

PROGRAMMABLE PACKET EDITOR FOR MULTI-GIGABIT NETWORK DEVICES

Jan Pazdera

Master Degree Programme (1), FIT BUT

E-mail: xpazde08@stud.fit.vutbr.cz

Supervised by: Tomáš Martínek

E-mail: martinto@fit.vutbr.cz

ABSTRACT

This paper deals with design and implementation of the Procedural Unit for network traffic analysis and editation. Its purpose is to analyse an incoming network traffic and perform packet header editations to provide the proper packet delivery. The designed architecture has the following characteristics. It is based on the stream processor concept which allows to process independent stream elements (i.e. packets) in parallel. Multiply stream clients can be used to process the same stream data concurrently. The stream clients can be driven either autonomously or by program. The packets are processed according to the incoming metadata and program and after editation, transmitted to the output. The Procedural Unit has been implemented in VHDL language. The target technology is Field Programmable Gate Array (FPGA).

1 ÚVOD

V posledním desetiletí jsme svědky výrazného rozvoje Internetu. Objem přenášených dat každým dnem roste a to společně se zvyšujícím se počtem uživatelů přináší stále vyšší nároky na systémy zabezpečující chod celosvětové sítě (dále jen uzly).

Mezi jednotlivými uzly mohou být přenášená data interpretována různými kódy. Data mohou být komprimována nebo šifrována. Při přenosu paketů je nutné provádět úpravy jejich hlaviček, aby se zajistilo správné doručení. Realizaci těchto úkolů zajišťují paketové editory.

Vzhledem k objemům dat přenášených na páteřních sítích není možné paketové editory realizovat softwarově s využitím univerzálního procesoru PC. Z tohoto důvodu je vhodné přesouvat implementaci na úroveň aplikačně specifických obvodů (ASIC) nebo *programovatelných hradlových polí* (FPGA), které budou schopny řídit činnost celého zařízení s dostatečnou rychlostí. Technologie FPGA oproti ASIC navíc umožňuje průběžně vylepšovat vlastnosti systému. Tato práce se zabývá návrhem a implementací procesní jednotky pro editaci síťového provozu v FPGA (dále jen Procesní jednotka). Je realizována jako proudový procesor (stream processor) vybavený volitelným počtem 16-bitových výpočetních jader Gena [4] s architekturou RISC a řadiči jednotlivých proudů realizujících jejich řízení a komunikaci s externím úložištěm paketů. Každé výpočetní jádro může být vybaveno volitelným počtem různých koprocesorů, které mohou autonomně vykonávat některé výpočetně náročné operace.

2 PROCESNÍ JEDNOTKA

Procesní jednotka provádí editace paketů paralelně z N vstupních rozhraní a to na základě sady procedur, které definují postup pro zpracování různých druhů paketů. Informace o jejich druhu je obsažena v metadatech, generovaných vnější analyzační jednotkou. Ta také nesou adresu do externí paměti, do níž je paket před zpracováním uložen. Zpracovávaná data jsou tedy organizována v N-krát dvou proudcích - proudě metadat a proudě dat paketů, generovaném z externí paměti. Do Datové cache Aplikačního procesoru se na základě druhu paketu nahraje příslušná sada procedur nezbytných pro jeho zpracování. Tato činnost je realizována autonomně pomocí proudových klientů bez pozornosti Aplikačního procesoru (viz níže).

Jakmile se ve vstupním proudě objeví první data, začne Aplikační procesor provádět instrukce ze své instrukční paměti, která obsahuje program prováděný nad každým paketem nezávisle na jeho druhu. Na konkrétních pozicích jsou přítomny odkazy na procedury v Datové Cache, jejichž provedení zajistí realizaci závislých editací.

2.1 ARCHITEKTURA

Procesní jednotka (PJ) se skládá z N editačních jednotek (*Editation Unit - EU*) a jednoho řadiče paměti procesní jednotky (*OPE Memory Controller*). Každá editační jednotka zpracovává pakety z jednoho vstupního rozhraní a je realizována jako proudový procesor [1][2]. Architektura PJ a EU je znázorněna na obrázku 1.

Paměť procesní jednotky obsahuje sady procedur, které definují způsob zpracování konkrétních paketů. Při jejich načítání jsou potřebné procedury přeneseny do Datové cache Aplikačního procesoru příslušné editační jednotky.

Jádrem každé editační jednotky je Aplikační procesor, postavený na generickém procesorovém jádru GENA [4], který vykonává program a řídí všechny proudové klienty. Aplikační procesor (AP) i proudoví klienti mohou společně přistupovat k registrům v *Proudovém registrovaném poli* (PRP).

2.2 PROUDOVÍ KLIENTI

Metadata Stream Controller (MSC), má na starosti komunikaci s analyzační jednotkou a načítání elementů metadat.

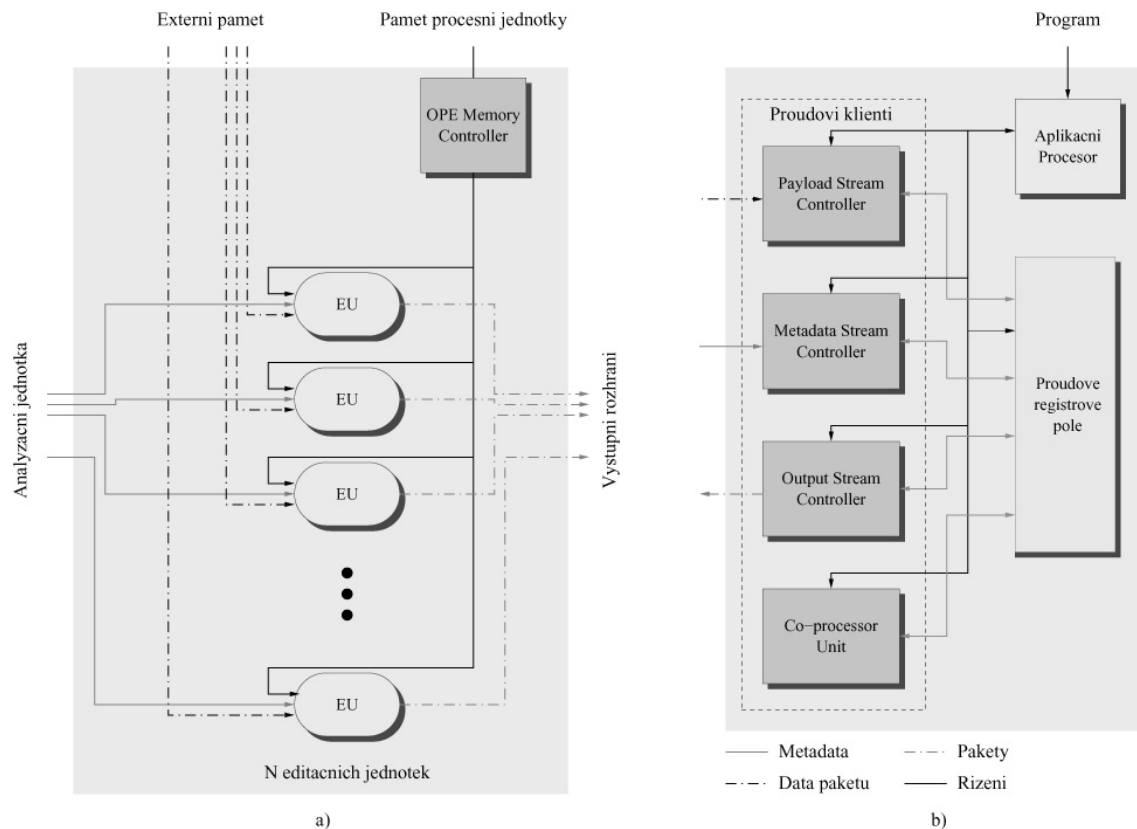
Payload Stream Controller (PSC) zajišťuje generování proudů dat a jeho správu. Obsahuje Vstupní buffer (VB), do něhož se ukládají data z proudů. VB je rozdělen do dvou banků, přičemž v jednom banku jsou data zpracovávaného paketu (aktivní bank) a do druhého se načítají data dalšího paketu.

Data v aktivním banku tvoří *okno* v datech paketu. K datům v okně může přistupovat AP, který může okno posouvat směrem ke konci paketu. Je-li okno posunuto, jsou data před oknem uvolněna z bufferu a na jejich místo se načtou nová data. Z tohoto důvodu nelze oknem pohybovat zpět.

S oknem lze manipulovat pomocí *operací vstupního proudů* (OVP) a souboru 16-bitových read-only registrů v PRP. OVP poskytují Aplikačnímu procesoru účinné prostředky pro práci nad proudem dat a jsou realizovány autonomně v PSC.

Output Stream Controller slouží ke generování výstupního proudů. Je řízen jednak některými OVP a také Aplikačním procesorem skrze registry v PRP.

Co-processor Unit obsahuje pomocné koprocesory Aplikačního procesoru. V současné aplikaci je využit pouze jeden koprocesor - Checksum Computer (CSC), který autonomně počítá



Obrázek 1: a) Architektura Procesní jednotky b) Architektura Editační jednotky

kontrolní součet nad odesílanými daty a výsledek uchovává v registru v PRP.

3 ZÁVĚR

Procesní jednotka je implementována v rámci projektu Liberouter [3] a je součástí výzkumného záměru CESNETu Programovatelný hardware. Dosahuje vysoké propustnosti v řádu několika Gb/s v závislosti na složitosti editačního programu. Použitý koncept proudového procesoru poskytuje výbornou škálovatelnost navržené architektury a umožňuje zpracovávat více toků zároveň bez snížení výkonnosti a tím dále zvyšovat propustnost.

REFERENCE

- [1] Kapasi, U., J., Rixner, S., Dally, W., J., Khailany, B., Ahn, J., H., Mattson, P., Owens, J., D.: Programmable Stream Processors, Published by the IEEE Computer Society, Srpen 2003
- [2] Khailany, B., Dally, W., J., Rixner, S., Kapasi, U., J., Owens, J., D., Towles, B.: Exploring the VLSI Scalability of Stream Processors, Stanford University a Rice University
- [3] The Liberouter Webpage, 2006. Dokument dostupný na <http://www.liberouter.org>
- [4] Mikušek, P.: Generic Nanoprocessor (GENA), 2006. Dokument dostupný na http://www.liberouter.org/vhdl_design/generated/index.php.