

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PŘIZPŮSOBENÍ PLATFORMY NETCOPE PRO KARTU NETFPGA

BAKALÁŘSKÁ PRÁCE

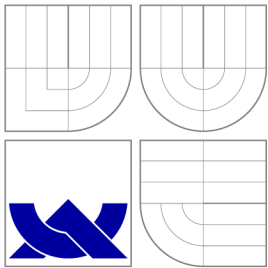
BACHELOR'S THESIS

AUTOR PRÁCE

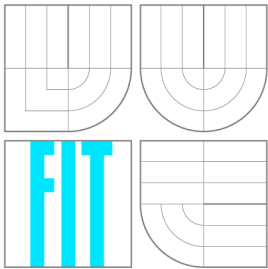
AUTHOR

KAREL KORANDA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PŘIZPŮSOBENÍ PLATFORMY NETCOPE PRO KARTU NETFPGA

ADAPTING THE NETCOPE PLATFORM FOR THE NETFPGA CARD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

KAREL KORANDA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVOL KORČEK

BRNO 2011

Abstrakt

Tato práce se zabývá přizpůsobením platformy NetCOPE určené pro rychlou tvorbu hardwarově akcelerovaných síťových aplikací pro karty NetFPGA-10G. Obecně seznamuje s cílovou technologií FPGA čipů a popisuje rozdíly mezi kartami rodiny COMBOv2 a NetFPGA. Cílem práce je především navrhnout, provést a zdokumentovat nezbytné úpravy platformy NetCOPE, testování cílového hardwarového zařízení s již přizpůsobenou platformou a zhodnocení možností vložení aplikačního jádra.

Abstract

This thesis deals with adapting the NetCOPE platform for rapid development of hardware-accelerated network applications for the NetFPGA-10G card. Its general purpose is to introduce target technology of FPGA chips and describe differences between the families of COMBOv2 and NetFPGA cards. The goals of this thesis are to design, implement and document the necessary adjustments of the NetCOPE platform, as well as testing of target hardware device with already ported platform and evaluation of options for inserting the core application.

Klíčová slova

COMBOv2, NetCOPE, NetFPGA, programovatelný hardware, hardwarově akcelerovaná síťová aplikace

Keywords

COMBOv2, NetCOPE, NetFPGA, programmable hardware, hardware-accelerated network application

Citace

KORANDA, K. *Přizpůsobení platformy NetCOPE pro kartu NetFPGA*. Brno: FIT VUT v Brně, 2011. Bakalářská práce.

Přizpůsobení platformy NetCOPE pro kartu Net-FPGA

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavola Korčeka. Uvedl jsem všechny literární zdroje, které jsem při tvorbě této práce použil. Další informace mi poskytli kolegové z projektu Liberouter.

.....

Karel Koranda

16. května 2011

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu své bakalářské práce Ing. Pavolu Korčekovi za cenné připomínky a odborné rady, kterými přispěl k vypracování této práce. Zároveň bych rád poděkoval kolegům z projektu Liberouter, zvláště Ing. Vlastimilu Košařovi, za poskytnutí informací a programového vybavení.

© Karel Koranda, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	5
2 Technologie FPGA	7
2.1 Struktura FPGA čipu	8
2.1.1 Programovatelné logické bloky CLB	8
2.1.2 Vstupně-výstupní bloky IOB	9
2.1.3 Hodinový signál a příslušné moduly	9
2.1.4 Udržení konfigurace čipu	9
2.2 Virtex-5	9
2.3 Konfigurace čipu rodiny Virtex-5	10
2.3.1 Konfigurační posloupnost	10
2.3.2 Konfigurační módy	11
3 Hardwarové vybavení	14
3.1 Karty rodiny COMBOv2	14
3.1.1 Karta COMBO-LXT	14
3.1.2 Přídavné karty ke COMBO-LXT	15
3.2 Karty NetFPGA	16
3.2.1 Karta NetFPGA-10G	17
4 Platforma NetCOPE	18
4.1 Sběrnice a komunikační protokoly	19
4.1.1 Generická interní sběrnice	19
4.1.2 MI32	25
4.1.3 FrameLink a MultiLink	26
4.2 HW moduly platformy NetCOPE a jejich rozhraní	31
4.2.1 Síťový modul	31
4.2.2 DMA moduly	33
4.2.3 Modul uživatelské aplikace	35
4.3 Ostatní moduly	36
4.3.1 ID komponenta	36
4.3.2 Hodinový modul	37
4.3.3 Timestamp jednotka	37
4.4 Softwarová část platformy NetCOPE	38
4.4.1 Vrstva ovladačů	38
4.4.2 Vrstva knihoven	38
4.4.3 Nástroje	39

5	Přizpůsobení platformy NetCOPE na kartu NetFPGA	40
5.1	Portovaná aplikace	40
5.1.1	Přizpůsobení síťového modulu	41
5.1.2	Úprava DMA modulu	41
5.2	Úprava hodinového systému	41
5.2.1	Hodinový systém na NetFPGA	42
5.2.2	Charakter úpravy	42
5.3	Úprava konfiguračního systému FPGA čipu	42
5.3.1	Systém konfigurace na COMBOv2	42
5.3.2	Zvolené řešení na NetFPGA	43
5.4	Odpojení specifických částí pro COMBOv2	43
5.5	Úprava identifikace karty	43
5.6	Příprava a vytvoření designu	44
5.6.1	Požadavky	44
5.6.2	Výsledky překladu	44
6	Testování	45
6.1	Ověření funkčnosti karty NetFPGA-10G	45
6.1.1	NetFPGA-10G Production Test	45
6.1.2	NetFPGA-10G 10G Ethernet Interface Loopback Test	47
6.1.3	NetFPGA-10G 1G Ethernet Interface Loopback Test	47
6.2	Simulace úpravy hodinového modulu	47
6.3	Simulační verifikace síťové karty	48
6.4	Existující testy platformy NetCOPE	48
7	Závěr	49
A	Výsledky simulace úpravy hodinového modulu	54
B	Výsledky simulační verifikace síťové karty	55
C	Výsledky překladu - využití zdrojů	57
C.1	Výsledky pro NetFPGA-10G na frekvenci 200 MHz	57
C.2	Výsledky pro COMBOv2-10G2 na frekvenci 200 MHz	58
D	Obsah přiloženého CD	60

Seznam obrázků

2.1	Struktura čipu FPGA	7
2.2	Struktura CLB a SLICE	8
2.3	Posloupnost fází konfiguračního procesu FPGA Virtex-5	10
2.4	Schéma zapojení sériové konfigurace	11
2.5	Sériová konfigurace - zapojení Daisy Chain a Ganged	12
2.6	Schéma zapojení Master SelectMAP	13
2.7	Schéma zapojení Slave SelectMAP	13
3.1	Karta COMBO-LX155T s připojenou COMBOI-10G2 kartou	15
3.2	Karta NetFPGA-10G	16
4.1	Hardwarová architektura platformy NetCOPE	18
4.2	Struktura generické interní sběrnice	19
4.3	Formát hlaviček paketů přenášených po interní sběrnici	24
4.4	Znázornění struktury síťového modulu (pro 10 Gbps síťová rozhraní)	31
4.5	Softwarová architektura platformy NetCOPE	38
5.1	Síťová karta nad platformou NetCOPE	40
5.2	Znázornění struktury síťového modulu pro kartu NetFPGA-10G	41
6.1	Schéma zapojení karty pro Production test	46
6.2	Schéma zapojení testu 10G Ethernet Interface Loopback	47
A.1	Simulace hodinového modulu na ComboV2	54
A.2	Simulace hodinového modulu na NetFPGA	54

Seznam tabulek

4.1	Adresový prostor interní sběrnice	20
4.2	Signály rozhraní interní sběrnice	20
4.3	Rozhraní uživatelské komponenty pro připojení k IB	21
4.4	Signály BusMaster rozhraní	22
4.5	Typy transakcí a paketů přenášených na interní sběrnici	23
4.6	Signály rozhraní MI32	25
4.7	Rozhraní FrameLink	26
4.8	Pořadí bajtů na FrameLinku	27
4.9	Nové a upravené signály MultiLink oproti FrameLink	27
4.10	Adresový prostor síťového modulu	32
4.11	Adresový prostor registrů jednotky IBUF	33
4.12	Adresový prostor registrů jednotky OBUF	33
4.13	Konfigurační adresový prostor DMA modulu	34
4.14	Datový adresový prostor DMA modulu	34
4.15	Adresový prostor ID komponenty	36
5.1	Nastavení nových hodnot registrů ID komponenty	43
5.2	Využití zdrojů a maximální dosažená frekvence	44

Kapitola 1

Úvod

Jedním z trendů dnešní doby je kladení stále většího důrazu na zvyšování rychlosti počítačového zpracování dat. V důsledku tohoto jevu se ujímá myšlenka urychlení aplikací jejich hardwarovou formou, umožňující předzpracování dat a ulehčení práce CPU počítače. Typickou oblastí, ve které se neustále zvyšují požadavky na rychlost přenosu dat a softwarové zpracování začíná být nedostačující, jsou počítačové sítě. V současnosti se mnoho velkých společností začíná zaměřovat na výzkum a vývoj platform umožňujících jednoduchou tvorbu hardwarově akcelerovaných aplikací. S tím souvisí také rozšiřování takových technologií, které tento výzkum usnadňují, a mezi něž pochopitelně patří technologie programovatelných čipů. Odborníků, kteří však práci s těmito technologiemi a platformami rozumí, je poměrně nízký počet. Proto jsou tyto technologie často výhodně poskytovány vysokým školám pro výzkumnou, vývojovou a výukovou činnost.

Mezi organizace, které se zabírají zrychlováním zpracování dat na počítačových sítích je i CESNET z.s.p.o. [2]. Toto sdružení vytvořilo v rámci svého výzkumného záměru „Programovatelný hardware“ [8] a projektu Liberouter [7] platformu pro tvorbu hardwarově akcelerovaných síťových aplikací zvanou NetCOPE [14]. Tato platforma vznikala na speciálně navržených FPGA kartách, označovaných jako karty rodiny COMBOv2. Kromě sdružení CESNET se shodnou oblastí zabývá také vývojový tým NetFPGA [4], který navrhl modernější obdobu těchto karet – NetFPGA-10G. Pro tyto karty však není v současné době vytvořen dostatečný firmware, což je motivací pro tvorbu této práce.

Tato bakalářská práce se v rámci projektu *Inovační Voucher*, zastřešeného výzkumnou skupinou ANT@FIT [1], zabývá přizpůsobením platformy NetCOPE, určené pro vývoj hardwarově akcelerovaných síťových aplikací, na karty NetFPGA-10G. Cílem této práce je popsat platformu NetCOPE a navrhnout, provést a otestovat nezbytné úpravy tak, aby ji bylo možné používat na cílovém hardwaru a umožnit nad ním vývoj příslušného aplikačního jádra.

Kapitola 2 stručně seznamuje čtenáře se základní koncepcí technologie programovatelných čipů FPGA, stručně představuje řady čipů technologie Virtex-5 uvedenou na trh společností Xilinx, a orientuje se především na konfiguraci FPGA čipů této technologie. Kapitola 3 popisuje hardwarová zařízení, na kterých byla platforma NetCOPE vyvíjena (karty rodiny COMBOv2) a pro která bude určen výstup této práce (karta NetFPGA-10G). Následující kapitola 4 se zaměřuje na představení platformy NetCOPE, především její základní moduly, jednotky pro zpracování dat a komunikační protokoly a příslušná rozhraní pro napojení na sběrnice. Návrh a implementaci úprav nutných pro přizpůsobení platformy pro cílové NetFPGA karty popisuje kapitola 5, kapitola 6 se pak zabývá popisem základních testů a simulací pro ověření funkčnosti těchto úprav, včetně ověření hardwarového zařízení.

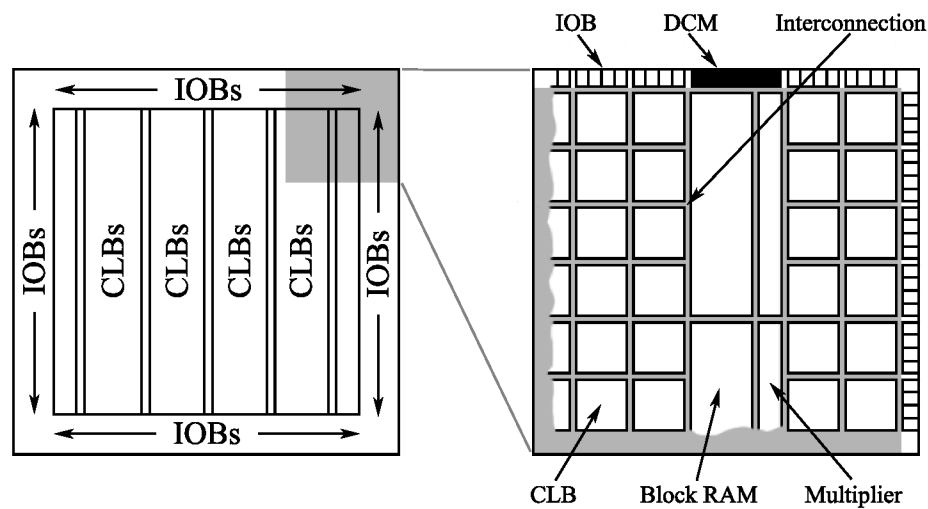
Kapitola 7 nakonec hodnotí přínos a další možnosti pokračování práce a možnosti vložení aplikačního jádra.

Kapitola 2

Technologie FPGA

Základem programovatelného hardwaru je speciální technologie čipů založená na integrovaných obvodech umožňujících změny vnitřní konfigurace podle požadavků vývojáře, přičemž obvykle je tato konfigurace či struktura specifikována jazyky pro popis hardwaru (tzv. *HDL – Hardware Description Languages*). Programovatelné obvody jsou souhrně označovány *PLD (Programmable Logic Device)* a z konkrétních typů technologie lze jmenovat například *SPLD (Simple PLD)*, *CPLD (Complex PLD)* a *FPGA (Field-Programmable Gate Array)*.

Technologie FPGA [19] je založena na generátorech logických funkcí realizovaných pomocí vyhledávacích tabulek označovaných *LUT (Look-Up Table)*, klopných obvodech a mnoha konfigurovatelných propojích, přičemž pro uchování konfigurace je nejčastěji používána technologie paměti *SRAM*. Zjednodušené schéma FPGA čipu je znázorněno na obrázku 2.1. Mezi největší výrobce FPGA patří společnosti Xilinx, Altera a Lattice Semiconductor.



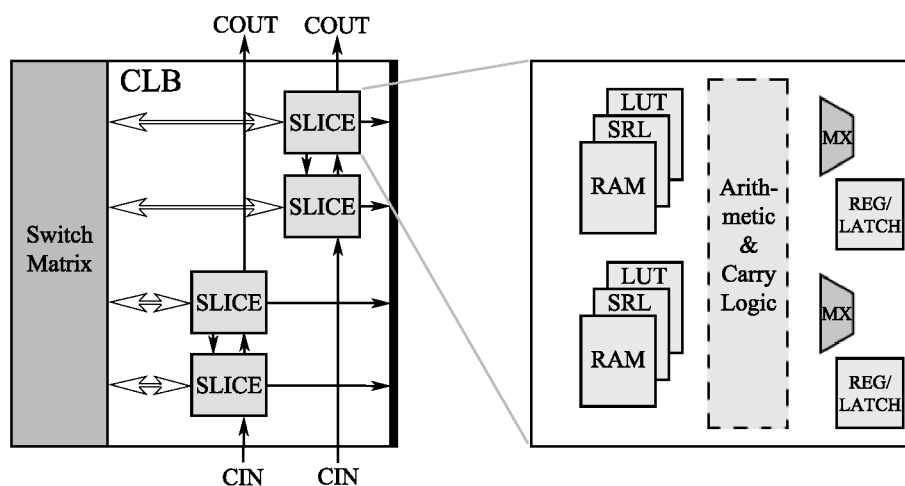
Obrázek 2.1: Struktura čipu FPGA

2.1 Struktura FPGA čipu

Základními stavebními kameny FPGA obvodu jsou programovatelné logické bloky *CLB* (*Configurable Logic Block*), programovatelné vstupně-výstupní bloky, označené jako *IOB* (*Input/Output Block*), a programovatelné horizontální a vertikální propoje. Důležitou součástí jsou také rozvody hodinového signálu s možností jejich úpravy pomocí speciálních integrovaných obvodů *DCM* (*Digital Clock Manager*) nebo *PLL* (*Phase-Locked Loop*). V dnešní době se na čipech FPGA vyskytují kromě těchto jednotek také blokové paměti *BlockRAM*, násobičky nebo dokonce ASIC procesory.

2.1.1 Programovatelné logické bloky CLB

Struktura CLB je znázorněna na obrázku 2.2. Je patrné, že tento obvod se skládá z menších logických jednotek označovaných jako *SLICE*. V závislosti na architektuře se jejich počet uvnitř CLB může lišit. Platí, že *SLICE* má více logických vstupů nežli výstupů a také vstupy a výstupy *carry* logiky umožňující konstrukci sčítaček. Propojení mezi jednotlivými CLB zajišťují programovatelné přepínací matice horizontálních a vertikálních propojů (*Switch Matrix*).



Obrázek 2.2: Struktura CLB a SLICE

Architektura SLICE

Na obrázku 2.2 je také zjednodušeně znázorněna architektura prvků *SLICE*. Jednotka *SLICE* obsahuje určitý počet generátorů logických funkcí *LUT*, které jsou realizovány pamětí, dále několik klopných obvodů (typu *Register/Latch* v závislosti na synchronizaci), multiplexorů, podpůrných aritmetickologických prvků (členy *AND*, *OR*) a *carry* logiky. Počet, propojení a vlastnosti jednotlivých prvků závisí na konkrétní technologii, například *SLICE* technologie FPGA Xilinx Spartan-3 obsahuje dvě *LUT* [31], Xilinx Virtex-5 čtyři [33]. Obdobně se mohou lišit i počty vstupů jednotlivých *LUT*, multiplexorů a podobně.

2.1.2 Vstupně-výstupní bloky IOB

Programovatelné vstupně-výstupní bloky FPGA reprezentují fyzické piny čipu připojené k dalším periferiím. Díky jejich schopnosti konfigurace lze určit, které piny jsou vstupní, výstupní či obousměrné. Taktéž je možné nastavit přítomnost registrů, které mohou poskytnout lepší časování a synchronizaci s okolím čipu. Při vytváření *konfiguračního bitstreamu*¹ FPGA je nutné taktéž určit, které piny patří ke vstupu/výstupu konkrétních navržených komponent.

2.1.3 Hodinový signál a příslušné moduly

Zdrojem hodinového signálu FPGA čipu je tzv. *krystal*, oscilátor, který generuje signál střídající hodnoty *log. 0* a *log. 1* s určitou frekvencí. Tento signál je na FPGA napojen pomocí speciální pinů k jednotkám, které ze signálu odstraní případné záskoky a zkraslení. Je důležité, aby hodinový signál byl rozveden rovnoměrně po čipu a zpravidla jsou tyto rozvody rychlejší, než běžné vodiče.

Součástí FPGA čipů dnes bývají i moduly umožňující úpravy hodinového signálu, například pro případ, že je nezbytné rozvést po čipu signál s jinou frekvencí, než má zdrojový krystal, případně pro tvorbu dalších, odvozených hodinových signálů pro některé komponenty. Jedná se o DCM nebo PLL obvody, které umožňují korekce zpoždění, násobení či dělení frekvence signálu, případně fázové posuny. Samozřejmě jsou i tyto moduly do určité míry programovatelné.

2.1.4 Udržení konfigurace čipu

O uložení konfigurace čipu se starají nejčastěji paměti SRAM, které umožňují neomezené množství přeprogramování. Jejich nevýhodou je podmínka udržení napájení, aby se data „neztratila“. Z tohoto důvodu jsou součástí vývojových desek s FPGA čipy paměti typu EEPROM nebo Flash, na nichž je konfigurační bitstream uložen pro případ odpojení napájení. Většina FPGA také podporuje konfiguraci přes programové rozhraní standardu JTAG (definováno v *IEEE Std 1532-2002* [11]).

O průběhu konfigurace FPGA se lze více dočíst v sekci 2.3, která se zaměřuje na konfiguraci čipů Xilinx Virtex-5.

2.2 Virtex-5

Technologie Virtex-5 [33] je produkt společnosti Xilinx. FPGA čipy této rodiny jsou rozdělené na pět platform, přičemž každá je rozdílně vybavena pro konkrétní požadavky na tvorbu hardwarově akcelerovaných aplikací. Tyto platformy jsou označovány jako *LX*, *LXT*, *SXT*, *TXT* a *FXT*. Bližší specifikaci jednotlivých řad a podrobné rozdíly lze najít v [29].

Řada LX je určena pro obecné logické aplikace.

Řada LXT je určena pro tvorbu logických aplikací s prostředky pro pokročilou sériovou komunikaci.

¹**Konfigurační bitstream** je soubor či posloupnost logických hodnot nesoucí informace o konfiguraci programovatelných obvodů FPGA čipu. Tento soubor je někdy označován taktéž jako *design* FPGA čipu.

Řada SXT poskytuje pokročilou sériovou konektivitu a prostředky pro aplikace na zpracování signálů.

Řada TXT je určena především pro vysokorychlostní telekomunikační a datové komunikační systémy.

Řada FXT se liší od ostatních řad tím, že má integrovány další vestavěné systémy, jako například ASIC mikroprocesory.

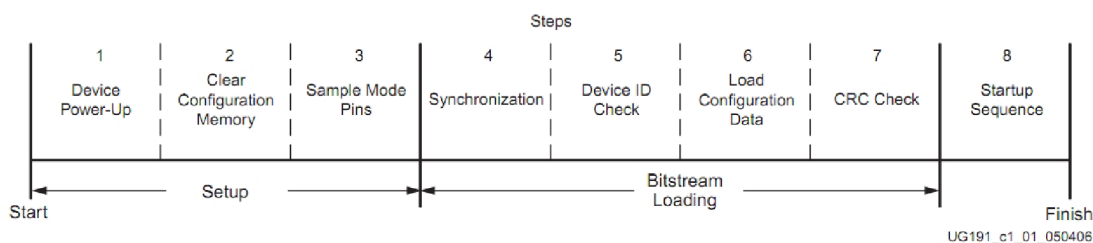
Čipy řad disponujících vybavením pokročilé sériové konektivity (tedy xXT) mají integrovány Ethernetové MAC bloky, čehož je v platformě NetCOPE pochopitelně využito při konstrukci síťového rozhraní. Dalším podstatným blokem, který je v rámci platformy využit, je vestavěný koncový (*Endpoint*) blok pro PCI Express sběrniceový systém. Na čipech se také nachází jednotka *SYSMON* (*System Monitor* [30]), která umožňuje sledování dostupnosti a spolehlivosti jednotlivých komponent čipu a jeho okolí.

2.3 Konfigurace čipu rodiny Virtex-5

Technologie čipů rodiny Virtex-5 umožňuje několik druhů konfigurací FPGA. Konfigurace spočívá v přenosu bitstreamu konfiguračních dat z určité paměti přes zvláštní rozhraní čipu. Pokud tento proces proběhne úspěšně, pak je vnitřní struktura a funkce FPGA nastavena příslušným způsobem.

2.3.1 Konfigurační posloupnost

Konfigurační proces se dá rozdělit na tři fáze, z nichž každá má definovaný počet kroků, které v ní musí být provedeny. V závislosti na vybraném způsobu konfigurace může mít v režii některé části návrhář (a již při návrhu zvažuje přítomnost příslušného mikrokontroleru nebo jiného dalšího pomocného čipu, který je schopen konfiguraci řídit) nebo si celý proces řídí samotné FPGA. Posloupnost celého procesu je znázorněna na obrázku 2.3.



Obrázek 2.3: Posloupnost fází konfiguračního procesu FPGA Virtex-5 [32]

Fáze Setup

Tento proces je stejný ve všech konfiguračních módech (2.3.2). Sestává se ze tří kroků:

Device Power-Up je krok, ve kterém je spuštěno napájení FPGA čipu.

Clear Configuration Memory provede vymazání konfigurační paměti čipu.

Sample Mode Pins je posledním krokem fáze Setup, jedná se o zjištění konfiguračního módu, na jehož základě se bude odvíjet další fáze.

Fáze Bitstream Loading

Tato fáze je nejpodstatnější částí celého procesu konfigurace FPGA, protože v ní probíhá přenos konfiguračních dat z paměti do čipu.

Synchronization je krok, který provádí detekci šířky datové sběrnice v závislosti na vybraném konfiguračním módu.

Check Device ID provede načtení ID konfigurovaného zařízení a kontrolu, zda požadovaná konfigurační data jsou pro tento typ zařízení.

Load Configuration Data Frames je nejdůležitějším krokem. Jak název napovídá, zde se provádí přenos konfiguračních dat do FPGA.

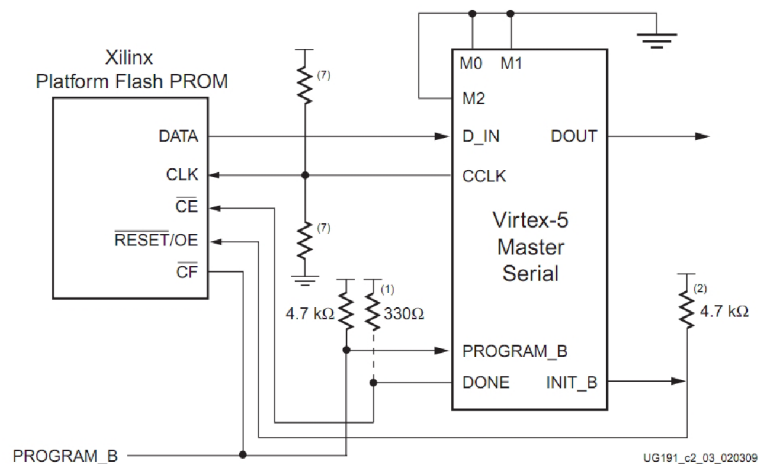
Cyclic Redundancy Check je ověření, zda byla přenesena korektní data.

Fáze Startup

Tato fáze má jen jeden krok. V této fázi se ukončí konfigurace a nastavením příslušných signálů FPGA oznámí, že je nakonfigurováno a připraveno k použití.

2.3.2 Konfigurační módy

Režim konfiguračních módů se rozděluje na *Master* a *Slave*. Režim Master zjednodušuje podstatným způsobem práci vývojáři, neboť přenos konfiguračních dat si řídí samotná FPGA. Je potřeba zmínit, že ačkoli zjednodušují vynaloženou práci, jejich výběr nemusí být vždy praktický nebo umožnitelný (záleží například na dalším konkrétním technologickém vybavení). Navíc vývojář nemá možnost do tohoto procesu zasáhnout. Naproti tomu Slave módy dosahují vyšších rychlostí a rozsáhlejších možností, vyžadují však přítomnost mikrokontroleru nebo automatu, který přenos bitstreamu řídí [32].



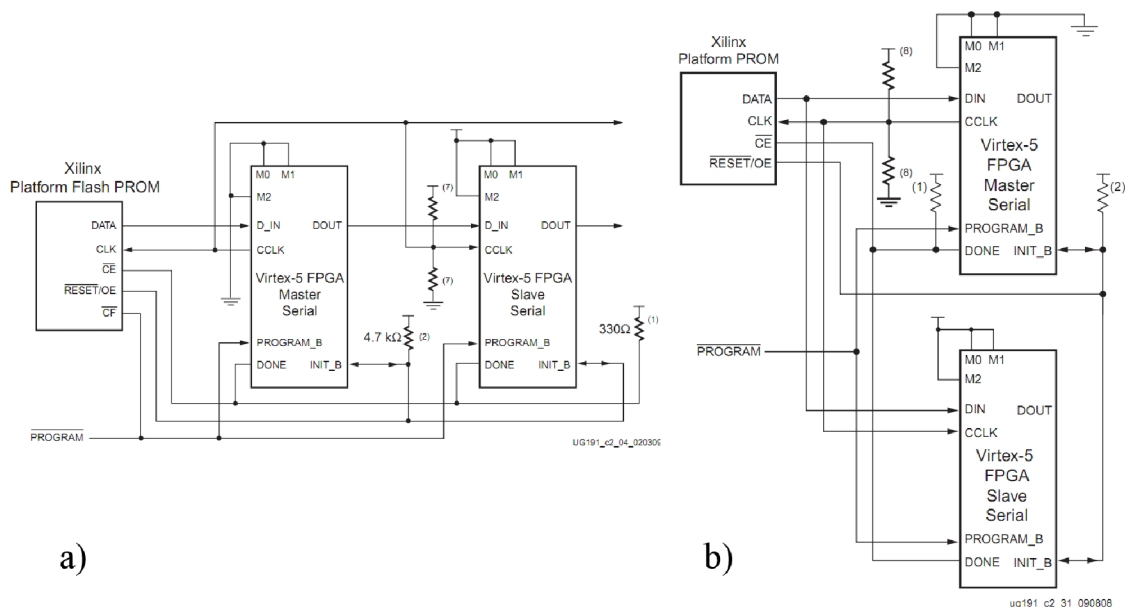
Obrázek 2.4: Schéma zapojení sériové konfigurace [32]

Podrobnější informace o konfiguraci FPGA rodiny Virtex-5 lze najít v [32], o Slave režimech pak také v [18] a o konfiguraci s využitím paměti Xilinx Platform Flash XL v [28].

Sériová konfigurace

Zapojení FPGA a paměti (zde je závislost na konkrétním typu paměti Xilinx Platform Flash), v níž je bitstream uložen, je znázorněno na obrázku 2.4. Data jsou přenášena bit po bitu, proto je tato konfigurace poměrně pomalá. V Master režimu si FPGA řídí přenos z paměti samo, zdrojem hodinového signálu je krystal na čipu. Naproti tomu ve Slave režimu je potřebná přítomnost externího mikroprocesoru nebo CPLD čipu, který bude přenos řídit. V takovém případě je zapotřebí také externí zdroj hodinového signálu.

Technologie Virtex-5 umožňuje sériovou konfiguraci také pro více čipů najednou – jedná se o tzv. zapojení *Daisy Chain*. V takovém případě je v Master módu jedno z FPGA, všechna ostatní jsou v režimu Slave. Schéma zapojení je znázorněno na obrázku 2.5 a). Podobným způsobem lze provést také zapojení označené *Ganged*, viz obrázek 2.5 b), při takovém zapojení však narozdíl od *Daisy Chain* budou mít všechny čipy po skončení procesu stejnou konfiguraci.



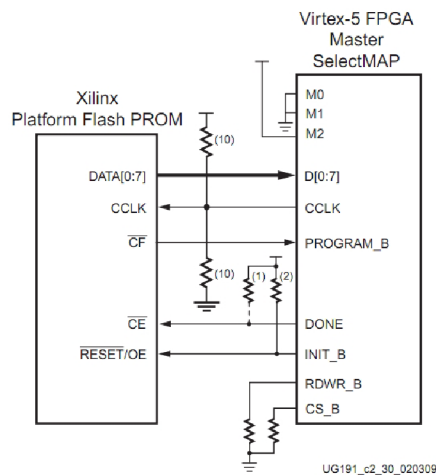
Obrázek 2.5: Sériová konfigurace: a) zapojení Daisy Chain, b) zapojení Ganged [32]

SelectMAP konfigurace

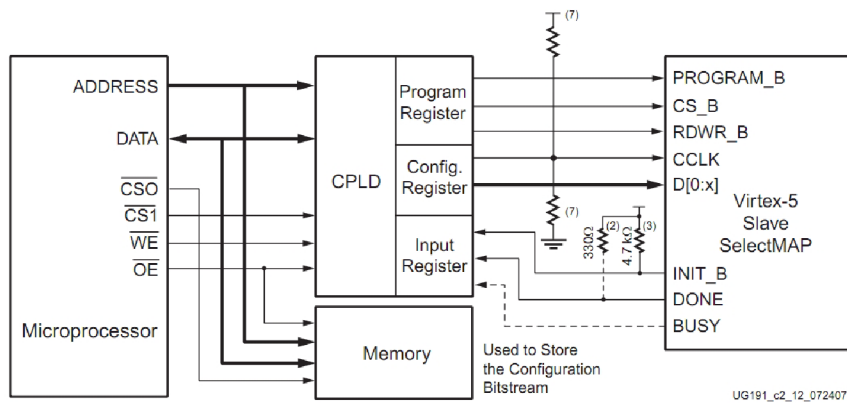
Konfigurace v režimu SelectMAP umožňuje paralelní přenos bitstreamu přes datovou sběrnici o šířkách 8, 16 nebo 32 bitů. Takový přenos je rychlejší, než výše zmíněný sériový přenos.

Obdobně jako pro sériovou konfiguraci i u tohoto typu platí rozdíly mezi Master a Slave režimy. Konfigurace SelectMAP pro jediné zařízení v Master režimu je závislé na uložení bitstreamu v paměti Platform Flash XL nebo Xilinx Platform Flash. Schéma zapojení zařízení v Master SelectMAP módu je znázorněno na obrázku 2.6, jedna z možností stejného zapojení v Slave režimu je na obrázku 2.7. SelectMAP konfigurační mód v režimu Slave je doporučen výrobcem [32].

Podobně jako v sériové konfiguraci, i v tomto případě je možné zapojení více zařízení jako *Daisy Chain*, resp. *Ganged* zapojení.



Obrázek 2.6: Schéma zapojení Master SelectMAP [32]



Obrázek 2.7: Schéma zapojení Slave SelectMAP [32]

SPI a BPI konfigurace

Módy typu SPI a BPI podporuje technologie Virtex-5 kvůli možnosti připojení i dalších typů pamětí, než jen Flash pamětí společnosti Xilinx.

Kapitola 3

Hardwarové vybavení

Cílem této práce je přizpůsobit platformu NetCOPE pro nové hardwarové karty NetFPGA. Původní hardwarové vybavení, na kterém byla platforma vyvíjena a pro které je primárně určena, jsou karty rodiny COMBOv2. Rozdíly mezi původní a cílovou architekturou jsou značné, nicméně díky tomu, že základní součástí obou rodin karet jsou FPGA čipy technologie Virtex-5, a díky dostatečné abstrakci platformy NetCOPE nebudou potřeba rozsáhlé koncepční úpravy. Rozdíly mezi zdrojovým a cílovým hardwarem budou popsány v této kapitole.

3.1 Karty rodiny COMBOv2

Karty rodiny COMBOv2 byly navrženy sdružením CESNET a jejími partnery. Hlavní myšlenkou vytvoření těchto karet je dát vývojářům možnost pracovat s hardwarem podobným způsobem a širokými možnostmi, jaké nabízí open-source software. Proto základem těchto karet je vždy jeden nebo více opětovně programovatelných čipů FPGA, konektory pro připojení paměťových karet a dalších nezbytných komponent, jako jsou síťové moduly a další periferie. Toto poskytuje vývojářům poměrně rozsáhlou oblast využití COMBOv2 karet, neboť díky technologii FPGA není problém v krátkém okamžiku změnit jejich konfiguraci na jinou hardwarově akcelerovanou aplikaci, což je možno využít u mnoha různých projektů. Příkladem takových projektů mohou být *NIFIC* [24], jehož cílem je vývoj hardwarového prostředku pro filtrování komunikace a směrování v počítačových sítích, nebo *Flexible FlowMon* [21], který se naopak zabývá vývojem sondy pro sběr statistik o datových tocích v počítačové síti (tzv. *NetFlow*). Také vývoj hardwarově akcelerovaných aplikací je jednodušší v důsledku snažšího a rychlejšího testování vytvářených designů. Stinnou stránkou těchto řešení jsou poněkud značné finanční náklady způsobené právě cenou potřebné FPGA technologie.

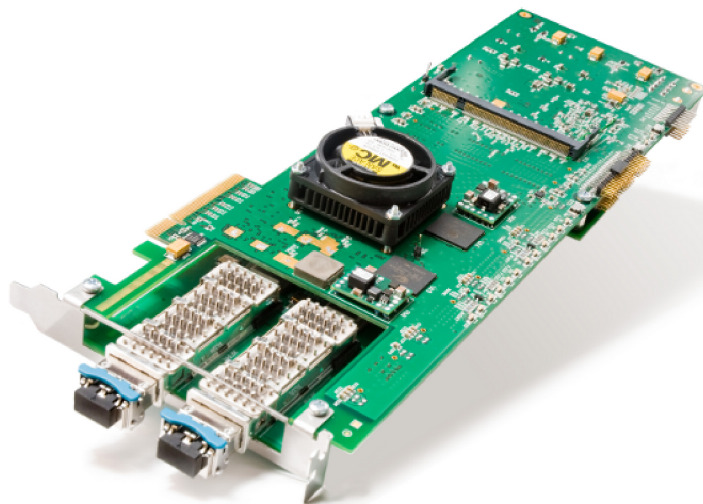
3.1.1 Karta COMBO-LXT

Karta COMBO-LXT¹ je základní deskou rodiny COMBOv2. Jedná se o kartu s PCI Express x8 připojením na základní desku počítače. Je vybavena těmito prvky:

- FPGA čip Virtex-5 XC5VLX110T (případně FX100T či LX155T),

¹Označení **LXT** nese karta z důvodu přítomnosti FPGA čipu Virtex-5 řady LXT. Pokud karta obsahuje čip řady FXT, je označena odpovídajícím způsobem.

- FPGA čip Spartan-3 XC3S1200E,
- dvě paměti QDRII RAM CY7C1513AV18,
- SODIMM konektor pro DDR2 paměť (až do velikosti 2 GB),
- čtyři nízkorychlostní konektory LSC s datovou propustností až 8 Gbps,
- dva vysokorychlostní konektory IFC s datovou propustností až 28 Gbps obousměrně.



Obrázek 3.1: Karta COMBO-LX155T s připojenou COMBOI-10G2 kartou

3.1.2 Přídavné karty ke COMBO-LXT

Karta COMBO-LXT v základu neobsahuje žádné síťové konektory, ale umožňuje připojení přídavných karet rodiny COMBOv2. To umožňuje poměrně rozsáhlé možnosti modularity, neboť díky tomuto systému lze k programovatelnému FPGA čipu připojit prakticky jakoukoli periférii. V současné době se využívá hlavně přídavných karet zaměřených na síťovou komunikaci, teoreticky je ale možné vytvořit a připojit kartu s grafickým či jiným rozhraním.

Karta COMBOI-10G2

Karta COMBOI-10G2 obsahuje dva 10 Gbps síťové porty. Karta obsahuje také digitální senzory pro snímání teploty.

Karta COMBOI-1G4

Karta COMBOI-1G4 obsahuje čtyři 1 Gbps síťové porty. Na této kartě jsou také integrovány dva digitální teplotní senzory.

Karta COMBOL-GPS

Karta COMBOL-GPS je vytvořena pro umožnění přijímání GPS signálu. Je potřeba pro vytváření přesných časových značek, které mají značné uplatnění v síťové komunikaci.

Karta COMBOI-TEST

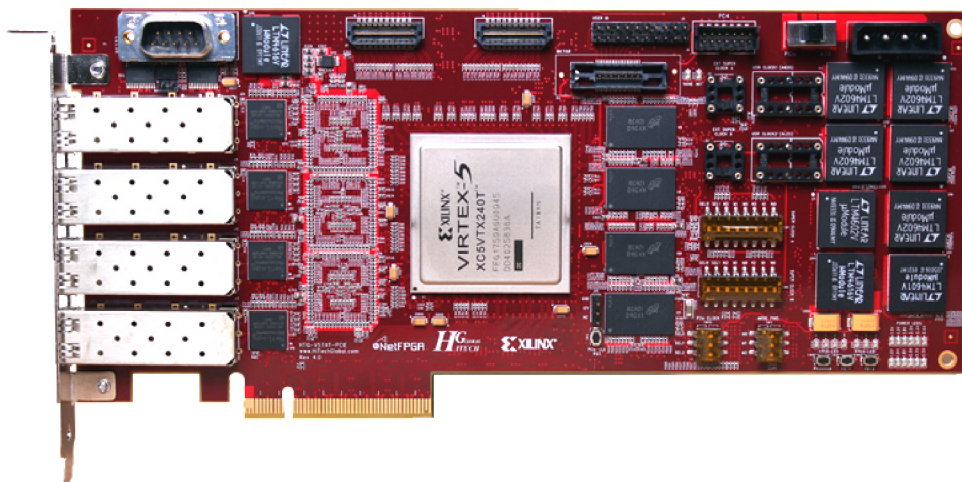
Tato karta slouží jako testovací pro ověření funkčnosti IFC a LSC konektorů karty COMBO-LXT. Její uplatnění ve vytváření aplikací nad platformou NetCOPE není tedy příliš velké.

Karta COMBOI-10G4TXT

Karta COMBOI-10G4TXT je nejnovější přídatnou kartou rodiny COMBOv2. Kromě čtyř 10 Gbps síťových portů obsahuje také další FPGA čip – Virtex-5 XC5VTX150T. Kromě jiného tato karta má také integrovaný konektor IFC umožňující připojení další přídatné karty se čtyřmi 1 Gbps nebo jedním 10 Gbps síťovým portem. Díky dalšímu FPGA čipu umožňuje rozsáhlejší možnosti vytváření aplikací nad platformou NetCOPE, než jakou poskytují výše popsané karty.

3.2 Karty NetFPGA

Obdobně jako karty rodiny COMBOv2 jsou i karty NetFPGA určeny pro vývoj hardwarově akcelerovaných síťových aplikací. Narozdíl od již představených karet nejsou karty NetFPGA modulovatelné – síťové konektory se nenachází na žádné další přídatné kartě. To má i své výhody, při zapojení do PCI Express slotu počítače zabírají podstatně méně místa. V současné době existují dvě varianty karet NetFPGA – starší NetFPGA-1G, kterou se tato práce nebude zabývat, a novější NetFPGA-10G.



Obrázek 3.2: Karta NetFPGA-10G

3.2.1 Karta NetFPGA-10G

Karta NetFPGA-10G (obrázek 3.2) obsahuje čtyři 10 Gbps síťové porty. Podobně jako karty rodiny COMBOv2 je k počítači připojena přes konektor PCI Express x8. Hlavní FPGA čip této karty je Virtex-5 XC5VTX240TFFG1759-2 (detaily viz [29]), který obsahuje více hardwarových zdrojů nežli čipy použité na kartách COMBOv2. Dalším rozdílem je použití čipu CPLD XC2C256 [27] namísto Spartan-3. Tato karta dále obsahuje tyto hardwarové prvky:

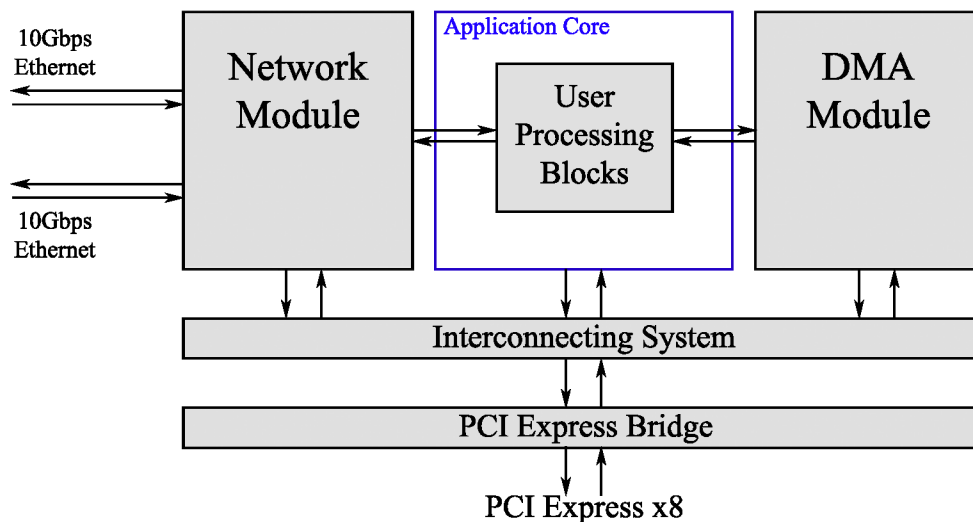
- tři paměti x36 QDR II CY7C1515JV18,
- čtyři paměti x32 RLDRAM II MT49H16M36HT-25,
- Mictor debugging konektor,
- dvě paměti Platform XL Flash (každá 128 MB),
- konektor DB9 (RS232).

Ke kartám NetFPGA v současné době neexistuje dostatečný firmware, který by tvorbu hardwarových aplikací usnadňoval. Cílem této práce je odstranit tento nedostatek zprovozněním platformy NetCOPE, která je pro vývoj HW akcelerovaných aplikací určená (viz dále kapitola 4), nad touto kartou.

Kapitola 4

Platforma NetCOPE

NetCOPE [14] je platforma určená pro rychlý vývoj akcelerovaných síťových aplikací na kartách rodiny COMBOv2, představených v kapitole 3.1. Tato platforma poskytuje několik funkčních bloků – modulů, s jejichž vývojem se návrhář síťové aplikace již nemusí zabýrat. Ke každému modulu je definováno vhodné rozhraní a komunikační protokol umožňující snadné zapojení a použití ve vytvořené aplikaci. Modulární architektura této platformy tak umožňuje vývojáři nezabývat se řešením nízkoúrovňových problémů, jako je například práce se síťovým rozhraním nebo přenos dat do paměti počítače. Hardwarová architektura platformy je naznačena na obrázku 4.1. Ačkoli je díky modulům práce při vývoji značně ulehčena, je přesto pro vývojáře pracujícího s touto platformou nezbytně nutné se seznámit s jednotlivými částmi platformy, rozhraním jednotlivých modulů a komunikačními protokoly, aby mohly být ve výsledné aplikaci správně využity.



Obrázek 4.1: Hardwarová architektura platformy NetCOPE

Platforma NetCOPE však nenabízí pouze hardwarové části, ale navíc poskytuje také softwarový základ – ovladače, které jsou nezbytné pro práci s kartou rodiny COMBOv2, a základní softwarové knihovny a nástroje pro testování a ověření funkčnosti výsledné aplikace.

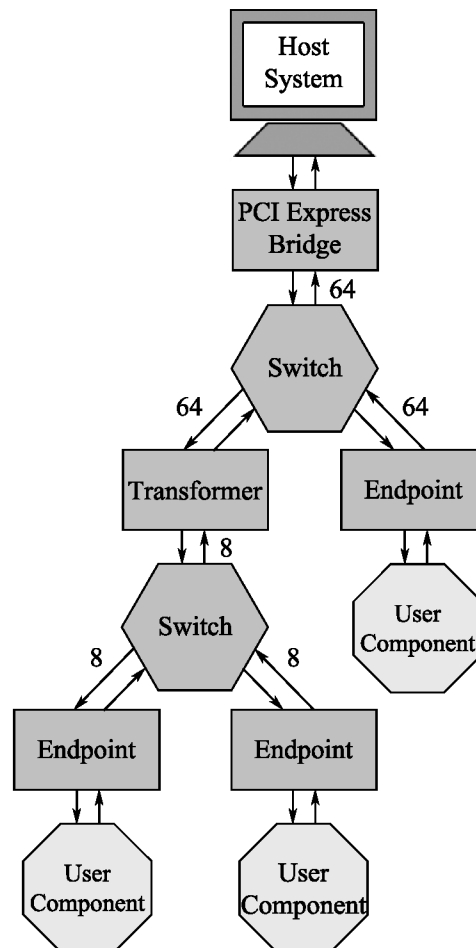
Při tvorbě této kapitoly bylo čerpáno z dokumentace projektu Liberouter [25] a [22].

4.1 Sběrnice a komunikační protokoly

Pro propojení jednotlivých komponent je v platformě NetCOPE vytvořeno několik druhů sběrnic. Ty se dají použít nejen pro připojení modulů k vytvářené aplikaci, ale jsou také využity k připojení jednotlivých částí na sběrnici PCI Express, přes kterou karta COMBO komunikuje s dalším vybavením počítače a se softwarovou částí této platformy.

4.1.1 Generická interní sběrnice

Generická interní sběrnice [16] (v platformě anglicky označovaná jako *Internal Bus* nebo *IB*) je hlavní propojovací sběrnici komponent v platformě NetCOPE, sloužící pro přenos příkazů a dat v navrhované hardwarově akcelerované aplikaci. Pro komunikaci se softwarovým vybavením počítače je tato sběrnice ukončena komponentou PCI Express Bridge, která převádí signály interní sběrnice na rozhraní PCI Express. Struktura interní sběrnice je znázorněna na obrázku 4.2.



Obrázek 4.2: Struktura generické interní sběrnice

Adresový prostor

Adresový prostor modulů připojených k interní sběrnici je adresován 32-bitovou adresou a je rozdělen tak, jak je popsáno v tabulce 4.1.

Od	Do	Velikost	Určení
0x00000000	0x000003FF	1 KiB	ID komponenta
0x00000400	0x000007FF	1 KiB	I2C/MDIO
0x00000800	0x00003FFF	14 KiB	neurčeno
0x00004000	0x00007FFF	16 KiB	Síťový modul
0x00008000	0x0000BFFF	16 KiB	Timestamp jednotka
0x0000C000	0x0000FFFF	16 KiB	DMA modul – kontrolní prostor
0x00010000	0x0001FFFF	64 KiB	InterFace Connector 1
0x00020000	0x0002FFFF	64 KiB	InterFace Connector 2
0x00030000	0x0003FFFF	64 KiB	Low Speed Connector 1
0x00040000	0x0004FFFF	64 KiB	Low Speed Connector 2
0x00050000	0x0005FFFF	64 KiB	Low Speed Connector 3
0x00060000	0x0006FFFF	64 KiB	Low Speed Connector 4
0x00070000	0x0007FFFF	64 KiB	neurčeno
0x00080000	0x01FFFFFF	31,5 MiB	Uživatelský adresový prostor (MI32)
0x02000000	0x02FFFFFF	3 MiB	DMA modul – datový a IRQ prostor
0x02300000	0x03FFFFFF	29 MiB	Uživatelský adresový prostor (IB)
0x04000000	0xFFFFFFFF	3,937 GiB	neurčeno
0xFFFFFFFF0	0xFFFFFFFF	16 B	PCI Express Bridge

Tabulka 4.1: Adresový prostor interní sběrnice

Komunikační protokol – rozhraní

Komunikační protokol interní sběrnice definuje dvě rozhraní – rozhraní sběrnice a rozhraní uživatelské komponenty, která je na sběrnici připojována. Signály rozhraní sběrnice a jejich datová šířka a směr jsou popsány v tabulce 4.2. Je potřeba podotknout, že signály definované v této tabulce se na rozhraní komponent vyskytují vždy dvakrát, ve dvou směrech (pokud chce komponenta data ze sběrnice přijímat i na ni zapisovat). Signály pak bývají označovány prefixy UP_ (z komponenty na sběrnici) a DOWN_ (ze sběrnice do komponenty).

Název	Šířka [bitů]	Směr
DATA	8–128	Zdroj → Cíl
SOF_N	1	Zdroj → Cíl
EOF_N	1	Zdroj → Cíl
SRC_RDY_N	1	Zdroj → Cíl
DST_RDY_N	1	Cíl → Zdroj

Tabulka 4.2: Signály rozhraní interní sběrnice

Signál DATA slouží pro samotný přenos datových struktur. Signály SOF_N a EOF_N identifikují začátek a konec přenášeného datového rámce. Signál SRC_RDY_N identifikuje připravenost zdroje odesílat data a určuje platnost signálů DATA, SOF_N a EOF_N. Signál DST_RDY_N

označuje připravenost cíle data přijímat. Oba dva posledně zmíněné signály využívají negativní logiky¹, v následujícím textu budou všechny signály označené příponou *_N* mít tuto vlastnost.

Součástí komunikačního protokolu je definováno také rozhraní uživatelské komponenty. Signály tohoto rozhraní jsou popsány v tabulce 4.3. Toto rozhraní využívá pouze jediná komponenta *Endpoint* (v tabulce označená jako EP), která bude zmíněna v následujícím textu v podsekcí 4.1.1.

Název	Šířka [bitů]	Směr
WR_REQ	1	EP → Komponenta
WR_RDY	1	Komponenta → EP
WR_DATA	8–128	EP → Komponenta
WR_ADDR	1–32	EP → Komponenta
WR_BE	1–16	EP → Komponenta
WR_LENGTH	12	EP → Komponenta
WR_SOF	1	EP → Komponenta
WR_EOF	1	EP → Komponenta
RD_REQ	1	EP → Komponenta
RD_RDY_ACCEPT	1	Komponenta → EP
RD_ADDR	1–32	EP → Komponenta
RD_BE	1–16	EP → Komponenta
RD_LENGTH	12	EP → Komponenta
RD_SOF	1	EP → Komponenta
RD_EOF	1	EP → Komponenta
RD_DATA	8–128	Komponenta → EP
RD_SRC_RDY	1	Komponenta → EP
RD_DST_RDY	1	EP → Komponenta

Tabulka 4.3: Rozhraní uživatelské komponenty pro připojení k IB

Signály s prefixem **WR_** jsou signály zápisové části rozhraní, prefix **RD_** značí čtecí rozhraní.

WR_REQ identifikuje požadavek na zápis a určuje platnost ostatních zápisových signálů.

WR_RDY signál používá uživatelská jednotka pro oznámení své připravenosti přijmout data.

WR_DATA je signálem nesoucím data určená k zápisu.

WR_ADDR je signálem nesoucím adresu místa v adresovém prostoru, kam mají být data zapsána.

WR_BE (*Byte Enable*) je povolovací signál zápisu. Jednička v určitém bitu tohoto signálu identifikuje požadavek na zápis konkrétního bajtu.

WR_LENGTH identifikuje celkovou délku transakce. Tento signál je vystaven po celou transakce a nemění se.

¹Negativní logika se vyznačuje tím, že je signál aktivní v *log. 0* a neaktivní v *log. 1*

WR_SOF (*Start of Frame*) označuje první slovo započaté transakce.

WR_EOF (*End of Frame*) označuje poslední slovo probíhající transakce.

RD_REQ určuje požadavek na čtení a platnost ostatních čtecích signálů.

RD_ARDY_ACCEPT identifikuje připravenost uživatelské jednotky přijmout požadavek.

RD_ADDR je signálem nesoucím adresu místa v paměti, ze kterého se mají data číst.

RD_BE (*Byte Enable*) je povolovacím čtecím signálem. Obdobně jako **WR_BE** jednotlivými bity označuje, který bajt má být z paměti vyčten.

RD_LENGTH označuje celkovou délku čtecí transakce.

RD_SOF (*Start of Frame*) označuje první požadavek transakce.

RD_EOF (*End of Frame*) označuje poslední požadavek transakce.

RD_DATA nese přečtená data.

RD_SRC_RDY (*Source Ready*) označuje připravenost uživatelské jednotky odeslat čtená data (označuje platnost **RD_DATA** signálu).

RD_DST_RDY (*Destination Ready*) je signálem určujícím připravenost *Endpoint* jednotky interní sběrnice.

Ze strany uživatelské komponenty je také definováno BusMaster rozhraní, tedy rozhraní, které umožňuje uživatelské komponentě řízení přenosů na sběrnici nejen v rámci čipu, ale v rámci celého počítače (tzv. *DMA přenos*). Šířky a směr signálů tohoto rozhraní jsou v tabulce 4.4 a jejich funkce je popsána v následujícím textu.

Název	Šířka [bitů]	Směr
BM_DATA	8–128	Komponenta → EP
BM_SOF_N	1	Komponenta → EP
BM_EOF_N	1	Komponenta → EP
BM_SRC_RDY_N	1	Komponenta → EP
BM_DST_RDY_N	1	EP → Komponenta
BM_TAG	8	EP → Komponenta
BM_TAG_VLD	1	EP → Komponenta

Tabulka 4.4: Signály BusMaster rozhraní

BM_DATA nese data vygenerované hlavičky zahájené transakce.

BM_SOF_N označuje první slovo dat.

BM_EOF_N označuje poslední slovo dat.

BM_SRC_RDY_N označuje připravenost uživatelské komponenty odesílat data a určuje platnost signálů **BM_DATA**, **BM_SOF_N** a **BM_EOF_N**.

BM_DST_RDY_N označuje připravenost *Endpoint* jednotky přijmout data.

BM_TAG nese značku (*tag*) transakce, která byla právě dokončena.

BM_TAG_VLD označuje platnost signálu **BM_TAG**.

Komunikační protokol – typy transakcí

Data na interní sběrnici jsou posílána ve formě datových jednotek označovaných jako *pakety* nebo *rámce*. Tato datová jednotka se skládá ze 128-bitové hlavičky, v závislosti na typu transakce pak mohou hlavičku následovat data o velikosti 1 až 4096 bajtů, přičemž maximální délka rámce je odvozena od specifikace PCI Express sběrnice a maximální velikosti stránky v paměti RAM. Hlavičky transakcí jsou obecně třech typů – *zápisové*, *čtecí* a *dokončovací*. Paket zápisové a dokončovací transakce obsahuje hlavičku i zapisovaná data, čtecí transakce pak obsahuje pouze hlavičku a označuje požadavek na čtení. Dokončovací transakce je pak odpovědí na tento požadavek.

Transakce se obecně rozdělují také na *lokální* a *globální*. Lokální transakce jsou transakcemi probíhajícími mezi jednotlivými komponentami na čipu v rámci navrhované hardwarově akcelerované aplikace. Globální transakce pak slouží k přenosu dat mezi čipem a dalším hardwarovým vybavením počítače (například operační paměť).

Typy transakcí a hlaviček paketů jsou shrnuty v tabulce 4.5.

Zkratka	Typ
L2LW	Lokální zápisová transakce
L2LR	Lokální čtecí transakce
L2GW	Globální zápisová transakce
G2LR	Globální čtecí transakce
RDC	Dokončovací transakce
RDCL	Poslední část dokončovací transakce

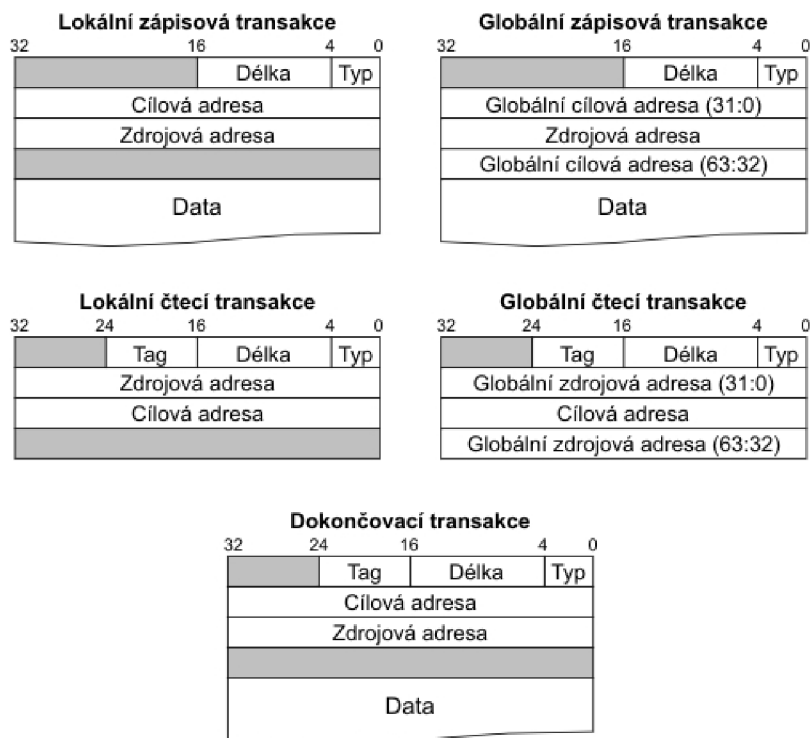
Tabulka 4.5: Typy transakcí a paketů přenášených na interní sběrnici

Poslední dva typy transakcí, tedy *RDC* a *RDCL* nesou odpověď na čtecí požadavek. Rozdíl mezi těmito dvěma transakcemi je pouze ten, že *RDCL* označuje poslední blok přenášených dat, *RDC* neobsahuje všechna data a nutně za ním musí následovat další paket.

Formát hlaviček paketů je znázorněn na obrázku 4.3 (pro datovou šířku 32 bitů). Jak již bylo zmíněno, hlavička má vždy délku 128 bitů.

Napojení na PCI Express

Pro napojení interní sběrnice ke sběrnici počítače PCI Express slouží komponenta IB Root, jinak také označovaná jako PCI Express Bridge. Tato komponenta je počátečním bodem interní sběrnice. Datová šířka interní sběrnice byla v tomto místě určena na 64 bitů. Hlavní úlohou této jednotky je obousměrný překlád transakcí mezi interní sběrnici a nadřazeným systémem PCI Express. Díky tomu, že záleží na konkrétní cílové systémové sběrnici, je část této komponenty, především dekodér a generátor datových jednotek systémové sběrnice, silně závislá na platformě cílového počítače. Pro usnadnění tvorby této komponenty je využito samostatného výstavbového bloku (tzv. *LogiCORE™ IP core*), který poskytuje výrobce cílového čipu Xilinx. Návrhem a tvorbou této jednotky se zabývá práce [13].



Obrázek 4.3: Formát hlaviček paketů přenášených po interní sběrnici [6]

V registrech této jednotky jsou také uloženy informace, které umožňují identifikaci karty na sběrnici hostitelského počítače a její „spárování“ s příslušným ovladačem. Mezi tyto informace patří především *Vendor ID* a *Subsystem Vendor ID*, která identifikují výrobce karty a jejího podsystému (tato ID jsou přidělována organizací PCI-SIG [5]), *Device ID*, které označuje jednotlivá PCI Express zařízení připojená ke sběrnici, *Subsystem Device ID*, které umožňuje rozlišovat různé aplikace na kartě (*NIC – Network Interface Card* [23] nebo např. již zmíněný NIFIC) a na jehož základě se vybírají příslušné ovladače.

Jednotka PCI Express Bridge je přístupná z interní sběrnice na adrese $0x\text{FFFFFFF0}$.

Komponenty

Aby bylo možno interní sběrnici rozvést po celém čipu a napojit na ni pro platformu nezbytné moduly, jsou navrženy další komponenty, které toto umožňují. Nejdůležitější z nich jsou IB Switch a IB Endpoint.

Switch je komponenta umožňující přepínání datových komunikačních jednotek na sběrnici podle jejich cílové adresy.

Endpoint je koncová komponenta sloužící pro napojení cílové jednotky na interní sběrnici. Pro napojení signálů komponenty slouží univerzální čtecí (výstupní) a zápisové (vstupní) rozhraní. Komponenta, která vyžaduje možnost řízení přenosů na sběrnici může využít také k tomuto určené BusMaster rozhraní.

Transformer je transformační komponentou pro převod transakcí mezi dvěma částmi datové sběrnice s různými šířkami datových spojů.

Kromě těchto základních komponent jsou součástí platformy NetCOPE ještě další komponenty sloužící pro lepší práci s touto sběrnicí a zajišťující její plnou a očekávanou funkčnost. Mezi tyto komponenty patří například jednotky zlepšující časování (Pipe), převod mezi rozhraními IB a MI (MI32 rozhraní je zmíněno v následující sekci 4.1.2) a jednotka sloužící pro značkování transakcí pro jejich seřazení (Tag Sequencer).

4.1.2 MI32

MI32 je sběrnicí určenou pro napojení uživatelem platformy navržených aplikačních modulů a konfiguračních registrů modulů platformy. Je napojena na interní sběrnicí, přičemž pro adresování komponent připojených na MI32 je vyhrazen určitý adresový prostor (jž popsán v tabulce 4.1). Standardní šířka datového a adresového nosiče je 32 bitů (od toho název MI32), teoreticky je však možné definovat datové i adresové rozhraní o šířce 1–32 bitů.

Komunikační protokol

Pro připojení komponent na sběrnicí MI32 je definováno rozhraní, jehož signály jsou popsány v tabulce 4.6.

Název	Šířka [bitů]	Směr
DWR	1–32	MI32 → Komponenta
ADDR	1–32	MI32 → Komponenta
BE	1–4	MI32 → Komponenta
RD	1	MI32 → Komponenta
WR	1	MI32 → Komponenta
ARDY	1	Komponenta → MI32
DRD	1–32	Komponenta → MI32
DRDY	1	Komponenta → MI32

Tabulka 4.6: Signály rozhraní MI32

DWR je signálem nesoucí zapisová data.

ADDR je adresový signál určující adresu zápisu či čtení.

BE (*Byte Enable*) je signál povolující zápis, resp. čtení konkrétního bajtu z přenášeného slova.

RD je povolující signál signalizující požadavek na čtení.

WR signalizuje požadavek na zápis.

ARDY identifikuje přijetí požadavku na zápis nebo čtení.

DRD je signál nesoucí čtená data.

DRDY určuje platnost signálu DRD.

Z popisu signálů tohoto rozhraní je patrné, že po této sběrnicí se nepřenášejí pakety obdobné těm na interní sběrnicí, což umožňuje jednodušší připojení komponent na MI32. Tato sběrnicí však umožňuje pouze adresování komponent v rámci čipu a není přes ni možné adresovat další vybavení počítače. MI32 je ale na interní sběrnicí napojena, což umožňuje za pomoci dekodérů vyčítat registry připojené na MI32 do softwaru počítače.

Komponenty

Pro práci s MI32 sběrnici je navrženo několik komponent usnadňujících její napojení na komponenty a rozvod po navržené aplikaci.

EPI to MI Converter je důležitou jednotkou sloužící pro napojení MI32 na interní sběrnici.

Pipe je komponenta sloužící pro zlepšení časování sběrnice čipu.

Splitter je tím, čím je pro interní sběrnici *Switch*. Provádí rozdělení řídicích signálů na více kanálů na základě určených bitů adresy. Důležité je, že narozdíl od *Switch* jednotky interní sběrnice je tato komponenta schopna rozdělovat i do více než dvou výstupů.

MI8 to MI32 Transformer je jednotkou transformující osmibitovou variantu MI32 sběrnice na její standardní 32-bitovou podobu.

4.1.3 FrameLink a MultiLink

FrameLink je dedikovanou sběrnici sloužící pro propojení rozhraní síťových vstupních a výstupních bufferů, popsaných více v sekci 4.2.1, a DMA modulu, popsaného v 4.2.2. Je často využíván komponentami určenými pro klasifikaci přenášených dat, je však nutné podotknout, že tyto komponenty ve své podstatě tuto sběrnici neovládají, pouze data sledují a provádějí nad nimi příslušné operace.

Komunikační protokol

Rozhraní tohoto protokolu je definováno v tabulce 4.7. Jeho základ je v protokolu LocalLink [26] navrženém společností Xilinx. Rozhraní je založeno na přenášení dat ve formách paketů definovaných platformou NetCOPE. Typicky se paket může skládat ze tří částí – hlavičky, dat a patičky – přičemž libovolná část může být vynechána, neboť sběrnice samotná nekontroluje, jaká data jsou po ní přenášena. Toto zajišťuje komponenta sama na základě řídicích signálů tohoto dedikovaného spoje.

Název	Šířka [bitů]	Směr
DATA	8–64	Zdroj → Cíl
REM	1–3	Zdroj <i>rightarrow</i> Cíl
SOF_N	1	Zdroj <i>rightarrow</i> Cíl
EOF_N	1	Zdroj <i>rightarrow</i> Cíl
SOP_N	1	Zdroj <i>rightarrow</i> Cíl
EOP_N	1	Zdroj <i>rightarrow</i> Cíl
SRC_RDY_N	1	Zdroj <i>rightarrow</i> Cíl
DST_RDY_N	1	Cíl <i>rightarrow</i> Zdroj

Tabulka 4.7: Rozhraní FrameLink

DATA je signál přenášející data.

REM (někdy označovaný jako **DREM** – *Data Reminder*) označuje počet platných bajtů v posledním přenášeném slově (platnost tohoto signálu určuje signál **EOP_N**).

SOF_N (*Start of Frame*) ohraničuje začátek přenášeného paketu.

EOF_N (*End of Frame*) označuje poslední slovo přenášeného paketu.

SOP_N (*Start of Part*) označuje první slovo přenášené části datového paketu.

EOP_N (*End of Part*) označuje poslední slovo přenášené části datového paketu.

SRC_RDY_N (*Source Ready*) identifikuje připravenost zdroje vystavit data na sběrnici.

DST_RDY_N (*Destination Ready*) identifikuje připravenost cíle přijímat data.

Platí, že přenos dat probíhá pouze, když jsou aktivní oba signály určující připravenost zdroje a cíle (**SRC_RDY_N** a **DST_RDY_N**). Se začátkem paketu **SOF_N** musí být aktivní také signál signalizující začátek první části paketu **SOP_N**. Obdobně je tomu se signály **EOF_N** a **EOP_N**.

Pořadí bajtů přenášených na FrameLinku je definováno v tabulce 4.8.

Bajt dle VHDL	$(n - 1 \text{ downto } n - 8)$...	$(15 \text{ downto } 8)$	$(7 \text{ downto } 0)$
Pořadí	$(n/8 - 1)$	1.	0.

Tabulka 4.8: Pořadí bajtů na FrameLinku (bajt definovaný v jazyce VHDL)

Rozdílem oproti LocalLink rozhraní je zarovnání dat v paketu. Ta jsou na FrameLinku vždy zarovnána (v jednom přenášeném slově nemohou být dvě části paketu). Explicitně není uvedeno, jaká je aktuálně přenášená část paketu, to je dáno jejím pořadím. Každá komponenta tak musí obsahovat příslušný dekodér. Dalším rozdílem oproti LocalLinku je přítomnost signálů **SOP_N** a **EOP_N**, které ohraničují začátky a konce jednotlivých částí paketu. Také se na FrameLinku používá rozdílné uspořádání přenášených bajtů, především kvůli snazšímu ukládání dat do paměti BlockRAM na FPGA čipu a jejich přenosu do softwarového vybavení počítače.

MultiLink

MultiLink je dedikovaný spoj odvozený od FrameLinku. Byl navržen pro potřeby generických DMA modulů, které umožňují distribuci dat na několik jader procesorů. Díky tomu, že nahrazuje několikanásobné propojení FrameLinku jen jediným datovým a konfiguračním tokem FrameLinkových signálů s určením adresy toku, ke kterému platná data náleží, je možné na čipu ušetřit zdroje nevyužitím několikanásobného propojení FrameLinkem. Při použití MultiLinku namísto FrameLinku je však potřeba se vypořádat s problémem hladovění, což částečně řeší příslušné komponenty platformy NetCOPE.

MultiLink má shodné rozhraní s FrameLinkem, které je částečně upraveno. Úpravy shrnuje tabulka 4.9. N označuje počet toků.

Název	Šířka [bitů]	Směr	Úprava
DST_RDY_N	N	Cíl \rightarrow Zdroj	změna šířky signálu z 1 na N
CHANNEL	$\log(N)$	Zdroj \rightarrow Cíl	nový signál, určuje adresu nebo hash aktuálních dat

Tabulka 4.9: Nové a upravené signály MultiLink oproti FrameLink

Komponenty

Protože FrameLink rozhraní je propojem mezi síťovým rozhraním a DMA modulem, který zajišťuje přenos dat mezi hardwarem a softwarem, je FrameLink právě tím místem, kde budou vyvíjeny uživatelské hardwarově akcelerované aplikace. Proto pro práci na tomto propoji je navržena poměrně rozsáhlá řada komponent.

Komponenty umístitelné na FrameLink, resp. MultiLink lze rozdělit do několika kategorií:

- **zpracování datových toků,**
- **změna dat v přenášeném rámci,**
- **ukládání dat rámce,**
- **ladění,**
- **ostatní.**

Komponenty pro zpracování toků obecně poskytují uživateli možnost data tekoucí v jednom datovém toku replikovat do více toků, přepínat mezi toky, provádět distribuci nebo rozdělení rámců. Také se sem zařazují komponenty sloužící pro převod FrameLinku na MultiLink, či FrameLinku o jisté datové šířce na jinou.

Binder spojuje několik datových toků do jednoho.

Fork replikuje vstupní tok do několika výstupních.

Pipe slouží pro zlepšení časování navrhované aplikace, svou instancí rozděluje kritickou cestu na FrameLinku na dvě části.

Sequencer umožňuje seřazení datových rámců z více toků do jednoho tak, aby bylo zachováno jejich pořadí při přijetí.

Splitter je komponentou, která rozdělí rámec přenášený v jednom toku do více toků (rozdělí ho na více částí).

Switch na základě informace v rámci rozděluje data z jednoho toku do více toků.

Distributor je jednotkou umožňující na základě informací v přenášeném rámci nasměrovat tento rámec do jednoho z výstupních toků.

Transformer slouží k propojení různých šířek FrameLink rozhraní.

Packet Binder spojuje dva datové rámce do jednoho.

Discard je jednotkou připojenou na upravené MultiLink rozhraní. Komponenta zajišťuje zahození přenášených dat, pokud je cílové úložiště zaplněno. Zároveň se jedná o jednotku umožňující počítání statistik zahozených a propuštěných dat.

Multiplexer je komponentou sloužící pro spojení více toků dat do jednoho multiplexovaného. Multiplexování probíhá na úrovni slov. Na vstupu má tato komponenta několik FrameLinkových rozhraní, výstupním rozhraním je pak MultiLink.

Komponenty pro změny dat v datech rámce běžně zahrnují uskutečnění takových operací, které pracují s první částí paketu – hlavičkou. Tu tyto komponenty buď vytvářejí či do ní doplňují data, nebo tato data modifikují či vyčítají a předávají jiným komponentám. Jsou tak základním prostředkem pro vytváření transakcí na FrameLinku a prostředkem použitelným pro klasifikaci přenášených dat.

Completer je jednotkou, která poskládá vstupní data do výsledného rámce na základě jejich správného umístění v rámci celku.

Cutter umožňuje vyčtení konkrétních dat z jednotlivých přenášených rámců a to konkrétní počet bajtů od určitého offsetu v rámci rámce. Umožňuje také tato data z rámce odebrat.

Extract funguje obdobně jako Cutter, neumožňuje však odebírat data.

Simple Marker řeší vložení či nahrazení dat o dané šířce ve FrameLinkovém toku. Vkládaná data jsou komponentě předávána za běhu.

Masker je komponentou sloužící k maskování jednotlivých bitů ve FrameLinkovém rámci.

Stamper vkládá identifikační známku o šířce jednoho slova (dle šířky DATA signálu) na začátek přenášeného rámce.

Trimmer je jednotkou sloužící k odstranění určité části (hlavičky nebo koncové části) FrameLink rámce.

Shortener slouží pro zkrácení části FrameLink toku a ponechá v ní pouze požadovaný počet bajtů.

First insert je komponentou vkládající část na začátek přenášeného FrameLink rámce o maximální délce jednoho slova.

Pro odložení zpracování dat, jejich přenos do softwaru, umožnění uložení přenášeného rámce tak, aby nebyl zahozen v důsledku zahlcení pracovní linky, zlepšení časování nebo pro uspokojení dalších podobných požadavků (například potřeby dočasného uložení dat pro převod mezi datovými šířkami sběrnice) slouží komponenty pro ukládání dat. Tyto komponenty jsou zpravidla založené na principu *FIFO*.

FIFO je standardní jednotkou FIFO se dvěma FrameLink rozhraními (RX a TX).

PFIFO (*Packet FIFO*) umožňuje rozdíl od FIFO jednotky zahazování čtených rámců.

PRFIFO (*Packet Releasing FIFO*) se chová jako PFIFO s tím rozdílem, že pokud ukládaná data přesáhnou limit bufferu, data rozukládaného rámce jsou z paměti uvolněna a zbylá příchozí data rámce jsou zahozena.

ASFIFO (*Asynchronní FIFO*) je komponentou sloužící pro převod FrameLinku mezi částmi designu, které běží na různých frekvencích.

Pro vývoj uživatelské aplikace vystává také často požadavek na komponenty umožňující sledování transakcí pro účely ladění.

RX Buffer umožňuje uložení dat tekoucích na FrameLinku a jejich odeslání na MI32 sběrnici.

Bus functional model je simulační jednotkou imitující a vytvářející FrameLinkové toky ze specifikovaného souboru.

Monitor je komponentou určenou výhradně pro použití v simulacích vytvářených aplikací. Shromažďuje informace o přenýšených paketech do specifikovaného souboru.

Watch je komponentou počítající propuštěné rámce a hlídá integritu přenášených dat na FrameLinku.

ICMP detacher zajišťuje kontrolu rámce na výskyt ICMP paketu a pouze tyto pakety propouští dále.

FL Stat je jednotkou sloužící k zjištění, které jednotky na FrameLinku způsobují zpoždění zpracování dat. To činí počítáním statistik připravenost a nepřipravenosti jednotek přijímat či odesílat data (SRC_RDY a DST_RDY signály).

Pro práci s FrameLink rozhraním existuje ještě několik dalších komponent, které není možno přesně zařadit do výše zmíněných kategorií. Těmito komponentami jsou Agregator a Decoder.

Agregator spojuje více příchozích rámců do jednoho výstupního. Jeho délka je dána timeoutem a maximální délkou rámce.

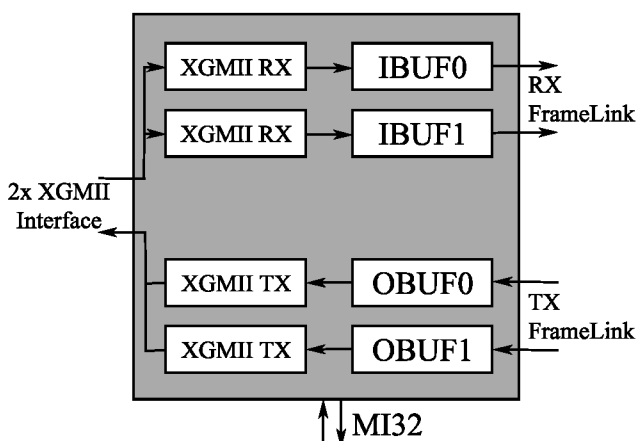
Decoder transformuje negativní logiku FrameLinku na pozitivní a přidává několik užitečných signálů identifikujících konkrétní části rámce (hlavičku, data a konec rámce).

4.2 HW moduly platformy NetCOPE a jejich rozhraní

Kromě systému sběrnic a nástrojů pro zpracování dat na nich bylo navrženo a implementováno několik základních modulů, které by se měly vyskytovat v každé aplikaci postavené nad platformou NetCOPE. Mezi ně patří nezbytný síťový modul sloužící jako rozhraní mezi aplikačním jádrem a fyzickou vrstvou počítačové sítě, DMA modul, který je spojnicí mezi aplikací a softwarovým vybavením počítače, a několik dalších modulů či jednotek.

4.2.1 Síťový modul

Síťovým modulem se rozumí ta část vytvářeného designu, která zajišťuje komunikaci vytvářené aplikace se síťovým rozhraním. Společnost Xilinx poskytuje jako součást FPGA čipů samostatné výstavbové bloky EMAC, GMII a XGMII², které zajišťují komunikaci na linkové vrstvě. Součástí síťových modulů jsou kromě těchto další potřebné bufferovací jednotky umožňující plynulé zpracování dat a zvýšenou spolehlivost přenosu. Další funkcí těchto jednotek je pak převod rozhraní EMAC/GMII/XGMII na jiné, které se užívá v rámci platformy NetCOPE. Tím je nejčastěji již zmíněný FrameLink (4.1.3).



Obrázek 4.4: Znárodnění struktury síťového modulu (pro 10 Gbps síťová rozhraní)

Ač je síťový modul jedním ze základních modulů platformy NetCOPE, jeho podoba se v rámci různých projektů může částečně lišit. Vždy však obsahuje EMAC/GMII/XGMII vstupně-výstupní rozhraní, rozhraní pro připojení na MI32 sběrnici, připojení na PACODAG (*Packet Control & Diagnostics*, jednotka sloužící pro generování kontrolních dat pro příchozí Ethernetové rámce), vzorkovací jednotku a v závislosti na počtu a typu kanálů také příslušný počet vstupních, resp. výstupních rozhraní FrameLink. Implementace síťového modulu se odvíjí od jeho napojení na uživatelskou aplikaci, některé mohou vyžadovat pouze odesílání síťových dat (např. generátor datového provozu) nebo pouze příjem (např. síťová sonda) nebo obojí (typickou ukázkou implementace takové aplikace je základní síťová karta).

²EMAC, GMII a XGMII jsou protokoly definující příslušné vlastnosti linkové vrstvy. Definují také příslušné fyzické síťové rozhraní.

Tvorbou síťových modulů pro platformu NetCOPE se zabývá práce [15].

EMAC

EMAC, neboli *Ethernet Media Access Controller*, je jedním z nezbytně nutných prostředků pro zprovoznění síťových aplikací. Tento do některých čipů již vestavěný modul umožňuje zpracování síťové komunikace na linkové vrstvě dle standardu *IEEE 802.3*. Je schopen zpracovávat síťovou komunikaci o rychlostech 10/100/1000 Mbps.

GMII a XGMII

GMII (*Gigabit Media Independent Interface*) a XGMII (*10 Gigabit Media Independent Interface*) jsou standardy definované podle *IEEE 802.3* umožňující síťovou komunikaci v rychlostech 1 Gbps a 10 Gbps. Podobně jako EMAC, bývají i tyto moduly poskytovány v rámci čipu výrobcem.

IBUF

IBUF (*Input Buffer*) je komponenta obsahující výstupní buffer EMAC/GMII/XGMII modulu. Data, která jsou zpracována v těchto modulech, jsou po příslušném rozhraní přenesené do tohoto bufferu, což umožní nejen zabránit zahazování dat v případě zahlcení linky, která tato data dále zpracovává, ale také umožňuje plynulost dalšího zpracování v této lince. Toho je dosaženo tak, že do dalšího zpracování jsou data odeslána až při nahrání celé datové jednotky do bufferu.

OBUF

OBUF (*Output Buffer*) je komponenta obdobná IBUFu, jedná se však o vstupní buffer EMAC/GMII/XGMII modulu. Data jsou příslušnému modulu předána k odeslání, jakmile je celá datová jednotka uložena ve vstupním bufferu.

Adresový prostor

Adresový prostor síťového modulu začíná na adrese 0x04000 a je mu vyhrazen adresový blok o velikosti 16 384 bajtů. Tabulka 4.10 udává rozložení základních adres jednotlivých vstupních a výstupních bufferů v rámci celého adresového prostoru.

Adresa	Komponenta
0x00004000	OBUF0
0x00004100	OBUF1
...	...
0x00005000	IBUF0
0x00005100	IBUF1
...	...

Tabulka 4.10: Adresový prostor síťového modulu

K jednotlivým registrům vstupních (IBUF) a výstupních (OBUF) bufferů se přistupuje pomocí konkrétních offsetů adresy. Celková adresa registru konkrétního bufferu pak závisí na základní adrese. Offsety, které odpovídají příslušným registrům jsou uvedeny v tabulce 4.11 pro vstupní buffery a v tabulce 4.12 pro výstupní buffery.

Offset	Registr
0x00	Total Received Frames Counter – Low
0x04	Correct Frames Counter – Low
0x08	Discarded Frames Counter – Low
0x0C	Counter of Frames Discarded due to Buffer Overflow – Low
0x10	Total Received Frames Counter – High
0x14	Correct Frames Counter – High
0x18	Discarded Frames Counter – High
0x1C	Counter of Frames Discarded due to Buffer Overflow – High
0x20	Enable Register
0x24	Error Mask Register
0x28	Status Register
0x2C	Command Register
0x30	Minimum Frame Length Allowed
0x34	Frame MTU
0x38	MAC Address Check Mode
0x80	Memory of Available MAC Addresses

Tabulka 4.11: Adresový prostor registrů jednotky IBUF

Offset	Registr
0x00	Total Frames Counter – Low
0x04	Transmitted Frames Counter – Low
0x08	Discarded Frames Counter – Low
0x10	Total Frames Counter – High
0x14	Transmitted Frames Counter – High
0x18	Discarded Frames Counter – High
0x20	Enable Register
0x24	MAC Address Register – Low
0x28	MAC Address Register – High
0x2C	Command Register
0x30	Status Register

Tabulka 4.12: Adresový prostor registrů jednotky OBUF

4.2.2 DMA moduly

Generický DMA modul je jedním z hlavních modulů celé platformy. Tento modul zajišťuje přenos dat mezi počítačovou sítí a pamětí RAM počítače. Je nutně připojen na interní sběrnici (4.1.1) a pomocí BusMaster rozhraní řídí přenosy dat do operační paměti. Konfigurační registry DMA řadičů, případně jiných, v modulu instancovaných komponent (například jednotka pro zahazování dat, která se nevejdou do zaplněných bufferů) jsou přístupné ne přímo z interní sběrnice, ale z lokální sběrnice MI32 (4.1.2). Kromě rozhraní těchto dvou sběrnic DMA modul musí mít FrameLinkové rozhraní pro TX směr (počet FrameLinkových rozhraní je dán počtem TX kanálů) a MultiLinkové rozhraní pro směr RX.

Nad samotným generickým DMA modulem je velice často definována příslušná obálka zajišťující odpovídající rozhraní příslušnému síťovému modulu (v případě nejjednodušší ukázky uživatelské aplikace nad platformou NetCOPE – obyčejná síťová karta), případně

vstupně-výstupnímu rozhraní definovaného uživatelského aplikačního jádra. Taková obálka může například slučovat několik FrameLinkových rozhraní do jediného MultiLinkového a opačně.

Adresa	Komponenta
0x0000C000	RX DMA řadič 0
0x0000C040	TX DMA řadič 0
0x0000C080	RX DMA řadič 1
0x0000C0C0	TX DMA řadič 1
...	...
0x00002000	další jednotky

Tabulka 4.13: Konfigurační adresový prostor DMA modulu

Adresový prostor

Adresový prostor HW bufferů DMA modulu na interní sběrnici začíná na adrese 0x02000000, přičemž pro každý směr (RX a TX) je vyhrazena velikost adresového prostoru 1 MiB. Na adrese 0x02200000 je pak adresována paměť jednotky pro správu popisovačů SW paměti (*Descriptor Manager*) a na adrese 0x02280000, resp. 0x02280008, se pak nachází RX, resp. TX stavové registry. Podrobnější rozpis se nachází v tabulce 4.14.

Adresa	Komponenta
0x02000000	RX Buffer 0
0x02002000	RX Buffer 1
0x02004000	RX Buffer 2
0x02006000	RX Buffer 3
...	...
0x02100000	TX Buffer 0
0x02102000	TX Buffer 1
0x02104000	TX Buffer 2
0x02106000	TX Buffer 3
...	...
0x02200000	Descriptor Manager
0x02280000	RX stavový registr
0x02280008	TX stavový registr

Tabulka 4.14: Datový adresový prostor DMA modulu

Kontrolní prostor DMA modulu pak začíná na adrese 0x0C000 a je mu vyhrazen blok o velikosti 16 KiB, přičemž polovina adresového prostoru (tedy do adresy 0x0DFFF) je vyhrazena přímo DMA řadičům, druhá polovina pak případným uživatelským komponentám (například se zde může nacházet jednotka určená pro zahazování paketů, přičemž adresovány jsou její statistické registry). Jeho rozložení naznačuje tabulka 4.13. Z hlediska vnitřní struktury je tato část adresového prostoru DMA modulu přístupná přes MI32 sběrnici.

4.2.3 Modul uživatelské aplikace

Modul uživatelské aplikace se ve struktuře architektury nachází mezi síťovým a DMA modulem. Z tohoto je jasné, že vytvářená uživatelská aplikace bude umístěna na rozhraní FrameLink či MultiLink a s tím bude jistým způsobem pracovat. V sekcích věnujících se FrameLinkovému rozhraní (4.1.3) jsou definovány komponenty platformy NetCOPE, které jsou již pro práci na těchto místech navrženy a funkční. Zpravidla se na tomto místě provádí různé filtrování či sběr statistik síťové komunikace a získané poznatky pak mohou být přeneseny do počítače k dalšímu zpracování. Na těchto místech se lze ovšem přenosu do počítače také vyhnout, například v případě, že uživatelskou aplikací bude směrovač.

Je potřeba zmínit, že platforma NetCOPE je primárně navržena pro hardwarovou akceleraci síťových aplikací, pokud je to ale vyžadováno, softwarová část platformy (stručně popsána v 4.4) umožňuje sestavit takovou aplikaci v softwaru a tento modul (tedy hardwarové řešení) může být plně vynechán.

Adresový prostor

Pro modul uživatelské aplikace jsou vyhrazené z pohledu interní sběrnice dva poměrně rozsáhlé adresové bloky. Jeden začíná na adrese 0x00080000, má celkovou velikost 31,5 MiB a primárně je určen pro adresaci přes MI32 rozhraní. Druhý blok, určený pro adresování z interní sběrnice, pak začíná na adrese 0x02300000 a má velikost 29 MiB.

4.3 Ostatní moduly

Kromě síťového a DMA modulu je součástí platformy NetCOPE ještě několik dalších, důležitých modulů či jednotek, které se nedají nijak zvlášť zařadit. Mezi ně patří ID komponenta, hodinový modul a jednotka generující časové značky.

4.3.1 ID komponenta

ID komponenta je jednotkou sloužící k identifikaci nahraného firmware do FPGA čipu. Každý design pro COMBO kartu by tuto komponentu měl mít zapojenu, přičemž přístup k ní by měl být na adrese 0x00000000 adresového prostoru interní sběrnice. Tato komponenta obsahuje především informace o počtu RX a TX DMA kanálů, frekvenci vnitřního propojo-

Offset	Popis
0x00	Negace
0x04	SW minor
0x05	SW major
0x06	Projektové ID
0x08	HW minor
0x0A	HW major
0x0C	Registr připojený na COMMAND výstup komponenty
0x10	Hodnota STATUS vstupu komponenty
0x20	Textový název designu, řetězec ukončený nulovým bajtem
0x40	Počet RX DMA kanálů
0x41	Počet TX DMA kanálů
0x42	Frekvence propojovacího systému v MHz
0x44	Přepínání banku přístupu do SYSMON
0x48	ID minor
0x49	ID major
0x4A	NetCOPE minor
0x4B	NetCOPE major
0x4C	Čas vytvoření designu
0x50	UID – unixová identifikace uživatele, který design vytvářel
0x54	Registr přerušení
0x58	USER_GENERIC0 (libovolné použití)
0x5C	USER_GENERIC1 (libovolné použití)
0x60	USER_GENERIC2 (libovolné použití)
0x64	USER_GENERIC3 (libovolné použití)
0x68	Maximální MTU vstupních Ethernetových rámců v Bajtech
0x6C	Maximální velikost <i>SZE2</i> ³ datového bloku v Bajtech
0x70	Rezervováno
0x80	Sysmon

Tabulka 4.15: Adresový prostor ID komponenty

³ **SZE2** nebo jinak **szedata2** je definice formátu přenášených dat mezi hardwarovou částí platformy NetCOPE a odpovídajícím ovladačem operačního systému. Více v 4.4.

vacího systému, identifikaci verze NetCOPE, identifikátor projektu (či uživatelské aplikace), maximální velikost MTU pro síťová rozhraní a podobně. Tyto informace jsou obsaženy na celkové velikosti 256 bajtů adresového prostoru interní sběrnice (jednotlivé informace lze adresovat pomocí příslušného offsetu, což je rozepsáno v tabulce 4.15).

4.3.2 Hodinový modul

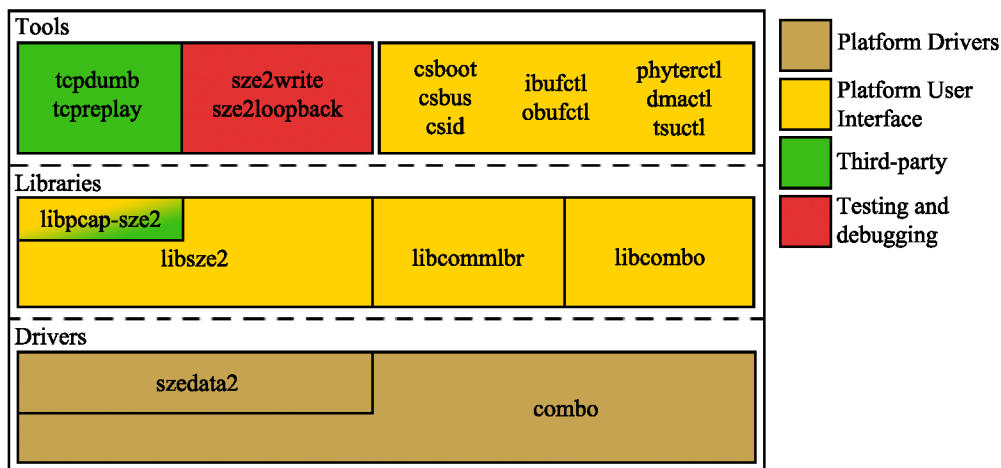
Hodinový modul je generátorem hodinových signálů o různých frekvencích, které jsou rozvedeny k jednotlivým jednotkám platformy NetCOPE na FPGA čipu. Tento modul je založen na vestavěných blocích PLL v FPGA. Kromě základních výstupních signálů modulu o frekvencích 62,5 MHz, 100 MHz, 166 MHz, 200 MHz a 250 MHz, které jsou využívány v rámci navržených jednotek platformy, hodinový modul umožňuje nastavit také dalších šest uživatelských hodinových signálů, přičemž první z nich je v rámci platformy NetCOPE využit jako hodinový signál interního propojovacího systému.

4.3.3 Timestamp jednotka

Tato jednotka je vytvořena pro generování časových značek (*timestamp*). Pro její funkčnost je potřebné její napojení na PPS signál příslušné rozšiřovací karty s modulem GPS. Formát časové značky je upřesněn až v rámci softwarového nástroje, který této jednotky využívá. Timestamp jednotka vytváří časovou značku na 64 bitech, přičemž horních 32 bitů identifikuje celou část počtu sekund, dolních 32 bitů pak vyjadřuje jejich desetinnou část. Co jsou to časové značky, k čemu se používají a jak fungují lze podrobněji nalézt například v RFC 3339 [12].

4.4 Softwarová část platformy NetCOPE

Nedílnou součástí platformy NetCOPE je také její softwarová část. Ta zastřešuje především ovladače operačního systému, balíček knihoven poskytujících uživateli základní funkce pro tvorbu síťových aplikací nad platformou a soubor základních nástrojů určených pro testování, případně sloužících pro vyčtení a nastavení řídicích a stavových registrů hardwarové části platformy. Struktura softwarové části platformy je znázorněna na obrázku 4.5, podrobnější informace se dají zjistit v [9].



Obrázek 4.5: Softwarová architektura platformy NetCOPE

4.4.1 Vrstva ovladačů

Vrstva ovladačů softwarové části platformy NetCOPE zajišťuje základní propojení mezi operačním systémem a hardwarovým a firmwarovým vybavením karty. V současné době ovladače podporují 64-bitový operační systém CentOS verze 5.

Ovladače combo jsou základní systémové ovladače poskytující řízení karet rodiny COMBO. Zajišťují základní vstupně/výstupní rozhraní pro knihovny vyšší vrstvy.

Ovladač szedata2 je systémový ovladač zajišťující rychlé DMA přenosy. Podrobnější informace o tomto ovladači jsou dostupné v [20].

4.4.2 Vrstva knihoven

Softwarové knihovny platformy poskytují vyšší míru abstrakce nad systémovými ovladači a definují rozhraní nezbytných funkcí pro tvorbu síťových aplikací nad platformou NetCOPE.

Knihovna libcommmlbr je základní knihovnou. V ní jsou definována základní makra, struktury, datové typy a funkce využívané pro ladění.

Knihovna libcombo poskytuje rozhraní pro práci s firmwarem karty, jeho bootování a vstupně/výstupní operace.

Knihovna libsze2 slouží k přijímání a odesílání dat přes *szedata2* rozhraní.

Knihovna libpcap-sze2 je knihovna *libpcap* doplněná o možnosti práce se *szedata2* rozhraním. S její pomocí lze využívat software třetích stran (jako například *tcpdump*) pracovat nad platformou NetCOPE bez nutnosti úprav.

4.4.3 Nástroje

Softwarové nástroje postavené nad knihovnami platformy NetCOPE jsou programy spustitelné v příkazové řádce, při ladění je často vhodné je spouštět v kombinaci s programem *watch*.

Nástroje csboot, csbus, csid jsou nástroje sloužící pro nahrání firmwaru do karty, pro získání informací o aktuálně nahraném firmwaru a pro zápis či čtení dat na konkrétní adresy.

Nástroje ibufctl, obufctl, phyterctl a tsuctl slouží pro monitorování a konfiguraci síťových rozhraní platformy.

Nástroje sze2write, sze2read a sze2loopback slouží pro testování správnosti a propustnosti *szedata2* a *libpcap* knihoven. Programy *sze2write* a *sze2read* umožňují odeslání a přijímání dat na síť, respektive z ní.

Nástroj dmactl slouží pro monitorování DMA řadičů.

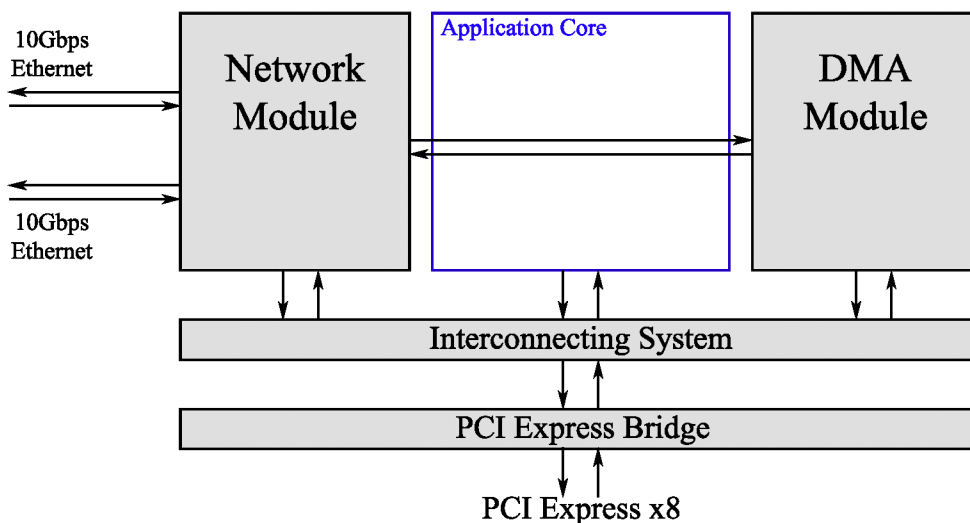
Kapitola 5

Přizpůsobení platformy NetCOPE na kartu NetFPGA

V rámci přizpůsobení platformy NetCOPE na cílovou kartu je možné se v této práci opřít od složitých, již vytvořených aplikací (například HW firewall), a práce se bude zabývat pouze nezbytně nutnými úpravami. Mezi tyto úpravy patří především přizpůsobení síťového modulu ze dvou na čtyři síťová rozhraní, odpovídajícím způsobem je potřeba upravit DMA modul. Důležitou součástí signálových rozvodů jsou také hodinové signály, jejichž zdroje se na cílové a zdrojové HW platformě mohou lišit, což je potřeba také zohlednit. Další důležitou změnou je úprava identifikace designu karty na sběrnici PCI Express a zvolení konfiguračního systému FPGA čipu. Zároveň bude v této kapitole vysvětlena příprava rozhraní designu FPGA čipu, budou popsány požadavky na syntézu a její výsledky.

5.1 Portovaná aplikace

Pro účely portace platformy NetCOPE na kartu NetFPGA bude využito té nejjednodušší hardwarově orientované aplikace nad touto platformou, tedy síťové karty [23]. Aplikace,

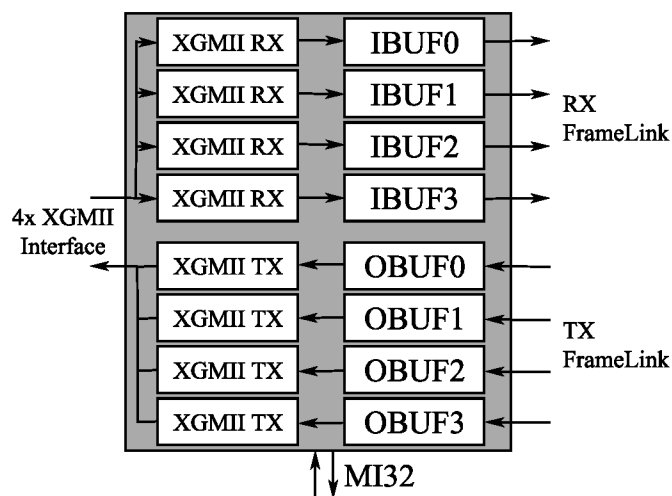


Obrázek 5.1: Síťová karta nad platformou NetCOPE

kteřá je vytvořena pro COMBOI-10G2, je však přizpůsobena na dva síťové porty. Karta NetFPGA poskytuje síťové konektory čtyři a proto bude nutné přizpůsobit síťový modul (popsán v 4.2.1), případně upravit DMA modul tak, aby data z jednotlivých síťových toků byla správně distribuována. Architektura síťové karty nad platformou NetCOPE je stručně znázorněna na obrázku 5.1.

5.1.1 Přizpůsobení síťového modulu

Síťový modul je ve své podstatě obálka nad příslušným hardwarovým blokem zajišťujícím komunikační protokol (v tomto případě XGMII) a příslušnými hardwarovými buffery. Pro karty COMBOI-10G2 jsou tyto bloky instancovány dva, pro kartu NetFPGA budou zapotřebí čtyři. S tím souvisí také přidání dalších jednotek IBUF a OBUF. Výslednou podobu síťového modulu znázorňuje obrázek 5.2.



Obrázek 5.2: Znáznornění struktury síťového modulu pro kartu NetFPGA-10G

5.1.2 Úprava DMA modulu

Obdobným způsobem jako síťový modul je potřeba upravit i modul DMA (popsaný v 4.2.2). Ten je pro kartu COMBOI-10G2 upraven pro distribuci dvou datových toků (v případě síťové karty). DMA moduly jsou v platformě NetCOPE navrženy genericky, tedy úprava spočívá v přenastavení příslušné generické konstanty a úpravou jeho obálky zajišťující korektní rozhraní pro napojení na síťový modul.

5.2 Úprava hodinového systému

Součástí platformy NetCOPE je hodinový modul (popsaný v 4.3.2) vytvářející a rozvádějící hodinové signály různých frekvencí. Výstupní signály tohoto modulu jsou odvozené z přivedeného externího zdroje (například oscilátoru nebo sběrnice PCI Express). Na kartách rodiny COMBOv2 je použit zdrojový hodinový signál o frekvenci 125 MHz. Kvůli zachování

co největší kompatibility všech částí platformy NetCOPE bude snaha o to, aby na výstupu hodinového modulu byly stejné frekvence.

5.2.1 Hodinový systém na NetFPGA

Na kartách NetFPGA-10G je několik zdrojů hodinového signálu. Prvním je nastavitelný generátor diferenciálního hodinového signálu, kterým lze ze zdrojů o frekvenci 25 MHz a 15 MHz vytvořit pomocí přepínače poměrně vysoký rozsah frekvencí. Dalšími zabudovanými zdroji jsou oscilátory generující hodinové signály o frekvencích 100 MHz a 25 MHz. Uživatel má také možnost ke třem portům, z nichž jeden se může podílet na vzniku diferenciálních hodin, připojit své vlastní externí zdroje. Distribuce hodin na NetFPGA je více popsána v [10].

Aby úpravy stávajícího modulu byly co nejmenší, bylo by ideální přivést k současnému hodinovému systému platformy NetCOPE stejný nebo podobný hodinový signál. Signál s frekvencí 125 MHz lze na NetFPGA získat pouze správným nastavením výše zmíněného generátoru diferenciálních hodin, s diferenciálním signálem však hodinový systém na vstupu nepočítá. Jako nejvhodnější se tedy jeví zvolit signál z externího oscilátoru o frekvenci 100 MHz.

5.2.2 Charakter úpravy

Použitý PLL modul pro konstrukci hodinového systému, který je součástí FPGA, má možnost použít kromě děliček frekvence také násobičku, již jsou všechny výstupní signály upraveny. To se dělá pomocí příslušného generiku. S využitím aplikace *Xilinx CORE Generator™ System*¹ bylo zjištěno, že přivedením signálu o frekvenci 100 MHz a příslušným nastavením této násobičky lze získat na výstupu shodné signály, jako s jejím implicitním nastavením a vstupním hodinovým signálem o frekvenci 125 MHz.

5.3 Úprava konfiguračního systému FPGA čipu

Průběh a vlastnosti konfigurace FPGA čipu rodiny Virtex-5 byly již popsány v sekci 2.3. Funkční řešení pro karty COMBOv2, které je součástí platformy NetCOPE, je však pro karty NetFPGA díky jejich odlišné konstrukci nepoužitelné. Konfigurační systém FPGA čipu je tedy také potřeba přizpůsobit.

5.3.1 Systém konfigurace na COMBOv2

Konfigurace čipů FPGA na kartách COMBOv2 probíhá automaticky po spuštění napájení. Nejprve proběhne automatická konfigurace čipu Spartan-3. Po jejím úspěšném dokončení přebírá tento čip řízení konfigurace hlavního FPGA Virtex-5. Základní konfigurační data pro oba dva čipy se nachází v paměti Flash Spansion S29GL01GP. Vytvořené designy je možné po načtení konfigurace čipu Spartan-3 přes sběrnici PCI Express vymazat či zapsat do daných pamětí.

Kromě paměti typu Flash je zde k dispozici ještě paměť PSRAM Micron MT45W4MW-16PC, která je využita k patentovanému systému konfigurace „za běhu“ karty. I k tomu

¹**Xilinx CORE Generator™ System** je nástroj umožňující zkrácení doby návrhu aplikací pro FPGA. Jeho součástí je katalog parametrizovatelných vestavěných bloků a komponent FPGA čipů. Více informací je dostupno v [34].

je zapotřebí již nakonfigurované pomocné FPGA. To v případě vyžádání si změny designu z počítače přes sběrnici PCI Express nejprve запиše cílová konfigurační data do této cílové PSRAM a následně spustí a řídí konfiguraci hlavního FPGA těmito konfiguračními daty. Je potřeba zmínit, že tato konfigurace je možná pouze díky rychlostem pro čtení a zápis, ve kterých zvolená PSRAM převyšuje používanou paměť Flash. Její nevýhodou ale je, že po ztrátě napájení jsou z ní data přirozeně odstraněna.

5.3.2 Zvolené řešení na NetFPGA

Jako řešení konfigurace FPGA čipu na kartách NetFPGA-10G byla prozatím zvolena konfigurace pomocí JTAG rozhraní, přesný postup je popsán v [10]. V současné době není možné zvolit lepší řešení s využitím cílového CPLD čipu, příčiny způsobené stranou výrobce, které k tomuto vedou, jsou diskutovány v kapitole 7.

5.4 Odpojení specifických částí pro COMBOv2

V rámci designů pro COMBOv2 jsou instancovány jednotky, které jsou specifické pro tyto karty. Na NetFPGA-10G jsou však tyto jednotky nepoužitelné a přebytečné. Mezi tyto jednotky patří Timestamp jednotka (využívající připojené karty COMBOL-GPS) a připojení čipu Spartan-3. Tyto části nebudou součástí platformy NetCOPE zprovozněnou pro kartu NetFPGA-10G.

5.5 Úprava identifikace karty

Identifikaci karty NetFPGA s portovanou platformou NetCOPE v počítači určují tyto tři faktory:

1. identifikační čísla určující nahraný firmware FPGA v ID komponentě, která je popsána v 4.3.1,
2. identifikační čísla určující kartu v generovaném PCI Express Core a
3. odpovídající promítnutí změn v ovladači na straně operačního systému, který identifikaci provádí.

Registr	Hodnota
SW minor	0x0c
SW major	0x05
Projektové ID	0x41c1
HW minor	0x0000
HW major	0x0007
Textový název designu	NIC_NetFPGA_10G4
Počet RX DMA kanálů	0x04
Počet TX DMA kanálů	0x04

Tabulka 5.1: Nastavení nových hodnot registrů ID komponenty

Registry ID komponenty budou pro identifikaci NetFPGA karty nastaveny způsobem, který uvádí tabulka 5.1, přičemž zvláště počet kanálů DMA patří mezi úpravy nezbytně

nutné (ovladač a software nad ním potřebuje „znát“ počet DMA kanálů). Co se týká dalších úprav především identifikace, která je součástí PCI Express Bridge a ovladačů, může být zachována její současná podoba na dobu nutnou pro ověření funkčnosti platformy NetCOPE na kartě.

5.6 Příprava a vytvoření designu

Poslední nutnou úpravou platformy NetCOPE pro její zprovoznění nad kartou NetFPGA je vytvoření tzv. *top-level* architektury a příslušných omezujících podmínek (*constraints*). Jedná se o popis propojení rozhraní navrhovaných komponent s jednotlivými piny FPGA čipu a zavedení určitých požadavků na frekvenci a maximální zpoždění určitých propojů, zvláště hodinových signálů.

5.6.1 Požadavky

Požadavkem a jedním bodem zadání této práce je mimo jiné ověřit agregovanou propustnost dat z počítačové sítě do paměti počítače. V rámci projektu Liberouter bylo experimentálně ověřeno [25], že tato propustnost se odvíjí od frekvence zpracování dat a je limitována propustností sběrnice PCI Express. Požadovaná plná agregovaná propustnost 10 Gbps bude splněna, pokud se podaří vytvořit design o frekvenci propojovacího systému alespoň 187,5 MHz. Tomuto požadavku budou přizpůsobeny omezující podmínky překladu a nastavení příslušných konstant pro hodinový modul.

5.6.2 Výsledky překladu

Výstupní design překladu všechny omezující podmínky splnil. Shrnutí potřebných zdrojů čipu XC5VTX240TFFG1759-2 pro vytvořené designy na frekvencích 187,5 MHz a 200 MHz jsou vypsané v tabulce 5.2, pro 200 MHz design podrobněji v příloze C.1 (jedná se o výsledky po fázi MAP překladu). Pro překlad byl využit překladový systém *Xilinx ISE 12.4*.

Zdroj	Počet využitých	
Slice Registers	46 573 (31 %)	46 587 (31 %)
Slice LUTs	40 504 (27 %)	40 489 (27 %)
BlockRAM/FIFOs	91 (28 %)	92 (28 %)
Dosažená frekvence	187,5 MHz	200 MHz

Tabulka 5.2: Využití zdrojů a maximální dosažená frekvence

Srovnání s výsledky na COMBOv2

Výsledné využití zdrojů po překladové fázi MAP designem pro frekvenci 200 MHz jsou pro obě karty uvedeny v příloze C. Je z nich patrné, že spotřeba zdrojů na NetFPGA-10G při stejné konfiguraci (s výjimkou síťového modulu a DMA modulu) oproti COMBOv2-10G2 narostla v některých případech na dvojnásobek. To se odvíjí od navýšení dvou síťových rozhraní na čtyři a příslušné „zvětšení“ DMA modulu, takže tyto výsledky nejsou překvapivé. Důležité je zmínit, že ačkoli procentuálně vycházejí výsledky překladu hůře pro čip FPGA na kartě NetFPGA, je na něm stále více zdrojů volných oproti čipu na kartě COMBO-LXT (v tomto případě XC5VLX155T).

Kapitola 6

Testování

V rámci fáze testování je potřeba ověřit funkčnost karty NetFPGA-10G, která je na trhu novým produktem. Mezi tyto testy patří především test karty a jejích periférií a test jejích síťových rozhraní. Balíček těchto testů dodává výrobce spolu s kartou.

Součástí testování musí být také ověření správnosti úprav platformy NetCOPE, která je nad cílovou platformou zprovoznována. K tomuto bude využito existujících simulací, případně simulačních verifikací, které byly vyvíjené spolu s platformou NetCOPE. Pro tyto simulace bude využito prostředí programu *ModelSim SE 6.6c*.

Další proces testování probíhá již na cílové kartě NetFPGA-10G, jejíž FPGA čip je nakonfigurovaný vytvořeným designem/bitstreamem. Pro tyto účely jsou součástí platformy NetCOPE rozličné testy popsané v 6.4.

6.1 Ověření funkčnosti karty NetFPGA-10G

Výrobce karet NetFPGA-10G dodává balíček základních testů umožňující ověření základní funkčnosti karty. Tyto testy jsou označeny jako *NetFPGA-10G Production Test* pro ověření periférií karty, *NetFPGA-10G 10G Ethernet Interface Loopback Test* a jeho 1 Gbps variantu pro ověření síťových rozhraní.

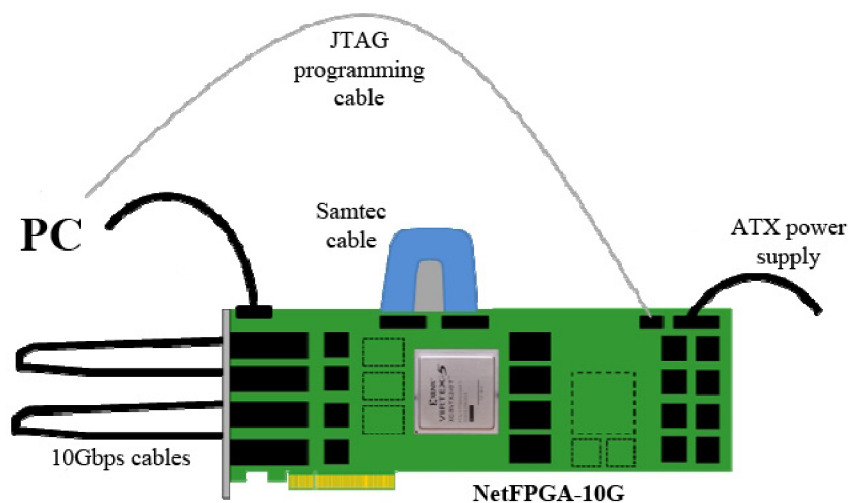
6.1.1 NetFPGA-10G Production Test

Tento základní balík testů umožňuje ověřit základní funkčnost a průceschopnost karty NetFPGA-10G, což provádí současným zatěžováním všech rozhraní karty. Kontrola může probíhat přes sběrnici PCI Express (Server mód) nebo přes UART kabel (Standalone mód). Před prvním použitím karty by tento test měl být vždy spuštěn. Schéma zapojení karty pro otestování je na obrázku 6.1, v [17] je možné se dočíst podrobnější informace k tomuto testu.

Standalone mód

Posloupnost akcí testu v tomto módu je následující:

1. Nejprve je potřeba spustit napájení karty. Zdrojem napájení je externí ATX zdroj. Po něm by měly všechny příslušné LED diody indikující napájení svítit.
2. Následně se pomocí programu Xilinx Impact nahrají přes JTAG kabel testovací konfigurace pro čipy CPLD a FPGA. Po proběhnutí této konfigurace indikují příslušné



Obrázek 6.1: Schéma zapojení karty pro Production test [17]

LED diody frekvenci hodinového signálu a funkčnost příslušných rozhraní (paměti, 10 Gbps síťová rozhraní a podobně).

3. Pomocí UART kabelu připojeného na příslušné rozhraní karty se provede připojení na terminálové rozhraní, například pomocí programu PuTTY. Toto rozhraní detailněji než stav LED diod popisuje výsledky testu. Přes toto rozhraní je také možné spustit test opakovaně.

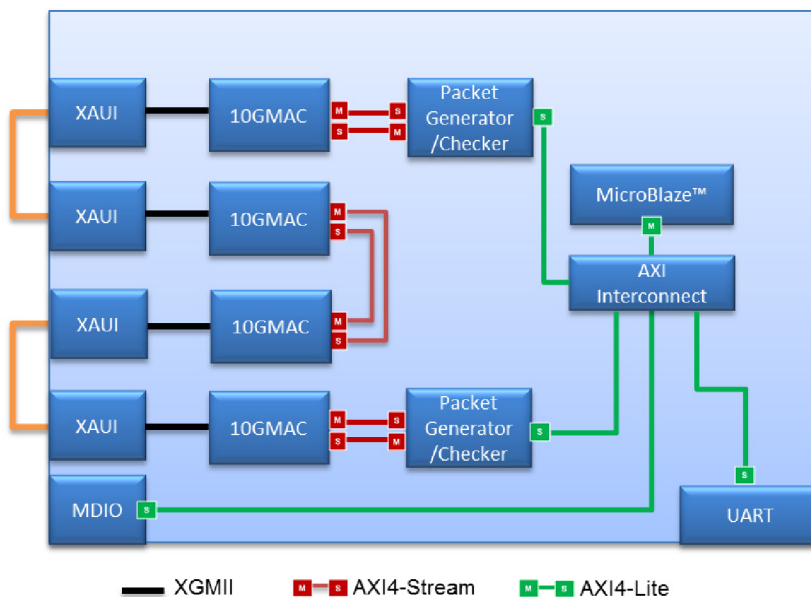
Server mód

Následující posloupnost akcí tohoto módu je sestavena pro operačním systému Linux.

1. Na konec současných nastavení bootování jádra OS (např. soubor `grub.conf`) je potřeba přidat příkaz `memmap=256$0x5f700000`. Ten slouží k alokaci místa v paměti, která bude použita pro PCI Express DMA test. Po této změně je potřeba počítač rebootovat.
2. Karta musí být připojena v počítači, příslušnými přepínači musí být nastaveno napájení karty ze sběrnice PCI Express.
3. Následuje naprogramování FPGA a CPLD obdobně jako ve Standalone módu.
4. Pomocí programu `lspci` je potřeba ověřit, zda je karta správně nainstalována a napájena.
5. V tuto chvíli by všechny diody indikující napájení měly být rozsvícené. Obdobně LED diody indikující správnost konfigurace FPGA a CPLD a funkčnost příslušných konektorů karty.
6. Příslušným skriptem je test spuštěn.

6.1.2 NetFPGA-10G 10G Ethernet Interface Loopback Test

Tento test ověřuje funkčnost síťových rozhraní a zároveň je jednou z ukázek práce s kartou NetFPGA-10G. Test spočívá ve vzájemném propojení jednotlivých síťových rozhraní kabely. FPGA je nakonfigurováno tak, aby plnilo funkci dvou generátorů datového provozu a dvou jednotek, které ověřují jejich správnost, vždy po dvou napojené na dvojici 10 Gbps síťová rozhraní. Podsystem MicroBlaze vrací statistiky generátorů a ověřovacích jednotek na UART rozhraní. Schéma zapojení karty je znázorněno na obrázku 6.2, podrobnější popis účastníků se komponent lze nalézt v [3].



Obrázek 6.2: Schéma zapojení testu 10G Ethernet Interface Loopback [3]

Posloupnost testu je následující:

1. Propojení síťového portu č.0 k portu č.1 a portu č.2 k portu č.3 přímým kabelem nebo 10GBASE-R optickým vláknem.
2. Následně se připojí UART kabelem karta k počítači a proběhne naslouchání na příslušném sériovém portu vhodným programem.
3. Do FPGA čipu je nahrána příslušná konfigurace. Na výstup terminálu budou vypsány další instrukce (inicializace, vypsání stavu, spuštění a zastavení testu) a testovací statistiky.

6.1.3 NetFPGA-10G 1G Ethernet Interface Loopback Test

Tento test je prakticky téměř shodný s testem zmíněným v 6.1.2, jediným rozdílem je přizpůsobená konfigurace FPGA 1 Gbps síťovým rozhraním.

6.2 Simulace úpravy hodinového modulu

Pro ověření chování nově upraveného hodinového modulu bude využito již existující simulace modulu připraveného pro karty COMBOv2. V souboru řídicím simulaci (*testbench*),

bude upravena frekvence vstupního hodinového signálu na požadovaných 100 MHz. Porovnáním frekvencí výstupních signálů nové simulace s výstupem simulace staré zjistíme, zda se upravený hodinový modul chová na výstupu požadovaným způsobem.

Výstupy simulace jsou znázorněny na obrázcích [A.1](#) (hodinový modul před úpravou) a [A.2](#) (modul po úpravě). Na první pohled je patrné, že vstupní hodinový signál CLKIN má u starého modulu periodu 8 ns, což odpovídá frekvenci 125 MHz. Naproti tomu je ve výstupu simulace [A.2](#) frekvence tohoto signálu 100 MHz. Ostatní signály se shodují v obou dvou simulacích a lze tedy konstatovat, že úprava hodinového modulu odpovídá požadavkům.

6.3 Simulační verifikace síťové karty

Nad základní aplikací platformy NetCOPE, tedy síťovou kartou, je vytvořena simulační verifikace pro ověření její funkčnosti a odhalení některých chyb. Celý simulační proces ověřuje správnost přístupu do registrů jednotlivých komponent a kontroluje průchod datových transakcí celým systémem. Součástí původní platformy je tato simulace připravena pro standardní dvě síťová rozhraní, což odpovídá aplikaci navržené pro kombinaci karty COMBO-LXT s COMBOI-10G2. Tuto simulaci lze jednoduchým způsobem upravit tak, aby proběhla pro čtyři síťová rozhraní. Frekvence interního propojovacího systému bude odpovídat frekvenci vytvářeného designu, tedy 187,5 MHz.

Simulace proběhla úspěšně pro necelých 20 000 transakcí ve směru příjmu a 40 000 transakcí ve směru odeslání. Textový výstup je možné nalézt v příloze [B](#). V něm je patrné, že některé transakce ověřovaným systémem neprošly a byly zahozené na vstupních bufferech, což je způsobeno limitovanou rychlostí propojovacího systému. Při vyšší frekvenci zpracování nebudou vstupní buffery tak zahlcené a transakcí projde systémem více.

6.4 Existující testy platformy NetCOPE

Součástí platformy NetCOPE jsou také různé testy, které jsou většinou automatizované, případně mohou vyžadovat například připojení síťových rozhraní karty na zařízení generující datový provoz. Testy se dají rozdělit do několika kategorií:

Funkční testy jsou testy ověřující základní funkčnost karty s nahranou platformou NetCOPE. Mezi tyto testy patří například test časových značek, správnosti příjmu a odeslání dat, správného označení poškozených paketů a podobně.

Výkonnostní testy jsou využívány pro zjištění propustnosti zařízení a zpracování dat, případně ztrátovosti.

Zátěžové testy ověřují stabilitu propojovacího systému, pozorují střední dobu mezi poruchami apod.

Crash test simuluje výpadky sítě (odpojením síťového kabelu) a jejich vliv na správnost příjmu dat.

Testy jsou zdokumentovány v rámci projektové dokumentace projektu Liberouter [\[25\]](#).

V rámci této práce bylo plánováno využití především výkonnostních testů pro ověření propustnosti. Protože však došlo k problému s doručením karty NetFPGA-10G, což je blíže referováno v kapitole [7](#), test na propustnost proveden nebyl, namísto něj byl stanoven požadavek na minimální frekvenci syntetizovaného designu, který může propustnost zajistit alespoň na teoretické úrovni.

Kapitola 7

Závěr

Cílem této práce bylo nastudovat a popsat platformu NetCOPE určenou pro vývoj hardwarově akcelerovaných aplikací na kartách rodiny COMBOv2, provést nezbytné úpravy pro její přizpůsobení na kartu NetFPGA-10G a ověřit funkčnost těchto úprav. Nejprve byla popsána technologie FPGA se zaměřením na konfiguraci čipů rodiny Virtex-5, poté zdrojové a cílové hardwarové platformy. Dále byla představena platforma NetCOPE především z hlediska jejích komunikačních rozhraní a základních modulů, které jsou její nezbytnou součástí. V další části práce je zdokumentováno přizpůsobení důležitých částí platformy NetCOPE na kartu NetFPGA-10G, simulační ověření provedených úprav a také základní testy pro ověření funkčnosti hardwarového zařízení a portované platformy.

Největším problémem, se kterým se tato práce potýkala, bylo nedodání karty NetFPGA-10G výrobcem. Karta byla objednána a zaplácena na konci ledna roku 2011 a do termínu vypracování této práce nebyla doručena ve výrobcem garantovaném termínu. Z tohoto důvodu nebylo možné úpravy otestovat v reálných podmínkách, ale pouze využitím simulační verifikace, která je nad portovanou aplikací vytvořena a která umožnila ověřit správnost úprav alespoň na teoretické rovině. Taktéž výrobce nebyl ani po rozsáhlé komunikaci ochoten dodat podrobnou dokumentaci, což bylo potřebné pro přesný popis rozhraní platformy na konkrétní FPGA čip.

Další pokračování práce se odvíjí od dodání objednané karty. Mezi první úkony by patřilo otestování základní funkčnosti jejích periférií za pomoci testů, které byly popsány v 6.1. Důležité je také ověření správného napojení platformy NetCOPE na rozhraní FPGA čipu podle kompletní dokumentace a schématu karty, zabezpečení lepší, automatické konfigurace FPGA za pomoci čipu CPLD, obdobně, jako je tomu na COMBOv2 kartách s čipem Spartan-3. Po ověření funkčnosti připojení karty na sběrnici PCI Express bude vhodné také upravit identifikaci karty nastavením správných hodnot, jak je popsáno v 4.1.1.

Výstupem práce je také vytvořený design pro hlavní modul karty NetFPGA s frekvencí interního propojovacího systému 200 MHz, který tímto na teoretické úrovni splňuje požadavek na plnou agregovanou propustnost do softwaru. Potřebných zdrojů FPGA je více, než tomu bylo na kartách rodiny COMBOv2, což odpovídá charakteru provedených úprav. Díky rozsáhlosti FPGA čipu na cílové kartě je však i přes tuto vyšší spotřebu stále více dostupných prostředků pro vytvářenou hardwarově akcelerovanou aplikaci nad platformou NetCOPE. Taktéž jsou možnosti dalšího zvyšování frekvence propojovacího systému platformy.

Literatura

- [1] *Accelerated Network Technologies at FIT homepage* [online]. 2011 [cit. 29. dubna 2011]. Dostupné na: <<http://merlin.fit.vutbr.cz/ant/>>.
- [2] *CESNET, z.s.p.o. homepage* [online]. 2011 [cit. 29. dubna 2011]. Dostupné na: <<http://www.cesnet.cz/>>.
- [3] *NetFPGA-10G 10G Ethernet Interface Loopback Test* [online]. Leden 2011 [cit. 29. dubna 2011]. Dostupné na: <<http://netfpga10g.pbworks.com/w/page/32283808/NetFPGA-10G%2010G%20Ethernet%20Interface%20Loopback%20Test>>.
- [4] *NetFPGA homepage* [online]. 2011 [cit. 29. dubna 2011]. Dostupné na: <<http://www.netfpga.org/>>.
- [5] *PCI-SIG homepage* [online]. 2011 [cit. 29. dubna 2011]. Dostupné na: <<http://www.pcisig.com/home>>.
- [6] BARTOŠ, V. *Verifikace generického propojovacího systému pro FPGA*. Brno: FIT VUT v Brně, 2009. Bakalářská práce.
- [7] CESNET, z.s.p.o. *Programmable hardware* [online]. 2002–2009 [cit. 29. dubna 2011]. Dostupné na: <<http://www.liberouter.org/>>.
- [8] CESNET, z.s.p.o. *Programovatelný hardware* [online]. 2011 [cit. 23. dubna 2011]. Dostupné na: <<http://www.cesnet.cz/projekt/02/>>.
- [9] HANK, A. *Návrh síťových aplikací na platformě NetCOPE*. Brno: FIT VUT v Brně, 2009. Diplomová práce.
- [10] HiTECH GLOBAL. *HTG-V5TXT-PCIE User Manual (preliminary)*. 2010.
- [11] IEEE COMPUTER SOCIETY. *IEEE Standard for In-System Configuration of Programmable Devices* [online]. leden 2003 [cit. 1. května 2011]. Dostupné na: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1176958>.
- [12] KLYNE, G. a NEWMAN, C. *RFC 3339 Date and Time on the Internet: Timestamps* [online]. Červenec 2002 [cit. 29. dubna 2011]. Dostupné na: <<http://www.ietf.org/rfc/rfc3339.txt>>.
- [13] KORČEK, P. *Jednotka pro řízení protokolu PCI Express*. Brno: FIT VUT v Brně, 2009. Diplomová práce.

- [14] MARTÍNEK, T. a KOŠEK, M. NetCOPE: Platform for Rapid Development of Network Applications. In *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*. Bratislava, SK: IEEE Computer Society, červen 2008. S. 219–224. Dostupné po registraci. ISBN 978-1-4244-2276-0.
- [15] MATOUŠEK, J. *Implementace a verifikace vstupních a výstupních síťových bloků*. Brno: FIT VUT v Brně, 2009. Bakalářská práce.
- [16] MÁLEK, T., MARTÍNEK, T. a KOŘENEK, J. GICS: Generic Interconnection System. In *2008 International Conference on Field Programmable Logic and Applications*. Heidelberg, DE: IEEE Computer Society, září 2008. S. 263–268. ISBN 978-1-4244-1960-9.
- [17] NETFPGA. *NetFPGA-10G Production Test*. Říjen 2010. Dostupné v rámci Git repozitáře projektu NetFPGA.
- [18] PEATTIE, M. *Using a Microprocessor to Configure Xilinx FPGAs via Slave Serial or SelectMAP Mode* [online]. Srpen 2009 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/application_notes/xapp502.pdf.
- [19] PINKER, J. a POUPA, M. *Číslicové systémy a jazyk VHDL*. 1. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-7300-198-5.
- [20] SLABÝ, J. *Rapid Data Transfers on COMBO Platform*. Brno: FI MU, 2008. Diplomová práce.
- [21] THE LIBEROUTER PROJECT TEAM. *Flexible FlowMon Probe 1.0.0 Handbook* [online]. Srpen 2009 [cit. 1. května 2011]. Dostupné na: <http://www.liberouter.org/fflowmon/handbook.html>.
- [22] THE LIBEROUTER PROJECT TEAM. *NetCOPE Platform Handbook* [online]. Prosinec 2010 [cit. 1. května 2011]. Dostupné na: <http://www.liberouter.org/netcope/handbook.html>.
- [23] THE LIBEROUTER PROJECT TEAM. *Network Interface Card Handbook* [online]. Prosinec 2010 [cit. 1. května 2011]. Dostupné na: <http://www.liberouter.org/nic/handbook.html>.
- [24] THE LIBEROUTER PROJECT TEAM. *NIFIC Handbook* [online]. Duben 2010 [cit. 1. května 2011]. Dostupné na: <http://www.liberouter.org/nific/handbook.html>.
- [25] THE LIBEROUTER PROJECT TEAM. *Projektová dokumentace*. 2010. Veřejně nedostupné.
- [26] XILINX. *LocalLink Interface Specification* [online]. Červen 2005. Dostupné po registraci. Dostupné na: http://www.xilinx.com/products/intellectual-property/LocalLink_UserInterface.htm.
- [27] XILINX. *XC2C256 CoolRunner-II CPLD* [online]. Březen 2007 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/data_sheets/ds094.pdf.

- [28] XILINX. *Platform Flash XL Configuration and Storage Device: User Guide* [online]. Prosinec 2009 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/user_guides/ug438.pdf.
- [29] XILINX. *VirteX-5 Family Overview* [online]. Únor 2009 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf.
- [30] XILINX. *VirteX-5 FPGA System Monitor User Guide* [online]. 2009 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/user_guides/ug192.pdf.
- [31] XILINX. *Spartan-3 Generation FPGA User Guide* [online]. Srpen 2010 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/user_guides/ug331.pdf.
- [32] XILINX. *VirteX-5 FPGA Configuration User Guide* [online]. Srpen 2010 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/user_guides/ug191.pdf.
- [33] XILINX. *VirteX-5 FPGA User Guide* [online]. Květen 2010 [cit. 29. dubna 2011]. Dostupné na: http://www.xilinx.com/support/documentation/user_guides/ug190.pdf.
- [34] XILINX. *Xilinx CORE Generator System* [online]. 2011 [cit. 29. dubna 2011]. Dostupné na: <http://www.xilinx.com/tools/coregen.htm>.

Seznam příloh

Příloha A Výsledky simulace úpravy hodinového modulu

Příloha B Výsledky simulační verifikace síťové karty

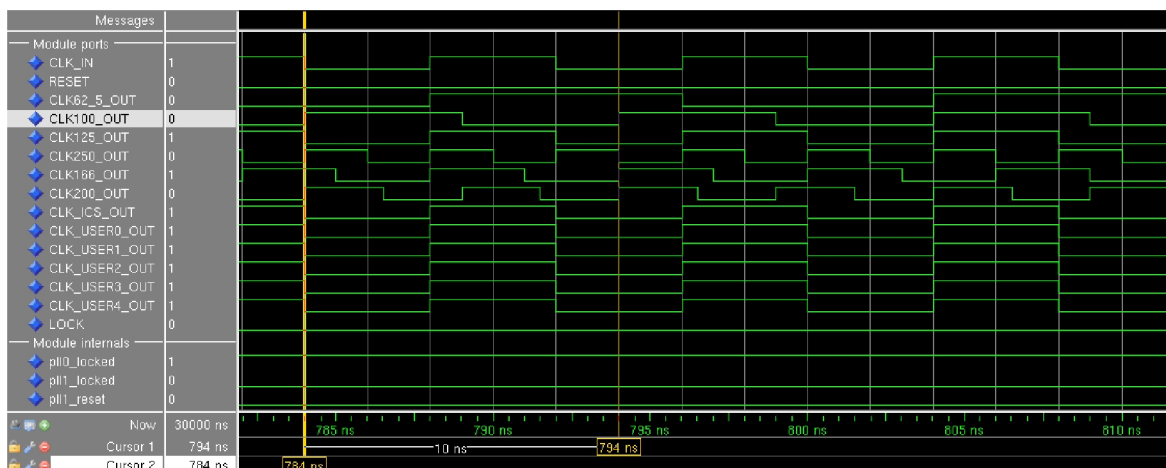
Příloha C Výsledky překladu - využití zdrojů

Příloha D Obsah přiloženého CD

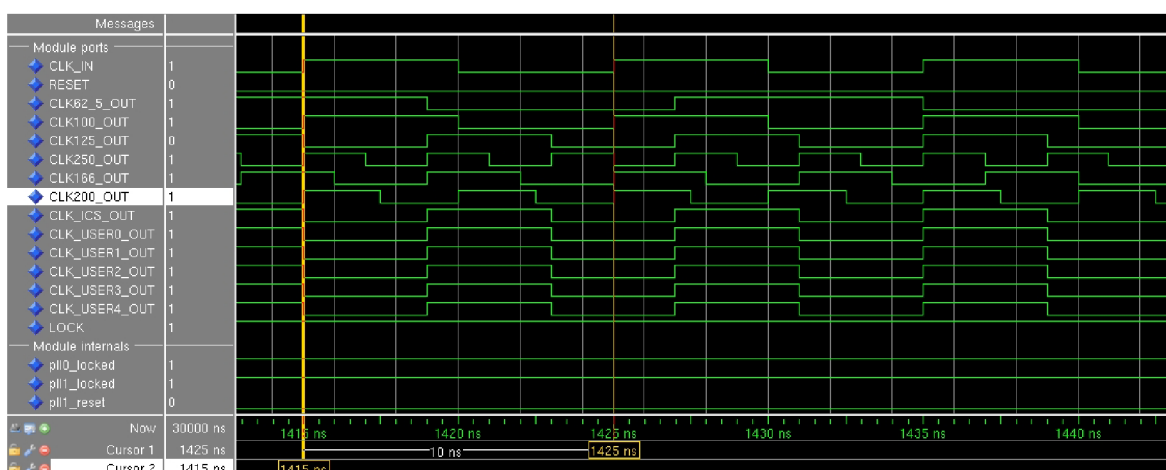
Příloha E CD nosič se zdrojovými kódy a elektronickou verzí této práce

Příloha A

Výsledky simulace úpravy hodinového modulu



Obrázek A.1: Simulace hodinového modulu na ComboV2



Obrázek A.2: Simulace hodinového modulu na NetFPGA

Příloha B

Výsledky simulační verifikace síťové karty

```
# -----  
# -- RX Scoreboard 0 TRANSACTION TABLE  
# -----  
# Size:                0  
# Items added:         5006  
# Items removed:       3876  
# Items discarded due to buffer overflow:    1130  
# -----  
# -----  
# -- RX Scoreboard 1 TRANSACTION TABLE  
# -----  
# Size:                0  
# Items added:         4954  
# Items removed:       4175  
# Items discarded due to buffer overflow:    779  
# -----  
# -----  
# -- RX Scoreboard 2 TRANSACTION TABLE  
# -----  
# Size:                0  
# Items added:         5001  
# Items removed:       3949  
# Items discarded due to buffer overflow:    1052  
# -----  
# -----  
# -- RX Scoreboard 3 TRANSACTION TABLE  
# -----  
# Size:                0  
# Items added:         4911  
# Items removed:       4221  
# Items discarded due to buffer overflow:    690  
# -----
```

```
# -----  
# -- TX Scoreboard 0 TRANSACTION TABLE  
# -----  
# Size:           0  
# Items added:    8596  
# Items removed:  8596  
# -----  
# -----  
# -- TX Scoreboard 1 TRANSACTION TABLE  
# -----  
# Size:           0  
# Items added:    10409  
# Items removed:  10409  
# -----  
# -----  
# -- TX Scoreboard 2 TRANSACTION TABLE  
# -----  
# Size:           0  
# Items added:    10041  
# Items removed:  10041  
# -----  
# -----  
# -- TX Scoreboard 3 TRANSACTION TABLE  
# -----  
# Size:           0  
# Items added:    10954  
# Items removed:  10954  
# -----
```

Příloha C

Výsledky překladač - využití zdrojů

C.1 Výsledky pro NetFPGA-10G na frekvenci 200 MHz

Slice Logic Utilization:

Number of Slice Registers:	46,587 out of 149,760	31%
Number used as Flip Flops:	46,586	
Number used as Latches:	1	
Number of Slice LUTs:	40,489 out of 149,760	27%
Number used as logic:	30,590 out of 149,760	20%
Number using O6 output only:	26,169	
Number using O5 output only:	2,362	
Number using O5 and O6:	2,059	
Number used as Memory:	9,447 out of 39,360	24%
Number used as Dual Port RAM:	1,492	
Number using O6 output only:	162	
Number using O5 output only:	194	
Number using O5 and O6:	1,136	
Number used as Single Port RAM:	242	
Number using O6 output only:	188	
Number using O5 output only:	2	
Number using O5 and O6:	52	
Number used as Shift Register:	7,713	
Number using O6 output only:	7,710	
Number using O5 output only:	3	
Number used as exclusive route-thru:	452	
Number of route-thrus:	2,917	
Number using O6 output only:	2,808	
Number using O5 output only:	103	
Number using O5 and O6:	6	

Slice Logic Distribution:

Number of occupied Slices:	20,273 out of 37,440	54%
Number of LUT Flip Flop pairs used:	62,553	
Number with an unused Flip Flop:	15,966 out of 62,553	25%
Number with an unused LUT:	22,064 out of 62,553	35%
Number of fully used LUT-FF pairs:	24,523 out of 62,553	39%

Number of unique control sets:	3,000		
Number of slice register sites lost to control set restrictions:	5,966 out of 149,760	3%	

IO Utilization:

Number of bonded IOBs:	517 out of 680	76%
Number of LOcEd IOBs:	517 out of 517	100%
IOB Flip Flops:	4	
Number of bonded IPADs:	54	
Number of bonded OPADs:	48	

Specific Feature Utilization:

Number of BlockRAM/FIFO:	92 out of 324	28%
Number using BlockRAM only:	75	
Number using FIFO only:	17	
Total primitives used:		
Number of 36k BlockRAM used:	65	
Number of 18k BlockRAM used:	12	
Number of 36k FIFO used:	1	
Number of 18k FIFO used:	16	
Total Memory used (KB):	2,880 out of 11,664	24%
Number of BUFG/BUFGCTRLs:	17 out of 32	53%
Number used as BUFGs:	17	
Number of BSCANs:	1 out of 4	25%
Number of BUFDSs:	3 out of 24	12%
Number of CRC64s:	8 out of 48	16%
Number of GTX_DUALs:	12 out of 24	50%
Number of PCIEs:	1 out of 1	100%
Number of PLL_ADVs:	3 out of 6	50%
Number of SYSMONs:	1 out of 1	100%

C.2 Výsledky pro COMBOv2-10G2 na frekvenci 200 MHz

Slice Logic Utilization:

Number of Slice Registers:	21,713 out of 97,280	22%
Number used as Flip Flops:	21,705	
Number used as Latches:	8	
Number of Slice LUTs:	23,787 out of 97,280	24%
Number used as logic:	18,297 out of 97,280	18%
Number using O6 output only:	15,771	
Number using O5 output only:	1,319	
Number using O5 and O6:	1,207	
Number used as Memory:	5,356 out of 26,240	20%
Number used as Dual Port RAM:	1,202	
Number using O6 output only:	154	
Number using O5 output only:	188	
Number using O5 and O6:	860	
Number used as Single Port RAM:	243	

Number using 06 output only:	192
Number using 05 output only:	1
Number using 05 and 06:	50
Number used as Shift Register:	3,911
Number using 06 output only:	3,911
Number used as exclusive route-thru:	134
Number of route-thrus:	1,518
Number using 06 output only:	1,444
Number using 05 output only:	69
Number using 05 and 06:	5

Slice Logic Distribution:

Number of occupied Slices:	10,287 out of	24,320	42%
Number of LUT Flip Flop pairs used:	31,310		
Number with an unused Flip Flop:	9,597 out of	31,310	30%
Number with an unused LUT:	7,523 out of	31,310	24%
Number of fully used LUT-FF pairs:	14,190 out of	31,310	45%
Number of unique control sets:	1,508		
Number of slice register sites lost to control set restrictions:	2,893 out of	97,280	2%

IO Utilization:

Number of bonded IOBs:	498 out of	640	77%
Number of LOCed IOBs:	498 out of	498	100%
IOB Flip Flops:	16		
Number of bonded IPADs:	38		
Number of bonded OPADs:	32		

Specific Feature Utilization:

Number of BlockRAM/FIFO:	33 out of	212	15%
Number using BlockRAM only:	31		
Number using FIFO only:	2		
Total primitives used:			
Number of 36k BlockRAM used:	25		
Number of 18k BlockRAM used:	7		
Number of 36k FIFO used:	2		
Total Memory used (KB):	1,098 out of	7,632	14%
Number of BUFG/BUFGCTRLs:	11 out of	32	34%
Number used as BUFGs:	11		
Number of BUFDSs:	3 out of	8	37%
Number of CRC64s:	4 out of	16	25%
Number of DCM_ADVs:	2 out of	12	16%
Number of GTP_DUALs:	8 out of	8	100%
Number of PCIEs:	1 out of	1	100%
Number of PLL_ADVs:	3 out of	6	50%
Number of SYSMONs:	1 out of	1	100%

Příloha D

Obsah přiloženého CD

Adresář design-200mhz

Adresář `design-200mhz` obsahuje výsledky překladu aplikace síťové karty nad platformou NetCOPE upravenou pro karty NetFPGA-10G.

Adresář doc

Adresář `doc` obsahuje zdrojové kódy textu této práce a použité obrázky. Součástí je také soubor `Makefile`, který umožňuje práci vysázet s využitím systému L^AT_EX.

Adresář src

Adresář `src` je svn adresář projektu Inovační Voucher obsahující zdrojové kódy firmware síťové karty platformy NetCOPE přizpůsobené pro kartu NetFPGA-10G. Složitá adresářová struktura odpovídá vývoji platformy v rámci projektu Liberouter.

Soubor readme.txt

Soubor `readme.txt` obsahuje informace o spuštění simulace hodinového modulu, simulační verifikace síťové karty a o překladu aplikace. Taktéž stručně popisuje adresářovou strukturu projektu se zaměřením pouze na nezbytné soubory a adresáře, které byly v rámci této práce upravené či vytvořené.

Soubor xkoran01-bp.pdf

Soubor `xkoran01-bp.pdf` je vysázený text bakalářské práce s funkčními odkazy.