
5.2 VRSTE PAKETOV USB	17
5.2.1 Okvirni paket	17
5.2.2 Žetoni.....	18
5.2.3 Posebni žetoni za deljene transakcije	18
5.2.4 Podatkovni paket	19
5.2.5 Usklajevalni paket	19
5.3 TRANSAKCIJE USB.....	20
5.3.1 Kontrola zaporedja podatkovnih transakcij.....	20
5.3.2 Kontrola prenosa z protokolom PING.....	21
5.3.3 Masivni prenos	21
5.3.4 Kontrolni prenos	22
5.3.5 Prekinitveni prenos	24
5.3.6 Izohroni prenos	24
5.3.7 Pravilno nizanje izohronih paketov	25
5.3.8 Low-speed in deljene transakcije	26
6 KONFIGURIRANJE NAPRAVE USB	28
6.1 DESKRIPTORJI USB	28
6.1.1 Deskriptor naprave	29
6.1.2 Konfiguracijski deskriptor.....	30
6.1.3 Deskriptor vmesnika.....	31

6.1.4	Deskriptor končne točke.....	31
6.1.5	Opcijski deskriptor	32
6.2	ZAHTEVKI GOSTITELJA.....	32
6.2.1	Standardni ukazi gostitelja.....	33
7	USB PROGRAMSKA OPREMA.....	35
7.1	KLJUČNE PLASTI GOSTITELJA.....	35
7.2	GONILNIK USB	36
8	USB ON-THE-GO	38
8.1	PROTOKOL USB OTG	39
9	STANDARD USB 3.0.....	40
9.1	KOMPATIBILNOST STANDARDA USB 3.0 IN USB 2.0.....	41
9.2	FIZIČNA PLAST USB 3.0.....	42
9.2.1	Upravljanje z električno porabo	42
9.2.2	Povezovalni kabli	43
9.3	POVEZOVALNA PLAST	45
9.3.1	Okvirjanje paketa.....	45
9.3.2	Povezovalni ukazi.....	46
9.3.3	Diagram prehajanja stanj povezave.....	47
9.3.4	Primer komunikacije na povezovalnem nivoju	49
9.4	PROTOKOLNA PLAST	53
9.4.1	Paket za upravljanje povezave.....	53
9.4.2	Transakcijski paket.....	54
9.4.3	Podatkovni paket	55
9.4.4	Izohrona časovna znamka.....	56
9.5	TRANSAKCIJA PAKETOV PRI STANDARDU USB 3.0.....	56
9.5.1	Usmerjanje paketov po vodilu.....	56
9.5.2	Eksplozijske transakcije	57
9.5.3	Tokovi.....	57
9.5.4	Vrste prenosov super-speed.....	58
9.5.5	KONFIGURIRANJE NAPRAVE USB 3.0 IN DESKRIPTORJI.....	60
10	ZAKLJUČEK.....	62
11	LITERATURA.....	63

1 UVOD

Vhodno-izhodne (V/I) naprave povečujejo funkcionalnost računalnika. Za uspešno delovanje pa morajo znati komunicirati s pomnilnikom in centralno procesno enoto (CPE). Vse tri enote povezujejo prenosne poti. V računalniku so to običajno vodila, preko katerih se prenašajo podatki. Naprave lahko prenašajo podatke serijsko ali paralelno. Za paralelni prenos je značilno veliko število povezav in krajši kabli. Serijski prenos pa ima manjše število povezav, vendar zaradi lažje izvedbe so lahko povezave daljše. Ker pa je lahko na vodilo priključeno veliko naprav, je bilo potrebno vpeljati sistem, ki omogoča skladno komunikacijo s CPU in na računalnik priključenimi napravami. Včasih je bilo za delovanje priključenih naprav potrebno nastaviti prekinitveni zahtevek (angl. interrupt request, IRQ), V/I naslovni prostor, kanal za direktni dostop do pomnilnika (angl. Direct memory access, DMA). Pred predstavitvijo standarda USB je bilo na voljo veliko drugih vmesnikov, tako serijskih RS232, MIDI, PS/2 kot paralelnih LPT, ki pa niso bili preprosti za uporabo. Za uspešno nastavitvev pa si moral biti skoraj izkušen računalniški mojster. Potrebno je bilo ročno nastavljanje IRQ in DMA kanalov. Niso podpirali niti vključevanje oziroma izključevanje med delovanjem (angl. hot-plug) niti niso podpirali samodejno konfiguriranje (angl. Plug&Play). Poleg tega pa so imeli tudi okorne kable in nerodne priključke.

Zaradi kompleksnosti uporabe in velikega števila različnih dotedanjih priključkov se je pojavila potreba po novem standardu, ki bi poenostavil priključevanje naprav na računalnik. Razvijalci univerzalnega serijskega vodila so imeli cilj razviti:

- eno vrsto priključka za različne naprave,
- ena vrsta kabla,
- napajanje naprav preko kabla USB,
- samodejno zaznavanje in konfiguriranje naprave,
- hitrejši prenos podatkov,
- podporo za starejše naprave,
- možnost priključitve več naprav,
- cenovno ugodno vodilo.

Tako so proizvajalci želeli ustvariti računalnik brez zastarelih vmesnikov, da bi vseboval samo vmesnik ethernet, SCSI in USB. Toda proizvajalci, tudi vodilni pri oblikovanju standarda USB, so se težko v kratkem času odrekli starejšim vodilom. Leta 1998 je bil Apple iMac prvi, ki je poslal na trg računalnik, ki je vseboval samo vodilo USB in FireWire.

1.1 KRONOLOŠKI PREGLED TEHNOLOGIJE USB

Razvoj standarda USB se je pričel leta 1994. Pri tem je sodelovalo sedem računalniških in telekomunikacijskih podjetij Compaq, Digital Equipment Corporation, IBM, Intel, Northern Telecom, Microsoft in NEC. Leto kasneje se je tem podjetjem pridružilo še okoli 350 podjetij, ki so skupaj ustanovila neprofitno organizacijo USB Implementers Forum ali na kratko USB-IF. Glavne naloge foruma so vzdrževanje in podpora razvijalcem, prodaja, promocija in skrb za skladnost standarda. Danes je v USB-IF včlanjenih že preko 1000 podjetij.

USB 1.0 je prvi predstavil Intel leta 1995, uradne specifikacije pa so bile predstavljene januarja 1996. USB 1.0 je omogočal prenos podatkov s hitrostjo 1,5 Mb/s in se danes imenuje nizko-hitrostni (angl. low-speed, krajše LS) USB. USB 1.0 je bil najprej podprt v Windows 95 OSR2. Leto po predstavitvi je bilo na trgu okoli 500 naprav, ki so omogočale priklop preko vodila USB. Čeprav gre za današnje čase za zelo počasno verzijo, se ga še vedno uporablja za nezahtevne naprave.

USB 1.1 je bil predstavljen leta 1998 in je odpravil nekaj implementacijskih napak, ki so ovirale stabilno komunikacijo standarda USB 1.0. Nova verzija je omogočila tudi izohrono komunikacijo. USB 1.1 omogoča prenos podatkov s hitrostjo 12 Mb/s, za njega pa se uporablja izraz polno-hitrostni (angl. full-speed, krajše FS) USB. USB 1.1 je bila verzija, ki se je začela množično uporabljati in leta 2000 zaslovela kot prevladujoče vodilo.

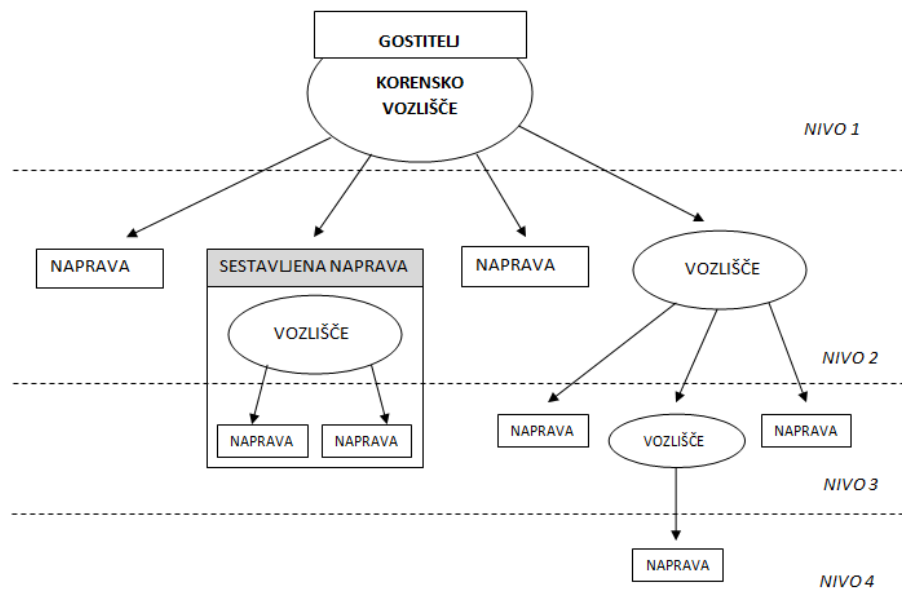
USB 2.0 specifikacije so bile objavljene aprila 2000. Nova verzija je omogočala 40x hitrejši prenos podatkov in tako dosega 480 Mb/s. Za USB 2.0 se uporablja oznaka visoko-hitrostni (angl. high-speed, krajše HS) USB. Z prihodom USB 2.0 so ukinili USB 1.1, obdržali pa so obe hitrosti, tako je potrebno pri nakupu paziti na oznako hitrosti FS ali HS.

USB On-The-Go specifikacije so bile predstavljene 2001. Za priklop naprav USB je bil do tedaj vedno potreben računalnik, ki je predstavljal gostitelja. USB OTG pa omogoča povezovanje dveh perifernih naprav pri kateri ena prevzeme vlogo gostitelja, tako računalnik ni več potreben. Na primer povezava fotoaparata s tiskalnikom. USB OTG omogoča prenos podatkov LS, FS in HS pri manjši porabi električne energije.

USB 3.0 specifikacije so bile določene novembra 2008. Prva naprava pa je prišla na trg januarja 2010. V primerjavi s starejšimi verzijami se je komunikacijski protokol močno spremenil. Tako krmilniki USB 3.0 vsebujejo tudi ločen krmilnik USB 2.0. Prenos podatkov se sedaj vrši s hitrostjo 5,0 Gb/s in se imenuje super-hitri (angl. super-speed, krajše SS). USB 3.0 pa omogoča poleg večje hitrosti tudi manjšo porabo električne energije.

2 PREGLED ARHITETKTURE USB

Fizično naprave USB priključujemo v drevesno strukturo, kot prikazuje *slika 1*. Naprave USB so fizično povezane na vozlišče. Vozlišče je posebna naprava USB, ki omogoča preko prostih priključkov priklop drugih naprav. Na najvišjem nivoju imamo korenisko vozlišče, ki je fizično povezano neposredno na gostitelja in je del njega. Vse povezave naprav ustrezajo principu točka-v-točka, zato v ena sama vrata (angl. port) ne moremo povezati dve napravi. Vsako vozlišče ali naprava predstavlja en nivo. Zaradi časovnih zakasnitev, ki jih ustvari dolžina kabla in vozlišča, je največje število nivojev vključno s koreniskim vozliščem, omejeno na sedem. To pomeni, da je lahko med gostiteljem in napravo največ pet vozlišč. Na sedmi nivo lahko priključimo samo napravo.



Slika 1: Vsaka naprava zasede določen nivo v topologiji USB, pri tem sestavljena naprava vedno zasede dva nivoja.

Terminologija USB predstavlja tudi sestavljene naprave (angl. compound device). Gre za napravo USB, ki jo sestavlja več vgrajenih naprav USB in se jih fizično ne da ločiti. Vedeti je potrebno tudi, da se sestavljene naprave štejejo za dva nivoja, čeprav so na zunaj videti kot ena naprava. Na *sliki 1* lahko vidimo, da en nivo predstavlja vozlišče v sestavljeni napravi, drugi nivo pa predstavlja napravo. Iz perspektive gostitelja zglada sestavljena naprava enako kot naprava, ki ima ločeno vozlišče na katerega je priključeno več naprav. Predstavniki take naprave je tipkovnica z vgrajeno sledilno ploščico. Gostitelj vidi sestavljeno napravo, kot vozlišče na katerega sta priključeni dve napravi, tipkovnica in sledilna ploščica. V tem primeru bo gostitelj sestavljeni napravi dodelil tri različne naslove.

Glede na fizično vezavo naprav v drevesno strukturo, lahko dobimo precej razvejano omrežje. Toda z logične perspektive gostitelj komunicira z vsako napravo, kot bi bil z njo neposredno povezan, ne glede na to ali je ta naprava (ali pa tudi vozlišče) priključena še na kakšno vozlišče.

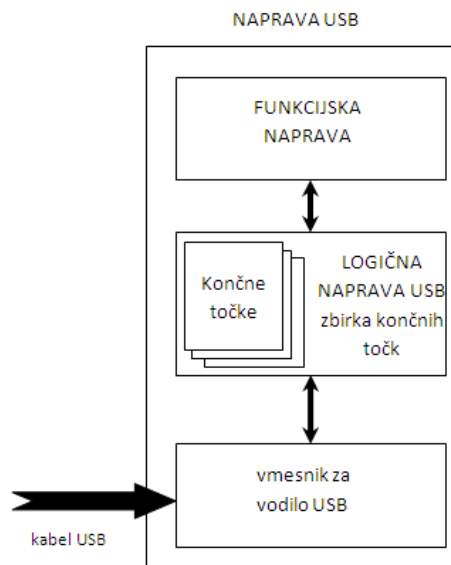
2.1 USB GOSTITELJSKI KRMILNIK

Naloga gostiteljskega krmilnika je voditi listo transakcij in jih razvrščati na vodilo. Vodi tudi stanje vsake transakcije (ali je opravljena, v zastoju na čakanju...). Ko je bil aktualen standard USB 1.1 sta obstajala dva ločena gostiteljska krmilnika (angl. host controller interface, HCI) Compaqov Open HCI (krajše: OHCI), ki ga je USB-IF vzel za svojega in Intelov Universal HCI (krajše: UHCI), ki za njega zahteva licenco. Oba krmilita prenose LS in FS. UHCI uporablja poleg Intela še podjetje VIA. Razlika med krmilnikoma je v tem, da je UHCI bolj programsko voden, ima večje zaglavje in je procesorsko bolj zahteven. Kot dobra stran pa je, da ga je strojno lažje implementirati, saj ima manj logičnih vrat v primerjavi z krmilnikom OHCI. S prihodom USB 2.0 je USB-IF zahteval samo en krmilnik, tako je nastal Extended HCI (krajše EHCI), ki skrbi za komunikacijo HS. Ne glede na tip gostiteljskega krmilnika, v osnovi vsi opravljajo isto nalogo. Pomembno je samo, da imamo ustrezen gonilnik, ki mora biti prilagojen OHCI, UHCI ali EHCI strojni implementaciji gostiteljskega krmilnika. Gostiteljski krmilnik tako med drugim poskrbi za:

- generacijo (mikro)okvirjev,
- možnost postavljanja naprav v stanje pripravljenosti in oddaljenega obujanja,
- procesiranje zahtevkov za sprejem in prenos podatkov,
- serijalizacija in deserijalizacija podatkov,
- detekcija in korekcija napak.

2.2 NAPRAVA USB

Vse naprave USB so deljene v razrede naprav (angl. device classes). Teh razredov je 19, nadalje pa jih lahko razdelimo tudi v podrazrede. Najbolj znani predstavniki določenega razreda so tiskalniki, sekundarni pomnilniki (USB ključi, prenosni diski), vozlišča USB, naprave HDI (miške, tipkovnice).



Slika 2: Ključni deli vsake naprave USB.

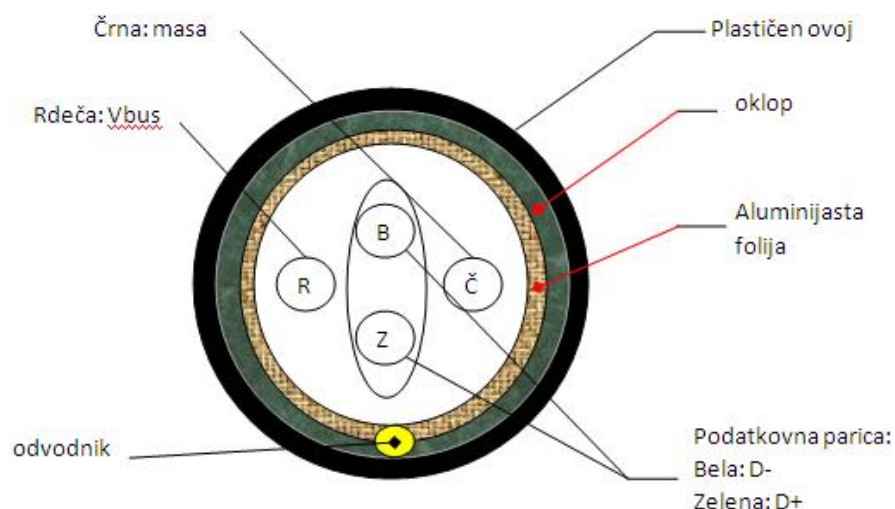
Vsaka naprava mora vsebovati informacijo za samopredstavitev. To informacijo nosi v posebni podatkovni strukturi imenovani deskriptor, ki tudi razkrije zmogljivosti in način delovanja naprave. Kot prikazuje *slika 2*, napravo USB sestavljajo trije nivoji:

- Vmesnik za vodilo USB (angl. USB bus interface): del naprave, ki sprejema in pošilja pakete po protokolu, ki ga narekuje standard USB.
- logična naprava USB: del naprave, ki usmerja podatke med vmesnikom in različnimi končnimi točkami. Končne točke so izvor ali ponor vseh paketov, ki se prenašajo med gostiteljem in napravo.
- Funkcijska naprava: je najvišji nivo naprave USB. Dejansko predstavlja fizično napravo, kot jo vidi končni uporabnik in omogoča novo funkcionalnost gostitelju (kamera, miška, tiskalnik, USB ključ...).

2.3 POVEZOVALNI KABLI

USB prenaša podatke in napajanje po štiri žilnem kablju, prikazanem na *sliki 3*. Dve žili se uporabljata za napajanje in sta označeni kot Vbus in GND (masa). Prenos podatkov med vozliščem in napravo pa se vrši po podatkovnih linijah D+ in D-, ki sta med seboj prepleteni.

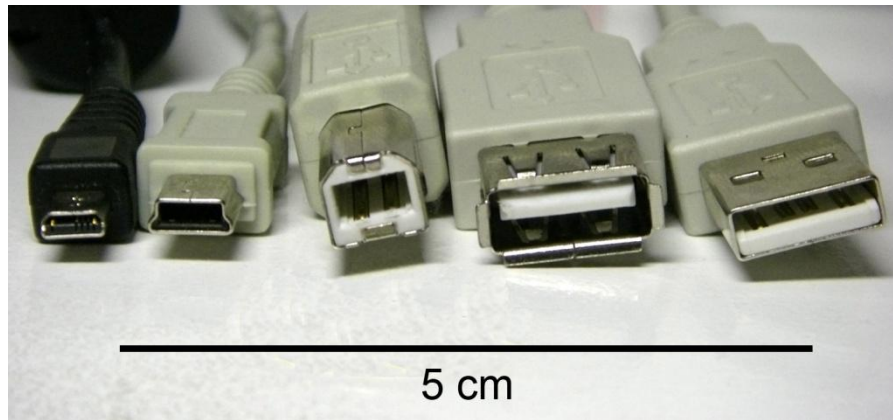
Kabel USB ima na enem koncu drugačen priključek, kot na drugem. Ta princip so razvijalci standarda izbrali zato, da uporabnik naprave ne more napačno priključiti. Standardni USB kabel ima zakasnitev signala 5,2 ns/m. Največja zakasnitev signala med napravo in vozliščem, ki jo standard USB predpisuje, je 26 ns. To pa pomeni, da je lahko standarden kabel dolg največ $26\text{ns}/5,2\text{ns/m} = 5\text{ m}$. To razdaljo lahko podaljšamo, če uporabimo vozlišča, ki delujejo kot ponavljavci. Toda več kot 5 vozlišč ne moremo uporabiti zaradi omejitev časovne kontrole. S tem dosežemo največjo možno razdaljo 30 m.



Slika 3: Zgradba kabla USB.

Glede na smer povezovanja standard USB deklarira priključke A in B. A vtič se uporablja na strani vozlišča, oziroma na tako imenovanih vhodnih vratih (angl. upstream port). Naprave, ki uporabljajo kabel z vtičem A, imajo pogosto drugi konec kabla trajno vgrajen v napravi in ga

ni mogoče odklopiti, to sta na primer tipkovnica in miška. Priključek B je kvadratne oblike in se ga vključi v periferne naprave oziroma v izhodna vrata (angl. downstream port), na primer tiskalnik. Priključka A in B sta namenoma drugačne oblike in ju je nemogoče napačno priključiti, saj bi s tem lahko povzročil kratek stik.



Slika 4: Iz leve proti desni si sledijo: mikro A vtič, mini A vtič, standardni B vtič, standardni A podaljšek, standardni A vtič.

Glede na velikost priključkov USB deklarira standardni, mini in mikro priključek. Standardni priključki se običajno uporabljajo za naprave, pri katerih ni prostorske stiske, torej za vse namizne in večje prenosne računalnike. Standardni priključki imajo deklarirano življenjsko dobo 1500 priključnih ciklov. Mini priključki merijo okoli 3x7 mm in se uporabljajo, kjer je prostor bistvenega pomena, recimo mobilni telefoni, kamere itd. V letu 2007 so razvijalci standarda USB predstavili tudi priključek mikro, ki naj bi zamenjal priključek mini. Na sliki 4 lahko vidimo, da je fizična razlika med priključkom mikro in mini v tem, da je priključek mikro v višino za približno polovico manjši. Priključek mikro ima tudi daljšo življenjsko dobo, saj naj bi zdržal 10000 priključnih ciklov, medtem ko ima mini deklarirano življenjsko dobo 5000 ciklov.

USB definira še posebno vtičnico AB, ki jo uporabljajo naprave OTG. V to vtičnico lahko vključimo tako vtič A kot tudi B. Pri tem ni bojzani, da bi napačno priključili napravo. OTG je namreč posebne vrste naprava USB, ki lahko deluje kot gostitelj ali pa kot periferne naprava.

3 FIZIČNA PLAST USB

3.1 LOGIČNI NIVOJI IN PREPOZNAVANJE HITROSTI VODILA

USB 2.0 za prenos podatkov uporablja tri različne hitrosti z standardnimi nazivi:

- high-speed (HS) podatki se prenašajo s hitrostjo 480 Mb/s,
- full-speed (FS) podatki se prenašajo s hitrostjo 12 Mb/s,
- low-speed (LS) podatki se prenašajo s hitrostjo 1,5 Mb/s.

USB za logične nivoje »0« ali »1« uporablja oznaki »J« in »K«. Interpretacija posameznega logičnega nivoja je opisana v tabeli 2, saj se interpretacija spreminja glede na signalizacijo LS ali FS/HS. Poleg omenjenih stanj USB pozna še asimetrično povezavo (AP), ki se uporablja za posebne nizkonivojske ukaze, kot je na primer reset naprave (tabela 1).

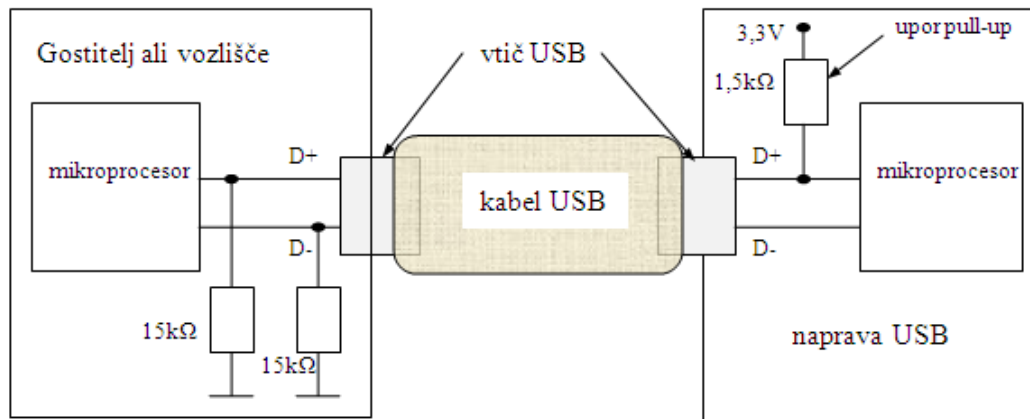
Logično stanje	Stanje na liniji
Logična »0«	D- visoko, D+ nizko
Logična »1«	D+ visoko, D- nizko
Asimetrična povezava »1« (AP1)	D+ visoko, D- visoko
Asimetrična povezava »0« (AP0)	D+ nizko, D- nizko

Tabela 1: Stanje signala na liniji glede na logično stanje.

J stanje	Stanje na liniji
LS	Logična »0«
FS ali HS	Logična »1«
K stanje	Stanje na liniji
LS	Logična »1«
FS ali HS	Logična »0«

Tabela 2: Interpretacija logičnega stanja glede na hitrost signalizacije.

Krmilnik USB ugotavlja hitrost vodila preko uporov pull-up, priključenih na podatkovno linijo D+ ali D- in 3,3 V napetostni izvor. Naprave, ki podpirajo hitrost FS (*slika 5*) to predstavijo s padcem napetosti na upor priključenem na D+, kot prikazuje *slika 5*. Naprave, ki podpirajo hitrost LS, pa s padcem napetosti na upor, priključenem na D-. V primeru, da na podatkovno linijo ni priključenega upora pull-up, USB smatra, da na vodilo ni priključena nobena naprava.



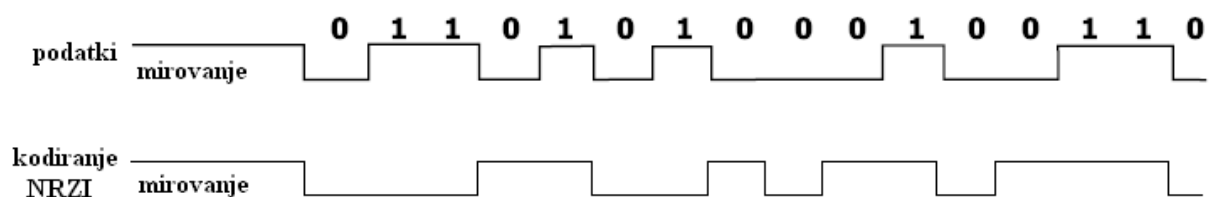
Slika 5: Naprava FS se predstavi z padcem napetosti preko 1,5 k Ω upora pull-up.

Naprava HS se ne more preprosto predstaviti preko upora pull-up. Za to se ob priklučitvi predstavi kot naprava FS. Vozlišče vedno po priklopu naprave FS izvede postopek za ugotavljanje ali je priključena naprava zmožna komunikacije HS. Dejansko vozlišče pošlje več zaporednih nizov »JK« v signalizaciji HS, ki jih mora naprava uspešno sprejeti. Če jih uspešno sprejme, vozlišče izvede reset vodila in nadaljuje komunikacijo v načinu HS.

3.2 KODIRANJE PODATKOV IN GRADNJA PAKETA

USB ne uporablja posebne linije za sinhronizacijo ure. Da sta oddajnik in sprejemnik pri pošiljanju in sprejemanju sinhronizirana, se uporablja kodiranje NRZI (non-return-to-zero inverted) z vrivanjem bitov. Primer kodiranja NRZI prikazuje *slika 6*, in deluje po principu:

- logična »1« je predstavljena z ohranitvijo logičnega stanja
- logična »0« je predstavljena s spremembo logičnega stanja

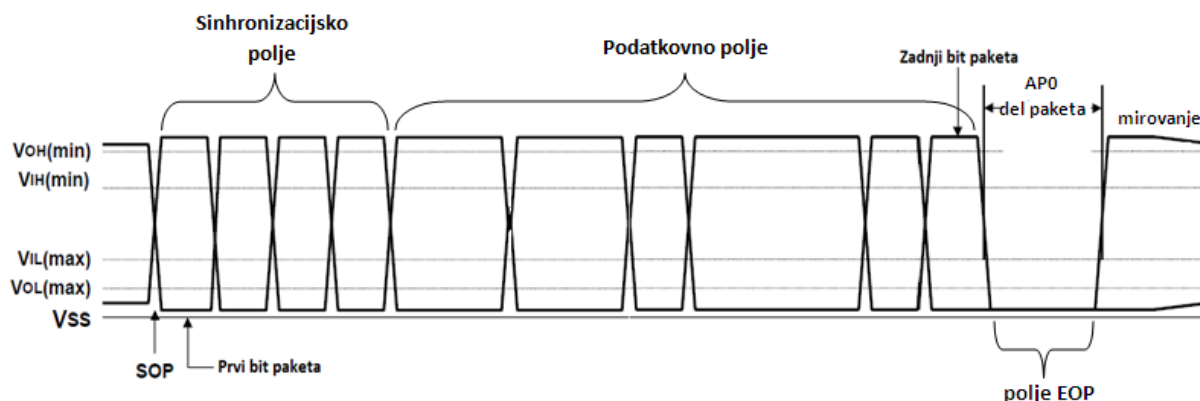


Slika 6: Kodiranje NRZI.

Dobra stran kodiranja NRZI je, da je medij najboljše izkoriščen. Slabo stran pa predstavljajo dolgi nizi logičnih »1«, ki predstavljajo slabo informacijo za sinhronizacijo ure sprejemnika in oddajnika, zato se uporablja vrivanje bitov. Oddajnik zazna, kdaj se pošlje 6 zaporednih logičnih »1«. V tem primeru bo sedmi bit vedno vrinjena logična »0«. Sprejemnik bo vrinjeno logično »0« zaznal in jo odstranil pred nadaljnjo obdelavo podatkov. Za dodatno sinhronizacijo sprejemnika in oddajnika poskrbi tudi sinhronizacijsko polje znotraj vsakega poslanega paketa (angl. SYNC field). Polje SYNC predstavlja začetek vsakega paketa USB. Polje SYNC je sestavljeno z zaporednim preklapljanjem logičnih nivojev (za HS 15 nizov »JK« preklapov, ki mu sledi 2x »J« preklap).

Pri prenosu se eden ali več paketov pošilja znotraj okvirja. Vsak paket pa je zgrajen znotraj dveh mej imenovanih začetek paketa (angl. start-of-packet, SOP) in konec paketa (angl. end-of-packet, EOP). SOP se prične, ko liniji D+ in D-, ki sta trenutno v mirovnem stanju (angl. idle), prožimo v nasprotno logično stanje. To predstavlja prvi bit polja SYNC.

Biti EOP predstavljajo konec paketa. USB to predstavi z asimetrično povezavo v nizkem stanju (AP0) za časovno enoto dveh bitov in nato za časovno enoto enega bita v J stanje. *Slika 7* prikazuje podatkovni paket.



Slika 7: podatkovni paket USB, ki ga sestavljajo polje SYNC, podatkovno polje in biti EOP. Polje SYNC dejansko sestavlja minimalno 8 bitov.

3.3 NAPAJANJE IN SUSPENDIRANJE NAPRAV

Ena od bistvenih prednosti vodila USB pred drugimi vodili je, da lahko napaja priključene naprave. Tako lahko energijsko nezahtevne naprave delujejo brez zunanega napajanja. Vse naprave priključene na vodilo USB se napajajo preko Vbus napetostnega izvora. Napetost na korenskem vozlišču znaša $5\text{ V} \pm 5\%$. Zaradi možnega nihanja napetosti (izključevanje, vključevanje naprav) standard USB veleva, da morajo vse naprave delovati tudi pri $4,4\text{ V}$. USB uvaja poseben koncept dodajanja napajanja določenim napravam, imenovan enota obremenitve (angl. unit load). Ko se naprava priključi na vodilo, dobi 100 mA toka, kar predstavlja eno obremenitveno enoto. Naprava lahko šele v času konfiguracije zahteva dodatne obremenitvene enote vendar največ pet, kar pomeni, da lahko črpa do 500 mA toka. Glede na električno porabo lahko delimo naprave na:

- varčne naprave (angl. Low power device),
- požrešne naprave (angl. High power device),
- naprave z zunanjim napajanjem (angl. self-powered device).

Ob priklopu se za vsako napravo predpostavlja, da je varčna, kar pomeni, da ne more črpati več kot 100 mA toka. Za požrešne naprave se smatrajo tiste, ki za svoje normalno delovanje potrebujejo več kot 100 mA toka. Vseeno se mora požrešna naprava uspešno zagnati in vstopiti v fazo konfiguracije pri toku 100 mA . Svoje energetske potrebe nato poda med konfiguracijo, za normalno delovanje pa ne sme porabiti več kot 500 mA . Naprave, ki

potrebujejo več kot 500 mA toka, potrebujejo zunanje napajanje. Take naprave v primeru izpada zunanjega napajanja iz vodila ne bodo mogle črpati več toka, kot so ga črpale v času, ko so še imele zunanje napajanje.

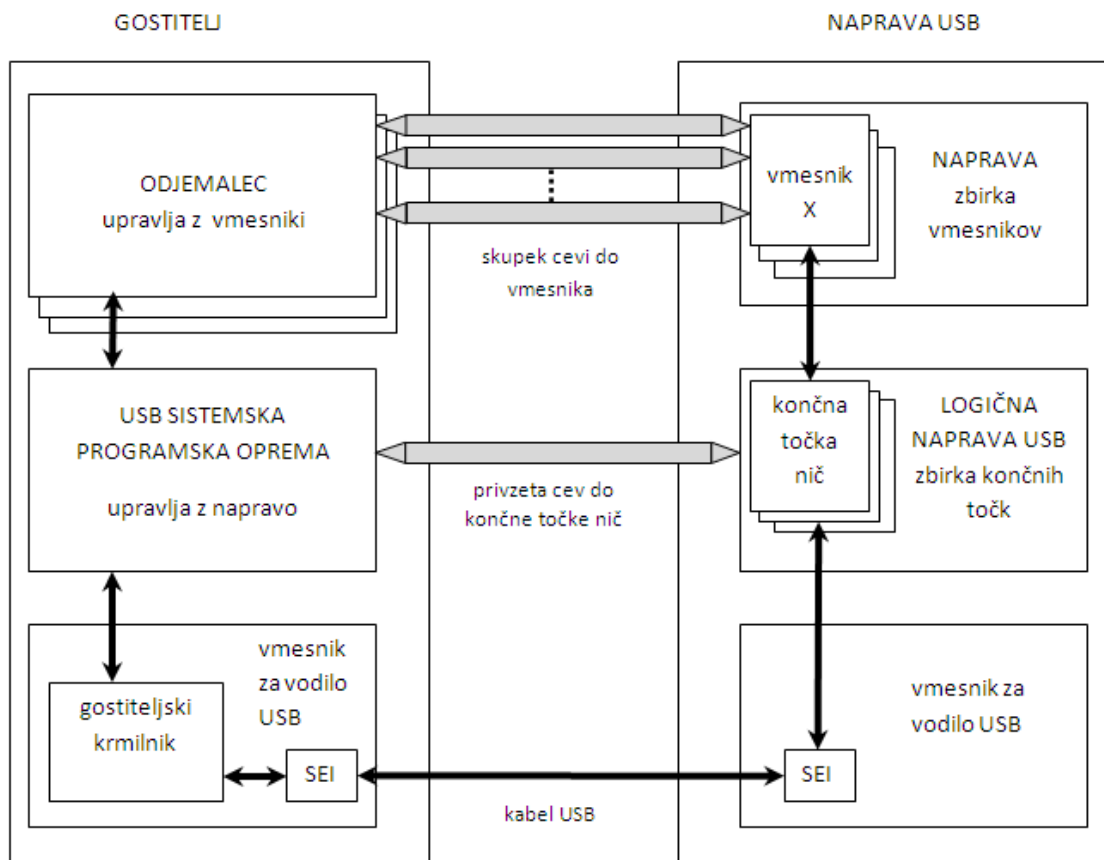
Vsaka naprava USB mora podpirati stanje mirovanja. V to stanje preide vsaka naprava, pri kateri ni aktivnosti na vodilu dlje kot 3 ms. Nato ima naprava 7 ms časa, da opravi aktivnosti za prehod v stanje mirovanja. Specifikacije USB narekujejo 500 μ A porabe toka, ko je naprava v mirovanju. Pogosto pa želimo, da naprava ne preide v mirovanje, predvsem zaradi dolgotrajnega prehoda iz mirovanja. To dosežemo s periodičnim pošiljanjem paketov za začetek okvirja (angl. start-of-frame, SOF) proti napravi.

USB 2.0 pozna selektivno in globalno stanje mirovanja. Pri selektivnem mirovanju lahko programsko določimo, katere naprave bodo prešle v stanje mirovanja. Pri globalnem mirovanju pa prenehajo z aktivnostmi vse naprave USB. Ko gostitelj ali vozlišče izvede ukaz obujanja naprave, se mora naprava odzvati v 50 μ s, nato pa gostitelj počaka 20 ms, da se naprava pripravi za aktivnosti. Globalno mirovanje lahko prekine tudi posamezna naprava, na primer modem ob klicu obudi računalnik.

4 MODEL PODATKOVNEGA TOKA

Končni uporabnik ne vidi več kot naprave s priključnim kablom in mu tudi ni potrebno, saj je zaradi dobro zasnovanega standarda priključena naprava na voljo za uporabo brez dodatnih nastavitvev. Bolj podroben pregled gostitelja in naprave razkrije naslednje glavne sestavne dele (slika 8):

- naprava USB: s katero želimo pridobiti dodatno funkcionalnost,
- odjemalec: je program, ki se izvaja na gostitelju in omogoča komunikacijo s ciljno napravo USB,
- USB sistemska programska oprema: je programska oprema prilagojena določenemu operacijskemu sistemu in omogoča izvedbo komunikacije z USB napravo,
- gostiteljski krmilnik USB: je strojna oprema s programsko opremo, ki omogoča priklop naprave na gostitelja.



Slika 8: Blok diagram gostitelja USB in naprave s ključnimi sestavnimi deli. Prikazane so tudi cevi, ki predstavljajo logično povezavo med napravo in gostiteljem.

Različne naprave, ki jih priključimo na vodilo USB, imajo različne zahteve po komunikaciji. USB omogoča dobro izrabo vodila, saj omogoča delitev različnih komunikacijskih tokov med napravo in gostiteljem. Gostitelj začne in vodi vse transakcije.

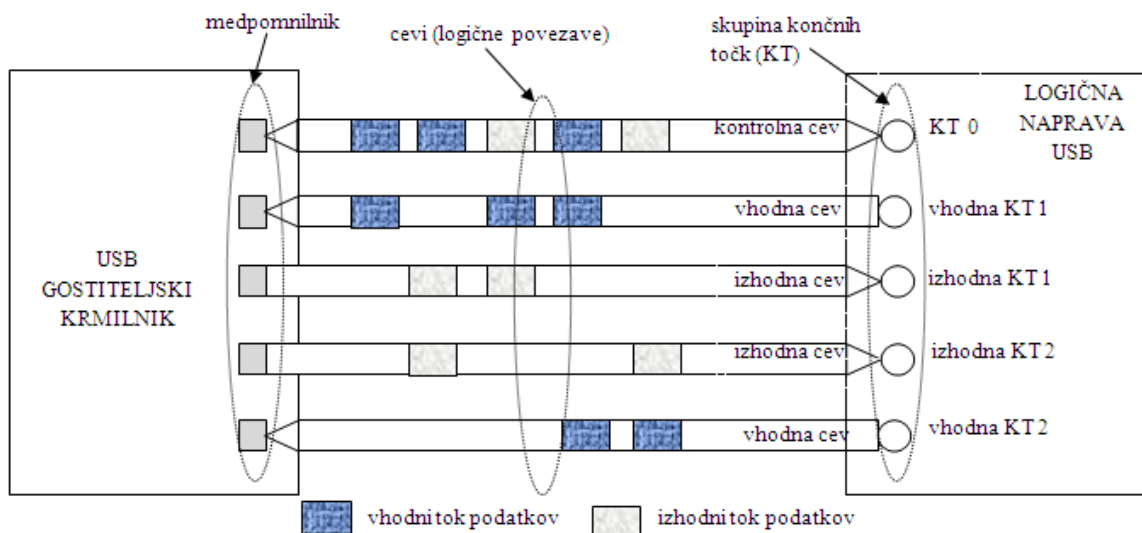
4.1 KONČNE TOČKE V NAPRAVI

Ob priklopu vsaka naprava USB dobi unikatni naslov, ki ga določi gostitelj ob priklučitvi. Napravo sestavlja skupina neodvisnih končnih točk (angl. device endpoint) te pa so naslovljive z unikatno številko. Končna točka je enota v napravi USB, kjer se konča ali začne komunikacijski tok med gostiteljem in napravo (slika 9). Končne točke dobijo svojo stalno številko že ob snovanju naprave USB. Standard USB definira štiri vrste končnih točk, ki ustrezajo določeni vrsti prenosa podatkov:

- masivna končna točka (angl. bulk endpoint),
- prekinitvena končna točka,
- izohrona končna točka,
- kontrolna končna točka.

S kombinacijo naslova naprave in številko končne točke lahko dostopamo do vsake končne točke. Samo kontrolna končna točka omogoča prenos podatkov v obe smeri. Ostale vrste končnih točk pa omogočajo povezavo half-duplex, kar pomeni, da podatkovni tok teče le v eno smer ali od naprave proti gostitelju ali obratno. Če želimo komunikacijo v obe strani, potrebujemo dve ločeni končni točki. Končna točka določa način prenosa med odjemalcem in končno točko in nosi informacije o:

- zahtevi po dostopnem času,
- zahtevi po pasovni širini,
- številki končne točke,
- načinu rokovanja z napakami,
- največji velikosti paketa, ki ga je sposobna prejeti oz. poslati,
- vrsti podatkovnega prenosa,
- smer podatkovnega toka.



Slika 9: Končne točke predstavljajo izvor in ponor podatkovnega toka. Prenos podatkov med končno točko in gostiteljem pa poteka po logičnih povezavah imenovanih cevi. Cev povezuje končno točko z odjemalcem, ki teče na gostitelju.

Lastnosti in število končnih točk so gostitelju neznane in tudi nedosegljive, vse dokler se ne izvede konfiguracija naprave. To pa ne velja za posebno vrsto končne točke, imenovano končna točka nič (angl. endpoint zero).

Končna točka nič mora biti v napravi USB vedno implementirana in je vedno dostopna sistemski programski opremi, ko napravo priključimo na vodilo. Za dostop do končne točke nič morajo imeti vse naprave USB implementirano privzeto kontrolno metodo (angl. default control method). S to metodo lahko USB sistemska programska oprema izvede konfiguracijo naprave ali pa jo uporabi za reševanje naprave USB ob nepredvidenih dogodkih.

Funkcijska naprava ima lahko več končnih točk, ki so namenjene odjemalcu. Končna točka nič je vedno prisotna, medtem ko je število ostalih končnih točk odvisno od potreb same naprave. Vseeno veljajo omejitve, in sicer LS naprava je omejena na do dve dodatni končni točki. FS in HS naprave pa imajo lahko do 15 vhodnih in ločeno 15 izhodnih končnih točk.

4.2 LOGIČNE POVEZAVE MED GOSTITELJEM IN NAPRAVO

Logične povezave med gostiteljem in napravo standard USB poimenuje cevi (angl. pipe). Predstavljajo tok podatkov skozi celotno strukturo USB, od odjemalca na gostitelju skozi podatkovni medpomnilnik in se zaključijo v končni točki v napravi. Standard USB definira dve vrsti cevi:

- tokovna cev (angl. stream pipe),
- sporočilna cev (angl. message pipe).

Cevi imajo parametre povezane s končnimi točkami, na primer koliko pasovne širine potrebujejo, vrsta in smer prenosa, velikost paketa. Končno točko nič vedno povezuje privzeta kontrolna cev (*slika 9*), ki spada pod sporočilno cev. Ta cev se kreira takoj ob priklopu naprave USB na vodilo. Uporablja se za konfiguracijo naprave in za reševanje naprav iz zastojev (angl. stall). Cevi, ki povezujejo druge končne točke, pa se kreirajo po končani konfiguraciji naprave USB.

Odjemalčeva programska oprema zahteva transakcije preko zahtevkov V/I (angl. i/o request packets, IRP) do cevi in je obveščena, ko je transakcija zaključena. V primeru napake pri izvajanju zahtevka IRP, oziroma če pride do nepredvidenega stanja, pošlje naprava gostitelju paket STALL. Prekine se izvajanje trenutnega in tudi vseh čakajočih zahtevkov IRP. Da si naprava opomore, je potrebna programska intervencija preko privzete kontrolne cevi, do takrat pa so vse ostale cevi v mirovanju.

Tokovne cevi ne vsebujejo značilne USB podatkovne strukture. Prenos podatkov ustreza principu FIFO, smer prenosa pa je vnaprej določena z žetonom, ki ga definira protokolni nivo. Tokovne cevi podpirajo masivno (angl. bulk), izohrono in prekinitveno vrsto prenosa podatkov.

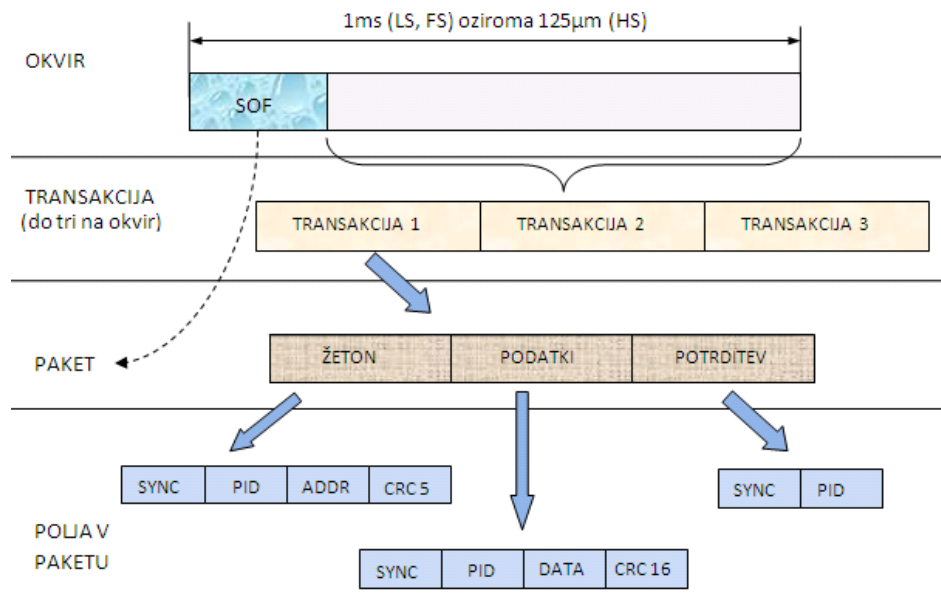
Sporočilne cevi imajo za razliko od tokovnih cevi določeno USB strukturo podatkov, kar omogoča, da se ukazi zanesljivo prepoznajo in izvedejo. Najprej se pošlje zahteva proti napravi. Zahtevi lahko sledi podatkovni tok v vnaprej določeni smeri. Na koncu pa se pošlje še statusna faza. Sporočilne cevi omogočajo komunikacijo v obeh smereh, čeprav v praksi

prevladuje komunikacija v eno smer. Privzeta kontrolna cev je vedno sporočilna cev. Sporočilno cev za prenos podatkov uporabljajo tudi kontrolne končne točke.

5 PROTOKOL USB

Prenos informacije skozi vodilo USB se imenuje transakcija. Znotraj enega okvirja se lahko izvedejo največ tri transakcije, odvisno od vrste prenosa. Gostiteljski krmilnik ima listo vseh transakcij, ki čakajo na prosto vodilo. Vse transakcije začne gostitelj. Transakcijo USB običajno sestavljajo trije paketi (*slika 10*):

- žeton napove s kom želi gostitelj komunicirati in smer prenosa podatkov,
- podatkovni paketi,
- statusni del, ki nudi informacijo o uspešnosti izvedbe transakcije.



Slika 10: Komunikacija po protokolu USB. Transakcije se izvajajo znotraj okvirja. Posamezni paket sestavljajo različna polja.

5.1 ZGRADBA PAKETA USB

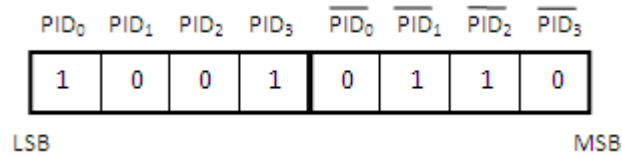
Najmanjša enota informacije, ki se prenaša po vodilu USB je paket in je skupek organiziranih bitov. Velikost kot tudi zgradba posameznega paketa je odvisna od namembnosti paketa (*slika 10*). V splošnem pa lahko paket vsebuje naslednja polja:

- sinhronizacijsko polje (angl. SYNC field),
- identitetno polje (angl. packet ID, PID field),
- naslovno polje (angl. address, ADDR field),
- podatkovno polje (angl. DATA field),
- CRC biti (angl. CRC check).

Vsak paket USB se ne glede na smer prenosa začne s poljem SYNC, ki služi za ponastavitev lokalne ure, oziroma za sinhronizacijo oddajnika in sprejemnika. Polje predstavlja 8 bitov pri LS in FS oziroma 32 bitov pri HS. Pri tem se zadnja dva bita uporabljata za identifikacijo konca polja SYNC, s tem sprejemnik ve, da bo sledilo naslednje polje. Polje SYNC ni

zaščiten s CRC biti. Ker je to polje prisotno na začetku vsakega paketa, ga v nadaljevanju zaradi lažje predstavitve ne bom posebej omenjal.

Polje PID sledi takoj za poljem SYNC in je ravno tako sestavni del vsakega paketa. Polje PID pove kakšno funkcijo bo imel poslani paket (*tabela 3*). Od polja PID je odvisna tudi struktura paketa. Polje sestavlja 8 bitov. Polje PID ni zaščiten s CRC biti, ampak so razvijalci uvedli eniški komplement prvih 4 bitov, ki prav tako služijo za detekcijo napak (*slika 11*).



Slika 11: Dano polje PID predstavlja vhodni žeton. Zadnji štirje biti predstavljajo eniški komplement prvih štirih bitov.

Nedefinirana vrednost polja PID se smatra kot napaka. V tem primeru sprejemnik celoten paket zavrže. Možen je tudi dogodek, da ima polje PID definirano vrednost, toda sprejemnik te operacije ne podpira, tak paket sprejemnik ignorira. Na primer, če v končno točko, ki podpira samo izhodne transakcije prispe žeton za vhodne transakcije.

Vrste paketov	Oznaka	Opis
Žeton	Izhodni žeton	Transakcija bo potekala od gostitelja proti napravi
	Vhodni žeton	Transakcija bo potekala od naprave proti gostitelju
	Začetek okvirja (SOF)	Najava začetka okvirja
	Nastavitveni paket (SETUP)	Začetek nastavitvene transakcije
Podatkovni	DATA0	Pri dolgih podatkovnih nizih s tem poljem PID preverjamo pravilno zaporedje prispelih paketov
	DATA1	
	DATA2	
	MDATA	
Usklajevanje	ACK	Potrjevanje pravilnega paketa
	NAK	Zavrnitev poškodovanega paketa. Ali pa naprava ne more poslati oz prejeti paket.
	STALL	Zastoj končne točke
	NYET	Sprejemnik se trenutno ne more odzvati
Posebni	Preambula (PRE), napaka (ERR)	Za deljene transakcije pri napravah LS
	SPLIT	Zahteva za deljeno transakcijo
	PING	Za potrebe HS masivnih ali prekinitvenih transakcij

Tabela 3: Pregled različnih vrst paketov.

Naprave USB in končne točke se naslavljajo preko naslovnega polja, ki vsebuje naslov naprave dolžine šestih bitov in številko končne točke dolžine štirih bitov. Paket, ki ga pošlje

gostitelj prispe do vseh naprav, ki so priključene na vodilo. Nato pa vsaka naprava, ki je že konfigurirana, prebere naslovno polje paketa in če le to ustreza njenemu naslovu naprave, ga sprejme, sicer pa zavrže. Ker je naslovno polje dolgo sedem bitov, lahko USB naslovi največ 127 naprav (naslov 0 je rezerviran). Na novo priključena naprava ima vedno privzeti naslov 0. Šele po končani konfiguraciji dobi unikaten naslov. Za naslavljanje končnih točk pa so v naslovnem polju rezervirani 4 biti. Končna točka nič je prisotna še preden naprava dobi unikaten naslov.

Podatkovno polje je lahko veliko od 0 pa do največ 8192 bitov. Dejanska velikost podatkovnega polja je odvisna od vrste prenosa podatkov. Naprava je lahko izvor ali ponor podatkovnih paketov.

CRC biti se uporabljajo za detekcijo napak in pokrivajo vsa polja znotraj paketa razen polja SYNC in PID. CRC biti se izračunajo preden se izvede potencialno vrivanje bitov. Sprejemnik odstrani vrinjene bite preden se izračunajo CRC biti. CRC biti omogočajo zaznavanje enojnih in tudi dvojnih napak in v primeru le teh se celoten paket zavrže. Za žetone se uporablja 5 CRC bitov, medtem ko se za podatkovni paket uporablja 16 CRC bitov.

5.2 VRSTE PAKETOV USB

USB 2.0 definira štiri različne vrste paketov:

- okvirni paket,
- žetoni,
- podatkovni paket,
- usklajevalni paket.

Vsaka transakcija USB 2.0 se začne z žetonom, nato običajno sledi podatkovni paket in na koncu usklajevanje. Transakcije se lahko izvajajo samo znotraj okvirja. Začetek okvirja sporoči okvirni paket. Vsi paketi se končajo s tri bitno EOP (angl. end-of-paket) mejo, ki sledi na koncu paketa. V primeru, da pride do sprejemnika veljaven paket, vendar ne dobi zaključne bite EOP, se tak paket ignorira.

5.2.1 Okvirni paket

Gostitelj napove začetek okvirja z okvirnim paketom. Ta paket gostitelj periodično pošilja proti napravam vsako milisekundo pri signalizaciji LS in FS. Pri signalizaciji HS pa se okvir pošilja v intervalu 125 μ s. Ti paketi vsebujejo polje PID, ki mu sledi 11 bitno polje za vpisovanje številke okvirja in na koncu še CRC biti. Številka okvirja se inkrementira za vsak poslani okvir, za kar poskrbi gostiteljski krmilnik. Ko 11 bitno polje prekorači največjo vrednost 2047, začne šteti ponovno od 0 naprej. Čeprav se pri signalizaciji HS ti paketi generirajo na vsake 125 μ s, se številka okvirja spremeni šele na vsako milisekundo. Pri tem mora naprava sama šteti ostalih osem okvirnih paketov, če je za napravo število okvirjev pomemben parameter. Naprave, ki imajo ostre časovne zahteve lahko dekodirajo polje PID (ostalo vsebino paketa zavržejo) in tako dobijo informacijo, da se je začel nov okvir.

5.2.2 Žetoni

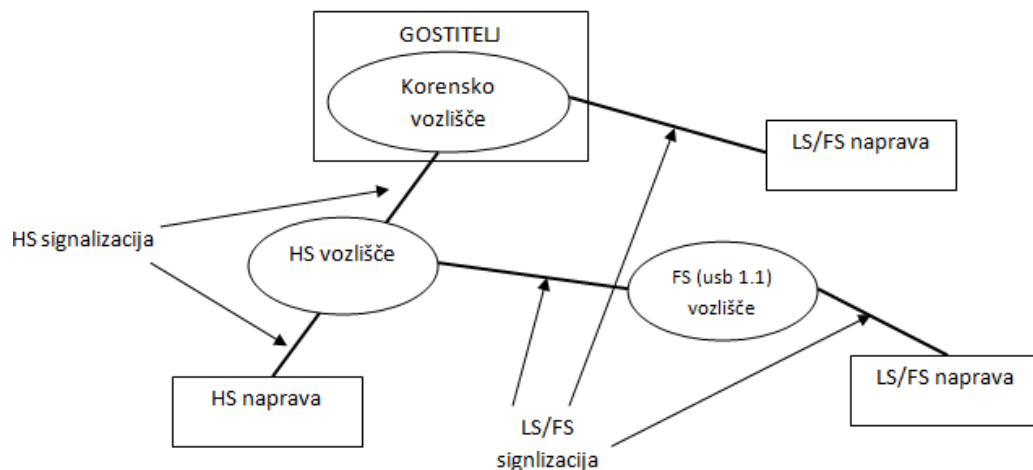
Žetone lahko generira samo gostitelj. Z njimi narekuje smer prenosa podatkov, ki bo potekal po prejetju žetona. Žeton je lahko vhodni, izhodni ali nastavitveni, kar določa polje PID v žetonu. Nato sledi naslovno polje naprave in številka končne točke na koncu pa še CRC biti. V skupino žetonov spada tudi posebni žeton PING, ki vsebuje enaka polja.

Z izhodnimi ali nastavitvenimi žetoni gostitelj izbere prejemnika podatkovnih paketov, ki sledijo od gostitelja proti napravi. Pri vhodnih žetonih pa se z naslovnim poljem določa katera končna točka v napravi bo oddala podatkovne pakete proti gostitelju. V primeru, da se paket žetona med prenosom pokvari, ga naprava ignorira in počaka na naslednji uspešno prenesen žeton.

5.2.3 Posebni žetoni za deljene transakcije

Standard USB definira tudi poseben žeton za deljene transakcije. Namen deljenih transakcij je, da naprave LS in FS s svojo počasno komunikacijo ne bi ovirale komunikacije HS. Ta žeton se uporablja samo za transakcije med gostiteljem in vozliščem HS. Naprave, ki so priključene na to vozlišče pa lahko delujejo z signalizacijo LS in FS (slika 12). Ta princip komunikacije je zelo uporaben kadar imamo na vozlišče priključene naprave, ki komunicirajo z različnimi hitrostmi, saj najpočasnejša naprava ne vpliva na hitrost komunikacije najhitrejših naprav. Žeton za deljene transakcije je dolg 4B, kar je za 1B več kot navaden žeton, saj nosi dodatno informacijo o napravi, ki potrebuje deljeno transakcijo. Žeton za deljeno transakcijo se deli na dve transakciji:

- Začetek deljene transakcije (angl. start-split transaktion, SSPLIT),
- Končanje deljene transakcije (angl. complete-split transaction, CSPLIT);



Slika 12: S pomočjo deljenih transakcij počasne naprave LS ali FS ne ovirajo hitre komunikacije naprav HS.

Začetek deljene transakcije sestavljajo dva do štiri paketi, žeton za deljeno transakcijo, »navaden« žeton, nato pa odvisno od smeri in vrste prenosa lahko sledijo še podatki in usklajevanje. Končanje deljene transakcije sestavljajo trije paketi, žeton za deljeno transakcijo, »navaden« žeton, nato pa ponovno odvisno od smeri in vrste prenosa lahko sledijo podatki ali usklajevanje.

5.2.4 Podatkovni paket

Podatkovni paket sestavljajo tri polja, polje PID nato do 8192 bitov podatkov odvisno od hitrosti vodila in vrste končne točke. Podatkovni paket ščiti 16 CRC bitov. Polje PID definira štiri različne vrste podatkovnih paketov. To sta paketa DATA0 in DATA1, ki se uporabljata v navezi s podatkovno sekvenčnim bitom (angl. data toggle synchronization). Za izohroni prenos se uporabljata še dva dodatna polja PID, to sta DATA2 in MDATA. Ta polja PID se uporablja za preverjanje pravilnega vrstnega reda podatkovnih transakcij (angl. PID sequencing) večkratnih transakcijah v istem okvirju.

5.2.5 Usklajevalni paket

Z razliko od drugih paketov usklajevalni paket (angl. handshake) vsebuje samo polje PID za sporočanje statusa poslanih podatkov. Z usklajevalnim paketom imamo nadzor nad pretokom podatkov. Ugotovimo ali so bili podatki uspešno sprejeti in ali je prišlo do zasičenja naprave USB. Tudi usklajevanje je omejeno z biti EOP in v primeru, da sprejemnik ne dobi bite EOP, se tako usklajevanje spregleda. Protokol USB definira pet različnih usklajevanj.

ACK usklajevanje pove, da so bili podatki poslani uspešno brez napak. Usklajevanje ACK se pri masivnih prenosih upošteva tudi kot sporočanje, da je naprava zmožna sprejemati podatke v primeru, da pred tem ni bila sposobna. Usklajevanje ACK lahko vrača gostitelj za vhodne transakcije ali pa naprava za izhodne, nastavitvene in transakcije PING. Gostitelj lahko pošilja vedno samo ACK in nobenega drugega usklajevanja. Tudi v primeru, ko dobi neveljavne podatke, le te zavrže, napravi pa ne vrne nobenega odziva.

NAK usklajevanje pove, da naprava od gostitelja ni bila zmožna sprejeti podatkov ali pa naprava nima podatkov za gostitelja (pri vhodnih transakcijah). Usklajevanje NAK lahko pošlje samo naprava pri podatkovni fazi za vhodne transakcije ali pri izhodnih PING transakcijah. Gostitelj ne more vrniti usklajevanja NAK.

STALL usklajevanje sporoča, da je prišlo do nepredvidenega dogodka v končni točki in ta ni sposobna prejeti ali poslati podatkov. Končna točka sama ne more preiti iz stanja STALL in je vedno potrebna gostiteljska intervencija. Gostitelj ne more nikoli vrniti usklajevanja STALL. Standard USB definira tudi protokolno usklajevanje STALL, ki se lahko izvede pri kontrolnih transakcijah. Protokolno usklajevanje STALL ne zahteva intervencijo gostitelja.

NYET usklajevanje se uporablja samo pri signalizaciji HS, kadar HS naprava ni pripravljena za sprejem podatkov, zaradi polnega medpomnilnika. NYET lahko vrne tudi vozlišče, ki izvaja deljeno transakcijo in naprava priključena na vozlišče še ni končala svoje transakcije z vozliščem.

ERR usklajevanje lahko sporočajo samo vozlišča HS. S tem usklajevanjem vozlišča HS prijavljajo napake na vodilih LS ali FS in se vračajo v sklopu deljene transakcije.

5.3 TRANSAKCIJE USB

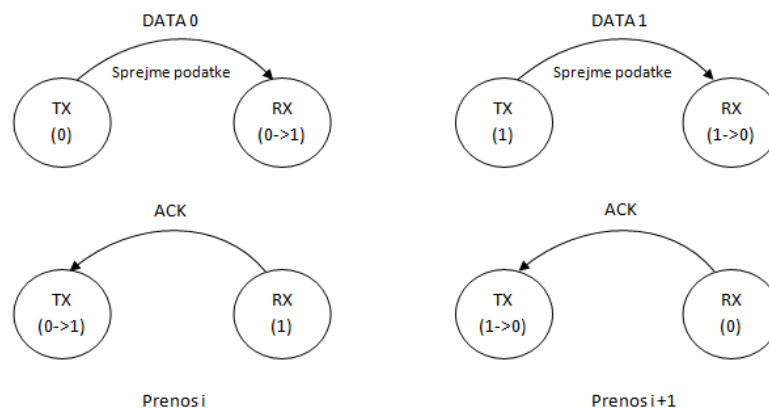
Kako bo potekala transakcija med gostiteljem in napravo je odvisno od vrste končne točke v napravi. USB ima definirane štiri vrste končnih točk in s tem tudi različne vrste prenosov podatkov:

- masivni prenos,
- prekinitveni prenos,
- izohroni prenos,
- kontrolni prenos;

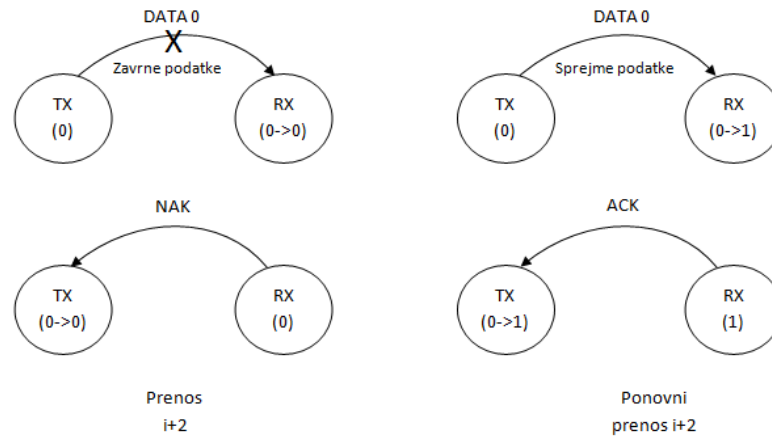
Vsaka vrsta prenosa ima značilne lastnosti, ki ustrezajo določeni napravi. Spletna kamera zahteva zanesljivo pasovno širino, po drugi strani pa računalniška miška zahteva zagotovljen dostopni čas. Lastnosti določenega prenosa diktirajo končne točke. Tako je naloga razvijalca naprave, da vgradi prave končne točke. Periodični FS prenosi (to so prekinitveni in izohroni) lahko rezervirajo za svoje potrebe do 90 % posameznega okvirja, medtem ko je pri HS ta omejitev postavljena na 80 % okvirja. S tem je v vsakem okvirju zagotovljen dostop do vodila tudi ostalim vrstam prenosa.

5.3.1 Kontrola zaporedja podatkovnih transakcij

USB vsebuje mehanizem, ki zagotovi, da se pri večkratnih transakcijah paketi dostavijo v pravilnem vrstnem redu. Pravilni vrstni red paketov zagotovi polje PID z DATA0, DATA1 oznako in podatkovno sekvenčni bit, ki se nahaja tako v sprejemniku kot tudi v oddajniku. Sprejemnik podatkovno sekvenčni bit proži samo v primeru, ko dobi nepoškodovan paket s pravilnim poljem PID, medtem ko oddajnik proži podatkovno sekvenčni bit samo takrat, ko za poslani podatkovni paket dobi veljavno usklajevanje ACK (*slika 13* in *14*). Pred podatkovno transakcijo morata imeti oba, tako sprejemnik kot oddajnik sinhroniziran podatkovno sekvenčni bit. Kontrolo sekvence podatkovnih transakcij uporabljajo vse vrste prenosa razen izohronega. Ta uporablja drugačen princip kontrole (poglavje 5.3.7).



Slika 13: Princip delovanja sekvence podatkovnih transakcij pri uspešnem prenosu podatkov.



Slika 14: Princip delovanja sekvence podatkovnih transakcij pri pokvarjenemu paketu DATA0.

5.3.2 Kontrola prenosa z protokolom PING

Naprave LS in FS, ki uporabljajo masivne ali kontrolne končne točke, lahko potrebujejo precej časa za obdelavo prispelih podatkov. Zato bo končna točka, ki še obdeluje prispele podatke odgovorila z usklajevanjem NAK, če proti njej sledijo dodatne izhodne transakcije. Saj za prihajajoče podatke trenutno nima dovolj prostora v medpomnilniku. V takem primeru gostitelj za določen čas preneha s pošiljanjem podatkov k tej končni točki. Problem nastane, ker v takem primeru gostitelj po nepotrebnem pošilja podatke, ki jih naprava trenutno ni sposobna prejeti. S tem pa se slabša izkoristek vodila še posebno, če imamo na vodilo priključenih več naprav.

Zato vse naprave HS podpirajo izboljšano usklajevanje NAK v kombinaciji s žetonom PING. V primeru, da končna točka trenutno ni sposobna prejeti podatkov, vrne NAK. Vsa nadaljnja povpraševanja »zasičene« končne točke gostitelj poizveduje z žetonom PING, vmes pa se lahko izvajajo druge transakcije. Na žeton PING lahko naprava odgovori na tri načine:

- NAK: končna točka še ni pripravljena za nove podatke,
- ACK: končna točka je pripravljena za nove podatke,
- NYET: končna točka je uspešno prejela podatke, toda ima trenutno poln medpomnilnik in zato ni pripravljena na nove podatke. Nadaljnja povpraševanja zasičene končne točke se izvajajo s žetonom PING.

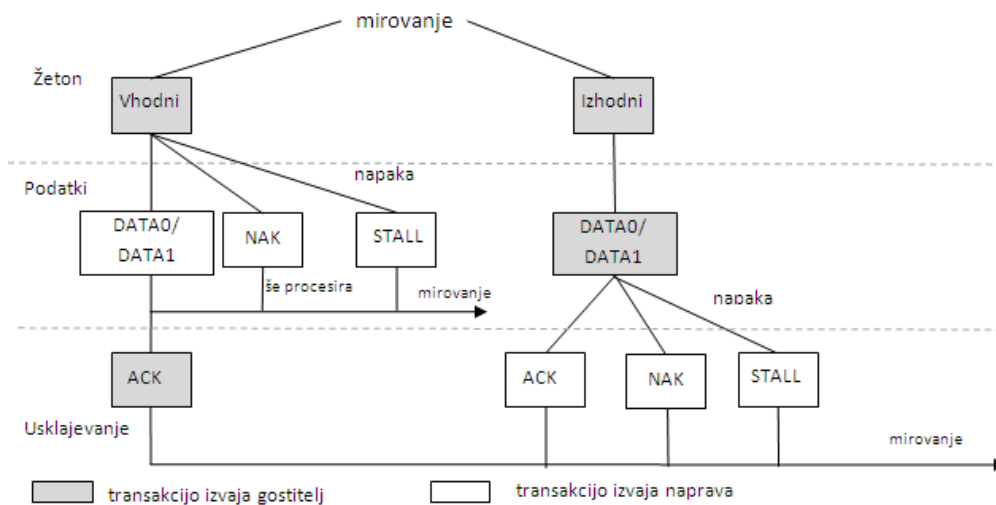
Žeton PING lahko naprava tudi ignorira. Kako pogosto se zasičena naprava odzove z usklajevanjem NAK, je zabeleženo v končni točki. Protokol PING podpirajo tudi vozlišča HS pri deljeni transakciji, medtem ko naprave LS in FS protokola PING ne podpirajo.

5.3.3 Masivni prenos

Masivne prenose uporabljajo naprave, ki prenašajo velike količine podatkov v precej variabilnem času. Masiven prenos porabi vso razpoložljivo pasovno širino, ki je niso zasedle druge vrste prenosov. V sistemih, kjer je veliko zahtevnih naprav USB (spletne kamere, prenosna podatkovna skladišča) lahko prenos traja dalj časa, toda podatki bodo prišli zanesljivo do sprejemnika. Za masivne prenose je značilno:

- dostop do vodila USB z vso razpoložljivo pasovno širino,
- v primeru napak se zahteva ponoven prenos,
- variabilni dostopni čas in variabilna pasovna širina.

Masivne cevi so enosmerne tokovne cevi. Zato v primeru, da potrebujemo dvosmerno masivno komunikacijo potrebujemo dve ločeni končni točki. Količina podatkov pri prenosih FS je omejena na 64 B na paket, medtem ko je največja količina podatkov za prenose HS omejena na 512 B na paket. Točna količina podatkov je zabeležena v posamezni končni točki. Standard USB uradno ne podpira masivne LS končne točke. Vsak uspešen masiven prenos se konča, ko sprejemnik dobi točno pričakovano količino podatkov ali kadar se prenese paket, ki vsebuje manjšo količino podatkov od maksimalne deklarirane vrednosti. Potek masivnih transakcij prikazuje *slika 15*.



Slika 15: Diagram komunikacije pri masivnih prenosih.

Masivni prenos ima najnižjo prioriteto med prenosi. Vse ostale oblike prenosa imajo prednost pri dostopu do vodila, zato masivni prenos nima nobenega časovnega zagotovila. V primeru, da imamo več masivnih prenosov, se vodilo deli po načelu pravičnosti.

Masivna končna točka nadzira pravilni vrstni red podatkovnih transakcij z uporabo podatkovno sekvenčnega bita. Če v končni točki pride do zastoja (zaradi nepredvidenega dogodka), vrne paket STALL, nato se odstrani vse čakajoče zahteve IRP za to končno točko. Za odstranitev zastoja je potrebna intervencija gostitelja, ki vzpostavi kontrolno cev do te naprave.

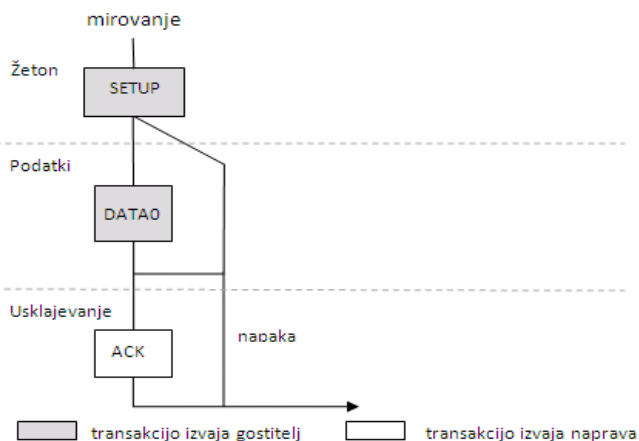
5.3.4 Kontrolni prenos

Kontrolni prenos se uporablja za konfiguriranje naprave USB. Vsaka naprava mora imeti implementirano privzeto kontrolno cev. To cev uporablja USB sistemska programska oprema za konfiguriranje, pregled statusa in kontrolno informacijo naprave. Kontrolni prenos se uporablja tudi za reševanje naprave, kadar je končna točka v zastoju. Naprava USB ima lahko tudi dodatne kontrolne končne točke za lastne potrebe.

Kontrolni prenosi LS in FS lahko porabijo največ 10 % posameznega okvirja, medtem ko lahko kontrolni prenos HS porabi do 20 % okvirja. Izohroni in prekinitveni prenosi imajo

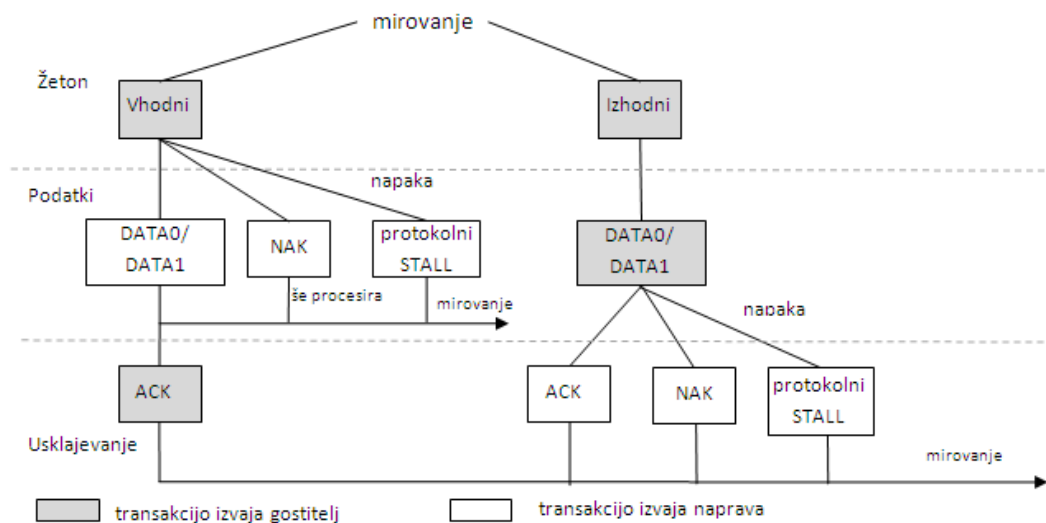
višjo prioriteto, zato se izvršijo pred kontrolnim prenosom. Kontrolni prenos deluje preko dvosmerne komunikacije skozi sporočilno cev. Ko je le ta vzpostavljena, se uporabljata dve končni točki, ena za vhodno in druga za izhodno transakcijo. Količina podatkov znotraj enega paketa je omejena na 8 B za naprave LS in do 64 B za naprave FS in HS. Kontrolne končne točke HS poznajo tudi protokol PING. Kontrolni prenos je običajno sestavljen s treh faz.

Nastavitvena faza se začne s žetonom SETUP, nato sledi ukaz, ki naj ga naprava izvede (*slika 16*). Podatkovni del nastavitvene faze vedno vsebuje DATA0 polje PID. Naprava ne sme nikoli zavrniti nastavitvenega paketa in mora vedno vrniti usklajevanje ACK. V primeru pokvarjenega paketa se ta zavrže in ne odda nobenega usklajevanja.



Slika 16: Format nastavitvene faze kontrolnega prenosa.

V podatkovni fazi kontrolnega prenosa se izvede različno število transakcij (*slika 17*), ki ustrezajo pravilu masivnega prenosa. Vse transakcije so enosmerne ali proti gostitelju ali proti napravi. Število prenesenih podatkov, velikost podatkovnega paketa, kot tudi smer transakcije je vnaprej določena z nastavitveno fazo. V primeru, da v podatkovni fazi sprejemnik dobi večjo ali manjšo količino podatkov kot je bilo dogovorjeno v nastavitveni fazi, bo končna točka v statusni fazi vrnila protokolno STALL usklajevanje. Podatkovna faza ni vedno prisotna, to je odvisno od ukaza, ki ga naprava izvede.



Slika 17: Diagram komunikacije v podatkovni fazi.

Statusna faza je zadnja faza kontrolnega prenosa, ki sporoči uspešnost izvedbe ukaza. Možni odzivi statusne faze so:

- ACK: potrdi uspešno izvedbo,
- NAK: naprava še izvaja ukaz,
- protokolni STALL: z razliko od »navadnega« STALL usklajevanja, ne ustavi končne točke, ampak sporoči da naprava ne podpira zahtevan ukaz ali pa je prišlo do napake pri izvajanju ukaza.

5.3.5 Prekinitveni prenos

Prekinitveni prenos je namenjen za javljanje posebnih dogodkov v napravi in se izvaja relativno redko, toda v strogih časovnih intervalih. Gostiteljski krmilnik periodično programsko izprašuje (angl. polling) prekinitvene končne točke, če imajo pripravljen prekinitveni zahtevek. Prekinitveni prenosi zahtevajo:

- prednostno obravnavanje preko prekinitvene cevi,
- zanesljivo dostavo podatkov,
- v primeru napake na vodilu se prenos ponovi v naslednji periodi.

Za prekinitveni prenos se uporablja tokovno cev, zato je vedno enosmerna. Prekinitvena FS končna točka omogoča največjo količino podatkov 64 B na paket, končna točka HS pa omogoča prenos do 1024 B na paket. Prekinitveni prenos je mogoč tudi na končnih točkah LS, pri tem je količina podatkov omejena z 8 B na paket. Določen prekinitven prenos lahko zahteva več transakcij, ki jih nato izvaja periodično. Prekinitvena končna točka HS lahko v enem okvirju izvede do tri transakcije, torej lahko prenese do 3072 B podatkov na okvir. Prekinitvena cev se vzpostavi samo v primeru, če je na vodilu dovolj časa za izvedbo določene transakcije. Konča pa se takrat, ko se prenese točno pričakovana količina podatkov ali v primeru, da je količina podatkov v paketu manjša od največje (v končni točki) določene količine podatkov. Naprava, ki nima javljenega prekinitvenega dogodka, to sporoči z usklajevanjem NAK. Intervali programskega izpraševanja LS in FS prekinitvene končne točke so določeni z $1\text{ms} * (2^{\text{interval}-1})$ oziroma za HS z $1\mu\text{s} * (2^{\text{interval}-1})$, pri tem je *interval* lahko vrednost od 1 do 16.

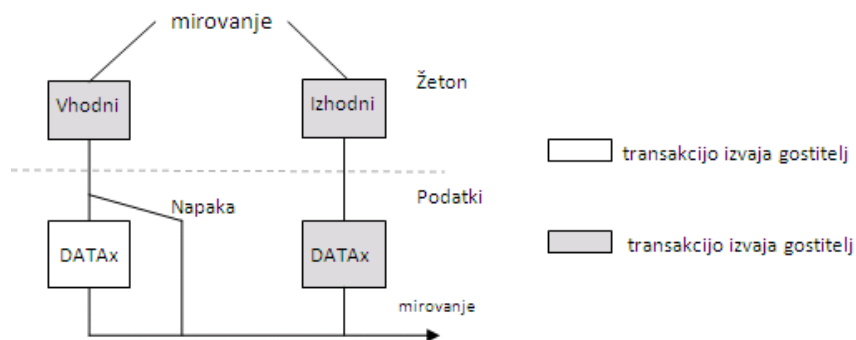
Prekinitveni prenosi se izvajajo po istem pravilu kot masivni prenosi (*slika 15*). Odjemalec na gostitelju mora izrecno zahtevati programsko izpraševanje določene prekinitvene končne točke preko zahtevka IRP. V primeru, da tega zahtevka gostiteljski krmilnik ne dobi, ne bo izpraševal končne točke, kljub podanemu intervalu.

5.3.6 Izohroni prenos

Izohroni prenos uporabljajo naprave, ki delujejo z neprekinjenim podatkovnim tokom. Najbolj tipični predstavnik je kamera, ki neprekinjeno prenaša avdio/video podatke. Značilnosti izohronega prenosa so:

- zagotovljen dostopni čas do vodila,
- konstanten tok podatkov dokler so podatki na voljo,
- v primeru napak med prenosom se ne zahteva ponoven prenos.

Izohrona cev je vedno enosmerna. Če potrebujemo dvosmerno izohrono komunikacijo, je potrebno uporabiti dve ločeni cevi z dvema izohronima končnima točkama. Končne točke LS ne poznajo izohronega prenosa. Izohroni FS prenos omogoča največjo količino podatkov 1023 B na paket ali 1024 B na paket pri prenosu HS. Dejanska količina podatkov na paket je zabeležena v končni točki. USB sistemska programska oprema uporablja informacijo o količini podatkov med vzpostavljanjem cevi. V primeru, da na vodilu ni dovolj pasovne širine za izohrono transakcijo, se cev ne vzpostavi. Rezervacija vodila za izohroni prenos je definirana z isto enačbo, kot je definiran interval prekinitvenega prenosa. Izohrona HS končna točka omogoča do tri transakcije na okvir. Izohrone transakcije omogočajo variabilno količino podatkov v paketih, tudi manjšo količino, kot je deklarirano v končni točki.



Slika 18: Diagram komunikacije pri izohronem prenosu

Z razliko od drugih vrst prenosa izohroni prenos ne pozna usklajevanja (slika 18). Za pokvarjene pakete se ne zahteva ponovni prenos, saj je neprekinjen pretok podatkov pomembnejši. Izohroni prenos vsebuje tudi mehanizem za preverjanje pravilnega nizanja podatkovnih transakcij, tako lahko zazna ali je bil določen paket izgubljen.

5.3.7 Pravilno nizanje izohronih paketov

Pri izohronih transakcijah se za ugotavljanje pravilnega vrstnega reda paketov znotraj okvirja uporablja polje PID. Polje PID v izohronih podatkovnih paketih ima lahko vrednost DATA0, DATA1, DATA2 za vhodne transakcije ali vse prej naštetje in dodatno MDATA za izhodne transakcije.

Za vhodni izohroni HS prenos s tremi transakcijami v enem okvirju si posamezna polja PID sledijo DATA2 nato DATA1 in na koncu DATA0. Ker je bilo zahtevano, da se bodo izvedle tri transakcije, mora končna točka vedno najprej poslati PID DATA2 in s tem gostitelj ve, da bosta sledili še dve dodatni transakciji znotraj istega okvirja. V primeru, da zahtevamo samo eno transakcijo ima polje PID znotraj paketa vrednost DATA0. Če sprejemnik zazna napačni vrstni red ali manjkajoče polje PID, se vse nadaljnje transakcije znotraj tega okvirja zavrže. Ravno tako se po tem, ko sprejemnik prebere polje PID DATA0 zavrže vse morebitne naslednje izohrone transakcije znotraj tega okvirja.

Za izhodne izohrone transakcije je princip pošiljanja malo drugačen. Če uporabljamo dve transakciji v okvirju, je polje PID prve transakcije MDATA druge pa DATA1. Za tri transakcije sta prvi dve transakciji MDATA zadnja pa DATA2. Vidimo lahko, da polje PID

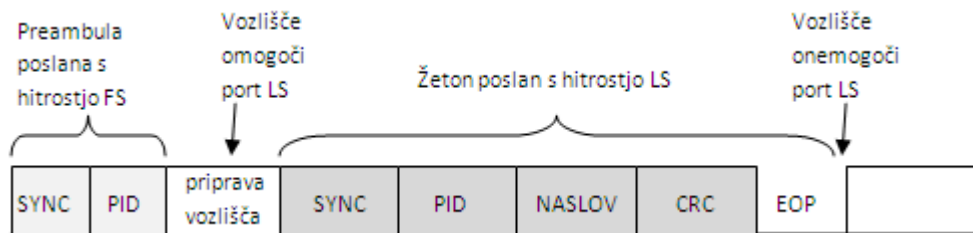
zadnje transakcije pove, koliko paketov MDATA je bilo poslanih. V primeru ene same transakcije se uporablja polje PID DATA0.

Glede na to, da izohroni prenos ne uporablja usklajevanja, lahko z zgoraj opisano logiko ugotovimo ali je vrstni red paketov pravilen.

5.3.8 Low-speed in deljene transakcije

Transakcije LS se vršijo s hitrostjo 1,5Mb/s, kar je dovolj za nezahtevne naprave. Gostitelj izključi izhodna vrata (angl. downstream port) na vozlišču, kjer so priključene naprave LS, ko komunicira z napravami FS ali HS. To zato, da se onemogoči napravam LS morebiten odziv na signalizacijo FS ali HS. Ko želi gostitelj komunicirati z napravo LS mora najprej poslati preambulo. Preambula je sestavljena iz polja SYNC in PID polja PRE (preambula). Preambulo gostitelj vedno pošlje s hitrostjo FS do vozlišča, kjer je priključena naprava LS. Samo vozlišča znajo prebrati PID polje PRE, pri tem vozlišče omogoči vrata LS za naprave, ki delujejo z hitrostjo LS. Sedaj lahko vozlišče pošlje žeton (slika 19), podatkovni paket in usklajevanje v hitrosti LS. Konec paketa zaznamujejo biti EOP, pri tem se zapre vrata LS na vozlišču. Protokol za signalizacijo LS ima še nekatere omejitve:

- količina podatkov v paketu je omejena na 8 B,
- podpora samo za prekinitvene in kontrolne prenose,
- naprave LS ne morejo sprejeti paketov SOF.



Slika 19: Primer poslanega žetona s hitrostjo LS preko vozlišča FS ali HS.

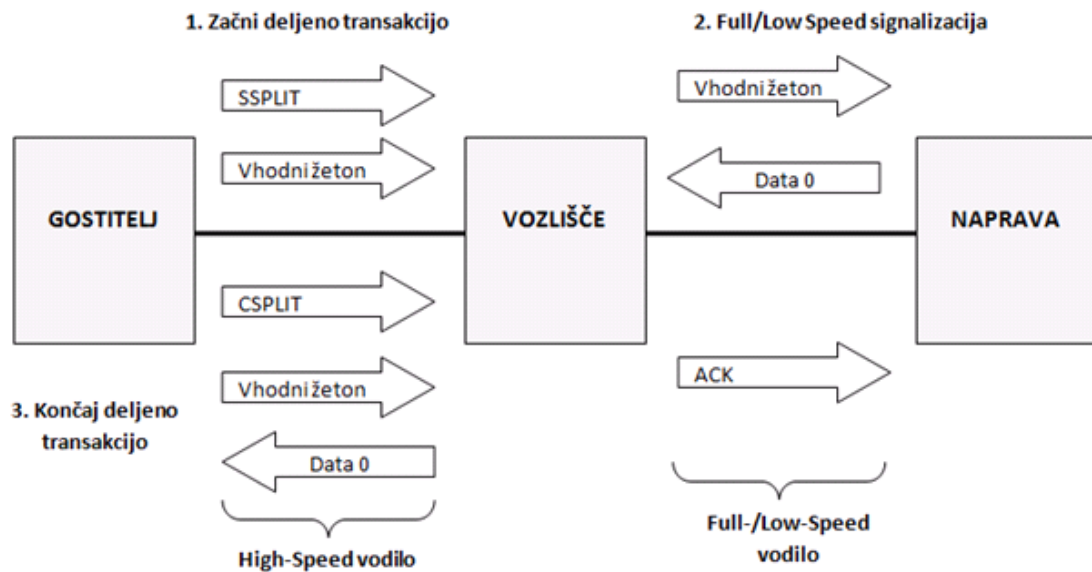
Deljene transakcije je standard USB 2.0 uvedel, da naprave LS in FS s svojo počasno komunikacijo ne bi ovirale komunikacije HS. Deljena transakcija je sestavljena z dveh žetonov:

- začetek deljene transakcije: za pripravo vozlišča na komunikacijo LS in FS,
- končanje deljene transakcije: za prevzem podatkov, ki jih je naprava LS ali FS dostavila na vozlišče.

Delovanje deljene transakcije si lahko pogledamo na primeru (slika 20). USB ključ imamo priključen neposredno na korensko vozlišče, tipkovnica pa je povezana preko dodatnega vozlišča.

Krmilnik, vozlišče in USB ključ delujejo s komunikacijo HS. Na vozlišče pa imamo priključeno tipkovnico, ki deluje s komunikacijo LS. Po določenem času, ko gostitelj želi pridobiti podatke od tipkovnice, pošlje žeton za začetek deljene transakcije in tudi »navaden« vhodni žeton proti vozlišču. Po tem prenosu ima gostitelj čas za transakcije z USB ključem, saj se vozlišče pripravlja na komunikacijo LS. Nekoliko kasneje vozlišče pošlje žeton v signalizaciji LS proti tipkovnici. Naprava odgovori s podatki, vozlišče pa odgovori z

usklajevanjem. Po tem, ko je lahko gostitelj z USB ključem ponovno opravil nekaj transakcij, pošlje proti vozlišču žeton za končanje deljene transakcije in »navaden« žeton za prevzem podatkov, ki jih je v vozlišče poslala tipkovnica.



Slika 20: Primer prikazuje vhodno transakcijo s pomočjo deljene transakcije.

6 KONFIGURIRANJE NAPRAVE USB

Ko se na vodilo priključi ali izključi napravo USB, se izvede enumeracija vodila, s katerim gostitelj ugotovi kakšna naprava je bila priključena na vodilo. Vozlišče preko statusnega registra vodi evidenco na katera vrata so priključene naprave. Vrata na katere ni priključene naprave so onemogočena. Pri vklopu naprave se izvede naslednji postopek:

1. Ko priključimo napravo na vozlišče, dobi osnovno napajanje. Vrata na katera je priključena naprava so še vedno onemogočena.
2. Gostitelj periodično bere statusni register in s tem ugotovi na katera vrata je priključena nova naprava.
3. Gostitelj počaka 100 ms, da napajanje proti napravi postane stabilno in omogoči vrata na katera je priključena naprava. Po tem se naprava napaja z do 100 mA toka. Vse številce in registre ima nastavljene na privzeto vrednost.
4. Do naprave se vzpostavi privzeta kontrolna cev preko privzetega naslova, rezerviranega za napravo, ki še ni bila konfigurirana. Gostitelj sedaj poizveduje v napravi, kolikšna je največja količina podatkov na paket, ki ga bo uporabljala privzeta kontrolna cev.
5. Gostitelj napravi dodeli unikatni naslov.
6. V zadnjem koraku gostitelj prebere vse možne konfiguracije naprave preko konfiguracijskega deskriptorja, ki se nahaja v napravi. S tem gostitelj izve, kakšne so zmogljivosti in potrebe naprave. Če naprava podpira več različnih konfiguracij, gostitelj izbere tisto, ki najbolje ustreza trenutnemu stanju. Določena naprava lahko deluje z zunanjim napajanjem ali pa samo z napajanjem preko vodila. Glede na pridobljeno napajanje lahko deluje drugače oziroma z drugo konfiguracijo.

Po tem postopku je enumeracija vodila končana in naprava je pripravljena za uporabo. Pri odklopu naprave se ponovno spremeni stanje v statusnem registru, kar prebere gostitelj. Z odklopom naprave gostitelj vrata onemogoči in posodobi informacijo o stanju topologije omrežja.

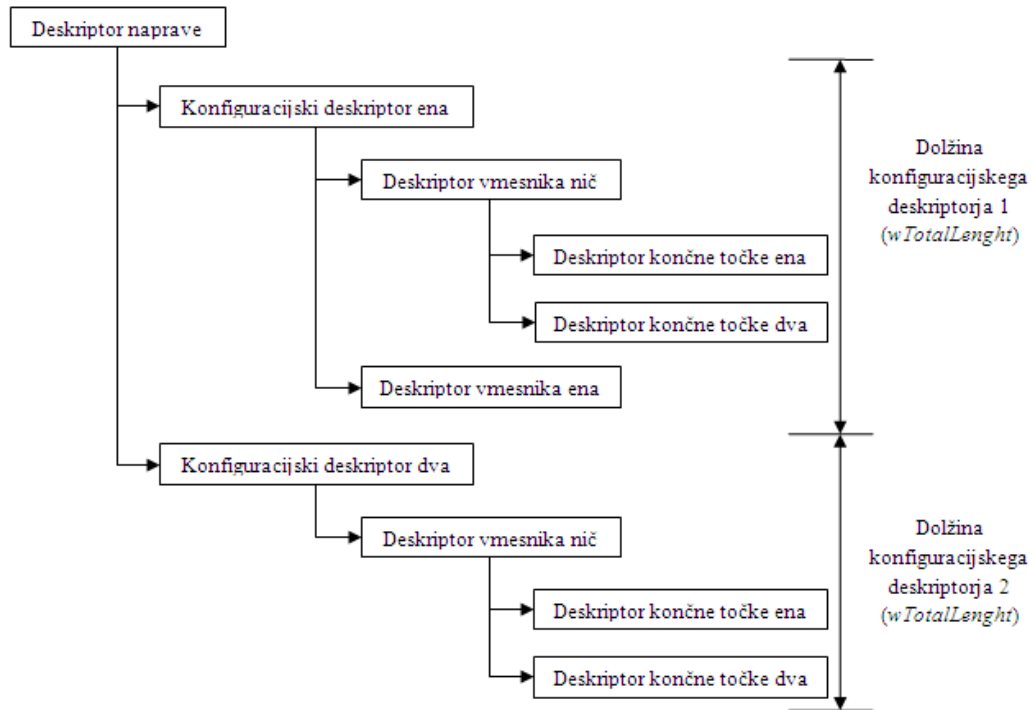
6.1 DESKRIPTORJI USB

Vse naprave USB vsebujejo hierarhijo deskriptorjev (*slika 21*), ki so podatkovna struktura s točno določenim formatom. Deskriptorji se uporabljajo za predstavitev naprave gostitelju. Med drugim opišejo kakšna je naprava (miška, tiskalnik...), kdo je proizvajalec, lastnosti in število končnih točk itd. Vsak deskriptor se začne z 8 bitnim poljem *bLength*, ki sporoči skupno dolžino deskriptorja, nato pa sledi 8 bitno polje, ki sporoči vrsto deskriptorja. Nadaljnja polja pa so odvisna od vrste deskriptorja. Štirje deskriptorji so vedno prisotni v vsaki napravi USB:

- deskriptor naprave (angl. device descriptor),
- konfiguracijski deskriptor (angl. configuration descriptor),
- deskriptor vmesnika (angl. interface descriptor),
- deskriptor končne točke (angl. endpoint descriptor).

Definirani pa so tudi opcijski deskriptorji:

- Tekstovni deskriptor (angl. string descriptor),
- Device Qualifier deskriptor,
- Other Speed configuration deskriptor.



Slika 21: Hierarhija deskriptorjev. Dolžina konfiguracijskega deskriptorja vključuje tudi podrejene deskriptorje.

6.1.1 Deskriptor naprave

Deskriptor naprave predstavi napravo USB kot celoto. Vsaka naprava USB lahko vsebuje samo en deskriptor naprave. Ta predstavi osnovno informacijo o proizvajalcu, ID naprave, število možnih konfiguracij... Celoten deskriptor naprave sestavlja 17 Bajtov podatkov in vsebuje 12 polj.

bcdUSB polje: Opisuje katera je najvišja verzija standarda USB oziroma hitrost signalizacije, ki jo podpira naprava. Vsebina polja je zapisana v binarno kodirani desetiški kodi s formatom JJMN. Pri tem je JJ verzija standarda USB, MN pa je podverzija standarda. Tako bo naprava USB 1.1 kodirana z 0x0110 in USB 2.0 z 0x0200.

bDeviceClass, **bDeviceSubClass**, **dDeviceProtokol** polja: Ta polja se uporabljajo za umestitev naprave v določen razred. S tem poljem si pomaga operacijski sistem, da najde primeren gonilnik za napravo. Razredov je 19, v kateri razred spada naprava pa določa USB-IF. USB ključ na primer spada v razred sekundarnih pomnilnikov (ang. mass storage) in ima oznako 08 (šestnajstiško). Večina naprav se raje predstavi na nivoju vmesnika, s tem dosežemo, da lahko naprava pripada različnim razredom v primeru, da je multifunkcijska.

bMaxPacketSize polje: Sporoči največjo velikost paketa za končno točko nič s katero se začne prva komunikacija. Vse naprave morajo imeti implementirano končno točko nič, saj se ta uporablja takrat, ko naprava še ni konfigurirana in druge končne točke še niso dostopne. Možne velikosti podatkov so 8, 16, 32, 64 bajtov na paket.

idVendor, idProduct polja: V prvem polju je zapisan ID proizvajalca naprave. ID proizvajalca dodeli USB-IF. V *idProduct* pa je zapisan ID naprave, ki jo dodeli proizvajalec.

bcdDevice polje: V polju je zabeležena verzija naprave, za kar je odgovoren proizvajalec. Polje je kodirano v binarno desetiški kodi.

iManufacturer, iProduct, iSerialNumber polja: Vsebujejo kazalec, ki kaže na tekstovni deskriptor. V tem deskriptorju je opisana naprava v človeku razumljivem besedilu. V primeru, da se proizvajalec odloči, da ne bo imel opisa naprave, morajo polja imeti vrednost »0«.

bNumConfiguration polje: To polje vrne število možnih konfiguracij, ki jih naprava podpira.

6.1.2 Konfiguracijski deskriptor

Vsaka naprava ima lahko različno število konfiguracij, čeprav redke naprave uporabljajo več kot eno konfiguracijo. V tem deskriptorju je opisan način napajanja, maksimalna poraba naprave, število vmesnikov. Naprava ima lahko znotraj konfiguracije različne vmesnike, ki vsebujejo različne končne točke (*slika 21*). Naprava ima lahko zunanje napajanje, lahko pa deluje tudi z napajanjem preko vodila. V tem primeru bi naprava vsebovala dva konfiguracijska deskriptorja in glede na to katero konfiguracijo bi naprava trenutno uporabljala, bi bila funkcionalnost naprave nekoliko drugačna. Med enumeracijo gostitelj pregleda vse možne konfiguracije in z ukazom *setConfiguration* izbere določeno konfiguracijo. Konfiguracijski deskriptor vsebuje šest polj.

wTotalLength polje: To polje vrne velikost vseh deskriptorjev, ki so znotraj posamezne konfiguracije. Tako se upošteva tudi velikost vseh vmesniških deskriptorjev, kot tudi deskriptorjev končnih točk. Pri branju konfiguracijskega deskriptorja se vrne celotna hierarhija deskriptorjev od konfiguracijskega deskriptorja navzdol.

bNumInterface polje: Prikaže število vmesnikov za določeno konfiguracijo.

bConfigurationValue polje: To polje se uporablja pri ukazu *SetConfiguration* za izbiro določene konfiguracije.

iConfiguration polje: Kazalec na tekstovni deskriptor, ki opisuje konfiguracijo v človeku berljivi obliki.

bmAttributes polje: Določa napajalne karakteristike naprave. Ali ima naprava zunanje napajanje ali se napaja preko vodila. Polje tudi pove ali naprava podpira funkcijo oddaljenega prebujanja (angl. remote wakeup).

bMaxPower polje: Definira tok, ki ga naprava potrebuje preko vodila za normalno delovanje tudi, če gre za požrešno napravo. Ena enota se šteje za 2 mA toka, največji možni tok, ki ga lahko vodilo USB priskrbi posamezni napravi je 500 mA.

6.1.3 Deskriptor vmesnika

Ta deskriptor opisuje določen vmesnik znotraj konfiguracije. Vmesnikov je lahko več. Vsak vsebuje skupino končnih točk, ki predstavlja del funkcionalnosti naprave. Naprava USB ima lahko aktivnih več vmesnikov naenkrat z razliko od konfiguracijskega deskriptorja. Z vmesniki lahko gostitelj izvaja določene naloge na napravi USB. V primeru multifunkcijske naprave, ki združuje fax, skener in tiskalnik, bi se vmesnik ena uporabljal za fax, vmesnik dva za skener in tretji vmesnik za tiskalnik.

Končne točke si različni vmesniki ne morejo deliti. Lahko pa vmesnik vsebuje alternativne nastavitve. Te nastavitve lahko velevajo vmesniku, da uporablja različne vrste končnih točk, kot tudi število končnih točk. Torej imamo lahko dva vmesnika z isto oznako, toda z različnimi alternativnimi nastavitvami in s tem tudi, različno funkcionalnostjo.

Do deskriptorja vmesnika ne moremo neposredno dostopati z ukazom *getDescriptor()*. Njegova vsebina se vrne preko ukaza *getConfiguration()*, torej kot del konfiguracijskega deskriptorja. Poleg polja, ki pove dolžino deskriptorja vmesnika, vsebuje še sedem polj.

bInterfaceNumber polje: Vrne indeks na izbrani deskriptor vmesnika. Če imamo samo en vmesnik, je označen z indeksom »0«. Indeks pa narašča za vsak dodatni vmesnik.

bAlternateSetting polje: Določa alternativne nastavitve vmesnika. Če imamo več alternativnih nastavitvev, se označujejo od »0« naprej. Te nastavitve se nato izbere z ukazom *SetInterface*.

bNumEndpoints polje: Prikaže število končnih točk, ki pripadajo določenemu vmesniku. Končna točka nič se ne prišteva k *bNumEndpoints* polju.

bInterfaceClass, ***bInterfaceSubClass***, ***bInterfaceProtocol*** polja: Prikaže razrede v katere spada naprava. S to funkcionalnostjo lahko večina naprav uporablja privzete »razredne« gonilnike in s tem odpade potreba po pisanju določenih gonilnikov.

iInterface polje: Kazalec na tekstovni deskriptor, ki opisuje vmesnik v človeku berljivi obliki.

6.1.4 Deskriptor končne točke

Z deskriptorjem končne točke dobi gostitelj informacijo o smeri in vrsti prenosa, količini podatkov na paket in drugih zahtevah, ki jih ima določena končna točka. Tako kot deskriptor vmesnika tudi ta deskriptor ne moremo klicati neposredno, ampak dobimo deskriptor končne točke preko konfiguracijskega deskriptorja. Deskriptor končne točke vsebuje štiri pomembna polja.

bEndpointAddress polje: Pove številko končne točke in tudi smer prenosa podatkov, torej ali je vhodna ali izhodna končna točka.

bmAttributes polje: Definira vrsto končne točke (masivni, kontrolni, prekinitveni ali isohronski).

wMaxPacketSize polje: Definira največjo možno količino podatkov v paketu, ki ga lahko končna točka sprejme ali pošlje. Če gre za izohrono ali prekinitveno HS končno točko, to polje pove tudi največje možno število transakcij znotraj okvirja.

bInterval polje: Deklarira interval programskega izpraševanja za prekinitvene in izohrone prenose, ter interval usklajevanja NAK pri poizvedbi s paketom PING pri masivni končni točki.

6.1.5 Opcijski deskriptor

Tekstovni deskriptor ni obvezen del vsake naprave USB. Je pa dobrodošel, saj prinaša informacije o napravi, zapisane v človeku berljivi obliki. V primeru, da ga naprava ne vsebuje, morajo imeti vse reference, ki kažejo na tekstovni deskriptor, vrednost nič. Besedilo je zapisano v abecedi Unicode, naprava pa lahko vsebuje informacijo v različnih jezikih.

Other Speed Configuration deskriptor vsebujejo samo naprave HS. V njemu je zapisano kako naj naprava HS deluje, ko komunikacija poteka v signalizaciji LS ali FS. Struktura deskriptorja je enaka konfiguracijskemu deskriptorju.

Device qualifier deskriptor vsebujejo samo naprave HS. Opisuje kakšne sposobnosti ima naprava HS v primeru, da deluje z hitrostjo LS ali FS, npr. največji paket, ki ga lahko pošlje pri LS ali FS signalizaciji.

6.2 ZAHTEVKI GOSTITELJA

Vse naprave USB se morajo odzvati na zahteve gostitelja preko privzete kontrolne cevi. Te zahteve se pošiljajo s kontrolnim prenosom. Zahteve in parametri se napravi pošljejo v nastavitvenem paketu. Vsak zahtev se začne z nastavitvenim paketom dolžine 8 bajtov. Nastavitveni paket sestavlja pet polj.

bmRequestType polje: dolžine 8 bitov deklarira smer prenosa v podatkovni fazi, proti gostitelju ali proti napravi. 1. in 2. bit pove tudi kdo je prejemnik končna točka, vmesnik ali naprava. 7. in 6. bit razkrije vrsto zahteve, standardna, razredna ali tovarniška. Razredne zahteve so značilne za naprave, ki spadajo v določen razred. Tako imajo naprave, ki spadajo v razred sekundarnih pomnilnikov drugačne razredne zahteve od naprav, ki spadajo v razred tiskalnikov. Tovarniški ukazi, ki jih implementira razvijalec sam, pa so značilni samo za določeno napravo. Standardne ukaze pa morajo podpirati vse naprave USB.

bRequest polje: deklarira dejanski ukaz, ki naj se izvede na napravi.

wValue in ***wIndex*** polje: Uporablja se za prenos parametrov k napravi in katera končna točka ali vmesnik naj izvrši ukaz. Vrednost vsebine se spreminja glede na ukaz, ki se bo izvedel.

wLength polje: Sporoča dolžino poslanih podatkov v drugi fazi kontrolnega prenosa torej podatkovni fazi. Proti gostitelju se ne sme nikoli prenesti večja količina podatkov, kot je deklarirano, lahko pa se prenese manj. Proti napravi pa se mora poslati vedno točno toliko podatkov kot je bilo deklarirano. Dolžina podatkov se šteje v bajtih. V primeru, da ukazu ne sledi podatkovna faza, je polje ***wLength*** nastavljeno na »0«.

6.2.1 Standardni ukazi gostitelja

Vse naprave USB se morajo znati odzvati na standardne ukaze gostitelja, tudi v primeru, ko še niso konfigurirane. S temi operacijami lahko beremo stanja naprave in jih tudi poljubno nastavljamo, da delujejo v skladu z našimi zahtevami.

Vsi ukazi z izjemo *SetAddress()* se začnejo izvajati takoj za tem, ko naprava vrne ACK po prejetju nastavitvenega paketa. Zahtevki pa se mora obdelati preden se uspešno zaključi statusna faza. USB postavlja zgornjo mejo 5 sekund za izvedbo ukaza, čeprav se mora večina ukazov izvršiti v precej krajšem času.

Standardni ukazi gostitelja, ki ne vsebujejo podatkovno, fazo morajo biti končani v 50 ms. Čas se šteje od prejete zahtevke do uspešno vrnjene statusne faze. Za standardne zahtevke, ki vsebujejo podatkovno fazo namenjeno gostitelju, morajo poslati prvi podatkovni paket v roku 500 ms. Ko je podatkovna faza končana, ima naprava nato 50 ms časa, da uspešno konča statusno fazo. Zahtevku, ki mu sledi podatkovna faza proti napravi se upošteva 5 sekundna omejitev do uspešne izvedbe statusne faze. V primeru, da se katera operacija ne izvede v predpisanem času, se sproži časovna kontrola. Standard USB 2.0 definira osem standardnih ukazov gostitelja.

get_status: naprava, kot odziv na ta ukaz vrne podatek dolžine 2 bajta v podatkovni fazi. Pomembna sta samo prva dva bita. Bit 0 pove ali se naprava napaja preko vodila ali preko zunanjega napajanja, bit 1 pa pove ali ima omogočeno funkcijo oddaljenega prebujanja. Ostali biti so rezervirani in zato postavljeni na 0.

clear_feature in ***set_feature***: Ukaz je namenjen za omogočanje ali onemogočanje funkcije oddaljenega prebujanja in pa za testni način delovanja. S testnim načinom delovanja lahko preverimo delovanje določenih delov naprave. Testiranje podatkovne linije J in K, pošiljanje testnega paketa. Specifične teste za napravo lahko implementira tudi razvijalec.

set_address: Ukaz se uporablja med enumeracijo vodila, s katerim gostitelj določi unikaten naslov napravi. Naslov naprave je deklariran v polju *wValue* katerega vrednost je lahko največ 127. To je edini ukaz, ki se izvede šele po uspešni vrnitvi statusne faze. Vsi drugi ukazi se morajo izvesti pred statusno fazo.

get_descriptor in ***set_descriptor***: S tem ukazom nastavimo ali dobimo vrnjen zahtevan deskriptor. Vedeti je potrebno, da z ukazom ne moremo dobiti neposredno deskriptorja tipa vmesnik ali tipa končna točka. Ta dva deskriptorja lahko dobimo vrnjena preko konfiguracijskega deskriptorja, ki vrne celotno hierarhijo deskriptorjev.

get_configuration in ***set_configuration***: s tem ukazom dobimo v podatkovni fazi vrnjen 1 B podatkov. V primeru, da je vrednost vrnjenih podatkov »0« pomeni, da naprava ni konfigurirana. Z ukazom *set_configuration* nastavimo določen konfiguracijski deskriptor s katerim želimo konfigurirati napravo. Če je konfiguracij naprave več, uporabimo polje *wValue*, s katerim izberemo določeno konfiguracijo.

Poleg opisanih standardnih ukazov naprave, standard USB definira še standardne zahtevke vmesnika in končne točke. S slednjim lahko z ukazom *get_status* pregledamo stanje končne točke ali je končna točka v zastoju. V primeru, da želimo končno točko namenoma ustaviti ali jo ponovno aktivirati, lahko to storimo z ukazom *set/get_feature*. S standardnim vmesniškim ukazom se najbolj pogosto uporablja ukaz *set/get_interface* s katerim določamo vmesnik. Če jih imamo več, izberemo določenega z poljem *wIndex*. Z ukazom *set_interface* lahko s poljem *wValue* izberemo tudi alternativno nastavitvev določenega vmesnika.

Ne glede na to, da so opisani zahtevki standardni za vse naprave, se lahko pripeti, da v polju *wValue* ali *wIndex* podamo vrednost, ki je naprava ne more izvesti. Na primer, da naprava podpira samo eno konfiguracijo v polju *wValue* pa podamo zahtevo za drugo konfiguracijo. V takem primeru pride do napake v zahtevi (angl. error request) in končna točka bo v podatkovni fazi vrnila usklajevanje STALL. Naprava vrne napako v zahtevi za vsak ukaz, katerega ne more uspešno izvesti.

7 USB PROGRAMSKA OPREMA

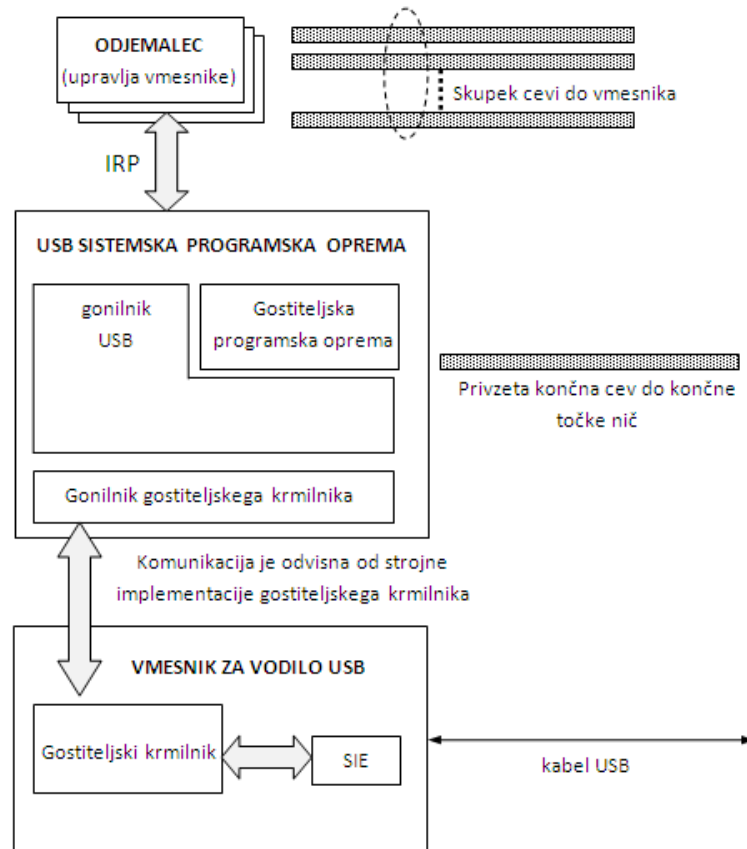
Gostiteljska programska oprema prinaša vmesnik med odjemalčevim gonilnikom in napravo s katero želi odjemalec komunicirati. Odjemalec se ne zaveda implementacije USB in nima podatkov o zmožnosti ali omejitvah standarda USB. Zato vse zahteve, ki jih pošlje odjemalec, »prestreže« USB programska oprema in jih prilagodi zahtevam protokola USB. Gostitelj USB upravlja z naslednjimi dejavnostmi:

- zaznavanje priključenih in izključenih naprav,
- upravljanje s kontrolnim prenosom med gostiteljem in napravo USB pri enumeraciji vodila in konfiguraciji naprave,
- upravljanje prenosa podatkov med gostiteljem in napravo USB,
- zbiranje statistik aktivnosti in statusov pri prenosu podatkov med gostiteljem in napravo USB,
- upravljanje z električno porabo.

7.1 KLJUČNE PLASTI GOSTITELJA

Gostitelja USB sestavljajo tri plasti (*slika 22*):

- USB vmesnik za vodilo (angl. USB bus interface),
- USB sistemska programska oprema,
- odjemalec.



Slika 22: Shema plasti gostitelja USB.

USB vmesnik za vodilo prestra zahtevke IRP, ki jih pošlje USB sistemska programska oprema in jih pretvori v transakcije. Transakcijo se nato prilagodi v ustrezne logične nivoje in se jo serijsko pošlje preko kabla do naprave USB. USB vmesnik za vodilo sestavlja:

- serijski vmesnik (angl. serial interface engine, SIE), ki se nahaja tudi v napravi USB in skrbi za serijski prenos podatkov po vodilu.
- gostiteljski krmilnik je integrirano vezje, ki skrbi za razvrščanje transakcij na vodilo.

USB sistemska programska oprema poskrbi za konfiguriranje novo priključenih naprav in za upravljanje z viri, kot so npr. napajanje in pasovna širina. Ko odjemalec zahteva prenos podatkov za določeno napravo, to zahtevo prejme USB sistemska programska oprema in jo prilagodi zahtevam gostiteljskega krmilnika. USB sistemska programska oprema je sestavljena iz gonilnika USB in gonilnika gostiteljskega krmilnika. Opcijsko pa lahko vsebuje tudi gostiteljsko programsko opremo, ki predstavlja dodaten programski vmesnik med gonilnikom USB in odjemalcem.

Odjemalec vsebuje gonilnik periferne naprave. Ti gonilniki so pobudniki transakcij po vodilu USB in so potrebni, da lahko naprava opravlja svojo funkcijo. Odjemalec lahko uporablja generične razredne gonilnike, ki so lahko vgrajeni v operacijski sistem, ali razvijalec napiše lastne gonilnike za svojo napravo. Če želimo dostopati do naprave, se to ne glede na vrsto gonilnika izvede preko zahtevka IRP, ki se ga posreduje proti USB sistemske programske opremi.

7.2 GONILNIK USB

Do želene naprave USB lahko dostopamo samo preko gonilnika USB. Implementacija gonilnika USB je specifična za posamezen operacijski sistem. Gonilnik USB sodeluje v naslednjih funkcijah:

- vodenje konfiguracije,
- upravljanje z viri,
- vodenje podatkovnega prenosa,
- nudenje storitev odjemalcu preko vmesnika gonilnika USB.

USB sistemska programska oprema podpira avtomatično konfiguriranje naprav USB. Pri konfiguriranju sodelujejo različne programske komponente. Pri priklopu naprave dobi vozliščni gonilnik obvestilo, da se je priključila nova naprava. Vozliščni gonilnik s tem obvestilom zahteva od gonilnika USB identifikator naprave. Gonilnik USB pripravi privzeto cev za priključeno napravo USB in vrne identifikator naprave k gonilniku vozlišča. Preden je naprava pripravljena za uporabo se morajo izvršiti tri konfiguracije:

- konfiguracija naprave: to obsega dostopanje do deskriptorjev naprave, ugotavljanje potrebnih virov za napravo, vzpostavlanje cevi do posameznih končnih točk.
- konfiguracija USB: da se cevi lahko uporablja, jih je potrebno še nastaviti s strani odjemalca. Pri tem se odjemalcu podajo zahteve in omejitve pri uporabi cevi, kot npr. velikost podatkov, ki jih bo odjemalec prenesel na posamezen IRP, pogostost servisnih intervalov...
- konfiguracija funkcijske naprave: ko se izvedeta zgornji dve konfiguraciji, so cevi pripravljene za uporabo. Vseeno pa nekatere naprave potrebujejo specifično

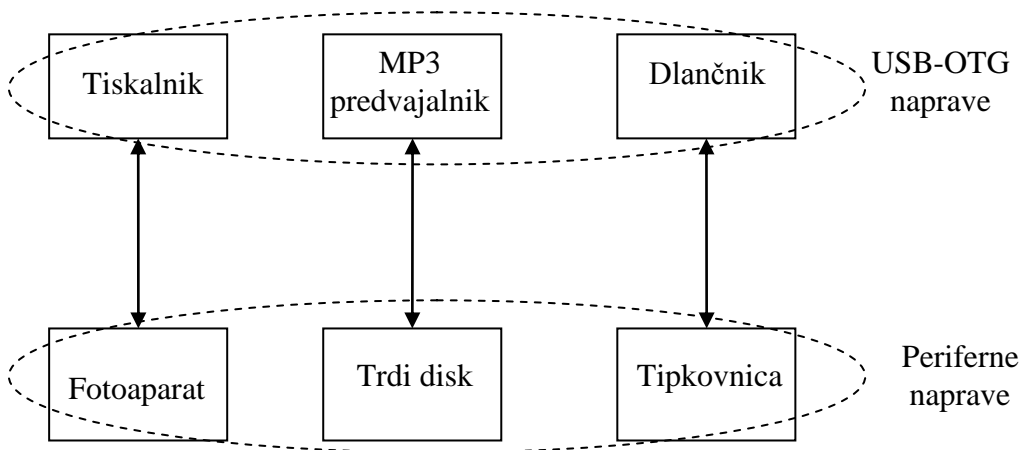
konfiguracijo, ki jo posreduje dobavitelj naprave USB in ni standardni del gonilnika USB.

Gonilnik USB mora pred vzpostavitvijo cevi do končne točke določiti ali bo sploh na voljo dovolj pasovne širine za komunikacijo. Vsaka končna točka ima v deskriptorju zapisano koliko pasovne širine zahteva. Ta podatek pa pove samo količino podatkov na paket ne pa tudi drugih parametrov, ki vplivajo na čas prenosa podatkov. Zato mora gonilnik USB izračunati potreben čas za prenos podatkov po vodilu. Če je ob upoštevanju vseh predpostavk na voljo dovolj časa na vodilu, gonilnik USB vzpostavi cev in doda transakcijo v razvrščevalnik. Te predpostavke so:

- vrsta prenosa: npr. izohroni prenos potrebuje zagotovljeno pasovno širino vodila, medtem ko ima prekinitveni prenos vedno zagotovljenih 10 % vodila,
- latenca gostitelja (angl. host recovery time): čas za pripravo nove transakcije, ki ga potrebuje gostiteljski krmilnik po končani zadnji transakciji,
- LS nastavitve na vozliščih: za prenose LS je potreben preambula paket in dodaten čas, da vozlišče pripravi vrata LS,
- čas za morebitno vrivanje bita pred kodiranjem NRZI,
- topologija vodila: čas, ki je potreben, da podatki pridejo do naprave je tudi odvisen od števila vozlišč oziroma kako globoko v topologiji je priključena naprava.

8 USB ON-THE-GO

Na koncu prejšnjega desetletja so se na trgu vse bolj pogosto pojavljale pametne naprave, ki so omogočale prenos podatkov. Čeprav je takrat USB dobro rešil prenos podatkov z navezo naprava-računalnik, se je pojavila potreba prenosa podatkov neposredno med perifernimi napravami. Proizvajalci so bili primorani uporabljati dokaj drago vodilo Fire-Wire ali pa so razvili lastne rešitve za prenos podatkov brez posredovanja računalnika. To pa je vodilo k nekompatibilnosti različnih naprav. Zato je USB-IF leta 2001 predstavil dodatek standardu USB 2.0, ki se imenuje USB OTG. Dodatek omogoča priključitev in komunikacijo dveh perifernih naprav brez posredovanja računalnika, tako imenovano povezavo točka-v-točko (slika 23). Pri tem ena od perifernih naprav prevzame vlogo gostitelja, ki vodi komunikacijo.



Slika 23: Primer povezave perifernih naprav brez posredovanja računalnika.

USB OTG dodatek standardu definira dve vrsti naprav OTG, naprave z dvojno vlogo (angl. dual role device, DRD) in periferne naprave. Naprave DRD imajo naslednje lastnosti:

- delujejo lahko kot gostitelj ali kot periferna naprava,
- imajo vtičnico mini-AB,
- kot gostitelj morajo biti sposobne generirati vsaj 8mA toka pri napetosti 4,25V,
- kot periferna naprava ne smejo porabiti več kot 8mA toka,
- lahko sprožijo ali začnejo protokol za začetek seje (angl. session request protokol, SRP),
- lahko sprožijo ali začnejo protokol za izmenjavo gostitelja (angl. host negotiation protocol, HPN).

Periferne naprave imajo naslednje lastnosti:

- nimajo gostiteljskih zmožnosti,
- imajo vtičnico mini-B,
- porabijo največ 8mA toka (oziroma več, če uporabljajo zunanje napajanje),
- lahko sprožijo protokol SRP.

8.1 PROTOKOL USB OTG

Za OTG dodatek standardu USB so razvili tri nove protokole, ki omogočajo vzpostavitev, komunikacije med dvema perifernima napravama:

- protokol za detekcijo vključitve (angl. attach detection protocol, ADP),
- protokol za začetek seje (SRP),
- protokol za izmenjavo gostitelja (HNP).

ADP ima funkcijo zaznavanja priključene naprave. Namreč naprava OTG periodično meri kapacitivnost linije na priključku. Tako lahko zazna ali je priključek prost ali pa je na njega priključena naprava tudi, če na liniji ni prisotne napetosti Vbus. Ko priključimo periferno napravo na linijo, naprava-A (OTG naprava) zazna spremembo kapacitivnosti linije in sproži napajanje Vbus, da poišče na novo priključeno napravo-B (periferna naprava).

Protokol SRP omogoča, da naprava-B zahteva od naprave-A, da začne dovajati električno energijo Vbus na vodilo in s tem začne sejo. To je zelo pomembno, saj se običajno naprava-A napaja preko baterije in lahko izključi napajalno linijo Vbus, kadar med napravama ni aktivnosti. Tako je pri napravah USB OTG s pomočjo protokola SRP na liniji Vbus napajanje samo, kadar poteka komunikacija med napravama, sicer pa je Vbus napajanje izključeno.

Protokol HNP omogoča izmenjavo funkcije gostitelja, kadar imamo neposredno povezani dve napravi DRD, ne da bi pri tem preklapljali kable.

9 STANDARD USB 3.0

Osem let po predstavitvi standarda USB 2.0, je 17. novembra 2008 USB-IF predstavil specifikacije standarda USB 3.0 oziroma marketinško poimenovanega USB SuperSpeed (USB SS). Doba osmih let je v računalniškem svetu precej dolga, toda ko pregledamo specifikacije postane hitro jasno, zakaj so potrebovali toliko časa (*tabela 4*). Odkrijemo namreč, da je princip delovanja za končnega uporabnika enak USB 2.0. Obdržali so eno izmed glavnih lastnosti USB, namreč združljivost s starejšim standardom in tako prehod ni problematičen. Kljub temu pa so pri USB 3.0 povišali hitrost prenosa podatkov za faktor 10 in zdaj znaša 5 Gb/s. Z naprednim upravljanjem napajanja so znižali porabo naprave USB 3.0, ko je v mirovanju. Po drugi strani pa so za 80 % povišali maksimalni napajalni tok glede na standard USB 2.0. Podroben pogled razkrije, da se je princip komunikacije drastično spremenil. Standard USB 3.0 je še vedno gostiteljsko voden prenos podatkov. Ta poteka sedaj asinhrono po principu full-duplex. To pomeni, da USB 3.0 uporablja dve ločeni podatkovni liniji, eno za prenos podatkov proti gostitelju in drugo linijo za prenos podatkov proti napravi. Obe podatkovni liniji pa lahko prenašata podatke hkrati. USB 3.0 sestavljajo štiri plasti:

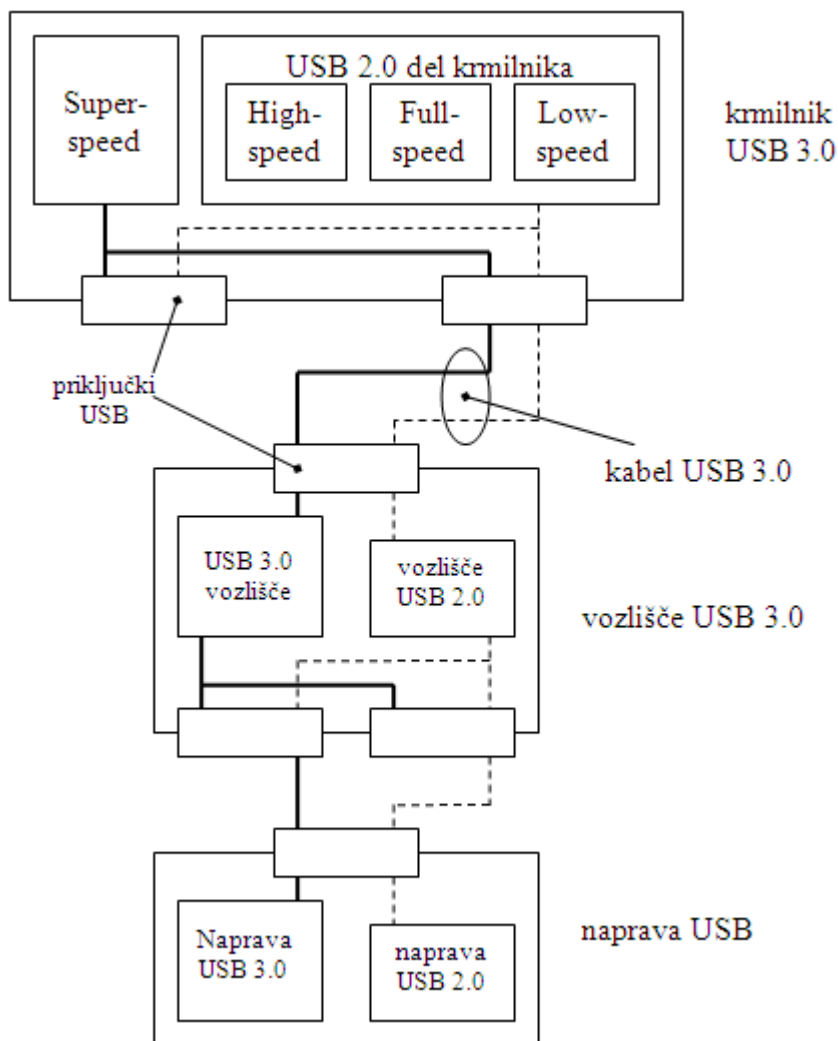
- fizična plast: predstavlja elektronska vezja in komunikacijo med njimi,
- povezovalna plast: je novost v standardu USB in predstavlja komunikacijo med sosednjimi vrati (angl. link partner), ki sta neposredno povezani s kablom,
- protokolna plast: vodi podatkovni tok med gostiteljskim krmilnikom in napravo USB,
- programska plast: vsebuje aplikacije in gonilnike za upravljanje naprave USB.

Karakteristika	USB 2.0	USB 3.0
Hitrost prenosa	LS = 1,5 Mb/s, FS = 12 Mb/s, HS = 480 Mb/s	SS = 5.0 Gb/s
Podatkovni vmesnik	Half-duplex	Full-duplex
Število podatkovnih linij	Dve za komunikacijo HS/FS/LS	Šest: 4 za komunikacijo SS in 2 za HS/FS/LS
Transakcijski protokol	Voden s strani gostitelja, prenos podatkov se izvede s programskim izpraševanjem, paketi se pošljejo vsem napravam	Voden s strani gostitelja, asinhroni prenos podatkov, paketi se usmerjajo
Napajanje naprav iz vodila	100 mA oziroma 500 mA, ko je naprava konfigurirana	150 mA oziroma 900 mA, ko je naprava konfigurirana.
Vrsta prenosa podatkov	Masivni, izohroni, prekinitveni, kontrolni	Izohroni, prekinitveni, kontrolni, masivni (slednji omogoča tokovni in eksplozijski prenos)

Tabela 4: Primerjava nekaterih lastnosti standardov USB 2.0 in USB 3.0.

9.1 KOMPATIBILNOST STANDARDA USB 3.0 IN USB 2.0

Standard USB 3.0 omogoča kompatibilnost s standardom USB 2.0. Pri tem uporablja arhitekturo z ločenimi linijami za prenose s hitrostjo USB 3.0 in prenose, ki potekajo s hitrostjo USB 2.0 (*slika 24*). Gostiteljski krmilnik vsebuje logično vezje za krmilnik USB 3.0 in paralelno dodatno logično vezje za krmilnik USB 2.0. Sedaj lahko na USB 3.0 priključimo tako napravo USB 2.0 kot napravo USB 3.0 in obe bosta delovali neodvisno drug od druge. Pri tem bo vsaka naprava delovala z najvišjo hitrostjo, ki jo omogoča ali pa s hitrostjo vozlišča, če deluje z nižjo hitrostjo kot naprava.



Slika 24: USB 3.0 uporablja ločen krmilnik za izvedbo transakcij USB 2.0.

Osnovna topologija vodila USB 3.0 je enaka kot pri vodilu USB 2.0. Naprave priključujemo v drevesno strukturo. Na najnižjem nivoju je en sam gostitelj s korenskim vozliščem na katerega priključujemo druge naprave. Vsaka naprava ima eno povezavo proti gostitelju in v primeru vozlišča več povezav za naprave.

9.2 FIZIČNA PLAST USB 3.0

V primerjavi z USB 2.0 so zaradi zanesljivejšega in hitrejšega prenosa podatkov zamenjali in dodali določene komponente. Tako sta novi pomembni komponenti:

- mešalec (angl. scrambler),
- kodirnik 8b/10b.

Mešalec vsebuje premikalni register s povratno vezavo (angl. linear feedback shift register, LFSR). Mešalec prestreže ves podatkovni tok, ki pride iz povezovalne plasti in spremeni zaporedje bitov tako, da se zmanjša verjetnost pojava neželenih bitnih zaporedji. To omogoča boljše delovanje vezja za restavriranje urinih impulzov pri sprejemu in bolj razpršen močnostni spekter signala in s tem posledično manjši presluh med podatkovnimi linijami v povezovalnem kablu. Mešalec ni namenjen kriptiranju prenašanih podatkov, temveč samo popravljanju tehničnih parametrov signala.

Tako kot USB 2.0 tudi USB 3.0 v kablu nima posebne linije za prenos ure. Pred kodiranjem NRZI se pri USB 3.0 uporablja dodatno kodiranje 8b/10b. Kodiranje 8b/10b deluje na principu preslikave 8 bitnih podatkov v 10 bitne podatke. Vsako 8 bitno število ima že vnaprej določeno 10 bitno kodo ali tako imenovane simbole. Posamezni simbol vsebuje največ šest logičnih 1 ali 0 in največ štiri zaporedne 1 ali 0. Tako zaporedje bitov predstavlja dobro uravnotežene logične nivoje in posledično dobro informacijo za sinhronizacijo ure sprejemnika in oddajnika. Kodiranje 8b/10b loči dve vrsti simbolov:

- D-simbole ali podatkovne simbole, ki prenašajo podatke uporabne za napravo USB,
- K-simbole ali kontrolne simbole. To so nizkonivojski ukazi, ki se uporabljajo pri nazoru in upravljanju komunikacije. USB 3.0 uporablja 12 kontrolnih simbolov za upravljanje povezave in okvirjanje paketov.

Kodiranje 8b/10b porabi precej pasovne širine. Če želimo poslati 8 bitov, se bo dejansko po kodiranju na linijo poslalo 10 bitov. USB 3.0 ima maksimalno teoretično kapaciteto 5 Gb/s. Zaradi učinka kodiranja 8b/10b se bo teoretično skozi vodilo preneslo maksimalno $5 \text{ Gb/s} * 8/10 = 4 \text{ Gb/s}$ uporabnih podatkov. Pri tem je potrebno upoštevati, da okoli 2 % poslanih podatkov predstavljajo glave in okvirji paketov. Tako lahko preko vodila pošljemo okoli 3,9 Gb/s uporabnih podatkov.

Fizična plast ima tudi nalogo oddajanja nizkofrekvenčne periodične signalizacije (angl. low frequency periodic signaling, LFPS). To je pravokotni signal, ki se oddaja med sosednimi vrati, za izvajanje posebnih nizkonivojskih ukazov (reset, komunikacija v stanju mirovanja...).

9.2.1 Upravljanje z električno porabo

V primerjavi z USB 2.0 je USB 3.0 naredil velik korak v smeri varčevanja z električno energijo. Paketi niso več razpršeno oddani vsem napravam (broadcasting) ampak USB 3.0 uporablja princip pošiljanje paketa samo tisti napravi, kateri je bil paket namenjen (unicasting). Tako je lahko aktivna samo tista naprava s katero bo dejansko potekala komunikacija.

Kot USB 2.0 tudi USB 3.0 omogoča varčevanje na nivoju naprave. Največja novost pa je več nivojsko varčevanje na fizični in povezovalni plasti, tako imenovana stanja *U*. Višja kot je

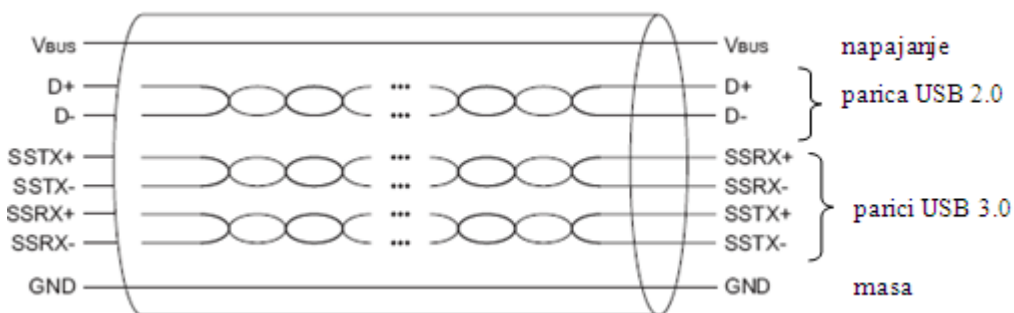
stopnja stanja U , bolj ostri so varčevalni ukrepi. Toda s tem se hkrati povečuje latenca, pri vzpostavitvi aktivnega stanja. USB 3.0 definira naslednja stanja U :

- stanje $U0$: predstavlja aktivno povezavo med sosednjimi vrati.
- stanje $U1$: sprejemnik in oddajnik preideta v stanje mirovanja. Prehod v to stanje se sproži, ko poteče določen čas brezdolja med sosednjimi vrati. Prehod v $U1$ stanje lahko zahtevajo tudi sama vrata, toda za dejanski prehod morajo sosednja vrata zahtevo potrditi lahko pa jo tudi zavržejo. Čas za prehod v aktivno stanje se meri μs .
- stanje $U2$: vezje za generacijo ure se postavi v mirovanje. Čas za prehod v aktivno stanje se meri v μs -ms.
- stanje $U3$: povezava med sosednjimi vrati je suspendirana (angl. link suspend). Vrata dobijo samo nujno napajanje za zaznavo reseta in oddaljenega obujanja. Čas za prehod v aktivno stanje se meri v ms.

Poleg varčevanja je pri USB 3.0 povečana tudi zgornja meja napajalnega toka. USB 3.0 lahko konfiguriranim napravam dostavi 80 % več električne energije v primerjavi s predhodnikom. Napravam ki še niso bile konfigurirane, pa lahko USB 3.0 dostavi za 50% večji napajalni tok v primerjavi z USB 2.0. Sedaj ob priklopu naprave dobijo 150 mA toka. Po konfiguraciji pa lahko preko deskriptorja zahtevajo še dodatne napajalne enote, vse do 900 mA toka oziroma do 1000 mA, če uporabljajo napajalni B kabel. Naprava v suspendiranem stanju pa lahko dobi do 2,5 mA toka. V primeru, ko je naprava USB 3.0 priključena na USB 2.0 gostiteljski krmilnik, se napajanje ravna po specifikaciji USB 2.0. Torej pred konfiguracijo dobi 100 mA toka in po konfiguraciji največ 500 mA. Čeprav se naprave USB 3.0 napajajo preko 5 V Vbus linije, standard zahteva delovanje tudi pri 4 V, zaradi možnih nihanj napajanja.

9.2.2 Povezovalni kabli

Pri razvoju novih kablov so morali razvijalci misliti na možnost prenašanja podatkov s 5 Gb/s. Linija mora biti čim bolj neobčutljiva na elektromagnetne motnje, hkrati pa mora biti dovolj poceni.

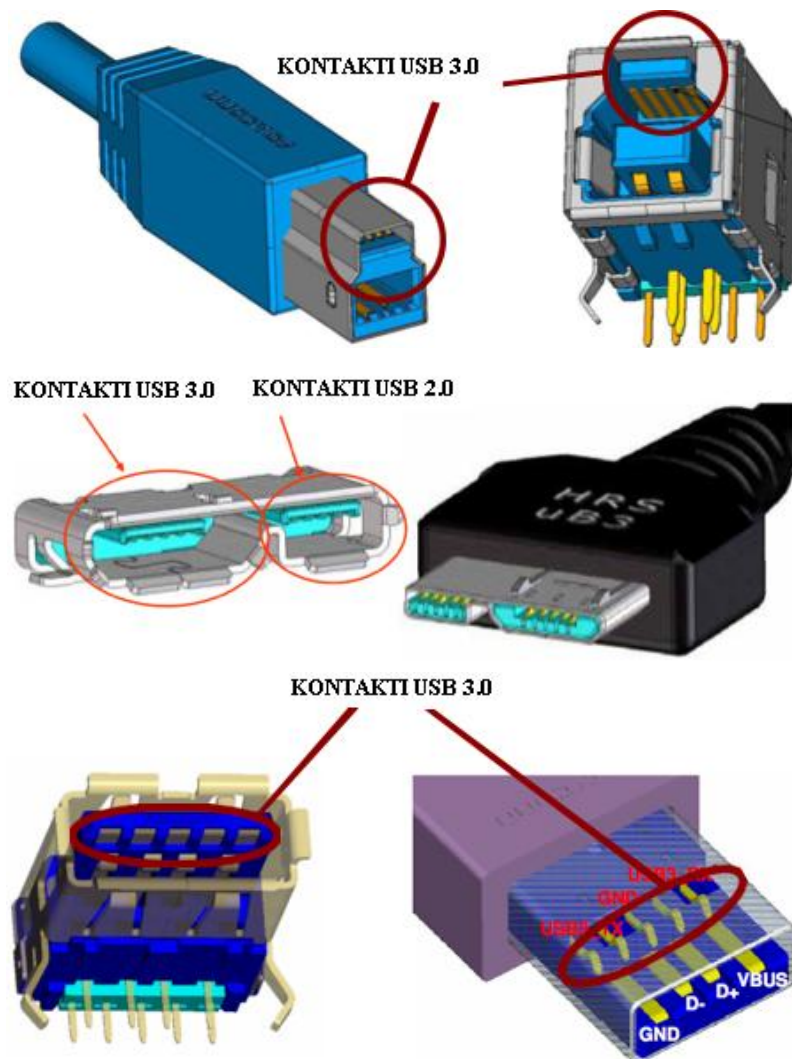


Slika 25: Zgradba kabla USB 3.

USB 3.0 uporablja za komunikacijo 8 žilni kabel, ki mora biti oklopljen (slika 25). Za kompatibilnost s starejšim standardom se ena parica uporablja za komunikacijo z napravo USB 2.0, to sta D- in D+. Dodatno pa vsebuje še dve parici, linija SSRX+, SSRX- se uporablja za sprejem podatkov in linija SSTX+ in SSTX- za pošiljanje podatkov. Nato ostaneta še dva vodnika za napajanje Vbus in masa GND.

Zaradi dodatnih linij v kablu USB 3.0 so novi vtiči in vtičnice nekoliko kompleksnejši (slika 26). Razvijalci so nove linije v priključkih razporedili tako, da so kompatibilni med seboj, zato bodo vse naprave USB 3.0 delovale preko kabla USB 2.0, seveda s temu primerno nižjo hitrostjo. Medtem, ko kabli USB 3.0 niso predvideni za povezavo naprave USB 2.0, zato se določene naprave USB 2.0 ne da priključiti preko kabla USB 3.0. Razvijalci standarda so zato predlagali proizvajalcem, da vtiče kablov USB 3.0 obarvajo turkizno in s tem uporabnika že vizualno opozorijo.

Na sliki 26 se vidi, da imajo vsi priključki USB 3.0 dva nabora kontaktov. Štiri za naprave USB 2.0 in 5 za naprave USB 3.0. Pri standardnih A priključkih so kontakti USB 3.0 postavljeni globlje v vtiču in kontakti USB 2.0 na začetku vtiča.



Slika 26: Priključki USB 3.0 z razporeditvijo kontaktov. Od zgoraj navzdol si sledijo: standardni B priključek, standardni mikro priključek, standardni A priključek.

Standardni B USB 3.0 vtič je ločen z dvema prekatoma, v spodnjem so štirje kontakti za naprave USB 2.0 v zgornjem prekatu pa je pet kontaktov za naprave USB 3.0. Standardni B USB 3.0 vtič je nekoliko višji od standardnega B USB 2.0 vtiča in ga zato ne moremo vključiti v standardno B USB 2.0 vtičnico. USB 3.0 je definiral še USB napajalni B kabel (angl. powered B), ki je vedno nepomično vgrajen v napravo in ima dve dodatni liniji za napajanje. Ta omogoča, da dobi naprava do 1000 mA toka.

Mikro B priključek USB 3.0 ima ravno tako kot standardni B priključek dva prekata. Levi služi za komunikacijo z napravo USB 2.0 in desni za komunikacijo z napravo USB 3.0, zato je širši od mikro B USB 2.0 priključka in ni kompatibilen s starejšimi napravami.

9.3 POVEZOVALNA PLAST

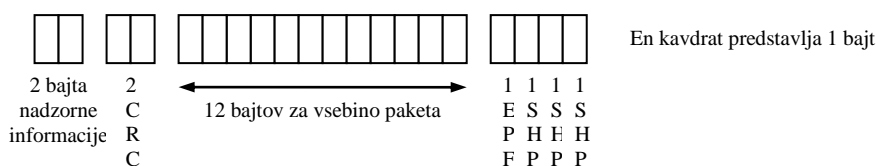
Povezovalna plast ima nalogo vzdrževati povezavo med sosednjimi vrati in s tem zagotoviti zanesljivo komunikacijo med njima. Med sosednjimi vrati se prenašajo paketi in povezovalni ukazi (angl. link commands). Medtem ko so paketi namenjeni napravi USB ali gostitelju, se povezovalni ukazi uporabljajo za komunikacijo med sosednjimi vrati. Za okvirjanje se uporablja K-simbole, ki so osnovni gradniki okvira. K-simbol predstavlja kontrolni del simbolov pri kodiranju 8b/10b.

Zanesljiv prenos podatkov med sosednjimi vrati omogoča okvir paketa, ki tolerira napako enega simbola. Vsebina paketa pa je dodatno zaščiten s CRC biti, ki poskrbijo za nadaljnjo integriteto podatkov. Povezovalna plast skrbi za naslednje funkcije:

- okvirjanje paketa,
- nadzor toka podatkov,
- vzpostavljanje in upravljanje povezave med sosednjimi vrati,
- upravljanje z napajanjem.

9.3.1 Okvirjanje paketa

Protokolna plast med komunikacijo posreduje pakete proti povezovalni plasti. Na tej plasti se vse 14 bajtne glave paketov opremi z okvirjem dolgim štiri simbole, ki se prenašajo med sosednjimi vrati. Za tem se prenesejo še podatki (angl. packet payload), če gre za podatkovni paket. Tudi podatki podatkovnega okvirja se okvirijo, in sicer z osmimi simboli. Prvi štirje bajti vsakega okvirja napovejo začetek paketa (*slika 27*). Trije simboli SHP (angl. start header packet) in en EPF (angl. end packet framing) služijo kot sinhronizacijsko polje saj USB 3.0 ne pozna polja SYN, kot ga je imel USB 2.0.



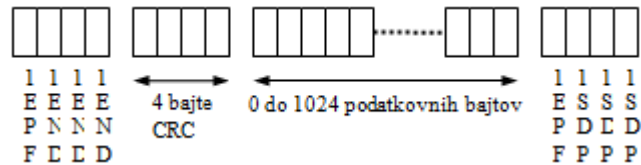
Slika 27: Struktura okvirja 3x SHP in 1x EPF. V okvirju se nahaja 12 B glava paketa in 2 B nadzorne informacije.

Po okvirju sledi glava paketa, ki jo protokolna plast pošlje povezovalni plasti. Glava paketa je dolga 14 B, prvih 12 B vsebuje informacijo o glavi, zadnja 2 B pa so CRC biti. Na koncu sledita še 2 B nadzorne informacije (angl. Link control world), ki ima funkcijo nadzora povezave med sosednjimi vrati, kot tudi za nadzor povezave na protokolni plasti. Z nadzorno informacijo lahko ugotovimo:

- številko poslani glave paketa in s tem ugotovimo, kateri paket je potrebno ponovno poslati v primeru napake,

- ali so vrata, kamor so bili namenjeni podatki, v varčevalnem stanju U1 ali U2,
- Informacijo o zakasneni glavi paketa, kadar je potrebno paket poslati ponovno zaradi polnega medpomnilnika ali okvare paketa.

Podatkovne transakcije vsebujejo poleg glave podatkovnega paketa tudi podatke v podatkovnem paketu. Podatki v podatkovnem paketu so struktura v kateri se dejansko pošljejo koristni podatki za napravo USB ali gostitelja (*slika 28*). Največja količina poslanih podatkov na paket je enaka kot pri USB 2.0 in znaša 1024 B, ne glede na velikost pa jo vedno spremlja 32 CRC bitov. Podatki v podatkovnem paketu se pošljejo takoj za glavo podatkovnega paketa in med njima ni »prazne« (idle) periode.

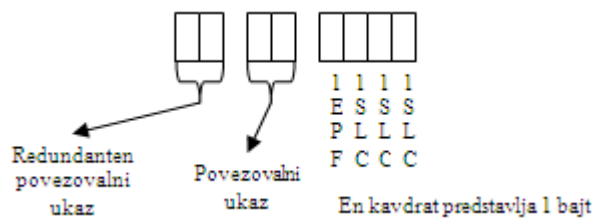


Slika 28: Podatke podatkovnega paketa spremlja 8 simbolov okvirja. Okvir se začne s tremi simboli SDP (angl. start data packet) in nato še en simbol EPF. Sledijo podatki in nato zaključitev okvirja s tremi END in enim simbolom EPF.

Na prenosni poti med sosednjimi vrati se lahko okvir pokvari. V primeru napake v okvirju ima USB 3.0 možnost zaznave in odprave napake velikosti enega simbola. Tak pokvarjen simbol se bo v sprejemniku nadomestil s SKP (angl. skip) K-simbolom in ne bo motil nadaljnje obdelave. V primeru okvare več simbolov se tak okvir z vsebino zavrže in zahteva ponovno pošiljanje.

9.3.2 Povezovalni ukazi

Povezovalni ukazi se uporabljajo za kontrolno komunikacijo med sosednjimi vrati in imajo funkcijo nadzora pretoka podatkov, upravljanje z napajanjem in zagotovitev skladnosti podatkov. Ti ukazi so vedno fiksne dolžine 8 B. Integriteto zagotovi redundanten ukaz znotraj poslanega okvirja in CRC biti (*slika 29*).



Slika 29: Okvir sestavlja 3x SLC (angl. start link command) in simbol EPF. Nato pa sledi 11 bitni ukaz zaščiten s petimi CRC biti, takoj za tem pa še en identičen ukaz.

Če je potrebno, se lahko pošlje več zaporednih povezovalnih ukazov. Vedno pa morajo počakati na prosto vodilo in jih ne moremo vrvati v glave drugih paketov. Trenutno je definiranih 11 povezovalnih ukazov, med najbolj pomembnimi so:

- **LGOOD_n**: vrne oddajniku informacijo o uspešni dostavi glave paketa, ki je brez CRC napak. Pri tem »n« predstavlja številko glave paketa, ki je bila uspešno prenesena. Številka glave se inkrementno povečuje od 0 do 7 in nato ponovno od 0 naprej.
- **LCRD_x**: Služi za kontrolo pretoka, kot tudi za signalizacijo prostega medpomnilnika pri sprejemniku. Vsaka vrata morajo imeti vgrajene štiri medpomnilniška mesta. »x« označuje mesta medpomnilnika od A do D. Vsako medpomnilniško mesto je rezervirano za eno glavo paketa. Ta ukaz z ustrezno medpomnilniško črko se pošlje oddajniku po vsakem uspešnem prejetju glave paketa, za sprostitev medpomnilnika. Oznaki medpomnilnika v oddajniku in sprejemniku morata biti usklajeni in se za vsako glavo paketa povečujeta za eno črko.
- **LBAD**: vrne oddajniku informacijo o pokvarjeni glavi paketa. Oddajnik bo ponovno poslal glavo paketa od zadnjega uspešno prejetega ukaza **LGOOD_n**.
- **LRTY**: pošlje oddajnik po tem, ko je dobil ukaz **LBAD**. Z ukazom **LRTY** sporoči sprejemniku, da bo poskusil ponovno poslati prej pokvarjeni paket.
- **LGO_{Un}**: predlog vrat za vzpostavitev varčevalnega stanja *U*. Pri tem je »n« globina stanja *U* in je lahko med 1 in 3. Sosednja vrata lahko odgovorijo z ukazom **LAU**, kar pomeni sprejem stanja *U* ali pa **LXU** za odklon.

9.3.3 Diagram prehajanja stanj povezave

Diagram prehajanja stanj na povezovalnem nivoju opiše vsa možna stanja, v katerih se lahko nahaja povezava med sosednjimi vrati (*slika 30*). Dejansko ima vsako spodaj omenjeno stanje še več podstanj, ki podrobneje opisujejo stanje v katerem se nahaja povezava. Pri določenih stanjih se uporablja tudi posebna vrsta komunikacije to je nizkofrekvenčna periodična signalizacija (angl. low Frequency Periodic Signaling, LFPS). To je stranska komunikacija in se uporablja:

- ko sta sosednji vrati v stanju varčevanja,
- med usklajevanjem povezave med sosednjimi vrati,
- kot ukaz za topli reset.

Signal LFPS je periodičen pravokoten signal, perioda in čas trajanja signala pa je odvisna od ukaza in stanja v katerem sta sosednji vrati.

časovnem okvirju ne zaznajo svoja sosednja SS vrata, preide povezava v stanje *SS.Disabled*. V nasprotnem primeru povezava preide v stanje *Polling*.

Polling: V tem stanju sosednji vrati s poskušanjem ugotovita najprimernejše parametre za prenos (angleško link training). To se izvaja s pomočjo signalizacije LFPS in z branjem skupine K-simbolov.

U0: To je normalno operativno stanje v katerem se lahko pošilja in sprejema pakete.

U1: Je varčevalno stanje v katerega morata privoliti obe vrati. V tem stanju pošiljanje paketov ni mogoče. Periodična komunikacija pa poteka preko LFPS. Če želi povezava preiti v aktivno U0 stanje, je najprej preusmerjena v stanje *Recovery*. Povezava lahko preide v varčevalno stanje U2, če poteče časovna kontrola za neaktivnost v stanju U1. V primeru, da poteče časovna kontrola za usklajevanje z LFPS signalom za prehod v stanje *Recovery*, povezava preide v stanje *SS.Inactive*.

U2 in U3: Sta varčevalni stanji, ki omogočata naprednejše varčevalne ukrepe, toda z večjo latenco, ko je potreben prehod v stanje *Recovery* in nato v stanje U0.

Recovery: Stanje je namenjeno ponovni pripravi povezave med vrati po prehodu iz varčevalnega stanja. Priprava poteka hitreje kot v stanju *Polling*, saj se določeni parametri ohranijo še od prvotne priprave povezave. V to stanje lahko preide povezava tudi v primeru, ko vrata v stanju U0 zaznajo, da sosednja vrata ne deluje pravilno in je potrebna faza ponovne priprave povezave. V tem primeru se vrednosti števecv znotraj vrat ohranijo, tako se komunikacija po vrnitvi v stanje U0 nadaljuje kjer se je prej končala.

Compliance Mode in Loopback: Stanji sta namenjeni za testiranje povezave, kot so testiranje pogostosti napake v komunikaciji, preverjanje napetostnih in časovnih karakteristik.

Hot Reset: Vroči reset lahko izvedejo samo izhodna vrata. Z njim lahko pobrišemo trenutne vrednosti števca napak pri povezavi, sekvenčna števca obeh vrat in nastavljene vrednosti časovnih kontrol za prehod v varčevalna stanja, medtem ko konfiguracija vrat ostane nespremenjena. Vroči reset se izvede s pomočjo K-simbolov. Razlika med vročim in toplim resetom je, da slednjega lahko izvedemo iz katerega koli stanja preko signalizacije LFPS, povezavo med sosednjimi vrati pa je nato potrebno na novo vzpostaviti.

9.3.4 Primer komunikacije na povezovalnem nivoju

Za primer pogledimo prenos glave paketa, ki ga prikazujeta sliki 31 in 32. Oba primera predpostavljata, da je povezava prešla iz stanja *Polling*, kar pomeni, da so vsi števci postavljeni na privzeto začetno vrednost. Če povezava preide iz stanja *Recovery*, se ohranijo števci in komunikacija se nadaljuje od tam, kjer se je končala pred prehodom v stanje *Recovery*. Komunikacija lahko poteka hkrati v obe smeri in ne samo v eno smer, kot poenostavljeno prikazuje primer. Preden preidemo na primer, je potrebno še razumeti najpomembnejše števce znotraj vrat. Vsa vrata imajo pet pomembnih polj ali števecv preko katerih nadzirajo pretok podatkov.

Tx header sequence number (THSN): Števec označi vsako poslano glavo s številko od 0 do 7 (in nato ponovno od 0 naprej). Tako lahko sprejemnik s pomočjo RHSN prepozna, če je dobil glavo paketa v pravilnem vrstnem redu.

ACK Tx header sequence number (ATHSN): Sprejemnik za vsak uspešno prejet paket pošlje potrditev *LGOOD_n*. »n« mora ustrezati števcu THSN, to pa preverja polje ATHSN. Ko potrdi pravilnost potrditve, lahko oddajnik sprosti medpomnilnik, ki vsebuje to glavo. Do zdaj jo je moral oddajnik zadržati, če bi jo zaradi neuspešne dostave moral poslati ponovno.

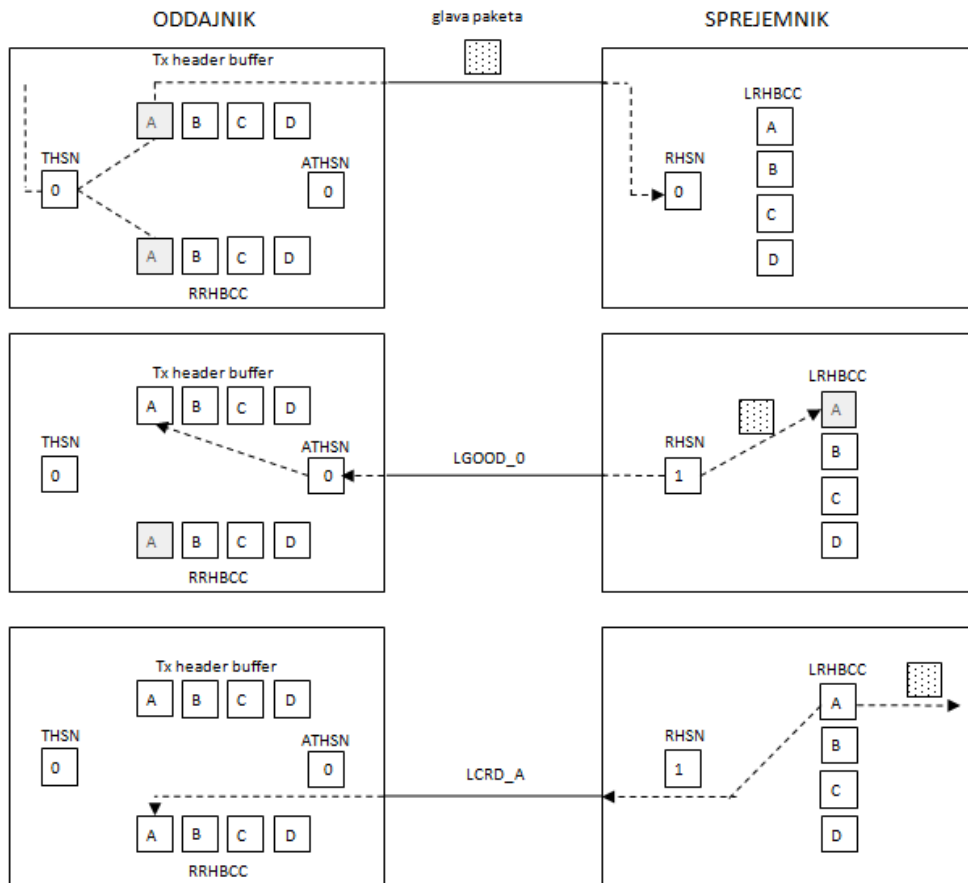
Rx header sequence number (RHSN): Števec pove katera glava paketa je naslednja, ki jo sprejemnik pričakuje in je s števcem THSN vedno sinhroniziran, tako lahko sprejemnik preveri ali dobiva pakete v pravilnem zaporedju.

Local Rx Header Buffer Credit count (LRHBCC): Vsa vrata imajo medpomnilnik v katerega lahko sprejmejo do štiri glave paketov, preden jih pošljejo v nadaljnjo obdelavo. Ta števec razkrije za koliko glav paketov je v sprejemniku še prostega medpomnilnika. Medpomnilniška mesta so označena z A,B,C in D.

Remote Rx Header Buffer Credit count (RRHBCC): Oddajnik vodi tudi evidenco prostih medpomnilniških mest v sprejemniku. Sprejemnik sporoča informacijo o svojem medpomnilniku preko ukaza *LCRD_x*. »x« je lahko A,B,C ali D odvisno kateri del medpomnilnika se je sprostil. Sprejemnik ukaz *LCRD_x* pošlje po tem, ko uspešno konča procesiranje prejetega paketa, takrat se tudi v sprejemniku sprosti eno od štirih mest medpomnilnika.

Prvi primer (*slika 31*) prikazuje uspešen prenos ene glave paketa:

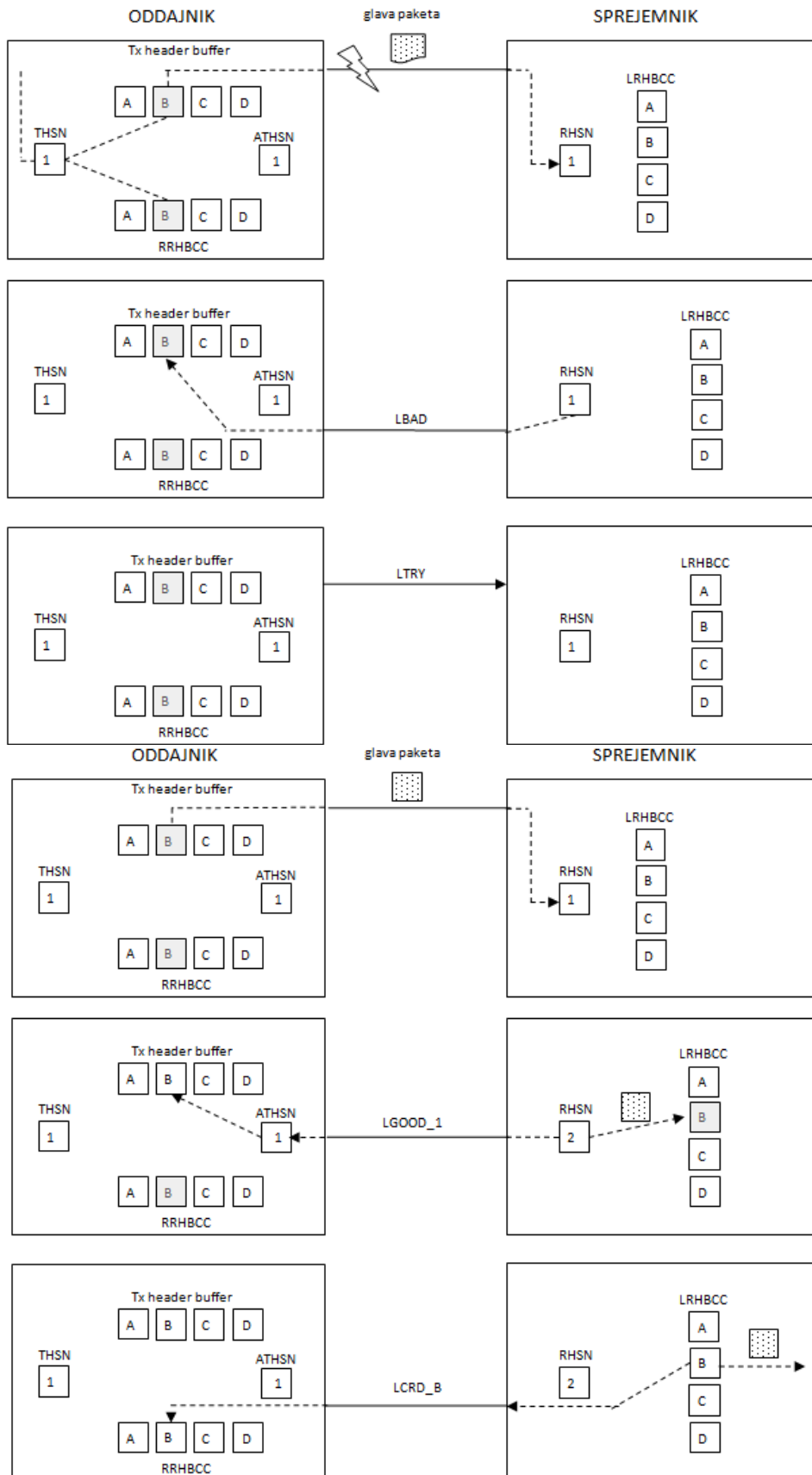
1. Povezava je prešla iz stanja *Polling*, zato so števci obeh vrat postavljeni na privzete vrednosti.
2. Glava paketa zasede prosto mesto v medpomnilniku oddajnika. THSN opremi paket z zaporedno številko glave paketa in ga pošlje sosednjim vratom. Sprejemnik preveri zaporedno številko in s tem zagotovi pravilni vrstni red. Realno oddajnik pogosto pošlje več glav paketov preden dobi potrditev za prvega, toda največ štiri, odvisno koliko je prostega medpomnilnika.
3. Če paket prestane preverjanje CRC in zaporedna številka ustreza prejemnikovi RHSN, ga sprejemnik sprejme v medpomnilnik A. Oddajniku signalizira *LGOOD_0*, le ta pa lahko nato sprosti medpomnilnik, kjer je bila shranjena glava paketa.
4. Ko sprejemnik obdela glavo paketa, signalizira oddajniku *LCRD_A*. S tem oddajnik dobi informacijo, da je sprejemnik pri sebi sprostil medpomnilniško A mesto, kjer se je zadržala poslana glava paketa. V primeru, da ima sprejemnik polna vsa štiri medpomnilniška mesta, mora obvezno počakati na odziv *LCRD_x*, šele nato lahko oddajnik pošlje novo glavo paketa.



Slika 31: Poenostavljen primer uspešne komunikacije na povezovalna plast.

Drugi del primera je nadaljevanje od prvega, le da se tokrat paket pokvari (slika 32):

1. Tokrat THSN inkrementira zaporedno številko glave paketa.
2. Sprejemnik preveri CRC bite. Ugotovi, da je paket pokvarjen, zato ga zavrže in oddajniku pošlje *LBAD*, medpomnilnik sprejemnika pa ostane nezaseden. Če sprejemnik 3x zapored ni zmožen sprejeti glave paketa preide v stanje *Recovery*.
3. Oddajnik sporoči sprejemniku z *LTRY*, da bo poskusil s ponovnim pošiljanjem. Če je bilo poslanih več glav paketov, se ponovno pošlje vse glave paketov, ki niso bile potrjene z *LGOOD*. Oddajnik ima glavo paketa že pripravljeno v medpomnilniku še preden je dobil signal *LBAD*, saj se ta del medpomnilnika izbriše po tem, ko dobi ustrezen signal *LGOOD_x*.
4. Oddajnik ponovno pošlje glavo paketa z enako zaporedno številko, kot jo je imel prej pokvarjen paket.
5. Sprejemnik je zdaj glavo paketa uspešno prejel in oddajniku to signalizira z *LGOOD_1*. V primeru, da se vrstni red potrditve *LGOOD* zameša, preide povezava v stane *Recovery*, enako velja za pomešan vrstni red potrditev *LCRD*.
6. Po uspešni obdelavi glave paketa sledi še informacija o sprostitvi prejemnikovega medpomnilnika, tokrat *LCRD_B*. Glave paketov vedno zasedajo medpomnilnik inkrementno od A do D. Enako velja za potrditve o sproščenem medpomnilniku, tako se lahko zavaruje pred morebitnim izgubljenem signalom *LCRD_x*.



Slika 32: Poenostavljena komunikacija na povezovalni plasti s pokvarjenim paketom.

9.4 PROTOKOLNA PLAST

Protokolna plast za razliko od povezovalne plasti, ki upravlja komunikacijo med sosednjimi vrati, upravlja komunikacijo med gostiteljem in napravo. Protokolna plast predpostavlja, da povezovalna plast zanesljivo dostavlja pakete na svojem nivoju. Protokolna plast pa nato dodatno poskrbi za zanesljiv prenos paketov med gostiteljem in napravo. USB 3.0 uporablja asinhrono unicast komunikacijo, kar ga na protokolni plasti močno razlikuje v primerjavi z USB 2.0. Transakcija USB 2.0 je običajno sestavljen iz treh paketov: žeton, podatki in usklajevanje. USB 3.0 pa je ta princip nekoliko optimiziral. Za izhodno transakcijo je žeton vgrajen v glavo podatkovnega paketa, medtem ko je vhodni žeton zamenjan z usklajevanjem.

USB 3.0 uporablja štiri osnovne vrste paketov, vsak pa lahko ima še dodatne podvrste. Vsi paketi imajo v glavi dve skupni polji. Prvi je tip polje (angl. type field), ki pove za kateri paket gre in pa 16 CRC bitov. Na koncu glave paketa se k vsakemu doda še 2 B polje za nadzor povezave (angl. link control word). Ostala polja v glavi paketov so specifična za določeno podvrsto paketov. Štirje osnovni paketi so:

- paket za upravljanje povezave,
- transakcijski paket,
- podatkovni paket,
- izohrona časovna znamka.

9.4.1 Paket za upravljanje povezave

Ti paketi (angl. Link management packet, LMP) so namenjeni upravljanju povezave med vrati. V paketu ne nosijo naslovne informacije, zato se jih ne usmerja po topologiji. Ti paketi se običajno generirajo kot posledica ukazov vozlišča svojim sosednjim vratom. Trenutno je definiranih 6 podvrst paketov LMP.

Set link function LMP: je paket, ki nastavlja možnost brezpogojnega prehoda v varčevalno stanje U1 ali U2 v primeru povezovalnega ukaza *LGO_U1* ali *LGO_U2*. Ta ukaz se običajno uporablja za testne namene, saj lahko sicer precej upočasnijo komunikacijo med vrati.

U2 inactivity Timeout LMP: s paketom lahko nastavimo čas mirovanja povezave med vrati, preden se izvede prehod iz varčevalnega stanja *U1* v stanje *U2*.

Vendor device test LMP: paket je specifičen za posameznega dobavitelja in namenjen za potrebe testiranja.

Port capabilities LMP: Paket opisuje karakteristike vrat. Ta paket morata obvezno poslati obe sosednji vrati ko se vzpostavi povezava. Med drugim paket opisuje ali so vrata sposobna delovati s komunikacijo SS, velikost medpomnilnika za sprejem in oddajo glave paketov, in identifikacijo ali gre za vhodna ali izhodna vrata.

Port configuration LMP in Port configuration response LMP: Paketa pošiljata vhodna in izhodna vrata za potrditev komunikacije v načinu SS. V primeru, da vrata ne dobijo odziva v predpisanem časovnem okvirju, preide povezava v stanje *SS.Disable*. Kasneje pa se poizkuša vzpostaviti povezava s hitrostjo USB 2.0.

9.4.2 Transakcijski paket

Ti paketi (angl. transaction packet, TP) se uporabljajo za upravljanje povezave med gostiteljem in končno točko v napravi. Torej za potrjevanje paketov, začetek vhodnega prenosa in javljanje zastojev. Vsak transakcijski paket, ki ga pošlje gostitelj, je namenjen samo določeni napravi. Naslov naprave, ki bo dobila paket je zabeležen v naslovnem polju. Če pa paket pošlje naprava, bo v naslovnem polju zabeležen naslov naprave, ki je poslala paket.

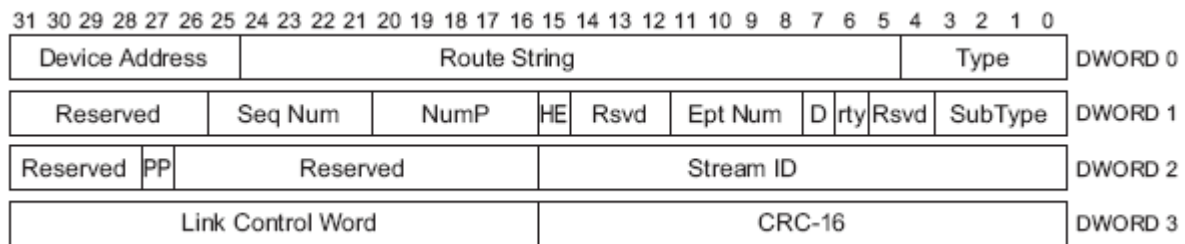
ACK TP je pri USB 3.0 nadomestil klasični žeton, ki se je uporabljal pri USB 2.0, zdaj je žeton vgrajen v paketu ACK. Interpretacija tega paketa je odvisna ali paket dobi vhodna ali izhodna končna točka. V vhodno končno točko paket ACK pošlje gostitelj s katerim poda zahtevo po prenosu podatkov od naprave proti gostitelju, ali pa kot potrditev, ko gostitelj uspešno sprejme podatke. Izhodna končna točka uporablja ta paket za potrditev uspešno prejetih podatkovnih paketov s strani gostitelja proti napravi.

ACK TP se dejansko uporablja tudi za potrditev neuspešno poslanih paketov, saj USB 3.0 ne pozna paketa NACK. V tem primeru se oddajniku pošlje ACK TP, ki ima v polju *Sequence Number* nastavljeno številko niza paketov, ki naj se pošljejo ponovno, torej niz paketov, ki so prispeli pokvarjeni. USB 3.0 pozna 8 podvrst transakcijskih paketov:

ACK TP je eden bolj pomembnih paketov, saj poleg potrditev in zahtevkov po prenosu nosi tudi druge pomembne informacije za učinkovito delovanje. Njegova polja prikazuje *slika 33* in jih bom opisal nekoliko podrobneje, saj so pomembna pri razumevanju principa transakcij USB 3.0.

- *Usmerjevalna informacija* (angl. *Route string*): Polje je namenjeno vozlišču za usmerjanje paketov po topologiji vodila USB.
- *Naslov naprave* (angl. *Device address*): Polje vsebuje naslov naprave, ki ga dodelil gostitelj med enumeracijo in je lahko vrednost med 1 in 127.
- *Podvrsta* (angl. *Subtype*): Polje opisuje za kakšno podvrsto TP gre. Torej v našem primeru gre za ACK TP.
- *Retry Data Packet* (*rtty*): Enobitno polje poda zahtevo za ponovno pošiljanje enega ali več paketov, če je bil prispeli paket pokvarjen ali pa sploh ni prispel. Polje se uporablja v kombinaciji s poljem *Sequence Number*.
- *Smer* (*D*): Polje definira smer prenosa ACK TP, torej od naprave proti gostitelju ali obratno.
- *Številka končne točke* (*Ept Num*): Polje pove kateri končni točki je namenjen paket. V primeru, da paket potuje od naprave proti gostitelju, je v polju zabeležena končna točka, ki je poslala paket.
- *Napaka gostitelja* (*HE*): Enobitno polje lahko sproži samo gostitelj, ko ne more sprejeti paketa zaradi nepredvidenega dogodka znotraj gostitelja. Hkrati mora biti postavljeno tudi polje *Retry Data Packet*.
- *Število paketov* (*NumP*): Polje sporoči velikost medpomnilnika za podatkovne pakete, ki jih lahko sprejemnik naprave trenutno sprejme. Vrednost ne sme biti večja od vrednosti *max_Burst_size* (največje število zaporednih transakcij brez vmesnega potrjevanja), ki je zabeležena v deskriptorju končne točke.
- *Sequence number* (*Seq Num*). To polje pove številko paketa, ki bo sledil v naslednji transakciji.
- *Stream ID*: Polje se uporablja samo pri masivnih transakcijah, ki podpirajo tokove. Polje ima lahko vrednost od 1 do 65533, tolikšno je lahko največje možno število tokov. Če končna točka ne podpira tokove, je vrednost polja 0.

- *Packet Pending (PP)*: Enobitno polje lahko sproži samo gostitelj in s tem sporoči napravi, da ima pripravljen nadaljnji paket za to napravo. V primeru, da ima polje vrednost 0, naprava ve, da k njej ni namenjen noben paket več in lahko takoj preide v varčevalno stanje *U1* ali *U2*.



Slika 33: Polja v ACK transakcijskem paketu.

NRDY TP: Paket lahko pošiljajo samo naprave, toda ne preko izohronih končnih točk. Izhodna končna točka signalizira, da nima več na voljo prostega pomnilnika za podatkovne pakete, ki jih pošilja gostitelj. Vhodne končne točke pošljejo NRDY v primeru, da ne morejo poslati podatkovnih paketov in je lahko odziv na ACK TP, ki ga pošlje gostitelj.

ERDY TP: S tem paketom naprava obvesti gostitelja, da je pripravljena za sprejem ali pošiljanje paketov. Paket pošljejo naprave, ki so predhodno signalizirale NRDY. Paket lahko pošiljajo samo naprave, toda ne preko izohronih končnih točk.

STATUS TP: Paket lahko pošlje samo gostitelj in sporoča, da naprava vstopa v statusno fazo pri kontrolni transakciji.

STALL TP: Paket lahko pošlje samo naprava, ki sporoča, da je prišlo do zastoja v končni točki ali do napake, če se izvaja kontrolni prenos.

PING TP: Pri USB 2.0 je imel paket PING funkcijo izpraševanja zasičene naprave. Pri USB 3.0 pa ima drugačno funkcijo. Paket se uporablja pred izvedbo izohronega prenosa za postavitve vseh povezav med gostiteljem in izohrono končno točko v aktivno stanje *U0*.

PING_RESPONSE TP: Ta paket pošlje naprava kot odziv na vsak PING TP in s tem signalizira gostitelju, da so vse povezave prešle iz varčevalnega stanja v stanje *U0*.

DEV_NOTIFICATION TP: Poseben paket za obveščanje gostitelja o asinhronih spremembah v napravi. Na primer obveščanje katera naprava je izvedla oddaljeno obujanje.

9.4.3 Podatkovni paket

Vsi podatkovni paketi so sestavljeni iz glave podatkovnega paketa in podatkov podatkovnega paketa in nimajo nobene podvrste paketa. V normalni situaciji naprava vrne podatkovni paket, kot odziv na ACK TP, ki ga pošlje gostitelj. Podatki v podatkovnem paketu so lahko dolžine 0 bitov, toda podatkovni paket bo še vedno vseboval 32 CRC bitov. V glavi podatkovnega paketa imamo poleg informacije, komu je paket namenjen, še nekatera bolj pomembna polja:

- *end of burst*: enobitno polje se uporablja pri neizohronih končnih točkah in označuje zadnji paket eksplozijskega prenosa,

- *sequence number*: polje se uporablja za beleženje niza podatkovnih paketov. Števec je dolg pet bitov in ko prešteje do 31 začne šteti od začetka,
- *setup*: polje sprejemniku pove, da gre za nastavitveni paket pri kontrolnem prenosu,
- *data length*: 16-bitno polje v katerem je zabeležena velikost podatkovnega bremena.

9.4.4 Izohrona časovna znamka

USB 3.0 ne uporablja več paketa SOF, ki je v USB 2.0 naznanjal začetek posameznega časovnega okvirja transakcije. Napravam USB 3.0 dodeljuje časovno informacijo preko izohrone časovne znamke (angl. Isochronous timestamp packet, ITP). Ta paket ne vsebuje naslovnega kot tudi ne usmerjevalnega polja. Samo gostitelj lahko pošilja ITP. Gostitelj periodično pošilja ITP korenskemu vozlišču, ta pa nato razpošlje paket vsem napravam, ki imajo povezavo v aktivnem stanju *U0*. Če imajo naprave povezavo v varčevalnem stanju, paketa ITP ne prejmejo. Če je ITP bistvenega pomena za delovanje naprave, mora le ta sama poskrbeti, da povezava ne preide v varčevalno stanje. Naprave se na ITP ne odzivajo in ga lahko, če ne potrebujejo časovne informacije, ignorirajo. Pakete ITP pošilja gostitelj na vsake 125 s. Če ima paket ITP v glavi postavljen *delayed* bit, lahko tak paket ignorira tudi naprava, ki potrebuje časovno informacijo, saj je tak paket zakasnen in ne nudi dobre časovne informacije. Prekinitvene in izohrone končne točke imajo v deskriptorjih zabeležen servisni interval, ki pove na koliko paketov ITP morajo biti postrežene.

9.5 TRANSAKCIJA PAKETOV PRI STANDARDU USB 3.0

Kot pri USB 2.0 tudi pri USB 3.0 vse transakcije vodi gostitelj, ki proti napravi pošlje podatke ali podatke od naprave zahteva. Transakcije se končajo, ko naprava pošlje ali podatke ali potrditev. Gostitelj USB 3.0 lahko začne več izhodnih transakcij, medtem ko čaka, da se prva transakcija konča. Pri vhodnih transakcijah pa gostitelj vedno čaka, da se trenutna transakcija konča (uspešno ali neuspešno) in med tem ne bo začel nove vhodne transakcije.

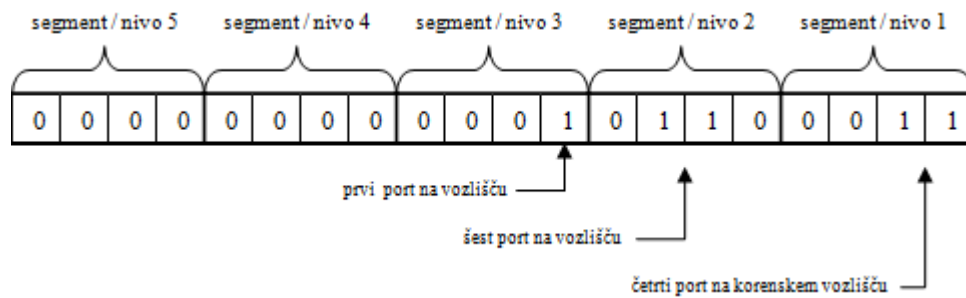
Zaradi uvedbe full-duplex komunikacije USB 3.0 izboljšuje izrabo vodila z opustitvijo izpraševanja naprav. Zdaj uporablja asinhrono komunikacijo. Ko želi določena naprava izvesti prenos podatkov proti gostitelju, torej vhodno transakcijo, pošlje do gostitelja paket ERDY. Nato pa v primernem času gostitelj napravi posveti pozornost.

9.5.1 Usmerjanje paketov po vodilu

Za dostavo podatkovnih, kot tudi večine drugih paketov uporablja USB 3.0 tako imenovani naslovni trojček, ki je sestavljen iz treh polj v glavi paketa. Ta polja so naslov naprave, številka končne točke in smer prenosa. Ta informacija je zadostna, da se izvede prenos podatkov proti točno določeni napravi. V primeru, da paket pošlje naprava proti gostitelju, bo v naslovnem polju zabeležen naslov naprave in končne točke, ki je izvedla transakcijo. Ker USB 3.0 dostavlja pakete točno določeni napravi, je znotraj paketa potrebna tudi usmerjevalna informacija (*slika 34*), da se lahko paket usmerja skozi vozlišča.

Usmerjevalno informacijo berejo vozlišča, da ugotovijo na katera izhodna vrata v vozlišču naj paket usmerijo. Polje sestavlja 20 bitov, ki bi jih lahko razdelili na 5 segmentov po 4 bite. Vsak segment predstavlja usmerjevalno informacijo za en nivo. Segmentov je 5, kar

predstavlja 5 nivojev oziroma največjo globino v topologiji vodila USB. Posamezni segment je sestavljen iz 4 bitov, zato lahko v posameznem vozlišču naslovimo do 16 izhodnih vrat. Ko se vozlišče vključi v vodilo USB, dobi v enumeracijskem procesu informacijo o globini vozlišča, kar dejansko predstavlja nivo v topologiji USB.



Slika 34: Primer usmerjevalne informacije. Ko paket prispe do vozlišča, ta pomnoži vrednost globine vozlišča s štiri. Rezultat pove kateri segment v usmerjevalni informaciji naj bere in s tem na katera vrata naj pošlje paket.

9.5.2 Eksplozijske transakcije

USB 3.0 pri transakcijah uvaja zelo pomembno novost, to so eksplozijske transakcije (angl. burst transaction). Eksplozijske transakcije omogočajo, da naprava ali gostitelj nepretrgoma pošilja podatkovne pakete pri tem pa pošiljatelju ni potrebno čakati na potrditveni paket. Največje število zaporednih transakcij, ki jih lahko končna točka pošlje brez prejete potrditve, je zabeleženo v deskriptorju. Za eksplozijske transakcije veljajo naslednja pravila:

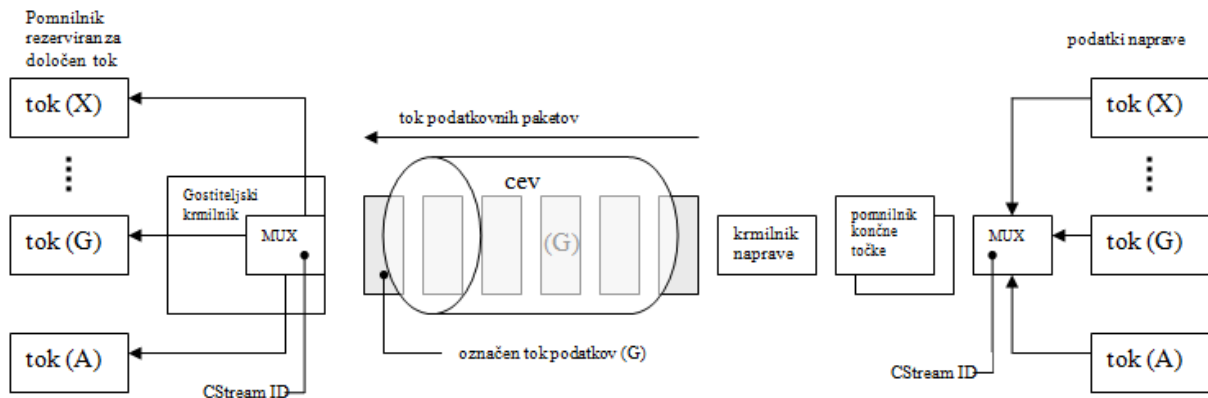
- Največje število nepretrgoma poslanih paketov preden sprejemnik pošlje ACK TP, je omejeno na največje število eksplozijskih transakcij, ki jih omogoča končna točka in vrednosti polja *NumP* v ACK TP. Polje *NumP* pove koliko prostega medpomnilnika ima sprejemnik pripravljenega za sprejem paketov. Vrednost *NumP* se lahko dinamično spreminja za vsak poslani ACK TP.
- Eksplozijske transakcije se končajo, ko je polje *NumP* enako nič. Lahko tudi kadar oddajnik pošlje paket, ki je manjši od *maxPacketSize*. Ali kadar je postavljena zastavica *End of burst* v glavi podatkovnega paketa, kar pomeni, da oddajnik nima več podatkov za to napravo.

9.5.3 Tokovi

Naslednja velika novost pri prenosu podatkov so tokovi (angl. Streams). USB 2.0 je imel možnost prenosa enega samega toka med gostiteljem in napravo. USB 3.0 pa je dodal novo funkcionalnost na ravni protokola, namreč za masivne prenose lahko uporabljamo vse do 65533 tokov. Navadni masivni paketi se ne razlikujejo od tokovnih paketov. Tokovni protokol upravlja masivne tokovne prenose samo preko manipulacije polja *Stream ID*.

Polje *Stream ID* je prisotno v ACK, NRDY, ERDY in podatkovnih paketih. To 16-bitno polje ima rezervirane tri vrednosti *Stream ID*, ki se uporabljajo za upravljanje tokovnih prenosov. Ostale vrednosti pa se uporabljajo za označevanje različnih tokov. Glede na število tokov, ki jih uporablja določena naprava, potrebujemo tudi tolikšno število ločenega rezerviranega medpomnilnika. Tokovni protokol omogoča vzpostavitev trenutnega toka (angl. Current Stream ID, CStreamID) preko ACK TP. Gostitelj uporablja *CStream ID* za pripravo ustreznega

medpomnilnika na strani gostitelja, preko katerega se bo nato prenašal tok podatkov. Naprava pa s *CStream ID* pripravi medpomnilnik na strani naprave, ki bo v uporabi.



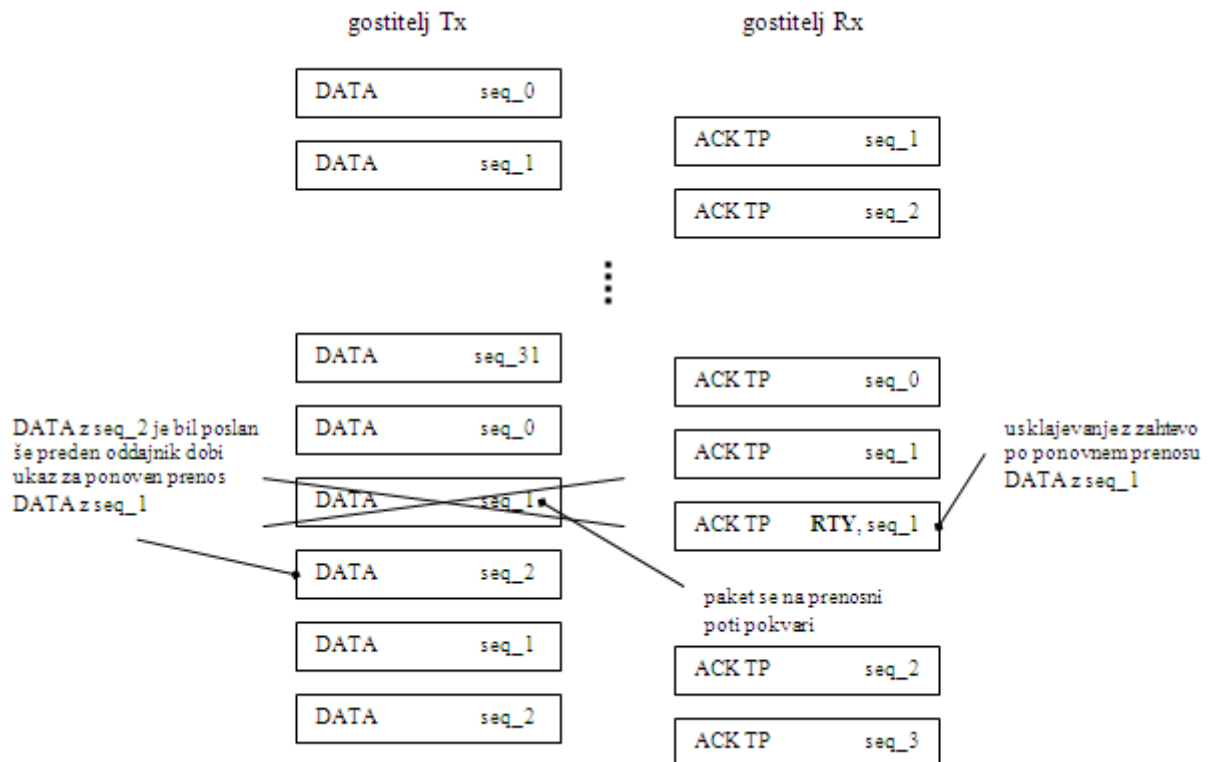
Slika 35: Shema prikazuje primer masivnega tokovnega prenosa.

Slika 35 prikazuje vhodne tokovne transakcije. Ko ima naprava pripravljene podatke za določen tok (G v našem primeru) uporabi ERDY paket v katerem nosi tudi informacijo o *CStream ID*. Gostitelj bo poslal vhodni ACK TP napravi kateri pripada *CStream ID*. Zdaj lahko naprava pošlje podatkovne pakete, označene s *CStream ID* proti gostitelju. Ko gostitelj dobi podatke, prebere *CStream ID*, da preusmeri podatke k ustreznemu medpomnilniku. Ko naprava pošlje gostitelju vse podatke, prekine tok. V našem primeru lahko tudi gostitelj prekine tok podatkov, če nima na voljo več prostega medpomnilnika. Če se to zgodi lahko kasneje gostitelj, ko se sprost predpomnilnik, napravo asinhrono obvesti, da je sprostil medpomnilnik in naprava lahko prenese preostale podatke.

9.5.4 Vrste prenosov super-speed

USB 3.0 pozna enake vrste prenosov kot standard USB 2.0. Da pa lahko USB 3.0 zagotovi hitrejše prenose podatkov, je princip transakcij nekoliko drugačen. Slika 36 prikazuje primer izhodne masivne transakcije. Ko je gostitelj pripravljen na transakcijo, pošlje enega ali več podatkovnih paketov. Če je transakcija uspešna, pošlje naprava ACK TP z zaporedno številko podatkovnega paketa, ki ga je uspešno prejela (kot potrditev) in število podatkovnih paketov, ki jih lahko naprava prejme brez takojšnje potrditve. V primeru, da naprava trenutno ne more sprejeti podatkovni paket, pošlje gostitelju NRDY TP. Gostitelj se lahko nato posveti drugim transakcijam, dokler od naprave ne dobi ERDY TP. Gostitelj vedno označi prvi podatkovni paket z zaporedno številko 0. Zaporedne številke podatkovnih paketov nato inkrementira vse do 31, ko začne ponovno šteti od 0 naprej. Naprava potrdi vsak uspešno prejet paket, toda ne nujno takoj po prejetju paketa. V primeru, da se podatkovni paket na vodilu pokvari, ga naprava zavrže. Gostitelju pa pošlje ACK s postavljenimi zastavico *retry* in zaporedno številko paketa, ki ga mora ponovno poslati. Masivna transakcija se konča, ko so bili poslani vsi podatkovni paketi. V tem primeru bo imel zadnji paket postavljeno zastavico *end of burst*, ki označuje konec prenosa. Masivna transakcija se konča tudi v primeru, da se na vodilo pošlje paket krajši od *MaxPacketSize*, saj se predpostavlja, da je bil to zadnji paket.

Vhodno masivno transakcijo gostitelj vedno začne z ACK TP proti končni točki, ta pa mu nato odgovori s podatki. Masivni prenos pri USB 3.0 omogočajo 1024 B podatkov na paket, USB 2.0 je omogočal do 512 B.



Slika 36: Primer masivne transakcije.

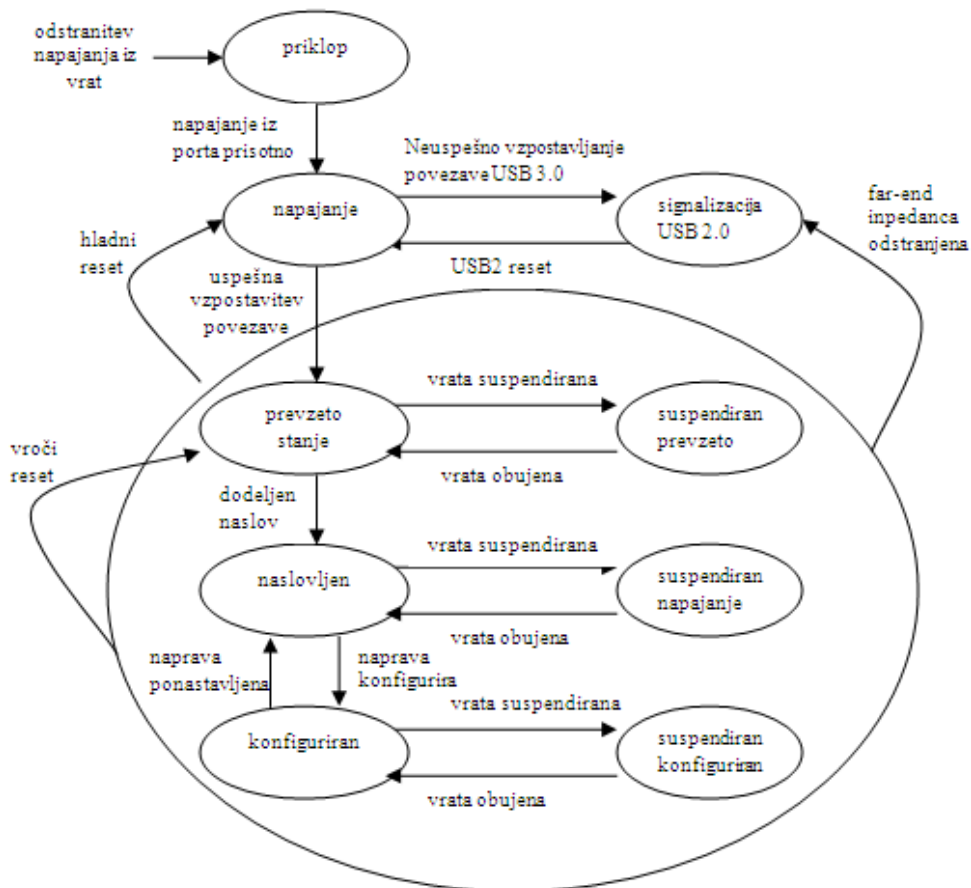
Kontrolni prenos je kot pri USB 2.0 vedno sestavljen z nastavitveno in statusno fazo, glede na vrsto ukaza pa lahko vsebuje tudi podatkovno fazo. Za prenos podatkovnih paketov v podatkovni fazi se uporablja masivni prenos, ki pa ne sme uporabljati eksplozijskih transakcij. Torej se mora potrditi vsak poslan paket, preden se pošlje novega.

Prekinitvene transakcije se izvajajo samo ob določenih servisnih intervalih in imajo do vodila zagotovljen dostop. Kako pogosti so servisni intervali, je zabeleženo v deskriptorju končne točke. Najmanjša enota servisnega intervala je 125 μ s. Prekinitvene transakcije uporabljajo masiven prenos pri tem omogočajo do tri eksplozijske transakcije, preden mora sprejemnik poslati potrditveni paket. V primeru pokvarjenega paketa se bo paket poslal ponovno, toda ne nujno v istem servisnem intervalu. V primeru, da pride do zastoja v končni točki in začasno ne more prejeti ali poslati paket, to sporoči z NRDY TP. Zdaj gostitelj v servisnih intervalih ne bo več poskusil komunicirati s to končno točko, dokler ta ne pošlje ERDY TP.

Izohroni prenos izvaja transakcije v servisnih intervalih, ki jih ima določene v deskriptorju končne točke. V enem servisnem intervalu lahko pošlje do 48 podatkovnih paketov. Poslani paketi se ne potrjujejo, prispeli pokvarjeni paketi pa se zavržejo. Ko izohroni končni točki zmanjka podatkovnih paketov, se zadnji podatkovni paket opremi z zastavico *last packet flag*, kar pomeni, da je končna točka ali gostitelj končal s pošiljanjem.

9.5.5 KONFIGURIRANJE NAPRAVE USB 3.0 IN DESKRIPTORJI

Naprava USB 3.0 lahko deluje hitreje z manjšo porabo električne energije, kot naprava USB 2.0. Če pa pride do situacije, da naprava USB 3.0 ni sposobna delovati s signalizacijo SS, npr. zaradi omejitve gostitelja, le ta preklopi na signalizacijo HS. Po priklopu naprave USB 3.0 se izvede podoben postopek prehajanja stanj kot pri napravi USB 2.0 (slika 37).



Slika 37: Diagram prehajanja stanj pri vključitvi naprave USB 3.0.

Priklop: naprava USB je ravnokar fizično vključena v vozlišče.

Napajanje: V tem stanju naprava dobi napajanje Vbus iz korenkega vozlišča. Nekatere naprave lahko dobijo zunanje napajanje, še preden se priključijo na vrata USB, toda v stanje napajanje preidejo šele takrat, ko dobijo napajanje Vbus.

Privzeto stanje: Po tem, ko naprava USB 3.0 prejme napajanje, ne bo izvedla nobene transakcije preden povezava ne prestopi uspešnega postopka pripravljanja povezave (angl. link training). V privzetem stanju je naprava dostopna preko privzetega naslova. V primeru, da sosednji vrati ne uspeja pripraviti povezave pri signalizaciji SS, poizkuša naprava vzpostaviti povezavo s signalizacijo USB 2.0.

Naslavljanje: Vse naprave po pridobljenem resetu dobijo privzeti naslov. Nato pa v tem stanju dobijo unikatni naslov, ki ga napravi dodeli gostitelj. Tudi v primeru, ko je naprava suspendirana, še vedno obdrži unikatni naslov.

Konfiguriranje: Preden lahko napravo uporabimo, jo je potrebno konfigurirati. To pomeni, da naprava uspešno izvede ukaz *SetConfiguration()*. Pri tem se ponastavijo vse vrednosti, ki so povezane s končnimi točkami na privzeto vrednost. Ko naprava preide v to stanje, je pripravljena na polno komunikacijo.

Suspend: Je varčevalno stanje U3, ko naprava trenutno ni v uporabi. Ko je naprava v tem stanju, ne izgubi naslova in trenutne konfiguracije. Naprava preide iz tega stanja, ko dobi signal obujanja. Če ima naprava omogočeno funkcijo oddaljenega obujanja, lahko zahteva od gostitelja izhod iz suspendiranega stanja preko signala *resume*. To funkcijo je potrebno omogočiti med konfiguracijo naprave.

Deskriptorji USB 3.0 predstavljajo nadgradnjo obstoječih deskriptorjev USB 2.0. Vsaka naprava USB 3.0 mora biti sposobna delovati tudi pri eni od signalizacij USB 2.0, kar pomeni, da mora vsebovati tudi vse deskriptorje USB 2.0. Naprava USB 3.0 ima lahko različne konfiguracije, ki so odvisne od hitrosti signalizacije. Lahko, da so določene funkcije možne le pri signalizaciji USB 3.0. Zato mora naprava ob zahtevi vrniti samo deskriptorje, ki so namenjeni določeni hitrosti signalizacije. Vedeti je tudi potrebno, da je vozlišče USB edina naprava, ki lahko hkrati deluje pri signalizaciji USB 2.0 in USB 3.0. USB 3.0 uvaja dodatne deskriptorje, ki jih morajo vsebovati vse naprave USB 3.0.

Binary Device Object Store (BOS) je razširljivo ogrodje za opis dodatnih zmožnosti poleg tistih, ki so del standardnih specifikacij USB. Je podoben konfiguracijskemu deskriptorju. Eden izmed bolj pomembnih deskriptorjev, ki se nahajajo nižje v hierarhiji, je *Super Speed device capability* deskriptor. V njem se med drugim nahajajo informacije o:

- katere hitrosti signalizacije so podprte poleg SS,
- ali je naprava sposobna generacije Latency tolerance Messages. Gre za upravljanjem s časom, ki poteče po tem, ko naprava pošlje paket ERDY gostitelju in ko se gostitelj na ta paket odzove. Funkcija doprinese k varčevanju z energijo,
- ali je naprava polno funkcionalna, ko deluje s hitrostjo nižjo od SS,
- čas v μs , ki je potreben, da naprava preide iz U1 ali U2 v varčevalno stanje U0. Ta čas je mišljen za vrata na napravi in ne za ostala vozlišča, ki vodijo do gostitelja.

Zaradi novih funkcionalnosti v transakcijah USB 3.0, kot so eksplozijske in tokovne transakcije je bilo potrebno dodati še en pomemben deskriptor, to je *SuperSpeed endpoint companion*. Ta deskriptor vsebuje karakteristike končnih točk SS, kot so:

- največje število eksplozijskih transakcij, ki jih podpira končna točka. Največje število zaporednih transakcij je lahko 16, najmanjše pa ena kar pomeni, da končna točka ne podpira eksplozijskih transakcij,
- koliko tokov podpira masivna končna točka. Vrednost 0 pomeni, da ne podpira tokov,
- za izohrone končne točke definira največje število paketov znotraj servisnega intervala,
- končne točke, ki potrebujejo periodični dostop do vodila sporočijo količino bitov, ki bodo poslani znotraj servisnega intervala.

10 ZAKLJUČEK

Danes si brez univerzalnega serijskega vodila ne moremo več predstavljati sodobnega računalnika oziroma katerokoli bolj napredne naprave. Prvih nekaj let po predstavitvi so mnogi razmišljali ali bo to še en standard, ki bo delal gnečo na zadnji strani računalnika. Čeprav je bil sprva namenjen povezovanju računalnika s telekomunikacijsko opremo, je s časoma postal resnično univerzalen gostitelj za periferne naprave vseh vrst tudi take, ki nimajo nobene povezave z računalništvom. Po nekaterih podatkih so do danes prodali več kot 3 milijard naprav, ki vsebujejo priključek USB. Menim, da je za takšen uspeh vodila USB zaslužen predvsem dejstvo, da so se pri projektu združila vodilna podjetja in so tako našli skupni jezik. Dodatno pa je k popularnosti prispevala še odprtost standarda.

Med pisanjem diplomske naloge sem dobil dobro predstavo, kako se je standard USB razvijal, kakšne omejitve so imele prve revizije standarda in v probleme s katerimi se soočajo razvijalci pri razvijanju standarda. Dobil sem tudi dober vpogled kako deluje komunikacija na nivoju protokola. Iz protokola USB 2.0 se lepo vidi, da so razvijalci potrebovali novi koncept komunikacije na nivoju protokola, saj je že predstavljal ozko grlo pri hitrih transakcijah. To so pokazali z drastično spremembo protokola pri standardu USB 3.0, ki ima po besedah razvijalcev še nekaj prostora za izboljšave. Razvijalci so se pri prehodu na USB 3.0 tudi držali osnovnega cilja, to je združljivost s starejšimi napravami. Na koncu ni potrebno posebej poudarjati, da je standard USB velika zgodba o uspehu. Danes USB najdemo praktično v vsaki napravi, ki deluje na električno energijo. Menim, da si je USB 3.0 dokončno pridobil veliko prednost pred konkurenčnim FireWire. Če pogledamo na trg, trenutno ni pravega konkurenta vodilu USB. Glede na zmogljivosti USB 3.0 pa se še nekaj časa ni potrebno bati za njegovo prihodnost.

11 LITERATURA

USB 2.0:

- [1] (2010) Universal Serial Bus Revision 2.0 Specification. Dostopno na:
<http://www.usb.org/developers/docs/>
- [2] (2010) The USB Vision: 10 Years Later. Dostopno na:
<http://www.everythingusb.com/timeline.html>
- [3] (2010) USB Background. Dostopno na:
<http://www.totalphase.com/support/kb/10047/>
- [4] (2010) USB Made Simple. Dostopno na:
<http://www.usbmadesimple.co.uk/index.html>
- [5] (2010) USB in a NutShell. Dostopno na:
<http://www.beyondlogic.org/usbnutshell/usb1.shtml>
- [6] (2010) Wikipedia: Universal Serial Bus. Dostopno na:
http://www.en.wikipedia.org/wiki/Universal_Serial_Bus
- [7] Don Anderson, Universal Serial Bus System Architecture, Mindshare Inc., Addison-Wesley, 2001

USB 3.0:

- [7] (2010) SuperSpeed USB 3.0: Ubiquitous Interconnect for Next Generation Consumer Applications. Dostopno na:
<http://www.design-reuse.com/articles/20937/superspeed-usb-3-0-ip.html>
- [8] (2010) Inside USB 3.0. Dostopno na:
http://www.reghardware.com/2009/05/25/superspeed_usb_3_guide/print.html
- [9] (2010) USB 3.0 SuperSpeed Overview. Dostopno na:
<http://www.inno-logic.com/resourcesUSB3.html>
- [10] (2010) Universal Serial Bus 3.0 Specification. Dostopno na:
http://www.usb.org/developers/docs/usb_30_spec_060910.zip
- [11] (2010) SuperSpeed USB (USB 3.0): More than just a speed increase. Dostopno na:
<http://www.eetimes.com/design/analog-design/4201110/SuperSpeed-USB--USB-3-0--More-than-just-a-speed-increase>

USB On-The-Go:

- [12] (2010) USB OTG: A tutorial on USB On-The Go. Dostopno na:
http://www.stericsson.com/sales_marketing_resources/USB_OTGBR_1.pdf

- [13] (2010) Wikipedia: USB On-The-Go. Dostopno na:
http://en.wikipedia.org/wiki/USB_On-The-Go

- [14] (2010) Universal Serial Bus Revision 2.0 Specification: On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification. Dostopno na:
<http://www.usb.org/developers/docs/>

- [15] Bojan Voglar, diplomsko delo: Dopolnitev specifikacij USB 2.0 z dodatkom OTG, Fakulteta za računalništvo in informatiko, 2004.