

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF MICROELECTRONICS



MĚŘENÍ PARAMETRŮ KOMUNIKACE PŘES PCI EXPRESS

MEASURING COMMUNICATION'S PARAMETERS OVER PCI EXPRESS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE

ONDŘEJ DUJÍČEK

ODBORNÝ VEDOUCÍ PRÁCE

ING. JIŘÍ MATOUŠEK

VEDOUCÍ PRÁCE

ING. MARIÁN PRISTACH

BRNO 2016

Bakalářská práce

bakalářský studijní obor **Mikroelektronika a technologie**
Ústav mikroelektroniky

Student: Ondřej Dujíček

ID: 155630

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Měření parametrů komunikace přes sběrnici PCI Express

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte specifikaci sběrnice PCI Express verze 3.0. Zaměřte se především na parametry ovlivňující maximální propustnost. Navrhněte jednotku pro měření parametrů komunikace přes sběrnici PCI Express, která bude určena pro čip FPGA na akcelerační kartě s PCI Express rozhraním. Navrženou jednotku implementujte v jazyce VHDL. S využitím implementované jednotky proveďte měření parametrů komunikace přes sběrnici PCI Express. Zhodnoťte naměřené výsledky a diskutujte možnosti dalších rozšíření.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 8.2.2016

Termín odevzdání: 2.6.2016

Vedoucí práce: Ing. Marián Pristach

Konzultant bakalářské práce: Ing. Jiří Matoušek

doc. Ing. Jiří Háze, Ph.D., *předseda oborové rady*

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato práce se zabývá parametry ovlivňujícími propustnost PCI Express sběrnice a jejím výsledkem je návrh a implementace jednotky pro měření parametrů komunikace přes PCI Express sběrnici. Jednotka je implementovaná v jazyce VHDL a zaměřuje se na generování provozu až o rychlosti 100 Gb/s a jeho měření. Implementovaná jednotka je schopná pracovat při frekvenci 200 MHz v čipu FPGA Virtex 7 umístěném na akcelerační kartě COMBO-100G. Implementovaná jednotka je řízena ze software přes rozhraní MI32 a je schopna měřit množství přenesených paketů, množství přenesených dat přijatých a odeslaných. Tyto informace dále exportovat do software pomocí rozhraní MI32.

KLÍČOVÁ SLOVA

PCI Express 3.0, VHDL, FPGA, VIRTEX 7, NETCOPE, COMBO-100G, FIRMWARE

ABSTRACT

This bachelor thesis deals with parameters affecting throughput of PCI Express bus and its main result is a design and implementation of a unit for measuring parameters of communication over PCI Express bus. The unit is implemented in VHDL language and its support on generating and measuring traffic at speeds up to 100 Gbps. Unit's operation frequency, when implemented in Virtex 7 available at COMBO-100G , is 200 MHz. The implemented unit is controlled from software through MI32 interface and it is able to measure the amount of transferred packets and data in both receive and transmit directions. This information can be exported into software using MI32 interface.

KEYWORDS

PCI Express 3.0, VHDL, FPGA, VIRTEX 7, NETCOPE, COMBO-100G, FIRMWARE

DUJIČEK, O. *Měření parametrů komunikace přes sběrnici PCI Express*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 37 s. Vedoucí bakalářské práce Ing. Marián Pristach.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma *Měření parametrů komunikace přes PCI EXPRESS* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

Poděkování

Děkuji odbornému vedoucímu Ing. Jiřímu Matouškovi, vedoucímu Ing. Mariánu Pristachovi a kolegům z projektu Liberouter Ing. Viktoru Pušovi Ph.D., a Ing. Martinu Špinlerovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování bakalářské práce.

V Brně dne 2.6.2016

.....

podpis autora

Obsah

Seznam obrázků	9
SEZNAM TABULEK	10
ÚVOD	11
1 PCI Express	12
1.1 Přehled Vrstev PCI Express	13
1.1.1 Transakční vrstva	14
1.1.2 Linková vrstva.....	15
1.1.3 Fyzická vrstva	15
1.2 Transakce PCI Express.....	16
1.2.1 Pakety transakční vrstvy.....	16
1.2.2 Transakce linkové vrstvy	17
1.3 Rychlost přenášení dat PCI Express.....	18
1.4 Režie PCI Express	18
1.5 Komunikační režie	19
1.6 Vliv parametrů systému	19
1.6.1 Maximální velikost obsahu paketu.....	19
1.6.2 Maximální velikost požadavku pro čtení	20
1.7 Systém kódování PCI Express	21
2 Vstupní a výstupní rozhraní	22
2.1 DMA rozhraní.....	22
2.2 MI32 rozhraní(<i>Memory Interface</i>).....	23
3 Měřené parametry PCI Express	25
3.1 Propustnost.....	25
3.2 Latence	25
3.3 Doba neschopnosti přijímání informací.....	25
4 Návrh komponenty pro měření parametrů komunikace přes Pci Expres	26
4.1 Komponenta FIFO_IN_MEM	27
4.2 Komponenta TRANS_GEN	27
4.3 Komponenta RX_TX.....	28
4.4 Komponenta TRANS_MARKER.....	28

5	Implementace	29
5.1	Principy použité při implementaci	29
5.2	Popis implementace	29
5.3	Problémy vzniklé při implementaci	30
5.4	Implementované řízení jednotky	30
5.5	Komponenta TRANS_GEN	32
5.6	Komponenta RX_TX	33
5.7	Komponenta TRANS_MARKER	34
5.8	Spotřeba zdrojů implementované jednotky	35
6	Závěr	36
	Seznam literatury	37
	Seznam symbolů, veličin a zkratk	38

SEZNAM OBRÁZKŮ

Obr. 1: PCI Express propojení (Full-Duplex) [2]	12
Obr. 2: Koncept vrstev PCI Express [2]	13
Obr. 3: Obsah paketu na jednotlivých vrstvách [2]	13
Obr. 4: Obecný Formát TLP [2]	14
Obr. 5: Obecný tvar DLLP [2].....	15
Obr. 6: Ukázka TLP v PCI Express generace 3 [2]	18
Obr. 7: Efektivita přenášených paketů [1]	20
Obr. 8: Zapisování přes rozhraní MI [4]	24
Obr. 9: Čtení přes rozhraní MI [4]	24
Obr. 10: Schéma jednotky pro měření parametrů komunikace přes PCI Express.....	26
Obr. 11: Schéma implementované jednotky pro měření parametrů komunikace přes PCI Express	30
Obr. 12: Schéma proudění informací přes TRANS_GEN.....	32
Obr. 13: Schéma proudění informací v komponentě RX_TX	33
Obr. 14: Schéma proudění informací v TRANS_MARKER	34

SEZNAM TABULEK

Tabulka 1: Maximální teoretická přenosová rychlost pro jeden směr (Gb/s).....	18
Tabulka 2: ukázka kódování 8b/10b [1]	21
Tabulka 3: signály rozhraní DMA [5]	22
Tabulka 4: Formát hlavičky DMA_UP [5].....	23
Tabulka 5: Formát hlavičky DMA_DOWN [5]	23
Tabulka 6: Signály MI rozhraní [4]	23
Tabulka 7: Parametry generátoru TRANS_GEN	27
Tabulka 8: Využívané zdroje při konfiguraci pro jeden PCI Express x8	35

ÚVOD

Při vývoji jakékoliv technologie jsou potřeba nástroje, jedním z těchto nástrojů by se měl stát výsledek této bakalářské práce. Cílem bakalářské práce bylo navrhnout a implementovat komponentu v jazyce VHDL pro měření parametrů komunikace přes PCI Express.

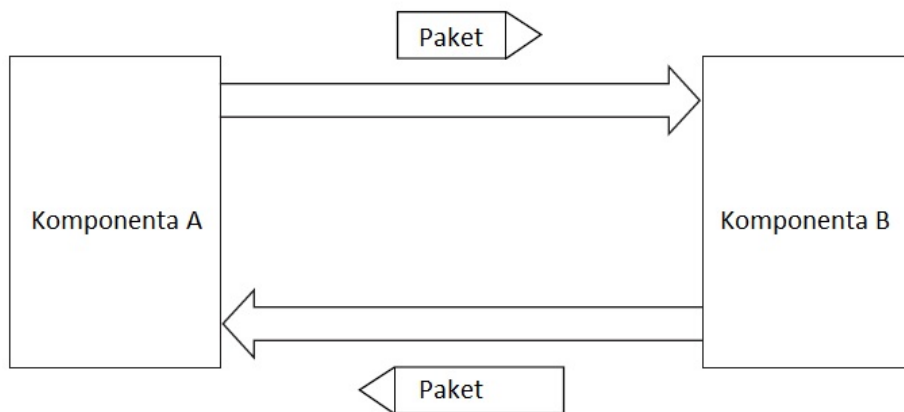
Důvodem vzniku této bakalářské práce je povaha PCI Express systému. Systém PCI Express se skládá z několika vrstev, přičemž každá z vrstev vytváří svoji režii. Reálný provoz uskutečňovaný přes PCI Express je velmi proměnlivý a tím znemožňuje přesný výpočet přenesených dat a určení celkového chování. Dostupným řešením v projektu Liberouter je PCI Express analyzátor generace 3, který monitoruje dění, ale postrádá možnost generování provozu a testování PCI Express linky.

Implementovaná jednotka má sloužit v projektu Liberouter, který se zabývá vývojem vysokorychlostním zabezpečením a monitorování sítě. Použití jednotky se plánuje v čipu FPGA Virtex 7 na akcelerační kartě COMBO-100G.

Bakalářská práce je rozdělena do šesti hlavních kapitol. V první kapitole je představen systém PCI Express a přesně rozebrány věci, které se týkají propustnosti. V další kapitole je popsáno rozhraní, ve kterém je navržena jednotka umístěna. V kapitole tři jsou uvedeny parametry komunikace, které jsou předmětem měření. V kapitole čtyři je rozepsán návrh jednotky pro měření parametrů komunikace. V kapitole pět je popsán způsob implementace a problémy vzniklé při implementaci jednotky pro měření parametrů komunikace. Kapitola šest uzavírá bakalářskou práci a shrnuje dosažené výsledky včetně postupu dalšího vývoje implementované jednotky.

1 PCI EXPRESS

PCI Express je vysokorychlostní sériová sběrnice, která se stala rozšířeným standardem v informatice. Reprezentuje *Full-Duplex* komunikaci zařízení (viz. Obr. 1).



Obr. 1: PCI Express propojení (Full-Duplex) [2]

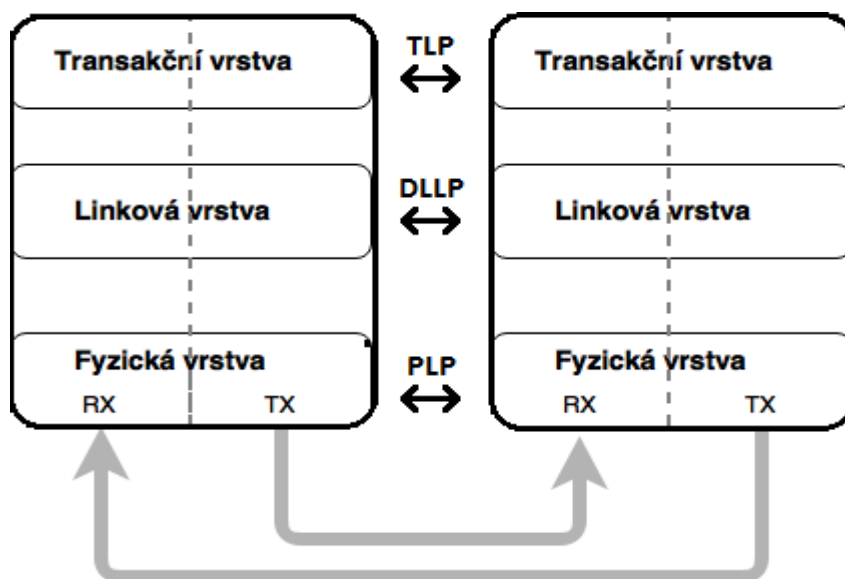
Hlavní vlastnosti a pravidla PCI Express

Základní zapojení komponent se skládá z dvojice jednosměrných spojů. Úroveň rychlosti signalizace je definována jedenkrát pro přenosovou strukturu. Každá generace podporuje jinou maximální přenosovou rychlost pro linku (viz tabulka. 1). Každá linka představuje dvojici diferenciálních párů, jeden diferenciální pár pro vysílání a jeden pro přijímání.

Zvýšení přenosové rychlosti lze dosáhnout spojením více linek, jejichž počet se značí xN , kde N je počet spojených linek (např. spojením 8 linek s propustností 2,5 Gb/s dosáhneme přenosové rychlosti 20 Gb/s v obou směrech). Tato přenosová rychlost značí všechna přenesená data včetně režie.

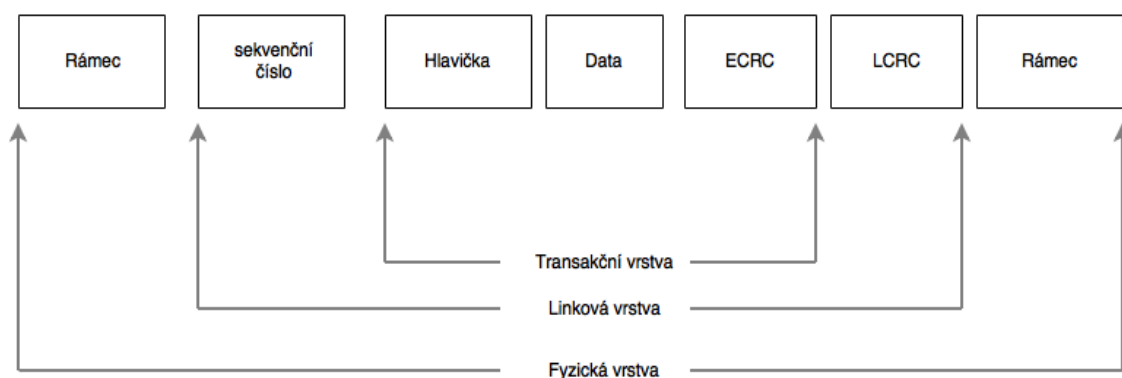
Při inicializaci zařízení, každá PCI Express linka nastavuje frekvenci odeslání transakcí mezi koncovými body, šířku sběrnice pro kterou realizují komunikaci. V době nastavování není připojený firmware ani operační systém. Symetrie sběrnice musí být dodržena pro oba směry, tj. x16 sběrnice musí mít 16 diferenciálních signálových párů v obou směrech. PCIe 3.0 používá vrstvý komunikační model na rozdíl od předchozích verzí. [2]

1.1 Přehled Vrstev PCI Express



Obr. 2: Koncept vrstev PCI Express [2]

PCI Express používá transakce pro komunikaci koncových zařízení mezi sebou. Na obr. 2 je koncept vrstev PCI Express, kde je i znázorněn tok paketů, které protékají přes jednotlivé vrstvy. Transakce jsou ve formě paketů, které jsou přenášeny pomocí linkové vrstvy od vysílací strany k přijímací. Vzhledem k tomu, že paketový tok je přenášen přes další vrstvy, přidávají se další nezbytné informace pro manipulaci s pakety na těchto vrstvách. Přijímací strana má proces opačný. Pakety jsou transformovány z paketu fyzické vrstvy do paketu linkové vrstvy a dále pak do paketu transakční vrstvy. [2]

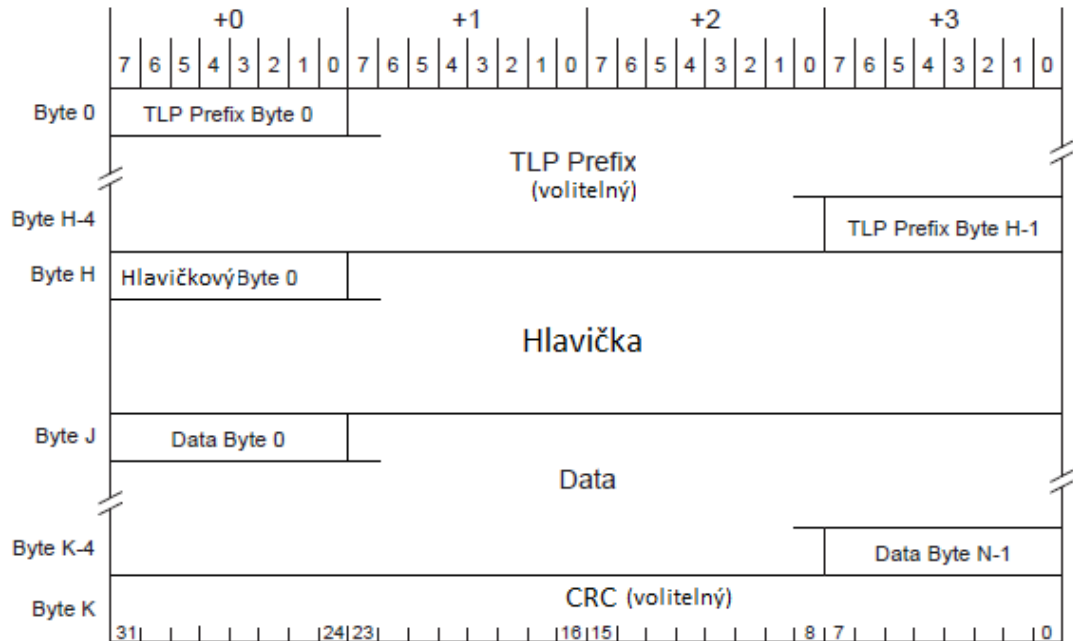


Obr. 3: Obsah paketu na jednotlivých vrstvách [2]

Na obr. 3 je vidět rezie přidána jednotlivými vrstvami, kterými prochází paket. Model na obr. 3 je zjednodušený pro obecnou představu o provozu, který probíhá na jednotlivých vrstvách. V dalších podkapitolách jsou popsány funkce jednotlivých vrstev. [2]

1.1.1 Transakční vrstva

Horní vrstva architektury, zodpovídá za složení a rozložení transakce do jednotlivých paketů *TLP* (*Transaction Layer Packets*). Pakety *TLP* se používají jako komunikační prostředek k vysílání požadavků pro čtení a zápis a informuje o dalších událostech.



Obr. 4: Obecný Formát TLP [2]

Transportní vrstva je také zodpovědná za řízení kreditového systému Flow Control pro jednotlivé *TLP*.

Každý *TLP* má jedinečný identifikátor, což umožňuje odpovídat přímo žadateli. Pakety jsou ve formátu, který podporuje rozdílné formy adresování v závislosti na typu transakce. [2]

Řízení toku

Řízení toku *FC* (Flow Control) je funkce transakční vrstvy, která jej používá jako ochranný prostředek proti přetečení vyrovnávací paměti u přijímací strany. Mechanismus *FC* je založen na kreditovém systému, který informuje o velikosti fronty nebo velikosti volného místa ve vyrovnávací paměti na druhé straně linky. *FC* využívá linkovou vrstvu pro odesílání informačních zpráv ve formě paketů *DLLP* (*Data Link Layer Packets*). [2]

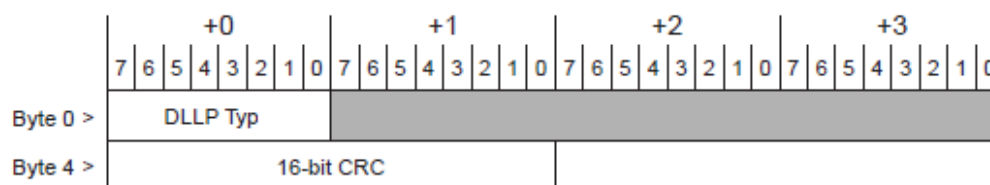
1.1.2 Linková vrstva

Linková vrstva slouží jako prostředník mezi transakční vrstvou a fyzickou. Jejím hlavním účelem je řízení linky a udržení integrity dat. Vkládá detekci poruch a opravy chyb.

Odesílací strana linkové vrstvy přijímá *TLP* složené transportní vrstvou, vypočítá a aplikuje ochranu dat a vloží sekvenční číslo *TLP*. Poté předloží upravený *TLP* fyzické vrstvě a ta jej odešle přes linku.

Přijímací strana linkové vrstvy je zodpovědná za kontrolu integrity přijatého *TLP* a jeho přeložení transportní vrstvě pro další zpracování. Při detekování porušeného *TLP* je vrstva zodpovědná za vydání požadavku pro opětovné odeslání *TLP* do té doby, dokud není *TLP* správně přijat nebo není rozhodnuto o jeho neúspěšném přijetí *TLP*. Linková vrstva také generuje a přijímá pakety, které slouží k řízení provozu linky.

Pro rozeznání od *TLP* se pro tyto pakety používá termín *Data Link Layer Packet* (*DLLP*). Na obr. 5 je ukázka obecného tvaru *DLLP*, v kterém je prázdným polem vyznačena oblast která se mění pro každý typ *DLLP*. [2]



Obr. 5: Obecný tvar *DLLP* [2]

1.1.3 Fyzická vrstva

Fyzická vrstva obsahuje přenosové rozhraní, vstupní a výstupní vyrovnávací paměti, sériově paralelní převodníky, fázové závěsy a obvody pro impedanční přizpůsobení. Také obsahuje logické operace pro inicializaci a údržbu rozhraní.

Fyzická vrstva je zodpovědná za transformaci informací získaných od linkové vrstvy do vhodné podoby sériového výstupu pro přenos přes PCI Express na kompatibilní frekvenci a šířce sběrnice s druhou stranou. Fyzická vrstva realizuje kódování 8b/10b nebo 128b/130b u PCI Express generace 3. [2]

1.2 Transakce PCI Express

Transakce slouží k výměně informací mezi zařízeními. Na obou stranách PCI Express jsou připojeny 2 typy jednotek *requester* a *completer*. *Requester* reprezentuje vysílací stranu v PCI Express, která se dotazuje pomocí transakcí typu *request*. *Completer* přijímá transakce *request* a jeho úkolem je splnit *request* a odeslat zpátky odpověď v podobě *completion* transakce. *Requester* a *completer* je na úrovni transakční vrstvy.

Transakce *request* a *completion* patří mezi *TLP* a jejich obecný formát je na obr. 3. Pro řízení odesílání *TLP* slouží pakety linkové vrstvy. [2]

1.2.1 Pakety transakční vrstvy

Transakce transakční vrstvy jsou rozděleny podle přenášených informací. Např. paměťové transakce přenášejí data z RAM paměti a zapisovat do RAM paměti.

Paměťové transakce jsou:

- Čtení *request/completion*
- Zápis *request*
- Atomické transakce *request/completion*

Paměťové transakce používají 2 rozdílné typy formátů adresování

- Krátký adresovací formát: 32 bitová adresa
- Dlouhý adresovací formát: 64 bitová adresa

Transakce Vstup/Výstup

PCI Express podporuje připojení vstupů a výstupů pro zachování kompatibility se staršími zařízeními. Používá 32 bitový formát adresování.

- Čtení *request/completion*
- Zápis *request/completion*

Konfigurační transakce

Konfigurační transakce se používají pro přístup do registru funkcí zařízení.

Konfigurační transakce zahrnují:

- Čtení *request/completion*
- Zápis *request/completion*

Transakce Zpráva

PCI Express systém umožňuje definovat vlastní zprávy pro komunikaci mezi zařízeními.

1.2.2 Transakce linkové vrstvy

Transakce linkové vrstvy se odesílají v podobě *DLLP* (viz. obr. 5) a jsou rozděleny podle funkce, kterou realizují. Například transakce *acknowledgement* zkráceně *ack* informuje o správně přijatém a nepoškozeném *TLP*. [3]

Transakce *acknowledgement* (ACK)

Přijímací strana linkové vrstvy odešle *ACK* v případě, že v přijaté sekvenci *TLP* nebyla detekována chyba kontrolního součtu (*CRC*) a fyzickou vrstvou byly správně přijaty všechny *TLP* v této sekvenci. Odesílací strana si uchovává každou sekvenci odeslaných *TLP*, dokud není přijat *ACK* s číslem této sekvence. [3]

Transakce *no acknowledgement* (NAK)

Přijímací strana linkové vrstvy odesílá *NAK* v případě, že v některém *TLP* byla detekována chyba v kontrolním součtu nebo fyzická vrstva špatně přijala *TLP*. V takovém případě musí odesílací strana znovu poslat všechny *TLP*, které patřily do této sekvence. [3]

Transakce *INIT_FC*

Po zapnutí nebo po resetu linky je nezbytné inicializovat kredity *FC*. *Init_FC* má dva druhy. Jeden inicializuje kredity v systému PCI Express a nazývá se *Init_FC*, *Init_FC2* potvrzuje inicializované kredity. [2]

Transakce *UPDATE_FC*

Přijímací strana linkové vrstvy průběžně odesílá stav dostupných *FC* kreditů. Tyto kredity vyjadřují množství dat, které je přijímací strana schopna přijmout. Když přijímací strana neodesílá *FC_UPDATE*, tak to znamená, že bylo překročeno množství kreditů a vysílací zařízení přestává odesílat *TLP*. [2]

Řízení spotřeby

Linková vrstva je zodpovědná za integritu dat na přenosové lince. K tomuto patří i napájení. Řízení spotřeby linkové vrstvy umožňuje zjistit vliv funkce na spotřebu, řídit režimy napájení, oznamovat současnou spotřebu a zapínat zařízení. [2]

1.3 Rychlost přenášení dat PCI Express

Rychlost přenášení dat je definována jako množství dat za jednotku času. Tato rychlost vyjadřuje, kolik bitů je PCI Express schopna přenést přes jednu linku (viz. Tabulka 1). Toto číslo neznámá množství přenesených dat, ale celkové přenesené množství bitů odeslaných z fyzické vrstvy.

Další možností, jak zvýšit rychlost, je rozšířit sběrnici. V Tabulce 1 je ukázka přenosových rychlostí pro generaci 1.x, 2.x a 3.0 PCI Express. S rostoucí šířkou roste lineárně i přenosová rychlost. Tyto hodnoty jsou přepočtené po kódování 8b/10b a 128b/130b pro generaci 3. V tabulce 1 jsou hodnoty uvedené hodnoty pro jeden směr přičemž PCI Express má oddělenou přijímací a vysílací stranu. [1]

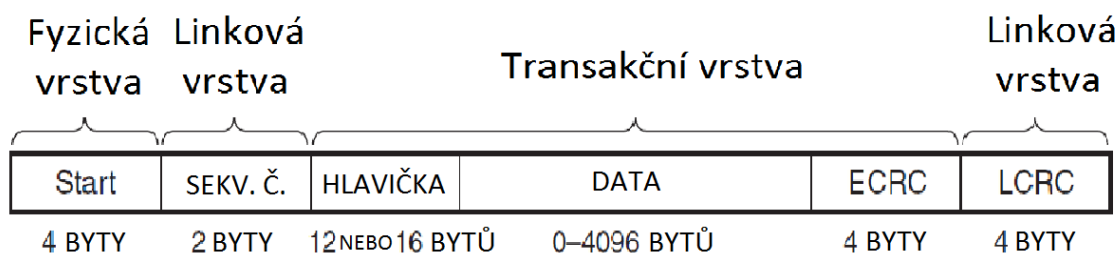
Tabulka 1: Maximální teoretická přenosová rychlost pro jeden směr (Gb/s)

	Šířka sběrnice			
	x1	x2	x4	x8
Generace 1.x	2.0	4.0	8.0	16.0
Generace 2.x	4.0	8.0	16.0	32.0
Generace 3.0	8.0	15.8	31,6	63.2

1.4 Režie PCI Express

Transakční vrstva, linková vrstva a fyzická vrstva přidávají režii, která snižuje efektivitu přenášení dat. Transakční vrstva přidává k datům hlavičku a kontrolní součet (ECRC). Linková vrstva připojuje svůj vlastní kontrolní součet (LCRC) pro garanci úspěšného přenosu přes linku a sekvenční číslo paketu. Fyzická vrstva označuje začátek paketu a jeho konec.

Režie TLP se pohybuje okolo 20 až 30 bytů v závislosti na použitém druhu adresování ve variantách 32 bitová nebo 64 bitová adresace a zvoleném ECRC. Druh adresování se projevuje ve velikosti hlavičky, jak je vidět na Obr. 6. [2]



Obr. 6: Ukázka TLP v PCI Express generace 3 [2]

1.5 Komunikační režie

Řízení odesílání a přijímání paketů je hlavní práce linkové vrstvy. Linková vrstva kontroluje, jestli pakety které odeslala, byly správně přijaty a zasílá informace o správném přijetí. Z důvodů zjednodušení popisu komunikace mezi transakční a linkovou vrstvou se *TLP* dělí na *posted* a *non-posted* transakce.

Non-posted jsou transakce, kterým je vrácena *TLP completion*, které obsahují data nebo bez informací v případě, že je nemůže odeslat. [2]

Non-posted transakce:

- Čtení paměti
- Čtení konfigurace
- Zápis konfigurace
- Čtení vstupu a výstupu
- Zápis do vstupu a výstupu

Na *posted* transakce linková vrstva nevrací *completion*, ale pouze *DLLP ACK* nebo *NAK*.

Posted transakce:

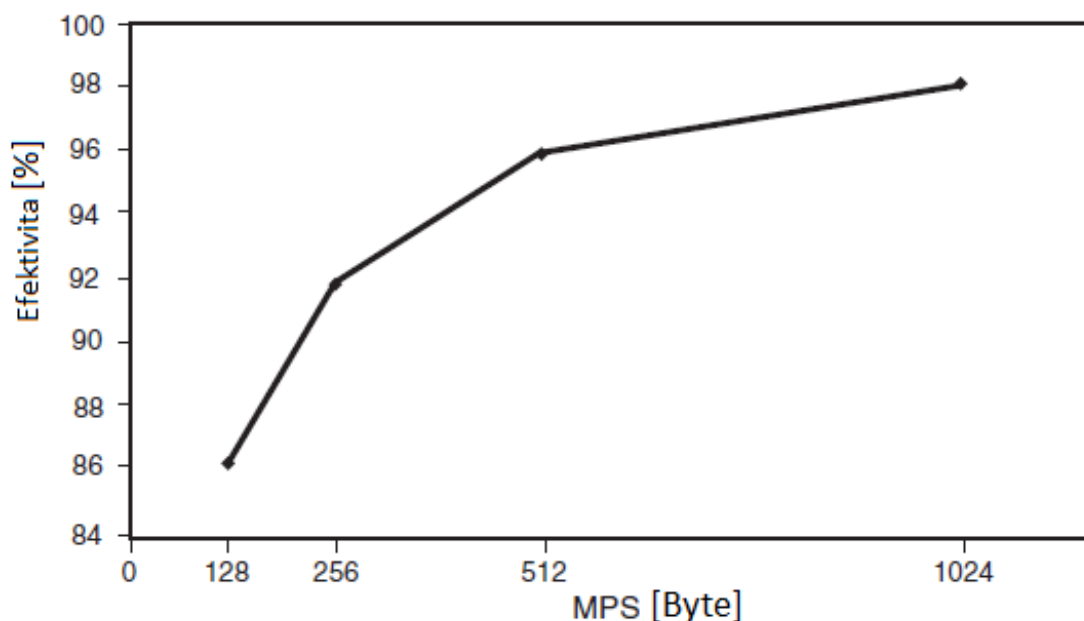
- Zpráva
- Zápis paměti

1.6 Vliv parametrů systému

Kromě hlaviček a typů transakcí existují další prvky, které mají vliv na výkon PCI Express systém. Jsou to parametry *Maximum Payload Size (MPS)* a *Maximum Read Request size (MRRS)*.

1.6.1 Maximální velikost obsahu paketu

Maximální velikost obsahu paketu *MPS (Maximum Payload Size)* je maximální množství dat, které může obsahovat jedna transakce. Systém PCI Express umožňuje posílání transakcí s maximální velikosti až 4096 bytů. Software musí zajistit nepřekračování tohoto limitu. V případě, že odesílací strana má poslat více než je *MPS* tak transakce rozdělena do více *TLP*. Každé zařízení v hierarchii má nastavený tento parametr na stejnou hodnotu, kterou se určuje podle zařízení s nejmenší hodnotou *MPS*. Systém nastaví *MPS* při konfiguraci tak, že každé zařízení vyše zprávu o svých vlastnostech a *MPS* je nastaven na nejnižší přijatou hodnotu. [1]



Obr. 7: Efektivita přenášených paketů [1]

Hodnota *MPS* také vyjadřuje minimální množství *TLP*, které je potřebné k přenesení dat. S větším *MPS* je pro určité množství dat potřeba méně odeslaných *TLP*, čímž je dosažena vyšší efektivita přeneseného paketu, protože ke každému *TLP* se připojuje režie. Velikost režie se pohybuje okolo 20 bytů a přesnou efektivitu jednoho paketu lze spočítat podle vzorce (1.1). Na obr. 7 je zobrazen průběh efektivity přenášených *TLP*. [1]

$$Efektivita\ TLP = \frac{MPS}{MPS + režie} \times 100\ [%] \quad (1.1) \quad [1]$$

1.6.2 Maximální velikost požadavku pro čtení

Maximální velikost požadavku pro čtení *MRRS* (*Maximum Read Request Size*) vyjadřuje maximální velikost dat, o které může být požádáno. Během konfigurace je softwarem nastaven maximální požadavek pro čtení do každého kontrolního registru v zařízení. Hodnota tohoto parametru se pohybuje od 128 do 4096 B s rozestupy 128 B. *MRRS* není závislé na *MPS*.

MRRS také ovlivňuje výkon PCI Express, protože určuje množství odeslaných požadavků potřebných k načtení dat. *Read Request* transakce představuje 100% režie, protože neobsahují žádná data.

Pro příklad:

Pro čtení 4 kB dat, při *MRRS* 512 B a *MPS* 256 B je potřeba 8 *read request* transakcí a data přijdou v 16 *completion* transakcích o velikosti 256 B dat a 20 B režie.

Pro čtení 1kB dat, při *MRRS* 1kB, *MPS* 256B je vyslán jedna *read request* transakce a data přijdou v čtyřech *TLP*. [1]

1.7 Systém kódování PCI Express

PCI Express generace 1 a generace 2 používá při přenášení informací kódování 8b/10b, generace 3 využívá kódování 128b/130b. Toto kódování se vkládá kvůli synchronizaci přenášených dat. Princip kódování je založen na transformaci 8 bitů na 10 bitů, aby 10 bitový výstup obsahoval stejné množství jedniček a nul (viz. tabulka 2) . Toho nejde dosáhnout vždy, proto se používá i poměr 6 jedniček a 4 nuly nebo 6 nul a 4 jedničky. U kódování 128b/130b je to podobné.

Tento nadbytek dat snižuje efektivitu přenosu o 20% u kódování 8b/10b a 1.538% u kódování 128b/130b. [1]

Teoretickou propustnost při kódování lze vypočítat podle vzorce (1.2)

$$Teor. propustnost = \frac{2.5Gb/s \times 2(2 směry) \times šířka Linky}{10 bit/8 bit} [Gb] \quad (1.2) \quad [1]$$

Pro generaci 1, která dokáže přenášet 2,5 Gb/s s kódováním 8b/10b je efektivní šířka pásma 2,0 Gb/s pro jednu přenosovou linku viz. tabulka 1. [1]

Tabulka 2: ukázka kódování 8b/10b [1]

8 bit	10bit	10 bit negovaných
00000000	1001110100	0110001011
00000001	0111010100	1000101011
00000010	1011010100	0100101011
...	...	...
11111101	1011100001	0100011110
11111110	0111100001	1000011110
11111111	1010110001	0101001110

2 VSTUPNÍ A VÝSTUPNÍ ROZHRAŇÍ

Navrhovaná jednotka bude připojena k zařízení prostřednictvím několika rozhraní. V této kapitole jsou popsána rozhraní, jež budou použita pro navrhovanou jednotku. Navržená jednotka musí dodržet specifikaci i v případě, že nevyužívá některé signály nebo vlastnosti rozhraní. Uvedená rozhraní jsou používána v projektu Liberouter.

2.1 DMA rozhraní

Rozhraní slouží pro větvení do různých koncových jednotek. Každá jednotka připojená na sběrnici:

- Nesmí odesílat zápisové transakce větší než *Maximum Payload Size*.
- Nesmí odesílat požadavky na čtení větší než *Maximum Read Request Size*.
- Musí počítat s tím, že odpověď na čtecí požadavek (*Completion*) může přijít rozdělený do menších paketů. [5]

Tabulka 3: signály rozhraní DMA [5]

Jméno signálu	Šířka [bit]	Směr	Popis
DATA	256	Z→C	data paketu
HDR	96 nebo 32	Z→C	hlavičku paketu
SOP	1	Z→C	start paketu
EOP	1	Z→C	konec paketu
SRC_RDY	1	Z→C	Zdroj připraven. 1 znamená, že všechny zdrojové signály obsahují platná data.
DST_RDY	1	C→Z	Cíl připraven. 1 znamená, že cílová jednotka je připravena na přenos.

V tabulce 3 jsou zobrazeny dva datové signály, jeden pro data a druhý pro hlavičku, a čtyři signály, kterými je řízeno odesílání. V sloupci Směr jsou písmena „Z“ jako zdroj a „C“ jako cíl. Formát a délka hlavičky v signálu HDR se liší pro směr UP (zápisový/čtecí požadavek z FPGA do hostitelského systému) a DOWN (odpověď na čtecí požadavek zpět z hostitelského systému do FPGA). Velikost hlavičky se liší podle typu pro směr DMA_UP je velikost 96 bitů a pro DMA_DOWN je velikost 32 bitů. Signál DATA obsahuje přenášené informace paketu. Signál SOP označuje začátek paketu. Signál EOP označuje konec paketu. Data se přenášejí pouze v případě, že signály SRC_RDY a DST_RDY jsou v logické 1. [5]

Tabulka 4: Formát hlavičky DMA_UP [5]

Bity	Popis
0 až 10	Počet 32bitových slov (DWORD).
11	Typ transakce: 0 = čtení z RAM, 1 = zápis do RAM.
12 až 13	Rezervováno.
14	Relaxed Ordering bit .
15	Rozlišení PCI Express core 0 / 1.
16 až 23	Tag transakce (pro typ transakce = 1 nemá význam).
24 až 31	ID jednotky.
32 až 96	Adresa dat v RAM.

Tabulka 5: Formát hlavičky DMA_DOWN [5]

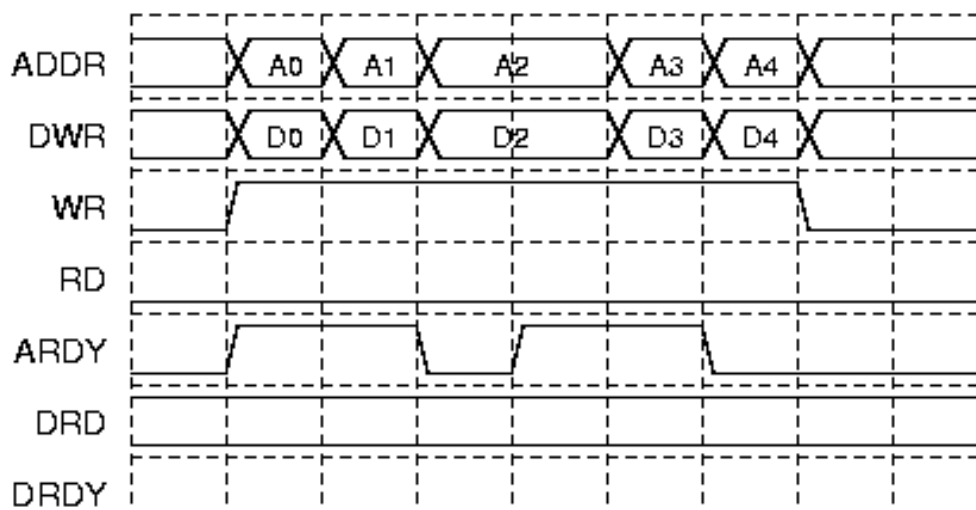
Bity	Popis
0 až 10	Počet 32bitových slov (DWORD).
11	Typ transakce: 0 = čtení z RAM, 1 = zápis do RAM.
12 až 15	Rezervováno.
16 až 23	Tag transakce (pro typ transakce = 1 nemá význam).
24 až 31	ID jednotky.

2.2 MI32 rozhraní(*Memory Interface*)

Rozhraní MI32 slouží k zapisování nebo čtení z paměti pomocí 32 bitů. Chování MI sběrnice je zobrazeno na diagramech obr. 8 a obr. 9.

Tabulka 6: Signály MI rozhraní [4]

Jméno	Šířka [bit]	Typ	Popis
DWR	32	Vstup	Data pro zápis.
ADDR	32	Vstup	Adresa pro čtení/zápis.
RD	1	Vstup	Požadavek na čtení.
WR	1	Vstup	Požadavek na zápis.
BE	4	Vstup	Maska platných bytů signálu DWR.
DRD	32	Výstup	Vyčtená data.
ARDY	1	Výstup	Potvrzení přijetí adresy pro čtení/zápis.
DRDY	1	Výstup	Indikace platnosti vyčtených dat.

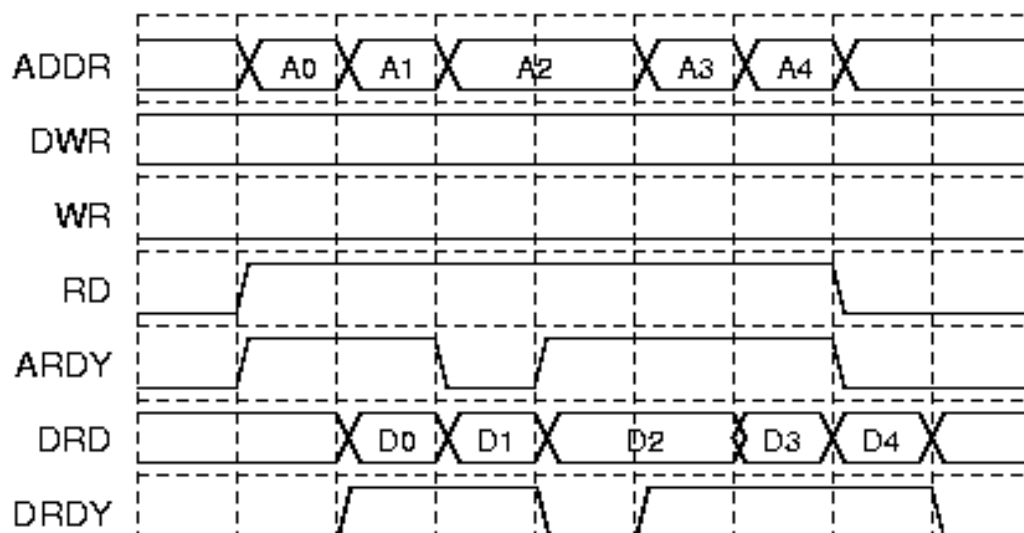


Obr. 8: Zapisování přes rozhraní MI [4]

Při čtecí operaci signál DRDY signalizuje připravenost výstupních dat. Signál ARDY při čtecích nebo zápisových požadavcích slouží k pozastavení vstupního požadavku - přestane se generovat následující adresa do doby, než se ARDY znovu aktivuje. Signály ARDY a DRDY jsou na sobě navzájem nezávislé.

ARDY může být nastaveno jen, když je RD nebo WR aktivní. DRDY je absolutně nezávislé. Počet odeslaných signálu RD musí odpovídat počtu vygenerovaných DRDY signálu. DRDY může být nastaven hned při prvním RD ve stejném cyklu. Port ADDR udává adresy vždy v bytech tzv. při zarovnaném 32 bitovém přístupu jsou vždy spodní dva bity nulové. Naopak při nezarovnaném přístupu jsou spodní dva bity nastaveny na nezarovnanou část adresy.

Při zápisových i čtecích operacích se pro každých 8 bitů dat vyhrazuje jeden BE signál. Tento BE signál vyjadřuje skutečnost, zda se příslušných 8 bitů účastní čtecí nebo zápisové operace. [4]



Obr. 9: Čtení přes rozhraní MI [4]

3 MĚŘENÉ PARAMETRY PCI EXPRESS

PCI Express podporuje různé druhy adresování, přenáší různě velké *TLP* a má vlastní systémové omezení (viz kapitola 1.6). Kvůli nepravidelnému chování nelze přesně určit přenosovou rychlost PCI Express.

Měření bude probíhat na transakční vrstvě při šířkách sběrnice x8 a x16. Při plném vytížení PCI Express 3.0 zapojení dvakrát x8 nebo jedenkrát x16 je rychlost datového toku 128 Gb/s v obou směrech. Cílem měření je zjistit maximální množství čistých dat, které jsme schopni přenášet.

3.1 Propustnost

Měření propustnosti dat se vypočítá jako množství přenesených dat za čas. Měření času začne v okamžiku, kdy protistrana bude schopna přijímat nebo vysílat. V případě, že protistrana nebude schopna přijímat nebo odesílat data, spustí se měření intervalu do okamžiku, než je protistrana opět schopna přijímat nebo vysílat. Tyto intervaly se budou průběžně zaznamenávat a ve výsledku se odečtou od celkové doby přenášení dat.

3.2 Latence

Latenci je možné měřit pouze u čtecí transakce *read request*, proto se při odeslání tohoto požadavku zapíše čas odeslání. Při přijetí *completion* transakce se stejným identifikačními údaji, jako byl požadavek na čtení, je zapsán čas přijetí.

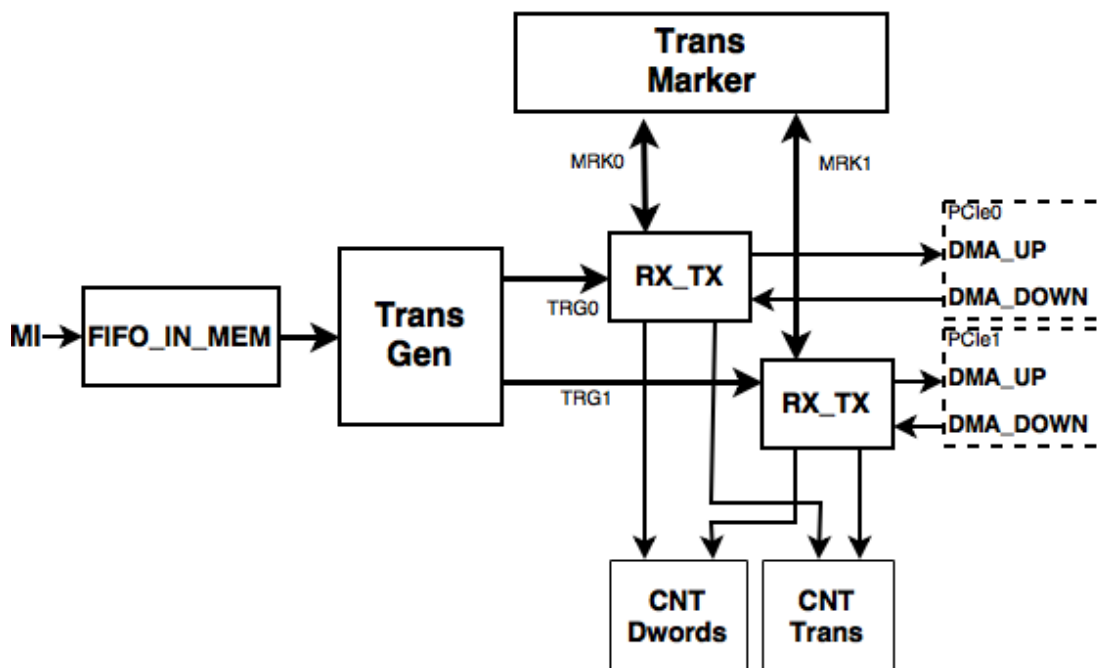
Přesnost měření času souvisí s rychlostí čítačů respektive jejich pracovní frekvenci. V použitém obvodu FPGA se maximální doba jednoho taktu je 5 ns a průměrná hodnota odezvy na PCI Express se pohybuje okolo 400 ns. Kvůli očekávané odezvě, která je o 2 řády větší než maximální doba jednoho taktu, není kritická přesnost měření.

3.3 Doba neschopnosti přijímání informací

Při odesílání *TLP* je na *DMA* vždy kontrolována dostupnost místa, na které jsou *TLP* doručeny. Při neschopnosti přijímání informací dochází k zastavení provozu na celé lince. Toto zdržení má za následek snížení propustností celého systému. Počítáním taktů neschopnosti přijímání informací dostaneme dobu, kdy jsme byli schopni vysílat ale na datové lince PCIe došlo ke zdržení.

4 NÁVRH KOMPONENTY PRO MĚŘENÍ PARAMETRŮ KOMUNIKACE PŘES PCI EXPRES

Návrh komponenty se zaměřuje na generování transakcí a sběr informací nad provozem PCI Express. Komponenta by měla být schopná generovat provoz až 128 Gb/s a tento provoz by také měla být schopna zachytit. Jednotka bude konfigurovatelná ze software skrze rozhraní MI32 a ke koncovému zařízení PCI Express bude připojena prostřednictvím rozhraní *DMA_UP* a *DMA_DOWN*.



Obr. 10: Schéma jednotky pro měření parametrů komunikace přes PCI Express

Na Obr. 7 je zobrazeno blokové schéma navržené komponenty. Komponenta obsahuje tři hlavní komponenty **TRANS_GEN**, **RX_TX**, **TRANS_MARKER**, dva čítače a paměť **FIFO_IN_MEM**. Blokové schéma obsahuje i rozhraními pro připojení k jednotlivým koncovým zařízením PCI Express a pro přístup ze software skrze rozhraní MI32. Funkce jednotky začíná na MI32 rozhraní, přes které jsou nahraná konfigurace do vstupní paměti **FIFO_IN_MEM**. Podle této konfigurace je nastaven generátor, který poté vysílá jednotlivé hlavičky *DMA* paketů na odesílací část komponenty **RX_TX**. Detailnější popis jednotlivých částí je popsán v dalších podkapitolách.

4.1 Komponenta FIFO_IN_MEM

FIFO_IN_MEM je paměť typu FIFO a obsahuje dva typy parametrů. Jeden pro řízení generátoru a druhý pro řízení odesílání paketu. Přehled parametrů je uveden v tabulce 7.

Tabulka 7: Parametry generátoru TRANS_GEN

Jméno	Velikost [bit]	Popis
tran_num	16	Počet transakcí k odeslání.
adr_inc	10	Počet adres o které se má inkrementovat.
adr	64	Adresa v RAM paměti.
traffic	7	Nastavení provozu.
tag	8	Tag transakce.
type	1	typ transakce pro zápis/čtení
dwords	11	Počet dvojslov (dwords) transakce.
send_delay	7	Nastavuje počet taktů pro vytvoření mezery mezi pakety
tx_mode	6	Určuje vlastnost každého odeslaného paketu

Paramet **tran_num** slouží pro nastavení množství paketu, které mají být odeslané. **Ard_inc** nastavuje počet adres, o které se má generátor inkrementovat. **Adr** slouží k nastavení adresy v RAM paměti ,na které má paket uložit svoje data nebo odkud bude data vyčítat. Zápis a četní paměti je řízeno parametrem **type**, který nabývá hodnoty logické 1 pro zapisování do RAM paměti a logická 0 pro čtení z RAM podle DMA protokolu viz. kapitola 2. Poslední parametr v hlavičce paketu je **dwords**, který udává množství dvojslov v paketu. Další parametr **traffic** slouží pro tvorbu provozu jiného než 100%. Parametr je **send_delay**, který pracuje v komponentě **TRANS_MARKER**. Tento parametr udává počet taktů mezi odesláním paketů. Poslední parametr je **tx_mode** obsahuje informace pro řízení paketu.

4.2 Komponenta TRANS_GEN

Jednotka **TRANS_GEN** vyvádí hlavičku *DMA* paketu z konfigurace v **IN_FIFO_MEM** a doplňuje ji o parametr **id**, který není uložený v paměti, protože je definován pro každé zařízení. Data získaná z paměti jsou naformátována do *DMA* formátu a odeslána na dostupné rozhraní jednotky **RX_TX**. Komponenta **TRANS_GEN** má dvě výstupní rozhraní pro **RX_TX** komponenty, kterým může zaslat vygenerované pakety. Nastavení generátoru zahrnuje možnosti inkrementaci adresy, počet *DMA* paketů, které mají být odeslány v této konfiguraci a náhodnou velikost *DMA* paketů. Pro *DMA* sběrnici je nutný i **tag**, který je unikátní pro každý odeslaný paket. **TRANS_GEN** generuje **tag** pro každý *DMA* paket.

4.3 Komponenta RX_TX

Komponenta **RX_TX** se dělí na 2 části, jedna slouží pro odesílání přes rozhraní DMA_UP a druhá slouží pro přijímání odpovědí z DMA_DOWN. Další funkce **RX_TX** komponenty je odesílání informací o množství paketu a dvojslov, jenž byly odeslány nebo přijaty.

Funkce **RX_TX** souvisí s měřením neschopností odesílání paketu viz kapitola 3.3. Pro toto měření bude jeden čítač, kterému po odeslání začátku paketu bude umožněno kontrolovat dostupnost cílového zařízení. Při každém nedostupném taktu bude čítač inkrementován o jedna a po skončení testování bude možno přičíst si celkovou dobu neschopnosti přijímání cílové oblasti.

Další vlastností odesílání je úprava celkového provozu pomocí *LFSR* registru. Parametr **traffic** nastavuje mezní hodnotu při které má komponenta odesílat. Při hodnotě *LFSR* registru větší než parametru **traffic** se pozastaví odesílání. Chování parametru **traffic** neovlivňuje dobu nedostupnosti, protože při počítání doby neschopnosti přijímání se porovnává i povolením pro odesílání.

4.4 Komponenta TRANS_MARKER

Funkcí komponenty **TRANS_MARKER** je výpočet latence z přijatých odpovědí. **TRANS_MARKER** obsahuje paměť a čítač času. Při odeslání monitorovaného paketu jsou do paměti zapsány identifikační údaje a čas odeslání. Naopak při přijetí transakce je hodnota **tagu** porovnána s pamětí a posléze vypočítána latence. Další funkce je pozdržení odesílání. Pomocí parametru **send_delay** jsme schopni vytvořit časovou mezeru mezi odeslaným paketem a dalším následujícím paketem připraveným pro odeslání. Komponenta **TRANS_MARKER** pracuje pro oba PCIE koncové body.

5 IMPLEMENTACE

Tato kapitola se zabývá realizací navržené jednotky a podrobněji popisuje problémy, které nastaly při implementaci. Podkapitoly 5.1 a 5.2 se věnují hlavním představám implementace. Podkapitola 5.3 se věnuje vzniklým problémům při implementaci. V podkapitole 5.3 jsou popsány možnosti řízení komponenty. V podkapitolách 5.5, 5.6 a 5.7 je rozkreslený detailnější popis jednotlivých komponent. Na obrázcích obr. 12, obr. 13, obr. 14 není implementované zapojení, ale pouze způsob jakým si komponenty předávají informace. Uvedené obrázky slouží pouze pro obecnou představu funkce jednotlivých parametrů. V podkapitole 5.8 je stručný souhrn využitých zdrojů FPGA čipu.

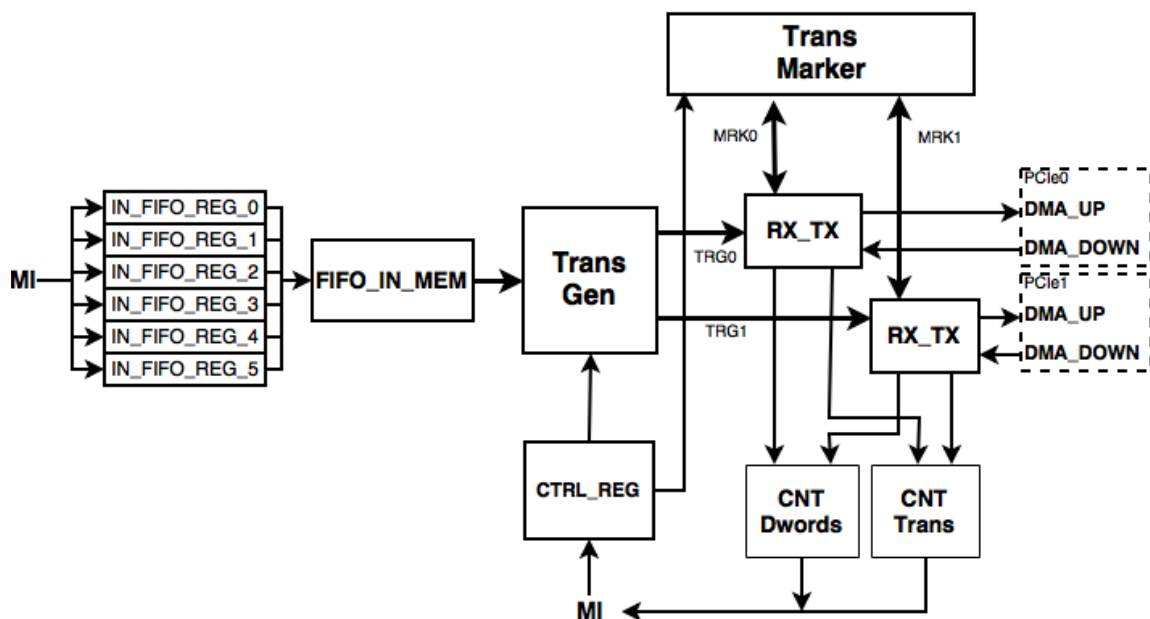
5.1 Principy použité při implementaci

Hlavní principy použité při implementaci jednotky se zaměřovaly na využívání registrů se snahou potlačit používání stavových automatů. Všechny operace, které na sobě přímo nejsou závislé, byly implementovány paralelně a jejich vstupy a výstupy jsou řízeny stejnými hodnotami signálů. Dalším důvodem využití paralelního zpracování je doba, kterou je nutné pro vytvoření hlavičky. Pro generování provozu i na nejkratších DMA paketech je doba odesílání pouze jeden takt. Tato doba odesílání omezuje dobu generování hlavičky na maximálně jeden takt.

5.2 Popis implementace

Implementace této jednotky vychází z návrhu jednotky pro měření parametrů komunikace přes PCI Express, viz kapitola 4. Na obr. 11 je vidět, že oproti návrhu přibýly vstupní registry pro zápis do paměti **FIFO_IN_MEM** a řídicí registr **CTRL_REG**, který je popsán v kapitole 5.4. Šířka datové sběrnice u paměti **FIFO_IN_MEM** je 192 bitů, proto je potřeba 6 registrů o velikosti 32 bitu pro synchronní zápis do **FIFO_IN_MEM**. Velikost registrů vychází z rozhraní MI32. Řídicí registr, byl přidán kvůli chybějící možnosti ovládání jednotky skrze software. Výhoda řízení skrze rozhraní MI32 je větší schopnost kontroly jednotky. Vlastnosti řídicího registru jsou popsány v kapitole 5.4.

Další vhodnou cestou při implementaci jednotky byla možnost se řídit velikostí datových signálů DWR a DRD u rozhraní MI32 a přizpůsobit jim velikost veškerých možných čítačů a registrů. Jedinou výjimkou jsou čítače přijatých nebo odeslaných dvojslov, kvůli jejich možnému přetečení. Velikost čítačů a registrů pro zapisování do **FIFO_IN_MEM** je tedy 32 bitů a velikost čítačů dvojslov je 64 bitů.



Obr. 11: Schéma implementované jednotky pro měření parametrů komunikace přes PCI Express

5.3 Problémy vzniklé při implementaci

Během implementace bylo zjištěno, že navržená jednotka vkládá mezi odesílané pakety nadbytečné mezery. Vzniklá mezera snižovala propustnost a pro pakety menší než 32 dvojslov nebylo možné dosáhnout propustnosti 100 Gbit/s. Tato chyba byla způsobena zbytečným počtem registrů v komponentě **RX_TX**, který vedl ke zpožděným reakcím obvodu. Opravení chyby znamenalo předělání rozhraní mezi komponentami **TRANS_GEN** a **RX_TX**. Také byl pozměněn způsob generování DMA paketů tak, aby reagoval na současný stav řídicích signálů.

5.4 Implementované řízení jednotky

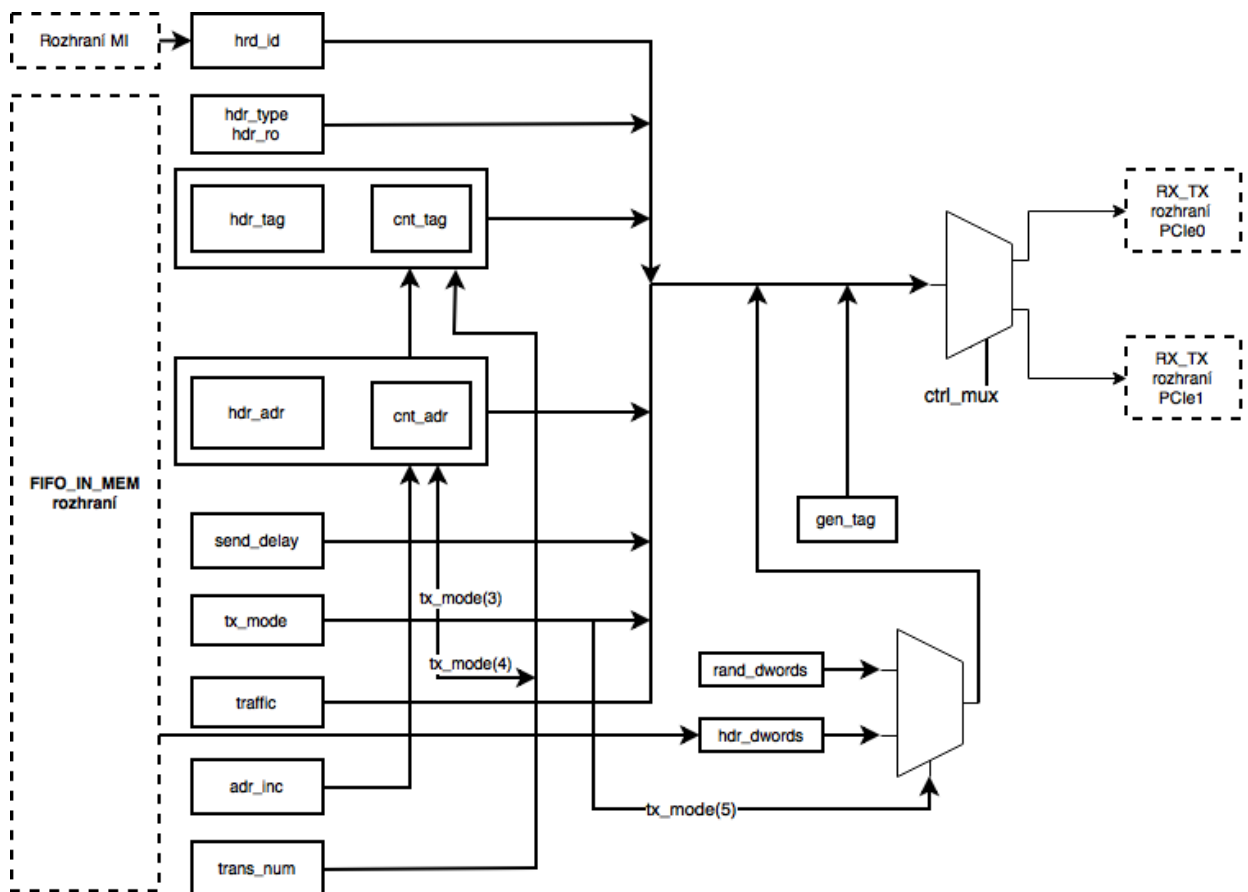
Schopnost kontrolovat chování jednotky představuje řízení přijímání a vysílání informací. U implementované jednotky je možnost nastavovat odesílání pouze na jeden PCI Express koncový bod, zapínání generátoru **TRANS_GEN**, zapínání a resetování čítače času v komponentě **TRANS_MARKER** a označení PCI Express koncového bodu, na který byla transakce odeslána. Tyto informace jsou uloženy v 14bitovém registru **CTRL_REG**. Do **CTRL_REG** je možné zapisovat přes rozhraní MI32.

Řízení komponenty se skládá také z signálu **tx_mode**. Tento 6 bitový signál určuje použití jednotlivých parametru jako **trans_num**, **adr_inc**, **traffic**. Bity jsou označeny od spodního 0 až po horní po číslo 5. **Tx_mode(0)** určuje zda má být zapsán **tag** a čas odeslání paketu. **Tx_mode(1)** zapíná funkci zpoždění odesílání jehož velikost určena parametrem **send_delay**. **Tx_mode(2)** spouští regulaci provozu tj. vytváření provozu jiného než 100%, určeného parametrem **traffic**. **Tx_mode(3)** zapíná inkrementaci odesílané adresy, při logické 1 je adresa každého odeslaného paketu inkrementována o 1. **Tx_mode(4)** zapíná odesílání více paketů v nastavení, které se

skládá z ostatních parametrů. Počet odeslaných paketů je určen parametrem **trans_num**. **Tx_mode(5)** zapíná náhodnou velikost paketů. Logická 1 u **tx_mode(5)** spouští náhodné generování dvojslov pomocí LFSR registru a hodnota LFSR registru se vkládá jako velikost paketu místo parametru **dwords**. Každý bit funguje jako značka a při jeho logické 1 je zapnuta uvedená funkce, která spouští použítá řízených parametru.

5.5 Komponenta TRANS_GEN

Na obr. 12 je schematicky naznačen tok informací, proudící přes jednotlivé registry. Jak je naznačeno na schématu informace, tak jsou přijímány z paměti **FIFO_IN_MEM** a z rozhraní MI. Rozhraní MI v této komponentě slouží k zapsání parametru **id**, který slouží pro identifikaci jednotlivých zařízení v systému PCI Express. Parametry s předponou **hdr** slouží pro vytvoření hlavičky DMA paketu. Z přijatých informací je seskládána hlavička a společně s konfiguračními údaji pro komponentu **RX_TX** je odeslána na dostupné rozhraní jedné z komponent **RX_TX**. Dostupnost komponent **RX_TX** je dána volným místem na výstupním registru komponenty **TRANS_GEN**. Pokud je volné místo ve výstupním registru na rozhraní **RX_TX**, je multiplexor přepojen signálem **ctrl_mux** na tuto pozici a je okamžitě zaplněn. Informace v těchto výstupních registrech čekají, dokud si je nepřevzme komponenty **RX_TX**. Komponenta **TRANS_GEN** generuje přijatou hlavičku za dvou předpokladů. První předpoklad je, že počet transakcí, řízen parametrem **tran_num** je větší než 0. Druhý předpoklad je, že počet adres, o které se má hlavička inkrementovat je větší než 0. V případě splnění jedné z těchto podmínek je z komponenty generována hlavička ostatním nastavením pro další komponenty.



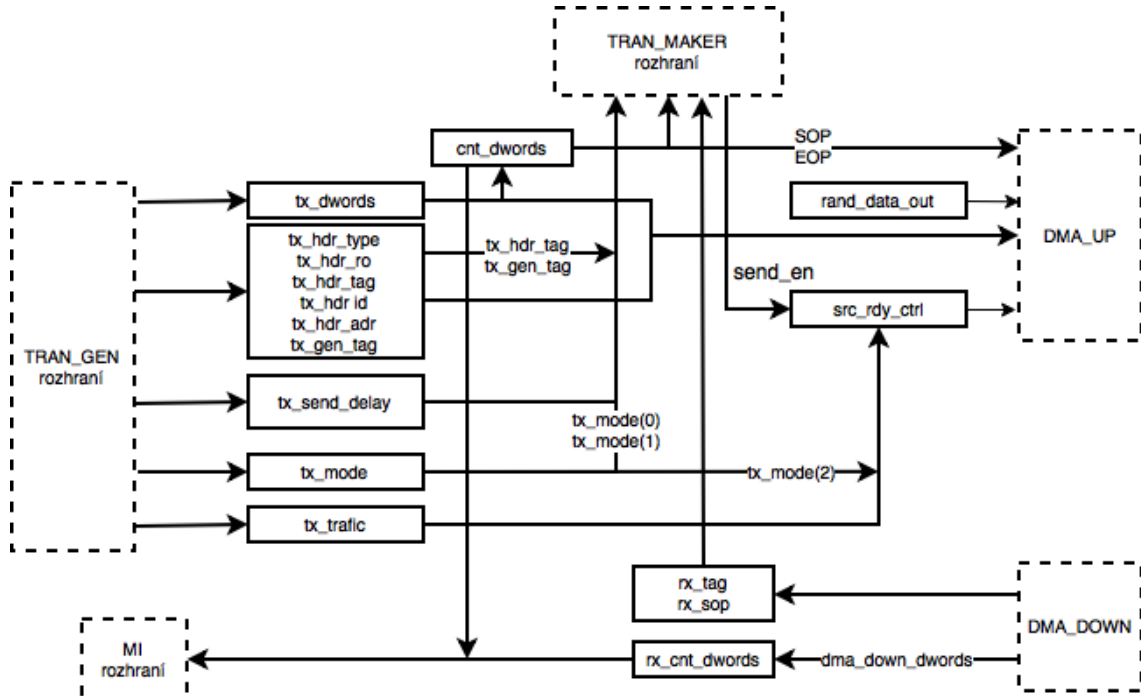
Obr. 12: Schéma proudění informací přes TRANS_GEN

5.6 Komponenta RX_TX

Komponenta **RX_TX** slouží převážně k přijímání a vysílání DMA paketů. Na obr. 13 je schématicky popsán tok informací přes komponentu **RX_TX**. Komponenta odesílá vygenerované hlavičku pakety v případě, že je dostupné na výstupním rozhraní **TRANS_GEN**. K hlavičce připojuje náhodně vygenerovaná data v množství definované parametrem **tx_dwords**. Takto vytvořený DMA paket je odeslán na rozhraní **DMA_UP** a zároveň jsou odeslány identifikační údaje do komponenty **TRANS_MARKER**. Komponenta **RX_TX** přeposílá parametr **send_delay** s parametrem **tx_mode(0)** a **tx_mode(1)** souběžně s identifikačními údaji. V komponentě **RX_TX** je odesílání je řízeno parametry **traffic** a **send_delay**. Pokud je parametr **traffic** aktivní, tak povoluje odesílání v případě, že hodnota LFSR registru je menší než zadaný parametr **traffic**. Pokud je aktivní parametr **send_delay**, je jeho hodnota odeslána do komponenty **TRANS_MARKER** a čeká se na odpověď z komponenty **TRANS_MARKER**. Po dobu čekání komponenta **RX_TX** neodesílá pakety. Pokud ani jeden z parametru **traffic** nebo **send_delay** není aktivní tak odesílání paketů z komponenty **RX_TX** je omezeno pouze přijímací stranou, na rozhraní **DMA_UP**.

Druhá část komponenty **RX_TX** řeší přijímání DMA paketů z rozhraní **DMA_DOWN**. Jednotka nepřijímá data pouze jejich identifikační údaje a přeposílá je k vyhodnocení do komponenty **TRANS_MARKER**.

V Komponentě **RX_TX** jsou čítače odeslaných a přijatých dvojslov v čítačích **cnt_dwords** a **rx_cnt_dwords**. Do těchto čítačů je umožněn přístup přes rozhraní **MI**.

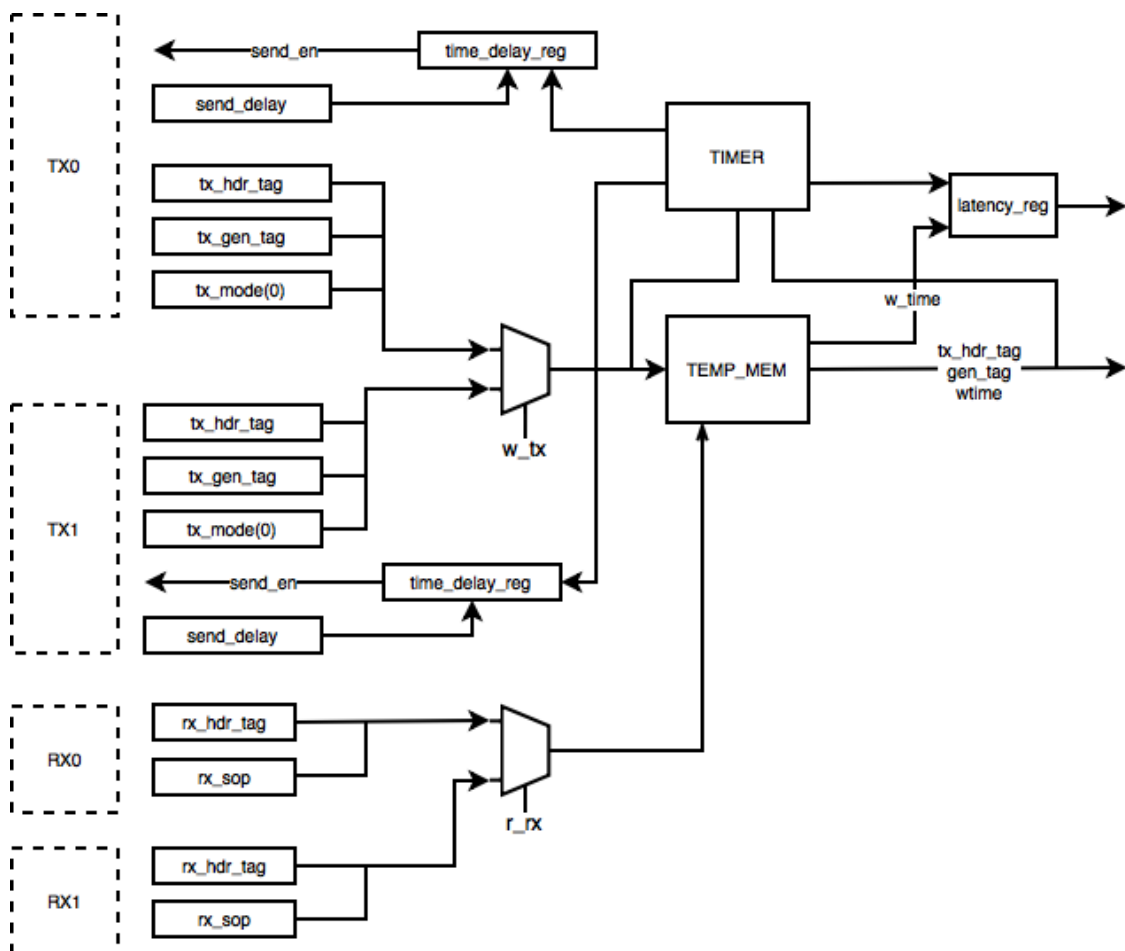


Obr. 13: Schéma proudění informací v komponentě **RX_TX**

5.7 Komponenta TRANS_MARKER

Rozhraní komponenty **TRANS_MARKER** je rozděleno na RX a TX část pro každý PCI E koncový bod. Komponenta **TRANS_MARKER** přijímá identifikační údaje odesílaných DMA paketů na dva PCI Express koncové body. Ukládá si jejich hodnotu v případě, že jsou označeny parametrem **tx_mode(0)** logickou 1. Pokud je na TX rozhraní doručena aktivní parametr **send_delay** je okamžitě vypnut signál **send_en**. Hodnota parametru je zapsána a sečtena s aktuálním časem a uložena. Signál **send_en** je opět nastaven na logickou 1 po uplynutí času. Tato funkce je rozdělena pro každý PCI Express koncový bod.

Hlavní funkce komponenty **TRANS_MARKER** je výpočet latence. V případě, že parametr **tx_mode(0)** je v logické 1, tak se zapíší do paměti **TEMP_MEM** identifikační údaje a čas odeslání DMA paketu. Každá hodnota přijatá z RX rozhraní je pak porovnávána. Pokud se najde shoda s uloženými identifikačními údaji, tak je současná hodnota čítače času **TIMER** a uložená hodnota času odeslána do registru **LATENCY_REG**, kde je vypočítán rozdíl mezi časy.



Obr. 14: Schéma proudění informací v TRANS_MARKER

5.8 Spotřeba zdrojů implementované jednotky

Výsledná implementace jednotky spotřebovává množství zdrojů uvedené v tabulce 8. a v tabulce 9. Splňuje časování s rezervou 1,087 ns pro jeden PCI Express x8 a 1.117 ns rezervou pro dva PCI Express koncové body při frekvenci 200 MHz. Použitý čip FPGA je Virtex 7 s rychlostní kategorií -2. Přesné označení použitého čipu je xc7vh580thcg1931. Další dohledatelné parametry čipu FPGA lze najít pod označením h580t na odkaze [6].

Tabulka 8: Využívané zdroje při konfiguraci pro jeden PCI Express x8

TYP ZDROJE	POČET	Využití FPGA [%]
LUT	474	0.15
klopné obvody	1003	0.13
BRAM paměti	4	0.43

Tabulka 9: Využívané zdroje při konfiguraci pro dva PCI Express x8

TYP ZDROJE	POČET	Využití FPGA [%]
LUT	569	0.16
klopné obvody	1373	0.19
BRAM paměti	4	0.43

6 ZÁVĚR

Implementovaná jednotka je ve fázi ladění chyb, z čehož vyplívá, že není 100% funkční. Chyba, která přetrvává je nesprávné nastavování signálu SOP, EOP při odesílání DMA paketů z komponenty **RX_TX**. Tato chyba vznikla po opravě chyby popsané v kapitole 5.3. Chybu se nepodařilo odstranit do termínu odevzdání BP, oprava bude pokračovat.

V implementované jednotce je plně funkční generátor hlaviček řiditelný skrze rozhraní MI32 a jeho výstupy je možné odesílat pro 2 koncové zařízení. Kvůli rozesílání mezi 2 PCI Express koncové body při krátkých paketech (menších než 8 dvojslov) nelze dosáhnout propustnosti 100 GB/s. Jednotka je generická ve vztahu k velikostem parametrů a lze nastavit její použití pouze pro jeden nebo dva PCI Express koncové body. Jednotka je řiditelná ze software pomocí rozhraní MI32 a podporuje typy adresování RAM paměti pro 32 a 64bitů. Po dokončení současných návrhů se plánuje rozšíření funkcí o možnost tvorby histogramu a vytváření průběžného měření pomocí přídavné paměti typu *FIFO*.

V tabulce 8 a tabulce 9 jsou zobrazeny využívané zdroje a celkové využití čipu FPGA Virtex 7. Implementovaná jednotka má při konfiguraci pro dva PCI Express koncové body má o 36.9 % větší využití klopných obvodů a o 20 % více spotřebovaných LUT jednotek než konfigurace pro jeden PCI Express koncový bod

Využívání navržené jednotky umožňuje měřit množství přenesených dat, latency a vytvářet záznamy chování paketů procházející přes PCI Express sběrnici i bez použití komponenty drahých zařízení jako je třeba PCI Express analyzátor. Použití komponenty se plánuje pro měření parametrů komunikace na akceleračních kartách COMBO-100G s čipem FPGA Virtex 7.

SEZNAM LITERATURY

- [1] LAWLEY, Jason. *Understanding Performance of PCI Express Systems* . October 28, 2014.
PCI Express Base Specification Revision 3.0 [online]. 2010 [cit. 2016-06-2]. Dostupné z:
[2] http://composter.com.ua/documents/PCI_Express_Base_Specification_Revision_3.0.pdf
- [3] BUDRUK, Ravi. *PCI express system architecture*. Boston: Addison-Wesley, c2004. ISBN 0321156307.
- [4] MATOUŠEK, Jiří. *IMPLEMENTACE A VERIFIKACE VSTUPNÍCH A VÝSTUPNÍCH SÍŤOVÝCH BLOKU*. Brno, 2009. Bakalářská Práce. FIT VUT. Vedoucí práce Ing. Jiří Tobola.
- [5] Liberouter. *Návrh firmwaru - NetCOPE pro kartu Combo-CG*. 2013. [Neveřejné].
- [6] *7 Series FPGAs Overview* [online]. 2015 [cit. 2016-06-02]. Dostupné z:
http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf
- [7] Cards. Liberouter [online]. [cit. 2016-06-02]. Dostupné z:
<https://www.liberouter.org/technolgies/cards/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

<i>CRC</i>	Cyklicky redundantní součet (<i>Cyclic redundancy check</i>)
<i>DLLP</i>	Paket linkové vrstvy (<i>Data Link Layer Packet</i>)
<i>DMA</i>	Přímý přístup do paměti (<i>Direct memory acces</i>)
<i>DWORD</i>	Dvojslovo, jednotka propustnosti o velikost 32 bitů
<i>EOP</i>	Konec paketu (<i>End of Packet</i>)
<i>FC</i>	Řízení toku (<i>Flow Control</i>)
<i>FIFO</i>	Paměť jejíž vstupní informace jsou odesílány na výstup ve stejném pořadí jak přicházejí. (<i>First In First Out</i>)
<i>FPGA</i>	Programovatelné hradlové pole (<i>Field Programable Gate Array</i>).
<i>LFSR</i>	Posuvný registr s lineární zpětnou vazbou (<i>Linear-feedback shift register</i>)
<i>MI</i>	Pamětové rozhraní (<i>Memory Interface</i>)
<i>MPS</i>	Maximální velikost obsahu paketu (<i>Maximum Payload Size</i>)
<i>MRRS</i>	Maximální velikost požadavku pro čtení (<i>Maximum Read Request Size</i>)
<i>PCI</i>	Periférní připojení komponent (<i>Peripheral Component Interconnect</i>)
<i>PLP</i>	Paket fyzické vrstvy (<i>Physical Layer Packet</i>)
<i>RAM</i>	Náhodný přístup paměti (<i>Random Access Memory</i>)
<i>SOP</i>	Začátek paketu (<i>Start of Packet</i>)
<i>TLP</i>	Paket transakční vrstvy (<i>Transaction Layer Packet</i>)