VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZVYŠOVÁNÍ VÝKONNOSTI SÍŤOVÝCH APLIKACÍ

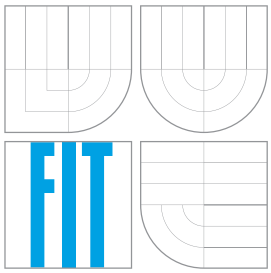
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN ŠTOURAČ

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZVYŠOVÁNÍ VÝKONNOSTI SÍŤOVÝCH APLIKACÍ

NETWORK APPLICATIONS ACCELERATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAN ŠTOURAČ
AUTHOR

VEDOUCÍ PRÁCE — Ing. VIKTOR PUŠ
SUPERVISOR

BRNO 2010

Abstrakt

Pod záštitou sdružení CESNET vznikla rodina karet COMBO s programovatelným hradlovým polem, jež je navržena pro akceleraci zpracování toku dat na vysokorychlostních sítích. Nad těmito kartami je postavena platforma NetCOPE, která má za cíl zrychlit a zjednodušit vývoj síťových aplikací. Tato práce se zabývá podrobnou analýzou datové propustnosti této platformy a navrhuje některá možná zlepšení, jež by mohla mít pozitivní vliv na celkový dosažený výkon.

Abstract

There has been developed family of cards called COMBO under the auspices of the CESNET association. These cards carry programmable field array and their purpose is intended into accelerating of net traffic. There is also a platform called NetCOPE which is based on these cards and it's purpose is to accelerate and simplify the development of network applications. This thesis contains a detailed analysis of data throughput through a whole platform and describes some possible improvements which should reflect in a better performance of the whole platform.

Klíčová slova

NetCOPE, Liberouter, programovatelný hardware, Ethernet, FrameLink, Interní sběrnice, propustnost sítí, paket.

Keywords

NetCOPE, Liberouter, programmable hardware, Ethernet, FrameLink, Internal Bus, nets throughput, packet.

Citace

Jan Štourač: Zvyšování výkonnosti síťových aplikací, bakalářská práce, Brno, FIT VUT v Brně, 2010

Zvyšování výkonnosti síťových aplikací

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Viktora Puše. Uvedl jsem všechny literální prameny a publikace, ze kterých jsem čerpal

.....
Jan Štourač
19. května 2010

Poděkování

Za odborné vedení, cenné rady a náměty při psaní této práce bych chtěl poděkovat Ing. Viktoru Pušovi.

Obsah

Kapitola 1

Úvod

S pokračujícím rozvojem informačních technologií jdou ruku v ruce rostoucí nároky na rychlost internetových služeb a s tím internetového připojení. Jak obyčejní uživatelé, tak především firmy, stále více využívají možností moderního Internetu. Jedná se například o internetovou telefonii, videohovory a videokonference přes Internet, cloud computing a další, především multimediální služby. Aby bylo možné vyhovět všem těmto požadavkům, je nutné zlepšovat a zrychlovat celou internetovou infrastrukturu a budovat silnější páteřní sítě, které poskytují dostatečný výkon. Jedná se jak o samotná fyzická média přenášející signál, síťové karty, schopné pracovat na rychlostech v řádech desítek Gb/s, tak i prvky sloužící pro správu sítě a její monitoring. Právě za posledním jmenovaným účelem vznikla v rámci výzkumné aktivity Liberouter sdružení CESNET platforma NetCOPE. Její účel je nabídnout nástroj, který dovolí uživateli (před)zpracovávat data ze sítě podle jeho konkrétních požadavků přímo na rychlosti linky. Což v případech, kdy se jedná o 10 Gb/s rychlosti a vyšší, by s obyčejným procesorem nebylo možné.

Příkladným využitím této platformy je projekt FlowMon, který je rovněž zaštiťován sdružením CESNET, a který slouží k monitorování sítě a vytváření statistik o síti. Dá se pomocí něj zjistit vytíženost linky, kudy data po síti putují, jak rychle jsou zpracovávána, délky toků a podobně. Se zjištěnými daty pak může administrátor sítě nakládat, jak aktuálně potřebuje. Může je využít například k identifikaci přetížených zařízení na síti nebo k detekci vzorů napadení.

Tato bakalářská práce obsahuje vysvětlení základních pojmů a nutný úvod do problematiky počítačových sítí, popis sběrnice PCI Express, stručný popis technologie FPGA, popis platformy NetCOPE a sběrnic, jež jsou v ní použity. Dále se zabývá podrobným měřením a analýzou jednotlivých sběrnic, využitých v platformě NetCOPE, především z pohledu jejich propustnosti a možnostmi dalšího zrychlení celé platformy.

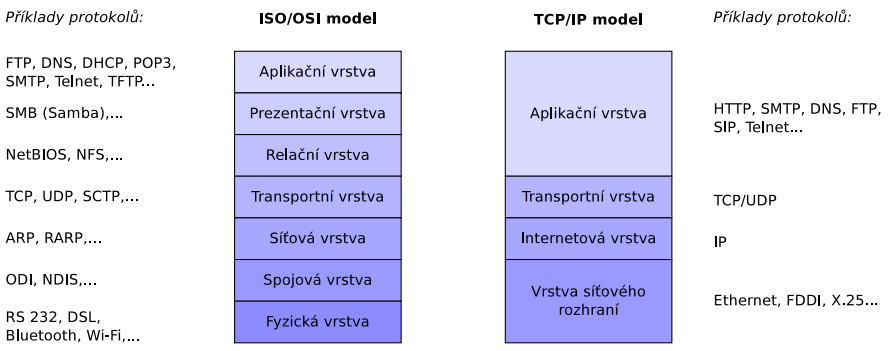
Kapitola 2

Teoretický rozbor

V první části této práce se zaměřím na vysvětlení základní teorie týkající se řešených problémů, na niž se budu v dalších částech odkazovat.

2.1 Referenční model ISO/OSI a TCP/IP

Pokud chceme komunikovat mezi počítači nebo obecně mezi nějakými dvěma zařízeními, je nutné je propojit tak, aby byla schopná předávat si vzájemně data. Komunikace mezi počítači, potažmo mezi jednotlivými aplikacemi, které na nich běží, není sama o sobě triviální záležitost. Je nutné zajistit uložení dat do vhodného formátu a přidat takové informace, aby data mohla doputovat k cíli, kde si je cílová stanice převezme a na základě přidaných informací rozpozná, jaké konkrétní aplikaci byla data adresována. Proto vzniknul model ISO/OSI [12]. Tento model vypracovala organizace ISO jako hlavní část snahy o standardizaci počítačových sítí nazvané OSI a v roce 1984 ho přijala jako mezinárodní normu ISO 7498. Jeho úlohou je poskytnout základnu pro vypracování norem pro účely propojování systémů. Norma nespecifikuje implementaci (realizaci) systémů, ale uvádí všeobecné principy sedmivrstvé síťové architektury. Popisuje vrstvy, jejich funkce a služby. Nejsou zde zařazeny žádné protokoly, které by vyžadovaly zbytečně mnoho detailů. V praxi se ale model ISO/OSI příliš neujal kvůli své složitosti. Používá se proto především jako referenční model. Naopak v rámci dnes nejrozšířenější počítačové sítě - Internetu - se rozšířil model TCP/IP [13] někdy také označovaný jako Internetový model. Na obrázku lze vidět popis vrstev obou modelů, jejich srovnání 2.1.



Obrázek 2.1: Srovnání modelů ISO/OSI a TCP/IP na jednotlivých úrovních.

2.2 Ethernet

Ethernet [4] a [8] je technologie, která se používá pro budování lokálních sítí (LAN). Představuje fyzickou a spojovou vrstvu v modelu ISO/OSI a vrstvu síťového rozhraní v modelu TCP/IP. V rámci LAN sítí je to velice rozšířený způsob propojení síťových zařízení, především díky jednoduchosti protokolu a jeho snadné implementovatelnosti.

2.2.1 Verze Ethernetu

Verze Ethernetu jsou následující [5]:

Ethernet	přenosová rychlost 10 Mb/s
Fast Ethernet	přenosová rychlost 100 Mb/s
Gigabit Ethernet	přenosová rychlost 1 Gb/s
10 Gigabit Ethernet	přenosová rychlost 10 Gb/s
40 Gigabit Ethernet a 100 Gigabit Ethernet	zatím neschválené standardy ve fázi návrhu

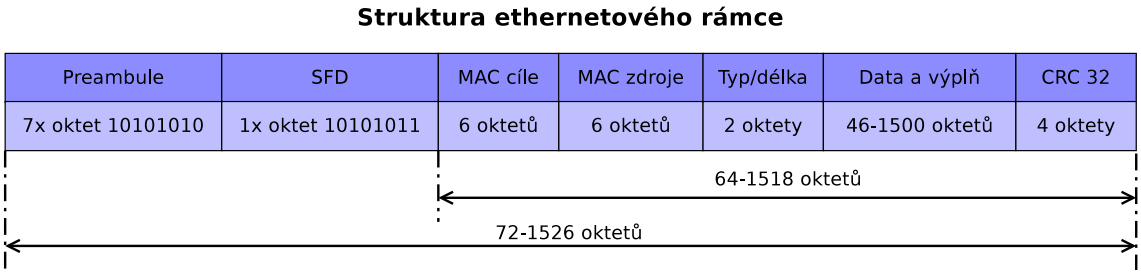
2.2.2 Komunikace v Ethernetu

Základní verze Ethernetu pracuje na rychlosti 10 Mb/s a používá sběrnicovou topologii. To znamená, že komunikace probíhá přes sdílené médium, kdy každé zařízení slyší všechno a vysílá se pouze tehdy, když nikdo jiný nevysílá. Znamená to ošetřovat případy, kdy se náhodou v jeden okamžik rozhodnou vysílat dvě zařízení. V této práci nás budou ale zajímat především sítě o rychlostech v řádech Gb/s, čili nadále se zaměřím na bližší popis gigabitového Ethernetu a 10 Gigabitového Ethernetu. Jejich hlavní rozdíl, kromě řádově vyšší rychlosti, spočívá v tom, že se u nich ustoupilo od klasické sběrnicové topologie a je možné komunikovat pouze typu half-duplex nebo full-duplex (u 10 Gigabitového Ethernetu dokonce již jen druhou ze zmíněných variant), čímž se omezila možnost, že dvě různá zařízení začnou vysílat v jeden okamžik.

Komunikace v Ethernetu probíhá v takzvaných rámcích (frames). Jsou to uspořádané sekvence dat, které jsou uvozeny takzvanou preambulí a SFD (viz 2.2) a ukončeny mezirámcovou mezerou, čímž jsme schopni rozlišit, kde jeden rámec začíná a kde končí. Aby bylo možné doručit posílané rámce správnému adresátovi, musí mít každé zařízení v Ethernetu vlastní adresu. Tato adresa se nazývá MAC (Media Access Control) adresa a je danému zařízení přiřazena již ve výrobě. Vždy když zařízení začne přijímat rámec, podívá se, jestli jsou určena pro něj. Pokud ne, tak daný rámec jednoduše zahodí. V opačném případě načte zbytek rámce, zkontroluje pomocí CRC kontrolního součtu správnost dat a ta pošle dál ke zpracování. Struktura rámce je vidět na obrázku 2.2. Pro popis velikosti jednotlivých bloků je využito slovo oktet, které znamená 8 bitů. Je to z toho důvodu, že na různých architekturách může být typ byte jiné délky než 8 a mohlo by tak dojít k nekompatibilitě.

2.2.3 Deficit idle count - DIC

V době, kdy se na ethernetové lince nevysílá, a mezi samotnými rámci musí být takzvaná mezirámcová mezera. Ta při plném vytížení linky musí být podle specifikace nejméně 12 oktetů. V případě pak, že se nevysílá, je mezirámcová mezera na lince stále. Deficit idle count je technika, která se využívá pro zefektivnění komunikace v Ethernetu. Umožňuje zkrátit mezeru mezi rámci v takových případech, kdy by vyšla zbytečně dlouhá a snižovala



Obrázek 2.2: Struktura ethernetového rámce. [4]

Popis polí:

Preambule	slouží k synchronizaci hodin příjemce
SFD	označení začátku rámce (Start of Frame Delimiter)
MAC cíle	MAC adresa cílového síťového rozhraní
MAC zdroje	MAC adresa zdrojového síťového rozhraní
Typ/délka	pro Ethernet II toto pole určuje typ vyššího protokolu pro IEEE 802.3 udává délku pole dat
Data a výplň	pole dlouhé minimálně 46 a maximálně 1500 oktetů, minimální délka je nutná pro správnou detekci kolizí v rámci segmentu, pokud data nemají potřebnou minimální délku (46 B), je nutné přidat nějakou výplň, aby se této hodnoty dosáhlo.
CRC32	kontrolní součet (Frame Check Sequence, FCS), počítá se ze všech polí s výjimkou preambule a FCS. Slouží k ověření správnosti dat.

by efektivitu komunikace, přičemž přesto zachovává průměrnou mezirámcovou mezeru v komunikaci na hodnotě 12 oktetů, jak je vyžadováno ve specifikaci. Konkrétně DIC dovoluje minimální mezirámcovou mezeru mezi rámci od 9 do 15 bytů.

2.3 Měření propustnosti v síťových zařízeních

Každé zařízení na síti je schopno pracovat určitou rychlostí. Často je však nutné zjistit u síťových zařízení, jakou rychlost linky jsou schopny zpracovávat, aby bylo možno určit, které z nich je nejužším článkem řetězce. Pro to, aby tato měření byla jednoznačná a pokud možno univerzální, vznikly definice, jež zavádí různé způsoby měření propustnosti v síťových zařízeních. V této práci využiji dva postupy, definované v RFC 2544 [1] a v RFC 2285 [2]. Jedná se o takzvaný Throughput test a FRMOL test.

2.3.1 Throughput test

Tento test je popsán v RFC 2544 a je založen na posílání rámců o určité velikosti danou rychlostí na testované zařízení a měří se počet zpracovaných rámců. Hledá se taková nejvyšší rychlost, na jaké dokáže zařízení zpracovat kompletně všechny testovací rámce, aniž by nějaký z nich zahodilo.

2.3.2 FRMOL test

FRMOL (Forwarding rate at maximum offered load) je narozdíl od předchozího testu založen na odesílání rámců určité délky nejvyšší možnou rychlostí, jakou linka nabízí, a následným zjištěním propustnosti vypočtením poměru rámců, jež bylo zařízení schopno zpracovat a jež naopak bylo nuceno kvůli vysoké vstupní rychlosti zahodit.

2.4 PCI Express

PCI Express [11] [14], označovaná také jako PCI-E nebo PCIe je standardní sériová sběrnice se stromovou topologií, sloužící ke komunikaci mezi periferními zařízeními v PC a pro připojení dalších rozšiřujících karet. Její první verze byla uvedena v roce 2004 a je to nástupce sběrnic PCI a PCI-X. Narozdíl od nich se ale jedná o sériovou sběrnici a ne paralelní. V každém směru vedou dva diferenciální vodiče (tato čtveřice tvoří jednu linku), čímž se výrazně snížila možnost ovlivnění šumem. Také se odstranil i dosavadní problém při zrychlování frekvencí minulých sběrnic - clock skew. To bylo způsobováno tím, že hodiny na samostatném vodiči nemusely díky rozdílným vlastnostem na vodiči hodin a vodičích dat dojít v ten samý okamžik a mohlo se stát, že přijatá data byla špatně vyhodnocena. Proto je využito vestavěných hodin (embedded clock), kdy synchronizační signál vede po stejném vodiči jako data a je do nich zakódován. Samotná data jsou posílána v paketech a pro jejich přenos je použito kódování 8/10 bitů. Další výhodou této sběrnice je možnost škálovatelnosti - pokud potřebujeme vyšší přenosovou rychlost, tak jednoduše zvýšíme počet linek. Počet použitých linek se pak uvádí za názvem sběrnice např. PCI-E x16 (především grafické karty), PCI-E x1, PCI-E x8, atd.

2.4.1 Verze PCI-E

1.0	původní verze, rychlost 250 MB/s na lane (obousměrně 500 MB/s)
2.0	přenosová rychlost zvýšena na 500 MB/s na lane (obousměrně 1000 MB/s)
2.1	zachovává rychlost předchozí verze, ale přidává některé prvky, které jsou připravovány pro verzi 3.0.
3.0	v době psaní této práce je připravovaný standard. Čeká se opět zvýšení rychlosti (1000 MB/s na lane), zpětná kompatibilita, náhrada kódování 8/10 bitů za 128/130 bitů, což sníží režii a další optimalizace.

2.5 FPGA

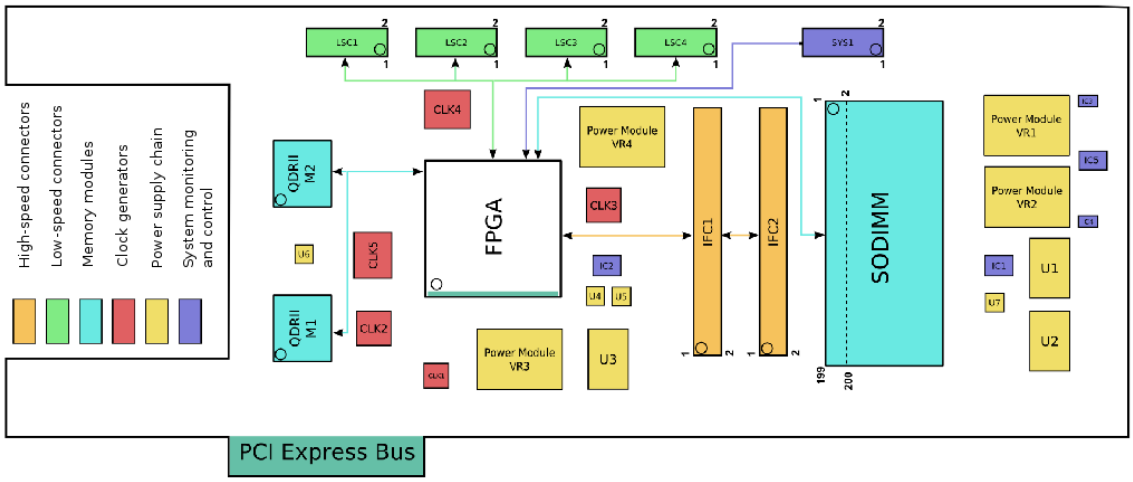
FPGA (Field-programmable gate array) jsou čipy, které obsahují programovatelná hradlová pole. Takový čip není z výroby omezen pouze k použití pro daný specifický účel, ale naopak díky programovatelné logice, kterou obsahuje, dokáže fungovat podle toho jakou konfiguraci do něj nahrajeme. Díky této vlastnosti se FPGA čipy hodí do zařízení, kde se mění potřeby na vykonávanou činnost daným zařízením. Popis funkce obvodu, do kterého se má čip nakonfigurovat, se pro takovéto čipy provádí v nějakém z HDL (hardware description language) jazyků jako například VHDL nebo Verilog. Z daného popisu se pomocí speciálních nástrojů vytvoří konfigurace (vysyntetizuje design), která se nahraje do čipu, čímž se uvnitř něj vytvoří propojení, vytvářející obvod, který provádí požadovanou činnost.

2.6 Rodina karet COMBO

Rodina karet COMBO [9] byla vyvinuta skupinou CESNET v rámci výzkumného projektu Liberouter. Hlavní myšlenkou bylo dát možnost vývojářům pracovat s takzvaným „open hardware“ a používat ho podobně jako se používá open-source software. Proto jádrem každé COMBO karty je jeden nebo více FPGA čipů. Dále disponují nezbytnými paměťmi, vstupně výstupními konektory a čipy a podobně. Díky FPGA čipům je možné během chvíle změnit funkčnost dané COMBO karty jen nahráním jiné verze konfigurace (designu) do čipu FPGA. Tímto je dosaženo, že jedna stejná karta může být použita pro více účelů a ve více projektech. Na stránkách www.liberouter.org v sekci Our Hardware je možné najít více detailů ke každé z dostupných karet. Původní rodina COMBO karet byla postavena na čipech Virtex II a Virtex II PRO firmy Xilinx. Novější karty rodiny COMBOv2 jsou pak založeny na FPGA čipech typu Virtex 5 rovněž od firmy Xilinx. Tato nová technologie dovolila zrychlení designu a zvětšila množství dostupných zdrojů na čipu. Dále budu popisovat pouze tuto novou rodinu karet.

2.6.1 Použití karet COMBOv2

Použitelný celek se většinou skládá ze dvou karet. Jedna takzvaná mateřská karta plus jedna add-on karta, která se připojí na kartu mateřskou. Mateřská deska 2.3 obsahuje čip FPGA, paměť, konektory a do PC se připojuje pomocí PCI-E x8. Karta může nést různé FPGA čipy, takže je možné si zvolit optimální desku podle potřeby aplikace (rychlost designu, velikost a podobně). Add-on karta pak obsahuje konektory pro připojení do sítě - například 4x SFP pro připojení na 4 gigabitové linky nebo 2x XFP pro připojení na 2 10 Gigabitové linky. Add-on karta tedy slouží pouze pro příjem či vysílání dat ze sítě/na síť.



Obrázek 2.3: Schéma mateřské karty COMBOv2. Obrázek převzat z [16]

2.7 FrameLink

FrameLink je protokol pro přenos dat ve formě rámců, který vychází z protokolu LocalLink od firmy Xilinx, je však oproti LocalLinku zjednodušen a podporuje pouze zarovnané

transakce. Informace o tomto protokolu jsem čerpal z privátních webových stránek projektu Liberouter [6].

2.7.1 Rozhraní protokolu

Struktura signálů tohoto protokolu je vidět v tabulce 2.1. Všechny signály, končící na „_N“ jsou aktivní v nule.

DATA	Přenášená data o šířce N*8 bitů
REM	Zbytek (remainder). Určuje kolik bytů z datové části je v aktuální transakci platných. Tento signál je platný pouze, pokud je aktivní EOP_N. Jeho bitová šířka je log(N). Například při datové šířce 32 b a hodnotě REM = „00“ je platný poze nultý bajt dat (0. - 7. bit). Při hodnotě REM = „10“ jsou pak platné spodní tři bajty (0. - 23. bit).
SOF_N	Start of frame, značí začátek rámce. Rámec samotný se pak může skládat z více částí (parts), které jsou ohraničeny pomocí SOP_N a EOP_N. Zároveň se SOF_N musí být vždy nastaven SOP_N první části rámce.
EOF_N	End of frame, značí konec rámce. Pokud je aktivní zároveň se SOF_N, pak je velikost rámce menší nebo rovna N bajtů. Vždy s tímto signálem musí být také nastaven EOP_N, který značí konec poslední části rámce.
SOP_N	Start of part, značí začátek části rámce. Každý rámec může obsahovat libovolné množství částí, většinou se však používají dvě nebo tři (header, payload, footer).
EOP_N	End of part, značí konec částí paketu, znamená „valid“ pro signál REM. Pokud je aktivní zároveň se SOP_N, je daná část menší nebo rovna N bajtů.
SRC_RDY_N	Source ready. Signalizuje připravenost vysílacího zařízení vysílat data - má nějaká data na odeslání a data jsou připravena na výstupu. Zároveň určuje platnost všech ostatních signálů.
DST_RDY_N	Destination ready. Zařízení, které přijímá data, dává tímto signálem najevo, že je schopno data přijmout. Přenos dat probíhá pouze když jsou oba SRC_RDY_N i DST_RDY_N aktivní.

Tabulka 2.1: Rozhraní protokolu FrameLink - výčet signálů a jejich funkce.

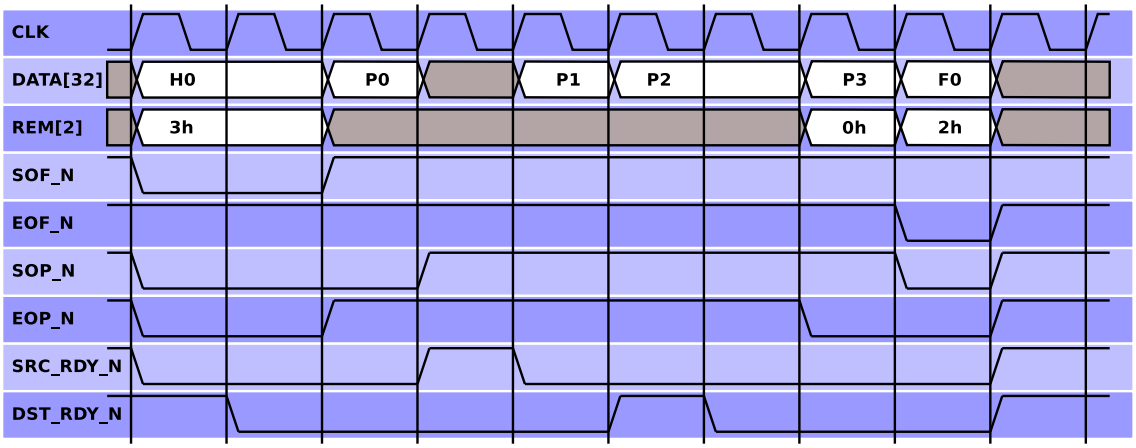
2.7.2 Struktura rámce

Každý rámec je ohraničen na začátku aktivním signálem SOF_N a na konci EOF_N. Dále je možné, aby se rámec skládal z více částí. V tom případě každá část musí být ohraničena navíc svojí dvojicí SOP_N a EOP_N. Jsou podporovány pouze zarovnané transakce, takže při aktivním signálu SOF_N a SOP_N je vždy nultý bajt dat na nulté pozici. Znamená to také, že v jednom slově dat nemohou být data ze dvou různých částí či dokonce rámců. Podporováno je neomezené množství částí v rámci, prakticky se ale používají maximálně tři části - header (hlavička), payload (náklad), footer (patička).

2.7.3 Přenos dat

Přenos dat (transakce) je řízen signály SRC_RDY_N a DST_RDY_N. Pokud jsou oba aktivní, je možné provést transakci. Pokud je alespoň jeden z nich neaktivní, je vložen čekací

stav. Vysílací komponenta připravuje data k odeslání na výstup a nechává je tam tak dlouho, dokud neproběhne přenos. Na obrázku 2.4 je příklad komunikace.



Obrázek 2.4: Příklad přenosu dat FrameLink protokolem.

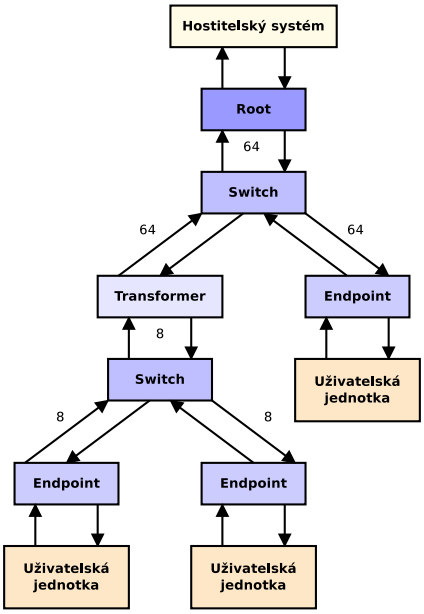
2.8 Internal Bus

Internal Bus, nebo-li Interní sběrnice [17], je sběrnicí, navrženou specielně pro použití v čipech FPGA. Její návrh vytvořil Ing. Tomáš Málek v rámci své magisterské práce na FIT VUT v Brně. Z jeho práce jsem čerpal informace a taktéž z privátní wiki projektu Liberouter [7].

2.8.1 Architektura

Interní sběrnice je navržena tak, aby dovolovala komunikaci jednotlivých zařízení (jednotek) na čipu FPGA mezi sebou a zároveň mezi interními komponentami a prvky hostitelského systému. Topologie systému je založená na stromové architektuře, která je vidět z obrázku 2.5. Na samotném vrcholu stojí kořenová komponenta (root), která je napojena na systémovou sběrnici hostitelského systému a na přepínací komponentu. Na konci každé přepínací komponenty (switche) může být další switch, či transformační komponenta, která se stará o převod transakcí mezi dvěma částmi Interní sběrnice s různými datovými šířkami, nebo přímo koncová komponenta (endpoint), starající se o převod rozhraní interní sběrnice na univerzální čtecí/zápisové rozhraní a pomocí níž se na Interní sběrnici připojují samotné jednotky na čipu. Interní sběrnice je tvořena duplexními směry (UPSTREAM LINK a DOWNSTREAM LINK). Po downstream lince přicházejí data od root komponenty k samotným jednotkám, jež jsou zapojeny na sběrnici. Upstream linka je naopak určena pro přenos dat ve směru od jednotek k root komponentě. Zajímavý je způsob připojení na upstream linku. Ta je provedena pomocí jednoduchého *or* spoje namísto třístavové sběrnice, což je mnohem jednodušší a rychlejší způsob (především k použití navrhovaného obvodu v FPGA čipu, kde by vytvoření třístavové sběrnice zabralo mnohem více zdrojů). Toto zjednodušení však klade jednu podmínku pro napojené jednotky, a to v podobě, že v okamžiku kdy daná jednotka nekomunikuje, musí na svých výstupech mít vždy hodnotu logické 1, aby došlo ke správnému vyhodnocení posílaných dat na upstream linku. Samostatná jednotka arbitru, v

případě připojených více jednotek, zde není nutná, jelikož samotná arbitrace probíhá přímo v každé switch kompononetě.



Obrázek 2.5: Popis architektury Interní sběrnice.

2.8.2 Rozhraní protokolu

Rozhraní protokolu je v podstatě shodné s protkolem FrameLinku 2.1, obejde se však bez signálů REM, SOP_N a EOP_N. Zbylé signály z protokolu FrameLinku jsou součástí protokolu Interní sběrnice a nabývají stejného významu jako v protokolu FrameLinku, proto je zde nebudu znovu vypisovat.

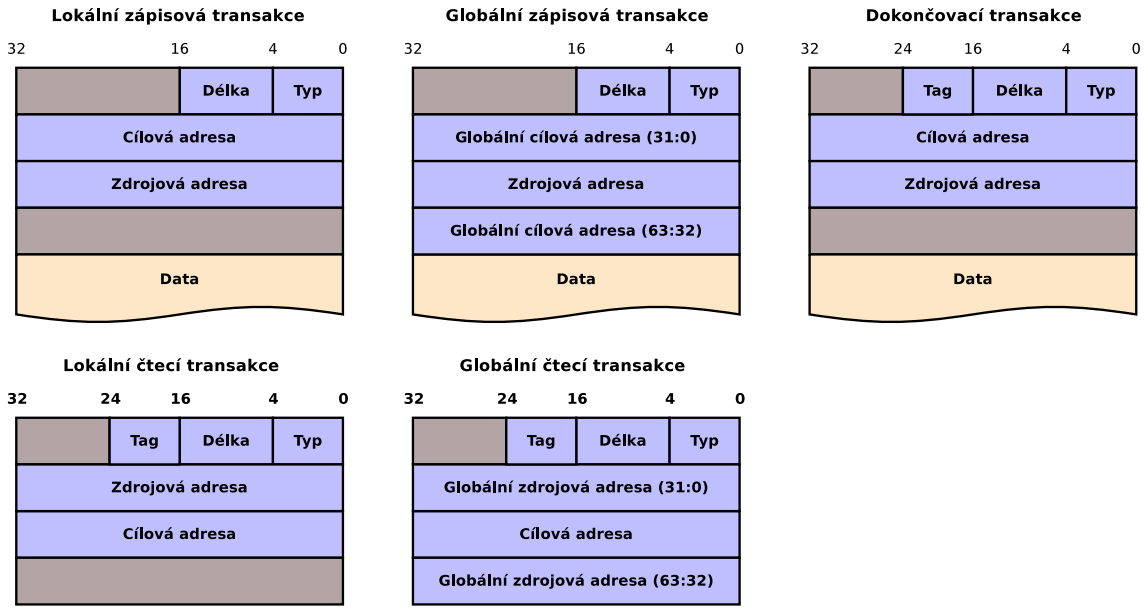
2.8.3 Typy transakcí a struktura rámce

Opět struktura rámce je podobná struktuře FrameLink rámce. Z důvodu absence signálu REM není možné určit, kolik bytů dat je v dané transakci platných, když je aktivní signál EOF_N. Tato informace lze získat z údaje z hlavičky. Vzhledem k tomu, že také oproti FrameLinku chybí signály SOP_N a EOP_N, není možné rozdělovat samotný rámec na jednotlivé části, a tedy rámec není nijak dělen. Dělení však probíhá v rámci jedné části - podle typu transakce je přenášený rámec členěn na hlavičku a případná užitečná data. Hlavička má délku 128 bitů a délka případných dat je omezena na 1 B až 4096 B. Existují tři základní typy transakcí:

- Zápisová
- Čtecí (požadavek na čtení)
- Dokončovací (odpověď, přečtená data)

Z těchto třech typů pouze čtecí transakce obsahuje jen hlavičku bez dat. Zbylé dvě obsahují hlavičku i data. Čtecí a dokončovací transakce pak navíc nesou i ideintifikační číslo

(tag), který je spojuje. Dále se transakce na Interní sběrnici rozdělují na lokální a globální. Lokální transakce jsou používány v případech, kdy spolu komunikují dvě jednotky přímo na čipu, kdežto globální transakce jsou naopak používány v situacích, kdy komunikuje jednotka na čipu s nadřazeným systémem. Tento druhý druh transakcí je nutný realizovat skrz platformově závislý bridge (například PCI-E bridge a podobně). Na obrázku 2.6 je ukázka složení rámců jednotlivých druhů transakcí a v tabulce C.3 je popsán význam jednotlivých polí.



Obrázek 2.6: Ukázky složení dat v rámcích protokolu Interní sběrnice.

2.8.4 Přenos dat

Zde se opět odkazuji na princip přenosu dat v rámci FrameLinku, s absencí rozdělení framu na části 2.4.

2.9 Local Bus

Local Bus (Lokální sběrnice) slouží pro konfiguraci jednotek na čipu. Jejím účelem není rychlá přeprava dat, ale především úspora zdrojů. To je důvod, proč není využita ke konfiguraci jednotek Interní sběrnice, jež je sice o poznání rychlejší, ale má vyšší požadavky na zdroje. Pro tento účel rychlost Lokální sběrnice plně dostačuje. Informace jsem čerpal z privátní wiki projektu Liberouter [10]

2.9.1 Architektura

Architektura Lokální sběrnice má podobně jako Interní sběrnice stromovou strukturu. Je složena z root komponenty, switche a endpoint komponent. Opět má duplexní propojení, rozdělené na UPSTREAM a DOWNSTREAM linky. Kořenový prvek inicializuje veškeré čtecí i zápisové transakce. Switch komponenty jsou narozdíl od switchů Interní sběrnice

Typ (4 bity)	Určuje, jaké vlastnosti přenášený paket má a jakou reprezentuje operaci.
Zdrojová a cílová adresa (32/64 bitů)	Udává, odkud a kam bude daný paket putovat po sběrnici. Globální adresa je 64 bitů široká, stejně jako prostor celého systému počítače. Lokální adresa je 32 bitů široká, stejně jako lokální adresový prostor propojovacího systému. V případě, kdy by tento prostor nebyl dostatečný, dají se rezervovat některé nevyužité bity hlavičky pro případné rozšíření lokální adresy.
Délka (12 bitů)	Udává počet bajtů užitečných dat, která jsou v paketu přenášena (v případě zápisové a dokončovací transakce) nebo která mají být přečtena (v případě čtecí transakce). Speciální hodnota 0x000 udává maximální délku paketu, která je 4096 B (což odpovídá jedné stránce paměti počítače).
Tag (8 bitů)	Reprezentuje příznak, který jednoznačně identifikuje konkrétní čtecí požadavek a k němu příslušející odpověď. Pro zápisové transakce nemá (na interní sběrnici) význam. V případě generování transakcí pomocí bus-master rozhraní endpointu má tag význam i u zápisové transakce - je využit pro indikaci úspěšného odeslání požadované transakce na sběrnici.

Tabulka 2.2: Popis významu jednotlivých polí v transakcích Interní sběrnice.

jednoduché a nedisponují arbitrační logikou. Jakákoliv transakce po DOWNSTREAM lince dorazí ke všem jednotkám zapojených na dané sběrnici. Předpokládá se, že na čtecí transakci odpoví pouze jedna jednotka a její odpověď pošle zpět prostřednictvím switchů po UPSTREAM lince. Pokud by na čtecí transakci reagovalo více jednotek, došlo k chybě (překrytí adresových prostorů), a switch nastaví signál určující chybu, jež propaguje do root komponenty, kde je tato chyba zalogována v registru.

2.9.2 Rozhraní protokolu

Výčet signálů rozhraní a jejich popis lze vidět v tabulce 2.3

DWR	Datová část o šířce 16 bitů, na niž se zapisují data, která chceme odeslat konkrétní jednotce. Směr od root komponenty níž po stromu sběrnice.
BE	Bitová maska o šířce 2 bity, označující validitu přenášených dat. Každý bit určuje validitu jednoho bajtu. Pokud je příslušný bit nastaven na logickou úroveň 1, je odpovídající bajt platný.
DRD	Datová část o šířce 16 bitů, ze které se čtou data při požadavku na čtení. Směr k root komponentě.
ADS	Address strobe. Značí, že na portu DWR je vystavena adresa, na kterou se bude zapisovat nebo ze které se bude číst.
WR	Značí požadavek na zápis dat.
RD	Značí požadavek na čtení dat.
RDY	Definuje okamžik, kdy jsou požadovaná data připravena ke čtení.
ERR	Pokud nastane problém v komunikaci (např. více zařízení reaguje na stejnou adresu) je aktivován tento signál.
ABORT	Slouží ke zrušení aktuálně prováděné transakce.

Tabulka 2.3: Rozhraní Lokální sběrnice - výčet signálů a jejich funkce.

2.9.3 Typy transakcí

Na lokální sběrnici existují dvě transakce: čtecí a zápisová. Popíši je z pohledu root komponenty. Zápisová operace začíná vystavením adresy, na kterou se má zapisovat, na port DWR a aktivací signálu ADS. Po odeslání cílové adresy se postupně na port DWR vystavují odesílaná data. Současně s nimi je nastaven i signál WR a příslušné BE signály. Čtecí operace probíhá v prvním kroku stejně - vystavením adresy na port DWR a aktivací signálu ADS. Dále se však liší. Když chceme přečíst slovo dat, je nutné nastavit signál RD a s tím i BE signály. Jakmile je nastaven RDY, znamená to, že endpoint komponenta zpracovala náš požadavek a můžeme si data vyzvednout na portu DRD.

Pozastavování přenosu dat na lokální sběrnici je řešeno poněkud odlišně od Interní sběrnice i FrameLinku. Na rozdíl od nich nedisponuje lokální sběrnice žádnými signály, které by reprezentovaly připravenost jednotky data vysílat nebo přijímat. Je to tedy řešeno tak, že root komponenta vždy vystaví omezený počet (16) čtecích/zapisovacích transakcí a dále čeká, dokud nedostane RDY signál ze strany endpointu. Pozastavení z pohledu root komponenty je ošetřeno tak, že root komponenta si sama hlídá to, aby odeslala pouze takový počet požadavků, který je schopna zpracovat.

Kapitola 3

Vývojová platforma NetCOPE

Jak již bylo zmíněno v úvodu, platforma NetCOPE byla navržena pro urychlení a usnadnění vývoje síťových aplikací, akcelerovaných pomocí čipů FPGA. Jako nosný hardware jsou využity karty COMBO. Vlastní zjednodušení spočívá ve využití abstraktní vrstvy na čipu FPGA, která odstiňuje nízkoúrovňové hardwarově závislé vlastnosti vybrané hardwarové karty s FPGA čipem a poskytne snadný přístup k jejím zdrojům, jako jsou síťové rozhraní, přípojení k hostitelskému počítači, kontroléry pamětí a podobně. Daná aplikace tuto abstraktní vrstvu uchopí a použije, čímž se může rovnou zaměřit pouze na specifické funkce, jež implementuje. První aplikací, která vždy demonstruje funkčnost platformy je síťová karta (Network Interface Card - NIC). Na obrázku níže 3.1 jsou vidět základní části, ze kterých je platforma složena a především hlavní tok dat skrze celou platformu.

3.1 Základní seznámení s architekturou

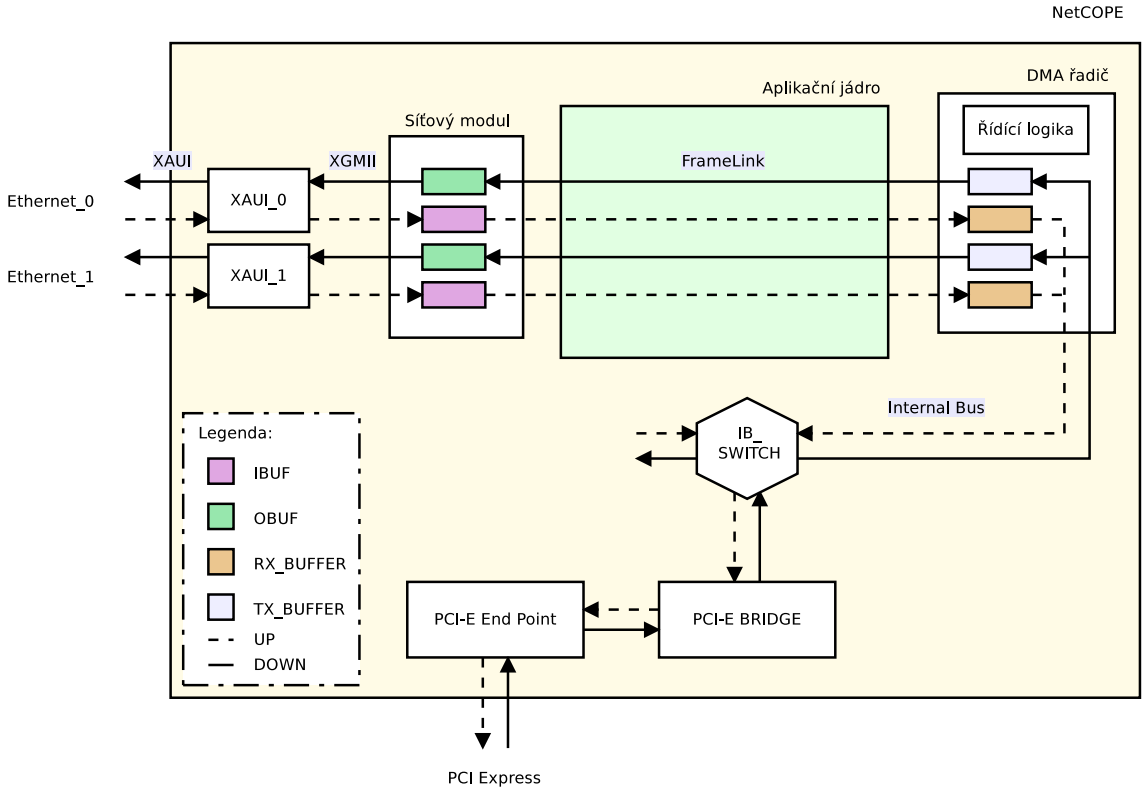
V centru výše zmíněného obrázku je vyhrazeno zelené místo pro aplikační jádro. Zde se očekává umístění specifické aplikace nad NetCOPE platformou postavené. Samozřejmě v případě potřeby mohou vývojáři příslušné aplikace upravit i další části. Toto je však nejjednodušší cesta a jsou odstíněni od ostatních specifických vlastností čipu. Architektura dále zahrnuje síťové porty a síťový modul, jež se starají o příjem a odesílání dat z/do karty. Původní data z Ethernetu, nesená na rozhraní XAUI, jsou tak postupně převedena na rozhraní XGMII a posléze FrameLink. Odtud putují přes aplikační jádro do DMA řadiče. Ten řídí přenos dat z karty do paměti příslušného počítače a naopak. K samotnému přenosu mezi kartou a pamětí hostitelského systému slouží sběrnice PCI Express, pro jejíž obsluhu jsou na čipu jednotky PCI-E bridge a PCI-E End Point. PCI-E End Point je IP core firmy Xilinx, jejíž FPGA čipy COMBO karty nesou.

3.2 Identifikace slabých míst

V této části se zaměřím na podrobnou analýzu využitých komunikačních protokolů na kartě, identifikaci možných omezení, jež skýtají a nakonec se pokusím navrhnout možná zlepšení.

3.2.1 Režie v Ethernetu

Cílem platformy NetCOPE je schopnost zpracovat všechna užitečná data přicházející ze sítě. Neznamená to však, že pokud je linka 10 Gb/s Ethernetu vytížena na maximum, musí nutně



Obrázek 3.1: Jednoduché schéma platformy NetCOPE zaměřené především na tok dat.

zpracovat všech 10 Gb každou vteřinu. Každý komunikační protokol obsahuje nějaká režijní data. Pro představu je níže graf 3.2 efektivní datové propustnosti na 10 Gb/s ethernetové lince v závislosti na délce ethernetového rámce. Efektivními daty myslím samotná data bez ethernetové hlavičky, CRC i preambule a SFD. Výpočet hodnot jsem provedl podle tohoto vzorce:

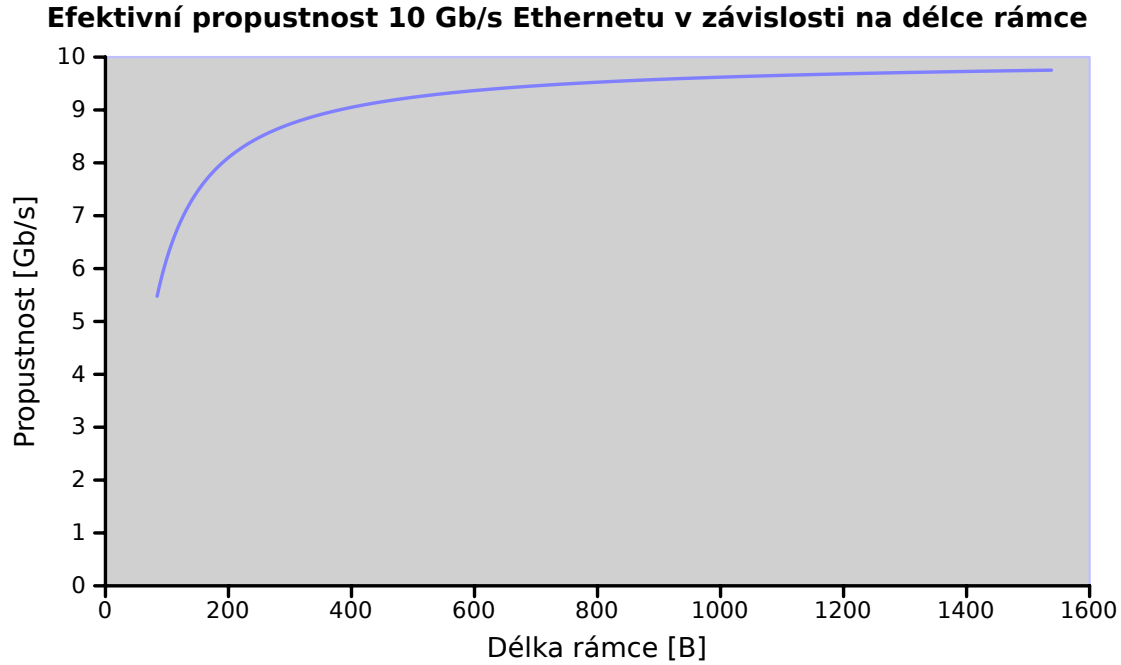
$$\frac{10 \cdot D}{H + D + M},$$

kde *10* znamená rychlost linky v Gb/s, H zastupuje hlavičku v Ethernetu (celý rámec bez vlastních dat, včetně CRC) plus nutnou preambuli a SFD - celkově tedy 26 oktetů. D je množství přenášených dat (46 až 1500 oktetů) a M zastupuje mezirámcovou mezeru (bráno pevných 12 oktetů). Výsledek rovnice je v jednotkách Gb/s, což je rychlost přenosu čistých dat při dané délce rámce (datové části v rámci). Z grafu je vidět, že efektivita komunikace je nejlepší při naximální délce rámce. Je to způsobeno tím, že je nutné posílat nejméně režijních dat (hlaviček rámce, preambulí, SFD a mezirámcových mezer) než v případě odesílání krátkých paketů.

3.2.2 Režie protokolů FrameLink a Internal Bus

Podobným způsobem jako v předchozím případě zde chci ukázat režii v komunikačních protokolech FrameLink a Interní sběrnici, využitých v platformě NetCOPE v závislosti na délkách přenášených dat.

Jelikož oba protokoly mají velice podobnou strukturu a zároveň oba dva nepodporují nezarovnané transakce, jejich režie je v případě použití stejné délky hlavičky totožná.

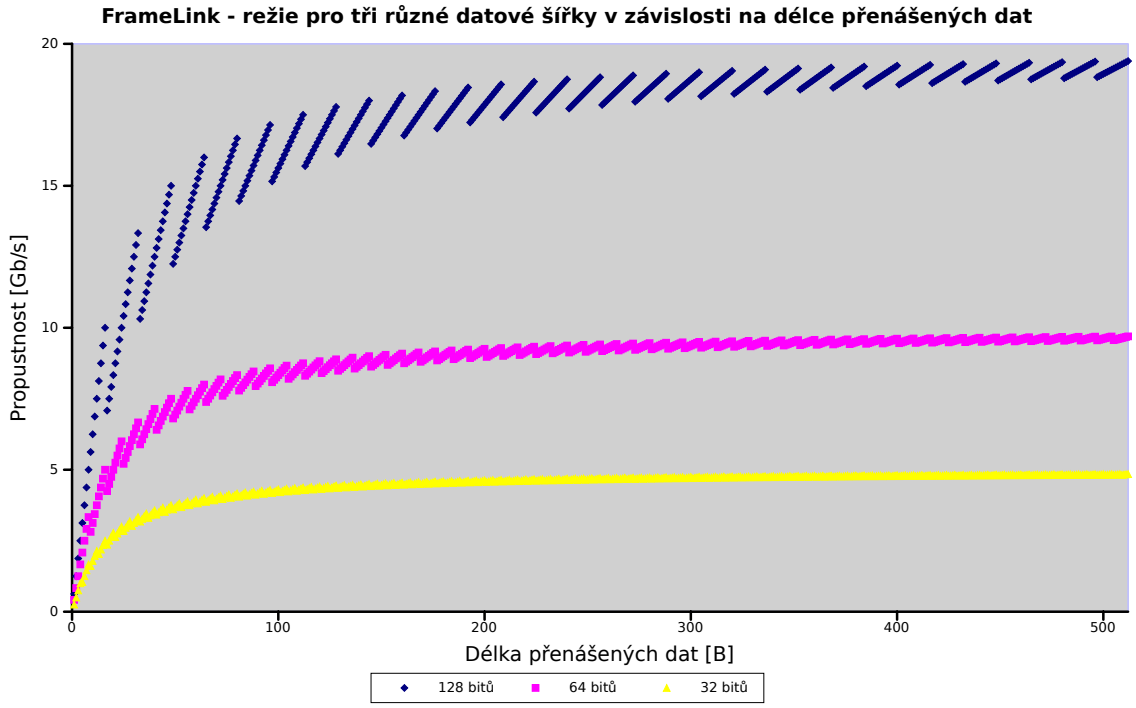


Obrázek 3.2: Graf efektivní datové propustnosti na 10 Gb/s ethernetové lince v závislosti na délce ethernetového rámce.

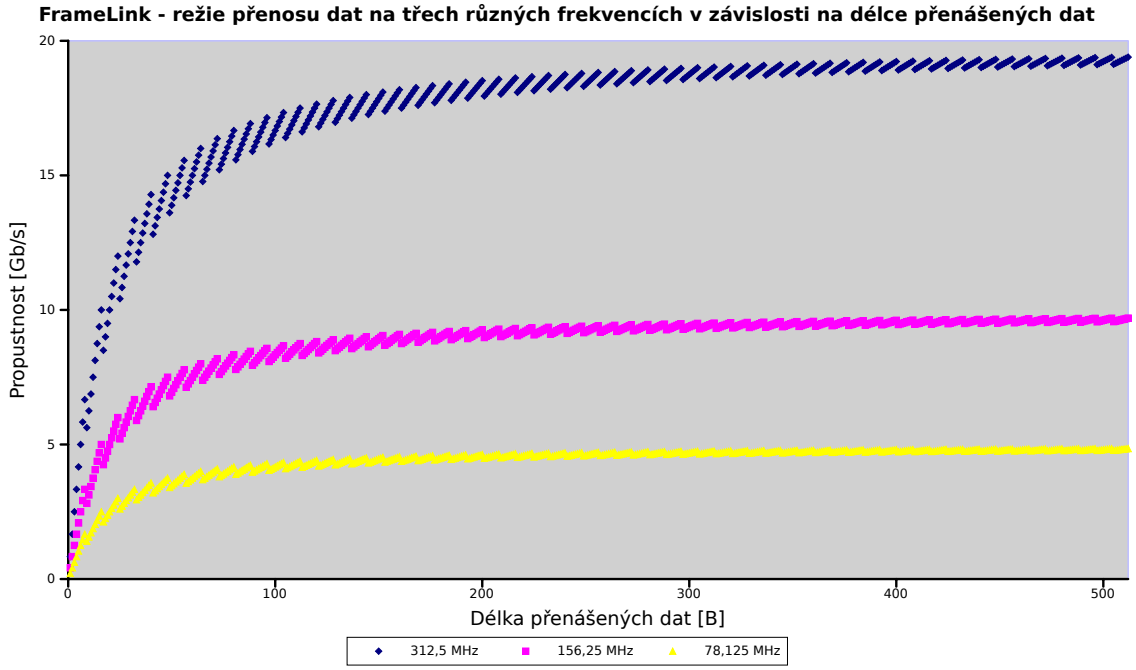
V protokolu FrameLink je velikost hlavičky v přenášeném rámci libovolná - lze ji určit podle potřeby. Rovněž délka jednoho rámce může být libovolně dlouhá. Oproti tomu je v Interní sběrnici velikost hlavičky určena napevno na hodnotu 128 bitů a maximální délka jedné transakce na 4096 B. V případě sběrnice FrameLink může být tedy díky možnosti přenášet oproti Interní sběrnici delší rámce jeho režie nižší.

Jednou z možností, jež lze na obou protokolech změnit, je jejich datová šířka. Na obrázku 3.3 jsou vidět grafy pro tři různé datové šířky těchto protokolů - 32, 64 a 128 bitů. Na první pohled jsou na grafu patrné zuby, které se u režie Ethernetu nevyskytovaly. Důvodem je to, že data jsou po ethernetové lince posílána sériově, kdežto v těchto sběrnicích jsou data posílána paralelně a díky absenci podpory nezarovnaných transakcí tak dochází ke skokovým úbytkům/nárůstům v režii. Lze jednoduše poznat, že čím vyšší datová šířka je použita, tím větší skoky v režii nastávají. Vždy totiž, když odesíláme data o velikosti různé jak $W*n$, kde W je datová šířka a n je počet slov, je platnými daty vždy poslední slovo obsazeno pouze z části. Proto čím větší datovou šířku zvolíme, tím větší část slova může zůstat nevyužita. V případě úvah o přechodu na sběrnici s větší datovou šířkou je tedy nutné tuto negativní vlastnot brát v potaz.

Další možnou úpravou je změna frekvence, na které dané protokoly běží. V takovém případě nedochází ke změně režie. Zvýší se pouze datová propustnost, přičemž velikost režie zůstává zachována. Lze to vidět na dalším obrázku 3.4, kde je znázorněna režie obou protokolů na různých frekvencích a stejných datových šířkách i o stejně dlouhých hlavičkách. Zvyšování frekvence je nejjednodušší možností zvýšení propustnosti daného protokolu. Ovšem její dosažení však již tak jednoduché nemusí být. V případě FPGA čipů se jedná o problémy toho typu, že nástroje pro syntézu nejsou schopné najít konfiguraci takovou, jež by splnila potřebné podmínky, aby byl výsledný design funkční. Je to způsobeno především



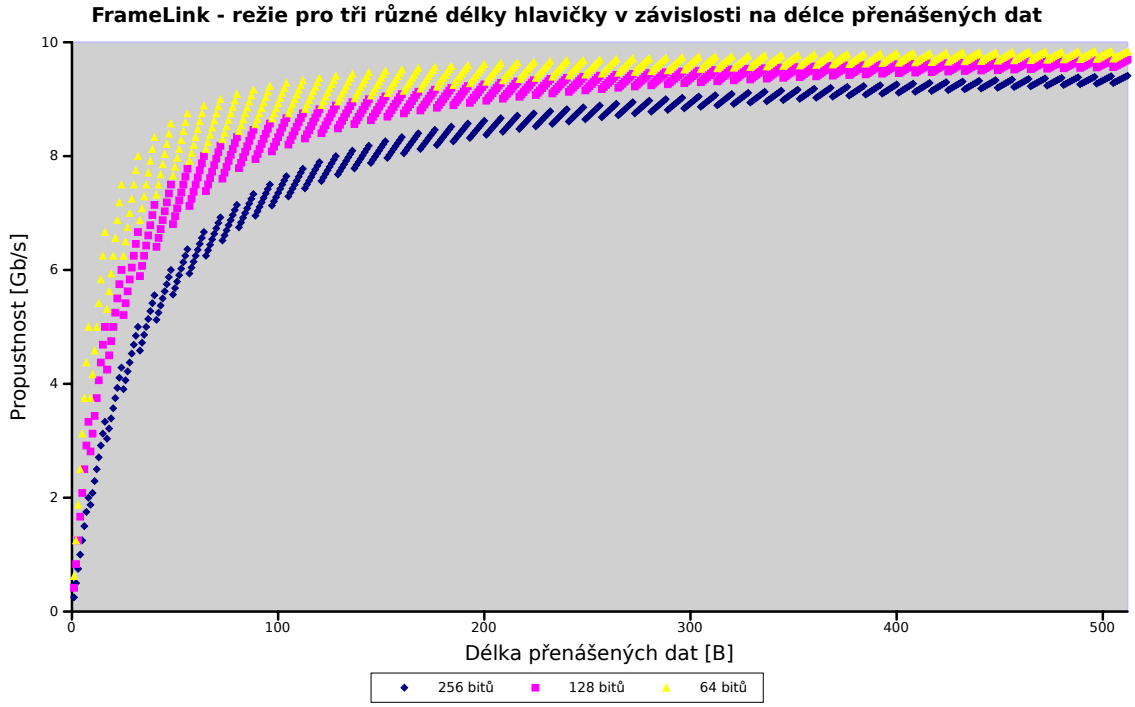
Obrázek 3.3: Graf efektivní datové propustnosti sběrnic FrameLink s hlavičkou 128 bitů a Internal Bus na frekvenci 156,25 MHz a třech různých datových šířkách v závislosti na velikosti odesílaných dat.



Obrázek 3.4: Graf efektivní datové propustnosti sběrnic FrameLink s hlavičkou 128 bitů a Internal Bus na třech různých frekvencích - 78,125, 156,25 a 312,5 MHz o stejné datové šířce (64 bitů) v závislosti na velikosti odesílaných dat.

složitostí vytvářeného designu.

U protokolu FrameLink lze mimo předchozích dvou modifikací ovlivnit i velikost hlavičky. Tato úprava má přímý vliv na velikost režie v protokolu. V dalším grafu 3.5 je ukázána efektivní datová propustnost protokolu FrameLink s využitím různě dlouhých hlaviček - 64, 128 a 256 bitů na datové šířce 64 bitů. Je zde patrné, že čím delší hlavičku protokol obsahuje, tím větší režie je přítomna a naopak. Délka hlavičky by vždy měla být minimální možná, ale zároveň musí obsahovat všechny potřebné údaje, aby nesená data mohla být v pořádku doručena. Někdy se do hlavičky přidávají doplňující informace k samotným neseným datům, například číslo vstupního portu, kterým data do karty přišla, časová značka a podobně.



Obrázek 3.5: Graf efektivní datové propustnosti sběrnice FrameLink na frekvenci 156,25 MHz a datové šířce 64 bitů s třemi různými délkami hlaviček v závislosti na velikosti odesílaných dat.

Rychlost komunikace na určitém protokolu je v zásadě ovlivněna těmito faktory: frekvencí, datovou šířkou přenášeného slova, možností podpory nezarovnaných transakcí a samozřejmě velikostí hlavičky a dalších metadat (režie). Z výše uvedených grafů lze vypozorovat, jaký vliv má na určitý protokol změna příslušných parametrů. Pro zvýšení datové propustnosti bez změny režie stačí zvýšit frekvenci, na které daný protokol běží. Naopak chceme-li zefektivnit samotný přenos a snížit režii, je nutné optimalizovat délku přenášených dat s datovou šířkou sběrnice nebo použít sběrnici s podporou nezarovnaných transakcí. Nejefektivnějším způsobem, jak snížit režii je však zřejmě agregace dat do delších transakcí. V případě posílání většího množství dat v jedné transakci se sníží počet odeslaných hlaviček a tím i velikost režijních dat.

3.2.3 Propustnost protokolů FrameLink a Internal Bus

V následující části bych chtěl znázornit nutnou datovou propustnost pro každý z výše uvedených protokolů takovou, aby bylo možné přijímat z 10 Gb/s ethernetové linky všechna efektivní data - to jsou čistě data bez preambule, SFD a mezipaketové mezery. Nejprve je však nutné si uvědomit, jakou cestou data putují, než dojdou na sběrnici PCI Express, odkud jsou transportována softwaru. Data přichází po ethernetové lince na vstupní port příslušné karty, kde se odstraní preambule a SFD a jsou přenesena na sběrnici FrameLink. Ta přidá k těmto datům vlastní hlavičku o velikosti 128 bitů. Dále data putují skrze čip, kde se na nich vykonávají příslušné operace. Pokud mají být data poslána do softwaru, jsou nejprve předána sběrnici Internal Bus, jež rovněž k těmto datům přidá vlastní hlavičku o velikosti 128 bitů a až teprve potom jsou odeslána přes PCI Express do softwaru. V průběhu cesty tedy data můžou podstatně nabýt rozměrů, obzvláště pokud přichází v krátkých paketech. V měřeních budu počítat s parametry obou protokolů, jaké jsou aktuálně využity na platformě NetCOPE, tedy hlavičky obou protokolů jsou 128 bitů dlouhé a datová šířka 64 bitů. Zároveň jako minimální délku paketu na protokolu FrameLink budu uvažovat 80 B (64 B data z Ethernetu a 16 B FrameLink hlavička). Z toho vyplývá i minimální velikost transakce na Interní sběrnici - 96 B.

Jak už bylo zmíněno, prvním přechodem mezi protokoly je transformace dat z ethernetové linky na protokol FrameLink. Při této operaci jsou zahozeny preambule, SFD i mezirámcová mezera. Naopak je k vlastním datům přidána hlavička protokolu FrameLink. První graf 3.6 zobrazje minimální nutnou propustnost protokolu FrameLink pro příjem dat z 10 Gb/s Ethernetu. Vstupní rychlost dat je odvozena z režie v Ethernetu v závislosti na délce přijímaných paketů. Na rozdíl od obrázku 3.2, kde se zaměřuji na čistá data v paketech, však zde jako režii počítám pouze preambuli, SFD a mezirámcovou mezeru, protože ostatní data jsou propagována dále. Proto vzorec pro tento výpočet je maličko odlišný od předchozího:

$$\frac{10 \cdot D}{R + D},$$

kde opět *10* znamená rychlost linky v Gb/s, D je množství přenášených dat, tentokrát však včetně hlavičky Ethernetu i CRC kontrolního součtu (tj. 64 až 1518 oktetů). R zastupuje mezirámcovou mezeru (bráno pevných 12 oktetů), preambuli a SFD. Výsledek rovnice je v jednotkách Gb/s. Největší možná dosažená efektivní rychlost na Ethernetu je pro nejdelší pakety, tedy o délce 1518 B. Avšak největší nutná propustnost pro FrameLink protokol je pro délku přijímaných dat 65 B a to 10,35 Gb/s. Dalo by se očekávat, že by to měla být hodnota 64 B, ale není tomu tak. Je to způsobeno právě zarovnáním, jelikož 64 B dlouhá data jsou zarovnána akorát na délku slova, kdežto v případě 65 B je odesílaný rámec o jedno slovo delší. Je z něho však využit pouze jeden bajt. S rostoucí délkou přijímaných paketů se nutná propustnost blíží k 10 Gb/s. Opět je to způsobeno tím, že je nutné odesílat méně režijních dat. Při výpočtu propustnosti jsem postupoval tímto způsobem: pro každou délku rámce v Ethernetu je nutné vypočítat délku rámce, jakou bude mít v protokolu FrameLink. Zde totiž dochází ke skokovým nárůstům kvůli nezarovnaným délkám. Výpočet lze provést podle této rovnice:

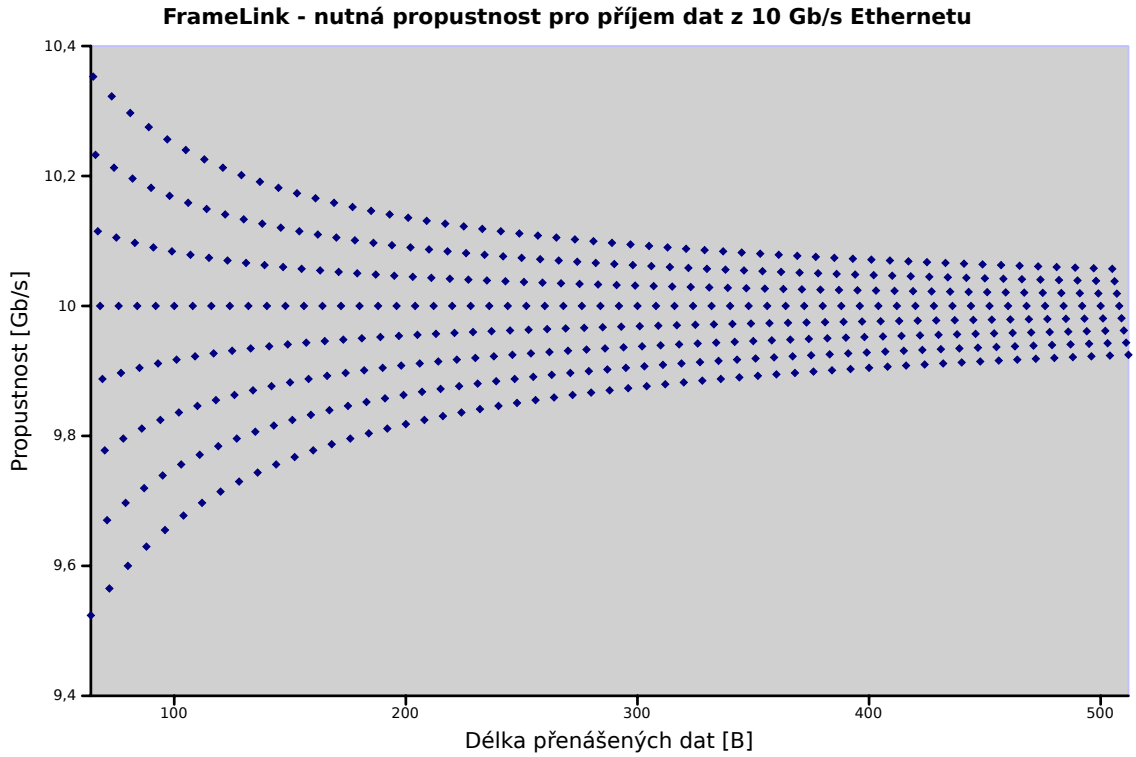
$$ceil(\frac{L}{W}) \cdot W + H,$$

kde L značí délku příslušného ethernetového rámce, W datovou šířku protokolu FrameLink a H je FrameLink hlavička. Funkce *ceil* pak značí zaokrouhlení na nejbližší vyšší celé číslo. Výsledkem rovnice je délka rámce v protokolu FrameLink pro danou délku ethernetového

rámce v B. Nyní známe efektivní propustnost Ethernetu na určité paketové délce a zároveň známe velikost paketu FrameLink protokolu, jež je nutná pro přenesení ethernetového rámce určité délky. Pro výpočet potřebné propustnosti protokolu FrameLink při příjmu efektivních dat z 10 Gb/s Ethernetu v závislosti na délkách paketů lze opět využít vzorec, jež jsme využili pro výpočet efektivní propustnosti Ethernetu. Vzorec však maličko upravíme:

$$\frac{S \cdot D_n}{D},$$

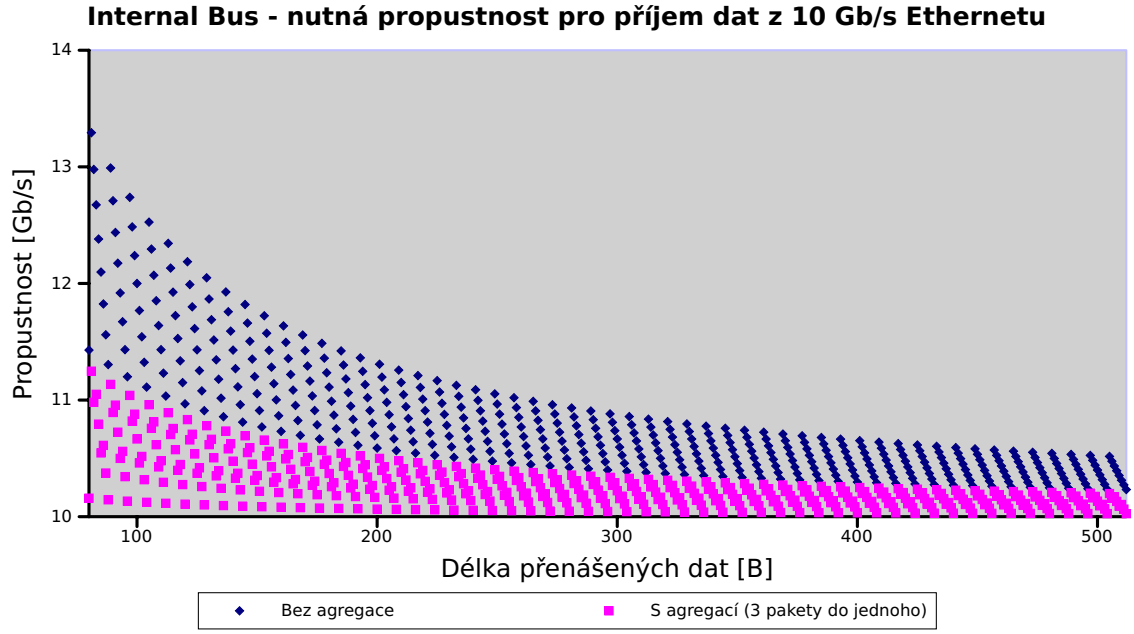
namísto S dosadíme propustnost Ethernetu při posílání paketů o délce $R+D$, za D_n dosadíme délku FrameLink paketu, jež je nutný pro přenesení efektivních dat z Ethernetu o délce D. Výsledek výpočtu pro délky ethernetových rámců od hodnoty 64 B do 512 B lze vidět v již zmiňovaném grafu 3.6. Z uvedeného grafu vyplývá, že pro úspěšný přenos dat z ethernetové linky na protokol FrameLink je v nejhorším případě nutná propustnost téměř 10,4 Gb/s. Ve velké většině případů pak postačuje propustnost menší než 10,2 Gb/s. V případě zarovnaných transakcí však ještě daleko méně a daná propustnost nepřekročí 10 Gb/s.



Obrázek 3.6: Graf zobrazující potřebnou propustnost FrameLinku o datové šířce 64 b a hlavičce 128 b, aby byl schopný přenést 10 Gb/s Ethernet.

Výpočty propustnosti pro sběrnici Internal Bus budou spíše orientační. Při příjmu paketů totiž dochází k ukládání přijatých dat do bufferů. Odtud jsou data odebírána podle toho, jak Interní sběrnice stíhá data odesílat. Může tedy docházet k tomu, že v jedné transakci se odešle více přijatých paketů naráz. Tato vlastnost příznivě ovlivňuje režii na Interní sběrnici a redukuje ji. K výpočtu použiji stejný postup a vzorec, jaký jsem použil výše pro výpočet potřebné propustnosti protokolu FrameLink, pouze patřičně nahradím data

Ethernetu za data FrameLinku. Čili za S dosadíme propustnost FrameLink protokolu při posílání paketů o délce D, za D_n dosadíme délku transakce Internal Bus, jež je nutná pro přenesení celého FrameLink rámce o délce D. Výsledek výpočtu pro délky rámců FrameLink protokolu od hodnoty 80 B do 512 B lze vidět v grafu 3.7. Vypočítaná potřebná propustnost se může zdát, především pro krátké paketové délky, poměrně značná. Proto je ve stejném grafu také naznačeno snížení režie, v důsledku využívání bufferů. Konkrétně tato křivka popisuje snížení v případě, že se odešlou tři přijaté pakety z FrameLink protokolu zároveň jako jedna transakce na Interní sběrnici.



Obrázek 3.7: Graf zobrazující potřebnou propustnost Internal Busu o datové šířce 64 b a hlavičce 128 b, aby byl schopný přenést data z FrameLink protokolu, jež přijímá plnou rychlostí data z 10 GE.

Vždy se tedy odesílá nejdelší možná transakce podle zaplnění bufferů. Například v případě, že se v bufferu nachází šest paketů o délce 81 B, odešlou se zároveň jako jedna dlouhá transakce a ušetří se režie v podobě $5 \cdot 16$ B dat (jedna hlavička v transakci být musí). Když se tedy podíváme do grafu, je to posun z přibližně 13,3 Gb/s propustnosti na 10,25 Gb/s. To je znatelný skok a bez tohoto principu práce s daty by propustnost celé platformy byla znatelně snížena.

3.3 Návrhy na zlepšení architektury

Celá platforma NetCOPE je co do velikosti poměrně rozsáhlá a vezme si relativně dost zdrojů na FPGA čipu. Vývojáři se potýkají s mnoha problémy, které jim brání lehce zvyšovat propustnost. Mezi základní přístupy, jak lze zvýšení propustnosti dosáhnout je buď zvýšením frekvence čipu (nebo jeho pomalejších částí), či zvýšením datové šířky zpracovávaných dat v každém taktu. Obě metody jdou proti sobě. V prvním případě je kladen požadavek na rozmístění prvků na čipu tak, aby všechny signály dorazily na určená místa

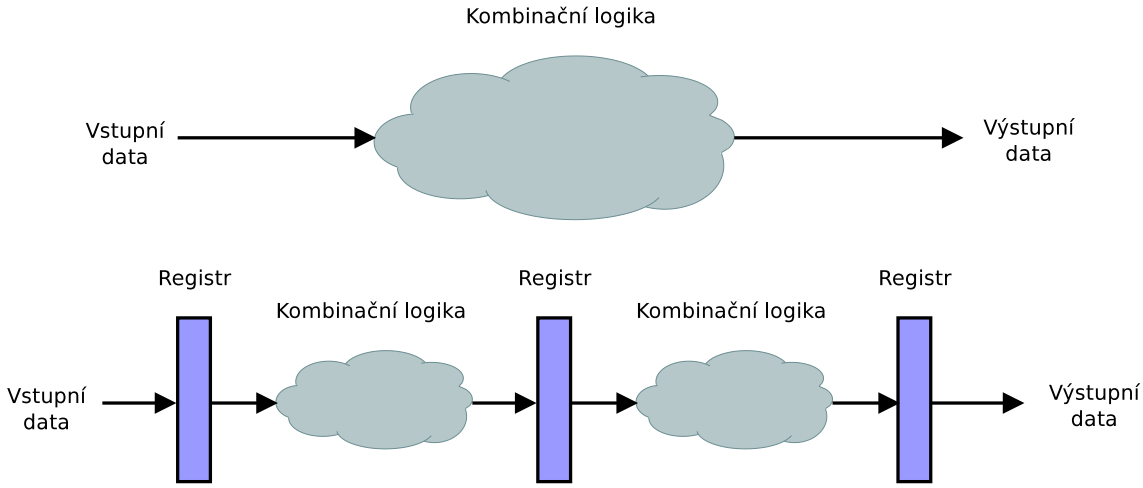
včas, což již při frekvenci 200 MHz a výše není zrovna triviální problém. V druhém případě je problém s razantním navýšením nároků na zdroje na čipu.

Pro splnění těchto všech požadavků jsou využívány nástroje, jež pomocí různých optimalizací hledají nejvhodnější řešení rozmístění prvků. Vyrobení jednoho designu platformy NetCOPE trvá podle zadaných kritérií řádově od jednotek hodin až po desítky. Samozřejmě není výjimkou, že konfigurace v požadovaných nastaveních není nalezena vůbec a je nutné hledat a odstraňovat kritické cesty, či změnit návrh některých komponent tak, aby se nástroji zjednodušila práce.

Mezi základní optimalizace, jichž tyto nástroje využívají, patří zejména pipelining a retiming a jejich kombinace. V případě pipeliningu je nutné tuto optimalizaci provést ručně, správným zásahem do zdrojových kódů. Důležité je pohlídat si, aby se případnou úpravou nezměnilo hlavní chování příslušného obvodu. Je nutné počítat s tím, že přidáním dalšího segmentu prodloužíme dobu zpracování minimálně o jeden takt. Naproti tomu retiming nástroje dokáží používat zcela automaticky přidáním příslušného přepínače, stejně jako i některé další optimalizace, o kterých se stručně zmíním v další části. Samozřejmě platí, že se zapnutím každé další optimalizace, jež se má na vytvářeném designu provádět, se také prodlouží délka syntézy tohoto designu, protože jsou prováděny další výpočetně náročné kroky navíc.

3.3.1 Pipelining

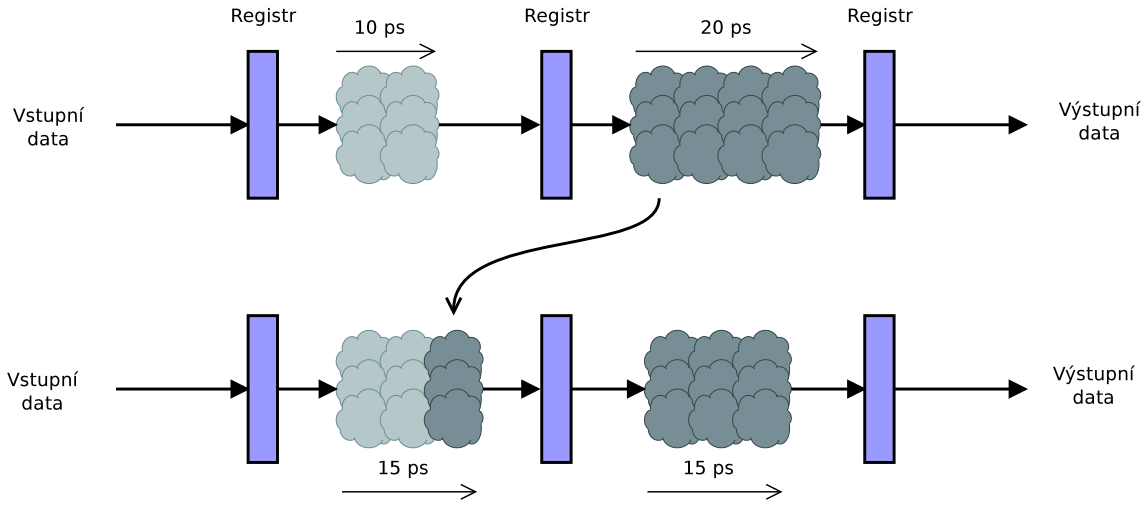
Jedná se o metodu, kdy základní úlohu, jež se skládá z kombinační logiky (bez registrů), rozdělíme na jednotlivé podúlohy tak, aby jejich provedení trvalo stejnou dobu [15]. Mezi každý z těchto dílčích bloků vložíme registr, čímž umožníme řetězcové zpracování dat. Názorně naznačeno to je na obrázku 3.8. Pokud jsme vložili *X* registrových prvků, doba zpracování se zkrátí *X+1* krát. Musíme však počítat s tím, že díky vloženým registrům trvá delší dobu, než se první data ze vstupu dostanou na výstup - vzniká latence.



Obrázek 3.8: Příklad použití pipeliningu. Obrázek převzat z [15]

3.3.2 Retiming

Tato technika navazuje na pipelining a je založena na přeuspořádávání kombinační logiky mezi jednotlivými stupni zřetězené architektury. Přeuspořádání je prováděno tak, aby se pokud možno vyrovnaly časové rozdíly ve zpracování všech kombinačních částí [15]. Díky tomu je pak možné zvýšit frekvenci, na které bude celá logika schopna pracovat. Situaci zachycuje obrázek 3.9.



Obrázek 3.9: Příklad použití retimingu. Obrázek převzat z [15]

3.3.3 Globální optimalizace (Global Optimization)

Jedná se o metodu [3] nástroje XST, jež je používán k syntéze designů NetCOPE platformy. Jejím základem je převedení již vytvořeného strukturálního popisu obvodu zpět do popisu behaviorálního a s využitím dokonalých znalostí samotného fyzického složení architektury FPGA čipu co možná nejlépe zoptimalizovat konkrétní obvod a nakonec jej zpět převést do strukturálního popisu.

3.3.4 Optimalizace logických propojů (Logic Optimization)

Tato metoda [3] pracuje s již vytvořeným strukturálním popisem a snaží se odstranit a optimalizovat kritické cesty v designu. Využívá k tomu přeskupování částí designu s následným postupným umísťováním na čip a postupnou časovou analýzou.

3.3.5 Nástroj SmartXplorer

Velmi vhodné je také zmínit se o nástroji SmartXplorer, jež je součástí balíku nástrojů firmy Xilinx - ISE. Tento nástroj má v sobě předdefinovány kombinace parametrů, jež zapínají příslušné optimalizace v průběhu syntézy a jež jsou osvědčeny jako úspěšné pro vytvoření designu. V případě nenalezení žádné z možných kombinací jako vyhovující, dokáže dokonce vybrat tu variantu, jež dopadla nejlépe a pokusí se ji ještě několikrát aplikovat s mírně upravenými parametry. Tento nástroj lze doporučit především při vytváření designů, jež se

nám nedaří vytvořit klasickou cestou. Je však nutné počítat s velmi dlouhou dobou trvání vytváření designu. Ideální je nechat tento nástroj spuštěný například přes noc.

Doposud zmíněné optimalizace byly součástí nástrojů, jež se samy postaraly o jejich provedení. Zároveň všechny z nich jsou v platformě NetCOPE s úspěchem využívány. Následují další možná zlepšení, jež se však neobejdou bez zásahů do architektury a bez nutnosti jejich návrhu, implementace a úspěšného odladění, zverifikování a otestování.

3.3.6 Podpora nezarovnaných transakcí na sběrnici

Jako další doporučení ke zrychlení propustnosti architektury lze navrhnout využití sběrnic, jež dokáží pracovat s nezarovnanými transakcemi. Velikost ušetřené režie je až přímoúměrná datové šířce příslušné sběrnice. Důvod, proč jsem zmínil slovo „až“ je ten, že záleží na délce posílaných dat. Pokud jsou například posílána data o ideální velikosti, jež se rovná násobku šířky sběrnice, jsou úvahy o využití sběrnic s podporou nezarovnaných transakcí zbytečné. V takovém případě by bylo jejich použití spíše na škodu, jelikož taková sběrnice je obvykle o něco náročnější na zdroje na čipu. V případě však, že by se tato vlastnost ukázala jako vhodná, je namístě navrhnout například v rámci platformy NetCOPE místo protokolu FrameLink použití protokolu LocalLink, z něhož FrameLink vychází a je jeho zjednodušenou verzí.

3.3.7 Variabilní datová šířka sběrnice

V současnosti je možné používat pouze jednu šířku Interní sběrnice pro všechny komponenty. To v určitých případech může být nevýhoda, když některé jednotky na čipu nepotřebují přenášet takové množství dat, jako třeba DMA jednotka. Pro tento případ se jeví jako vhodné navržení generické sběrnice, jež by dovolovala různé jednotky propojit různou datovou šířkou. Ušetřilo by se nemalé množství zdrojů na čipu a posléze by se tak mohlo dosáhnout vyšší frekvence. Právě tato vlastnost je jednou z klíčových milníků platformy pro tento rok. V současné době je návrh ve fázi verifikací a jeho nasazení se očekává v blízké době.

Kapitola 4

Měření a porovnání výsledků

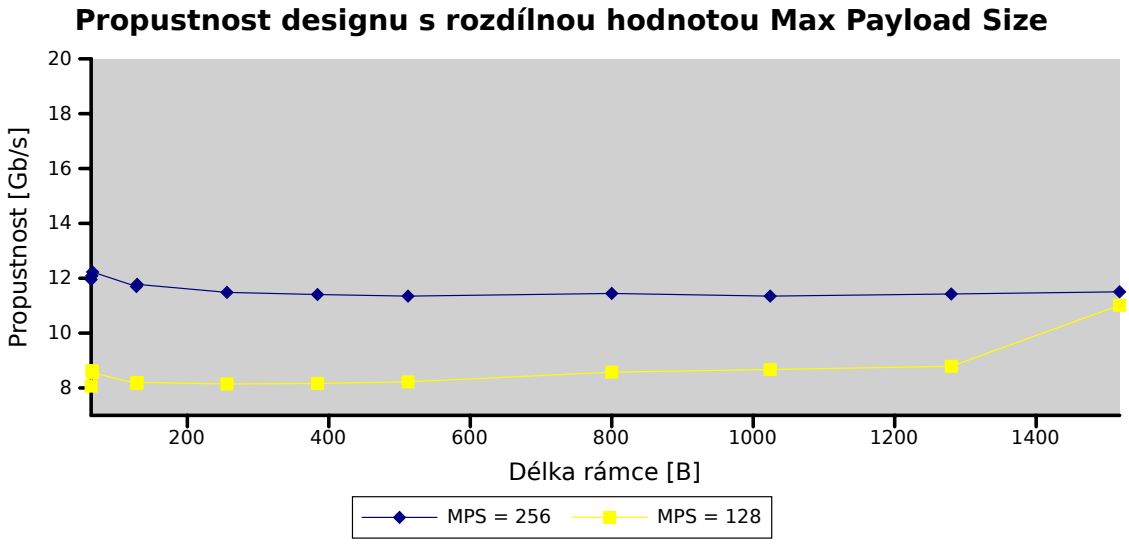
V této kapitole jsem si jako úkol dal za cíl proměřit propustnost platformy NetCOPE při dvou různých nastaveních Max Payload Size sběrnice PCI Express. Jedná se o maximální možnou velikost transakcí, jakými spolu mohou komunikovat dvě zařízení na sběrnici PCI Express. Platforma NetCOPE donedávna využívala pouze transakce o maximální velikosti 128 B. Pro zlepšení propustnosti do softwaru však došlo ke zvýšení této hodnoty na 256 B. PCI Express nabízí nastavení této hodnoty až do velikosti 4096 B. Je však nutné takovouto velikost používat opravdu v odůvodněných případech, protože se již vystavujeme riziku, že při takovýchto dlouhých transakcích můžou jiná zařízení na sběrnici čekat příliš dlouho na data. Tato měření dále zkombinuji s měřením propustnosti designů na různých frekvencích.

Měření propustnosti jsem provedl dvěma způsoby, definovanými v RFC 2544 a RFC 2285. Jedná se o metodu throughput 2.3.1 a metodu FRMOL 2.3.2. K samotnému měření jsem využil kartu COMBOv2 s FPGA čipem lx155t a dvěma 10 GE rozhraními. Dále generátor provozu Test Center Tc2000 firmy Spirent a připravené testovací skripty, jež využívají testeři platformy NetCOPE. Testy zahrnují samotné nastavení tc2000, zahájení testů a vyhodnocení dat. Nastavení testované karty požadovanou konfigurací podle příslušného testu jsem provedl ručně. Testovány byly designy platformy NetCOPE, vytvořené ze složky trunk v SVN projektu, aktuální k datu 9.5.2010. Všechna měření jsem prováděl na stejných předem zvolených paketových délkách. Samozřejmostí jsou délky 64 B a 1518 B a dále délka 65 B, jelikož z předcházejících grafů tato délka vychází jako nejproblematičtější. Nakonec ještě několik délek mezi těmito interavaly. Příjem dat probíhal z obou dvou 10 Gb/s rozhraní na kartě, a tudíž celkový datový tok, jež přicházel do karty byl 20 Gb/s.

4.0.8 Vliv MPS na propustnost

První měření se zabývá vlivem velikosti hodnoty Max Payload Size na propustnost platformy NetCOPE. Toto měření ověří, jestli je platforma omezována rychlostí přenosu dat sběrnicí PCI Express. Z výsledného grafu 4.1 měření pomocí testu throughput je vidět, že velikost MPS má na propustnost výrazný vliv. V průměru se propustnost při nastavení MPS na hodnotu 256 B zvýšila přibližně o 20%. Zajímavý je konec grafu, kde se křivka designu s hodnotou MPS rovnou 128 B výrazně přibližuje křivce s designem s MPS na hodnotě 256 B. To však odpovídá teoretickým výpočtům, jež jsme získali v grafu 3.7. Přestože výpočty propustnosti Interní sběrnice jsou značně nepřesné, lze z nich vyvodit, že s rostoucí délkou přijímaných paketů dochází ke snižování potřebné propustnosti na Interní sběrnici. Díky tomu se omezí i množství dat, jež je nutné posílat přes PCI Express.

Vliv MPS jsem ještě ověřil druhým testem - FRMOL. Oproti throughput testu, FRMOL



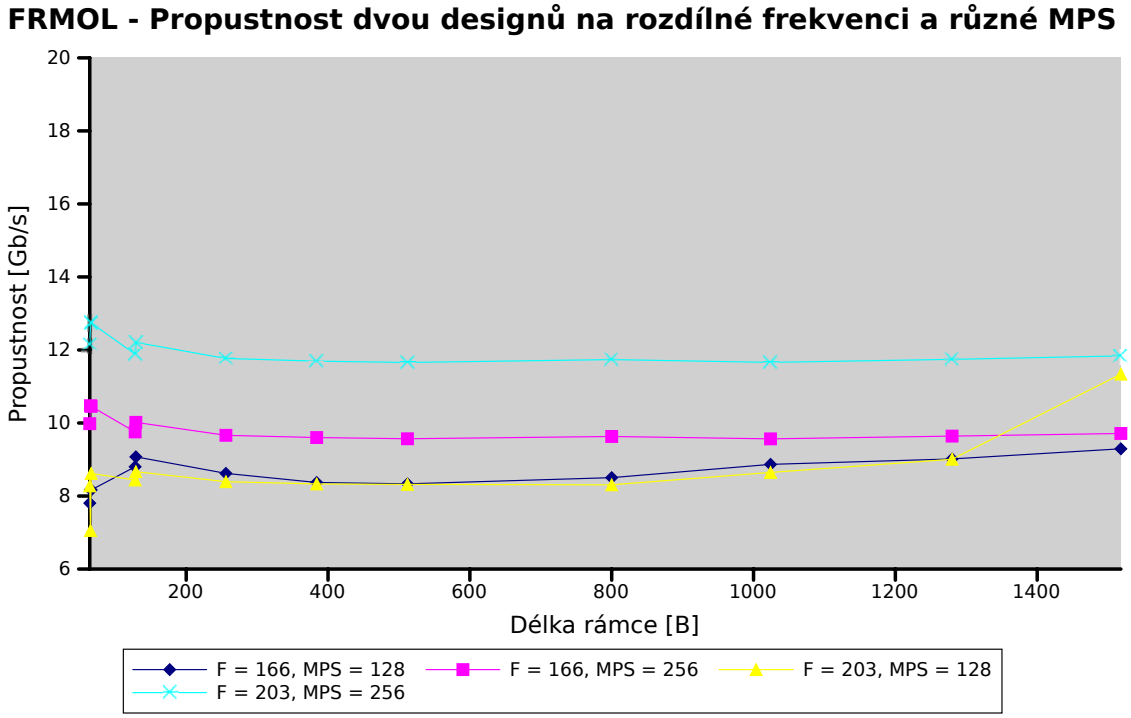
Obrázek 4.1: Ukázka propustnosti designu s různým nastavením MPS měřené pomocí throughput testu.

zatíží linku na maximum pakety dané délky a na konci iterace zjistí, kolik paketů dorazilo a kolik bylo zahozeno. Z tohoto výsledku se počítá ztrátovost/propustnost. Pro srovnání jsem navíc přidal design běžící na 166,7 MHz, viz graf 4.2. I zde jsou výsledky podobné jako v předchozím měření a konce grafů s hodnotou MPS nastavenou na 128 se v obou případech přibližují ke svým protějškům s MPS nastavenou na dvojnásobnou hodnotu. Co se týká samotného zlepšení propustnosti je vidět, že lépe si vede design na vyšší frekvenci, kde je velikost zlepšení přibližně o 20% jako tomu bylo v předchozím měření. Naopak u druhého designu, který běží na nižší frekvenci, se zlepšení projevilo přibližně o 10%. Z grafu lze rovněž vypozorovat to, že oba designy s MPS nastavenou na hodnotu 128 B si vedou s propustností velice podobně, ale jakmile nastavíme MPS na hodnotu 256 B, propustnost každého vzroste o různé hodnoty. Z toho lze usoudit, že v tomto případě je nejpomalejší člen v propustnosti platformy přenos dat přes PCI Express.

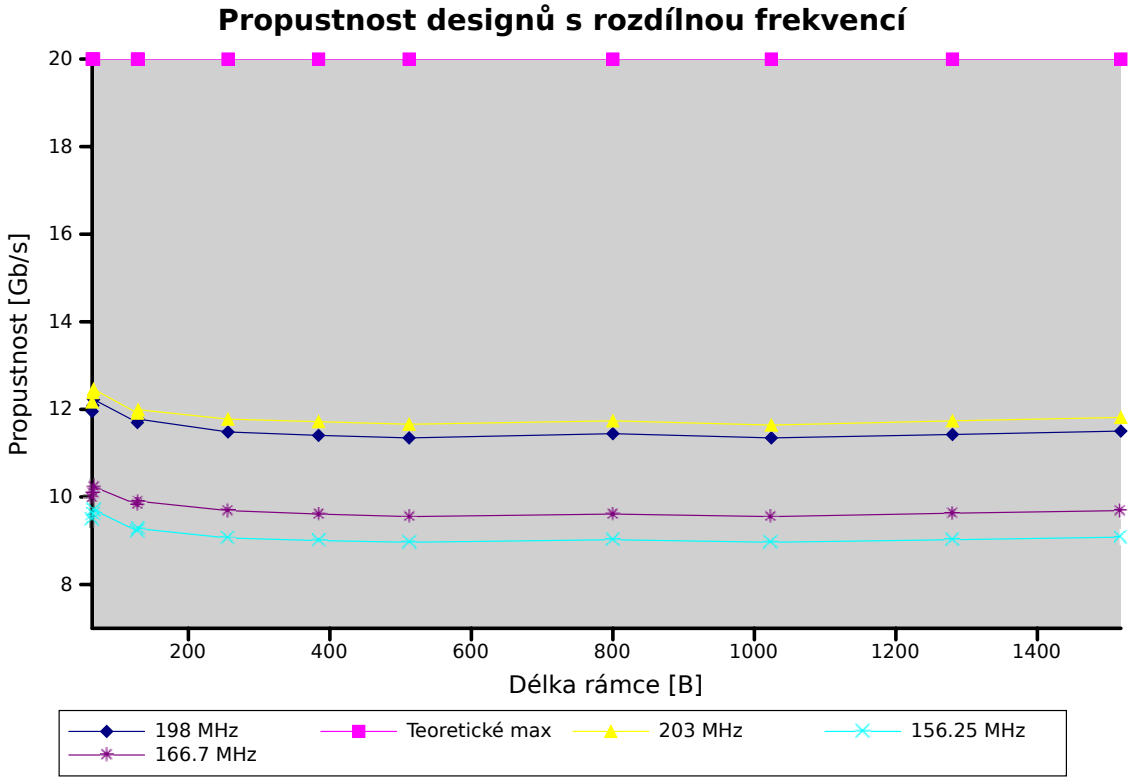
V druhé části měření jsem porovnal tři designy na různých frekvencích a jejich propustnost. V těchto případech jsem nechal MPS nastavené na hodnotě 256 B. Všechny designy se chovaly rovnoměrně na všech měřených paketových délkách. Rovněž rozestupy v propustnosti mezi nimi byly odpovídající jejich frekvencím. Výsledky měření jsou vidět v grafu 4.3.

Z provedených měření je znát, že MPS má na propustnost platformy zásadní vliv. Velikost MPS 128 B je pro NetCOPE platformu již nedostačující a způsobuje omezení z hlediska propustnosti celé architektury. Jako minimální možná velikost se v současnosti jeví hodnota 256 B. Nejvíce to bylo vidět v grafu 4.2, kde designy na různých frekvencích dosahovaly stejné propustnosti právě při této hodnotě MPS. Když se však hodnota MPS zvýšila, každý z designů zvýšil svoji propustnost způsobem, odpovídajícím velikosti jeho frekvence. Z toho lze tedy také usoudit, že zvyšování frekvence designu má rovněž přímý vliv na propustnost celé platformy, což dokazuje i graf 4.3.

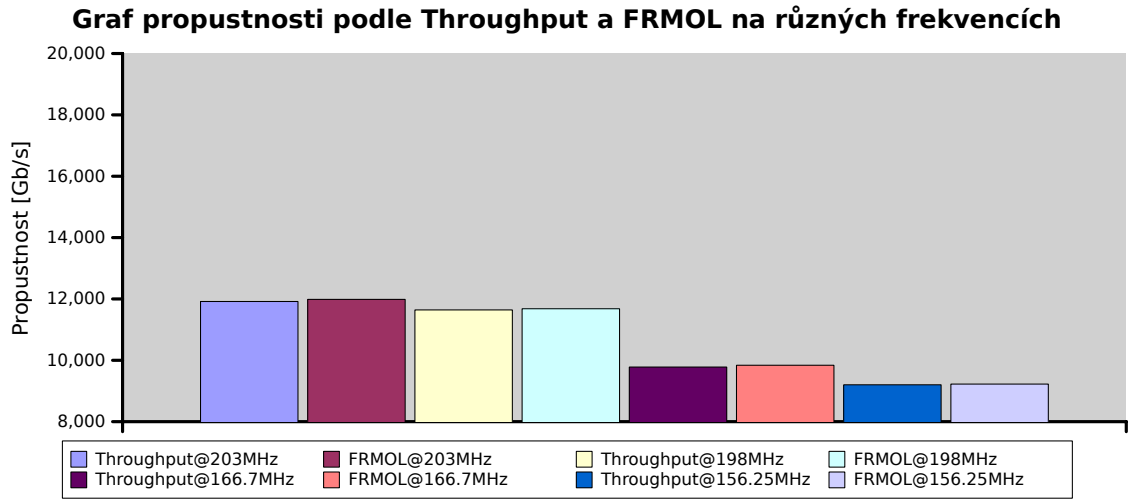
Nakonec jsem ještě vytvořil srovnání rozdílu výsledků mezi oběma typy testů - Throughput a FRMOL. Z grafu 4.4 lze vyčíst jen minimální rozdíly v naměřených hodnotách



Obrázek 4.2: Ukázka propustnosti dvou designů na různých frekvencích a s různou velikostí MPS, měřené pomocí FRMOL testu.



Obrázek 4.3: Ukázka propustnosti designu na různých frekvencích měřené pomocí throughput testu.



Obrázek 4.4: Srovnání propustnosti designů na čtyřech různých frekvencích oběma typy testů (Throughput a FRMOL).

propustnosti s tím, že výsledky testů typu FRMOL vychází vždy nepatrně lépe. Prakticky však obě metody poskytují stejné výstupy. Ze stejného grafu je rovněž ještě navíc viditelné srovnání designů na čtyřech různých frekvencích a jejich propustnost. Podle očekávání stoupá propustnost s velikostí frekvence, na které design běží.

Kapitola 5

Závěr

V této práci byla popsána rodina karet COMBO, která vznikla v rámci projektu Liberouter a je založena na FPGA čipech Virtex 5 firmy Xilinx. Dále zde byla popsána architektura platformy NetCOPE, jež je vyvíjena v rámci stejného projektu a jejímž cílem je zjednodušení a urychlení návrhu síťových aplikací, především v oblasti zpracování dat z vysokorychlostních sítí. Zaměřil jsem se na zjištění slabých míst v přenosu dat mezi jednotlivými částmi platformy. Provedl jsem podrobné výpočty a popsal jsem nároky na použité sběrnice a protokoly, které tyto musí splňovat při příjmu 10 Gb/s ethernetového síťového provozu ze sítě do karty na plné rychlosti linky. V rámci těchto nároků jsem navrhnul některá možná zlepšení, jež by měla pozitivní dopad na výkon celé platformy. Část z nich již byla v platformě využita nebo se aktuálně připravují pro začlenění - například generická šířka Interní sběrnice, jež by měla pomoci ušetřit zdroje na čipu. Další z nich mohou sloužit jako příklad pro zrychlení do budoucna. Například nahrazení protokolu FrameLink protokolem LocalLink by mohlo díky podpoře nezarovnaných transakcí pomoci zvýšit propustnost dat v aplikační části NetCOPE. Dále jsem v práci provedl testy vlivu nastavení maximální délky transakce přes PCI Express (Maximum Payload Size), kde jsem ověřil její vliv na celkovou propustnost platformy.

V dalším vývoji platformy je nutné stálé zvyšování propustnosti a rychlosti zpracování dat. V dohledné době je totiž plánováno začít používat karty se čtyřmi 10 GE rozhraními, kde se prakticky zdvojnásobí síťový datový tok oproti současnému stavu. Pro to aby bylo možné tyto cíle splnit, je nutné stále hledat slabá místa architektury a snažit se je odstraňovat, protože pouhé zvyšování frekvence je již v současnosti velmi obtížné.

Literatura

[1] Benchmarking Methodology for Network Interconnect Devices. WWW, [Online], [1999] [cit. 2010-04-29].
URL http://www.ietf.org/rfc/rfc2544.txt

[2] Benchmarking Terminology for LAN Switching Devices. WWW, [Online], [1998] [cit. 2010-04-29].
URL http://www.apps.ietf.org/rfc/rfc2285.html

[3] Command Line Tools User Guide. PDF, [Online], [2009-12-02] [cit. 2010-05-07].
URL http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/devref.pdf

[4] Ethernet. WWW, [Online], [cit. 2009-12-18].
URL http://cs.wikipedia.org/wiki/Ethernet

[5] Ethernet. WWW, [Online], [cit. 2010-01-22].
URL http://en.wikipedia.org/wiki/Ethernet

[6] FrameLink. WWW, [Online], [cit. 2010-01-13].
URL https://www.liberouter.org/wiki/index.php/FrameLink

[7] Generic Interconnection System (GICS). WWW, [Online], [cit. 2010-01-21].
URL https://www.liberouter.org/trac/netcope/wiki/netcope_doc_gics

[8] IEEE 802.3 LAN/MAN CSMA/CD (Ethernet) Access Method. WWW, [Online], [cit. 2010-04-19].
URL http://standards.ieee.org/getieee802/802.3.html

[9] Our Hardware - Description of COMBO cards. WWW, [Online], [cit. 2010-01-25].
URL http://www.liberouter.org/hardware.php?flag=2

[10] PCI bridge a systém sběrnic na čipu. WWW, [Online], [2007] [cit. 2010-03-11].
URL https://www.liberouter.org/wiki/index.php/PCI_bridge_a_syst%C3%A9m_sb%C4%9Brnic_na_%C4%8Dipu

[11] PCIe Base Specification 2.1. WWW, [Online], [cit. 2010-04-19].
URL http://www.pcisig.com/specifications/pciexpress/

[12] Referenční model ISO/OSI. WWW, [Online], [cit. 2009-12-01].
URL http://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/OSI

[13] TCP/IP. WWW, [Online], [cit. 2009-12-27].
URL http://cs.wikipedia.org/wiki/TCP/IP

[14] Budruk, R.; Anderson, D.; Solari, E.: *PCI Express System Architecture*. Pearson Education, 2003, ISBN 0321156307.

[15] Maxfield, C.: *The Design Warrior's Guide to FPGAs*. Orlando, FL, USA: Academic Press, Inc., 2004, ISBN 0750676043.

[16] Novotný J., Žádník M.: COMBOV2 Cards. PDF, [Online], [2008-02-10] [cit. 2010-05-02].
URL https://www.liberouter.org/data/presentation/2008-06-25-SeminarSlavonice/COMBO-V2-Novotny/COMBOV2.pdf

[17] Tomáš Málek: *Systém interních sběrnic pro čipy s technologií FPGA, diplomová práce*. FIT VUT v Brně, 2008.

Dodatek A

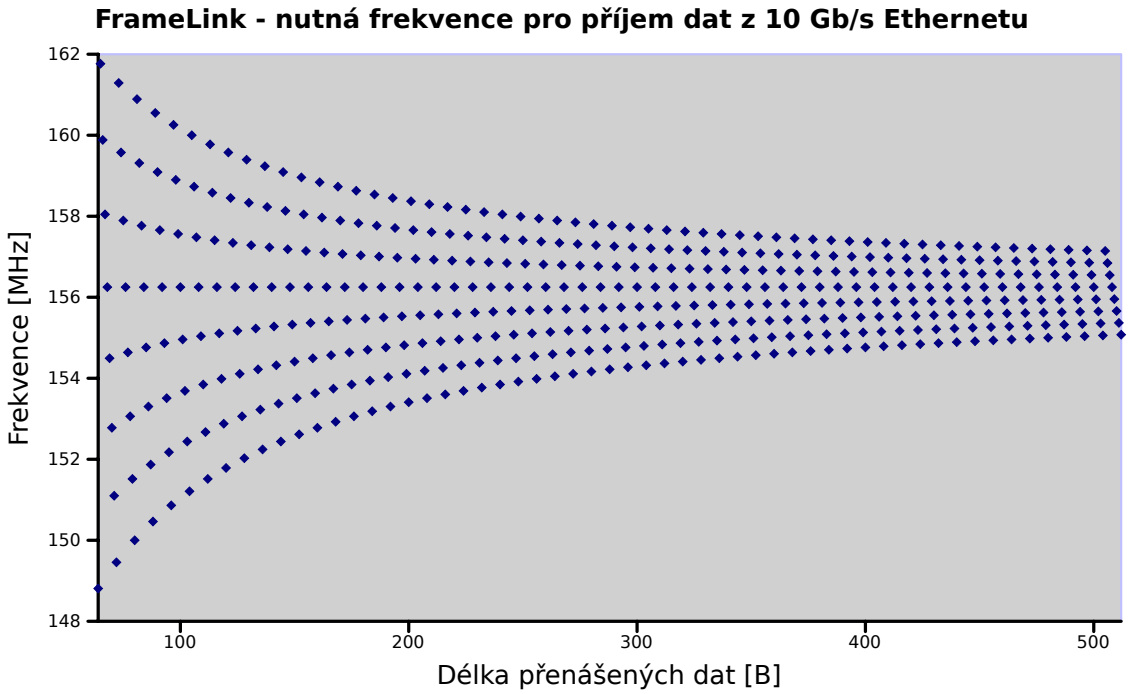
Obsah CD

Přiložené CD obsahuje elektronickou verzi této práce včetně všech grafických materiálů, tabulek a grafů s výpočty a výsledky měření, jež jsou v této práci použity.

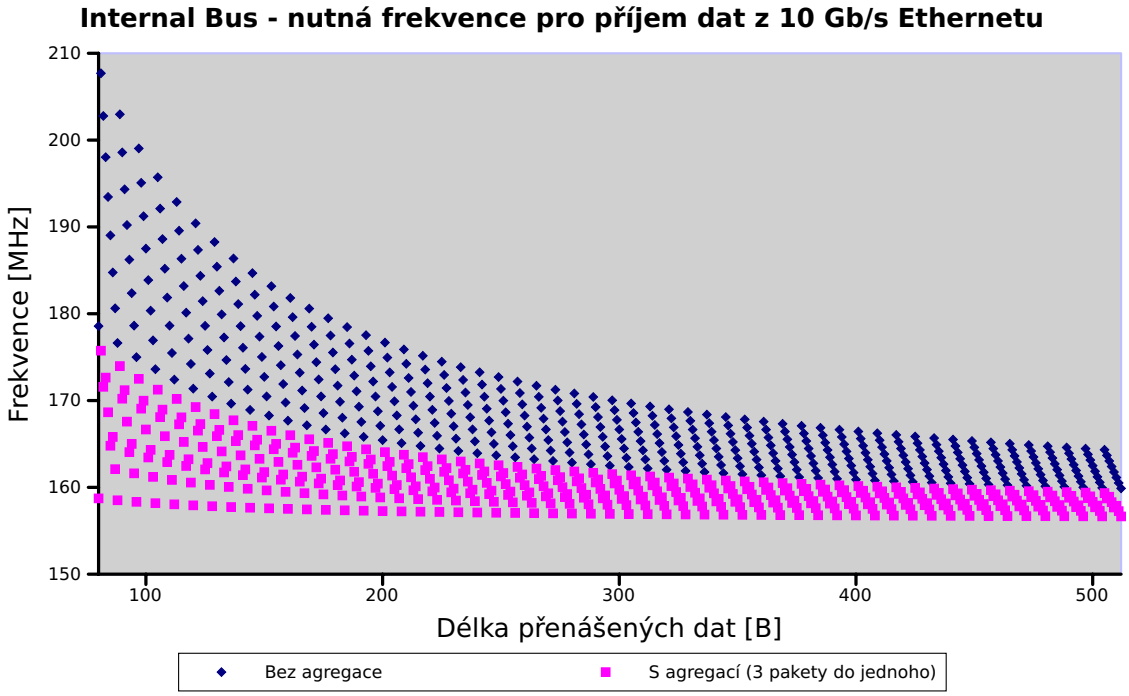
Dodatek B

Frekvence protokolů

V této části přikládám grafy frekvencí protokolů FrameLink a Internal Bus. Tyto frekvence jsou odvozené z potřebné propustnosti jak bylo probíráno v sekci 3.2.3



Obrázek B.1: Potřebná frekvence FrameLink protokolu při příjmu dat z 10 Gb/s Ethernetu. Souvisí s grafem 3.6



Obrázek B.2: Potřebná frekvence Interní sběrnice při příjmu dat z FrameLink protokolu. Souvisí s grafem 3.7

Dodatek C

Tabulky s daty jednotlivých měření

Propustnost [Gb/s], MPS = 256 B				
Délka paketu [B]	f = 156 MHz	f = 166,7 MHz	f = 198 MHz	f = 203 MHz
64	9,355188	9,98175	11,846666	12,160466
65	9,800302	10,474066	12,422818	12,743902
66	9,800014	10,455144	12,422566	12,73838
128	9,144306	9,755654	11,586458	11,885798
129	9,392642	10,018058	11,897326	12,207578
256	9,098084	9,662646	11,471052	11,768354
384	8,9965	9,601466	11,396284	11,693536
512	8,969506	9,56844	11,357104	11,658098
800	9,030526	9,630602	11,437372	11,736574
1024	8,96753	9,565366	11,357384	11,660518
1280	9,038894	9,640818	11,444968	11,742946
1518	9,107904	9,713484	11,531272	11,837058

Tabulka C.1: Výsledky měření propustnosti testem FRMOL pro všechny testované designy s MPS nastaveným na 256 B.

Propustnost [Gb/s], MPS = 128 B				
Délka paketu [B]	f = 156 MHz	f = 166,7 MHz	f = 198 MHz	f = 203 MHz
64	7,799148	7,80552	8,19538	8,27815
65	8,55871	8,177902	8,760314	7,053806
66	8,420752	8,16626	8,795022	8,619394
128	8,737672	8,801842	8,52952	8,439172
129	8,966152	9,07336	8,6804	8,666312
256	8,64456	8,620234	8,426988	8,392686
384	8,562074	8,371006	8,31444	8,330862
512	8,172572	8,33413	8,318706	8,318724
800	8,603424	8,503464	7,828872	8,3068
1024	8,569978	8,867056	8,782672	8,645922
1280	8,622998	9,01612	8,894394	9,007602
1518	8,719122	9,293542	11,038228	11,338304

Tabulka C.2: Výsledky měření propustnosti testem FRMOL pro všechny testované designy s MPS nastaveným na 128 B.

Propustnost [Gb/s], MPS = 256 B					MPS = 128 B
Délka pkt.[B]	f=156 MHz	f=166,7 MHz	f=198 MHz	f=203 MHz	f=203 MHz
64	9,492	9,980	11,953	12,168	8,066
65	9,609	10,098	12,090	12,383	8,633
66	9,707	10,234	12,227	12,461	8,555
128	9,219	9,824	11,699	11,914	8,164
129	9,277	9,902	11,777	11,992	8,203
256	9,062	9,688	11,484	11,777	8,145
384	9,004	9,609	11,406	11,719	8,164
512	8,965	9,551	11,348	11,660	8,223
800	9,023	9,609	11,445	11,738	8,574
1024	8,965	9,551	11,348	11,641	8,672
1280	9,023	9,629	11,426	11,738	8,789
1518	9,082	9,688	11,504	11,816	11,016

Tabulka C.3: Výsledky měření propustnosti testem Throughput pro všechny testované designy s MPS nastaveným na 256 B a jedním na 128 B.