



Fakulta elektrotechnická
Katedra elektroniky a informačních technologií

DIPLOMOVÁ PRÁCE

Využití rozhraní na bázi PCI-Express pro akviziční systémy s velkou datovou propustností

Autor práce: Bc. Martin Farkaš
Vedoucí práce: Ing. Michael Holík, Ph.D.

Plzeň 2021

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Martin FARKAŠ**
Osobní číslo: **E19N0055P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a aplikovaná informatika**
Téma práce: **Využití rozhraní na bázi PCI-Express pro akviziční systémy s velkou datovou propustností**
Zadávací katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

Předmětem diplomové práce je vývoj modulárního akvizičního rozhraní s vysokou propustností datového toku, s využitím moderních vysokorychlostních rozhraní.

1. Seznamte se s rozhraním PCI-Express a prostudujte jeho specifické vlastnosti
2. Porovnejte PCI-Express s ostatními rozhraními s vysokou datovou propustností, dostupnými na běžných osobních počítačích, které se jeví být vhodné pro nasazení v akvizičních systémech (například 1/10 Gbit ethernet, USB 3.x)
3. Zvolte vhodnou platformu pro realizaci testovací implementace (FPGA vývojový kit umožňující přímé připojení na pozici PCI-Express sběrnice osobního počítače. Vývojové prostředí FPGA firmwaru, které nabízí dostupné IP jádro PCI-Express k uživatelské implementaci)
4. Proveďte implementaci rozhraní PCI-Express na zvolené platformě a zprovozněte přenos dat mezi osobním počítačem a vývojovým kitem. Při vytváření implementace zohledněte specifický profil akvizičního systému (kontinuální sběr dat, optimalizace pro maximální propustnost a zejména pak pro asymetrický přenos kdy velikost přijímaného datového toku výrazně převyšuje velikost odesílaného datového toku). Dále při implementaci uvažujte možnost budoucí integrace do reálného akvizičního systému uzpůsobením designu a volbou vhodných rozhraní.
5. Proveďte testování na realizované implementaci, analyzujte a zhodnoťte dosažené výsledky.
6. Diskutujte postup pro budoucí integraci v reálném akvizičním systému.

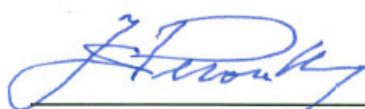
Rozsah diplomové práce: **40 – 60 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. J. Broulim et al 2019, „j-Pix ? A multiplatform acquisition package for Timapix3“, JINST 14 C06004

Vedoucí diplomové práce: **Ing. Michael Holík, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání diplomové práce: **9. října 2020**
Termín odevzdání diplomové práce: **27. května 2021**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Doc. Ing. Jiří Hammerbauer, Ph.D.
vedoucí katedry

V Plzni dne 9. října 2020

Abstrakt

Tato práce se věnuje přenosu dat mezi PC a FPGA za pomoci PCI Express (PCIe) rozhraní. V teoretické části diplomové práce je popsán přehled dnes běžně dostupných rozhraní pro přenos dat na straně PC. Dále je detailněji popsáno rozhraní PCIe. Druhá část diplomové práce se věnuje implementaci PCIe rozhraní v FPGA. Realizovaný systém je kombinací dostupných IP jader a vlastního kódu, který je napsán v jazyce VHDL. PC aplikace je napsána v jazyce C++. Na realizovaném systému je ukázán přenos dat mezi pamětí a PC. Maximální dosažená propustnost z FPGA do PC byla 1646 MB/s a z PC do FPGA byla maximální dosažená propustnost 1257 MB/s.

Klíčová slova

PCI Express, USB, ThunderBolt, FireWire, PolarFire, Microsemi, Libero SoC, Propustnost.

Abstract

Farkaš, Martin. *Application of the PCI-Express based interface in acquisition systems operating at high data transfer rate [Využití rozhraní na bázi PCI-Express pro akviziční systémy s velkou datovou propustností]*. Pilsen, 2021. Master thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Electronics and Information Technology. Supervisor: Michael Holík

This thesis deals with data transfer between PC and FPGA device using PCI Express (PCIe) interface. The theoretical part of the thesis describes an overview of commonly available PC interfaces for data transfers. The next chapter describes the PCIe interface in more detail. The second part of this thesis is devoted to the implementation of the PCIe interface in the FPGA kit. The implemented system is a combination of available IP cores and custom code, which is written in VHDL language. The PC application is written in C++ language. The implemented system demonstrates data transfer between FPGA memory and PC. The maximum achieved throughput from FPGA to PC was 1646 MB/s and from PC to FPGA the maximum achieved throughput was 1257 MB/s.

Keywords

PCI Express, USB, ThunderBolt, FireWire, PolarFire, Microsemi, Libero SoC, Throughput.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

V Plzni dne May 27, 2021

Bc. Martin Farkaš

.....

Podpis

Contents

Seznam obrázků	vii
Seznam tabulek	viii
Seznam symbolů a zkratk	ix
1 Úvod	1
2 Rozhraní PC s vysokou datovou propustností	2
2.1 Ethernet	2
2.1.1 Fast Ethernet	3
2.1.2 Gigabit Ethernet	3
2.1.3 10 Gigabit Ethernet	3
2.1.4 Nejnovější specifikace	4
2.1.5 Power over Ethernet	4
2.2 USB	4
2.2.1 USB 1.x a USB 2.0	5
2.2.2 USB 3	5
2.2.3 USB 3.1	6
2.2.4 USB 3.2	6
2.3 FireWire - IEEE 1394	6
2.3.1 Specifikace pro FireWire	7
2.3.1.1 FireWire 400	7
2.3.1.2 FireWire 800	8
2.3.1.3 FireWire S1600 a FireWire S3200	8
2.3.2 Konec FireWire	8
2.4 Thunderbolt	8
2.4.1 Thunderbolt generace	9
2.4.2 Thunderbolt 2	9
2.4.3 Thunderbolt 3	10
2.4.4 Thunderbolt 4	10
2.4.5 Šířka pásma pro data	10

3	PCI Express	11
3.1	Sériový přenos	11
3.1.1	Problém paralelního přenosu PCI	12
3.1.2	Překonání problémů paralelního přenosu	12
3.2	Přenos signálu v PCIe	13
3.2.1	Diferenciální signalizace	13
3.2.2	Paketový protokol	13
3.3	Topologie PCIe	14
3.3.1	Root Complex	14
3.3.2	Switch	15
3.3.2.1	ACS	15
3.3.3	Koncové zařízení	16
3.4	Přenosové vrstvy PCIe	16
3.4.1	Transakční vrstva	17
3.4.1.1	Quality of Service	17
3.4.1.2	Flow Control	18
3.4.2	Linková vrstva	18
3.4.2.1	Ack/Nak protokol	19
3.4.3	Fyzická vrstva	19
3.4.3.1	8b/10b kódování	19
3.4.3.2	Logická část	20
3.4.3.3	Elektrická část	20
3.4.4	Bázová adresa	20
3.5	Propustnost sběrnice PCIe	21
3.5.1	Výpočet propustnosti	21
3.5.1.1	Vliv MPS na přenos	23
3.5.1.2	Odhad reálné propustnosti systému	25
4	Realizace	27
4.1	Platforma pro realizaci	27
4.2	PCIe systém	29
4.2.1	Advanced eXtensible Interface	29
4.2.2	PCIe jádro	30
4.3	HW implementace	31
4.3.1	PCIe Endpoint subsystém	31
4.3.2	Reset a hodinové signály systému	32
4.3.3	Připojení bloků k PCIe jádru	33
4.3.4	Vnitřní registry FPGA	34
4.4	SW realizace	34
4.4.1	Device info	35
4.4.2	Zápis/čtení do BAR	36

4.4.3	Zápis/čtení pomocí DMA	36
4.5	Integrace do akvizičního systému	37
5	Naměřené výsledky	38
5.1	Postup měření	38
5.1.1	Naměřené hodnoty	39
5.2	Vyhodnocení výsledků	41
6	Závěr	42
	Reference, použitá literatura	44
	Přílohy	46
A	Připojení vývojového kitu do systému	46

List of Figures

2.1	Jednotlivé vrstvy modelu TCP/IP a referenčního modelu ISO/OSI [Převzat z [13]]	3
2.2	Naznačení vodičů v kabelu pro USB 1.x a USB 2.0 [Převzat z [12]]	5
2.3	Schéma kabelu pro USB 3.x [Převzat z [12]]	6
2.4	Princip funkce Thunderbolt rozhraní [Převzat z [11]]	9
2.5	Princip vedení pro Thunderbolt první generace [Převzat z [11]]	9
3.1	Problémy a omezení paralelního přenosu PCI [Převzat z [1]]	12
3.2	Jeden přenosový pár PCIe linky	13
3.3	PCIe topologie [Převzat z [2]]	14
3.4	Blokové schéma PCIe přepínače [Převzat z [5]]	15
3.5	Vrstvový model PCIe systému	16
3.6	Transakční paket (TLP) pro PCIe Gen2 [Převzat z [6]]	17
3.7	Struktura DLLP bez začátku a konce rámce [Převzat z [5]]	18
3.8	Závislost efektivity přenášeného paketu na jeho MPS	24
4.1	Komponenty na vývojové desce Everest DEV Board [Převzat z [14]]	28
4.2	Blokové schéma realizovaného systému	29
4.3	Architektura PolarFire PCIe jádra	30
4.4	Zapojení PCIe Endpoint subsystému	32
4.5	PCIe TL CLK subsystém	32
4.6	Propojení AXI slave bloků s PCIe přes AXI4Interconnect	33
4.7	IO Cntrl blok	34
4.8	Úvodní obrazovka konzolové aplikace	35
4.9	Výpis základních informací o zařízení	35
4.10	Zápis dat do LSRAM (vlevo) a následné vyčtení dat ze stejného místa v LSRAM (vpravo)	36
4.11	Možnosti DMA přenosů PC aplikace	37
5.1	Ukázka měření propustnosti (LSRAM 64 kB DMA přenos, 250 MHz AXI CLK)	39
A.1	Připojení kitu do PC systému pomocí prodlužovacího kabelu	46

List of Tables

3.1	Teoretická propustnost PCIe pro jednotlivé generace a pro různé počty linek (jednosměrný přenos).	21
3.2	Přenosová rychlost a typ kódování pro jednotlivé generace	21
3.3	Teoretická datová propustnost pro PCIe Gen2 (MPS = 128 B, RCB = 64 B)	25
5.1	Naměřené hodnoty propustnosti pro LSRAM (kontinuální DMA)	39
5.2	Naměřené hodnoty propustnosti pro DDR3 (kontinuální DMA)	40
5.3	Naměřené hodnoty propustnosti pro DDR3 (SG DMA)	40

Seznam symbolů a zkratek

PCI	Peripheral Component Interconnect
PCIe	PCI Express
TLP	Transaction Layer Packet. Transakční paket
DLLP	Data Link Layer Packet. Linkový paket.
TB	Theoretical Bandwidth. Teoretická propustnost $\left[\frac{GB}{s}\right]$.
TR	Transfer Rate. Přenosová rychlost. $\left[\frac{GT}{s}\right]$.
MPS	Maximum Payload Size. Maximální velikost užitečných dat.
CRC	Cyclic redundancy check. Cyklický redundantní součet.
AXI	Advanced eXtensible Interface.
BAR	Base Address Registers.
ATR	Address Table Register.

1

Úvod

Peripheral Component Interconnect Express (PCI Express) rozhraní jak je z názvu, patrně slouží pro připojení vysokorychlostních komponent. Dnes sloty pro připojení PCI Express (dále PCIe) komponent obsahuje každá základní deska osobního počítače. Slouží pro připojení grafické karty, SSD disků či síťové karty. Obecně je toto rozhraní vhodné pro různé typy rozšiřujících karet. Důležitým parametrem PCIe zařízení je, jakou PCIe generaci a šířku linky podporuje. Tyto dva parametry určují maximální teoretickou propustnost daného zařízení. V moderních základních deskách je dnes běžně podporovaná třetí a čtvrtá generace PCIe. Pro představu, slot pro grafickou kartu, který podporuje PCIe 4.0 x16 (šířka linky), může dosáhnout teoretické propustnosti 31.5 GB/s v jednom směru (63 GB/s pro obousměrný přenos dat). Skupina složená z více jak 800 společností s názvem PCI Special Interest Group (PCI-SIG) má na starosti údržbu a vývoj specifikací pro PCIe. Nejnovější vydaná specifikace je PCIe 5.0 a v roce 2021 je plánované vydání specifikace PCIe 6.0. Tato specifikace by měla v konfiguraci x16 dosahovat teoretické propustnosti 126 GB/s (jednosměrný přenos). Důvodem pro řešení diplomové práce je ověření možnosti využití rozšiřující PCIe karty v DAQ systémech. Tato rozšiřující karta by mohla provádět i předzpracování dat před předáním do PC aplikace a urychlit tak sběr dat. V teoretické části diplomové práce je proveden rozbor dnes běžně dostupných PC rozhraní. Dále je podrobněji popsáno rozhraní PCI Express. Cílem praktické části diplomové práce je realizovat přenos dat mezi PC a FPGA. Realizace je provedena na vývojové desce EVEREST-DEV-BOARD od společnosti Arrow Electronics. Tato deska využívá Polarfire FPGA čip MPF300TS-1FCG1152EES od Microsemi. Firmware je psán v jazyce VHDL a celkový systém je složen z vlastně psaného kódu a dostupných IP jader. Výsledný systém podporuje zápis a čtení dat z paměti FPGA (DDR3, LSRAM) a FPGA registrů. Dále je pro názornost možné ovládání 4 led diod, které se na desce nachází. Testovací software na straně PC využívá výrobcem dodaný driver a dll. Propustnost byla měřena pro druhou generaci PCIe s šířkou linky x4. V závěru práce jsou zhodnocené dosažené výsledky a je diskutována možná budoucí integrace do akvizičního systému. V diplomové práci se využívají předpony dle standardu SI (1 kB = 1000 B). Pokud se použije binární význam předpony (1 kB = 1024 B) bude to explicitně řečeno.

2

Rozhraní PC s vysokou datovou propustností

V této kapitole je popsán přehled rozhraní vhodných pro přenos dat do PC s vysokou datovou propustností. Přehled je zaměřen na historii a princip funkce rozhraní. Dále jsou v přehledu uvedeny různé verze, generace či standardy rozhraní s důrazem na přenosovou rychlost. V kapitole je užíván výraz *plug-and-play* a *hot swapping*, pro přehlednost je zde uvedena jejich definice:

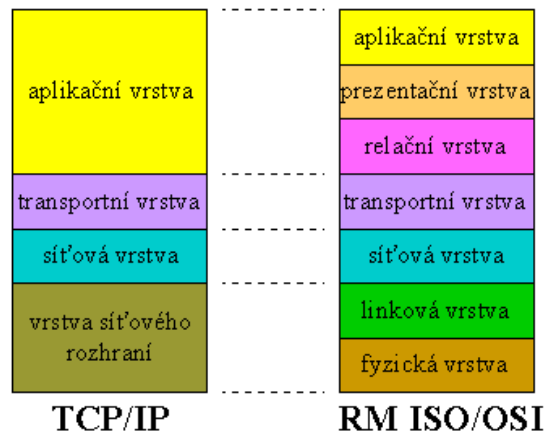
Plug-and-play (PnP) je schopnost zařízení fungovat hned po zapojení do systému. Uživatel nemusí instalovat ovladače zařízení ani systému sdělit, že bylo přidáno nové zařízení. *Hot swapping*, neboli *hot-plugging*, je možnost připojení zařízení do spuštěného systému bez nutnosti vypnutí napájení či restartu. Zatímco zařízení nepodporující hot swapping lze zapojit pouze do vypnutého systému.

2.1 Ethernet

Ethernet je protokol pro přenos dat definovaný ve specifikaci IEEE 802.3. Tato specifikace popisuje velké množství typů Ethernetu. Liší se například definovanou maximální přenosovou rychlostí či typem přenosového média (optické x metalické vedení).

Ethernet pracuje ve fyzické a linkové vrstvě referenčního ISO/OSI modelu (obr 2.1). V modelu TCP/IP spadá do vrstvy síťového rozhraní. Model TCP/IP je rozdělen na 4 vrstvy (obr. 2.1). Nejnižší vrstva síťového rozhraní má na starosti fyzickou přenosovou cestu společně s příjmem a vysláním datových paketů. Specifikace této vrstvy závisí na použité technologii (např. Ethernet). Další vrstva je vrstva síťová, která se stará, aby paket dorazil ke správnému příjemci (realizována pomocí Internet protokolu (IP)). Druhá nejvyšší vrstva se označuje jako transportní vrstva, která má za úkol přenášet aplikační data. Tato vrstva je realizována TCP (Transmission Control Protocol) nebo UDP (User Datagram Protocol) protokolem. TCP protokol zajišťuje spolehlivý přenos, kdy se před přenosem dat musí navázat spojení, ale má větší overhead než protokol UDP.

Oproti TCP má UDP nespolehlivý přenos dat, protože nepoužívá potvrzovací mechanismus. Díky tomu má UDP menší overhead, a tudíž je rychlejší. Nejvyšší vrstvou je aplikační vrstva, do které spadají aplikační protokoly (např. http). Při využití UDP protokolu se o spolehlivost přenosu stará aplikační program. Každá vrstva na vysílaná data přidává svůj overhead a výsledný paket je poté zabalen do Ethernet rámce.



Obr. 2.1: Jednotlivé vrstvy modelu TCP/IP a referenčního modelu ISO/OSI |Převzato z [13]|

2.1.1 Fast Ethernet

V roce 1995 byl představen standard IEEE 802.3u - Fast Ethernet (100BASE-X). Tento standard zvedl přenosovou rychlost z 10 Mbit/s (předchozí verze standardu) na 100 Mbit/s. Podporuje přenos po metalickém (100BASE-T) i optickém (100BASE-F) vedení. Standard podporuje plně duplexní přenos a plug-and-play.

2.1.2 Gigabit Ethernet

V roce 1998 byl vydán standard IEEE 802.3z, který definoval Gigabit Ethernet (GbE) po optickém vlákně. Tento standard zvedl přenosovou rychlost na 1 Gbit/s. Standard definující GbE po kroucené dvojlince byl vydán o rok později s názvem IEEE 802.3ab s označením 1000BASE-T.

2.1.3 10 Gigabit Ethernet

Ethernet s přenosovou rychlostí 10 Gbit/s byl poprvé definován v roce 2002 ve standardu IEEE 802.3ae (pouze po optice). Oproti předchozím standardům 10 Gigabit Ethernet (10GbE) definuje pouze plně duplexní přenos. Přenosová vzdálenost po metalickém vedení je za použití vhodných kabelů 100 m (kabely Cat 6A a výš).

2.1.4 Nejnovější specifikace

Výše uvedený výčet standardů zdaleka neobsahuje všechny dnes dostupné přenosové rychlosti. Poslední specifikace v době psaní této práce uvádí přenosovou rychlost 400 Gbit/s přes optické vlákno. Tyto přenosové rychlosti zatím nejsou určeny pro osobní počítače, a proto zde nejsou popsány.

2.1.5 Power over Ethernet

Power over Ethernet (PoE) označuje standardy pro přenos napájení společně s daty po Ethernetu přes kroucenou dvojlinku. Tři techniky jsou standardizovány pro realizaci PoE. Jsou to alternative A, alternative B a 4PPoE. Alternative A přenáší napájení a data po stejných vodičích. Tato technika je možná díky tomu, že Ethernet využívá diferenciální signalizaci a linky jsou galvanicky oddělené. Alternative B je určena pro přenosy, kde jsou využity jen 2 ze 4 signálových párů a zbylé dva jsou využity pro napájení. Poslední technika se nazývá 4PPoE a využívá pro přenos napájení všechny 4 páry, což umožňuje přenos vyšších hodnot výkonu. Nejnovější standard uvádí maximální možný přenesený výkon 71,3 W.

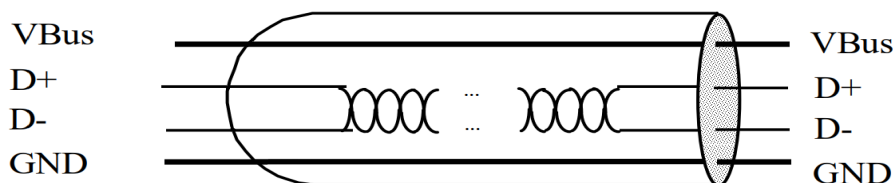
2.2 USB

V roce 1996 byla představena první specifikace standardu Universal Serial Bus (USB). Cílem bylo vytvoření levného univerzálního standardu, který by zjednodušil připojení externích periférií do PC a zredukoval počet různých portů (sériový, paralelní, PS/2...). USB, jak je z názvu patrné, je sériová sběrnice využívající stromovou topologii. Na začátku architektury je host (PC), který vždy obsahuje host root hub, ke kterému se připojují ostatní zařízení. Zařízení se dělí na rozbočovače USB (huby), sloužící pro rozvětvení sběrnice a koncové zařízení USB. Připojené USB zařízení lze na sběrnici identifikovat pomocí USB adresy, která je mu přiřazena po připojení na sběrnici. Maximální množství připojených zařízení je 127 (počítají se i huby). Rozhraní podporuje plug-and-play a hot swapping technologie.

USB podporuje 4 typy přenosů. Řídící přenos pro konfiguraci a ovládání zařízení. V případě chyby se přenos opakuje. Izochronní přenos určený pro zařízení vyžadující stálý přísun dat (zařízení jako je mikrofón či reproduktor). V případě chyby se přenos neopakuje, protože je důležité, aby data dorazila včas, nikoliv bezchybně. Další typ přenosu je blokový přenos. Tento přenos je určený pro přenos velkého množství dat. Chybný přenos se opakuje, protože jeho prioritou je bezchybný přenos (např. tiskárna). Poslední typ přenosu je přenos s přerušením. Přenos s přerušením je určený pro zařízení vyžadující periodický přenos malého množství dat (např. myš, klávesnice) s vysokou spolehlivostí.

2.2.1 USB 1.x a USB 2.0

USB 1.x (1.0, 1.1) lze provozovat ve dvou módech. Low speed (LS) s přenosovou rychlostí 1.5 Mbit/s a full speed (FS) s přenosovou rychlostí 12 Mbit/s. Původní myšlenka byla mít jen jednu rychlost, ale toto rozdělení umožnilo levnou realizaci v low speed zařízeních (klávesnice, myši, herní ovladače...). Pro připojení se specifikoval USB konektor typu A a typu B, kdy oba typy využívají 4 piny. Jak je vidět na obrázku 2.2, 2 piny se využívají pro data a 2 pro napájení (VBus = +5V).



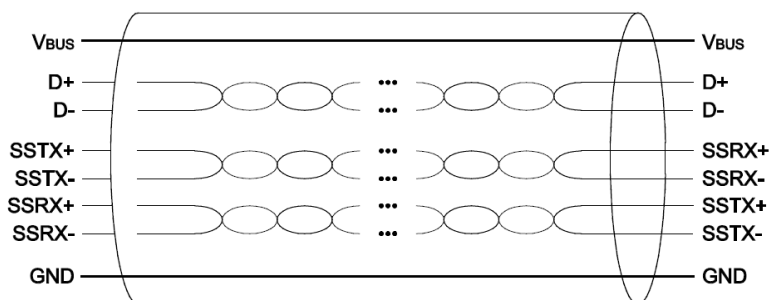
Obr. 2.2: Naznačení vodičů v kabelu pro USB 1.x a USB 2.0 [Převzato z [12]]

Specifikace USB 2.0 byla představena v roce 2000 s názvem HighSpeed. Jedná se o přímé vylepšení předchozí generace. K přenosovým módům LS a FS přidala high speed (HS) s přenosovou rychlostí 480 Mbit/s. K již specifikovaným konektorům A a B byly dále přidány konektory typu mini a micro. Specifikace dovoluje běžný odběr do 100 mA, a po ohlášení odběru až 500 mA. Byl také definován nabíjecí port, který umožnil odběr do 1.5 A.

2.2.2 USB 3

Specifikace USB 3.0 byla vydána v roce 2008 pod názvem SuperSpeed. USB 3.0 je plně zpětně kompatibilní s USB 2.0 (při užití správného konektoru). Jak je vidět na obr. 2.3, nová verze sběrnice pro přenos využívá 8 pinů. Piny D+ a D- slouží k zajištění zpětné kompatibility (přenosové módy LS, FS a HS). Nové piny SSTX(±) a SSRX(±) jsou diferenciální páry pro vysílání a přijímání dat v módu super speed (SS), který má přenosovou rychlost 5 Gbit/s. Pro přenos se používá 8b/10b kódování (20% overhead), z čehož vychází datová propustnost 4 Gbit/s. Odběr se zvýšil na 150 mA pro běžný a 900 mA pro ohlášený. Architektura rozhraní USB 3.0 byla vytvořena jako paralelní zapojení sběrnice USB 2.0 a USB SuperSpeed (zachována stromová topologie). Pokud je přenos v módu SS, využívá se strana USB SuperSpeed, a pokud se používá nižší rychlost, využije se strana USB 2.0. I když se jedná o paralelní zapojení dvou sběrnic, současný provoz není možný. Díky této konstrukci je možné užívat starší USB zařízení na řadičích USB 3.0.

Konektor typu A byl pro USB 3.0 doplněn o SS piny, ale zachovává si zpětnou kompatibilitu. Konektor typu B si zachoval jen částečnou zpětnou kompatibilitu, protože lze připojit starý kabel do nové zástrčky, ale naopak to možné není. V této specifikaci byl představen nový typ konektoru, a to konektor typu C. Konektor obsahuje 24 pinů a je



Obr. 2.3: Schéma kabelu pro USB 3.x |Převzato z [12]|

mechanicky symetrický (lze ho připojit i otočený o 180°). Doporučená maximální délka kabelu je 3 m.

2.2.3 USB 3.1

V roce 2013 byla představena nová verze specifikace USB 3.1. V této specifikaci byl zaveden nový mód rychlosti SuperSpeed USB 10 Gbps, označovaný jako USB 3.1 Gen 2 (SS+). Jak je z názvu patrné, zvýšila se tím přenosová rychlost na 10 Gbit/s. Tento mód již dále nepoužívá 8b/10b kódování, ale kódování 128b/132b (3% overhead). Dále došlo k přejmenování stávajícího SS módu na USB 3.1 Gen 1.

2.2.4 USB 3.2

V roce 2017 byla představena USB 3.2 specifikace, která zdvojnásobila šířku pásma pro SS módy. Stejně jako u USB 3.1, je USB 3.2 zpětně kompatibilní a definuje 4 SS přenosové módy. USB 3.2 Gen 1 (stejný jako USB 3.0 SS), USB 3.2 Gen 2 (stejný jako USB 3.1 Gen 2), USB 3.2 Gen 1x2 a USB 3.2 Gen 2x2.

2.3 FireWire - IEEE 1394

FireWire je sériové rozhraní pro přenos informací mezi dvěma zařízeními se zaměřením na vysokorychlostní komunikaci. Důvod pro vývoj rozhraní bylo vytvořit sériovou náhradu paralelního rozhraní SCSI a zároveň poskytnout možnost připojení audio a video techniky. FireWire byl vyvinut v roce 1995 společností Apple, ve spolupráci s mnoha organizacemi, primárně s firmou Sony a Panasonic. FireWire byl standardizován jako specifikace IEEE 1394 High Performance Serial Bus. Užití názvu FireWire bylo licencováno společností Apple, a to vedlo k různým obchodním názvům označující stejný standard (i.LINK pro Sony, Lynx pro Texas Instruments). Jako oficiální název se využívá IEEE 1394 dle stejnojmenného standardu. Při vytváření FireWire standardu měli návrháři několik základních cílů:

- Rychlý přenos dat.

- Možnost připojení několika zařízení na sběrnici.
- Levné kabely a implementace.
- Možnost napájení přes kabel.
- Podpora hot-plug a plug-and-play.
- Podpora izochronního a asynchronního přenosu.

FireWire se hojně užívalo k připojení digitálních kamer a zařízení pro ukládání dat. Rozhraní našlo využití i pro systémy počítačového vidění.

2.3.1 Specifikace pro FireWire

FireWire umožňuje propojení 63 externích zařízení do stromové nebo zřetězené (angl. daisy-chain) architektury. Daisy-chain znamená, že lze připojit několik zařízení do jednoho portu. Rozhraní podporuje peer-to-peer komunikaci mezi zařízeními. Lze tak provádět komunikaci mezi zařízeními bez nutnosti procesoru. FireWire využívá přenosu pomocí paketů, které obsahují 32 bitové entity nazývané quadlets [8]. FireWire je schopný pracovat v asynchronním nebo izochronním módu. V asynchronním módu paket obsahuje ID zdrojové a cílové stanice, zatímco u izochronního módu jsou pakety rozeznány podle přiřazeného čísla kanálu. Zaručuje to, že data přijdou v čas. 80% sběrnice je vyhrazeno pro izochronní cykly a zbývajících 20% je pro asynchronní data.

Specifikace uvádí základní rozdělení konektorů na alfa a beta konektor. Alfa konektory se užívají pro verzi FireWire 400 a beta konektory pro ostatní verze rozhraní. Standardní alfa konektor má 6 pinů (4 piny pro signál, 2 piny pro napájení). Dále je specifikován alfa konektor od Sony (i.LINK), který má 4 piny (neobsahuje piny pro napájení). Beta konektor má 9 pinů (4 piny pro signál + 2 referenční země, 2 piny pro napájení a 1 rezervovaný pin). FireWire má zpětnou kompatibilitu, ale vzhledem k odlišné konstrukci alfa a beta konektorů je nutné užít adaptéry či speciálních kabely. Jednotlivé specifikace byly v roce 2008 sjednoceny do standardu IEEE 1394-2008.

2.3.1.1 FireWire 400

FireWire 400 označuje původní specifikaci z roku 1995 (IEEE 1394-1995). Realizuje half-duplex komunikaci dosahující přenosové rychlosti 100, 200 nebo 400 Mbit/s. Tyto rozdílné přenosové rychlosti se často označují jako S100, S200 a S400. Maximální délka kabelu je 4,5 m, ale sběrnice FireWire umožňuje využití opakovačů, kdy jako opakovač může sloužit i jiné zařízení.

2.3.1.2 FireWire 800

Specifikace FireWire 800 (IEEE 1394b-2002) zdvojnásobila přenosovou rychlost oproti FireWire 400 a dovolila plně duplexní komunikaci. Délka kabelu se zvýšila na 100 m. Dále se v této specifikaci zavedlo využití 8B/10B kódování a beta konektorů.

2.3.1.3 FireWire S1600 a FireWire S3200

Verze FireWire S1600 a S3200 byly do jisté míry popsány v původní specifikaci IEEE 1394b-2002 a doplněny ve specifikaci IEEE 1394-2008. Jak je z názvu patrné, zvýšila se přenosová rychlost na 1,6 Gbit/s a 3,2 Gbit/s.

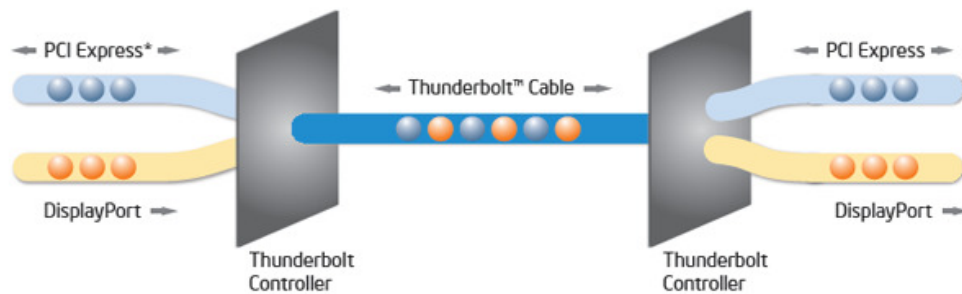
2.3.2 Konec FireWire

Výhoda rozhraní FireWire byla vyšší přenosová rychlost oproti USB 2.0, a to nejen ve verzi FireWire 800, ale i verze FireWire 400 byla v reálném provozu rychlejší (díky možnosti peer-to-peer komunikace). Bohužel FireWire byl dražší než USB a mimo audio a video techniku si rozhraní nezískalo velké zastoupení. FireWire byl využíván komerčně převážně společností Apple v produktech MacBook. Poslední řada s FireWire portem vyšla v roce 2012 a v roce 2014 už bylo toto rozhraní nahrazeno rozhraním Thunderbolt. Výhoda vyšší přenosové rychlosti oproti rozhraní USB již neplatí, a proto už nemá toto rozhraní v současnosti velké využití.

2.4 Thunderbolt

Thunderbolt je rozhraní pro připojení externích periférií k počítači vyvinuté společností Intel ve spolupráci se společností Apple. Thunderbolt kombinuje PCI Express (PCIe) a DisplayPort (DP) do jedné sériové linky. To je možné díky namapování těchto dvou protokolů do jednoho meta protokolu. Výhoda meta protokolu spočívá v tom, že operační systém vidí připojená zařízení jako PCIe nebo DP, a lze tedy použít standardní ovladače. Díky tomu umožňuje Thunderbolt vysokorychlostní přenos dat z periférií a zároveň HD přenos videa na stejném kabelu. Rozhraní lze realizovat na metalickém (do délky 3 m) i optickém vedení (do délky 50 m). Maximální délka je dána pro jeden kabel, ale zařízení je možné propojit na delší vzdálenost pomocí daisy chain propojení. Rozhraní Thunderbolt je založeno na switched fabric architektuře s plně duplexními kanály. Základní vlastnosti Thunderbolt rozhraní jsou:

- Video a data přes jeden kabel pomocí dvojitého protokolu (PCI Express a Display-Port).
- Oboustranný přenos dat.
- Podpora hot-plug a plug-and-play.

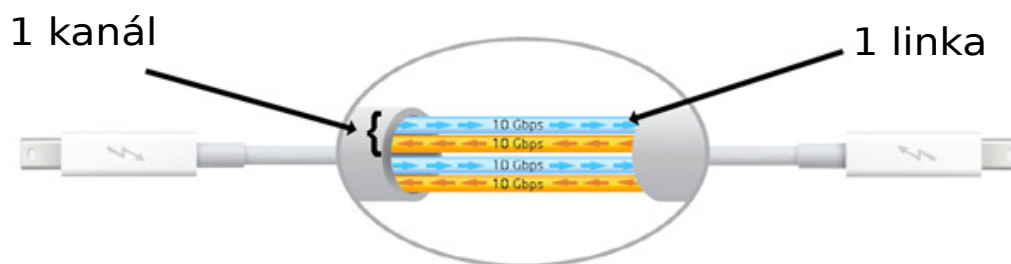


Obr. 2.4: Princip funkce Thunderbolt rozhraní |Převzato z [11]|

- Kompatibilita s existujícími DisplayPort zařízeními (kromě Thunderbolt 3).
- Užití nativních software ovladačů (PCIe a DP).
- Možnost napájení přes kabel.
- Nízká latence s přesnou časovou synchronizací.
- Možnost daisy chain propojení zařízení.

2.4.1 Thunderbolt generace

První generace rozhraní Thunderbolt byla představena v roce 2011. Rozhraní obsahuje 2 kanály, kde je každý kanál tvořen *upstream* a *downstream* linkou. Linka poskytuje 10 Gbit/s šířku pásma. Dohromady mají tedy 2 kanály 40 Gbit/s šířku pásma. Jako konektor se využívá Mini DisplayPort a umožňuje napájení po metalickém vedení do 10 W.



Obr. 2.5: Princip vedení pro Thunderbolt první generace |Převzato z [11]|

2.4.2 Thunderbolt 2

V roce 2013 Intel oznámil Thunderbolt 2. Dva kanály z předchozí generace se spojily v jeden a dosáhlo se tak data-rate 20 Gbit/s, kdy celková šířka pásma zůstala 40 Gbit/s.

Dále Thunderbolt 2 podporuje DisplayPort 1.2, a tím umožňuje přenos 4k videa do monitoru. Kabely i konektory pro Thunderbolt 2 jsou identické s těmi pro Thunderbolt a je tedy zachována zpětná kompatibilita.

2.4.3 Thunderbolt 3

Thunderbolt 3 byl představen v roce 2015. Jednou z hlavních změn oproti předchozím generacím je užití USB-C konektoru. Dále se šířka pásma zvýšila na 80 Gbit/s a obsahuje 2 plně duplexní kanály s přenosovou rychlostí 20 Gbit/s nebo 4 plně duplexní kanály s přenosovou rychlostí 10 Gbit/s. Výkon pro napájení se zvedl na 15 W, ale díky využití USB-C konektorů je možné implementovat specifikaci USB Power Delivery, která umožňuje výkon do 100 W. Dále kontrolér pro Thunderbolt 3 podporuje PCIe Gen3 a může podporovat buď x4, nebo x2 šířku linky (o PCIe více v následující kapitole). Produkty vyvinuté pro specifikaci Thunderbolt a Thunderbolt 2 budou fungovat i na Thunderbolt 3, ale je nutné užití adaptérů.

2.4.4 Thunderbolt 4

V roce 2020 byla oznámena specifikace Thunderbolt 4. Tato specifikace podporuje protokol USB4, DisplayPort 1.4. Šířka pásma zůstala na 80 Gbit/s.

2.4.5 Šířka pásma pro data

Z výše uvedených informací se zdá být rozhraní Thunderbolt se šířkou pásma 40 Gbit/s jako velice vhodné rozhraní pro přenos dat. Šířka pásma pro data je však velice odlišná. Thunderbolt kontrolér je připojen na CPU přes PCIe gen3 x4, a z toho vyplývá maximální šířka pásma přibližně 31.52 Gbit/s. Ovšem z dostupné dokumentace [10] je pro datový přenos uvedená šířka pásma jen 22 Gbit/s. Přesný důvod tohoto omezení není uveden a podrobnější dokumentace není dostupná.

3

PCI Express

Sběrnice PCI Express (dále PCIe) je nástupce sběrnice Peripheral Component Interconnect (PCI). Na rozdíl od svého předchůdce je PCIe sériová sběrnice, kdežto PCI je paralelní. PCI protokol byl vytvořen v roce 1993 a v době uvedení PCIe (2003) už byla tato sběrnice běžně používaná v osobních počítačích. Z toho důvodu byla zpětná kompatibilita se softwarem využívající PCI jeden z hlavních cílů při návrhu PCIe.

Tohoto cíle je u PCIe dosaženo tím, že se celý adresní prostor PCI přesunul do PCIe buď nezměněný, nebo rozšířený. Například konfigurační prostor byl rozšířen o mnoho nových registrů, které podporují moderní funkcionalitu. Software napsaný před mnoha lety na PCI bude fungovat i na dnešních PCIe zařízeních.

V kontextu této diplomové práce je přenosová rychlost brána jako údaj popisující kolikrát za sekundu je sběrnice schopná přenést data (nikoliv jak velké množství dat). Uvádí se v GT/s (Gigatransfer per second).

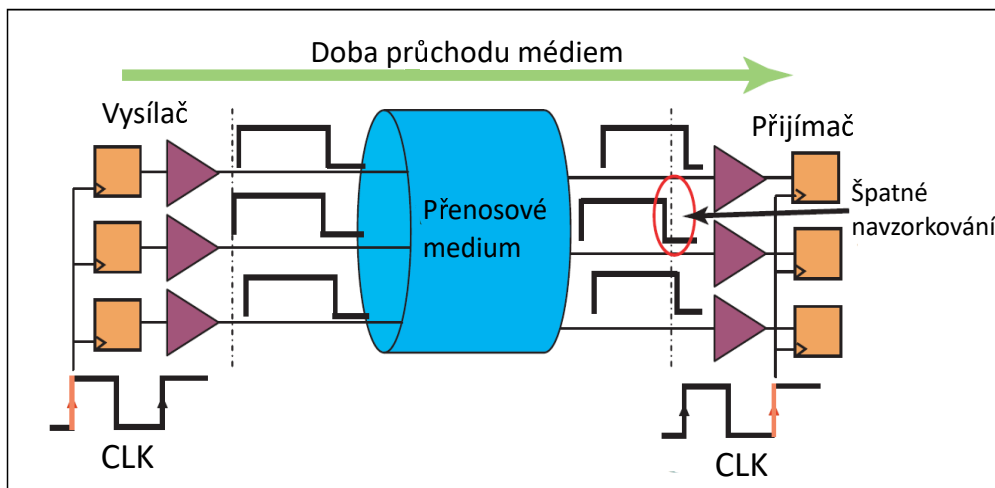
3.1 Sériový přenos

Přechod na sériový přenos z paralelního byl nevyhnutelný. PCI architektura umožňuje připojit teoreticky 32 zařízení na sběrnici. Prakticky je to ovšem okolo 5 zařízení při základní frekvenci 33 MHz a šířce sběrnice 32 bitů. Důvodem je užití signalizace odraženou vlnou (angl. reflected-wave signaling). Při zvýšení frekvence na 66 MHz a šířky na 64-bit se množství připojených zařízení sníží na 1 maximálně 2.

Propustnost 66 MHz PCI sběrnice (533 MB/s) je sice dvojnásobná oproti 33 MHz verzi (266 MB/s), ale další výrazné zlepšení už z podstaty PCI architektury nebylo možné, a proto bylo nutné přejít na přenos sériový. Aby sériový přenos překonal šířku pásma paralelního přenosu, musí využívat mnohem vyšší frekvence. Přenosové rychlosti 2.5 GT/s (Gen1) a 5.0 GT/s (Gen2) jsou na rozdíl od PCI pro PCIe snadno dosažitelné, protože zvyšování frekvence sériového přenosu nezpůsobí tak velké problémy jako u paralelního přenosu.

3.1.1 Problém paralelního přenosu PCI

Problémy paralelního přenosu začnou být jasné při snaze zvýšit přenosovou rychlost. Paralelní přenosy užívají společné hodiny (angl. common clock), to znamená, že na hranu prvního hodinového impulsu jsou data vyslána a na hranu druhého hodinového impulsu jsou přijata. Z toho vychází časová rezerva jednoho hodinového taktu. Čím rychlejší bude frekvence hodin, tím bude menší časová rezerva.



Obr. 3.1: Problémy a omezení paralelního přenosu PCI [Převzato z [1]]

První problém paralelního přenosu spočívá v různém zpoždění signálů (angl. signal skew). Když se vyšle více bitů zároveň, na každý bit působí trochu odlišné zpoždění a na přijímač dorazí v trochu odlišných časech. Při velkém rozdílu mohou být data chybně navzorkována (obr. 3.1). Druhý problém je se synchronizací hodin mezi více zařízeními (angl. clock skew). Moment, kdy přijde hrana hodin do jednoho zařízení není shodný s tím, kdy vejde do jiného a tato skutečnost dále snižuje časovou rezervu. Dalším problémem je doba, než projde signál z vysílače do přijímače. Tato doba průchodu signálu daným médiem (angl. flight time) musí být menší, než je perioda hodin nebo časová rezerva. Z toho plyne, že implementované signálové cesty musí být krátké, aby byla splněna výše uvedená podmínka pro dobu průchodu.

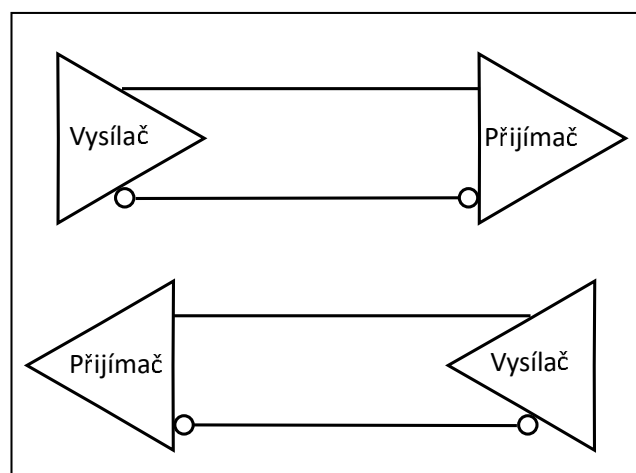
3.1.2 Překonání problémů paralelního přenosu

Doba průchodu signálu z vysílače do přijímače pro sériový přenos není problémem, protože hodiny jsou zabudovány v datovém toku a není třeba jejich externí reference. Ze stejného důvodu odpadá i problém zpoždění hodinových impulsů mezi zařízeními. Problém různého zpoždění signálu neexistuje v uspořádání s jedním přenosovým párem, protože v lince se posílá jen jeden bit dat. Problém se ovšem vrátí, pokud použijeme více přenosových párů. Ovšem toto zpoždění dokáže do jisté míry opravit přijímač.

3.2 Přenos signálu v PCIe

Dvě PCIe zařízení jsou spojena pomocí linky (angl. Link). Tato linka se skládá z jednoho nebo více přenosových párů. Jeden takový pár se anglicky nazývá *lane* a je tvořen diferenciálním přijímačem a vysílačem signálu (obr. 3.2). Počet párů v lince udává její šířku (angl. Link Width). Daný počet párů je zápisem reprezentován jako x1 (jeden pár), x2 (dva páry), x4, x8, x16 a x32. Pro přenos dat mezi zařízeními stačí jen jeden pár, přičemž ale větší počet párů znamená větší propustnost (více v kap. 3.5). I když je ve specifikaci pro PCIe [5] šířka linky x32 definována, tak v praxi se tato šířka nachází jen zřídka.

PCIe využívá oboustranné připojení, které umožňuje data současně přijímat i vysílat. Jedná se tedy o plně duplexní (angl. full-duplex) komunikaci.



Obr. 3.2: Jeden přenosový pár PCIe linky

3.2.1 Diferenciální signalizace

Každý přenosový pár využívá pro přenos diferenciální signál, jak je naznačeno na obr. 3.2. Diferenciální uspořádání sice zdvojnásobí počet pinů vůči signalizaci se společnou zemí, ale přináší dvě velké a důležité výhody pro vysokorychlostní signály. První je zvýšená odolnost vůči šumu, která vyplývá z podstaty diferenciálního přenosu a druhou výhodou je možnost užití nižšího signálového napětí.

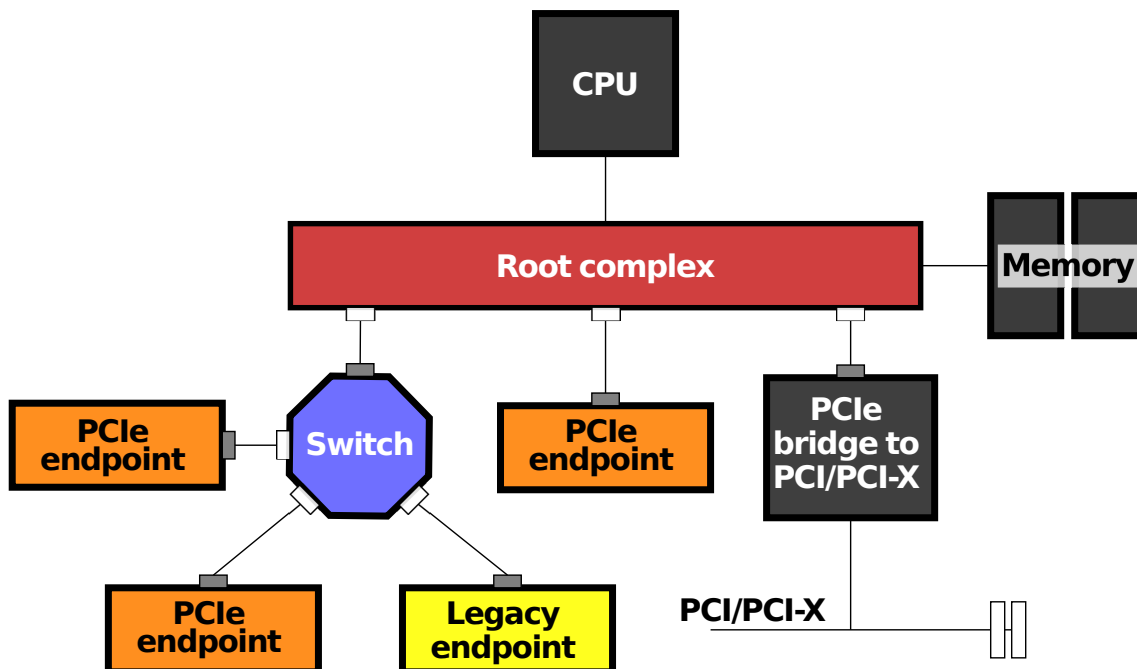
3.2.2 Paketový protokol

Přechod na sériový přenos značně redukoval počet datových pinů. U PCIe byly stejně jako u většiny sériových protokolů piny pro řídicí signály nahrazeny pomocí paketů. Pro řídicí signály byl přidán samostatný adresový prostor pro zprávy (angl. Message Space), který podporuje předchozí PCI řídicí signály (interrupty, power-management, ...). Lze si tak PCIe Message transakci představit jako virtuální řídicí vodiče.

V PCIe systému se vyskytují dva druhy paketů DLLP (Data Link Layer Packet) a TLP (Transaction Layer Packet). Samotná užitečná data (angl. payload) jsou přenášena jako součást TLP. Příjímač rozpozná začátek paketu a podle známé struktury jej dekoduje.

3.3 Topologie PCIe

Největší změnou a vylepšením PCIe oproti PCI je přechod ze sdílené sběrnice (angl. shared-bus) na dvoubodové spojení (angl. point-to-point). Ve dvoubodovém spojení je funkce sdílené sběrnice nahrazena pomocí přepínače (angl. switch), který provádí přepojování. Každé koncové zařízení (angl. Endpoint) má přímé spojení do přepínače pomocí linky a není proto nutné provádět arbitráž. Zařízení PCIe bridge to PCI/PCI-X zajišťuje připojení PCI/PCI-X systémů do PCIe systému.



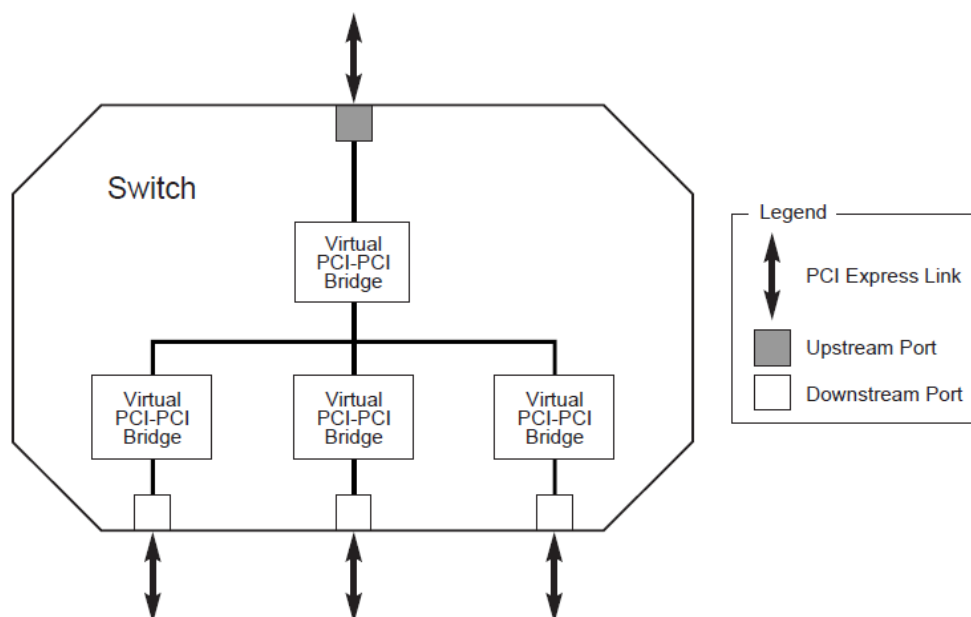
Obr. 3.3: PCIe topologie [Převzato z [2]]

3.3.1 Root Complex

V PCIe systému má Root Complex (RC) za úkol propojit procesor a paměť se zbytkem PCIe systému. RC může podporovat jeden nebo více PCIe portů. Na každý tento port může být připojeno jedno koncové zařízení nebo přepínač pro připojení více koncových zařízení. Zapojení bez použití přepínače se na základních deskách používá pro připojení grafické karty.

3.3.2 Switch

Přepínač umožňuje připojení více zařízení k jednomu PCIe portu (rozvětjuje PCIe topologii). Pro konfigurační software se přepínač jeví jako jeden nebo více PCI-PCI mostů (angl. PCI-PCI bridge). Přepínač může mít směrem od RC několik portů, ale směrem k RC musí mít pouze jeden port. V blokovém schématu na obr. 3.4 je tato vlastnost označena jako downstream a upstream port. Přepínač může spojit dvě PCIe zařízení na stejné úrovni, a tím jim umožní posílání dat bez průchodu RC (peer-to-peer komunikace).



Obr. 3.4: Blokové schéma PCIe přepínače [Převzato z [5]]

3.3.2.1 ACS

Jak již bylo zmíněno výše PCIe specifikace umožňuje peer-to-peer transakce. To znamená, že je možné a v některých případech žádané, aby jedno koncové zařízení poslalo data přímo do jiného koncového zařízení bez nutnosti průchodu RC. Tím je možný náhodný (i záměrně zlomyslný) zápis do neplatné/zakázané oblasti koncového zařízení, který způsobí problémy. Pro zabránění nechtěným transakcím byl zaveden mechanismus Acces Control Services (ACS), který nutí projít peer-to-peer transakce přes RC.¹

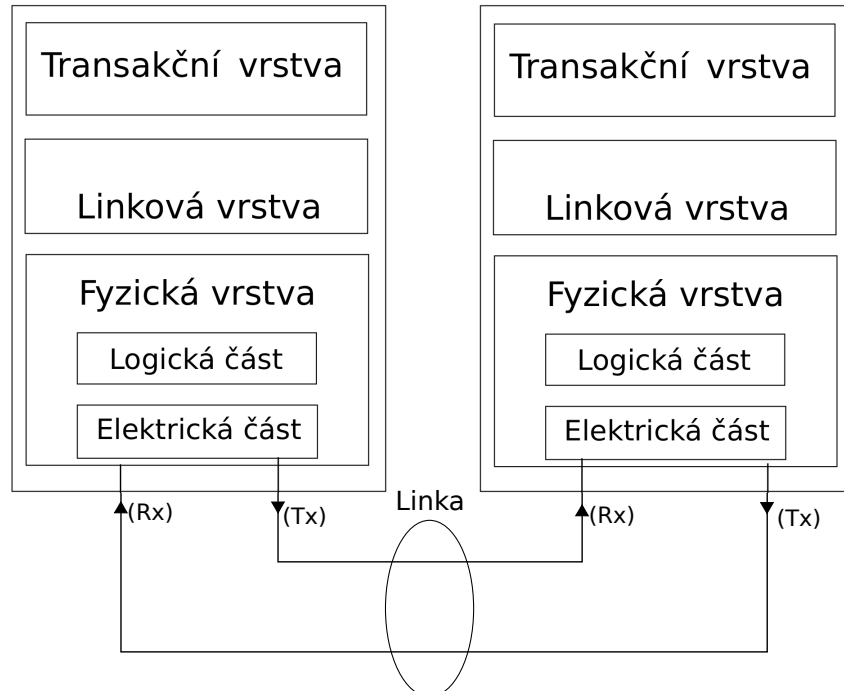
¹V dnešní době je ACS už součástí specifikací PCIe a většina PCIe přepínačů ACS podporuje. [7]

3.3.3 Koncové zařízení

PCIe topologie je zakončena koncovým zařízením (angl. Endpoint). Často tímto zařízením bývá grafická karta, ale může se jednat o libovolnou přídavnou kartu nebo zařízení, které je schopné vytvářet a přijímat požadavky PCIe přenosu. Aby zařízení bylo rozpoznáno jako koncové, musí mít v konfiguračním registru nastavený typ hlavičky 0 (angl. Header Type 0). Koncová zařízení mají vnitřní vstupně-výstupní registry namapované do adresového prostoru paměti (angl. memory mapped I/O - MMIO). Legacy endpoint existuje pro zachování zpětné kompatibility na zařízeních, která byla navržena pro fungování na starších sběrnicích (PCI-X). Zařízení pro legacy endpoint musí mít v konfiguračním registru příslušné nastavení. Legacy endpoint dovoluje použití věcí, které jsou v novějších návrzích zakázány. Těmi jsou např. využití IO prostoru nebo IO transakcí (novější návrhy dovolují pouze MMIO).

3.4 Přenosové vrstvy PCIe

PCIe specifikace definují vrstvou architekturu rozdělenou do 3 logických částí (vrstev). Transakční (angl. Transaction Layer), linkovou (angl. Data Link Layer) a fyzickou (angl. Physical Layer). Každá z těchto vrstev je dále rozdělena na přijímací a vysílací část. Každé zařízení (switch, endpoint) musí tyto vrstvy podporovat.



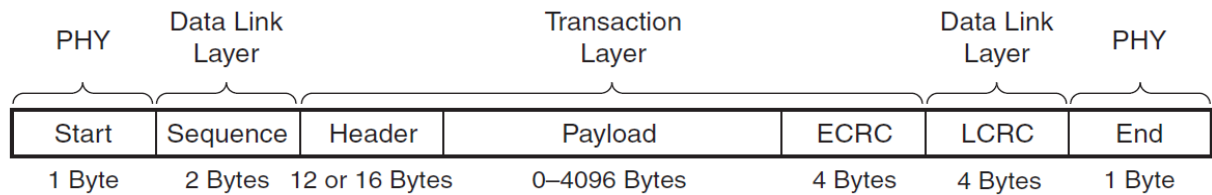
Obr. 3.5: Vrstvový model PCIe systému

Jak bylo popsáno v kap. 3.2.2 PCIe užívá pakety pro komunikaci mezi komponenty. Pakety jsou vytvořeny v transakční nebo linkové vrstvě a nesou informaci z vysílacího komponentu do přijímacího. Jak vyslaný paket postupně prochází přes jednotlivé vrstvy,

rozšiřuje se o další informaci, která je nutná pro správný přenos a zpracování paketu v jednotlivých vrstvách. Na přijímací straně se s procházejícím paketem děje přesný opak. Z paketu se postupně odstraňují nabalené informace, které pro následující vrstvu nemají význam.

3.4.1 Transakční vrstva

Horní vrstva architektury se nazývá transakční. Jejím úkolem je vytvoření transakčních paketů dále TLP (angl. Transaction Layer Packet) na vysílací straně a dekodování TLP na přijímací straně.



Obr. 3.6: Transakční paket (TLP) pro PCIe Gen2 |Převzato z [6]|

Jeden přenos (angl. Transaction) je definován jako kombinace paketu s požadavkem (angl. Request) do cílového zařízení a potvrzovacího paketu (angl. Completion) od cílového zařízení. Transakční vrstva podporuje 4 adresové prostory. Stejně jako PCI obsahuje paměťový, I/O a konfigurační prostor a přidává k nim prostor pro zprávy (viz. kap.3.2.2).

Požadavek se dále rozděluje na dva typy dle toho, jestli vyžaduje odpověď ve formě completion packetu (angl. non-posted) nebo ne (angl. posted). Výhoda posted transakcí spočívá v rychlosti, protože Requester nemusí čekat na odpověď, a tím odpadá overhead spojený s completion packetem. Requester v tomto případě nemá zpětnou vazbu o přenosu a neví, jestli došlo k chybě. Posted transakce stále využívají Ack/Nak protokol v linkové vrstvě.

3.4.1.1 Quality of Service

Dalším úkolem transakční vrstvy je aplikování QoS (Quality of Service). PCIe bylo navrženo s podporou časově citlivých přenosů jakou jsou audio nebo video aplikace, kde je důležité, aby data přišla včas. Systémy, které toto zajišťují, se označují QoS a v PCIe se skládají z několika částí. Nejdříve se softwarově každému paketu přiřadí priorita zápisem do 3-bitového pole Traffic Class (TC) uvnitř TLP hlavičky. Vyšší hodnota TC znamená vyšší prioritu.

Paket s daným TC se poté uloží do VC (Virtual Channel) bufferů. V každého portu se nachází několik těchto bufferů a podle TC se rozhodne do jakého VC se má paket uložit.

Jelikož má port na vysílání připraveno několik bufferů s pakety, je nutná arbitrážní logika pro zvolení z jakého VC dojde k přenosu.

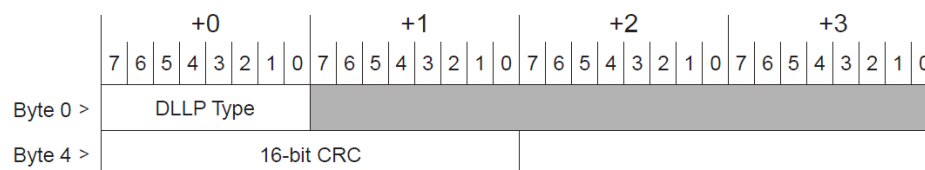
Finální část QoS tvoří přepínač, který musí zvolit jaký vstupní port dostane přístup na VC výstupního portu. Tato arbitráž portů (angl. Port Arbitration) může být nastavena hardwarem nebo programovatelná softwarově.

3.4.1.2 Flow Control

Poslední hlavní funkcí transakční vrstvy je Flow Control. Znamená to, že vysílač vyšle paket jen pokud je v přijímacím bufferu dostatek místa (příjímací buffer nemusí být nutně buffer cílového zařízení). Tím dojde k zlepšení výkonu, protože se odstraní události při kterých se musí přenos kvůli plnému přijímači opakovat. Aby se zabránilo čekání, musí dostatečně často přijímací buffer vysílat informaci o svém dostupném místě. K informaci o stavu (mimo jiné) se využívá paket DLLP (Data Link Layer Packet). TLP se nepoužívá, jelikož by nebyl při plném bufferu tento paket přijat a systém by přestal fungovat. DLLP může být vyslán a přijat nezávisle na stavu bufferu.

3.4.2 Linková vrstva

Linková vrstva (angl. Data Link Layer) má 3 hlavní funkce. Oprava TLP chyb, Flow Control a power management. Těchto funkcí dosahuje generováním linkových paketů DLLP. DLLP má jednoduchý formát s pevně danou velikostí 8 bajtů (i s rámcem), a na rozdíl od TLP nenese informaci o cílovém zařízení. DLLP se totiž používá jen na komunikaci s nejbližším zařízením (angl. nearest-neighbor). Pokud dojde k chybnému přenosu, nevysílá se DLLP znova, ale je zahozen a čeká se na další přenos. Samotný DLLP má vždy 6 bajtů. První bajt obsahuje typ DLLP (Ack/Nak, power management, ...) a poslední dva bajty jsou určeny pro CRC. Zbylé bajty jsou používány různě dle daného typu.



Obr. 3.7: Struktura DLLP bez začátku a konce rámce |Převzato z [5]|

Na vysílací straně linkové vrstvy je DLLP vytvořen, přidá se k němu 16-bit CRC a předá se do fyzické vrstvy. Tam se k DLLP připojí začátek a konec paketu (rámec) a odešle se. Na přijímací straně se nejdříve ve fyzické vrstvě od DLLP odstraní začátek a konec rámce. Zbytek paketu je následně předán do linkové vrstvy, kde se zkontroluje CRC a provedou se akce dané v paketu. DLLP je určen pro linkovou vrstvu, proto se nepřeposílá do transakční vrstvy.

Druhým typem paketu, který přes linkovou vrstvu prochází, je TLP. Vysílací strana přijme TLP vytvořené v transakční vrstvě a pro dané TLP se vytvoří CRC, sekvenční číslo a to se vyšle do fyzické vrstvy. Přijímací strana ověří integritu přijatého paketu. Pokud se v přenosu detekuje chyba, linková vrstva požádá o zopakování transakce.

3.4.2.1 Ack/Nak protokol

Každý vyslaný TLP obsahuje CRC a sekvenční číslo. Vysílač si ukládá kopii každého odeslaného TLP do Replay bufferu a má tuto kopii uloženou, dokud mu ze sousedního zařízení nepříjde potvrzení o přijmutí. Potvrzení přijímač posílá ve formě DLLP typu Ack s údajem o sekvenčním čísle posledního dobře přijatého TLP. Když vysílač přijme Ack DLLP, uvolní se z Replay bufferu TLP s daným sekvenčním číslem společně s TLP, které odeslal před ním.

Při detekci chyby přijímač přenášený TLP zahodí a vyšle vysílači DLLP typu Nak. Ten poté opakuje vysílání pro všechny nepotvrzené TLP. Tento proces se nazývá Ack/Nak protokol.

3.4.3 Fyzická vrstva

Fyzická vrstva je nejnižší vrstva v PCIe architektuře. Pakety typu TLP i DLLP jsou fyzickou vrstvou přijímány z linkové vrstvy pro přenos přes linku do fyzické vrstvy cílového zařízení. Specifikace rozděluje fyzickou vrstvu na 2 části, logickou a elektrickou. Logická část obsahuje digitální logiku, která připravuje pakety pro sériový přenos přes linku na vysílací straně a na straně přijímače upravuje přijatý paket pro horní vrstvy. Elektrická část je analogové rozhraní, které je připojeno na linku a obsahuje vazební kondenzátory, impedanční přizpůsobení a diferenciální přijímače/vysílače. Součástí fyzické vrstvy je i inicializace linky pomocí stavového automatu dle dostupných možností (nastavení šířky a rychlosti linky).

Kromě TLP a DLLP existuje ještě jeden typ přenosu, a to Ordered sets. Jedná se o přenos mezi fyzickými vrstvami dvou zařízení. Ordered sets vždy začínají COM symbolem, protože ho snadno přijímač rozezná. COM symbol jako jediný začíná dvěma bity stejné polaritý a následuje 5 bitů opačné polaritý. Ordered sets se posílají vždy přes každý přenosový pár v lince současně a používají se pro inicializaci linky, kompenzaci hodin a pro změnu power módů.

3.4.3.1 8b/10b kódování

PCIe² užívá 8b/10b kód pro přenos. U každého přenosového páru je 8b/10b enkodér, který převádí 8-bitové znaky na 10-bitové symboly.³ Jedním z úkolů kódování je udržet dostatečný počet hran v datovém toku, aby mohl přijímač z daného přenosu získat hodiny.

²Platí pro první dvě generace. Ostatní generace využívají 128b/130b.

³Patentováno firmou IBM v roce 1984.

V zakódovaném toku se nikdy neobjeví 5 stejných po sobě jdoucích bitů. Další vliv kódování je udržení nulové stejnosměrné složky. PCIe používá vazební kondenzátory v přenosové cestě pro odstranění stejnosměrné složky signálu. Pokud by se průměrně používalo v datovém toku více jedniček nebo nul, mohlo by dojít k nabití těchto kondenzátorů, což poškozuje integritu signálu. Enkodér sleduje, kolik měl předchodzí přenos jedniček nebo nul a podle toho zvolí, jak by měl další přenos vypadat.

3.4.3.2 Logická část

TLP i DLLP jsou přijímány do bufferu ve fyzické vrstvě a je k nim přidán začátek a konec (rámec paketu). V této vrstvě se každý bajt rozdělí mezi všechny přenosové páry, které jsou na lince dostupné. Tento proces se nazývá *byte striping* a lze si ho představit tak, že se každý přenosový pár chová jako nezávislá sériová cesta přes linku. Na přijímací straně se data zase složí dohromady.

První symbol reprezentující začátek rámce ("SDP" pro DLLP a "STP" pro TLP) musí být umístěn na lane 0 (první přenosový pár) a poslední odeslaný symbol musí být "END" (označuje konec rámce).

3.4.3.3 Elektrická část

Přijímač a vysílač jsou přes linku propojeny přes vazební kondenzátory (75-200 nF)[5]. Toto zapojení je užíváno ve více sériových přenosech, protože umožňuje rozdílné hodnoty referenčního napětí na straně přijímače a vysílače.

3.4.4 Bázová adresa

Každé zařízení v systému může mít různé nároky na potřebnou velikost a typ adresového prostoru. Typy adresového prostoru⁴ můžeme dělit na IO, MMIO s předběžným načtením dat (prefetchable MMIO (P-MMIO)) a MMIO bez předběžného načtení dat (non-prefetchable MMIO (NP-MMIO)). P-MMIO dovoluje zařízení předpřipravit data a zvýšit tak efektivitu přenosu, pokud předpřipravená data byla správná. P-MMIO je vhodné použít tam, kde nehrozí ovlivnění předběžným načtením dat. Pokud přečtení může ovlivnit daný paměťový prostor (např. registr, který po přečtení hodnoty sám danou hodnotu smaže), musí se použít NP-MMIO. PCIe zařízení sama o sobě nemohou rozhodovat jaká adresa má být použita k přístupu k jejich vnitřní paměti (o tom rozhoduje software). Zařízení potřebuje předat systémovému softwaru informaci o tom, jaký adresový prostor má být použit, aby mohl alokovat adresy požadovaného typu (IO, NP-MMIO nebo P-MMIO). Toto je zajištěno pomocí bázové adresy (angl. Base Address Registers (BAR)) v hlavičce konfiguračního prostoru. Hlavička typu 0 má k dispozici 6 BAR každý o velikosti 32 bitů (typ 1 má k dispozici 2 BAR).

Návrhář zařízení si je vědom jaké množství registrů/paměti by mělo být přístupné pomocí

⁴Myšlen je adresový prostor paměťový a vstupně výstupní (IO) nikoliv konfigurační

IO nebo MMIO (NP, P). Z těchto informací nastaví spodní bity (bity 0-3) BAR na určitou hodnotu, která udává typ a velikost adresového prostoru. Např. 4096 B adresový prostor lze zapsat nulou na bitech 4-11, protože $2^{12} = 4096$ B, kde 2^{12} označuje LSB adresového rozsahu. Horní bity jsou zapsány softwarem po přečtení spodních bitů a udávají bá-zovou adresu adresového rozsahu pro zařízení. Po naprogramování BAR lze přistupovat k vnitřním registrům/paměti přes rozsahy adres udaných v BAR.

3.5 Propustnost sběrnice PCIe

Výsledkem kombinace vysoké rychlosti a počtu linek je vysoká teoretická propustnost sběrnice. Propustnost sběrnice se určí z přenosové rychlosti, počtu linek a způsobu kódování. Pro zjednodušení se užívá označení generace a počet párů v lince např. Gen2 x4 (druhá generace PCIe, která má čtyři přenosové páry). Každá generace má dvojnásobnou propustnost oproti předchozí generaci a je zpětně kompatibilní. Propustnost pro jednotlivé konfigurace PCIe ukazuje tabulka 3.1, ve které jsou hodnoty vypočítány dle vzorce 3.1 (u generace 3 - 5 jsou pro přehlednost tabulky hodnoty zaokrouhleny na setiny)

Počet linek	x1	x2	x4	x8	x16	x32
Propustnost Gen1 (GB/s)	0,25	0,5	1	2	4	8
Propustnost Gen2 (GB/s)	0,5	1	2	4	8	16
Propustnost Gen3 (GB/s)	0,98	1,97	3,94	7,88	15,75	31,51
Propustnost Gen4 (GB/s)	1,97	3,94	7,88	15,75	31,51	63,02
Propustnost Gen5 (GB/s)	3,94	7,88	15,75	31,51	63,02	126,03

Tab. 3.1: Teoretická propustnost PCIe pro jednotlivé generace a pro různé počty linek (jednosměrný přenos).

3.5.1 Výpočet propustnosti

Pro výpočet teoretické propustnosti je nutné znát generaci PCIe (udává přenosovou rychlost a typ kódování) a šířku linky (počet přenosových párů).

Generace	Přenosová rychlost (GT/s)	Kódování
1	2,5	8b/10b
2	5	8b/10b
3	8	128b/130b
4	16	128b/130b
5	32	128b/130b

Tab. 3.2: Přenosová rychlost a typ kódování pro jednotlivé generace

$$TB(GB/s) = \frac{TR \cdot DIR \cdot LineWidth \cdot Encoding}{8} \quad (3.1)$$

- *TB* - Maximální teoretická propustnost (angl. Theoretical Bandwidth). Hodnota se nejčastěji udává v GB/s.
- *TR* - Přenosová rychlost (angl. Transfer Rate).
- *DIR* - 1 pro jednosměrný přenos, 2 pro duplexní přenos.
- *LineWidth* - Udává počet párů v lince (x1, x2, x4, ...).
- *Encoding* - Záleží na použitém kódování. První a druhá generace PCIe využívá 8b/10b kódování (20% overhead). Generace 3-5 využívá 128b/130b kódování (1,54% overhead).

Další omezení propustnosti dochází vlivem jednotlivých přenosových vrstev. Fyzická, linková i transakční vrstva přidá ke každému přenášenému paketu vlastní overhead, a tím dochází ke snížení efektivity přenosu. Transakční vrstva přidává povinnou hlavičku paketu a může přidávat i nepovinný end-to-end cyklický redundantní součet (ECRC). Linková vrstva (DLL) přidává k paketu sekvenční číslo (angl. sequence number) a linkové CRC (LCRC), což zajišťuje úspěšný přenos přes linku. Fyzická vrstva přidává informaci, která označuje začátek a konec paketu.

TLP overhead může mít velikost 20 - 28 bajtů. Přesná velikost závisí na použitém adresování a zda se použije ECRC. Pro 32-bit adresování je nutná velikost hlavičky 12 bajtů a pro 64-bit adresaci je hlavička o 4 bajty větší. Přenos velkého množství dat vyžaduje několik TLP, kdy každý má svůj overhead. I když má každý TLP vlastní overhead, užití většího množství TLP a větší velikosti payloadu vede ke zvýšení efektivity linky. Nastavení maximální velikosti payloadu (angl. maximum payload size (MPS)) v zařízení ovlivní maximální velikost TLP. Zvyšování MPS nutně nevede na odpovídající zvýšení efektivity linky, protože při zvyšování velikosti jednotlivých TLP začne efektivitu linky ovlivňovat provozní (angl. traffic) overhead.

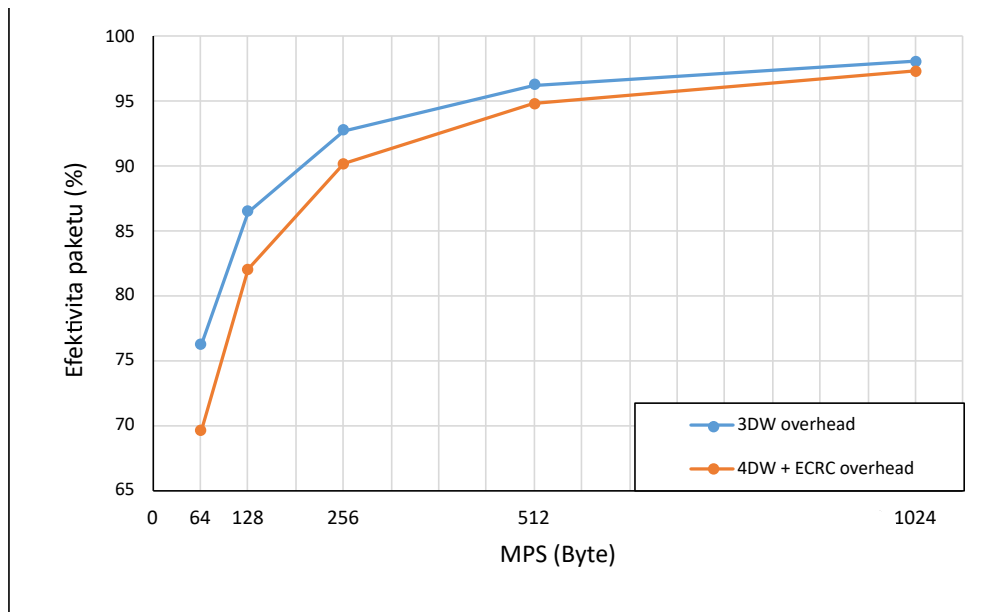
Fyzická i linková vrstva přidá traffic overhead. Za normálního provozu linky přidává fyzická vrstva skip-ordered sets pro kompenzaci přenosové rychlosti (angl. bit rate) mezi dvěma porty. Skip-ordered set má typicky 16 bajtů a vkládá se v intervalu mezi 1180 a 1538 symbol times. Symbol time udává dobu, za kterou se 10-bitů dostane na přenosový pár. Do středu TLP se nesmí vkládat jiné pakety nebo ordered sets, proto skip-ordered sets a DLLP musí čekat na mezeru mezi TLP přenosem. Tato skutečnost je jedním z důvodů, proč zvýšení MPS neodpovídá ekvivalentnímu zvýšení propustnosti linky. Zvyšující TLP znamená větší mezery mezi přenosem TLP, protože se do jednotlivých mezer musí vejít více DLLP a ordered set přenosů.

Pro každý přenesený TLP z jednoho zařízení do druhého musí být příjemcem vygenerován Ack/Nak DLLP a poslán do vysílače (blíže popsáno v sekci 3.4.2.1). Při velkém počtu odeslaných TLP je generováno i značné množství Ack/Nak DLLP, což omezuje dostupnou

šířku pásma linky. PCIe umožňuje, aby se několik DLLP stejného typu spojilo do jediného DLLP. Například, pokud vysílač vyšle 5 TLP a přijímač je úspěšně přijme, odešle Ack potvrzení se sekvenčním číslem posledního přijatého TLP. Vysílač po přijetí Ack potvrzení předpokládá, že se všechny TLP úspěšně přijaly. Výhodou je snížení množství traffic overheadu na dané lince. Pokud se Ack nevysílá dostatečně často, může dojít k zahlcení vysílacího replay bufferu. Flow control overhead záleží na častém posílání update FC DLLP paketů a jejich efektivnímu zpracování. Příliš frekventované vysílání update paketů zvyšuje traffic overhead, ale při nedostatečně častém vysílání dojde k zahlcení přijímacího bufferu.

3.5.1.1 Vliv MPS na přenos

Software musí zajistit, že každý paket nepřesáhne velikost parametru `Max_Payload_Size` všech zařízení po přenosové cestě. To znamená, že zařízení podporující vysoké MPS se musí podřídit systému s nižším MPS. Systémové MPS nastavení se získá během konfiguračního procesu. Každé PCIe zařízení má zapsanou hodnotu MPS v Device Capability registru, který se nachází v konfiguračním prostoru zařízení. Software si během konfigurace tento registr přečte u každého zařízení v systému a rozhodne, jaká hodnota MPS se bude používat. Vybraná hodnota MPS se zapíše do Device Capability registrů všech zařízení. Velikost MPS určuje, kolik TLP je nutných pro přenos daného množství dat. Jak se MPS zvyšuje, je třeba méně TLP pro přenos dat, a tím se zvýší efektivita přenosu daného paketu. Jak je vidět na obr. 3.8, zvyšování MPS nezvyšuje efektivitu lineárně.



Obr. 3.8: Závislost efektivity přenášeného paketu na jeho MPS

Hodnoty v grafu jsou vypočteny podle vzorce 3.2. Zobrazen je průběh pro nejmenší možný overhead 20 bajtů (3DW hlavička bez ECRC) a největší overhead 28 bajtů (4DW hlavička + ECRC).

$$EP(\%) = \frac{MPS}{MPS + Overhead} \cdot 100 \quad (3.2)$$

- *EP* - Efektivita paketu udává, kolik procent z teoretické šířky pásma lze využít pro přenos dat (maximální teoretická propustnost dat).
- *MPS* - Udává množství užitečných dat v TLP.
- *Overhead* - Je tvořen pevnou částí o velikosti 8 bajtů (fyzická a linková vrstva), proměnné velikosti závislé na použité adresaci, a zda je použito ECRC (12, 16 bajtů adresace + 4 bajty ECRC).

Během konfigurace je nastavena i maximální velikost čtení v řídicím registru (angl. maximum read request size (MRRS)). Tento parametr udává, o kolik dat si může jiné zařízení požádat. Nastavit lze maximálně 4096 bajtů (nižší hodnoty musí být násobky 128 bajtů) [6]. Nastavená hodnota může přesahovat MPS. Čtecí požadavek 512 bajtů může být poslán na zařízení, které má 128 bajtové MPS. Zařízení, které požadavek zpracuje, limituje Completion TLP na 128 bajtů a odešle těchto completion paketů více. Čtecí požadavek je 100% overhead, protože neobsahuje užitečná data. Čtení 64 kB dat při užití MRRS 128 bajtů znamená, že se vyšle 512 Memory Read TLP jen pro požadavky. Pro efektivní přenosy by množství požadovaných dat mělo odpovídat MRRS, aby se omezil počet čtecích požadavků.

Read Completion Boundary (RCB) umožňuje obsluhu jednoho čtecího požadavku několika completion pakety. Nastavení RCB udává, do jak velkých částí se data rozdělí (nejčastěji 64 nebo 128 bajtů).

Šířka linky	Zápis (MB/s)	Čtení (MB/s)
1	430	380
2	860	760
4	1720	1520
8	3440	3040

Tab. 3.3: Teoretická datová propustnost pro PCIe Gen2 (MPS = 128 B, RCB = 64 B)

Například pro MPS 128 bajtů je efektivita přenosu každého paketu 86%. Tabulka 3.3 byla vypočítána pro PCIe Gen2, kde teoretická propustnost je 500 MB/s na jeden přenosový pár (poloviční duplex). Pro výpočet zápisové propustnosti byl použit předpoklad MPS 128 bajtů. Pro čtení bylo počítáno s RCB 64 bajtů (76% efektivita) a byl použit předpoklad, že se požadavek splní vždy s násobkem 64 bajtů. Prvotní latence byla u čtení zanedbána.

3.5.1.2 Odhad reálné propustnosti systému

Podle uvedených faktorů je možné učinit odhad propustnosti systému. Propustnost je celkové množství přenesených dat za celkovou dobou přenosu.

$$\text{Propustnost (GB/s)} = \frac{\text{Transferred Data}}{\text{Transfer Time}} \quad (3.3)$$

Pro odhad propustnosti zařízení, které provede 300 zápisových transakcí (posted) přes PCIe Gen2 x4 s 32-bitovou adresací a MPS 256 bajtů počítáme s těmito parametry:

- 5 Gb/s přenosová rychlost. 1 bajt je tedy přenesen každé 2 ns (Symbol Time). Na lince jsou 4 přenosové páry, takže každé 2 ns lze odeslat 4 bajty.
- 32-bitová adresace bez ECRC znamená overhead 20 bajtů.

Z těchto parametrů lze vypočítat doba přenosu TLP a DLLP přenosu:

$$t_{TLP} = \frac{256 \text{ B payload} + 20 \text{ B overhead}}{4 \text{ B/clock}} \cdot 2 \text{ ns} = 138 \text{ ns} \quad (3.4)$$

$$t_{DLLP} = \frac{8 \text{ B}}{4 \text{ B/clock}} \cdot 2 \text{ ns} = 4 \text{ ns} \quad (3.5)$$

Aby bylo možné pokračovat ve výpočtu, musíme určit dobu přenosu. Pro odhad doby přenosu je nutné znát poměr vyslaných TLP ku Ack DLLP a FC DLLP. Tento poměr závisí mimo jiné na velikosti přijímacího bufferu. Pro další výpočet uvažuji 1 Ack DLLP každé 4 TLP a FC DLLP každé 3 TLP.

$$\textit{Transferred Data} = 300 \cdot 256 \text{ B} = 76800 \text{ B} \quad (3.6)$$

$$\textit{Transfer Time} = (300 \cdot 138 \text{ ns}) + (75 \cdot 2 \text{ ns}) + (100 \cdot 2 \text{ ns}) = 41750 \text{ ns} \quad (3.7)$$

Ze známé doby přenosu a přeneseného množství dat lze dosadit do rovnice 3.3.

$$\textit{Propustnost (GB/s)} = \frac{76800 \text{ B}}{41750 \text{ ns}} = 1.84 \text{ GB/s} \quad (3.8)$$

4

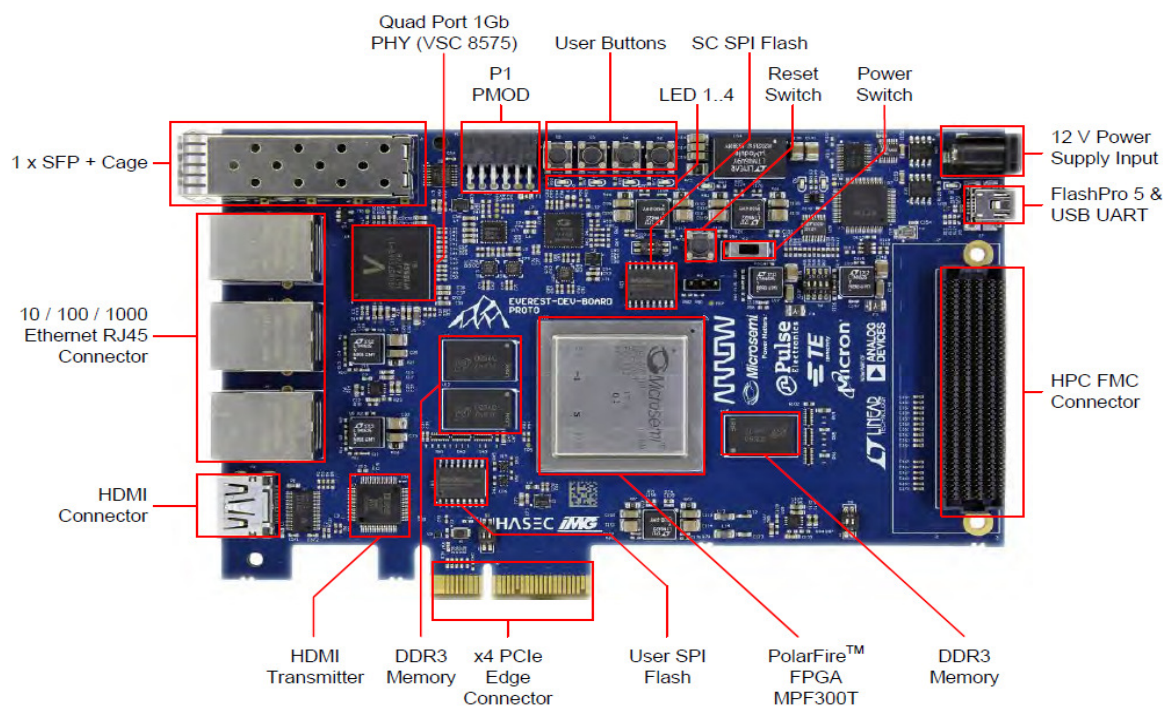
Realizace

Cílem diplomové práce je realizovat komunikaci mezi FPGA a PC s využitím rozhraní PCIe. Pro splnění tohoto úkolu bylo nutné zvolit vhodnou platformu pro realizaci s dostupným IP PCIe jádrem. V první části kapitoly je tato platforma popsána společně s dostupným IP PCIe jádrem. Dále je popsán protokol Advanced eXtensible Interface 4, jenž dané jádro využívá. Druhá polovina kapitoly se zabývá samotnou realizací systému v FPGA a užitým SW na straně PC.

4.1 Platforma pro realizaci

Pro realizaci praktické části diplomové práce je využita FPGA vývojová deska EVEREST-DEV-BOARD od společnosti Arrow Electronics. Deska využívá Polarfire FPGA čip MPF300TS-1FCG1152EES od Microsemi a obsahuje mnoho komponent pro usnadnění vývoje (obr. 4.1). V návrhu je využit PCIe konektor x4, paměť DDR3, led diody a tlačítka. Tento kit byl zvolen, protože parametricky vyhovuje požadavkům pro realizaci. Kit obsahuje PCIe konektor (x4), 3x RJ45 konektory (pro budoucí integraci do akvizičního systému). Dále dodávané firmwarové prostředí (Libero SoC) nabízí dostupné IP jádro pro PCIe. Softwarové nástroje využitě při realizaci DP:

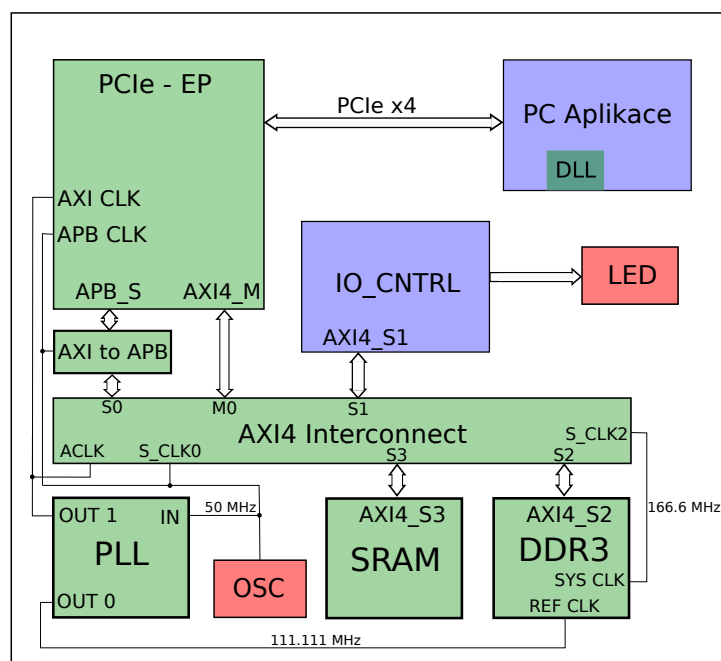
- Libero SoC v12.5 - Návrh, syntéza a nahrání realizovaného systému do FPGA.
- Visual Studio Code - Vývojové prostředí pro psaní VHDL kódu.
- Visual Studio 2019 - Vývojové prostředí pro psaní C++ kódu.
- ModelSim - Simulace VHDL kódu.



Obr. 4.1: Komponenty na vývojové desce Everest DEV Board [Převzato z [14]]

4.2 PCIe systém

Při návrhu byla využita kombinace vlastního kódu (psán v jazyce VHDL) a dostupných IP bloků. Navržený systém lze rozdělit na paměťovou část (DDR3, SRAM), ovládání I/O (led diody, čítače) a PCIe endpoint. Všechny tyto bloky jsou propojeny pomocí AXI4 Interconnect bloku. Bližší rozdělení je znázorněno v blokovém schématu na obr. 4.2. Dále lze na obrázku vidět rozdělení bloků na vlastní kód (fialová), IP bloky (zelená) a pevné komponenty (červená).



Obr. 4.2: Blokové schéma realizovaného systému

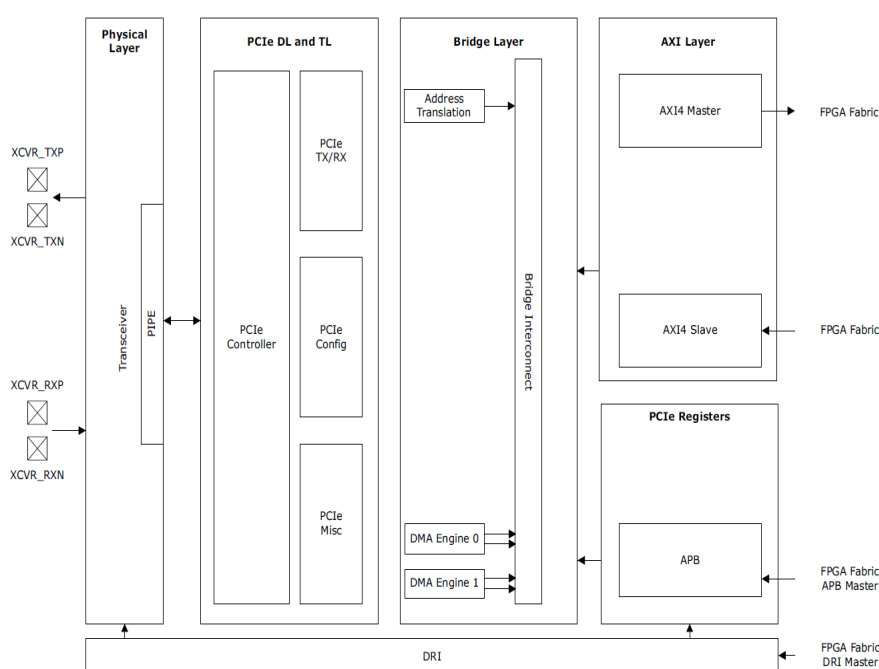
4.2.1 Advanced eXtensible Interface

K propojení jednotlivých bloků je využit protokol Advanced eXtensible Interface 4 (AXI4), který je součástí specifikace Arm Advanced Microcontroller Bus Architecture (AMBA). AXI4 realizuje spojení mezi master a slave blokem. AXI4 obsahuje 5 oddělených kanálů: Read Address, Write Address, Read Data, Write Data a Write Response. Každý kanál obsahuje mnoho různých signálů, ale všechny kanály obsahují valid a ready signály pro handshake. AXI4 lze rozdělit do tří kategorií: AXI4, AXI4-Lite a AXI4-Stream. AXI4-Lite má 5 oddělených kanálů stejně jako plný AXI4, ale na rozdíl od plného AXI4 (AXI4 podporuje 47 signálů) je počet signálů značně redukován (AXI4-Lite podporuje 21 signálů). Dále AXI4-Lite nepodporuje dávkové čtení či zápis (angl. burst read, burst write), zatímco AXI4 podporuje burst transakce do 256 datových přenosů. AXI4-Lite je vhodný pro jednoduchou komunikaci s nízkým nárokem na propustnost. AXI4-Stream protokol definuje pouze jeden kanál pro jednosměrný přenos dat. Tento protokol je vhodný pro

systémy pracující s daty, u kterých nezáleží na adresaci nebo nemá adresace smysl (např. FIFO).

4.2.2 PCIe jádro

Základem celého návrhu je hard IP PCI Express jádro obr. 4.3, které implementuje všechny tři přenosové vrstvy PCIe dle specifikace a podporuje přenosové rychlosti 2.5 Gbit/s (Gen1) a 5 Gbit/s (Gen2). K těmto přenosovým vrstvám je přidána Bridge vrstva, AXI4 vrstva a konfigurační vrstva. PCIe jádro podporuje šířku linky x1, x2 a x4¹. Podporovaná maximální velikost payloadu je 256 bajtů. V konfiguraci koncového zařízení (angl. endpoint) jádro podporuje až 6 BAR o velikosti 32 bitů nebo až 3 BAR o velikosti 64 bitů.



Obr. 4.3: Architektura PolarFire PCIe jádra

Bridge vrstva převádí PCIe adresu na AXI a naopak. Převod adres se provádí pomocí tabulky adres (ATR - Address table register). Vrstva obsahuje 6 ATR, které převádí adresu z PCIe bázeového adresového prostoru (BAR) na adresu pro AXI4 a 6 ATR pro převod adres z AXI4 do PCIe adresového prostoru. Každý 32-bitový BAR má vlastní ATR. Pokud je použit 64-bitový BAR, tak je využita jedna ATR a druhá je nevyužita (např. BAR01 využívá ATR0 a ATR1 je nevyužita). Adresu na kterou daná ATR převede PCIe adresu lze změnit zápisem do *TRSL_ADDR* registru.

Dále vrstva obsahuje dvě nezávislé DMA (DMA0 a DMA1). DMA0 má fixně nastavený

¹Každé PolarFire zařízení umožňuje vyžití dvou PCIe bloků (PCIe0 a PCIe1), tyto bloky sdílejí stejný x4 konektor. Je tedy možné užití dvou PCIe bloků s šířkou linky x1 a x2 nebo jednoho bloku s šířkou linky x4

zdroj dat jako PCIe linku s nastavitelnou cílovou destinací (slouží pro přenos dat z PC do FPGA). DMA1 má fixně nastavený zdroj dat jako AXI interface a jako fixní cílovou destinaci má PCIe interface (slouží pouze pro přenos dat z FPGA do PC). Obě DMA podporují dva módy přenosu. Mód přímého přenosu (angl. direct transfer mode), kdy DMA zapisuje/čte blok dat sekvenčně do kontinuálního bloku paměti. Počáteční DMA adresa se inkrementuje dle velikosti bloku dat. Druhý přenosový mód je scatter/gather. V tomto módu je zdrojová či cílová počáteční adresa ukazatel na zřetěžený list deskriptorů. Každý deskriptor obsahuje adresu a velikost bloku dat, společně s ukazatelem na následující deskriptor a umožňuje tak zápis do nekontinuálního bloku paměti.

AXI4 vrstva obsahuje AXI4 Master a Slave interface. Vrstva převádí AXI4 příkazy na PCIe pakety a PCIe požadavky na AXI4 příkazy. Velikost PCIe transakce se může měnit až do hodnoty MPS (maximum payload size) 256 bajtů. Šířka AXI4 Read/Write data signálů je 64 bitů a vrstva nepodporuje uživatelem definované signály v jednotlivých kanálech.

Konfigurační vrstva zajišťuje inicializaci konfiguračních registrů PCIe. Při zapnutí nebo restartu se do konfiguračních registrů nahraje nastavení zvolené při vytváření PCIe jádra v Libero SoC. Přístup k PCIe konfiguračním registrům je možný přes APB sběrnici.

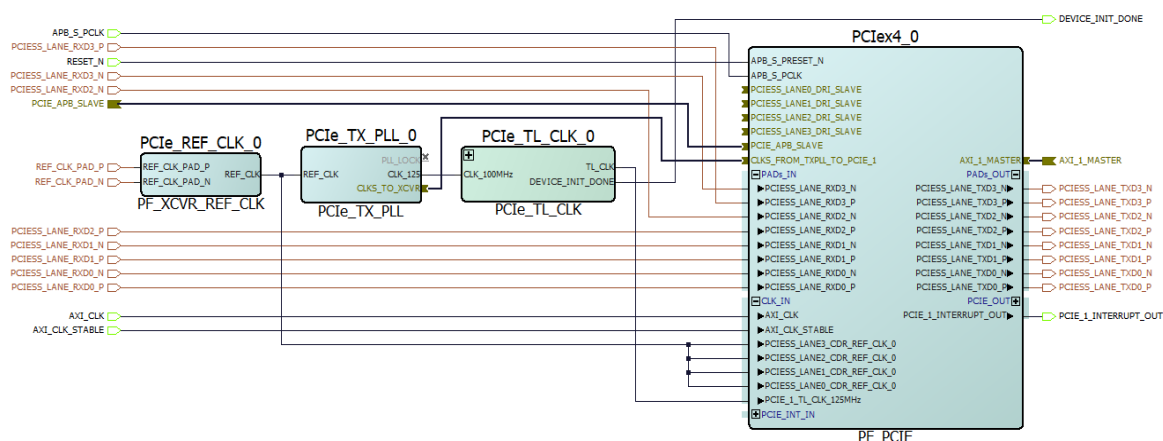
4.3 HW implementace

Přenos dat mezi PC a FPGA se provádí pomocí TLP paketů, kdy PC aplikace pošle TLP pro zápis do paměti (angl. memory write operation (MWr)) nebo TLP pro čtení z paměti (angl. memory read operation (MRd)). Tyto požadavky přijímá PCIe jádro a přemění je na AXI příkazy, které se dále zpracovávají v logických blocích realizovaného systému. Tato část kapitoly se věnuje popisu těchto logických bloků.

4.3.1 PCIe Endpoint subsystém

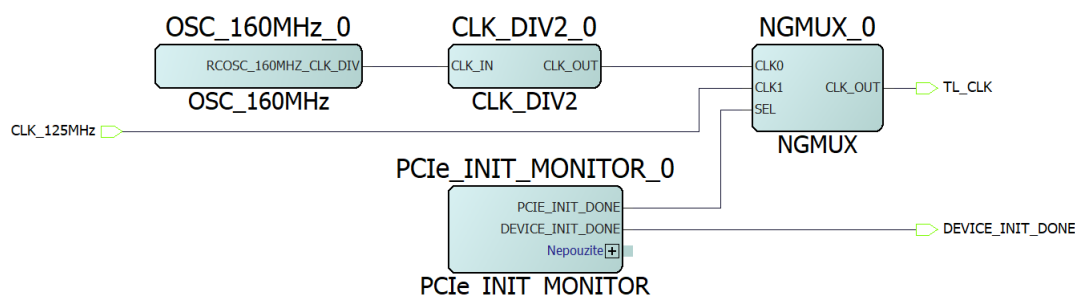
PCIe Endpoint subsystém (obr. 4.4) je tvořen výše popsáním PCIe jádrem a obvody zajišťující hodinový signál pro jádro. PCIe jádro je nastaveno jako PCIe EndPoint s přenosovou rychlostí 5 Gbit/s (Gen2) s šířkou linky x4. Referenční frekvence hodin je nastavená na 100 MHz. Při konfiguraci lze nastavit identifikační údaje o zařízení (Vendor ID, Device ID apod.). Tyto údaje jsou ponechány v defaultním nastavení. Jádro je konfigurováno jako AXI Master využívající dva 64 bitové BAR s povoleným předběžným načtením dat (angl. prefetchable memory). BAR0 slouží pro přístup ke konfiguračním a řídicím registrům PCIe jádra. Velikost adresového prostoru je nastavena na 64 kB. Počáteční adresa ATR0 je nastavena na 0x0300_0000. Přes BAR2 lze přistupovat k LSRAM a DDR3 paměti. Dále je přes tento BAR možný přístup k bloku IOCntrl. Velikost adresového prostoru je nastavena na 1 MB (zde je myšlen binární význam 1 MB = 1024 kB) a počáteční adresa ATR2 je nastavena na 0x1000_0000.

PCIe TL CLK slouží pro přívod referenčních hodin do jádra. Hodinový signál přenášený



Obr. 4.4: Zapojení PCIe Endpoint subsystému

po lince je v PolarFire zařízeních dostupný až po inicializaci. Z tohoto důvodu je nejdříve na referenční pin přivedeno 80 MHz a po dokončení inicializace se přepne na referenční hodiny přenášené po lince (100 MHz). Obvod realizující přepínání hodinového signálu je zobrazen na obr. 4.5.



Obr. 4.5: PCIe TL CLK subsystém

4.3.2 Reset a hodinové signály systému

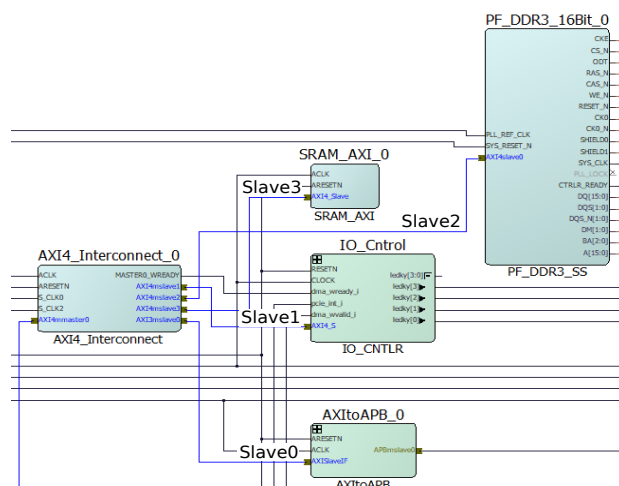
Systém obsahuje 3 zdroje hodinového signálu 50 MHz, 160 MHz oscilátor a referenční hodinový signál z PCIe linky. Hodinový signál z 160 MHz oscilátoru a PCIe linky je využit pro PCIE TL CLK, jak již bylo popsáno výše. 50 MHz oscilátor generuje vstupní signál do PLL, který generuje AXI CLK a REF CLK pro DDR3. AXI CLK² je zaveden do všech AXI bloků. DDR3 REF CLK má frekvenci 111,111 MHz a je využit pro PLL uvnitř DDR3 kontroléru.

Reset systému se provádí pomocí tlačítka S7 (pin T3). Signál z tlačítka je zaveden do bloku PF RESET, kde se dále rozvádí do reset vstupů ostatních bloků. Tento blok provádí resynchronaci resetu na AXI hodinový signál.

²Využitá frekvence AXI CLK je 200/250/307 MHz. Hodnota frekvence záleží na testované konfiguraci

4.3.3 Připojení bloků k PCIe jádru

PCIe jádro využívá pro přenos dat v FPGA AXI4 protokol. Pro připojení všech AXI slave bloků k PCIe AXI mastru je využito CoreAXI4Interconnect IP (dále jen interconnect). Interconnect je nakonfigurován pro propojení jednoho AXI master bloku a čtyř AXI slave bloků (obr. 4.6).



Obr. 4.6: Propojení AXI slave bloků s PCIe přes AXI4Interconnect

Pro všechny porty je šířka dat nastavená na 64 bitů a šířka adresy na 32 bitů. Konfigurace jednotlivých portů:

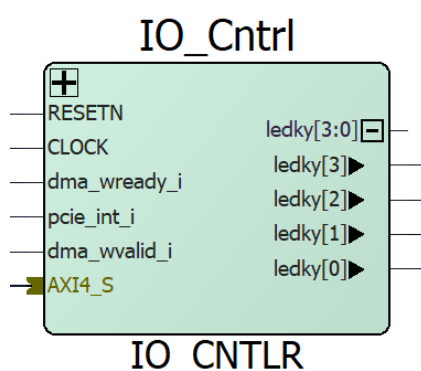
- Master0: PCIe jádro.
- Slave0 - Konfigurační a řídicí registry PCIe připojené přes AXItoAPB bridge. Rozsah adres (0x0000_0000 až 0x0FFF_FFFF).
- Slave1 - Ovládání led diod a vnitřní registry FPGA (IOCntl). Rozsah adres (0x1000_0000 až 0x1FFF_FFFF).
- Slave2 - AXI4 DDR3 kontrolér, který je připojen na externí 16 bitovou DDR3 paměť. Rozsah adres (0x2000_0000 až 0x2FFF_FFFF).
- Slave3 - AXI4 SRAM o velikosti 4 kB (4096 B). Rozsah adres (0x3000_0000 až 0x3FFF_FFFF).

Přístup ke konfiguračním registrům je realizován pomocí BAR0. Zadáním adresy (ofsetu) do tohoto prostoru je pomocí ATR0 PCIe adresa převedena na správnou AXI adresu. Pro přístup k ostatním slave blokům je využít BAR2, a tak pouhým zadáním ofsetu nelze přistoupit ke všem blokům. ATR2 je defaultně nastavená na počáteční adresu 0x1000_0000 a lze pomocí ofsetu provést přenos dat pouze se slave1. Proto se v případě potřeby přístupu k jinému slave bloku změní počáteční hodnota adresy, na kterou ATR2 převádí pomocí příslušného registru *TRSL_ADDR* (např 0x3000_0000). Po této změně

a vykonání transakce s daným blokem se znovu přepíše *TRSL_ADDR* registr na původní hodnotu 0x1000_0000.

4.3.4 Vnitřní registry FPGA

Vnitřní registry FPGA jsou realizovány blokem IOCtrl (obr. 4.7). Subsystem obsahuje dva 32 bitové čítače pro výpočet propustnosti, registry s hodnotou čítačů a frekvence. Dále je zde logika pro ovládání led diod. PC driver před DMA přenosem nejdříve nakonfiguruje daný DMA (zdrojová/cílová adresa, velikost dat) a během konfigurace nastaví i registry v IOCtrl pro výpočet propustnosti. Společně s inicializací DMA přenosu spustí driver i čítač pro dané DMA. Pro zastavení čítačů jsou využity signály *dma_wready_i*, *dma_wvalid_i* (pro DMA0) a *pcie_int_i* (pro DMA1). Po dokončení přenosu vyšle PCIe interrupt do driveru, a ten vyčte hodnoty registrů z IOCtrl. Blok je do systému připojen jako AXI slave s počáteční adresou 0x1000_0000. Ovládání led diod je přístupné přes offset 0x08. Registry obsahující parametry pro výpočet propustnosti jsou dostupné s offsetem 0x20 pro hodnoty čítačů a offsetem 0x50 pro hodnotu frekvence (registr obsahuje pevně danou hodnotu AXI CLK).



Obr. 4.7: IO Cntrl blok

4.4 SW realizace

Pro ověření funkčnosti je využit softwarový nástroj (C++ konzolová aplikace) využívající výrobcem dodaný driver a dll knihovnu. Úvodní obrazovka aplikace je na obr. 4.8. Pro ověření funkčnosti byla využita aplikace PolarFirePCIEGUI. Aplikace byla zkompileována a otestována v systému Windows 10 Home verze 2004. Pro testovací software bylo stanoveno několik hlavních funkcí:

- Vypis základních informací z konfiguračních registrů.
- DMA čtení/zápis libovolných dat do paměti (DDR3/LSRAM).
- Zápis a čtení do BAR prostoru (paměti, registry).

- Možnosti pro snadné ověření správné funkce (led diody, zápis/čtení vygenerovaných dat).

```
Farkas M. PCIe - DP app
-----
size of PCIState = 100
mpci_config_read success.
size of bytesReturned = 100
1. Device info
2. Read/Write do BAR prostoru
3. SG/C DMA
4. SGDMA test
5. LED
```

Obr. 4.8: Úvodní obrazovka konzolové aplikace

4.4.1 Device info

První volbou při spuštění aplikace je výpis základních informací o zařízení (Device info). Výpis těchto informací je zobrazen na obr. 4.9. Aplikace získává tyto informace z konfiguračních registrů PCIe jádra, vnitřních registrů (typ zařízení) a ze samotného driveru (driver info).

```
-----
size of PCIState = 100
mpci_config_read success.
size of bytesReturned = 100
firmware = DP - PCIe
device_status = 0x11AA
device_type = PolarFire Evaluation kit
driver_timestamp = 03:13:01 14/11/2017
driver_version = 6.1.7600.16385
supported_width = x4 (4 lanes)
n_width = x4 (4 lanes)
supported_speed = 5 Gbps (Gen 2)
n_speed = 5 Gbps (Gen 2)
num_bar = 2
bar0_adresa = fbfef00c
bar0_size = 10000
bar2_adresa = fbd0000c
bar2_size = 100000
```

Obr. 4.9: Výpis základních informací o zařízení

Základní informace z konfiguračních registrů:

- *supported_width* - maximální podporovaná šířka linky
- *n_width* - dohodnutá šířka linky
- *supported_speed* - maximální podporovaná rychlost linky
- *n_speed* - dohodnutá rychlost linky
- *bar_num* - počet dostupných BAR v FPGA
- *bar_size* - velikost daného BAR prostoru
- *bar_adresa* - počáteční adresa daného BAR prostoru

Podporovaná (angl. supported) rychlost a šířka linky udává, jakou konfiguraci zařízení podporuje. Ovšem to neznamená, že na dané rychlosti či šířce opravdu operuje. Rychlost a šířka linky s kterou bude zařízení komunikovat udává dohodnutá (angl. negotiated) šířka a rychlost linky. Konfigurační registry obsahují mnohem více údajů, ale ty už jsou celkem specifické. Pokud je třeba je znát, lze je vyčíst přes BAR0 MRd s adresou požadovaného údaje.

4.4.2 Zápis/čtení do BAR

Další funkce, kterou jsem od systému požadoval byla možnost zápisu/čtení dat z libovolného místa BAR prostoru (paměť, registry). Pro snadnou obsluhu je zápis do libovolného BAR zvlášť od volby pro zápis/čtení z paměťové části (volba 1 a 2 viz. obr. 4.10). Parametrem pro tuto volbu je požadovaná cílová paměť a offset. Možnost 3 a 4 umožňuje čtení/zápis z BAR prostoru. Tato volba vyžaduje typ BAR prostoru (BAR0/BAR2), adresní offset a v případě zápisu i hodnotu dat k zapsání. K pamětím lze přistoupit i přes možnost 3 a 4, ale před zápisem do požadované paměti je nejdříve nutno přepsat hodnotu *TRSL_ADDR* registru pro ATR2 (defaultní hodnota 0x1000_0000) v prostoru BAR0.

```

1 Read from memory
2 Write to memory
3 Read from bars space
4 Write to bars space
2
provide type 1 DDR, 2 LSRAM
2
Provide bar space offset
0x0
Provide the data to write
0xA0A

1 Read from memory
2 Write to memory
3 Read from bars space
4 Write to bars space
1
provide type 1 DDR-3, 2 LSRAM
2
Enter the offset
0x0
read value = 0xA0A

```

Obr. 4.10: Zápis dat do LSRAM (vlevo) a následné vyčtení dat ze stejného místa v LSRAM (vpravo)

Pro ovládání led diod lze použít zápis do BAR2 nebo využít funkci LED. Tato funkce má předvyplněný parametry BAR a offset, a proto stačí pouze zapsat hodnotu dat (např. 0 pro vypnutí všech led diod).

4.4.3 Zápis/čtení pomocí DMA

Poslední požadovaná funkce systému byla přenos většího množství dat pomocí DMA. Pro přenos dat z FPGA slouží DMA1 a pro přenos dat do FPGA slouží DMA0. DMA, jehož se bude přenos dat týkat, je inicializován pomocí driveru z PC. Tato inicializace obsahuje zdrojovou a cílovou adresu, typ DMA přenosu (kontinuální nebo scatter-gather) a velikost přenášených dat. DMA ukončí přenos, pokud vyčítaná/zapisovaná data dosáhnou nastavené velikosti. Po ukončení přenosu vyšle interrupt do PC driveru. V aplikaci je podporován zápis generovaných dat (např. samé 1 nebo 0), zápis vlastních dat a čtení dat (obr.4.11). Generovaná data jsou určena pro testování propustnosti a ověření správného

přenosu. Parametry pro tento mód jsou velikost generovaných dat a typ generovaných dat. Zápis vlastních dat vyžaduje pouze samotná data a pro čtení je nutné znát kolik dat se bude číst. Funkce SGDMA test slouží pro rychlý test propustnosti. Inicializuje se read/write SGDMA operace do DDR3 o velikosti dat 64 kB. Hodnota dat je generována inkrementálně od 0.

```

----- DMA write/read do DDR3/LSRAM -----
----- Gen. data write -----
1. Continuous Dma write do DDR3
2. Continuous Dma write do LSRAM
3. Continuous Dma write/read DDR3
4. Continuous Dma write/read LSRAM
5. SG Dma write do DDR3
6. SG Dma write/read DDR3
----- read data -----
7. Continuous Dma read z LSRAM
8. Continuous Dma read z DDR3
9. SG Dma read z DDR3
----- Data write -----
10. Data Dma write do DDR3/LSRAM
11. Data SG Dma write do DDR3/LSRAM

```

Obr. 4.11: Možnosti DMA přenosů PC aplikace

4.5 Integrace do akvizičního systému

Důvodem pro řešení diplomové práce je ověření přenosu dat mezi FPGA a PC pro možné využití v akvizičním systému (dále DAQ). Pro použití v DAQ systémech nelze PCIe přímo porovnávat s ostatními PC rozhraními, protože spíše než aby si tyto rozhraní konkurovaly, tak se doplňují. PCIe rozhraní bylo navrženo pro přímé připojení³ do systému. DAQ systém využívající PCIe by stále potřeboval využít některé jiné rozhraní pro přenos dat ze sensorů. Výhoda PCIe je jeho celková propustnost. Bylo by tedy možné vytvořit rozšiřující kartu, ke které by se připojilo několik sensorů (např. po Gigabit Ethernetu) a data z těchto sensorů by se poté posílala do PC přes PCIe. Testovací aplikaci by nahradil DAQ software (např. j-Pix [18]), který by využíval dll knihovnu k PCIe driveru. Dll knihovna může být upravena dle požadavků daného softwaru nebo FPGA systému.

Rozšiřující karta pro senzory využívající Ethernet by mohla přijímat data z několika sensorů. Tyto data by se do PC poslala pomocí PCIe. Oproti síťové kartě byl mohl systém v FPGA data před odesláním upravit. Toto předzpracování by mohlo probíhat paralelně pro každý sensor a ulehčit tak PC, který by dále data zpracovával.

³Je možné užití prodlužovacích kabelů. Pro PCIe Gen 2,3 lze použít kvalitní kabely pro prodloužení kolem 1 m. Pro PCIe Gen 5 se s prodlužovacími kabely nepočítá.

5

Naměřené výsledky

Vývojový kit je připojen do PC se základní deskou GA-P67A-D3-B3 (podpora PCIe Gen2) pomocí prodlužovacího kabelu (pro snazší manipulaci). Propustnost systému byla měřena pro tři různé hodnoty AXI CLK: 200 MHz, 250 MHz a 307 MHz. Pro napájení vývojového kitu je využívána PCIe linka. Nakonfigurovaná LSRAM má velikost 4 kB. Pokud je DMA zápis větší než 4 kB, dojde k přepsání dat v LSRAM (např. 32 kB přepíše celou LSRAM 8x). Velikost zápisu je omezena na 1 MB (dáno nastavenou velikostí v BAR2).

V této kapitole je použit binární i SI význam pro předpony. Binární je použit v souvislosti s velikostí paměti a velikostí přenášených dat. Pro hodnoty propustnosti je využíván SI standard.

5.1 Postup měření

Pro měření propustnosti slouží blok IOCntrl, jenž byl popsán v předchozí kapitole. Dva 32-bitové čítače, jenž jsou určeny pro výpočet propustnosti, jsou inicializovány společně s příslušným DMA. Před spuštěním je daný čítač vynulován. Pro zastavení DMA1 slouží signál *pcie_int_i*. PCIe jádro tento signál nastaví do log. '1' pokud došlo k ukončení DMA1 přenosu. Při log. '1' na *pcie_int_i* se čítač pro DMA1 zastaví. Během inicializace je do IOCntrl zapsána i velikost DMA přenosu. Této informace využívá čítač pro DMA0. Pro zastavení sleduje signály *dma_wready_i* a *dma_wvalid_i*. Pokud jsou oba signály v log. '1', tak DMA0 transakce stále probíhá a počítá se počet transakcí. Čítač pozná konec DMA0 přenosu, když se počet transakcí rovná nastavené hodnotě velikosti přenosu DMA0. Po dokončení přenosu vyšle PCIe interrupt do driveru a ten si vyčte hodnoty registrů (CLK_COUNT, CLK_FREQ) pro daný DMA přenos z IOCntrl. Pro výpočet propustnosti je využita rovnice 5.1.

$$Propustnost(MB/s) = \frac{TS \cdot CLK_FREQ}{CLK_NUM} \quad (5.1)$$

- *TS* - Udává množství přenesených dat (angl. Transfer Size) v bajtech.
- *CLK_FREQ* - Udává vnitřní frekvenci FPGA systému (AXI CLK). Pro dosažení výsledné propustnosti v MB/s se frekvence dosazuje v MHz. Hodnotu frekvence lze vyčíst pomocí offsetu 0x50.
- *CLK_NUM* - Udává počet hodinových cyklů od spuštění DMA přenosu. Počet cyklů pro zápis je dostupný s offsetem 0x20 a pro čtení je dostupný s offsetem 0x24.

5.1.1 Naměřené hodnoty

Propustnost byla měřena pro paměť LSRAM a DDR3. Pro přenos dat mezi pamětí LSRAM a PC byl využit kontinuální přenosový mód DMA. Paměť DDR3 byla testována s kontinuálním i SG DMA módem. Výsledné naměřené hodnoty propustnosti jsou zobrazeny v tabulkách 5.1, 5.2 a 5.3. Tyto hodnoty jsou výsledkem průměru z 10 měření. Při měření byl vývojový kit zapojen přímo do PC pro vyloučení vlivu prodlužovacího kabelu. Ukázka jednoho měření je na obr. 5.1

```
15997  15998  15999  16000
Read throughput = 1474 MBPS
local_dma.throughput_tx = 1233
local_dma.throughput_rx = 1474
```

Obr. 5.1: Ukázka měření propustnosti (LSRAM 64 kB DMA přenos, 250 MHz AXI CLK)

AXI CLK	Velikost přenosu	PC ->LSRAM	LSRAM ->PC
[MHz]	[kB]	Propustnost [MB/s]	Propustnost [MB/s]
200	4	800	1065
	32	1120	1173
	64	1156	1187
250	4	868	1293
	32	1124	1434
	64	1230	1474
307	4	906	1534
	32	1224	1614
	64	1257	1646

Tab. 5.1: Naměřené hodnoty propustnosti pro LSRAM (kontinuální DMA)

AXI CLK	Velikost přenosu	PC ->DDR3	DDR3 ->PC
[MHz]	[kB]	Propustnost [MB/s]	Propustnost [MB/s]
200	4	605	418
	32	761	425
	64	778	426
250	4	620	429
	32	788	455
	64	804	457
307	4	638	433
	32	819	456
	64	836	458

Tab. 5.2: Naměřené hodnoty propustnosti pro DDR3 (kontinuální DMA)

AXI CLK	Velikost přenosu	PC ->DDR3	DDR3 ->PC
[MHz]	[kB]	Propustnost [MB/s]	Propustnost [MB/s]
200	4	494	365
	64	763	421
	256	785	425
	1000	791	428
250	4	506	367
	64	789	451
	256	812	456
	1000	818	459
307	4	518	370
	64	821	452
	256	846	458
	1000	852	460

Tab. 5.3: Naměřené hodnoty propustnosti pro DDR3 (SG DMA)

5.2 Vyhodnocení výsledků

Pro zhodnocení výsledků je důležitá informace o maximální teoretické propustnosti systému. Pro PCIe Gen2 x4 je udávána hodnota 2 GB/s (pro jednosměrný přenos dat). Tato hodnota ale počítá jen s 20% overheadem, který vzniká 8b/10b kódováním. Pro určení maximální teoretické hodnoty je však důležitý i overhead přidáný jednotlivými přenosovými vrstvami PCIe. Pro určení tohoto overheadu je nutné znát množství užitečných dat v TLP (MPS), velikost read completion boundary (RCB), šířku adresace a zda je použit end-to-end cyklický redundantní součet (ECRC). Navržený systém má 256 bajtů MPS, 128 bajtů RCB, 64-bitovou adresaci a nepoužívá ECRC. Z těchto údajů vyplývá overhead 24 bajtů.

$$EP(\%) = \frac{MPS}{MPS + Overhead} \cdot 100 \quad (5.2)$$

Z rovnice 5.2 lze vypočítat efektivitu systému pro zápis a čtení dat. Po dosazení parametrů do rovnice vyjde efektivita 91,4% pro zápis a 84,2% pro čtení (do rovnice se místo MPS dosadilo RCB). Z toho plyne maximální teoretická propustnost pro zápis 1828,6 MB/s a 1684,2 MB/s pro čtení. Propustnost čtení dále snižují completion pakety. Naměřené hodnoty obsahují jak PCIe overhead, tak AXI4 a DMA overhead. Přenos DMA0 (PC -> LSRAM) je složen ze čtecích (MRd) a completion paketů, zatímco přenos DMA1 (LSRAM -> PC) je uskutečněn pomocí zápisových paketů (MWr).

Z naměřených hodnot je patrné, že zvyšování AXI CLK zvyšuje výslednou propustnost. Dále je vidět, že s rostoucí velikostí DMA přenosu nelineárně roste propustnost. Naměřená propustnost LSRAM odpovídá předpokladu, že přenos DMA1 (LSRAM -> PC) má vyšší propustnost než přenos DMA0 (PC -> LSRAM). Nejvyšší dosažená propustnost je při AXI CLK = 307 MHz a velikosti DMA přenosu 64 kB. Hodnota 1646 MB/s dosahuje 90% z maximální teoretické propustnosti zápisu pro danou konfiguraci. Pro čtení bylo dosaženo 74,6% z teoretické hodnoty (1257 MB/s). Odchylka od teoretické hodnoty je o něco větší než očekávaná. Pravděpodobně je zapříčiněna zanedbáním completion paketů. Naměřené hodnoty pro DDR3 danému předpokladu neodpovídají. Přenos DMA0 má vyšší propustnost než přenos DMA1. Naměřené hodnoty propustnosti jsou v porovnání s teoretickými poměrně nízké. Nejlepších výsledků bylo dosaženo při AXI CLK = 307 MHz a velikosti DMA přenosu 1 MB. Propustnost pro čtení dosáhla 50,6% z teoretické hodnoty a pro zápis dosáhla 25,16% z teoretické hodnoty. Pro nízké hodnoty propustnosti lze v dokumentaci najít vysvětlení. Nízké hodnoty jsou zapříčiněny limitací v bloku CoreAXI4Interconnect z důvodu křížení hodinových domén (CDC).

6

Závěr

V úvodu diplomové práce byl proveden teoretický rozbor PC rozhraní USB, Ethernet, Firewire a Thunderbolt. Dále byl proveden rozbor PCI Express (PCIe) rozhraní. Integrace do akvizičního systému byla diskutována ve 4. kapitole.

Systém byl realizován na FPGA vývojové desce EVEREST-DEV-BOARD od společnosti Arrow Electronics. Vývojová deska využívá Polarfire FPGA čip MPF300TS-1FCG1152EES od Microsemi. Pro realizaci systému byla využita kombinace dostupných IP bloků a vlastního kódu (psán v jazyce VHDL). Navržený systém byl popsán ve 4. kapitole. Systém podporuje přenos dat mezi pamětí FPGA (LSRAM, DDR3) a PC aplikací pomocí DMA přenosů. Dále tento systém podporuje zápis do registrů pomocí čtení/zápisu do BAR prostoru. Pro ověření funkce bylo realizováno ovládání led diod na vývojové desce.

Měření je prováděno pomocí bloku IOCtrl uvnitř FPGA systému. Měřené hodnoty propustnosti zahrnují overhead PCIe, DMA i AXI4 protokolu. Z naměřených hodnot vyšla nejlepší propustnost při zápisu dat z LSRAM paměti v FPGA do PC. Propustnost DMA zápisu byla 1646 MB/s. Této hodnoty bylo dosaženo DMA přenosem o velikosti 64 kB při frekvenci FPGA 307 MHz. Při stejné konfiguraci bylo dosaženo propustnosti 1257 MB/s pro čtení dat z PC do LSRAM. Z porovnání těchto výsledků s teoretickými hodnotami propustnosti (výpočet proveden v kapitole 5) lze říci, že hodnota 1646 MB/s dosahuje 90% z maximální teoretické propustnosti pro zápis. Hodnota 1257 MB/s dosahuje 74,6% z maximální teoretické propustnosti pro čtení.

Z naměřených výsledků vyplývá, že zvyšováním vnitřní frekvence FPGA se zvyšuje propustnost. Výsledky dále ukazují, že ke zvýšení propustnosti dochází i při zvyšování velikosti přenášených dat.

Výsledky řešení diplomové práce lze stručně shrnout do následujících bodů:

- Byl proveden rozbor běžně dostupných PC rozhraní.
- Bylo popsáno rozhraní PCI Express.
- Byla zvolena platforma pro realizaci PCI Express.
- Na dané platformě byl realizován PCIe systém a zprovozněna komunikace s PC.

- Na realizovaném systému bylo provedeno měření a byla provedena analýza výsledků.
- Byla diskutována možnost budoucí integrace v akvizčním systému.

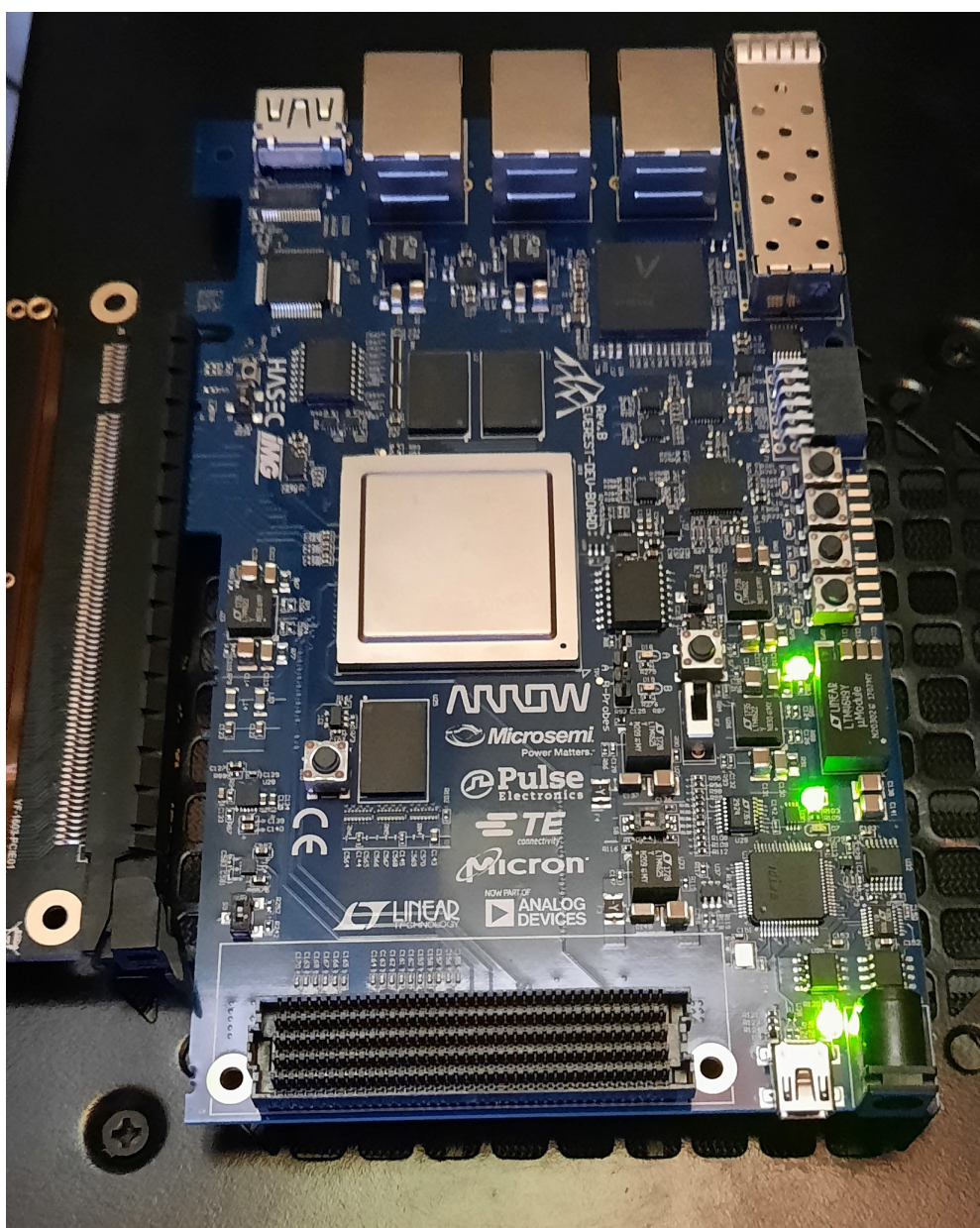
Bibliography

- [1] JACKSON, Mike a BUDRUK Ravi. *PCI Express Technology: Comprehensive Guide to Generations 1.x, 2.x and 3.0*. MindShare Press, 2012. ISBN: 978-0-9836465-2-5.
- [2] SOLOMON, Richard. *PCI Express Basics Background*. [online] [Cit. 23. 2. 2021] Dostupné z: https://pcisig.com/sites/default/files/files/PCI_Express_Basics_Background.pdf
- [3] STOKES, Jon. *PCI Express: An Overview* [online] [Cit. 18. 2. 2021] Dostupné z: <https://arstechnica.com/features/2004/07/pcie>
- [4] BLAKE, Bob. *Choosing the Right Programmable Logic Solution for PCI Express Applications* [online] [Cit. 19. 1. 2021] Dostupné z: <https://bit.ly/3p1nZA8>
- [5] PCI-SIG. *PCI Express Base Specification Revision 2.1*. [online] PCI-SIG, 2009.
- [6] LAWLEY, Jason. *Understanding Performance of PCI Express Systems* [online] [Cit. 27. 2. 2021] Dostupné z: https://www.xilinx.com/support/documentation/white_papers/wp350.pdf
- [7] Intel *An Introduction to SR-IOV Technology* [online] [Cit. 5. 2. 2021] Dostupné z: <http://www.intel.com/content/www/us/en/pci-express/pci-sig-sr-iov-primer-sr-iov-technology-paper.html>
- [8] IEEE 1394-2008. *IEEE Standard for a High-Performance Serial Bus* [online] [Cit. 2. 3. 2021] IEEE, 2008.
- [9] TEENER, J. Michael. *Technical Introduction to IEEE 1394* [online] Dostupné z: https://www.ieee802.org/802_tutorials/04-July/NewTechIntroTo1394.pdf
- [10] *Thunderbolt3 Technology Brief* [online] [Cit. 8. 3. 2021] Dostupné z: https://thunderbolttechnology.net/sites/default/files/Thunderbolt3_TechBrief_FINAL.pdf
- [11] KU, Andrew. *Everything You Need To Know About Thunderbolt* [online] [Cit. 8. 3. 2021] Dostupné z: <https://www.tomshardware.com/reviews/thunderbolt-performance-z77a-gd80,3205.html>

- [12] USB Implementers Forum. *Universal Serial Bus Specification Revision 2.0* [online] [Cit. 13. 3. 2021] Dostupné z: <https://www.usb.org/document-library/usb-20-specification>
- [13] PETERKA, Jiří. *Filosofie TCP/IP(II.)* [online][Cit. 5. 3. 2021] Dostupné z: <https://www.earchiv.cz/a96/a631k150.php3>
- [14] *Everest-DEV-Board User Guide* [online] [Cit. 14. 3. 2021] Dostupné z: <https://www.arrow.com/en/products/everest-dev-board/arrow-development-tools>
- [15] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. Praha: BEN - technická literatura, 2006. ISBN 80-7300-198-5.
- [16] *User Guide: PolarFire FPGA PCI Express* [online] Dostupné z: <https://www.microsemi.com>
- [17] *PCI Express High Performance Reference Design* [online] Dostupné z: <https://www.intel.com>
- [18] BROULÍM, Jan. aj. *j-Pix — A multiplatform acquisition package for Timepix3* JINST, 2019.

Appendix A

Připojení vývojového kitu do systému



Obr. A.1: Připojení kitu do PC systému pomocí prodlužovacího kabelu