

# Technická zpráva



Akademie věd České republiky  
Ústav teorie informace a automatizace AV ČR, v.v.i.

## Akcelérátor pro výpočet odezvy ADSL vedení

Ing. Jan Kloub, Ing. Antonín Heřmánek PhD.  
{kloub, hermanek}@utia.cas.cz, +420-2-6605 2511

### Obsah

1	Úvod	1
2	Implementace FIR filtru	1
3	Struktura akcelérátoru	3
4	Ověření funkce	5
4.1	Formát přenášených dat . . . . .	6
4.2	Ověření výsledků v prostředí Matlab . . . . .	7
5	Výsledky	9
6	Výpis obsahu CD-ROM	10

### Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	12.12.2007	Kloub	Vytvoření dokumentu
1			
2			
3			
4			

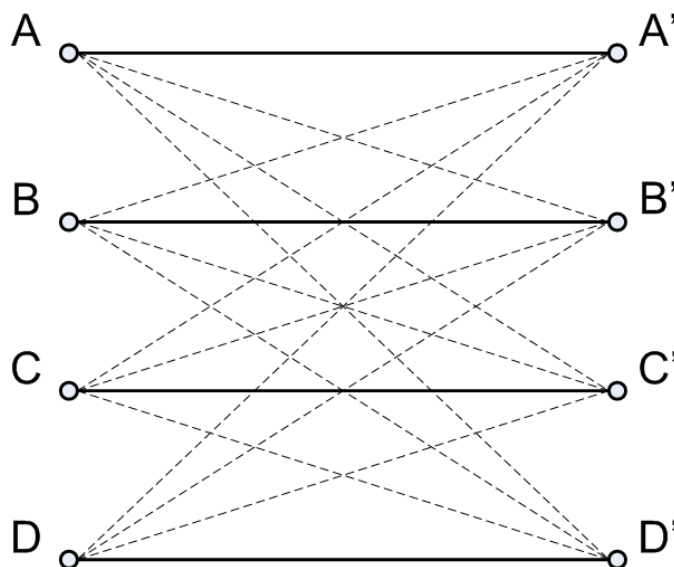
# 1 Úvod

Tento dokument popisuje implementaci akcelérátoru pro modelování ADSL vedení na obvodu FPGA.

ADSL vedení je složeno z několika párů vodičů, po kterých probíhá komunikace. Každý pár vodičů má svou přenosovou charakteristiku. Mezi vodiči dochází k přeslechům a proto je nutné při výpočtu odezvy celého vedení počítat s parazitními příspěvky od jednotlivých párů vodičů.

Výsledná odezva celého vedení je určena odezvami jednotlivých párů včetně parazitních příspěvků od ostatních párů vodičů.

Na obrázku 1 jsou znázorněny všechny možné přenosové cesty mezi čtyřmi páry vedení (parazitní příspěvky jsou znázorněny čárkovaně).



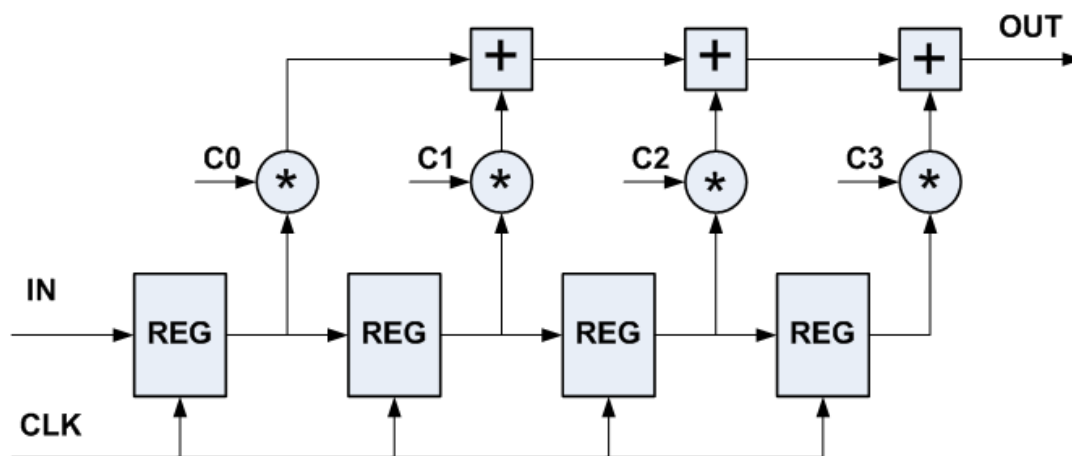
Obrázek 1: Možné přenosové cesty mezi čtyřmi páry vodičů

Struktura dokumentu je následující: v kapitole 2 jsou popsány příklady implementace filtrů akcelérátoru, v kapitole 3 je popsána struktura akcelérátoru a v kapitole 4 je popsán postup pro ověření funkce akcelérátoru.

**Poděkování:** Tato práce vznikla za podpory projektu číslo 1ET300750402 Grantové agentury AV ČR.

## 2 Implementace FIR filtru

Každou reálnou přenosovou cestu lze v modelu nahradit příslušným filtrem. Pro sestavení modelu jsou využity filtry s konečnou impulsní odezvou (FIR - Finite Impulse Response).



Obrázek 2: Realizace FIR filtru s paralelním výpočtem odezvy

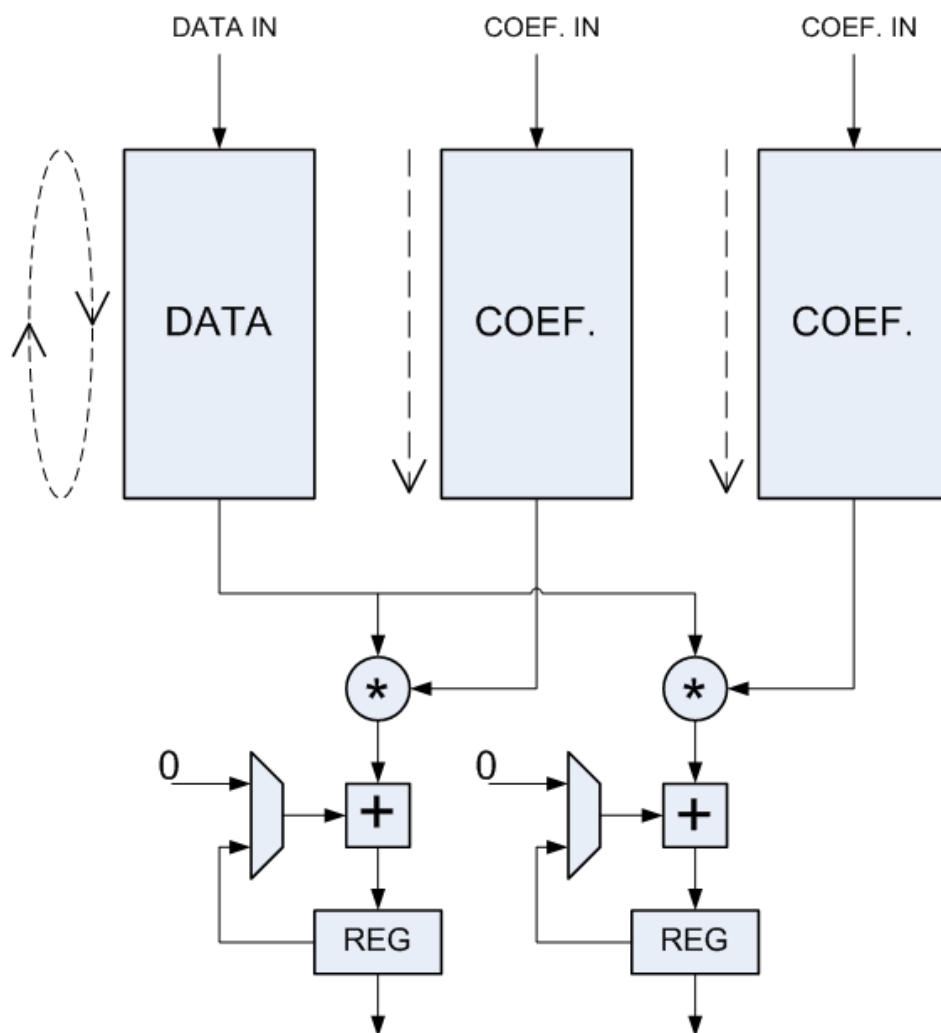
FIR filtr lze implementovat pomocí posuvného registru, který udržuje část historie příchozích vzorků. Jedna hodnota odezvy filtru je vypočítána na základě hodnot uchované historie vzorků a příslušných hodnot koeficientů filtru. Tato hodnota lze určit v jeden časový krok pomocí paralelního zapojení násobiček a sčítaček (obr. 2).

Paralelní zapojení využívá velké množství hardwarových zdrojů. Akcelerátor pro výpočet modelu vedení bude obsahovat poměrně velké množství filtrů, a proto toto zapojení nelze použít.

Pro každý filtr akcelerátoru bude použita jen jedna násobička a jedna sčítačka, které budou sdíleny pro postupný výpočet odezvy filtru. Výpočet bude rozdělen na několik sekvenčních kroků. Koeficienty a historie příchozích vzorků budou udržovány v blokových pamětech.

Pro každou přenosovou cestu bude implementován jeden filtr. Jak je vidět z obrázku 1 bude mít několik filtrů stejná vstupní data, a proto může být paměť pro ukládání vzorků sdílena několika filtry.

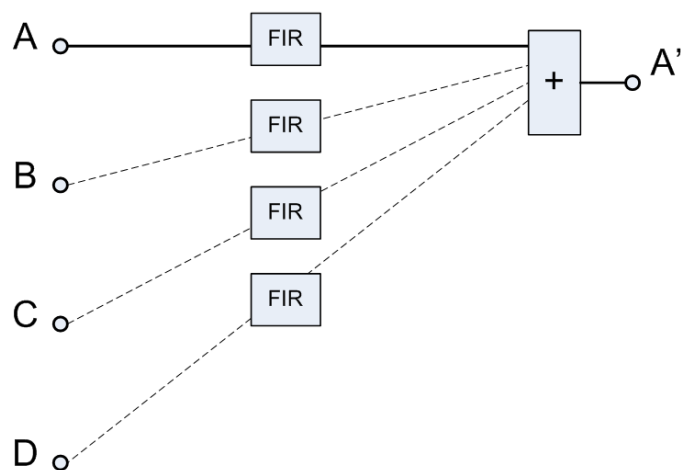
Na obrázku 3 je znázorněna struktura dvou filtrů se společnou pamětí pro vstupní data. Paměti pro data a koeficienty jsou dvoubranové (jedna brána je určena pro čtení a druhá pro zápis). Příslušné hodnoty koeficientů je nutné do pamětí nahrát před zahájením výpočtu. Paměť pro data je využita jako kruhový buffer, který udržuje část historie vstupních vzorků. Na výstupu filtrů je umístěn registr, do kterého se ukládají mezivýsledky výpočtu. Na začátku výpočtu jedné hodnoty odezvy je do registru zapsána hodnota prvního součinu a dále jsou další hodnoty součinů přičítány. Na konci průchodu přes všechny hodnoty koeficientů je v registru aktuální hodnota odezvy.



Obrázek 3: Realizace dvou sekvenčních FIR filtrů se společnou pamětí pro historii vzorků

### 3 Struktura akcelérátoru

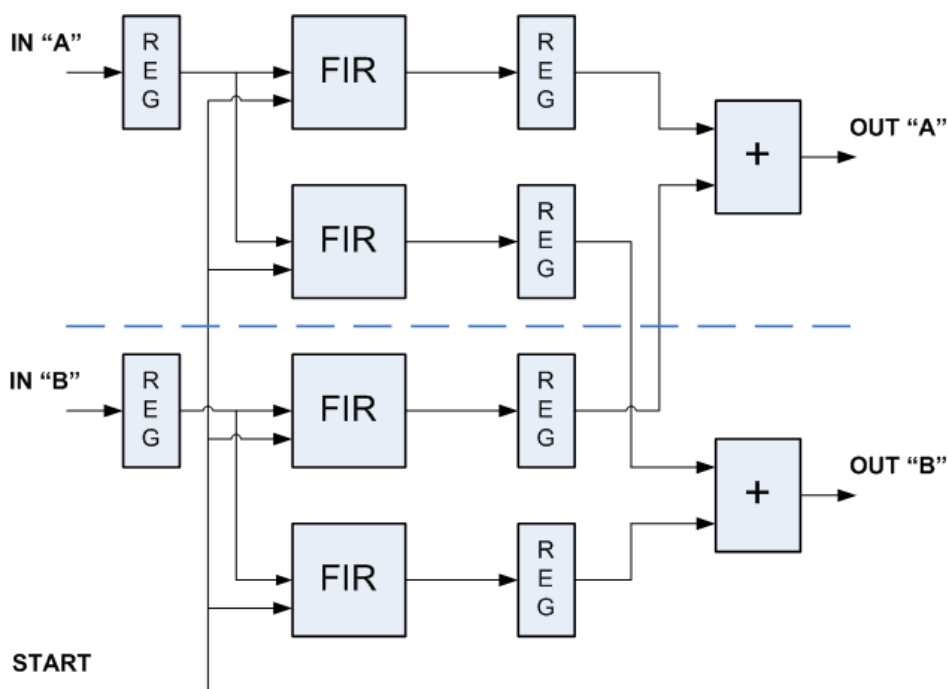
Akcelérátor umožňuje určit odezvu na výstupech modelu vedení na základě znalosti struktury a vlastností přenosových cest vedení. Na obrázku 4 je znázorněn příklad výpočtu odezvy jednoho výstupu vedení.



Obrázek 4: Způsob sčítání jednotlivých odezev filtrů

V závislosti na složitosti problému a vzhledem k omezeným zdrojům obvodů FPGA, je možné implementovat pouze určitou část přenosových cest. Výsledná odezva všech přenosových cest lze dekomponovat a akcelerátor použít v několika iteracích výpočtu. Koeficienty jednotlivých filtrů lze za běhu měnit.

Výpočet odezev jednotlivých filtrů je prováděn souběžně, a proto musí být všechny vstupní hodnoty platné v okamžik spuštění výpočtu. Po dokončení výpočtu jsou příslušné výstupy sečteny a lze zpracovávat další vstupní data. Obrázek 5 znázorňuje kompletní strukturu akcelerátoru pro dva páry komunikačních linek.



Obrázek 5: Struktura akcelerátoru pro dva páry komunikačních linek

Struktura akcelerátoru nemusí být symetrická. Podle zvolených parametrů lze například určit příspěvky z několika vedení na všechny ostatní vedení daného modelu atp.

Akcelerátor používá celočíselnou aritmetiku se znaménkem a je tedy nutné vstupní data příslušným způsobem kvantizovat.

Datová šířka výstupních dat je odvozena z šířky vstupních dat, šířky koeficientů a operací nad nimi, tak

aby nedošlo k přetečení výsledku. Ve výrazu 1 je vyjádřena výstupní datová šířka, kde SD je datová šířka vstupních dat, SC je datová šířka koeficientů filtrů, NC je počet koeficientů filtrů a NT je počet vstupních bodů modelu vedení.

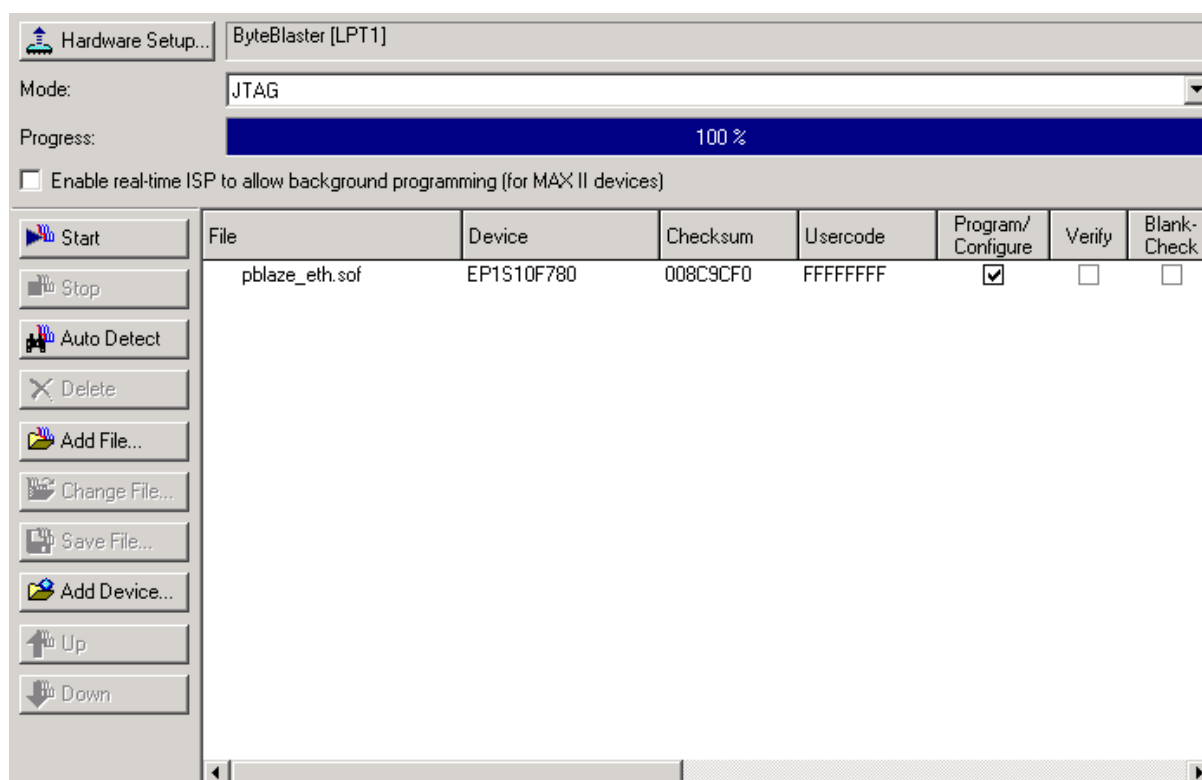
$$SD + SC + \lceil \log_2(NC) \rceil + \lceil \log_2(NT) \rceil \quad (1)$$

## 4 Ověření funkce

Ověření akcelerátoru je možné nahráním příslušného bitstreamu (.sof) do následující vývojové desky:

- Stratix II EP2S180 DSP Development Board (osazena obvodem Altera Stratix II EP2S180F1020C3)

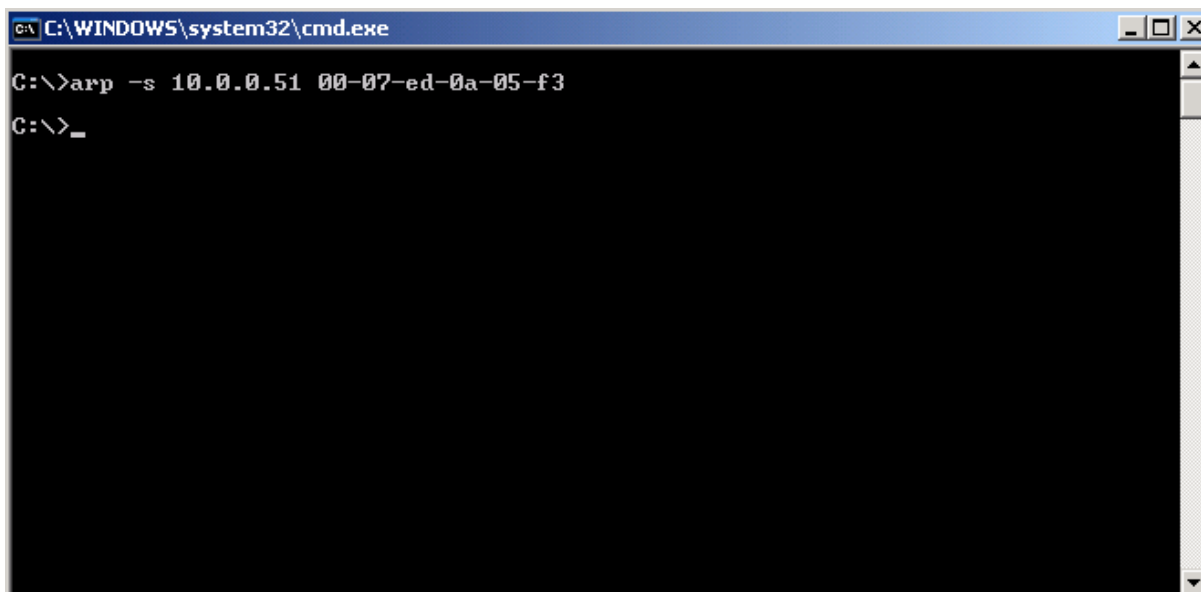
Bitstream je uložen na přiloženém datovém nosiči a lze jej nahrát pomocí downloaderu vývojového prostředí Quartus II (viz obrázek 6).



Obrázek 6: Dialogové okno downloaderu vývojového prostředí Quartus II.

Po nahrání bitstreamu vývojová deska čeká na příchozí ethernetový rámec obsahující data a po přijetí zobrazí na sedmi-segmentovém displeji počet přijatých vstupních vektorů, zpracuje data a vyše je přes ethernetové rozhraní. Rámec musí být ve formátu UDP a číslo cílového portu rovno 2000. Vývojová deska nemá přiřazenu žádnou IP adresu a nepodporuje protokol ARP. Pro komunikaci s vývojovou deskou je nutné přidat statický záznam v ARP tabulce. IP adresu zvolíme libovolnou a MAC adresa je 00:07:ed:0a:05:f3. V prostředí MS Windows vytvoříme statický záznam příkazem `arp -s XX.XX.XX.XX 00-07-ed-0a-05-f3`, kde XX.XX.XX.XX je libovolná zvolená adresa (viz obrázek 7).

Velikost dat zasílaných pro akcelerátor je omezena velikostí jednoho ethernetového rámce. Jedním ethernetovým rámcem lze pomocí protokolu UDP přenést maximálně 1272 bytů. Součástí zasílaných dat jsou i informace o významu zasílaných dat. Formát a význam dat je popsán v kapitole 4.1.



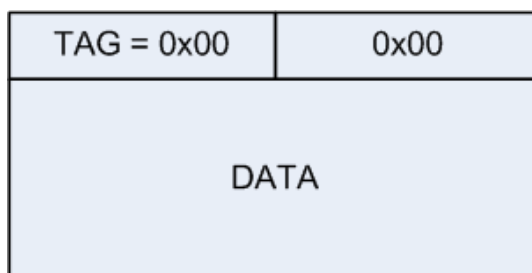
Obrázek 7: Vytvoření statického záznamu ARP tabulky v prostředí MS Windows.

#### 4.1 Formát přenášených dat

Akcelerátor připojený přes ethernetové rozhraní musí rozeznávat různý význam příchozích dat. Význam dat je určen prvními datovými slovy přijatých dat - takzvanými "tagy". Rozlišováno je mezi následujícími typy dat:

- Hodnoty koeficientů a určení filtru pro který jsou koeficienty určeny.
- Vstupní hodnoty akcelerátoru.
- Příkaz pro reset akcelerátoru a vynulování hodnot v paměti filtrů.

Rámec obsahující vstupní data pro akcelerátor začíná dvěma nulovými byty (obr. 8). Data jsou 16-ti bitová a jsou postupně čtena na vstup akcelerátoru, dokud nejsou platné všechny vstupy akcelerátoru. Po načtení hodnot pro všechny vstupy je spuštěn výpočet akcelerátoru. Po dokončení výpočtu jsou načteny další hodnoty z rámce a výpočet je opakován dokud zbývají vstupní data.



Obrázek 8: Formát rámce obsahující vstupní data akcelerátoru

Po zpracování všech vstupních dat, nebo pokud již akcelerátor naplnil ethernetový rámec, jsou výsledná data (nebo jejich část) zaslána zpět. Výsledná data jsou 64 bitová a jsou ukládána cyklicky přes všechny výstupní body modelu.

Koeficienty pro jednotlivé filtry jsou také zasílány pomocí datových rámců uvedeným tagem rovným hodnotě 1 a určením filtru. Na obrázku 9 je uveden formát datového rámce pro uložení koeficientů do daného filtru. Konkrétní filtr je určen dvojicí hodnot IN\_ID a FIR\_ID. Hodnota pole IN\_ID určuje skupinu filtrů pro daný vstupní bod modelu vedení a hodnota FIR\_ID určuje výstupní bod modelu.

TAG = 0x01	IN_ID
FIR_ID	XX
COEFICIENTS	

Obrázek 9: Formát rámce obsahující koeficienty pro filtry akcelérátoru

Přijetí koeficientů je potvrzeno rámcem se stejným tagem jaký byl v přijatém rámci s koeficienty.

Pro vynulování hodnot paměti všech filtrů je použit rámec s tagem rovným hodnotě 2 (viz obrázek 10). Na dalším obsahu rámce nezáleží. Reset je potvrzen obdobně jako v případě potvrzení přijetí koeficientů.

TAG = 0x02	XX
XX	

Obrázek 10: Formát rámce pro reset filtrů akcelérátoru

## 4.2 Ověření výsledků v prostředí Matlab

Správnost výpočtu akcelérátoru je možné ověřit pomocí prostředí Matlab. Pro účely ověření jsou napsány skripty a programy pro komunikaci s akcelérátorem a následné ověření výsledků.

Akcelérátor používá celočíselnou aritmetiku a je tedy nutné příslušné hodnoty vstupních dat a koeficientů kvantizovat.

Funkce pro komunikaci jsou implementovány v jazyce C. Zdrojové kódy lze přeložit v prostředí Matlab do MEX souboru. Pop přeložení lze funkce volat a předávat jim hodnoty z prostředí Matlab pomocí parametrů.

Překlad zdrojových kódů v Matlabu lze provést například následujícím příkazem:

```
mex -O sendudp.c C:\pf\MATLAB\R2007a\sys\lcc\lib\wsock32.lib -DWIN32
```

V jazyce C jsou implementovány tři funkce - funkce pro zasílání a příjem dat (sendudp), funkce pro zaslání koeficientů do všech filtrů akcelérátoru (sendcoef) a funkce pro reset akcelérátoru (sendrst). Parametry funkcí a jejich význam je uveden v tabulce 1

Výsledky výpočtu akcelérátoru lze porovnat s výsledky funkce implementované v Matlabu. Tato funkce se nazývá fir\_net a je dostupná M-souboru na přiloženém datovém nosiči. Význam parametrů funkce je uveden v tabulce 2.

V souboru fir\_ru\_test.m je napsán jednoduchý skript pro otestování akcelérátoru náhodnými hodnotami dat a koeficientů a porovnání výsledků s referenčním modelem popsáním v Matlabu.



<b>Funkce:</b>	<b>Význam parametrů</b>
<code>sendcoef(target, port, nf, nt, c_net);</code>	target - IP adresa akcelerátoru port - komunikační port akcelerátoru nf - počet výstupních bodů modelu nt - počet vstupních bodů modelu c_net - matice hodnot koeficientů akcelerátorů (každý řádek matice náleží jednomu filtru - pořadí řádků odpovídá postupně pro všechny filtry v prvním vstupním bodu modelu a podobně pro další vstupní body)
<code>sendrst(target, port);</code>	target - IP adresa akcelerátoru port - komunikační port akcelerátoru
<code>y_hw = sendudp(target, port, nf, nt, data);</code>	y_hw - výstupní data (každý řádek matice odpovídá jednomu výstupnímu bodu modelu) target - IP adresa akcelerátoru port - komunikační port akcelerátoru nf - počet výstupních bodů modelu nt - počet vstupních bodů modelu data - matice se vstupními daty pro akcelerátor (každý řádek matice odpovídá jednomu vstupnímu bodu modelu)

Tabulka 1: Funkce prostředí Matlab pro komunikaci s akcelerátorem a význam jejich parametrů

<b>Funkce:</b>	<b>Význam parametrů</b>
<code>y_sw = fir_net(data, c_net, nf, nt);</code>	y_sw - výstupní data (každý řádek matice odpovídá jednomu výstupnímu bodu modelu) data - matice se vstupními daty pro akcelerátor (každý řádek matice odpovídá jednomu vstupnímu bodu modelu) c_net - matice hodnot koeficientů akcelerátorů (každý řádek matice náleží jednomu filtru - pořadí řádků odpovídá postupně pro všechny filtry v prvním vstupním bodu modelu a podobně pro další vstupní body) nf - počet výstupních bodů modelu nt - počet vstupních bodů modelu

Tabulka 2: Funkce pro výpočet modelu vedení v prostředí Matlab a význam parametrů

## 5 Výsledky

Akcelerátor byl implementován pro 8 vstupních a 25 výstupních bodů modelu vedení. Jednotlivé FIR filtry mají 512 koeficientů a počítají 16-ti bitová celočíselná aritmetice se znaménkem. Výstupní data jsou reprezentována jako 64 bitová, celočíselná se znaménkem.

Akcelerátor je implementován na obvodu FPGA Stratix II EP2S180F1020C3 firmy Altera. Dosažené výsledky implementace na obvodu FPGA jsou shrnuty v tabulce 3

<b>Max. pracovní frekvence</b>	50 MHz
<b>ALUT</b>	56,174 / 143,520 ( 39 % )
<b>Paměť obvodu</b>	1,778,288 / 9,383,040 ( 19 % )
<b>DSP bloků</b>	400 / 768 ( 52 % )

Tabulka 3: Tabulka parametrů implementace akcelerátoru v obvodu FPGA Stratix II EP2S180F1020C3 firmy Altera

Doba výpočtu jednoho vektoru výstupů akcelerátoru lze určit z jeho známé struktury. Počet hodinových cyklů je dán vztahem 2, kde NC je počet koeficientů filtrů a NT je počet vstupních bodů modelu vedení.

$$NC + 3 + \lceil \log_2(NT) \rceil \quad (2)$$

Doba výpočtu, včetně režie přenosu dat přes ethernetové rozhraní a jeho uložení do paměti akcelerátoru a hostitelského počítače, je naměřena v prostředí Matlab pomocí funkcí `tic` a `toc`.

Naměřené hodnoty jsou uvedeny v tabulce 4.

<b>Velikost matice vstupních dat</b>	<b>Doba výpočtu akcelerátoru</b>	<b>Doba výpočtu Matlabu</b>
8 x 512	0.0233 s	0.0998 s
8 x 1024	0.0449 s	0.1793 s
8 x 2048	0.0883 s	0.3585 s
8 x 4096	0.1761 s	0.7141 s
8 x 8192	0.3496 s	1.4306 s

Tabulka 4: Tabulka naměřených hodnot doby výpočtu akcelerátoru

K době výpočtu je nutné přičíst dobu inicializace koeficientů akcelerátoru. Tato inicializace je potřeba provést před zahájením výpočtu a případně pak při jakémkoliv změně koeficientů. Pro stávající implementaci doba nahrání všech koeficientů, tedy odeslání  $8 * 25 * 512$  hodnot, trvá 0,066 s. Koeficienty mohou být zasílány pouze do vybraných filtrů akcelerátoru.

## 6 Výpis obsahu CD-ROM

Na CD se nachází text tohoto dokumentu, skripty pro Matlab a bitstream pro obvod FPGA osazený na vývojové desce "Stratix II EP2S180 DSP Development Board".

Přiložené CD má následující adresářovou strukturu:

```
.
|-- doc/                text dokumentu ve formátu PDF
|-- fir_8_25_512        implementace akcelerátoru
|   '- pblaze_eth
|       '- STRATIXII_EDIF  Projekt nástroje Quartus II a bitstream
|-- matlab/
|   |-- macrogen/        skript pro vygenerování makro procedur a parametrů
|   '- model/            Model akcelerátoru v Matlabu a příslušné funkce pro
|                           ověření funkce hardwarového akcelerátoru
'-- readme.txt
```

## Reference

- [1] Davídek, V; Sovka, P. *Číslicové zpracování signálů a implementace*.  
Praha: Vydavatelství ČVUT, 1996.  
ISBN 80-01-01530-0