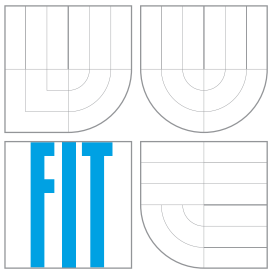
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VYSÍLÁNÍ PAKETŮ NA 100 GB/S ETHERNETU

PACKET TRANSMISSION AT 100 GB/S ETHERNET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VÁCLAV HUMMEL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ MATOUŠEK

BRNO 2014

Abstrakt

Platforma NetCOPE slouží pro rychlý vývoj hardwarově akcelerovaných síťových aplikací na COMBO kartách. Nezbytnou součástí této platformy je i výstupní síťový modul, který návrháři pomáhá s implementací linkové vrstvy síťového modelu ISO/OSI, především podvrstvy MAC. Tato bakalářská práce se zabývá návrhem, implementací a verifikací tohoto modulu pracujícího na rychlosti 100 Gb/s. Dále byla vytvořena aplikace nad platformou NetCOPE pro odesílání krátkých vzorků síťového provozu uložených ve statické paměti QDR. Odesílání je řízeno podle přesných časových značek. Celý systém byl nasazen na kartě COMBO a ověřen pomocí síťového analyzátoru.

Abstract

The NetCOPE platform is used for rapid developement of hardware accelerated network applications on the family of COMBO cards. An essential part of this platform is output network module which helps designers to implement Data Link Layer of the OSI reference model, especially the MAC sublayer. This bachelor's thesis focuses on design, implementation and verification of such a module operating at speed 100 Gb/s. Furthemore, an application on the NetCOPE platform was created. It is designed for transmitting short samples of network traffic stored in QDR static memory. Transmission is controlled by precise timestamps. The whole system was deployed on a COMBO card and verified by a network traffic analyzer.

Klíčová slova

výstupní síťový modul, CGMII, 100 Gigabit Ethernet, NetCOPE, QDR, časové značky, karta COMBO, Virtex-7

Keywords

network output module, CGMII, 100 Gigabit Ethernet, NetCOPE, QDR, timestamps, COMBO card, Virtex-7

Citace

Václav Hummel: Vysílání paketů na 100 Gb/s Ethernetu, bakalářská práce, Brno, FIT VUT v Brně, 2014

Vysílání paketů na 100 Gb/s Ethernetu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Matouška. Další informace jsem získal od kolegů z Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Václav Hummel
20. května 2014

Poděkování

Zde bych rád poděkoval panu Ing. Jiřímu Matouškovi za vedení mé bakalářské práce a kolegům z Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET, kteří mi poskytli technické zázemí a další odborné znalosti.

Obsah

Slovník

ASIC Application-Specific Integrated Circuit. 2

CGMII 100 Gigabit Media Independent Interface. 2, 9, 12–18, 20

CLB Configurable Logic Block. 7

CPU Central Processing Unit. 2

CRC Cyclic Redundancy Check. 5, 17

CSMA/CD Carrier Sense Multiple Access with Collision Detection. 5

DDR Dual Data Rate. 6, 7

DIC Deficit Idle Count. 14, 16, 17

DMA Direct Memory Access. 8, 21, 23–25, 27–30

EFD End of Frame Delimiter. 14

FIFO First In First Out. 7, 16, 25, 27, 30, 32

FLU FrameLink Unaligned. 2, 8–11, 15–17, 19, 20, 23–25, 30, 32

FPGA Field Programmable Gate Array. 2–4, 6, 7, 9–12, 19, 21, 22, 31–33

HVL Hardware Verification Language. 20

IP Internet Protocol. 5

LAN Local Area Network. 2

LED Light Emitting Diode. 16

LSB Least Significant Bit. 4, 5

LUT LookUp Table. 7

MAC Media Access Control. 4–6, 10, 14, 17

MI32 Memory Interface 32-bit. 2, 8–13, 15–17, 19–21, 23, 26, 28

MSB Most Significant Bit. 4, 5

NIC Network Interface Card. 2, 30

OBUF Output BUFfer. 8

PCAP Packet CAPture. 23, 33

PCI Express Peripheral Component Interconnect Express. 6–8

PCIe Peripheral Component Interconnect express. 7, 8

PCS Physical Coding Sublayer. 4, 15

PMA Physical Medium Attachment. 4, 15

QDR Quad Data Rate. 2, 6, 7, 23–26, 31, 33

RAM Random Access Memory. 7, 31

SDRAM Synchronous Dynamic Random Access Memory. 6, 7

SFD Start of Frame Delimiter. 5, 14

VHDL Very high speed integrated circuit Hardware Description Language. 7, 15, 19, 22, 30, 33

VLAN Virtual Local Area Network. 5, 6

XST Xilinx Synthesis Technology. 19, 31

Kapitola 1

Úvod

Počet připojených zařízení do sítě Internet se již začíná počítat v desítkách miliard [1] a neustále roste. Pryč jsou doby, kdy se do této síťě připojovaly výhradně vojenské a akademické instituce či velké firmy. Internet v domácnostech se již stává standardem dokonce i v rozvojových zemích. Díky rozvoji drátových a bezdrátových sítí může dnes být online skutečně cokoliv – stolní a přenosné počítače, tablety, mobilní telefony, termostaty, pračky, atd. Internet se stal nejběžnějším komunikačním prostředkem a jsou na něho kladeny čím dál tím větší nároky – bezpečnost, propustnost, doba odezvy, apod.

Nejpoužívanější technologií pro výstavbu lokálních sítí LAN s přenosem přes fyzické médium je Ethernet. Ethernet je definovám standardem IEEE 802.3 [9] a je to realizace fyzické a linkové vrstvy síťového modelu ISO/OSI. U koncových uživatelů je běžná propustnost až 1 Gb/s. Ná páteřních spojích se už pomalu ustupuje od 10 Gb/s technologie a začíná se nasazovat 100 Gb/s Ethernet [2]. V přípravě je již standard Ethernetu s propustností až 400 Gb/s.

Přenosové rychlosti v počítačových sítích rostou rychleji než výkon univerzálních výpočetních čipů – CPU. Běžné softwarové řešení je tedy nedostačující a musí se využívat hardwarová akcelerace. Síťové přepínače a směrovače využívají aplikačně specifických integrovaných obvodů – ASIC. Tato technologie má vysoký výkon, nízkou spotřebu, ale malou flexibilitu a vysoké počáteční náklady.

Alternativou k čipům ASIC je technologie FPGA. Má nižší výkon, vyšší spotřebu, ale daleko větší flexibilitu. Touto technologií se zabývá například Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Na tomto oddělení vznikla mimo jiné platforma pro rychlý vývoj hardwarově akcelerovaných síťových aplikací – NetCOPE.

Prvním cílem práce je vytvořit výstupní síťový modul s propustností 100 Gb/s, který bude nedílnou součástí platformy NetCOPE. S uživatelskou aplikací v rámci této platformy bude komunikovat přes protokol FLU [5]. Připojení na fyzickou vrstvu obstarává rozhraní CGMII [9]. K přístupu ke konfiguračním a stavovým registrům z hostitelského zařízení slouží sběrnice MI32 [5]. Implementace síťového modulu bude ověřena ve funkčním verifikačním prostředí vytvořeném v jazyce SystemVerilog a následně nasazena na COMBO kartě jako součást síťové karty – NIC.

Druhým cílem je vytvořit aplikaci nad platformou NetCOPE, která bude vysílat krátké vzorky síťového provozu na maximální rychlosti 100 Gb/s podle přesných časových značek z externí statické paměti QDR. Klasická síťová karta vytvořená nad platformou NetCOPE by teoreticky také dokázala vysílat na maximální rychlosti, nikoliv však podle přesných časových značek. Toto řešení by také zbytečně zatěžovalo hostitelské zařízení a v praxi by mělo potíže s maximální propustností [4].

Práce je členěna do 5 kapitol. V kapitole 2 jsou popsány použité technologie – Ethernet, FPGA Virtex-7 a platforma NetCOPE. V kapitole 3 je popsán celý proces tvorby výstupního síťového modulu. V kapitole 4 je popsána síťová aplikace pro vysílání paketů na 100 Gb/s Ethernetu. V poslední kapitole 5 jsou zhodnoceny dosažené výsledky a je diskutováno využití aplikace pro vysílání paketů.

Kapitola 2

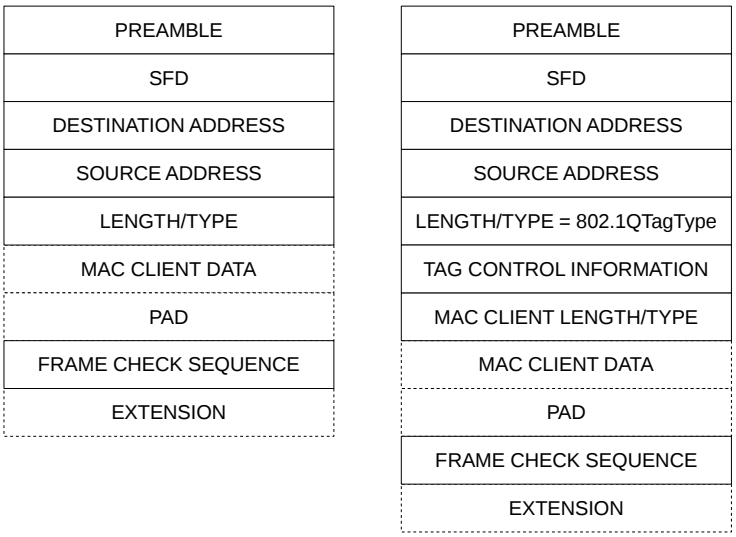
Teoretický rozbor

V této kapitole bude detailně popsán Ethernet, FPGA Virtex-7 a platforma NetCOPE. Spojení těchto technologií umožní vysílání paketů na rychlosti 100 Gb/s, což je cíl této práce.

2.1 Ethernet

Dominantně používanou technologií pro výstavbu lokálních a metropolitních sítí je Ethernet definovaný standardem IEEE 802.3 [9], který si své postavení vybudoval nízkou cenou a vysokou přenosovou rychlostí – v současnosti až 100 Gb/s. Ethernet zaujímá místo fyzické a linkové vrstvy referenčního modelu ISO/OSI. Na fyzické vrstvě řeší například kódování dat (podvrstva PCS), připojení k fyzickému médiu (podvrstva PMA), typ fyzického média (kroucená dvojlinka, optické vlákno). Na linkové vrstvě se řeší především řízení přístupu k médiu (podvrstva MAC).

Data z vyšších vrstev referenčního modelu ISO/OSI jsou přenášena v ethernetových rámcích (viz obrázek 2.1). Existuje více druhů ethernetových rámců: standardní rámec (basic frame), Q-tagovaný rámec (Q-tagged frame) [7] a další [8]. Jednotlivé bity jsou vysílány od LSB po MSB.



Obrázek 2.1: Standardní a Q-tagovaný ethernetový rámec

2.1.1 Standardní ethernetový rámec

Nejčastěji používaný ethernetový rámec se skládá z následujících polí.

Preamble (7 B) slouží k synchronizaci mezi vysílačem a přijímačem. Je to posloupnost 7 bytů `10101010` (od LSB po MSB).

SFD (1 B) má hodnotu 10101011 (od LSB po MSB) a okamžitě za ním začíná MAC rámec.

Destination Address (6 B) obsahuje MAC adresu cílového zařízení. Adresa může být individuální (*unicast*), skupinová (*multicast*) nebo všesměrová (*broadcast*).

Source Address (6 B) obsahuje MAC adresu odesílatele. Adresa je vždy individuální.

Length/Type (2 B) má dva významy odlišené numerickou hodnotou (uloženo ve formátu *Big-Endian*). Pokud je hodnota menší nebo rovna 1500 (0x05DC), pak toto pole udává délku pole MAC Client Data v bytech. Pokud je hodnota větší nebo rovna 1536 (0x0600) musí být toto pole interpretováno jako typ protokolu, který je zapouzdřen v poli MAC Client Data.

MAC Client Data a Pad (46-1500 B) nese užitečná data (*Payload*) ethernetového rámce. Pokud je její velikost menší než 46 B, je zarovnána položkou Pad na minimální velikost 46 B. Velikost této položky však nesmí překročit 1500 B. Minimální délka rámce je dána historicky, kdy Ethernet používal sdílené fyzické médium. Pro řízení přístupu se používal algoritmus CSMA/CD. Doba odeslání rámce minimální délky musela být větší než kolizní okénko (doba, než se signál rozšíří do celého fyzického média). Dnes se používají téměř výhradně dedikované spoje, avšak pro zachování zpětně kompatibility minimální délka rámce zůstává. Maximální délka rámce byla zvolena s ohledem na velikosti vstupních front aktivních síťových prvků a doby přenášení jednoho rámce.

Frame Check Sequence (4 B) obsahuje kontrolní součet (CRC), který je vypočítán z položek Destination Address, Source Address, Length/Type, MAC Client Data a Pad. Slouží ke kontrole integrity přenášeného rámce.

Extension se používá výhradně na rychlosti 1 Gb/s při režimu polovičního duplexu (*Half-Duplex*). Doba odeslání rámce minimální délky na této rychlosti je menší než kolizní okénko (doba než se signál rozšíří do celého média), a proto pole Extension zvyšuje minimální délku rámce.

2.1.2 Q-tagovaný ethernetový rámec

Slouží pro vytváření virtuálních lokálních sítí (VLAN). Tento mechanismus dovoluje v rámci jedné sítě vytvořit více logicky oddělených sítí. Využívá se například pro zajištění bezpečnosti u IP telefonie. Oproti obyčejným rámcům přibyla dvě pole (viz obrázek 2.1). Následuje popis polí specifických pro Q-tagované rámce.

Length/Type = 802.1QTagType (2 B) má hodnotu 0x8100 a je to identifikátor typu 802.1Q.

Tag Control Information (2 B) obsahuje informace ohledně VLAN. Význam bitů je uveden v [7].

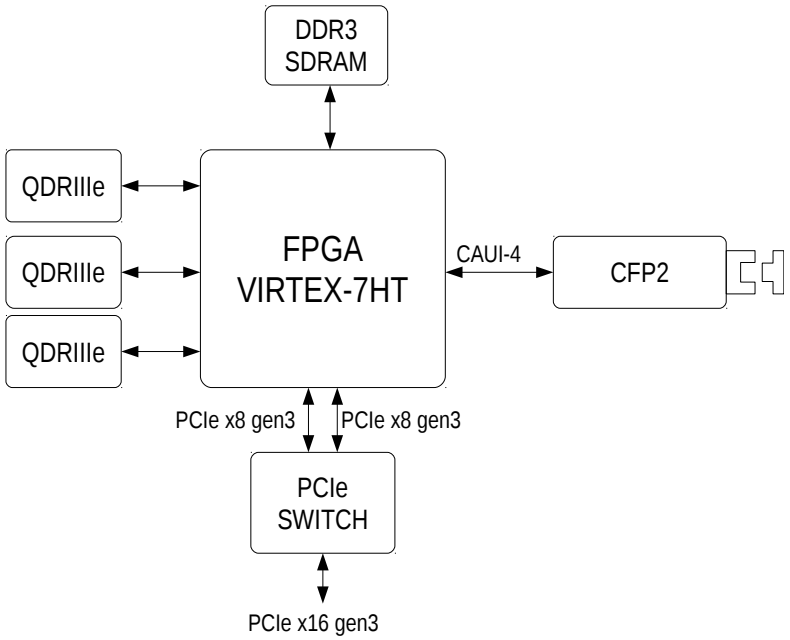
MAC Client Length/Type (2 B) odpovídá poli Length/Type obyčejného ethernetového rámce.

2.2 Rodina karet COMBO

Rodina karet COMBO je množina přídavných akceleračních karet určených do PCI Express slotu, které slouží ke zpracování síťových dat na vysokorychlostních linkách. Rodinu tvoří základní karty osazené FPGA čipem, externí pamětí, PCI Express konektorem a dalšími komponentami a dále pak přídavné karty, na nichž jsou umístěna síťová rozhraní. V současnosti se základní a přídavná karta spojily v jednu kartu, historicky ovšem stále patří do rodiny COMBO karet [3]. V následujícím textu je popsána nejnovější akcelerační karta COMBO-100G a její jádro – FPGA čip Virtex-7.

2.2.1 COMBO-100G

Schéma akcelerační karty je uvedeno na obrázku 2.2. Skládá se z čipu FPGA Virtex-7HT, optického rozhraní CFP2, PCI Express přepínače, 3 statických pamětí QDRIIIe a dynamické paměti DDR3 SDRAM [5]. Jednotlivé komponenty jsou blíže popsány v následujícím textu.



Obrázek 2.2: Schéma COMBO-100G karty [5]

FPGA Virtex-7HT je popsán v podkapitole 2.2.2.

CFP2 tvoří rozhraní mezi elektrickými a optickými signály. Rozhraní CAUI-4 tvoří 8 diferenciálních párů vodičů (4 pro každý směr), kde každý pár přenáší data rychlostí 25 Gb/s.

PCIe SWITCH spojuje dva koncové body PCI Express x8 uvnitř čipu FPGA v jeden x16. 16 linek PCI Express generace 3 dosahuje propustnosti přes 100 Gb/s.

QDRIIIe jsou statické, dvouportové paměti (jeden port pro čtení a druhý port pro zápis) s nízkou latencí, vysokou propustností a náhodným přístupem. Během jednoho taktu jsou schopny provést až 4 operace (technologie QDR). Pracují na frekvenci až 700 MHz s datovou šířkou 36 b – propustnost překračuje 50 Gb/s. Karta COMBO-100G obsahuje tři paměti QDRIIIe, každá s kapacitou 72 Mb.

DDR3 SDRAM je dynamická pamět, která má oproti statické paměti vyšší kapacitu, ale i vyšší latenci a ne úplně náhodný přístup. Tyto paměti se budou používat hlavně jako velké vyrovnávací paměti.

2.2.2 Virtex-7

Jádrem akcelerační karty COMBO-100G je FPGA čip Virtex-7 XC7VH580T od společnosti Xilinx. Klíčovými prvky tohoto čipu jsou konfigurovatelné logické bloky (CLB), blokové paměti RAM (BRAM), univerzální sériové přijímače/vysílače (*transceivers*) a integrované bloky pro PCI Express. V následujícím textu budou jednotlivé komponenty dále popsány ve vztahu ke 100 Gb/s Ethernetu.

Konfigurovatelné logické bloky tvoří základ čipu Virtex-7. Skládají se ze dvou bloků Slice a ty jsou složeny ze čtyř 6vstupých LUT, logiky pro rychlou realizaci aritmetického přenosu (*arithmetic carry logic*), 8 registrů (*flip-flops*) a 8 multiplexorů. Překladové nástroje mapují do těchto bloků aplikace zapsané v jazyce pro popis hardwaru (např. VHDL).

Blokové paměti RAM mají kapacitu 36 Kb a dva nezávislé porty s konfigurovatelnou datovou šířkou až 72 b. Tyto bloky je také možné použít jako asynchronní fronty FIFO pro přechod mezi hodinovými doménami.

Sériové přijímače/vysílače s propustností 28,05 Gb/s jsou naprosto klíčové pro vytvoření 4 kanálové 100 Gb/s (4krát 25 Gb/s) síťové karty [14].

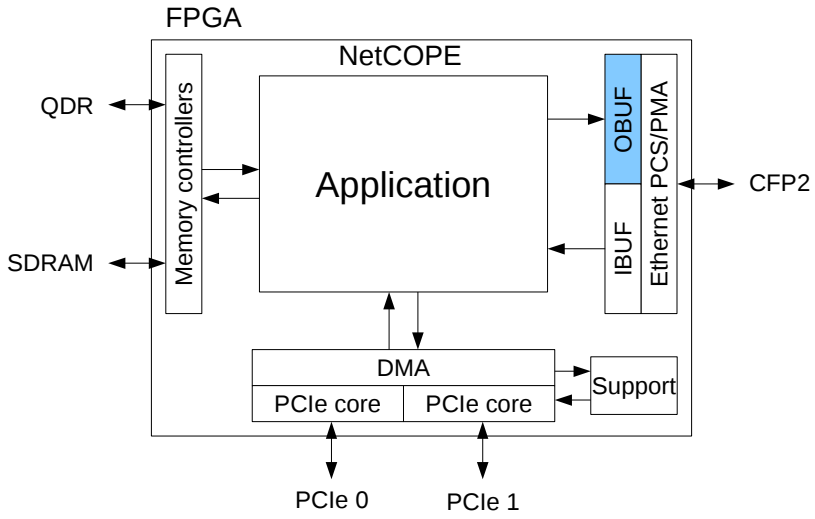
Integrované bloky pro PCI Express slouží pro přenos dat do hostitelského zařízení. Čip Virtex-7 XC7VH580T obsahuje dva integrované bloky pro PCI Express 3. generace s propustností 8 Gb/s na jednu linku. Každý blok podporuje až 8 linek, což dává celkem 16 linek s maximální teoretickou propustností 128 Gb/s. Požadavek na alespoň 100 Gb/s propustnost je tedy splněn.

2.3 Platforma NetCOPE

Platforma NetCOPE slouží pro rychlý vývoj hardwarově akcelerovaných síťových aplikací [10]. Je určena především pro rodinu karet COMBO, které vyvinulo zájmové sdružení CESNET. Je složena ze softwarové a firmwarové části. Pro komunikaci s uživatelskou aplikací v rámci platformy NetCOPE slouží komunikační protokoly a rozhraní FLU a MI32. V následujícím textu jsou jednotlivé vrstvy detailně rozebrány.

2.3.1 Firmware

Hlavní myšlenkou firmwarové části platformy NetCOPE je odstínit návrháře od hardwarově specifických vlastností akcelerační karty. Může se tak zaměřit na tvorbu samotné aplikace. Schéma firmwarové části platformy NetCOPE je na obrázku 2.3. Skládá se ze síťového modulu (OBUF, IBUF, Ethernet PCS/PMA), rozhraní PCI Express (PCIe core), DMA řadiče (DMA), paměťových řadičů (Memory controllers) a pomocných modulů (Support).



Obrázek 2.3: Schéma firmwarové části platformy NetCOPE

Síťový modul slouží k vysílání a přijímání dat ze síťě. Modul OBUF (výstupní síťový modul) bude vytvořen v rámci této práce, a proto je zvýrazněn.

PCIe core umožňuje obousměrný, vysokorychlostní, point-to-point přenost dat.

DMA slouží pro přímý přístup do paměti hostitelského zařízení. Tento řadič zajišťuje obousměrný, vysokorychlostní přenos dat mezi akcelerační kartou a hostitelským zařízením.

Memory controllers abstrahují fyzické rozhraní externích pamětí, a tím zjednodušují jejich používání.

Support umožňují identifikovat akcelerační kartu a její firmware, monitorovat teplotu a napájecí napětí, přistupovat ke stavovým a konfiguračním registrům, generovat přesné časové značky (*timestamps*) a další.

2.3.2 Software

Software pro komunikaci s akcelerační kartou je vytvořen v linuxovém prostředí (výhradně se používá platforma `x86/AMD64`) a je rozdělen do tří vrstev. Nejblíže hardwaru jsou ovladače, které poskytují základní rozhraní mezi hardwarem a softwarem. Nad nimi je vytvořena množina knihoven a těchto knihoven se využívá při tvorbě softwarových nástrojů. Důležité je, že tyto nástroje se mohou použít i ke komunikaci s vlastní firmwarovou aplikací – není nutné vytvářet další software.

2.4 Protokoly a rozhraní

Na vstupu výstupního síťového modulu směrem od uživatelské aplikace v rámci platformy NetCOPE bude použito rozhraní FLU. S výstupním síťovým modulem bude hostitelské zařízení komunikovat skrze sběrnici MI32. Rozhraní FLU se transformuje na rozhraní CGMII, které bude připojeno na vstup jednotky Ethernet PCS/PMA. Pochopení těchto rozhraní je nutné pro tvorbu výstupního síťového modulu. Rozhraní FLU a MI32 budou mít využití i v uživatelské aplikaci v rámci platformy NetCOPE.

2.4.1 FLU

FLU je vysokorychlostní, paketově orientovaný, point-to-point protokol. Používá se především pro přenos síťových rámců v rámci čipu FPGA, ale může se i použít pro přenos obecných dat. Rozhraní je definováno tabulkou 2.1. Význam jednotlivých signálů je uveden v následujícím textu.

Signál	Šířka (b)	Směr
CLK	1	vstup
DATA	$DATA\_WIDTH(16, 32, 64, \dots)$	zdroj → cíl
SOP	1	zdroj → cíl
SOP_POS	1 až $\log_2(DATA\_WIDTH/8)$	zdroj → cíl
EOP	1	zdroj → cíl
EOP_POS	$\log_2(DATA\_WIDTH/8)$	zdroj → cíl
SRC_RDY	1	zdroj → cíl
DST_RDY	1	cíl → zdroj

Tabulka 2.1: Signály rozhraní FLU

Clock (CLK) je signál, který slouží k synchronizaci.

DATA je sběrnice, po které se po částech přenášejí obecná data. Data jsou platná pouze, když je SRC_RDY = '1' a DST_RDY = '1'. Šířka této sběrnice je dána generickým parametrem $DATA\_WIDTH$.

Start Of Packet (SOP) říká, že v daném datovém slově se nachází začátek rámce. Je platný v logické 1 a pouze, když je SRC_RDY = '1' a DST_RDY = '1'.

Start Of Packet Position (SOP_POS) udává pozici začátku rámce v datovém slově. Jeho generická šířka, dána parametrem SOP_POS_WIDTH, ovlivňuje zarovnání začátků rámců. Zarovnání se vypočte podle vztahu $DATA\_WIDTH/(2^{SOP\_POS\_WIDTH})$. Signál je platný pouze, když je platný signál SOP.

End Of Packet (EOP) říká, že v daném datovém slově se nachází konec rámce. Je platný v logické 1 a pouze, když je SRC_RDY = '1' a DST_RDY = '1'.

End Of Packet Position (EOP_POS) udává pozici konce rámce v datovém slově. Jeho šířka je dána parametrem $DATA\_WIDTH$ a vypočte se vztahem $\log_2(DATA\_WIDTH/8)$. Signál je platný pouze, když je platný signál EOP

Source Ready (SRC_RDY) říká, že odesílatel je připraven odeslat data. Aby data odeslal, musí počkat na potvrzení od příjemce, což může, ale nemusí, nastat ve stejném taktu.

Destination Ready (DST_RDY) říká, že příjemce je připraven přijímout data. Data jsou platná pouze, když je připraven i odesílatel. Z toho vyplývá, že k přenosu dat dojde, když je SRC_RDY = '1' a zároveň DST_RDY = '1'.

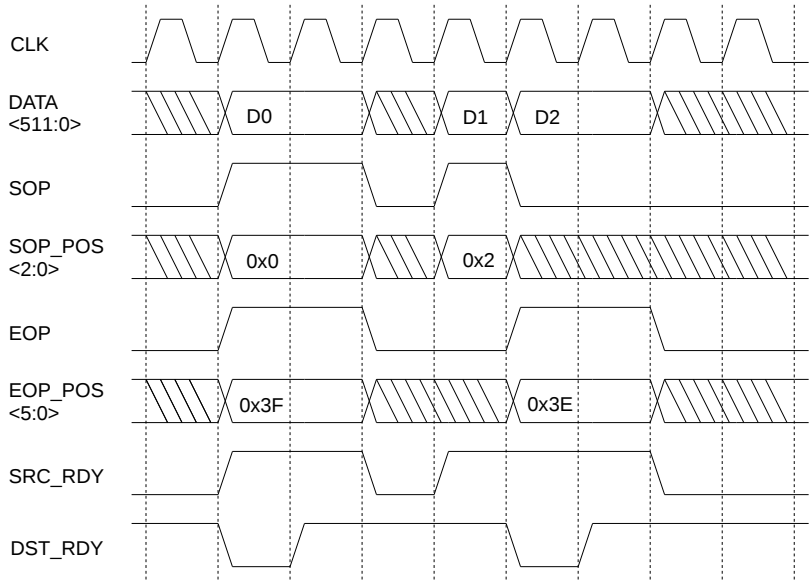
Ukázka komunikace podle komunikačního protokolu FLU je zobrazena na obrázku 2.4. V prvním taktu není připraven odesílatel, a proto nedošlo k přenesení datového slova. V druhém taktu není připraven příjemce, a proto opět nedojde k přenosu dat. Ve třetím taktu je odesílatel a příjemce připraven, a proto dojde k přenosu datového slova. V tomto taktu jsou také platné signály SOP i EOP, což znamená, že došlo k přenesení celého paketu. Podle kapitoly 2.1 je tento paket nejkratší platný MAC rámec, ale ve skutečnosti se skrze rozhraní FLU přenášejí MAC rámce bez kontrolního součtu, tedy nejkratší platný MAC rámec přenášený přes rozhraní FLU má délku 60 B. Ve čtvrtém taktu není připraven odesílatel. V pátém taktu je připraven odesílatel i příjemce a je tedy přeneseno první datové slovo paketu, ale protože signál EOP je neaktivní, nejedná se ještě o celý paket. V šestém taktu není připraven příjemce. V sedmém taktu je připraven odesílatel i příjemce a je tedy přeneseno druhé datové slovo paketu. Aktivní signál EOP znamená, že došlo k přenesení celého paketu. Hodnota generických parametrů rozhraní FLU není zvolena náhodně ($DATA\_WIDTH = 512, SOP\_POS\_WIDTH = 3$) – takto je používá platforma NetCOPE na kartě COMBO-100G.

2.4.2 MI32

Sběrnice MI32 slouží k přístupu ke konfiguračním registrům uvnitř jednotek v FPGA z hostitelského zařízení [13]. Nemá vysokou propustnost, ale nespotřebovává tolik zdrojů na čipu FPGA jako sběrnice FLU. Rozhraní je definováno tabulkou 2.2. Dále bude popsán význam jednotlivých signálů.

Clock (CLK) je signál, který slouží k synchronizaci.

Data for Write Bus (DWR) je sběrnice dat k zápisu do kontrolních a konfiguračních registrů uvnitř FPGA.



Obrázek 2.4: Ukázka komunikace protokolem FLU

Signál	**Šířka (b)**	**Směr**
CLK	1	vstup
DWR	32	vstup
ADDR	32	vstup
RD	1	vstup
WR	1	vstup
BE	4	vstup
DRD	32	výstup
ARDY	1	výstup
DRDY	1	výstup

Tabulka 2.2: Signály rozhraní MI32

Address Bus (ADDR) udává adresu čtených nebo zapisovaných dat. Rozlišuje se podle signálů RD a WR.

Read Request (RD) indikuje požadavek na čtení. Signál je platný v logické 1.

Write Request (WR) indikuje požadavek na zápis. Signál je platný v logické 1.

Byte Enable Bus (BE) umožňuje maskovat byty čtených nebo zapisovaných dat.

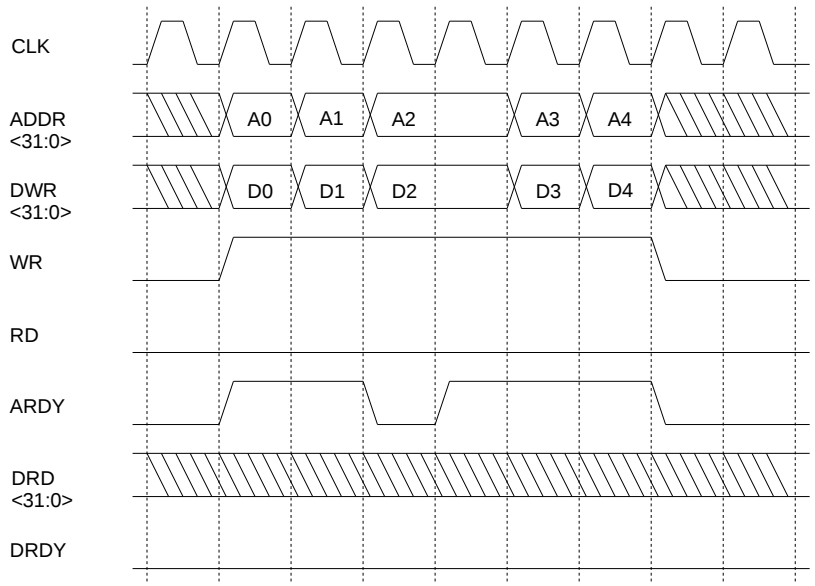
Data for Read Bus (DRD) je sběrnice, po které se přenášejí vyčtená data ze stavových registrů.

Address Ready (ARDY) slouží pro potvrzení čtecího nebo zápisového požadavku a adresy komponentou uvnitř FPGA. Signál je platný v logické 1.

Data Ready (DRDY) signalizuje přítomnost vyčtených dat na sběrnici DRD.

Následuje ukázka komunikace podle komunikačního protokolu MI32. Na obrázku 2.5 zobrazen zápis 5 slov o 32 bitech. V prvním taktu nejsou platné signály WR a RD, a nic se tedy neděje. Ve druhém taktu je platný signál WR a zároveň i signál ARDY, dojde tedy k zápisu datového slova. Ve třetím taktu dochází k zápisu dalšího datového slova. Ve čtvrtém taktu není připravena komponenta uvnitř FPGA (ARDY = 0), a proto nedojde k zápisu datového slova. V pátém, šestém a sedmém taktu dojde k zápisu datového slova.

Na obrázku 2.6 je zobrazeno čtení 5 slov o 32 bitech. V prvním taktu nejsou platné signály WR a RD, a nic se tedy neděje. Ve druhém taktu je platný signál RD, zároveň i signály ARDY a DRDY, dojde tedy k přečtení datového slova. Ve třetím taktu dochází k přečtení dalšího datového slova. Ve čtvrtém taktu není připravena komponenta uvnitř FPGA (ARDY = 0), a proto nedojde k přečtení datového slova. V pátém, šestém a sedmém taktu dojde k přečtení datového slova. Čtecí nebo zápisovou operaci inicializuje vždy hostitelské zařízení.

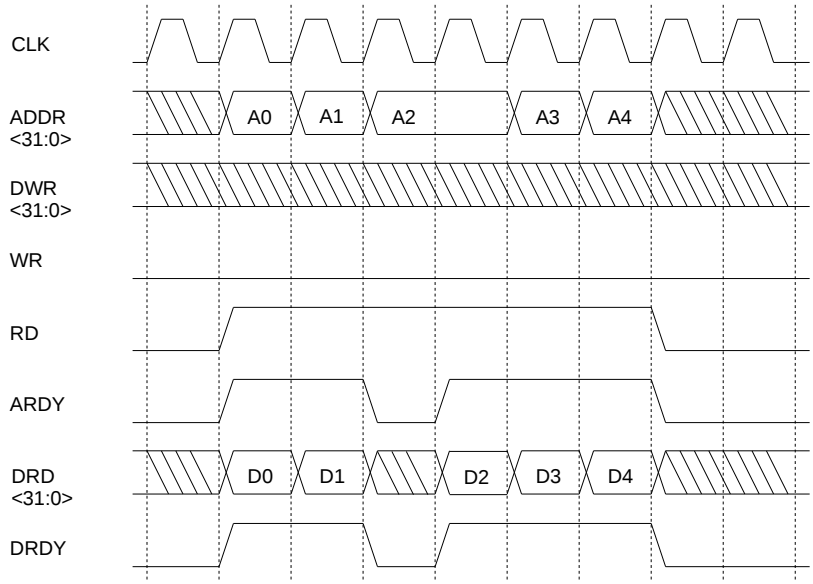


Obrázek 2.5: Ukázka zápisu datových slov na sběrnici MI32

2.4.3 CGMII

Rozhraní CGMII je definováno ve standartu IEEE 802.3 [9] na úrovni fyzické vrstvy referenčního modelu ISO/OSI. Z důvodu příliš vysoké pracovní frekvence 1562,5 MHz pro použití v FPGA bylo toto rozhraní interně upraveno – došlo ke zvětšení datové šířky z 64 bitů na 512 bitů, a tím i ke snížení frekvence na 195,3125 MHz. Na toto rozhraní jsou kladeny následující požadavky:

- rychlost 100 Gb/s v plně duplexním režimu
- nezávislé hodiny pro vysílání (TX) a příjem (RX)
- 512 bitové datové rozhraní pro vysílání i příjem



Obrázek 2.6: Ukázka čtení datových slov na sběrnici MI32

Je nutno poznamenat, že výstupní síťový modul používá pouze směr *TX*. Směr *RX* je zde uveden pouze pro úplnost. Signály rozhraní CGMII jsou uvedeny v tabulce 2.3.

Signál	**Šířka (b)**	**Směr**
TX_CLK	1	vstup
RX_CLK	1	vstup
TXD	512	výstup
TXC	64	výstup
RXD	512	vstup
RXC	64	vstup

Tabulka 2.3: Definice rozhraní CGMII

Význam jednotlivých signálů je následující:

TX_CLK a RX_CLK slouží pro synchronizaci. Pro rozhraní CGMII mají tyto signály frekvenci 195,3125 MHz.

TXD a RXD jsou datové signály, které jsou rozděleny do 64 8bitových linek. Každá linka má svůj řídicí signál.

TXC a RXC jsou řídicí signály. Udávají přítomnost/nepřítomnost řídicích znaků na odpovídajících datových linkách (přehled řídicích znaků viz tabulka 2.4)

Data se přes rozhraní CGMII přenášejí v podobném formátu jako ethernetové rámce popsané v kapitole 2.1. Tento formát se nazývá CGMII data stream a má následující podobu:

`<Inter-Frame><Preamble><SFD><Data><EFD>`

TXC/RXC	TXD/RXD	Význam
0	0x00 až 0xFF	Data
1	0x07	Idle
1	0xFB	Start
1	0xFD	Terminate
1	0xFE	Error

Tabulka 2.4: Řídicí znaky na rozhraní CGMII

<Inter-Frame> je mezera mezi rámci, která musí být v průměru alespoň 12 bytů, ale nesmí být menší než 5 bytů. Proměnlivá velikost mezery se řídí čítačem DIC a je dána tím, že položka <Preamble> musí být vždy zarovnána na 8 bytů. Mezera začíná řídicím znakem Terminate a za ním následují řídicí znaky Idle. Mezera vždy větší nebo rovna 12 bytům se nepoužívá (standard ji ale nezakazuje) z důvodu neefektivního využití přenosové linky, které souvisí se zarovnáním začátků rámců.

<Preamble> odpovídá stejnojmenné části ethernetového rámce s tím rozdílem, že 1. byte je kontrolním znakem Start. Začátek musí být zarovnaný na některou z linek 0, 8, 16, 24, 32, 40, 48 nebo 56.

<SFD> za tímto znakem začíná MAC rámec (viz 2.1).

<Data> odpovídají MAC rámci (viz 2.1).

<EFD> označuje konec rámce a odpovídá kontrolnímu znaku Terminate. Započítává se do mezery mezi rámci.

Kapitola 3

Výstupní síťový modul

V této kapitole bude popsán návrh (podkapitola 3.1), implementace (podkapitola 3.2) a testování (podkapitola 3.3) výstupního síťového modulu. V návrhu budou nejdříve definovány požadavky a poté popsána samotná architektura včetně popisu jednotlivých podkomponent. Implementace bude provedena v jazyce VHDL a budou diskutovány dosažené výsledky. Neméně důležitou částí je testování. Na Oddělení nástrojů pro monitorování a konfiguraci sdružení CESNET se v současnosti používají tři techniky testování – simulace, funkční verifikace a ověření v hardwaru. Nejinak tomu bude i v této práci.

3.1 Návrh

Návrh je z hlediska tvorby nejen výstupního síťového modulu velice důležitý. Opomenutí i zdánlivě malého problému může mít za následek opakování vývojového cyklu. V této podkapitole budou nejdříve zmíněny požadavky na výstupní síťový modul, detailně popsána jeho architektura a adresový prostor stavových a konfiguračních registrů včetně jejich popisu.

3.1.1 Požadavky

Výstupní síťový modul bude součástí platformy NetCOPE a z toho plynou použitá rozhraní. Požadavky jsou následující:

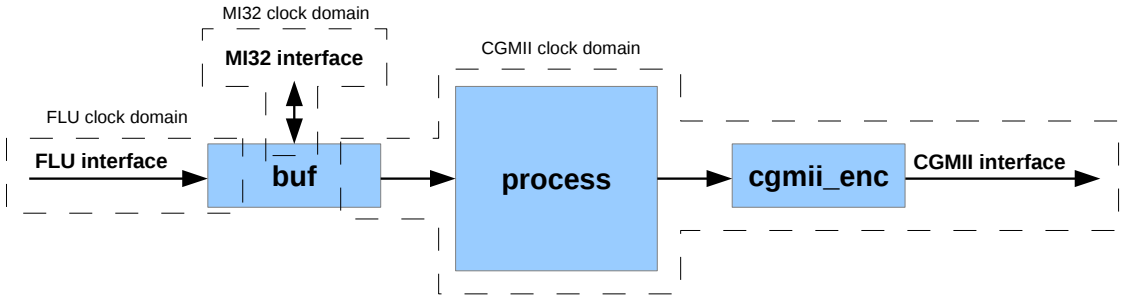
- CGMII rozhraní mezi výstupním síťovým modulem a implementací vrstvy PCS/PMA.
- FLU rozhraní mezi uživatelskou aplikací a výstupním síťovým modulem
- MI32 rozhraní pro přístup ke stavovým a konfiguračním registrům
- rozdílné hodinové domény a resetovací signály pro rozhraní CGMII, FLU a MI32
- bufferování odesílaných dat
- rozšiřování rámců na minimální délku
- možnost pozastavit/obnovit činnost výstupního síťového modulu
- čítač odeslaných rámců
- čítač odeslaných bytů

- výpočet kontrolního součtu Frame Check Sequence (CRC)
- implementace DIC pro efektivní využití přenosové linky
- signalizace odchozích rámců pro LED diodu

Více hodinových domén se používá z důvodu větší flexibility systému. Hodinová doména CGMII má pevně dané hodiny na frekvenci 195,3125 MHz. U hodinové domény FLU už záleží na konkrétní aplikaci v rámci platformy NetCOPE. Mohou se použít hodiny z domény CGMII, často je ale nutné použít jiné hodiny ať už s vyšší nebo nižší frekvencí. Propusnost sběrnice MI32 nebývá kritická, a proto se obvykle taktuje na nízké frekvence.

3.1.2 Architektura

Na základě požadavků a znalostí z kapitoly 2 vzniklo schéma výstupního síťového modulu na obrázku 3.1.



Obrázek 3.1: Schéma výstupního síťového modulu

Výstupní síťový modul je rozdělen do několika podkomponent, což má tu výhodu, že se návrhář může soustředit v jeden okamžik pouze na jednu z nich. Každá z těchto jednotek může být nezávisle na ostatních detailně navržena, implementována a také testována. Tento přístup se skutečně použije. Vzniklé chyby budou tedy při testování odhaleny mnohem dřív. Na závěr bude provedena integrace a důkladné testování. Nyní budou do podrobna popsány jednotlivé podkomponenty.

Jednotka buf

Jak už napovídá název, jednou z funkcí této jednotky je bufferování odesílaných dat. Není to ale jediná funkce. Vstupní FLU rozhraní je přivedeno na vstup fronty typu FIFO. Zde se asynchronně oddělí hodinová doména FLU a CGMII. Stále jde o rozhraní FLU, ale v hodinové doméně CGMII.

Tato fronta je typu *store & forward*, což znamená, že dokud nepřijme celý MAC rámec, signalizuje, že je prázdná. Tato funkce je nezbytná pro vytvoření datového toku CGMII, protože zde neexistuje možnost pozastavení odesílání rámce. Fronta je složena z blokových pamětí RAM, popsaných v kapitole 2.2.2, doplněných o další logiku.

Fronta typu *store & forward* má jednu nevýhodu. V případě, že by do ní přišel rámec větší, než je velikost fronty, zasekla by se. Na vstupu by hlásila, že je plná, kdežto na výstupu, že je prázdná. Nutno říct, že rámce takové délky by se vůbec na vstupním rozhraní neměly vyskytnout. Přesto se počítá s jednoduchou ochranou, která dlouhé rámce násilně rozdělí.

Rozdělené rámce sice nebudou dávat žádný smysl, avšak takto dlouhý rámec nejspíš také ne, pravděpodobně vznikl v důsledku nějaké chyby.

Dále jsou zde umístěny čítače odeslaných rámců a bytů, logika pro signalizaci odchozího rámce a stavové a konfigurační registry přístupné přes sběrnici MI32 z hostitelského zařízení. Je opět nutno použít asynchronní oddělení, protože sběrnice MI32 pracuje v jiné hodinové doméně.

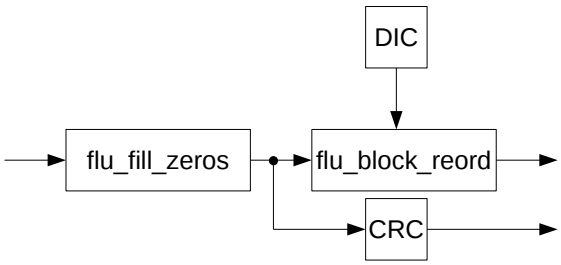
Jednotka process

Tvoří jádro výstupního síťového modulu. Převod mezi protokolem FLU a CGMII není úplně přímočarý. Rámce přenášené na rozhraní FLU jsou typu MAC (bez kontrolního součtu), zatímco na rozhraní CGMII se vyskytuje CGMII data stream podobný ethernetovým rámcům.

CGMII data stream obsahuje navíc oproti MAC rámcům položky `<Inter-Frame>`, `<Preamble>`, `<SFD>`, `<EFD>` (viz 2.4.3) a navíc je ještě nutno přidat kontrolní součet (Frame Check Sequence, viz 2.1.1). Tyto položky mají v součtu průměrně 24 bytů a ke každému MAC rámci se musejí přidat.

Jednotka process je dále dělena na podjednotky. Podjednotka `flu_fill_zeros` rozšiřuje MAC rámce na minimální délku. Podjednotka `flu_block_reord` zvětšuje mezeru mezi MAC rámci na FLU sběrnici takovým způsobem, aby byly snadno rozšiřitelné na CGMII data stream. K tomu využívá podjednotku `DIC`, což je 3bitový čítač, který si uchovává informaci, kolik kontrolních znaků Idle na CGMII data streamu bylo vynecháno z průměrných 12.Nakonec podjednotka `CRC` vypočítá kontrolní součet.

Výstupem jednotky `process` je FLU rozhraní, kde jsou MAC rámce uspořádány takovým způsobem, že konverze na CGMII data stream je už přímočará a rozhraní pro přenos CRC součtu. Vnitřní struktura je zobrazena na obrázku 3.2.



Obrázek 3.2: Schéma jednotky `process`

Jednotka cgmii_enc

Dokončuje konverzi na CGMII data stream. Je nutné si uvědomit, že jednotka `process` pouze přeuspořádá MAC rámce vhodným způsobem tak, aby je bylo snadné rozšířit na CGMII data stream. Pořád se ale jedná o platné MAC rámce bez kontrolního součtu. Tato jednotka tedy doplní položky `<Inter-Frame>` (kontrolní znaky Idle), `<Preamble>`, `<SFD>`, kontrolní součet CRC a `<EFD>`. Nakonec se zahodí kontrolní signály FLU rozhraní a vytvoří se kontrolní signál TXC rozhraní CGMII.

3.1.3 Adresový prostor

Pro komunikaci s hostitelským zařízením slouží stavové a konfigurační registry. V jednotce `buf` už bylo naznačeno o jaké registry se jedná, nic ale nebylo řečeno o tom, na jakých adresách se nacházejí. Absolutní adresa je dána bázovou adresou výstupního síťového modulu a posunutím adres jednotlivých registrů vůči této adrese. Adresový prostor je popsán tabulkou 3.1 a hned za ní jsou popsány jednotlivé registry.

Adresa	Registr	Velikost [B]	Čtení/Zápis
0x00	Total Sent Frames Counter, spodní část	4	Čtení
0x04	Octects Sent Counter, spodní část	4	Čtení
0x08	Rezervováno	8	Čtení
0x10	Total Sent Frames Counter, horní část	4	Čtení
0x14	Octects Sent Counter, horní část	4	Čtení
0x18	Rezervováno	8	Čtení
0x20	OBUF Enable Register	4	Čtení/Zápis
0x24	Rezervováno	8	Čtení
0x2C	Control Register	4	Zápis
0x30	OBUF Status Register	4	Čtení

Tabulka 3.1: Adresový prostor výstupního síťového modulu

Total Sent Frames Counter, spodní i horní část tyto dva registry slouží pro uložení hodnoty 64bitového čítače odeslaných rámců.

Octects Sent Counter, spodní i horní část tyto dva registry slouží pro uložení hodnoty 64bitového čítače odeslaných bytů.

OBUF Enable Register nejnižší bit tohoto registru zapíná (hodnota 0x1) nebo vypíná (hodnota 0x0) činnost výstupního síťového modulu. Při vypnutí sice nechochází k vysílání na rozhraní CGMII, ale k ukládání rámců do přítomných bufferů ano.

Control Register je speciální registr, do kterého se zapisují 8bitové instrukce pro práci s čítači. Instrukce jsou uvedeny v tabulce 3.2.

Hodnota	Název	Význam
0x01	OBUFCMD STROBE COUNTERS	uložení čítačů do zachytávacích registrů
0x02	OBUFCMD RESET COUNTERS	vynulování čítačů

Tabulka 3.2: Popis instrukcí pro práci s čítači

OBUF Status Register slouží pro uložení stavových informací o výstupním síťovém modulu. Umožňuje mimo jiné identifikaci výstupního síťového modulu. Na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET vzniklo více výstupních síťových modulů, které se liší rychlostí odesílání dat. V tabulce 3.3 je uveden úplný výčet výstupních síťových modulů.

Od (b)	Do (b)	Hodnota	Rychlost
4	5	0x0	10 Mb/s
4	5	0x1	100 Mb/s
4	5	0x2	1 Gb/s
4	5	0x3	10 Gb/s
4	6	0x4	40 Gb/s
4	6	0x5	100 Gb/s

Tabulka 3.3: Výčet výstupních síťových modulů a jejich identifikace

3.2 Implementace

Výstupní síťový modul navržený v podkapitole 3.1 byl naimplementován v jazyce VHDL. Při implementaci byly využity některé komponenty vytvořené na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Konkrétně asynchronní fronta typu *store & forward*, blok pro výpočet kontrolního součtu ethernetového rámce, generické multiplexory, asynchronní fronta pro MI32 sběrnici a jednotky pro zřetězené zpracování na rozhraní FLU. V rámci jednotky `process` nebylo naimplementováno rozšiřování na minimální délku. Tato funkce není nezbytně nutná a díky modulárnímu návrhu se může v budoucnu snadno doplnit.

3.2.1 Výsledky syntézy

Naimplementovaný výstupní síťový blok byl vysyntetizován nástrojem XST společnosti Xilinx verze 14.7. Tato první fáze při vytváření bitového toku (*bitstream*) pro čip FPGA mimo jiné ověří syntaktickou správnost kódu a odhadne maximální výslednou frekvenci a spotřebované zdroje. Syntetizovány byly i jednotlivé komponenty výstupního síťového bloku s cílem odhalit problémy s nízkou frekvencí a spotřebovanými zdroji co nejdříve.

Jako cílová technologie pro syntézu byl zvolen čip Virtex-7 typ XC7VH580T. Tento čip je osazen na kartě COMBO-100G popsané v kapitole 2.2.1. Výsledky syntézy jsou uvedeny v tabulce 3.4.

Zařízení	LUT	Registry	BRAM	Frekvence [MHz]	Nástroj
XC7VH580T	11787 (3 %)	5235 (0 %)	23 (2 %)	201,499	XST 14.7

Tabulka 3.4: Výsledky syntézy, v závorce je uvedeno procentuální využití čipu

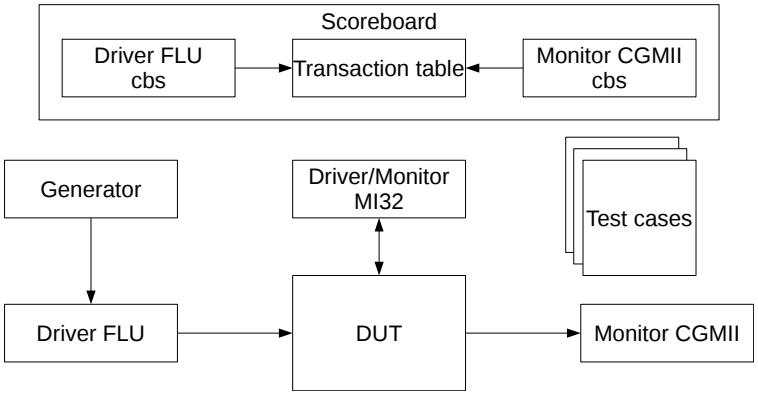
Z tabulky je patrné úzké hrdlo systému – maximální frekvence. Požadováno je alespoň 195,3125 MHz a rezerva není nikterak velká.

3.3 Testování

Testování je často opomíjenou fází vývoje, ale především u hardwarových komponent je tato fáze velice důležitá. Existuje více způsobů testování. Podle bakalářské práce [13] se používá simulace, funkční verifikace, hardwarové testování a formální verifikace. Liší se časovou náročností, úrovní abstrakce a úplností testů. Všechny způsoby testování, kromě formální verifikace, byly použity pro otestování implementovaného výstupního síťového modulu.

3.3.1 Funkční verifikace

Verifikace chování komponenty se dá nazvat automatizovanou simulací. Vytváří se v jazycích typu HVL – například v jazyce SystemVerilog, ve kterém bylo vytvořeno funkční verifikační prostředí pro výstupní síťový modul. Pro jazyky HVL je typická podpora pro náhodné generování testovacích hodnot. Tím se dosáhne otestování většího množství stavů testované komponenty. Vyhodnocení, zdali komponenta ve funkční verifikaci uspěla, či nikoliv, se provádí automatizovaně. Typický je modulární přístup, čímz se dosahuje vysoké znovupoužitelnosti kódu funkčního verifikačního prostředí. Verifikační prostředí výstupního síťového modulu je zobrazeno na obrázku 3.3. Nutno podotknout, že bylo využito funkční verifikační prostředí pro výstupní síťový modul s propustností 40 Gb/s vytvořený na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Převážně stačilo jenom změnit generické parametry tohoto prostředí a vytvořit množinu testovacích případů (*test cases*). V následujícím textu je popis jednotlivých komponent funkčního verifikačního prostředí.



Obrázek 3.3: Schéma funkčního verifikačního prostředí výstupního síťového modulu

Generator slouží pro generování vstupních transakcí metodou *constrained random generation*, což znamená, že vstupy jsou sice náhodné, ale musí splňovat předem stanovené podmínky.

Driver FLU transformuje vygenerované transakce na protokol FLU a dále je posílá na vstupní rozhraní testované komponenty. Po odeslání informuje pomocí callback funkce Scoreboard.

Monitor CGMII přijímá rámce na výstupním rozhraní CGMII a znovu z nich vytváří obecnější transakce. Poté informuje Scoreboard callback funkcí.

Scoreboard provádí automatické vyhodnocování funkční verifikace. Driver FLU pošle tomuto objektu vstupní transakci. Ve Scoreboardu se tato transakce převede na očekávanou výstupní transakci a uloží se do Transaction table. Monitor CGMII pošle výstupní transakci objektu Scoreboard a ten ji porovná s očekávanou výstupní transakcí. Pokud se shoduje, funkční verifikace pokračuje dál, jinak se zahlásí chyba.

Driver/Monitor MI32 se používá pro přístup ke stavovým a konfiguračním registrům výstupního síťového modulu skrze MI32 sběrnici.

Test cases je množina testovacích případů. Cílem je vytvořit takové testovací případy, které odhalí co nejvíce chyb.

Podle očekávání funkční verifikace odhalila množství chyb, které by byly v simulacích téměř nedohledatelné. Na konci funkčního verifikačního cyklu bezchybně prošlo přes výstupní síťový modul 7,2 milionu rámců. Ani tak ale není možno tvrdit, že komponenta je bezchybná. Toto tvrzení by mohla dát až formální verifikace. Nicméně se dá předpokládat, že většina chyb byla odhalena.

3.3.2 Ověření v hardwaru

Ověření v hardwaru spočívá ve vytvoření bitstreamu pro FPGA, jeho nahrání do akcelerační karty a následného testování za pomoci síťového analyzátoru. V průběhu testování se dále vyčítají stavové registry dostupné přes MI32 rozhraní. Na základě znalosti systému a výsledků ze síťového analyzátoru se zjistí, zdali systém v testu uspěl, či nikoliv.

V době psaní této bakalářské práce ještě nebyla k dispozici platforma NetCOPE pro kartu COMBO-100G, a proto byla využita testovací karta COMBO-80G. Výstup DMA modulu byl zapojen na vstup výstupního síťového modulu, jehož výstup byl zapojen na vstup vstupního síťového modulu. Výstup vstupního síťového modulu byl zapojen na vstup DMA modulu, čímž se uzavřela smyčka.

Pro testování byly použity softwarové nástroje platformy NetCOPE. Nástrojem `sze2write` byly odeslány pakety do karty, které se následně příjmuly nástrojem `sze2read`. Počet přijatých paketů pomocí nástroje `sze2read` byl porovnán s čítači výstupního a vstupního síťového modulu. Úspěšně se podařilo ověřit výstupní síťový modul v hardwaru.

Kapitola 4

Vysílání paketů

Ještě dříve než se síťová infrastruktura a její jednotlivé prvky uvedou do ostrého provozu, je nutno je řádně otestovat na plné rychlosti linky. Testování síťových prvků spočívá v přehrávání nebo generování síťového provozu, který se posílá na vstup testovaného zařízení. Výstup tohoto zařízení se zachytává pro další analýzu. Tento problém lze řešit třemi způsoby: softwarovými nástoji, speciálním hardwarovým testovacím zařízením nebo pomocí přídavné karty s programovatelným FPGA čipem do běžných serverových stanic.

První přístup lze realizovat pomocí volně dostupných softwarových nástojů (`tcpdump` pro zachytávání a `tcpreplay` pro přehrávání) a vyniká tak svou dostupností. Bohužel však toto řešení není schopno dosáhnout plné rychlosti linky a nelze zaručit přesný čas odeslání dat [4]. Specializované hardwarové zařízení oba problémy řeší, ale výrobce si je nechá také patřičně zaplatit, což je činí často nedostupnými. Alternativou je řešení založené na technologii FPGA. Časově kritické operace jsou zde implementovány přímo v hardware, a je tak zajištěná dostatečná rychlost zpracování. Přitom cena přídavných FPGA karet je oproti specializovanému hardware řádově nižší.

Na posledním uvedeném přístupu je založen systém pro zachytávání a vysílání paketů, který bude vytvořen v rámci této práce. Toto řešení je navrženo pro kartu COMBO-100G (podkapitola 2.2.1) a platformu NetCOPE (podkapitola 2.3). Vysílání může být řízeno pomocí přesných časových značek (časově kritické aplikace), nebo pomocí předem přednastavené rychlosti linky.

V této kapitole bude popsán návrh (podkapitola 4.1), implementace (podkapitola 4.2) a testování (podkapitola 4.3) systému pro zachytávání a vysílání paketů. V návrhu budou nejdříve definovány požadavky a poté popsána samotná architektura včetně popisu jednotlivých podkomponent. Implementace bude provedena v jazyce VHDL a budou diskutovány dosažené výsledky. Neméně důležitou částí je testování. Jednotlivé podkomponenty budou otestovány v simulacích, některé budou i funkčně verifikovány a celý systém bude otestován v hardwaru.

4.1 Návrh

Na základě znalostí platformy NetCOPE pro kartu COMBO-100G a problematiky testování síťových prvků byla navržena architektura systému pro zachytávání a vysílání síťového provozu. V této podkapitole budou nejdříve zmíněny požadavky na systém pro zachytávání a vysílání paketů, detailně popsána jeho architektura, adresový prostor stavových a konfiguračních registrů včetně jejich popisu a jednoduchá instrukční sada tohoto systému.

4.1.1 Požadavky

Ještě před samotným návrhem je třeba si ujasnit požadavky na navrhovaný systém. Požadavky jsou rozděleny do dvou skupin – požadavky na rozhraní a požadavky na funkce.

Požadavky na rozhraní:

- FLU rozhraní od/do vstupního/výstupního síťového modulu
- FLU rozhraní od/do DMA modulu
- MI32 rozhraní pro přístup ke stavovým a konfiguračním registrům
- rozhraní od statické paměti QDR pro čtení/zápis síťového provozu
- rozhraní od jednotky pro generování přesných časových značek

Požadavky na funkce:

- zachytávání síťového provozu ze síťového rozhraní
- zachytávání síťového provozu ze souboru typu PCAP
- vysílání síťového provozu na síťové rozhraní
- vysílání síťového provozu do souboru typu PCAP
- podpora vysílání podle časových značek s nanosekundovou přesností
- podpora vysílání podle předem přednastavené rychlosti linky
- uložení síťového provozu v externí paměti akcelerační karty
- ovládání pomocí instrukcí

Podpora vysílání podle časových značek znamená, že každému paketu musí být přiřazena časová značka. Na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET se každému paketu na rozhraní FLU předřazuje hlavička, která mimo jiné obsahuje časovou značku. Přesný formát hlavičky popisuje tabulka 4.1.

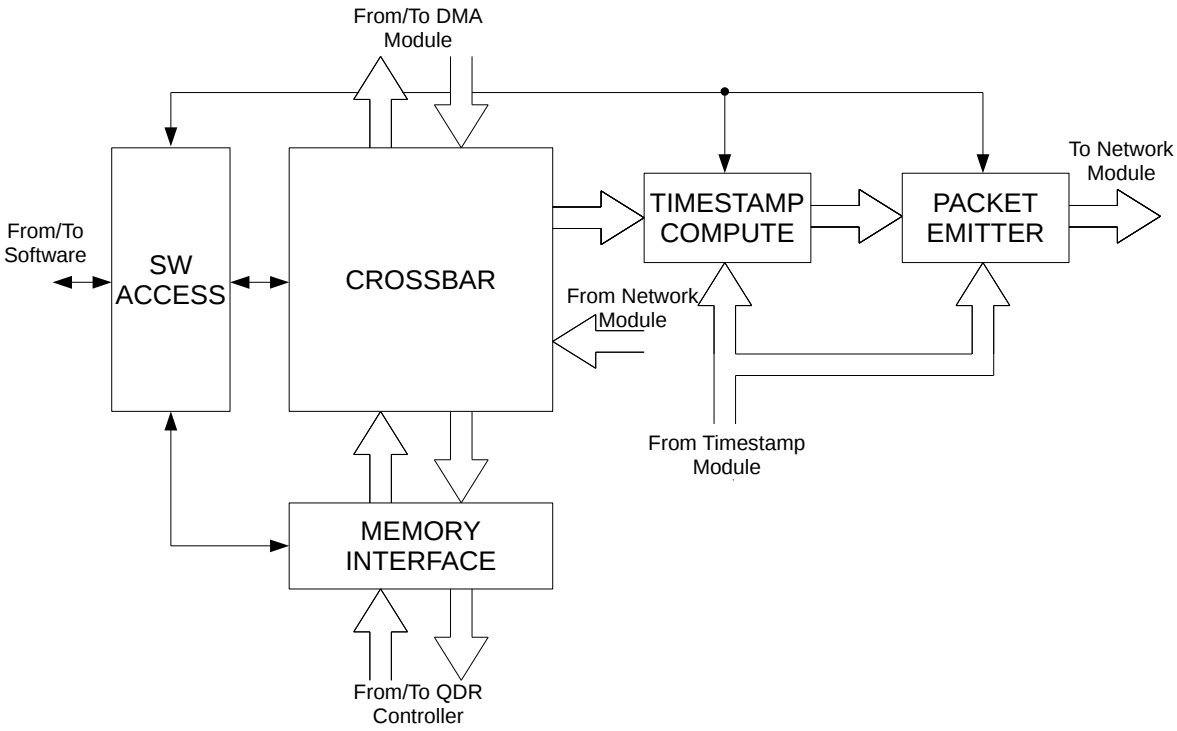
Od (b)	**Do (b)**	**Význam**
0	15	Velikost paketu včetně hlavičky v bytech
16	31	Velikost hlavičky v bytech od 32. bitu
32	63	Rezervováno
64	95	Počet uplynulých nanosekund v dané sekundě
96	127	Počet sekund od 1.1.1970

Tabulka 4.1: Popis hlavičky

První osmice bytů se používá pro rychlé DMA přenosy z akcelerační karty do paměti hostitelského zařízení. Druhá osmice obsahuje časovou značku s nanosekundovou přesností. Princip generování časových značek popisuje technická zpráva [11].

4.1.2 Architektura

Na základě znalostí z kapitoly 2 a požadavků z předchozí podkapitoly vznikla architektura pro zachytávání a vysílání síťového provozu, jejíž schéma je na obrázku 4.1. Navržený systém je škálovatelný na více síťových rozhraní pouhým zopakováním části schématu (TIMESTAMP COMPUTE a PACKET EMITTER).



Obrázek 4.1: Schéma systému pro zachytávání a vysílání síťového provozu

Byly využity výhody modulárního přístupu. Jednotlivé komponenty byly nezávisle na sobě navrženy, implementovány a testovány. Velký důraz byl tedy kladen na rozhraní komponent, aby nenastal problém při následné integraci. Nyní budou do podrobna popsány jednotlivé komponenty.

Jednotka MEMORY INTERFACE

Je abstrakcí externích statických pamětí QDR. K aplikaci v rámci platformy NetCOPE je tato jednotka připojena pomocí rozhraní FLU, což umožňuje přímé spojení se síťovým modulem a DMA modulem.

K externím statickým pamětem QDR je tato jednotka připojena skrze řadič QDR vygenerovaný nástrojem COREGEN společnosti Xilinx, jehož rozhraní je blízké nativnímu rozhraní samotné paměti. Pro dosažení propustnosti přes 100 Gb/s je nutno použít 3 QDR paměti. (Podle podkapitoly 2.2.1 by stačily 2, ale frekvence 700 MHz je spíše teoretická.)

Jednotka se skládá ze dvou podjednotek - `flu2qdr` a `qdr_adapter`. Podjednotka `flu2qdr` převádí 3 FLU datová slova o šířce 512 b včetně kontrolních signálů na 2 datová slova o šířce 864 b. Efektivita tohoto řešení se vypočte vztahem:

$$\frac{3 * (512 + 11)}{2 * 864} * 100\% = 90\,\%. \tag{4.1}$$

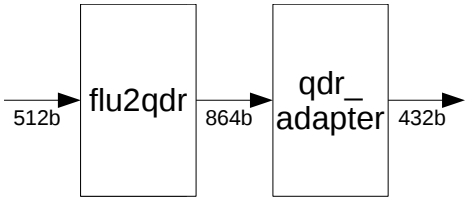
Naivní způsob by spočíval v uložení jednoho FLU slova o šířce 512 b včetně kontrolních signálů na 1 datové slovo o šířce 864 b. Efektivita tohoto neefektivního řešení se vypočte vztahem:

$$\frac{512 + 11}{864} * 100\% = 60\,\%. \tag{4.2}$$

Efektivní způsob ukládání dat 4.1 tedy využije statickou pamět QDR o 30 % lépe než neefektivní způsob 4.2.

Podjednotka `qdr_adapter` serializuje datová slova o šířce 864 b s poměrem 1:2 na datová slova o šířce 432 b. Výstupní rozhraní podjednotky `qdr_adapter` už odpovídá rozhraní 3 řadičů pamětí QDR, vygenerovaných nástrojem COREGEN společnosti Xilinx.

Jednotka MEMORY INTERFACE pracuje v několika režimech. Umí ukládat síťový provoz do externí paměti, číst síťový provoz z externí paměti (i opakovaně v cyklu) a najednou ukládat i číst síťový provoz z externí paměti, čímž se z externích pamětí stane fronta typu FIFO. Nechybí možnost zachycený síťový provoz v externích pamětech vymazat. Vnitřní struktura jednotky MEMORY INTERFACE je zobrazena na obrázku 4.2.



Obrázek 4.2: Vnitřní struktura jednotky MEMORY INTERFACE

Jednotka CROSSBAR

Dynamicky propojuje datové cesty. Tato jednotka je obousměrně propojena rozhraním FLU s DMA modulem, síťovým modulem (Network Module) a jednotkou MEMORY INTERFACE. Všechny smysluplné kombinace propojení budou popsány v podkapitole 4.1.4.

Jednotka TIMESTAMP COMPUTE

Vypočte před odesláním paketů nové časové značky pro řízení odesílání tak, že ke všem časovým značkám v zachyceném síťovém provozu připočte stejnou konstantu odpovídající rozdílu mezi první časovou značkou v zachyceném provozu a aktuálním časem. Původní časové rozestupy mezi zachycenými pakety tak zůstanou zachovány. Díky této jednotce není nutno časové značky přepočítávat v softwaru – tato funkce je hardwarově akcelerována.

Jednotka PACKET EMITTER

Posílá pakety do síťového modulu s respektováním časových značek nebo přednastavené rychlosti odesílání.

Jednotka SW ACCESS

Slouží pro obousměrnou komunikaci s hostitelskou stanicí. Je to aplikačně specifický procesor s velmi jednoduchou instrukční sadou a stavovými a konfiguračními registry, který

řídí celý systém na základě požadavků ze softwaru. Tyto registry jsou přístupné přes rozhraní MI32 a jejich absolutní adresa je dána bázovou adresou aplikace v rámci platformy NetCOPE a posunutím adres jednotlivých registrů vůči této adrese.

4.1.3 Adresový prostor

Adresový prostor je popsán tabulkou 4.2 a hned za ní jsou popsány jednotlivé registry. Nevyužité bity registrů jsou rezervovány pro budoucí použití.

Adresa	Registr	Velikost [B]	Čtení/Zápis
0x00	State Register	4	Čtení
0x04	Command Register	4	Zápis
0x08	Memory Start Pointer	4	Čtení
0x0C	Memory End Pointer	4	Čtení
0x10	Memory Pointer	4	Čtení
0x14	Iterations Register	4	Čtení/Zápis
0x18	Storage State Register	4	Čtení
0x1C	Preset Speed Register	4	Čtení/Zápis
0x20	Delta Timestamp Register	8	Čtení
0x28	In Packet Counter	4	Čtení
0x2C	Out Packet Counter	4	Čtení
0x30	Timestamp Register	8	Čtení

Tabulka 4.2: Adresový prostor systému pro zachytávání a vysílání síťového provozu

State Register identifikuje stav systému. Zaneprázdněnost systému indikuje 0. bit (platný v logické 1). V tomto stavu systém není schopen zpracovat další instrukci. Prázdnost externí paměti signalizuje 1. bit (platný v logické 1). Plnost externí paměti signalizuje 2. bit (platný v logické 1). Zkalibrování externích pamětí QDR signalizuje 3. bit (platný v logické 1). Na bitech 8-15 je uložena právě vykonávaná instrukce, což umožňuje detekovat nekorektní činnost konečného automatu v jednotce SW ACCESS. Instrukční sada je popsána v podkapitole 4.1.4. Jednotlivé bity tohoto registru jsou ještě pro větší přehlednost popsány v tabulce 4.3.

Od (b)	Do (b)	Význam
0	0	Příznak zaneprázdněnosti
1	1	Příznak prázdnosti externí paměti
2	2	Příznak plnosti externí paměti
3	3	Příznak připravenosti QDR pamětí
4	7	Rezervováno
8	15	Právě vykonávaná instrukce
16	31	Rezervováno

Tabulka 4.3: Popis registru State Register

Command Register slouží k zapisování instrukcí přes MI32 rozhraní z hostitelské stanice. Instrukční sadě se věnuje podkapitola 4.1.4.

Memory Start Pointer je ukazatel na počáteční adresu v externí paměti. Tento ukazatel slouží ke čtení datových slov. Při čtení se postupně zvyšuje až na hodnotu ukazatele Memory End Pointer.

Memory End Pointer je ukazatel na koncovou adresu v externí paměti. Tento ukazatel slouží k zápisu datových slov. Při zápisu se postupně zvyšuje až na hodnotu ukazatele Memory Start Pointer zmenšeného o jedničku.

Memory Pointer je pomocný ukazatel do externí paměti. Tento ukazatel slouží ke čtení datových slov v případě, kdy se data z externí paměti budou číst opakovaně.

Iterations Register je parametr pro instrukce, které čtou síťový provoz opakovaně. Tento registr udává počet iterací čtení z externí paměti.

Storage State Register identifikuje stav jednotky MEMORY INTERFACE. Seznam stavů shrnuje tabulka 4.4. Registr umožňuje detekovat nekorektní stav jednotky.

Hodnota	Název	Význam
0x0	STORAGE_DISABLE	deaktivováno, čtení i zápis z/do externí paměti je zakázáno
0x1	STORAGE_FIFO	fronta FIFO, současné čtení i zápis z/do externí paměti
0x2	STORAGE_CAPTURE	zachytávání, zápis do externí paměti
0x3	STORAGE_REPLAY	čtení, čtení z externí paměti
0x4	STORAGE_REPLAY_REPEATED	opakované čtení, opakované čtení z externí paměti
0x5	STORAGE_CLEAR	smazáno, smaže externí pamět a přejde do stavu deaktivováno

Tabulka 4.4: Popis stavů jednotky MEMORY INTERFACE

Preset Speed Register slouží k přednastavení rychlosti vysílání. Přípustné hodnoty jsou 0-100, které odpovídají procentuálnímu využití linky.

Delta Timestamp Register obsahuje časovou značku, která je rovna rozdílu mezi aktuálním časem a časem příjmu prvního paketu síťového provozu. Tato hodnota odpovídá konstantě, kterou přičítá jednotka TIMESTAMP COMPUTE k časovým značkám paketů při odesílání.

In Packet Counter je čítač příchozích paketů do systému z DMA modulu. Čítač příchozích paketů ze vstupního síťového modulu lze vyčíst nástrojem `ibufctl`.

Out Packet Counter je čítač odchozích paketů ze systému do DMA modulu. Čítač odchozích paketů do výstupního síťového modulu lze vyčíst nástrojem `obufctl`.

Timestamp Register obsahuje časovou značku, která je rovna aktuálnímu času.

4.1.4 Instrukční sada

Systém pro zachytávání a vysílání síťového provozu je řízen pomocí instrukcí. Instrukce posílá hostitelské zařízení skrze MI32 rozhraní do registru Command Register v jednotce SW ACCESS. Tato jednotka provede načtení, dekódování a vykonání instrukce. Všechny instrukce jsou uvedeny v tabulce 4.5, za níž následuje detailní popis každé instrukce.

Hodnota	Název
0x0	DISABLE
0x1	RESET
0x2	CAPTURE_IBUF
0x3	CAPTURE_DMA
0x4	REPLAY_OBUF
0x5	REPLAY_OBUF_TS
0x6	REPLAY_OBUF_PS
0x7	REPLAY_DMA
0x8	REPLAY_REPEATED_OBUF
0x9	REPLAY_REPEATED_OBUF_TS
0xA	REPLAY_REPEATED_OBUF_PS
0xB	REPLAY_REPEATED_DMA
0xC	NIC
0xD	NIC_FIFO
0xE	LOOPBACK
0xF	LOOPBACK_FIFO

Tabulka 4.5: Seznam instrukcí

DISABLE instrukce pozastaví zachytávání i vysílání síťového provozu tím, že uvede jednotku MEMORY INTERFACE do stavu STORAGE_DISABLE. Síťový provoz uložený v externí paměti zůstává nedotčen.

RESET instrukce vymaže externí paměť tím, že uvede jednotku MEMORY INTERFACE do stavu STORAGE_CLEAR. Deaktivuje vysílání podle časových značek a podle předem přednastavené rychlosti linky, čímž dojde k vyprázdnění veškerých front síťového provozu.

CAPTURE_IBUF instrukce zahájí zachytávání síťového provozu ze vstupního síťového modulu do externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup vstupního síťového modulu (IBUF) spojen se vstupem jednotky MEMORY INTERFACE. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_CAPTURE.

CAPTURE_DMA instrukce zahájí zachytávání síťového provozu z modulu DMA do externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup DMA modulu spojen se vstupem jednotky MEMORY INTERFACE. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_CAPTURE.

REPLAY_OBUF instrukce zahájí vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY.

REPLAY_OBUF_TS instrukce zahájí vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti podle časových značek. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY. Dále je aktivována jednotka TIMESTAMP COMPUTE, která přepočítá časové značky vzhledem k aktuálnímu času. Jednotka PACKET EMITTER je aktivována do stavu, kdy odesílá síťový provoz podle časových značek.

REPLAY_OBUF_PS instrukce zahájí vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti podle předem přednastavené rychlosti linky. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY. Jednotka PACKET EMITTER je aktivována do stavu, kdy odesílá síťový provoz podle předem přednastavené rychlosti linky v registru Preset Speed Register.

REPLAY_DMA instrukce zahájí vysílání síťového provozu do modulu DMA z externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem modulu DMA. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY.

REPLAY_REPEATED_OBUF instrukce zahájí opakované (počet iterací je dán hodnotou registru Iterations Register) vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY_REPEATED.

REPLAY_REPEATED_OBUF_TS instrukce zahájí opakované (počet iterací je dán hodnotou registru Iterations Register) vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti podle časových značek. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY_REPEATED. Dále je aktivována jednotka TIMESTAMP COMPUTE, která přepočítá časové značky vzhledem k aktuálnímu času. Jednotka PACKET EMITTER je aktivována do stavu, kdy odesílá síťový provoz podle časových značek.

REPLAY_REPEATED_OBUF_PS instrukce zahájí opakované (počet iterací je dán hodnotou registru Iterations Register) vysílání síťového provozu do výstupního síťového modulu (OBUF) z externí paměti podle předem přednastavené rychlosti linky. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem výstupního síťového modulu. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY_REPEATED. Jednotka PACKET EMITTER je aktivována do

stavu, kdy odesílá síťový provoz podle předem přednastavené rychlosti linky v registru Preset Speed Register.

REPLAY_REPEATED_DMA instrukce zahájí opakované (počet iterací je dán hodnotou registru Iterations Register) vysílání síťového provozu do modulu DMA z externí paměti. Jednotka CROSSBAR je nastavena tak, aby byl výstup jednotky MEMORY INTERFACE spojen se vstupem modulu DMA. Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_REPLAY_REPEATED.

NIC instrukce uvede systém do režimu běžné síťové karty NIC. Jednotka CROSSBAR je nastavena tak, aby byl výstup vstupního síťového modulu (IBUF) spojen se vstupem modulu DMA a výstup modulu DMA se vstupem výstupního síťového modulu (OBUF).

NIC_FIFO instrukce uvede systém do režimu běžné síťové karty NIC s použitím externí paměti jako vyrovnávací fronty typu FIFO. Jednotka CROSSBAR je nastavena tak, aby byl výstup vstupního síťového modulu (IBUF) spojen se vstupem jednotky MEMORY INTERFACE, výstup jednotky MEMORY INTERFACE se vstupem modulu DMA a výstup modulu DMA se vstupem výstupního síťového modulu (OBUF). Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_FIFO.

LOOPBACK instrukce propojí vstupní a výstupní síťový modul. Jednotka CROSSBAR je nastavena tak, aby byl výstup vstupního síťového modulu (IBUF) spojen se vstupem výstupního síťového modulu (OBUF).

LOOPBACK_FIFO instrukce propojí vstupní a výstupní síťový modul s použitím externí paměti jako vyrovnávací fronty typu FIFO. Jednotka CROSSBAR je nastavena tak, aby byl výstup vstupního síťového modulu (IBUF) spojen se vstupem jednotky MEMORY INTERFACE a výstup jednotky MEMORY INTERFACE se vstupem výstupního síťového modulu (OBUF). Jednotka MEMORY INTERFACE je uvedena do stavu STORAGE_FIFO.

4.2 Implementace

Systém pro zachytávání a vysílání síťového provozu navržený v podkapitole 4.1 byl naimplementován v jazyce VHDL. Při implementaci byly využity některé komponenty vytvořené na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Konkrétně generické multiplexory pro rozhraní FLU a jednotky pro zřetězené zpracování na rozhraní FLU. V podkapitole 4.1.4 byla navržena množina instrukcí, ze které byla implementována její podmnožina. Přehled všech implementovaných instrukcí shrnuje tabulka 4.6.

I když množství instrukcí nebylo implementováno, i tak se jedná o významné rozšíření zadání práce. Navíc do budoucna lze zbývající instrukce snadno implementovat, neboť s nimi počítá návrh. Implementována také nebylo jednotka TIMESTAMP COMPUTE, a proto je nutné časové značky přepočítávat v softwaru.

Hodnota	Název	Implementováno	Požadováno
0x0	DISABLE	ANO	ANO
0x1	RESET	ANO	ANO
0x2	CAPTURE_IBUF	ANO	NE
0x3	CAPTURE_DMA	ANO	ANO
0x4	REPLAY_OBUF	ANO	ANO
0x5	REPLAY_OBUF_TS	ANO	NE
0x6	REPLAY_OBUF_PS	NE	NE
0x7	REPLAY_DMA	NE	NE
0x8	REPLAY_REPEATED_OBUF	ANO	NE
0x9	REPLAY_REPEATED_OBUF_TS	NE	NE
0xA	REPLAY_REPEATED_OBUF_PS	NE	NE
0xB	REPLAY_REPEATED_DMA	NE	NE
0xC	NIC	ANO	NE
0xD	NIC_FIFO	NE	NE
0xE	LOOPBACK	ANO	NE
0xF	LOOPBACK_FIFO	NE	NE

Tabulka 4.6: Popis instrukční sady

4.2.1 Výsledky syntézy a implementace

Naimplementovaný systém pro zachytávání a vysílání síťového provozu byl vysyntetizován nástrojem XST společnosti Xilinx verze 14.7 pro cílovou technologii Virtex-7 typ XC7VH580T. Výsledky syntézy jsou uvedeny v přehledné tabulce 4.7.

Zařízení	LUT	Registry	BRAM	Frekvence	Nástroj
XC7VH580T	7416 (2 %)	8433 (1 %)	26 (3 %)	280,504 MHz	XST 14.7

Tabulka 4.7: Výsledky syntézy, v závorce je uvedeno procentuální využití čipu

Dosažená frekvence přes 280 MHz znamená, že se s vysokou pravděpodobností podaří dosáhnout požadovaných 200 MHz i po implementaci. Implementace systému pro zachytávání a vysílání síťového provozu nemohla být bohužel provedena na cílové akcelerační kartě COMBO-100G, a to z důvodu nedostupnosti platformy NetCOPE pro tuto kartu, a proto byla použita testovací karta COMBO-80G. Nasazení na testovací kartě COMBO-80G přináší jisté kompromisy. Tato karta obsahuje pouze jednu statickou pamět QDR, a proto byly požadované 3 statické paměti QDR modelovány uvnitř čipu FPGA pomocí 6 BRAM. Nicméně rozhraní platforem NetCOPE pro karty COMBO-80G a COMBO-100G jsou téměř totožná, což znamená, že systém pro zachytávání s vysílání paketů může být plnohodnotně otestován i na testovací kartě COMBO-80G. Výsledky implementace na testovací akcelerační kartě COMBO-80G s čipem Virtex-7 XC7VX690T nástrojem Vivado2014.1 společnosti Xilinx jsou uvedeny v tabulce 4.8 (aplikace je systém pro zachytávání a vysílání síťového provozu). Výsledkem implementace je *bitstream*, který lze nahrát přímo do čipu FPGA.

Zklamáním je nízká dosažená frekvence 160 MHz. Dosažená frekvence systému pro zachytávání a vysílání síťového provozu po systéze 4.7 je 280 MHz, což znamená, že výsledná frekvence 160 MHz po implementaci je dána platformou NetCOPE, jejíž vývoj ještě není zcela uzavřen.

Část	LUT	Registry	BRAM	Frekvence
Aplikace	5851(1 %)	8537 (1 %)	32 (2 %)	160 MHz
NetCOPE	124311 (29 %)	108452 (13 %)	360 (25 %)	160 MHz
Aplikace+NetCOPE	130162 (30 %)	116989 (14 %)	392 (27 %)	160 MHz

Tabulka 4.8: Výsledky implementace, v závorce je uvedeno procentuální využití čipu

4.3 Testování

Obecné principy testování již byly popsány v kapitole 3.3, a proto se k nim již tato podkapitola nebude vracet. Při testování byly použity techniky simulace, funkční verifikace a ověření v hardwaru.

V simulacích byly ověřeny komponenty MEMORY INTERFACE, CROSSBAR, PACKET EMITTER a SW ACCESS. Všechny jednotky v simulacích uspěly, a proto se mohlo postoupit do další fáze testování.

Klíčovou jednotkou systému pro zachytávání a vysílání paketů je jednotka MEMORY INTERFACE, a proto byla i funkčně verifikována. Díky velké znovupoužitelnosti kódu funkčního verifikačního prostředí byly použity moduly pro funkční verifikaci front typu FIFO na rozhraní FLU vytvořené na Oddělení nástrojů pro monitoring a konfiguraci sdružení CESNET. Výsledné funkční verifikační prostředí odhalilo některé chyby a na konci funkčního verifikačního cyklu úspěšně prošlo jednotkou MEMORY INTERFACE přes 5 milionů transakcí.

Při testování v hardwaru byly použity softwarové nástroje platformy NetCOPE. Pro přístup ke konfiguračním a stavovým registrům slouží nástroj `csbus` (bázová adresa pro aplikaci v rámci platformy NetCOPE pro kartu COMBO-80G je 0x2000000). Vstupní síťový modul se ovládá nástrojem `ibufctl`, výstupní síťový modul nástrojem `obufctl`. Pro zápis síťového provozu ze souborů typu PCAP se používá nástroj `sze2write`. Při práci s tímto nástrojem je třeba zvýšené opatrnosti, neboť systém pro zachytávání a vysílání paketů předpokládá, že každému paketu předchází hlavička. Vkládání ale bohužel tento nástroj nepodporuje. Je tedy nutné tyto hlavičky do souboru typu PCAP doplnit např. v binárním editoru. Jinak je třeba počítat s tím, že prvních 16 bytů každého paketu bude v systému zahozeno. Při vysílání podle časových značek dále dojde ke špatné interpretaci dat, a proto se paket nemusí vůbec odeslat. Úpravy v binárním editoru by bylo žádoucí nahradit nějakým vhodným softwarovým nástrojem, jehož vytvoření ale nebylo součástí této práce. Bitstream pro FPGA se do karty nahrává nástrojem `csboot`. Na akcelerační kartě COMBO-80G byla ověřena funkčnost všech implementovaných instrukcí (viz. tabulka 4.6).

Kapitola 5

Závěr

Prvním cílem této bakalářské práce bylo vytvořit výstupní síťový modul pro technologii 100 Gb/s Ethernetu. Tento modul je určen pro platformu NetCOPE na kartě COMBO-100G. Druhým cílem bylo vytvořit hardwarově akcelerovanou aplikaci v rámci platformy NetCOPE, která bude schopna vysílat krátké vzorky síťového provozu. Nad rámec zadání byl místo ní vytvořen komplexní systém pro zachytávání a vysílání paketů, který pro ukládání provozu používá externí statické paměti QDR.

Nejdříve bylo nutno nastudovat technologii 100 Gb/s Ethernetu, včetně formátu Ethernetových rámců. Dále se bylo nuté seznámit s rodinou karet COMBO, obzvláště s kartou COMBO-100G, včetně FPGA čipu Virtex-7. Následně platformou pro rychlý vývoj hardwarově akcelerovaných aplikací NetCOPE, včetně v ní používaných protokolů a rozhraní.

Veškeré tyto znalosti byly použity pro návrh výstupního síťového modulu. Implementace tohoto modulu byla provedena v jazyce VHDL. Veškeré komponenty a podkomponenty totoho modulu byly už v průběhu implementačních prací otestovány v simulacích. Celý výstupní síťový modul byl ověřen pomocí funkční verifikace vytvořené v jazyce SystemVerilog. Bezchybně při ní prošlo přes 7,2 milionu transakcí. Nakonec došlo k ověření tohoto modulu ve fyzické implementaci v rámci platformy NetCOPE.

Dále byly nastudovány metodiky testování síťových prvků. Pro testování síťových prvků byla navržena hardwarově akcelerovaná aplikace v rámci platformy NetCOPE, která umožňuje zachytávání/vysílání síťového provozu přímo z/do síťového rozhraní, a nebo z/do souborů typu PCAP, umístěných v hostitelské stanici akcelerační karty. Vysílání může být řízeno podle časových značek s nanosekundovou přesností, nebo předem přednastavené rychlosti vysílání. Pro ukládání síťového provozu se používají externí statické paměti QDR. Tento systém pro zachytávání a vysílání síťového provozu byl prezentován na studentské konferenci EEICT [6], dokonce se i umístil na 3. místě ve své kategorii.

Systém pro zachytávání a vysílání paketů byl implementován v jazyce VHDL. Veškeré komponenty a podkomponenty tohoto systému byly už v průběhu implementačních prací otestovány v simulacích. Důležité komponenty byly ověřeny i pomocí funkční verifikace, vytvořené v jazyce SystemVerilog. Celý systém byl nasazen na testovací kartě COMBO-80G, a tím tak ověřen ve fyzické implementaci.

Další práce by mohla spočívat v dokončení implementace instrukční sady systému pro zachytávání a vysílání paketů a nasazení systému na cílové kartě COMBO-100G. Dále pak v přidání hardwarové podpory pro generování syntetického síťového provozu [12]. Pro zvýšení komfortu uživatelů tohoto systému by bylo vhodné vytvořit grafické uživatelské rozhraní.

Literatura

[1] How Many Things Are Currently Connected To The Internet of Things (IoT)? [online]. `http://www.forbes.com/sites/quora/2013/01/07/how-many-things--are-currently-connected-to-the-internet-of-things-iot/`, červenec 2013 [cit. 2014-04-28].

[2] Nové adaptéry pro 40G a 100G technologie [online]. `http://www.cesnet.cz/sdruzeni/zpravy/tiskove-zpravy/nove-adaptery-pro%--40g-a-100g-technologie/`, únor 2014 [cit. 2014-04-28].

[3] Cards [online]. `https://www.liberouter.org/technologies/card-test/`, [cit. 2014-04-28].

[4] Covington, G. A., Gibb, G., Lockwood, J., aj.: A Packet Generator on the NetFPGA Platform. In *17th IEEE Symposium on Field Programmable Custom Computing Machines*, FCCM '09, 2009, s. 235–238.

[5] Friedl, Š., Puš, V., Matoušek, J., Špinler, M.: Designing a Card for 100 Gb/s Network Monitoring. Technická Zpráva 7, CESNET, 2013.

[6] Hummel, V.: Packet Capture & Replay on 100 Gb/s Ethernet. In *20th Student Electrical Engineering, Information and Communication Technologies*, EEICT 2014, 2014, s. 233–235.

[7] IEEE Computer Society: *Virtual Bridged Local Area Networks*. IEEE, std 802.1q-2005 vydání, 2005.

[8] IEEE Computer Society: *Ethernet Frame Expansion*. IEEE, std 802.3as vydání, 2006.

[9] IEEE Computer Society: *IEEE Standard for Ethernet*. IEEE, std 802.3-2012 vydání, 2012.

[10] Martínek, T., Košek, M.: NetCOPE:Platform for Rapid Development of Network Applications. In *11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*, DDECS 2008, 2008, s. 1–6.

[11] Martínek, T., Žádník, M.: Precise Timestamp Generation Module and its Applications in Flow Monitoring. Technická Zpráva 13, CESNET, 2009.

[12] MATOUŠEK, J.: *Network Traffic Simulation and Generation*. Diplomová práce, Brno, FIT VUT v Brně, 2011.

[13] Matoušek, J.: *Implementace a verifikace vstupních a výstupních síťových bloků*. Bakalářská práce, Brno, FIT VUT v Brně, 2009.

[14] Pužmanová, R.: Ethernet s kapacitou 40 a 100 Gbit/s [online].
http://archiv.cesnet.cz/sdruzeni/napsali-o-nas/2010/09/201009_ETM.html,
říjen 2010 [cit. 2014-01-19].

Příloha A

Obsah DVD

Přiložené DVD obsahuje tyto adresáře:

- /bitstreams – bitstreamy pro FPGA
- /cap_rep – zdrojové kódy systému pro zachytávání a vysílání síťového provozu
- /doc – elektronická verze tohoto textu
- /obuf – zdrojové kódy výstupního síťového modulu s propustností 100 Gb/s
- /tex – zdrojové kódy tohoto textu v elektronické verzi