



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

VYSOKORYCHLOSTNÍ KOMUNIKAČNÍ LINKA  
PRO AKVIZACI DAT

HIGH PERFORMANCE DATA ACQUISITION COMMUNICATION LINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Hadámek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Soběslav Valach

BRNO 2020

# Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. Jakub Hadánek

**ID:** 174301

**Ročník:** 2

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Vysokorychlostní komunikační linka pro akvizici dat

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je akvizice dat AD převodníkem a jejich přenos protokolem JESD204B/C do FPGA s následnou transformací a přenosem rozhraním Ethernet 100G nebo PCIe do PC.

- 1) Prostudujte vlastnosti a základní parametry AD převodníků
- 2) Prostudujte protokol JESD204B/C pro rychlosti do 16 Gbit/s na linku
- 3) Prostudujte rozhraní vhodná pro přenos dat rychlostí 100 Gb/s (Ethernet, PCIe)
- 4) Implementujte protokol JESD204B/C do FPGA
- 5) Ověřte integritu dat přicházejících rozhraním JESD204B/C
- 6) Implementujte rozhraní pro komunikaci s PC (Ethernet, PCIe) a vyberte vhodný protokol např. RoCE
- 7) Ověřte integritu datového spojení s PC
- 8) Proveďte transformaci dat z rozhraní JESD204B/C a odešlete je do PC
- 9) Demonstrujte funkcionální na reálném systému a zhodnoťte dosažené výsledky

### DOPORUČENÁ LITERATURA:

[1] JEDEC STANDARD: Serial Interface for Data Converters, JESD204C. EDEC Solid State Technology Association 2017. DECEMBER 2017

[2] <https://www.infinibandta.org/ibta-specifications-download/>

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 1.6.2020

**Vedoucí práce:** Ing. Soběslav Valach

**doc. Ing. Václav Jirsík, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Cílem práce je akvizice dat AD převodníkem a jejich přenos protokolem JESD204B do FPGA s následnou transformací a přenosem rozhraním 100G Ethernet nebo PCI Express do PC. První část práce je zaměřena na představení technologií a hardwaru a rozbor řešení. Druhá část práce popisuje samotné řešení a demonstruje jeho funkčnost. Byl vytvořen HDL design, který umožňuje použít pro přenos dat z převodníku obě výše zmíněná rozhraní. Dále byl napsán software pro OS linux, umožňující data přijímat a ukládat. Na závěr jsou zde prezentovány výsledky měření převodníkovou kartou a jejich rozbor.

## **Klíčová slova**

100Gbit Ethernet, RDMA, PCI Express, JESD204B, AD převodník, FPGA

## **Abstract**

The aim of this thesis is the acquisition of data from the AD converter and it's transfer via the JESD204B interface to FPGA with the following transformation and transfer to PC through 100G Ethernet or PCI Express interface. The first part of the thesis is focused on the introduction to used technologies and hardware and analysis of the solution of this project. Second part of the thesis describes solution and it's functionality. I created HDL design which allows to transfer data from AD converter using both of the interfaces mentioned above. I also created software application for OS Linux which allows to receive and store incoming data in PC. In the end, the results of the measurement using the converter board are presented and discussed.

## **Keywords**

100Gbit Ethernet, RDMA, PCI Express, JESD204B, AD converter, FPGA

## **Bibliografická citace**

HADÁMEK, Jakub. *Vysokorychlostní komunikační linka pro akvizici dat* [online]. Brno, 2020 [cit. 2020-06-01]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/126968>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Soběslav Valach.

## **Prohlášení**

„Prohlašuji, že svou diplomovou práci na téma Vysokorychlostní komunikační linka pro akvizici dat jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **31. května 2020**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Soběslavu Valachovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **31. května 2020**

.....  
podpis autora

# Obsah

1	Úvod.....	13
2	AD Převodníky.....	15
2.1	Parametry AD převodníku.....	15
2.2	Chyby.....	20
2.3	Zlepšení parametrů převodu.....	23
2.3.1	Oversampling.....	23
2.3.2	Prokládání.....	23
2.3.3	Průměrování.....	24
2.3.4	Dithering.....	24
3	Standard JESD204.....	26
3.1	Revize JESD204A.....	26
3.2	Revize JESD204B.....	27
3.2.1	Fyzická vrstva.....	28
3.2.2	Datová vrstva.....	32
3.2.3	Transportní vrstva.....	33
3.2.4	Deterministická latence.....	35
3.3	Revize JESD204C.....	36
4	Rozhraní 100G Ethernet.....	38
4.1	UDP/IP protokol.....	38
4.2	RDMA.....	38
4.3	RoCE.....	39
5	Sběrnice PCI Express.....	43
5.1	Topologie.....	44
5.2	Vrstvy.....	44
5.2.1	Fyzická.....	46
5.2.2	Linková.....	46
5.2.3	Transakční.....	47
5.3	Typy transakcí.....	47
5.4	Přenos dat.....	48
6	Hardware.....	49
7	Rozbor řešení.....	52
7.1	Získání dat z AD převodníku.....	53
7.2	Transformace dat.....	54
7.3	Přenos dat do PC.....	54
8	Design a Firmware.....	56
8.1	AD převodník, HMC7044 a JESD204B.....	57
8.2	Transformace dat.....	60

8.3	Aplikace využívající 100G Ethernet a RoCE .....	61
8.4	Aplikace využívající PCI Express .....	68
9	Software.....	71
9.1	RDMA aplikace .....	71
9.2	PCI Express aplikace.....	76
10	Výsledky.....	81
10.1	Ověření integrity dat rozhraní JESD204B.....	81
10.2	Ověření integrity dat rozhraní 100G Ethernet a PCI Express .....	82
10.3	Měření .....	83
11	Závěr .....	90
12	Literatura .....	92



# Seznam obrázků

Obrázek 1. Spojitý, diskrétní a digitální signál.....	16
Obrázek 2. Kvantování signálu .....	16
Obrázek 3. Frekvenční spektru periodického signálu .....	17
Obrázek 4. Frekvenční spektrum a aliasing.....	17
Obrázek 5. Změna SNR při rostoucí frekvenci signálu a konstantním jitteru [3] ....	19
Obrázek 6. Změna frekvenčního spektra signálu vlivem jitteru [3] .....	19
Obrázek 7. Vliv jitteru na vzorkování signálu [3] .....	20
Obrázek 8. Chyba offsetu .....	21
Obrázek 9. Chyba zisku.....	21
Obrázek 10. Nelinearita převodní charakteristiky (DNL) .....	22
Obrázek 11. Nelinearita převodní charakteristiky (INL) .....	22
Obrázek 12. Prokládání AD převodníků.....	23
Obrázek 13. Frekvenční spektrum bez a s použitím ditheringu [4] .....	25
Obrázek 14. Topologie rozhraní JESD204 .....	26
Obrázek 15. Topologie rozhraní JESD204A.....	27
Obrázek 16. Topologie rozhraní JESD204B.....	27
Obrázek 17. Vrstvy JESD204B .....	28
Obrázek 18. Signály použité v JESD204B.....	29
Obrázek 19. Serializér a deserializér .....	30
Obrázek 20. CDR blok přijímací části GTY transceiverů [7] .....	30
Obrázek 21. Blokové schéma GTY transceiverů [7].....	31
Obrázek 22. Místa vzorkování signálu v bloku CDR [7].....	31
Obrázek 23. Synchronizační fáze JESD204B .....	33
Obrázek 24. Zarovnání dat z více linek.....	33
Obrázek 25. Mapování vzorků v rámcích.....	34
Obrázek 26. Rámce v multirámcích.....	35
Obrázek 27. Synchronizace LMFC podtřída 1.....	35
Obrázek 28. Synchronizace LMFC podtřída 2.....	36
Obrázek 29. Topologie rozhraní JESD204C .....	36
Obrázek 30. Komunikace se serverem bez RDMA.....	39
Obrázek 31. Komunikace se serverem s RDMA.....	39
Obrázek 32. Struktura Infiniband paketu.....	39
Obrázek 33. Struktura RoCE paketu.....	40
Obrázek 34. Struktura RoCEv2 paketu .....	40
Obrázek 35. Infiniband – spojově orientované/neorientované přenosy .....	41
Obrázek 36. Infiniband Base Transport Header .....	41
Obrázek 37. Infiniband Datagram Extended Transport Header.....	42

Obrázek 38. PCI Express link a lane v konfiguraci x4 .....	43
Obrázek 39. Topologie sběrnice PCI Express.....	45
Obrázek 40. Propojení vrstev v PCIe.....	45
Obrázek 41. Složení PCIe paketu.....	46
Obrázek 42. Princip funkce BARů.....	48
Obrázek 43. Princip DMA přenosu.....	48
Obrázek 44. Blokové schéma CECILIE.....	49
Obrázek 45. Blokové schéma rozšiřující převodníkové karty .....	50
Obrázek 46. Blokové schéma řetězce s PCIe kartou Cecilie a rozšiřující převodníkovou kartou .....	50
Obrázek 47. PCIe karta Cecilie .....	50
Obrázek 48. Rozšiřující převodníková karta.....	51
Obrázek 49. Blokové schéma řešení.....	52
Obrázek 50. Tok dat v aplikaci pro FPGA.....	53
Obrázek 51. AXI4 Stream [16].....	54
Obrázek 52. Blokové schéma zapojení komponent AXI QSPI a SPI to ADI SPI .....	57
Obrázek 53. Blokové schéma zapojení JESD komponent v FPGA .....	59
Obrázek 54. Transformace dat z převodníku .....	60
Obrázek 55. Blokové schéma komponenty 100G UDP .....	62
Obrázek 56. Blokové schéma komponenty RoCE to 100G UDP .....	63
Obrázek 57. Blokové schéma aplikace RDMA APP.....	64
Obrázek 58. Typ write.....	65
Obrázek 59. Typ read .....	66
Obrázek 60. Typ info.....	66
Obrázek 61. Typ error.....	66
Obrázek 62. Simulace RDMA app s opakovaným zapínáním a vypínáním přenosu – strana aplikace.....	67
Obrázek 63. Simulace RDMA app s opakovaným zapínáním a vypínáním přenosu – strana 100G UDP modulu .....	68
Obrázek 64. Blokové schéma aplikace PCIE APP .....	70
Obrázek 65. Simulace PCIe app při aktivované synchronizaci a triggeru.....	70
Obrázek 66. Blokové schéma hlavního menu RDMA aplikace .....	73
Obrázek 67. Blokové schéma inicializace a průběhu RDMA přenosu.....	74
Obrázek 68. Aplikace cecilie_app po spuštění .....	75
Obrázek 69. Aplikace cecilie_app při hledání připojených karet .....	75
Obrázek 70. Aplikace cecilie_app při přenosu dat z karty do PC.....	76
Obrázek 71. Výsledek přenosu dat z karty do PC rozhraním 100G Ethernet .....	76
Obrázek 72. Blokové schéma hlavního menu PCIe aplikace .....	77
Obrázek 73. Blokové schéma inicializace PCIe přenosu jednoho kanálu.....	78

Obrázek 74. Blokové schéma inicializace PCIe přenosu obou kanálů.....	78
Obrázek 75. Blokové schéma vlákna pro přenos a uložení dat.....	79
Obrázek 76. Výsledky přenosu dat z karty do PC přes sběrnici PCIe.....	79
Obrázek 77. Aplikace pcie_app po spuštění.....	80
Obrázek 78. Aplikace pcie_app při nastavování triggeru.....	80
Obrázek 79. Aplikace pcie_app při přenosu dat z karty do PC .....	80
Obrázek 80. Kontrola PRBS IBERTem na JESD linkách.....	81
Obrázek 81. Inicializační fáze JESD přenosu zachycená logickým analyzátozem ....	82
Obrázek 82. Pilovitý průběh testovacího vzoru převodníku.....	82
Obrázek 83. Frekvenční spektrum signálu při koherentním vzorkování .....	84
Obrázek 84. Frekvenční spektrum signálu při nekoherentním vzorkování .....	84
Obrázek 85. Frekvenční spektrum signálu získané spektrálním analyzátozem .....	85
Obrázek 86. Velikosti významných složek frekvenčního spektra získané spektrálním analyzátozem.....	85
Obrázek 87. Sparkle code error.....	86
Obrázek 88. Současné měření na obou kanálech.....	87
Obrázek 89. Demonstrace použití triggeru.....	87
Obrázek 90. Frekvenční spektrum multi tónového signálu.....	88
Obrázek 91. Frekvenční spektrum při aliasingu .....	88
Obrázek 92. Soustava použitá pro měření a analýzu signálu.....	89
Obrázek 93. Soustava použitá pro měření s triggerem.....	89

## Seznam tabulek

Tabulka 1. Porovnání rychlostí jednotlivých generací PCIe .....	44
Tabulka 2. Sumarizace použitých HDL modulů a firmwaru (zelená – mnou vytvořeno, oranžová – nakonfigurováno, modrá – dostal jsem k dispozici) .....	56
Tabulka 3. Seznam a popis registrů aplikace RDMA APP .....	64
Tabulka 4. Typy zpráv podporovaných UDP registry .....	65
Tabulka 5. Seznam a popis PCIe registrů .....	69
Tabulka 6. Sumarizace použitého softwaru (zelená – mnou vytvořeno, oranžová – ostatní) .....	71

# 1 ÚVOD

Velké množství aplikací dnes potřebuje sledovat rychlé děje a generuje tak velké množství dat. Procesory již nejsou schopny tato data v reálném čase zpracovat, a jsou proto používána programovatelná hradlová pole známá jako FPGA (Field Programmable Gate Array). Tyto obvody nabízejí možnost konfigurace na míru dané aplikaci a jsou alternativou k drahým zákaznickým obvodům ASIC (Application Specific Integrated Circuit).

Tato práce se zabývá získáním dat z AD (Analog/Digital) převodníku přes vysokorychlostní rozhraní JESD204B a jejich následným uložením do paměti počítače. Cílovou platformou je PCIe karta Cecilie s převodníkovým FMC+ modulem, vyvinuté firmou DFC Design s.r.o. Karta obsahuje FPGA a nabízí možnost přenosu dat do nadřazeného systému přes rozhraní PCIe (Peripheral Component Interconnect Express), nebo 100G Ethernet.

Cílem je tedy do FPGA implementovat za účelem získávání dat z AD převodníku protokol rozhraní JESD204B a navrhnout a vytvořit moduly pro jejich přenos do PC. K nastavení převodníku bude použito rozhraní ADI-SPI, ovládané softcore procesorem Microblaze. Ten bude sloužit také pro monitorování stavu karty a konfiguraci dalších periférií. Přijatá data budou vhodně upravena a odeslána přes rozhraní PCIe nebo 100G Ethernet do PC. FPGA již obsahuje MAC (Media Access Control) vrstvu spolu s fyzickou vrstvou v podobě integrovaného bloku zvaného CMAC (C – římská číslice 100 – MAC). Stejně tak je již v FPGA obsažen i integrovaný blok pro PCIe. Použitým AD převodníkem je AD9208 od Analog Devices. Tento převodník je schopen vzorkovat s maximální rychlostí 3,1GSPS (GigaSamples Per Second) při 14bitovém rozlišení a s FPGA může být spojen až osmi 16Gbps linkami.

Teoretická část práce se věnuje především představení použitých technologií. V první kapitole jsou popsány základní parametry a chyby AD převodníků. Následuje popis rozhraní JESD204B, kde jsou vysvětleny principy synchronizace a dosažení deterministické latence. Část práce se zabývá rozhraním Ethernet a PCIe. Kapitola věnovaná Ethernetu se zaměřuje na popis protokolů pro přenos dat. Sběrnice PCIe je zde popsána v rozsahu potřebném pro pochopení jeho základních principů. V praktické části práce bude nejprve představen použitý hardware, kterým je, jak je výše zmíněno, PCIe karta Cecilie s převodníkovou FMC+ kartou. Poté bude proveden rozbor problému a návrh řešení. Samotné řešení se skládá ze dvou částí, a to FPGA designu spolu s potřebným firmwarem a softwaru, kterým jsou věnovány jednotlivé kapitoly. Následuje ověření integrity dat přenášených jednotlivými rozhraními (JESD204B, PCI Express a 100G Ethernet) a měření, jehož cílem je ověřit funkčnost převodníkové karty. Převodníková karta

byla netestovaným prototypem, a z toho důvodu se na ni nacházely chyby v zapojení. Konkrétně se jednalo o špatně zapojené převodníky napěťových úrovní. Chyby byly díky vhodně navrženému testovacímu designu brzy identifikovány a opraveny. Na závěr jsou diskutovány výsledky práce a je provedena úvaha nad zvoleným řešením a možnostmi dalšího vývoje.

## 2 AD PŘEVODNÍKY

V přírodě má většina fyzikálních jevů spojitý charakter a jsou obecně označovány jako spojité či analogové. Analogové signály je sice možné i analogově zpracovat, avšak takové řešení přináší chyby a neumožňuje je uložit a posléze s nimi pracovat. Proto se převádí do digitální podoby, ve které už nejsou dále degradovány a je mnohem jednodušší je dále zpracovávat. Tento přístup však přináší chybu samotného převodu. Signály mohou být spojité nebo diskrétní v čase i amplitudě, přičemž:

- Spojité v čase mohou měnit svou hodnotu v každém okamžiku.
- Spojité v amplitudě mohou nabývat všech hodnot.
- Diskrétní v čase mohou měnit svou hodnotu pouze v definovaných okamžicích (vzorkovaný signál).
- Diskrétní v amplitudě mohou nabývat pouze konečného počtu hodnot (kvantovaný signál).

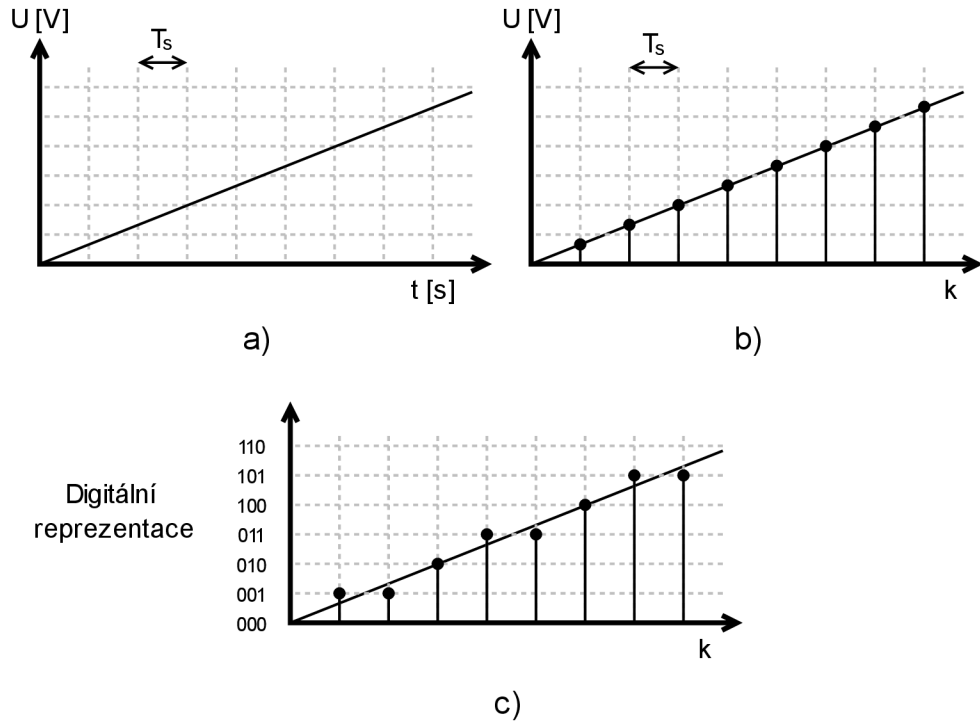
V počítačové technice je používán digitální signál, který je diskrétní jak v čase, tak v amplitudě. Může tedy nabývat pouze omezeného počtu hodnot, které se mohou měnit jen v určitých časových okamžicích.

AD převod je rozdělen na dvě části, kterými jsou vzorkování signálu a kvantování signálu. Vzorkování je proces diskretizace v čase, při kterém dochází k odběru vzorku vstupního signálu v definovaných okamžicích. Ty jsou dány vzorkovacími impulzy s periodou vzorkování  $T_s$ . Kvantování je diskretizace hodnoty signálu s kvantizačním krokem, který je dán referenčním napětím a rozlišením převodníku. Ukázka analogového, vzorkovaného a digitálního signálu je na obrázku 1.

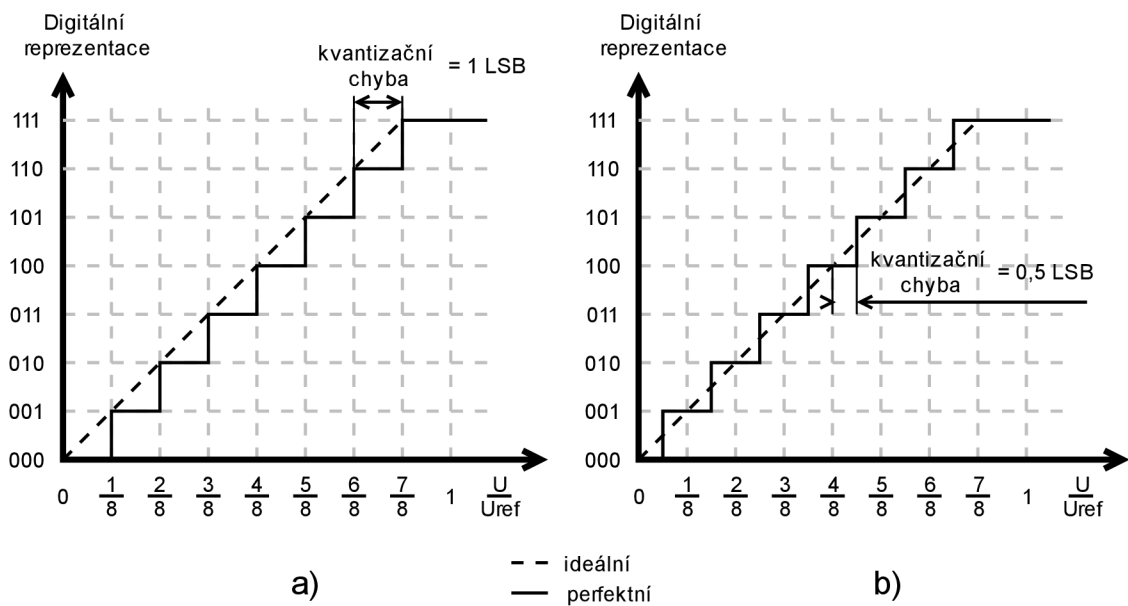
### 2.1 Parametry AD převodníku

Základními parametry AD převodníku jsou rozlišení, referenční napětí a perioda vzorkování. Rozlišení se udává v bitech a vyjadřuje počet převodníkem rozlišitelných úrovní. Počet bitů převodníku bývá označován jako  $N$ , rozlišitelných úrovní je pak  $2^N$ . Referenční napětí je porovnáváno s měřeným napětím a určuje rozsah AD převodníku. Spolu s rozlišením definuje velikost kvantizačního kroku, často označovaného jako LSB (Least Significant Bit). Polovina kvantizačního kroku je v závislosti na převodní charakteristice zpravidla nejnižší rozlišitelnou úrovní. Kvantování spolu přináší kvantizační chybu, která je stejně jako kvantizační krok závislá na použité převodní charakteristice. Na obrázku 2 je možné vidět dvě charakteristiky generující rozdílné kvantizační chyby. Charakteristika na obrázku b) vykazuje menší chybu [1], a proto je v dnešních AD převodnících

používaná. Ideální charakteristika odpovídá převodníku s nekonečným rozlišením, zatímco perfektní charakteristika odpovídá reálnému, chybami neovlivněnému převodníku.



Obrázek 1. Spojitý, diskrétní a digitální signál

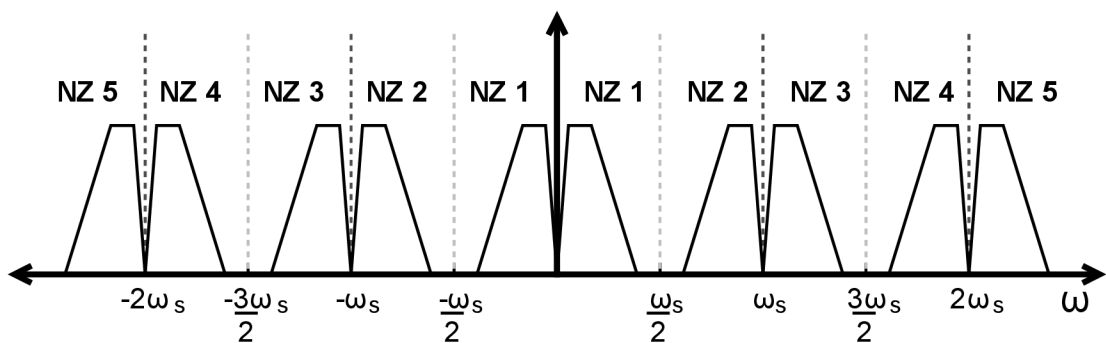


Obrázek 2. Kvantování signálu

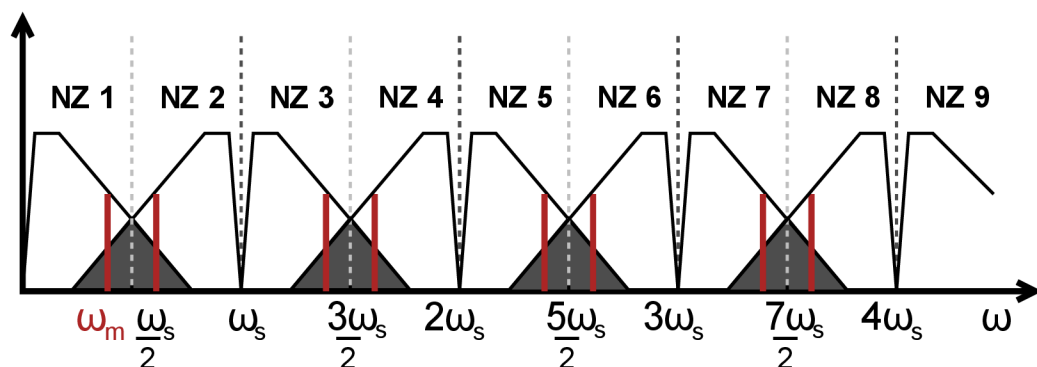


Perioda vzorkování určuje rychlost převodníku a měla by splňovat vzorkovací teorém. Ten vychází z Shannon-Kotělnikova teorému, který říká, že zpětná rekonstrukce spojitého signálu z jeho vzorků je možná pouze tehdy, je-li vzorkovací frekvence minimálně dvakrát vyšší než nejvyšší frekvence obsažená v signálu [2]. V případě nedodržení této podmínky vzniká tzv. aliasing efekt.

Spektrum signálu je ve frekvenční oblasti periodizováno s frekvencí vzorkování  $f_s$  a zrcadí se kolem její poloviny. Vznikají tzv. Nyquistovy zóny o šířce poloviny vzorkovací frekvence, které jsou ilustrovány obrázkem 3. Aliasing efekt vzniká překrytím původního a zrcadleného spektra. V časové oblasti to pak znamená, že není možné rozlišit například sinusový signál o frekvenci  $f_0$  a signál o frekvenci  $f_0 \pm kf_s$ , kde  $k$  je reálné celé číslo. Tohoto jevu lze využít při měření periodických signálů s frekvencí vyšší než polovina vzorkovací frekvence. Podmínkou je, že frekvenční spektrum signálu musí spadat do jediné Nyquistovy zóny. Šířka frekvenčního pásma měřeného signálu tedy může být maximálně polovina vzorkovací frekvence [2]. Aliasing efekt je demonstrován obrázkem 4.



Obrázek 3. Frekvenční spektru periodického signálu



Obrázek 4. Frekvenční spektrum a aliasing

Dalšími důležitými parametry charakterizující kvalitu AD převodu jsou odstup signálu od šumu (SNR = Signal to Noise Ratio) a celkové harmonické zkreslení (THD = Total Harmonic Distortion). SNR je definován jako poměr signálu a šumu na výstupu AD převodníku [1]. Může být vypočítán jak z efektivní hodnoty signálu, tak z jeho výkonu podle *rovnice 1*. Jeho hodnota se zpravidla udává v decibelech a může být vyjádřena *rovnicí 2*. SNR může být zlepšeno odfiltrováním vstupního šumu nebo použitím zařízení ve stálém prostředí.

$$SNR_P = \frac{P_{(signál)}}{P_{(šum)}} = SNR_V^2 = \left( \frac{U_{RMS}(signál)}{U_{RMS}(šum)} \right)^2 \quad (1)$$

$$SNR_P(dB) = 10 \log(SNR_P) = 20 \log(SNR_V) \quad (2)$$

Při průchodu signálu nelineárním zařízením dochází k zesílení jeho harmonických frekvencí a vznikají tak frekvenční složky, které se v původním signálu nevyskytovaly. Tento efekt může být vyjádřen parametrem THD a vypočten podle *rovnice 3*. Je definován jako poměr součtu harmonických složek a základní složky signálu [1]. Jeho hodnota roste spolu s maximální frekvencí vstupního signálu.

$$THD = \frac{\sqrt{U_2^2 + U_3^2 + \dots + U_n^2}}{U_1} \quad (3)$$

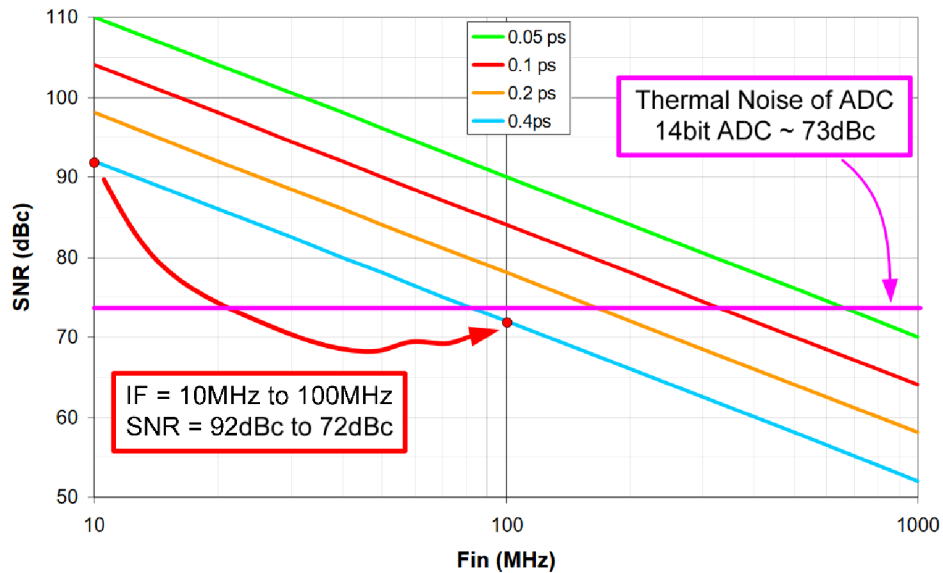
Kombinací SNR a THD vznikne parametr zvaný SINAD (Signal to Noise and Distortion), který je definován jako poměr efektivní hodnoty signálu a všech ostatních spektrálních složek [1]. Může být vypočten pomocí *rovnice 4*.

$$SINAD = -10 \log \left( 10^{-\frac{SNR}{10}} + 10^{-\frac{THD}{10}} \right) \quad (4)$$

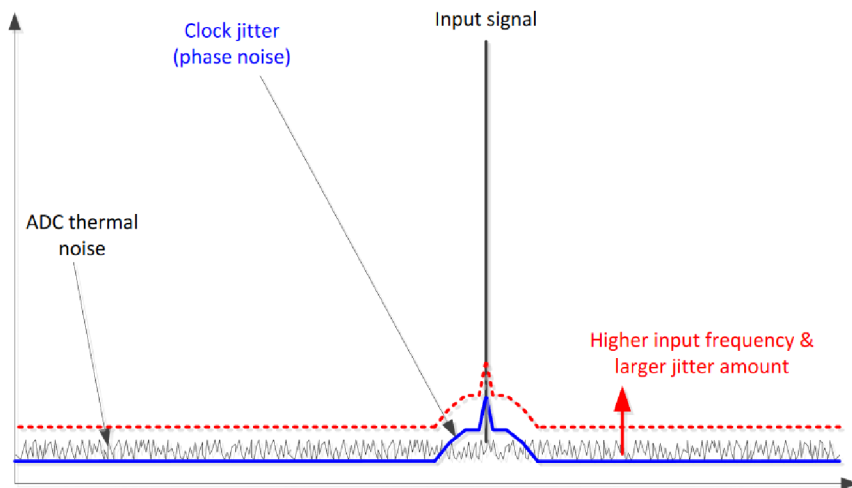
Posledním často používaným parametrem je ENOB (Effective Number Of Bits). Ideálně se rovná počtu bitů převodníku, ale ve skutečnosti bývá o něco nižší. Bere v úvahu chyby způsobené šumem a ostatními harmonickými složkami a vyjadřuje, kolik nejvyšších bitů převodníku má reálnou vypovídající hodnotu [1]. ENOB je možné vypočítat *rovnicí 5*.

$$ENOB = \frac{SINAD - 1,76}{6,02} \quad (5)$$

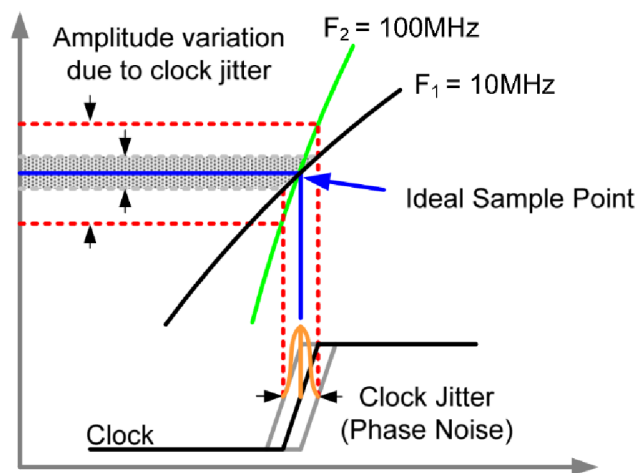
Na výše popsané parametry má vliv jitter vzorkovacího hodinového signálu, který je definován jako odchylka od jeho periodicity. Způsobuje chyby vedoucí ke zvýšení úrovně šumu ve frekvenčním spektru a zhoršení SNR, což je ilustrováno obrázky 5 a 6. Obrázek 7 demonstruje, jak chyba převodu způsobená jitterem roste spolu s frekvencí měřeného signálu [3].



Obrázek 5. Změna SNR při rostoucí frekvenci signálu a konstantním jitteru [3]



Obrázek 6. Změna frekvenčního spektra signálu vlivem jitteru [3]



Obrázek 7. Vliv jitteru na vzorkování signálu [3]

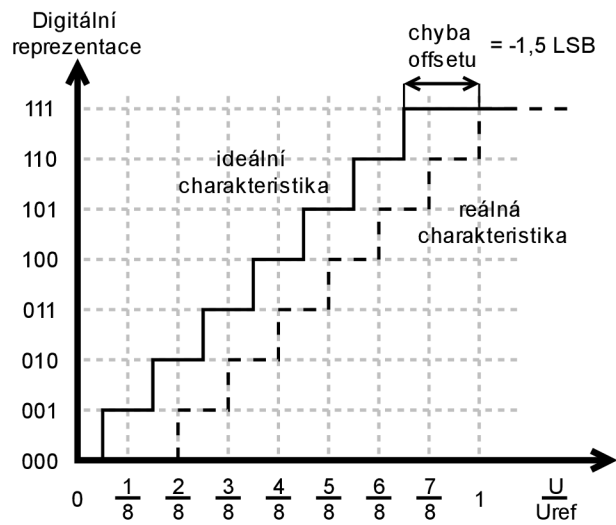
## 2.2 Chyby

Převod spojitého signálu na digitální provází chyby ovlivňující výslednou přesnost AD převodníku. Nejčastěji se jedná o chybu offsetu, chybu zisku a nelinearity. Často se chyby dělí na aditivní a multiplikační, přičemž aditivní chyby jsou stejné pro celý rozsah, zatímco multiplikační rostou spolu s hodnotou měřeného signálu.

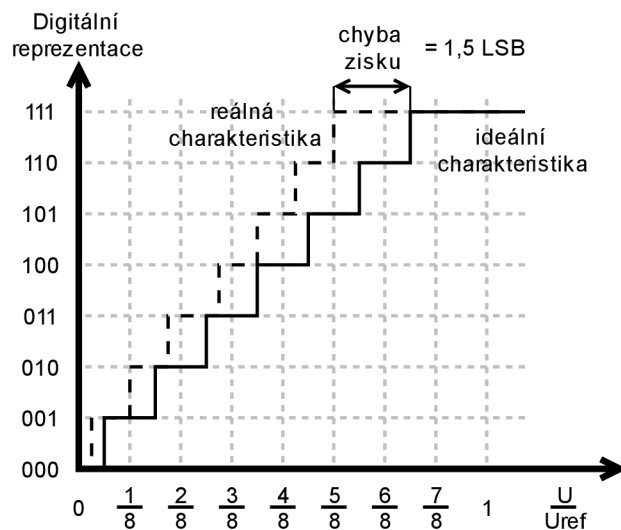
Chyba offsetu je definována jako rozdíl ideální charakteristiky převodníku a její reálné podoby v počátku [1]. Pokud reálná charakteristika leží pod ideální, jedná se o offset záporný, v opačném případě o offset kladný. Tato chyba může být změřena a kompenzována buď při převodu, nebo při následném zpracování dat. Jedná se o chybu aditivní a její grafické znázornění je na obrázku 8.

Chyba zisku je na rozdíl od chyby offsetu multiplikační, a tudíž roste spolu s hodnotou měřeného signálu. Je definována jako rozdíl hodnoty posledního kroku převodníku od jeho ideální charakteristiky po kompenzaci chyby offsetu [1]. Pokud se skutečná převodní charakteristika nachází nad ideální, jedná se o kladnou chybu a naopak. Chyba zisku je znázorněna na obrázku 9.

Velmi často je uváděna chyba z celého rozsahu, která říká, jak moc se na konci rozsahu liší skutečná převodní charakteristika od ideální. Při jejím výpočtu je brána v úvahu jak chyba z offsetu, tak chyba zisku.

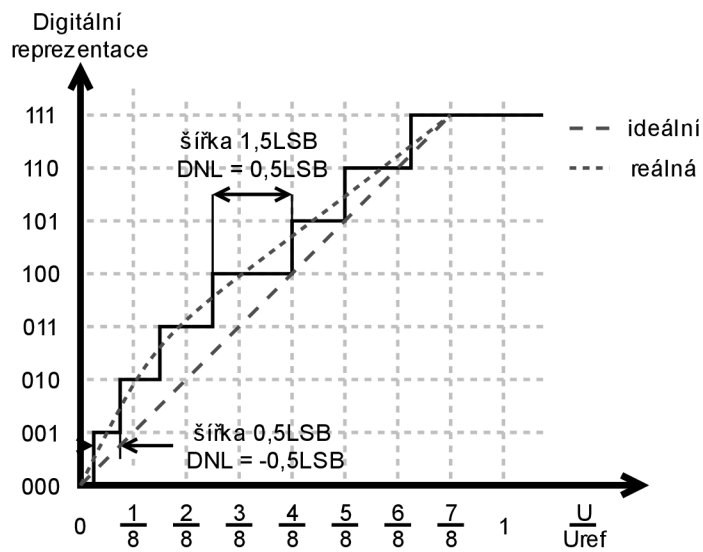


Obrázek 8. Chyba offsetu

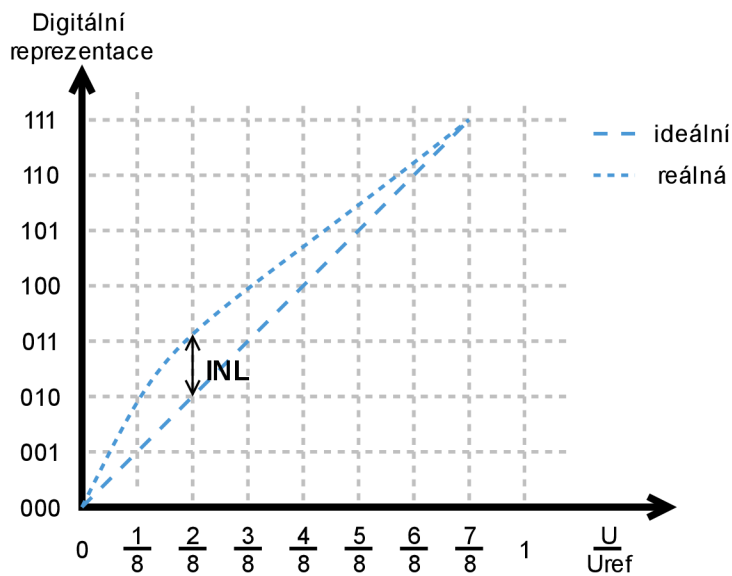


Obrázek 9. Chyba zisku

Nelinearita značně ovlivňuje kvalitu převodu analogové veličiny na digitální. I přes kompenzované chyby offsetu a zisku nebude skutečná převodní charakteristika totožná s ideální. Počáteční a koncové body budou stejné, avšak průběh bude lehce odlišný. To je dáno nepřesnostmi kvantovacích kroků, kdy některé mohou být delší a některé zase kratší. Existují dva základní typy nelinearity, a to diferenciální nelinearita (DNL) a integrální nelinearita (INL) [1]. Ukázky obou typů jsou zobrazeny na obrázcích 10 a 11.



Obrázek 10. Nelinearita převodní charakteristiky (DNL)



Obrázek 11. Nelinearita převodní charakteristiky (INL)

Diferenciální nelinearita je dána nejmenší a největší odchylkou skutečné šířky kvantizačního kroku od jeho ideální šířky. Integrovaná nelinearita je definována jako maximální vertikální rozdíl mezi skutečnou a ideální charakteristikou. Značí velikost odlišnosti obou charakteristik a může být interpretována jako suma odchylek všech kvantizačních kroků od jejich ideální šířky. Širší kvantizační kroky posouvají skutečnou převodní charakteristiku pod ideální, zatímco kratší ji analogicky posouvají nahoru.

## 2.3 Zlepšení parametrů převodu

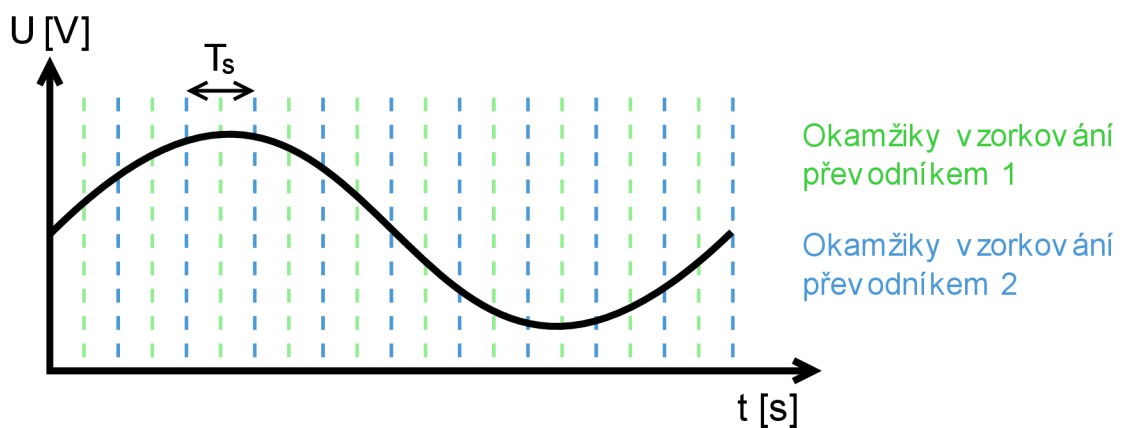
Jsou známy metody, jež umožňují změnit určité parametry převodu k lepšímu, avšak většinou za cenu zhoršení jiných parametrů. Některé z těchto metod zde budou popsány.

### 2.3.1 Oversampling

Oversampling je proces vzorkování analogového signálu s mnohem vyšší frekvencí, než jaká je dána vzorkovacím teorémem [1]. Tato technika přináší několik výhod, přičemž první je vyhnutí se problémům s aliasing efektem. Dalšími výhodami je vylepšení SNR a zvýšení rozlišení převodníku. Toho je na úkor výsledné vzorkovací frekvence dosaženo průměrováním většího množství vzorků. Požadované množství vzorků pro získání  $n$  bitů navíc je  $2^{2n}$ .

### 2.3.2 Prokládání

V aplikacích, ve kterých je zapotřebí vysoké vzorkovací frekvence je možné využít procesu prokládání [2]. Jedná se o využití více kanálů nebo samotných převodníků pro vzorkování jednoho analogového signálu. Princip spočívá ve vzorkování více převodníky, které mají vzájemně vzorkovací hodinový signál posunutý o půl periody (v případě dvou převodníků). Tím je získáno dvojnásobné množství vzorků a výsledná vzorkovací frekvence je tak zvýšena na dvojnásobek. Ukázka principu je na obrázku 12.



Obrázek 12. Prokládání AD převodníků

### 2.3.3 Průměrování

Tato technika počítá s použitím většího množství převodníků pro vzorkování jednoho analogového signálu [2]. Do převodníků je přiveden vzorkovací hodinový signál o stejné frekvenci a fázi a všechny tedy vzorkují současně. Tento přístup zlepšuje SNR o 3 dB a redukuje nekoreluující šum. Nekoreluující šum má náhodný charakter a většinou také nulovou střední hodnotu. Vyšší počet převodníků představuje větší zlepšení SNR. Rovnice 6 definuje výpočet výsledné hodnoty SNR při použití N převodníků.

$$SNR_{AVE} = -10 \cdot \log \left[ \frac{10^{\frac{-SNR_{ADC1}}{10}} + 10^{\frac{-SNR_{ADC2}}{10}} + \dots + 10^{\frac{-SNR_{ADCN}}{10}}}{N^2} \right] \quad (6)$$

Pokud je SNR všech použitých převodníků stejné, může být rovnice 6 zjednodušena na rovnici 7.

$$SNR_{AVE} = SNR_{ADC} + 10 \cdot \log N \quad (7)$$

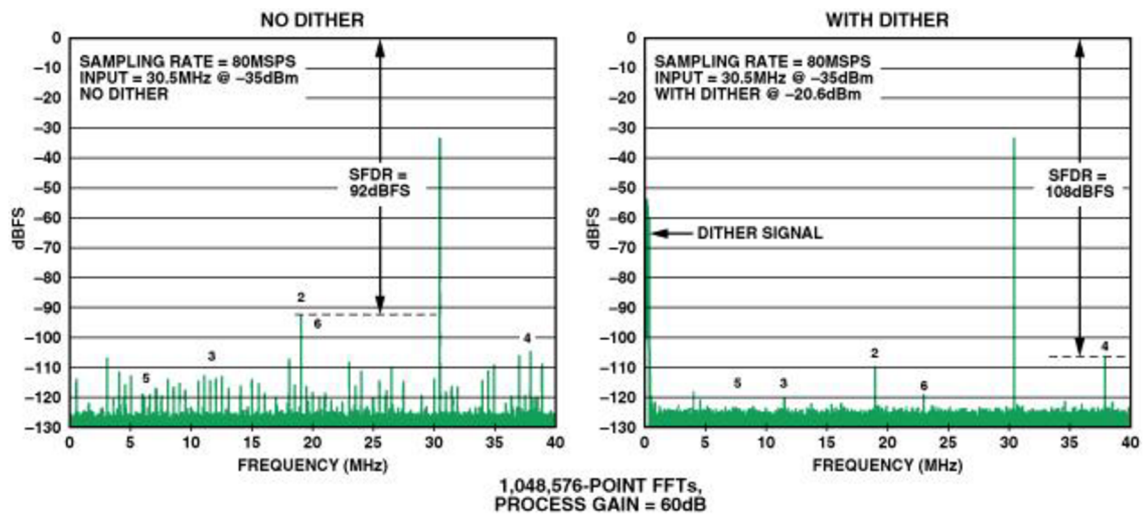
Je zapotřebí brát v potaz také jitter vzorkovacího hodinového signálu. Jitter představuje odchylku periodického signálu od jeho periodicity a jeho výsledná hodnota může být spočítána rovnicí 8.

$$Jitter_{Total}^2 = Jitter_{Ext\_clk}^2 + \left[ \frac{Jitter_{Aperture}}{\sqrt{N}} \right]^2 \quad (8)$$

### 2.3.4 Dithering

AD převodníky vykazují deterministické a systematické chyby, které se opakují při určitých úrovních signálu. Tyto chyby mohou být minimalizovány přidáním nízkourovňového náhodného šumu do měřeného signálu [2]. Může tak být redukována velikost harmonických složek, avšak za cenu zvýšení šumu na nízkých frekvencích. Ukázka frekvenčního spektra signálu bez a s použitím ditheringu je na obrázku 13.



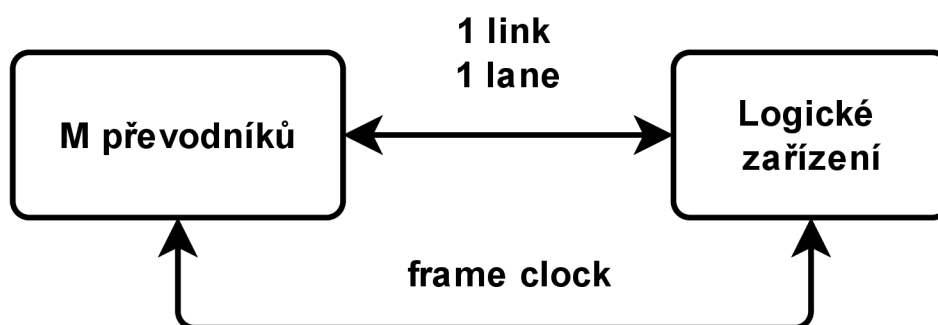


Obrázek 13. Frekvenční spektrum bez a s použitím ditheringu [4]

### 3 STANDARD JESD204

JESD204 je standard sériového rozhraní určeného pro komunikaci mezi AD nebo DA převodníky a nadřazeným systémem, kterým je nejčastěji FPGA nebo ASIC. Jeho první verze byla vydána v roce 2006 skupinou JEDEC, která jej od té doby udržuje a pravidelně přichází s novými revizemi, jimiž se snaží udržet krok se zvyšujícími se nároky na AD převodníky. Doposud vyšly tři revize původního standardu, a to JESD204A v roce 2008, JESD204B v roce 2011 a JESD204C v roce 2017. V současné době nejpoužívanějším rozhraním pro komunikaci s převodníky je JESD204B, které bude použito i v této práci.

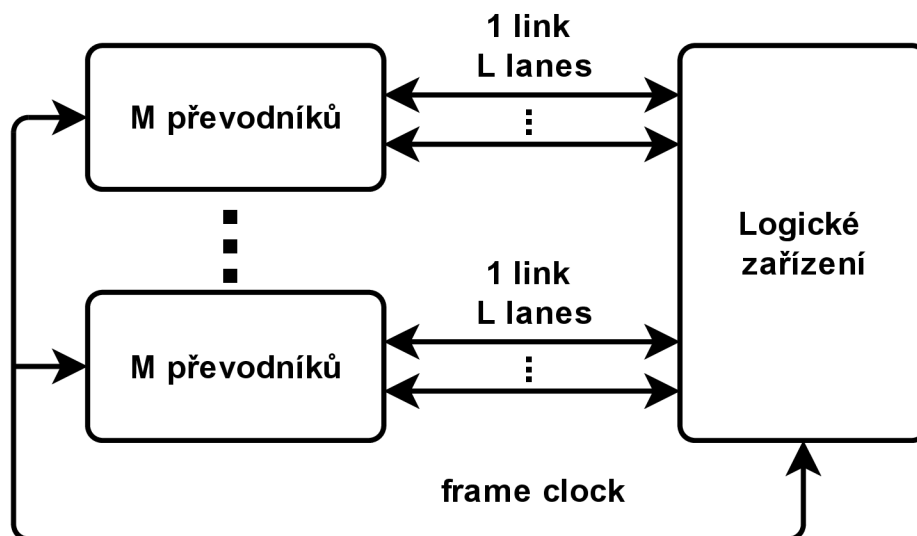
Původní standard definoval jednu linku mezi převodníkem a logickým zařízením tvořenou diferencíálním párem. Přenosová rychlost byla stanovena na 312,5 Mbps až 3,125 Gbps podle potřeby a k oběma zařízením byl přiveden společný referenční hodinový signál [5]. Tato přenosová rychlost ovšem brzy přestala být dostačující a byla proto vydána první revize. Topologie první verze JESD204 je na obrázku 14.



Obrázek 14. Topologie rozhraní JESD204

#### 3.1 Revize JESD204A

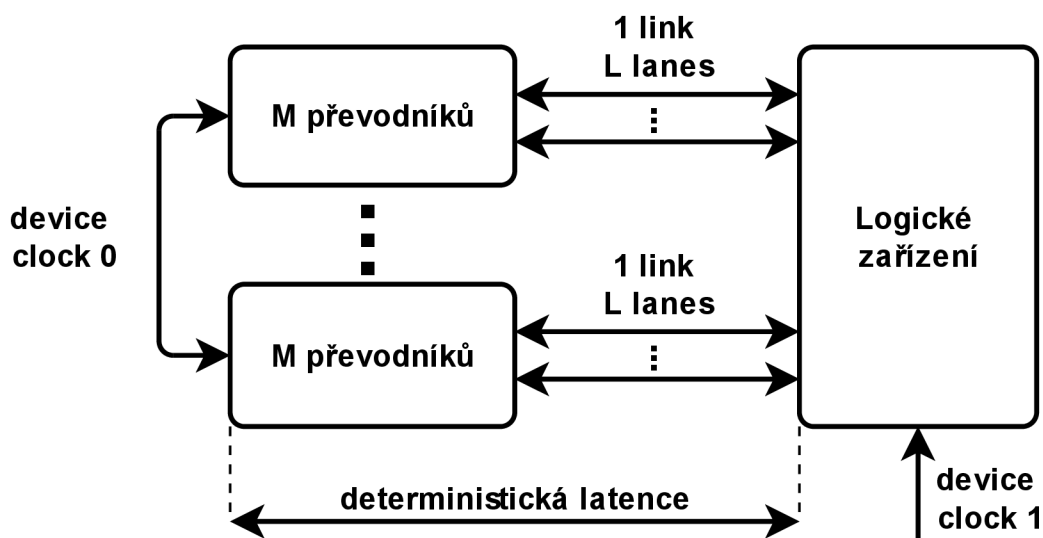
JESD204A umožnil použít pro komunikaci mezi převodníkem a logickým zařízením více linek při zachování stejné přenosové rychlosti. Mohlo tak být přeneseno větší množství dat než s JESD204 použitím většího počtu linek [5]. Přestože se zvýšila schopnost přenést větší množství dat, chyběla možnost deterministické latence mezi okamžikem navzorkování a okamžikem přijetí dat příjemcem. Mechanismus pro zajištění deterministické latence byl spolu s dalším navýšením přenosové rychlosti představen v následující revizi B. Topologie první revize JESD204 je na obrázku 15.



Obrázek 15. Topologie rozhraní JESD204A

### 3.2 Revize JESD204B

Pro další popis JESD204B bylo vycházeno z jeho standardu [6]. Revize JESD204B nabízí oproti předchozí revizi navýšení maximální přenosové rychlosti a mechanismus pro zajištění deterministické latence mezi okamžikem navzorkování signálu převodníkem a přijetí jeho digitální reprezentace příjemcem. Maximální rychlost je stanovena na 12,5 Gbps na jednu linku, avšak objevují se i zařízení s vyšší rychlostí. Další změnou je odpadající nutnost použít jeden společný referenční hodinový signál a každé zařízení tak může mít vlastní. Topologie druhé revize JESD204 je na obrázku 16.



Obrázek 16. Topologie rozhraní JESD204B

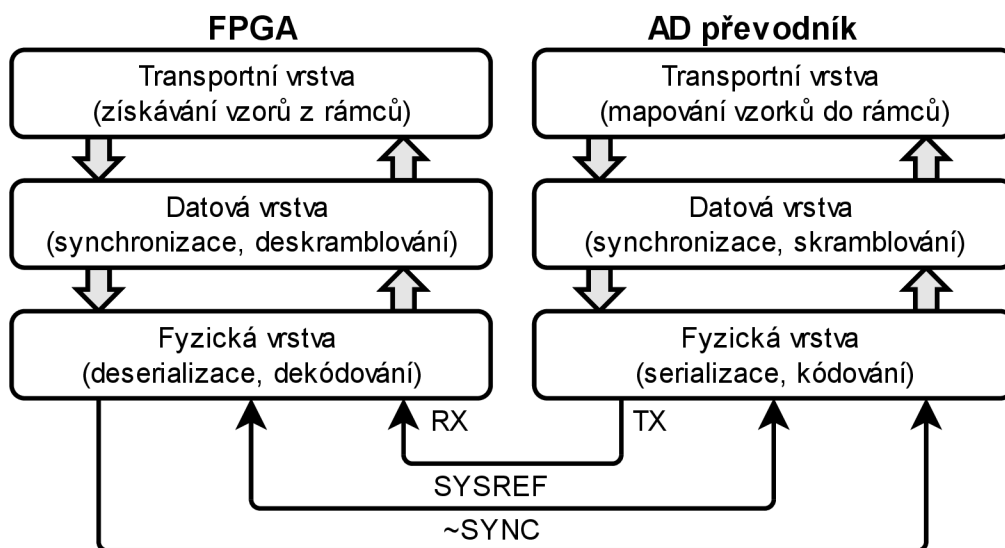
Deterministické latence může být dosaženo dvěma způsoby, a to pomocí signálu SYNC~ nebo SYSREF. JESD204B je tak rozděleno na tři podtřídy.

- Podtřída 0 neposkytuje mechanismus pro dosažení deterministické latence
- Podtřída 1 pro dosažení deterministické latence využívá signál SYSREF
- Podtřída 2 pro dosažení deterministické latence využívá signál SYNC

Více k jednotlivým mechanismům je uvedeno v kapitole Deterministická latence. Standard rozděluje JESD204B do čtyř vrstev:

1. Fyzická – definuje, jak jsou fyzicky přenášena data mezi vysílačem a přijímačem
2. Datová – kóduje paralelní data a má na starosti synchronizační a zarovnávací procesy.
3. Transportní – přidává k datům kontrolní bity a následně je rozděluje na skupiny po 4 bitech.
4. Aplikační – určená pro speciální případy, kdy dochází k atypickému mapování dat.

Typicky jsou použity pouze první tři vrstvy, jež jsou ilustrovány obrázkem 17. JESD204B není v současnosti nejnovější revizí. V roce 2017 byla představena revize C, o které je více v kapitole *Revize JESD204C*.

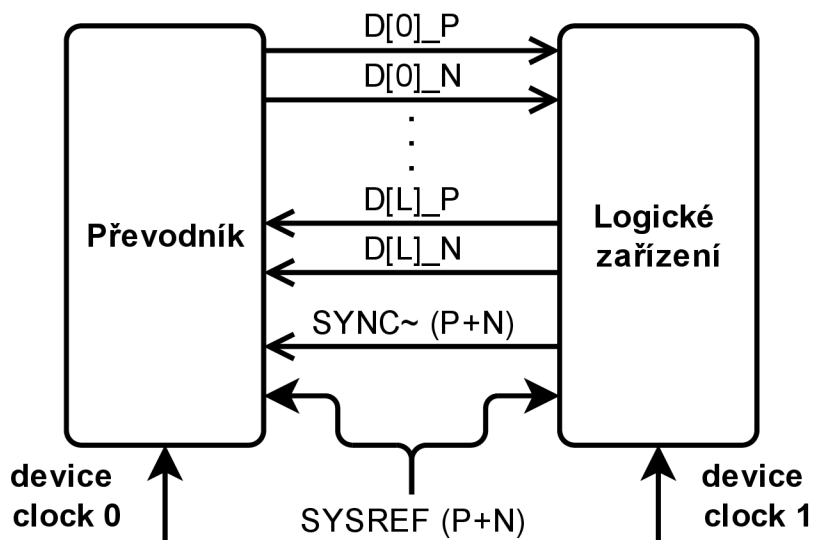


Obrázek 17. Vrstvy JESD204B

### 3.2.1 Fyzická vrstva

Fyzická vrstva definuje, jak jsou přenášena data mezi vysílačem a příjemcem. JESD204B používá pro přenos dat jednosměrné diferenciální páry pracující ve standardu CML (Current Mode Logic). Každé zařízení musí mít připojen referenční

hodinový signál označený jako *device clock* a signál *SYNC~*. Zařízení pracující v podtřídě 1 pak musejí mít připojen také signál *SYSREF*, o kterém lze nalézt více v kapitole *Deterministická latence*. Všechny signály potřebné pro implementaci rozhraní JESD204B jsou ilustrovány na obrázku 18.



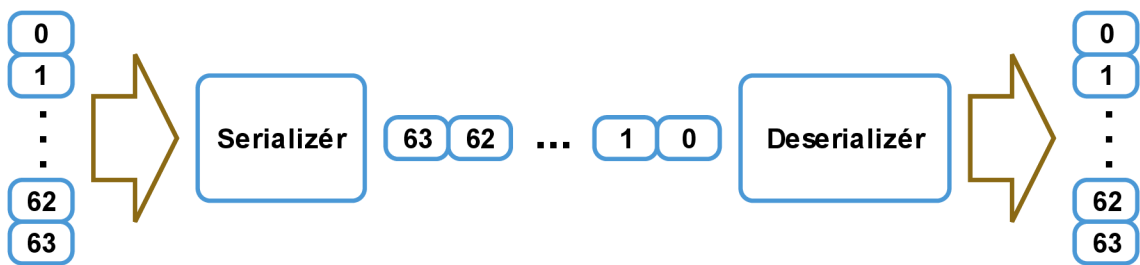
Obrázek 18. Signály použité v JESD204B

Revize JESD204B nabízí navýšení maximální přenosové rychlosti na jednu linku a definuje tři rychlostí třídy:

- 312,5 Mbps až 3,125 Gbps – kompatibilní s předchozími revizemi
- 3,125 Gbps až 6,375 Gbps
- 6,375 Gbps až 12,5 Gbps

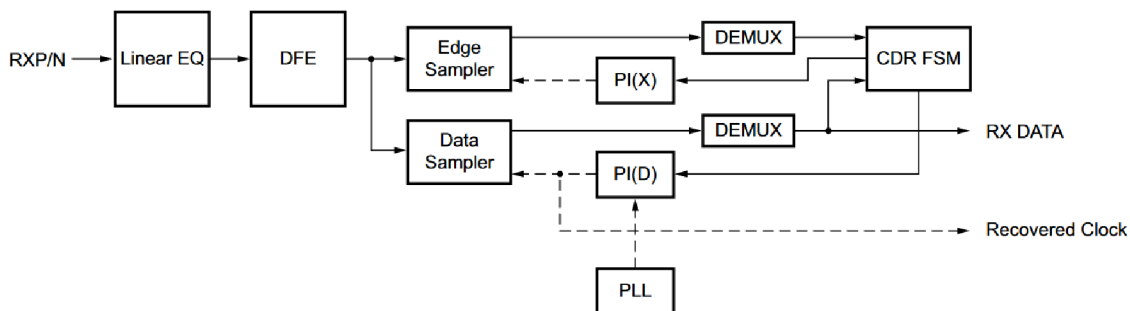
Maximální doporučená rychlost se odvíjí od fyzického provedení. Rozhraní je určeno pro propojení převodníku a logického zařízení spojem o délce kolem 200 mm. Pro rychlosti nad 3,125 Gbps by měl být v cestě maximálně jeden konektor a pro rychlosti pod 3,125 Gbps pak maximálně dva konektory.

Vysílače i přijímače obsahují logické obvody známé jako serializéry a deserializéry. Serializéry převádí vstupní paralelní data o šířce  $n$  generované s frekvencí  $f$  do sériové podoby s frekvencí  $n \cdot f$ . Deserializéry naopak provádí převod sériových dat na paralelní. Demonstrace procesu serializace a deserializace je na obrázku 19. V této práci bude implementován přijímač a proto se bude dále zbývat pouze jeho částí.



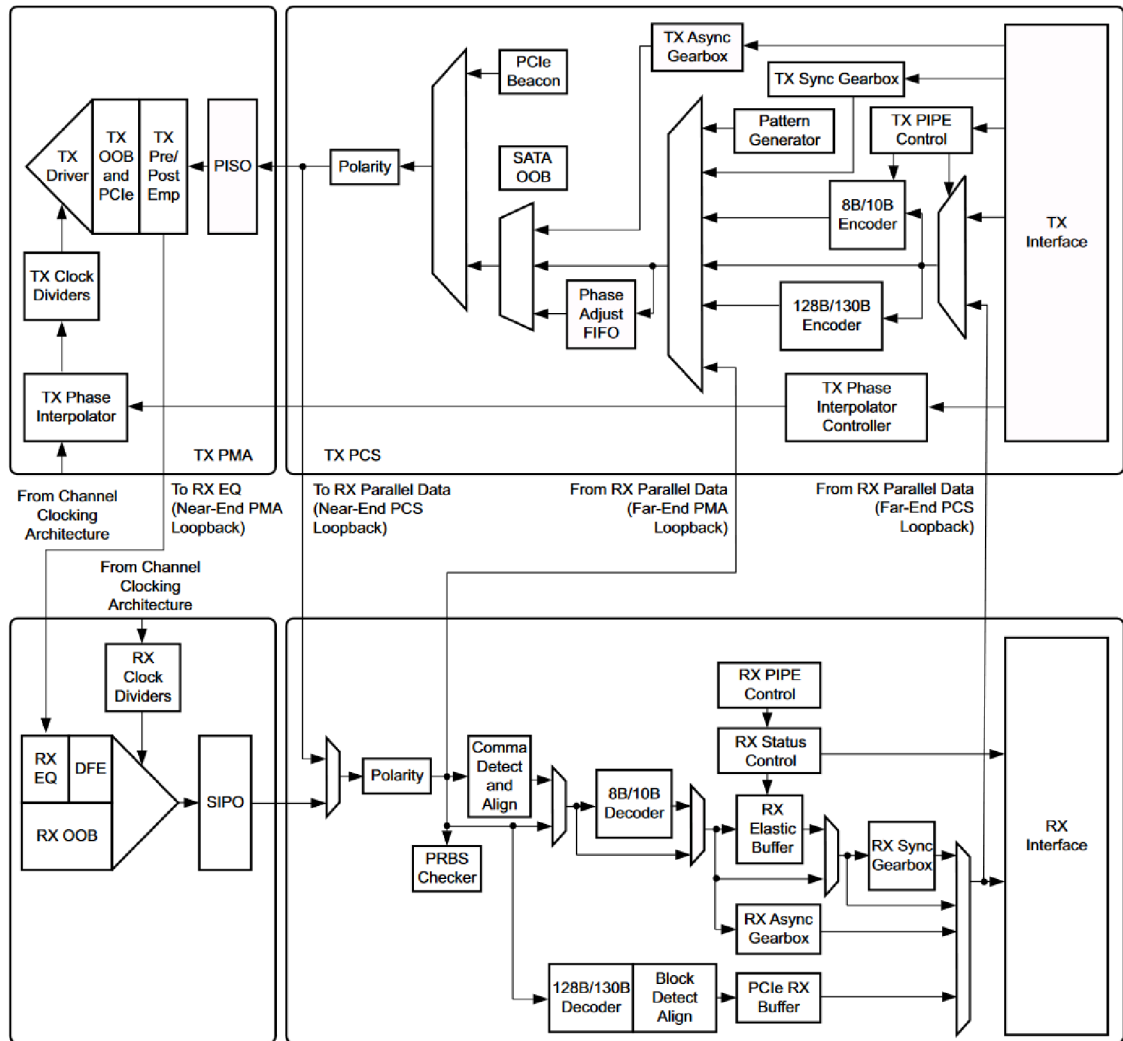
Obrázek 19. Serializér a deserializér

V FPGA jsou serializéry a deserializéry integrovány v tzv. multigigabitových transceiverech [7]. Ty obsahují další důležité prvky, jako například bloky CDR (Clock Data Recovery), kódování a dekódování a PLL (Phase Locked Loop) pro generování vysokých frekvencí. Kromě toho také umožňují nastavení parametrů výstupního signálu. Transceivery na straně přijímače se starají o deserializaci a následné dekódování dat (v případě JESD204B dekódování 10b/8b). Blokové schéma GTY transceiverů z FPGA od společnosti Xilinx je na obrázku 21. Přestože jsou kódovací a dekódovací bloky již obsaženy v transceiverech, jsou z hlediska rozdělení vrstev standardu JESD204B uvažovány až ve vrstvě datové.

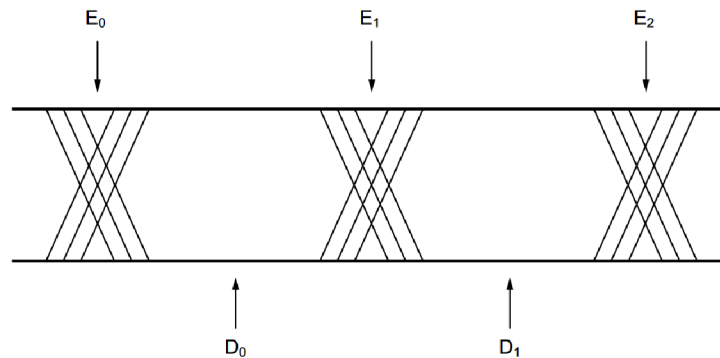


Obrázek 20. CDR blok přijímací části GTY transceiverů [7]

U paralelních rozhraní je zapotřebí vést spolu s daty také hodinový signál, ke kterému jsou data vztažena. To je při rychlostech v řádech gigabitů nemožné a využívá se proto procesu zvaného CDR, který umožňuje při dostatečném množství přechodů hodinový signál rekonstruovat z přenášených dat. Blokové schéma CDR bloku v přijímací části transceiverů je na obrázku 20. Princip spočívá v použití dvou vzorkovačů, přičemž jeden vzorkuje ve středu mezi dvěma hranami signálu a druhý na hranách signálu, jako je demonstrováno obrázkem 22. Vzorkovaná data jsou následně zavedena do stavového automatu, který vyhodnocuje fázi přijatého signálu a upravuje fázi vzorkovacích hodin [7].



Obrázek 21. Blokové schéma GTY transceiverů [7]



Obrázek 22. Místa vzorkování signálu v bloku CDR [7]

### 3.2.2 Datová vrstva

Datová vrstva standardu JESD204B pokračuje tam, kde fyzická vrstva končí. Vstupují do ní deserializovaná a stále ještě kódovaná data (přestože jsou dekódována v transceiveru, je z pohledu standardu dekódování až v datové vrstvě). Kódování dat má více důvodů a je velmi užitečné i navzdory faktu, že snižuje efektivní využití linky. V předchozí kapitole byl zmíněn proces zvaný CDR, který pro správnou funkčnost potřebuje v datovém streamu dostatečné množství přechodů. Kódování 8b/10b je jedním ze způsobů, jak tohoto dosáhnout, neboť zaručuje 3 až 8 přechodů na symbol [8]. Kromě toho se touto technikou dosáhne stejnosměrného vyvážení linky a předejde se tak problémům s offsetem. Poslední výhodou je možnost přenášet spolu s daty kontrolní znaky (symboly) sloužící k zarovnání dat (deserializovaná data tak nejsou bitově posunuta) a monitorování linky. JESD204B těchto kontrolních znaků využívá v inicializační fázi přenosu a posléze ke kontrole zarovnání linek. Kontrola probíhá nahrazováním dat při splnění určitých podmínek a jejich detekcí v přijímači. Kontrolními znaky použitými při synchronizaci a během přenosu dat jsou:

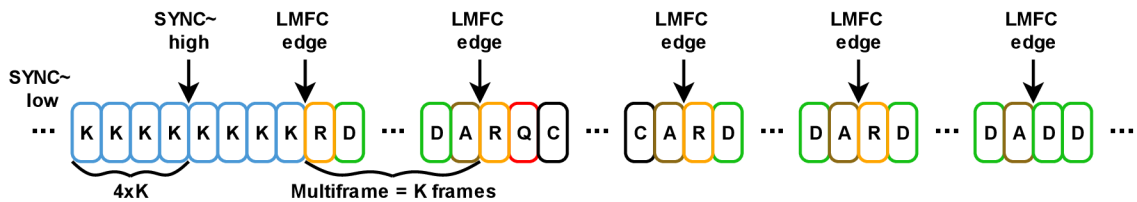
- /K/ = /K28.5/ - generovaný vysílačem během CGS fáze.
- /R/ = /K28.0/ - během ILAS fáze označuje začátek multirámce.
- /Q/ = /K28.4/ - během ILAS fáze oznamuje že následují konfigurační data.
- /A/ = /K28.3/ - při splnění určitých podmínek a během ILAS fáze nahrazuje poslední znak multirámce a slouží ke kontrole zarovnání LMFC.
- /F/ = /K28.7/ - při splnění určitých podmínek nahrazuje poslední znak rámce a slouží ke kontrole zarovnání linky.

Data jsou formována do rámců a multirámců. Multirámce jsou složeny ze skupiny rámců, přičemž každý rámec obsahuje vždy minimálně jeden celý vzorek. Hranice multirámců jsou dány multirámcovým hodinovým signálem (LMFC – Local Multiframe Clock), který bývá nejčastěji odvozen od referenčního hodinového signálu v každém zařízení. Před započítím přenosu dat proběhnou dvě inicializační fáze sloužící k synchronizaci vysílače, přijímače a jednotlivých linek. Těmito fázemi jsou CGS (Code Group Synchronization) a ILAS (Initial Lane Alignment Sequence) a jsou demonstrovány obrázkem 23.

Během CGS fáze si přijímač vyžádá synchronizaci uvedením signálu SYNC~ do stavu LOW a vysílač na všech linkách začne generovat symbol /K/. Poté co přijímač přijme na každé lince minimálně 4x /K/, uvede signál SYNC~ zpět do stavu HIGH a ukončí tak CGS fázi.

Následně začne spolu s nástupnou hranou LMFC fáze ILAS dlouhá minimálně čtyři multirámce. První, třetí a čtvrtý multirámec začínají symbolem /R/ a končí symbolem /A/. Druhý multirámec se od ostatních odlišuje tím, že obsahuje parametry spojení a přijímač tak může ověřit, že je nastaven správně.

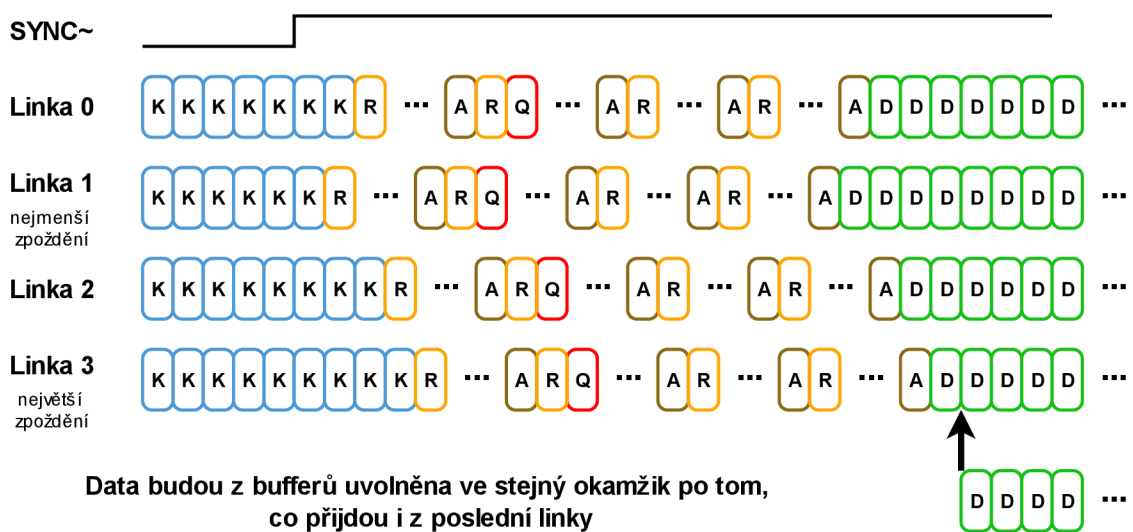




Obrázek 23. Synchronizační fáze JESD204B

Po skončení ILAS začne spolu s hranou LMFC odesílání samotných dat. Poslední znaky (oktety) v rámci a multirámci mohou být při splnění určitých podmínek nahrazeny symboly /F/ nebo /A/ a slouží ke kontrole zarovnání linky. Data mohou být také volitelně skramblována polynomem  $1 + x^{14} + x^{15}$ .

Přijímač obsahuje takzvané elastické buffery, do kterých jsou data po přijetí ukládána. Jejich účelem je vyrovnat rozdílné zpoždění mezi jednotlivými linkami a data jsou z nich uvolněna současně. Obrázek 24 demonstruje zarovnání linek a současné uvolnění dat.



Obrázek 24. Zarovnání dat z více linek

### 3.2.3 Transportní vrstva

Úkolem transportní vrstvy je na straně vysílače mapovat vzorky do rámců a na straně přijímače je z rámců získat zpět. Protože náplní této práce je získat data z AD převodníku, bude zde uveden popis z pohledu přijímače. Transportní vrstva získává od datové vrstvy paralelní data, ve kterých identifikuje rámce. Rámec může

obsahovat jeden a více vzorků z jednoho nebo několika převodníků. Může být také rozdělen mezi vícero linek pro dosažení potřebné přenosové rychlosti. Ta je dána rychlostí a rozlišením převodníku, rychlostí transceiverů a strukturou rámců.

Kromě vzorků mohou být přenášeny také kontrolní a „tail“ bity. Ty jsou zařazeny buď za každým vzorkem nebo na konci rámcu. Konfigurace JESD204B spoje bývá zpravidla popsána následujícími parametry.

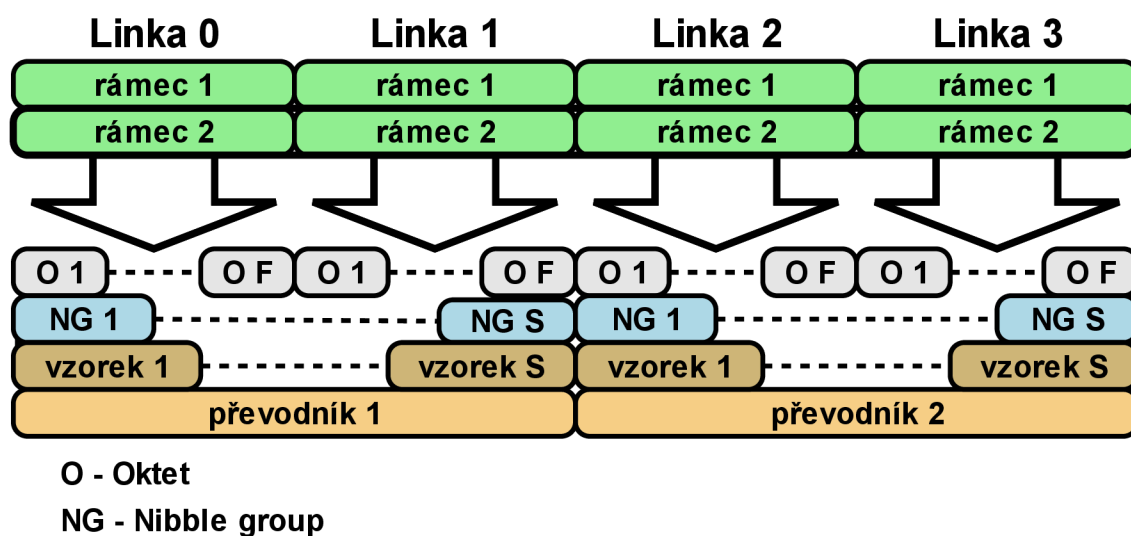
- rozlišení převodníku (N)
- počet použitých bitů na vzorek (N')
- počet převodníků (M)
- počet vzorků z jednoho konvertoru za rámeček (S)
- počet linek (L)
- počet oktětů v rámci (F)
- počet rámců v multirámcu (K)
- počet kontrolních bitů za každým vzorkem (CS)
- počet tail bitů za každým vzorkem (TT)

Počet oktětů v rámci může být spočítán pomocí následujících rovnic 9 a 10.

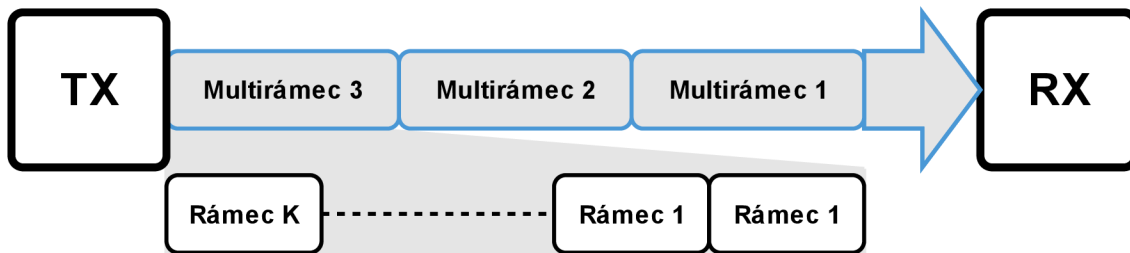
$$N' = N + CS + TT \quad (9)$$

$$F = \frac{M \cdot S \cdot N'}{8 \cdot L} \quad (10)$$

Počet bytů na rámeček není pevně dán specifikací, ale liší se od aplikace k aplikaci podle použitého převodníku a nastavením jeho parametrů. Struktura dat a separace vzorků je znázorněna na obrázcích 25 a 26.



Obrázek 25. Mapování vzorků v rámcích



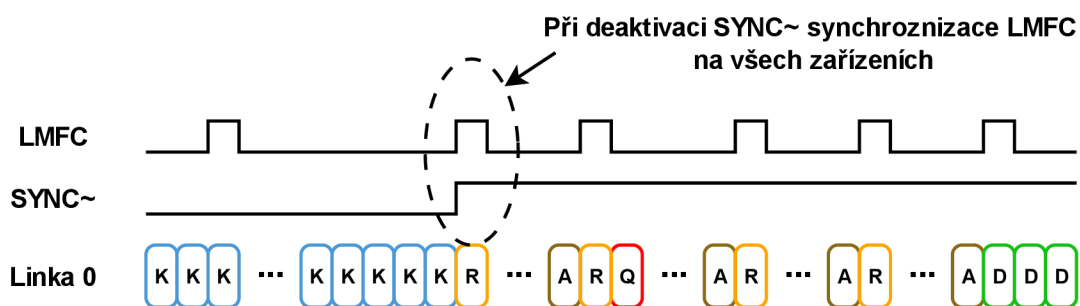
Obrázek 26. Rámce v multirámcích

### 3.2.4 Deterministická latence

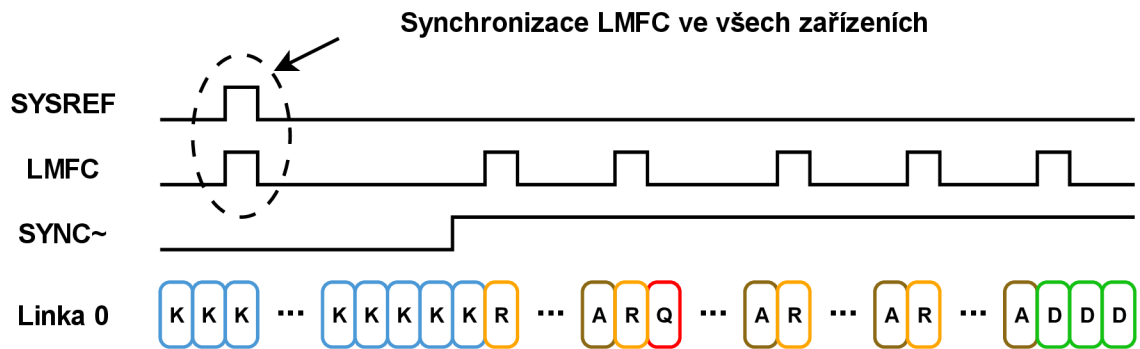
Některé aplikace potřebují, aby byla doba, za kterou se data dostanou od vstupu do převodníků na výstup přijímače vždy konstantní a známá. Tento koncept je znám jako deterministická latence a revize JESD204B představila mechanismus k jeho dosažení.

Deterministické latence je dosaženo ukládáním přijatých dat do elastického bufferu a jejich vypuštěním pouze v určitý časový okamžik. Tento okamžik je znám jako příležitost k uvolnění a je definován jako počet rámcových hodinových cyklů od hrany LMFC. Pokud jsou v tento okamžik ve všech bufferech platná data, mohou být vypuštěna. V případě že, tato podmínka splněna není, je nutné vyčkat jednu periodu LMFC na další příležitost.

Podtřídy 1 a 2 představují dva rozdílné mechanismy pro synchronizaci LMFC. Zařízení pracující v podtřídě 1 provedou synchronizaci svých LMFC ve chvíli, kdy přejde signál SYNC~ do stavu HIGH. Podtřída 2 naopak využívá signálu SYSREF. Při nalezení jeho nástupné hrany všechna zařízení provedou synchronizaci svých LMFC. Oba popsané principy jsou zobrazeny na obrázcích 27 a 28.



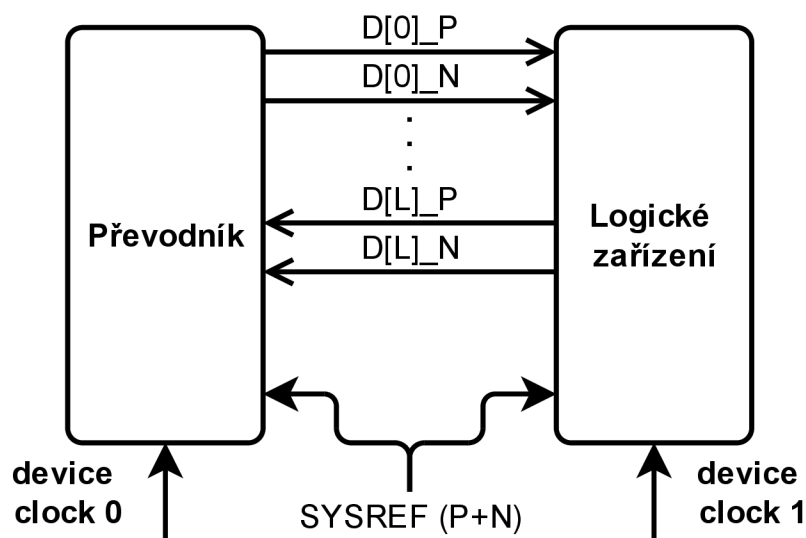
Obrázek 27. Synchronizace LMFC podtřída 1



Obrázek 28. Synchronizace LMFC podtřída 2

### 3.3 Revize JESD204C

Poslední revize standardu není v práci použita, avšak i přesto zde budou uvedeny hlavní rozdíly oproti jeho předchůdci. Nejvýraznější změnou je zvýšení maximální přenosové rychlosti z 12,5 Gbps na 32 Gbps. Spolu se změnou rychlosti dochází také ke změně kódování na efektivnější 64b/66b, kterému by se tedy mělo říkat spíše pseudokódování. Data totiž nejsou fakticky kódovaná, ale jsou k nim pouze přidány 2 bity. Z toho důvodu se stává povinným skramblování, neboť je zde jediným mechanismem zaručujícím stejnou vyváženost linky a dostatečné množství přechodů pro CDR. JESD204C dovoluje použít techniku nazývanou FEC (Forward Error Correction), která umožňuje opravu chyb v datech bez nutnosti jejich opětovného přeposlání. Signály potřebné pro implementaci rozhraní JESD204C jsou ilustrovány na obrázku 29.



Obrázek 29. Topologie rozhraní JESD204C

Velká změna nastala také v principu synchronizace zařízení a deterministické latence. Zařízení stále používají signály SYSREF a device clock, avšak zaniká signál SYNC a s ním i CGS a ILA fáze. Synchronizace zde probíhá pomocí dvou extra bitů na konci každého 64bitového slova. Z těchto dvou bitů je postupně složeno 32bitové synchronizační slovo. Zanikly také rámce a multirámce a byly nově definovány bloky, multibloky a rozšířené multibloky [9].

## 4 ROZHRANÍ 100G ETHERNET

Ethernet je dnes nejrozšířenějším komunikačním rozhraním ve všech možných typech počítačových systémů. Jeho 100gigabitová verze se oproti předchůdcům liší nejnižšími vrstvami, umožňujícími takto vysoké přenosové rychlosti. Kromě vysokých rychlostí dochází k vývoji procesů, jež zaručují vyšší míru bezchybnosti a spolehlivosti tohoto rozhraní. V této práci bude Ethernet, spolu s komunikačním protokolem RoCE (RDMA over Converged Ethernet), použit pro přenos zpracovaných dat z AD převodníku do PC nebo serveru. V krátkosti zde budou popsány protokoly vedoucí k implementaci RDMA (Remote Direct Memory Access) a použití RoCE.

### 4.1 UDP/IP protokol

Jednou z velkých výhod ethernetu je velké množství vyvinutých protokolů, jež umožňují jeho použití v širokém spektru aplikací. V této práci budou z obecně známých implementovány protokoly IP (Internet Protocol) a UDP (User Datagram Protocol).

IP protokol umožňuje směřovat pakety z jednoho zařízení do druhého i přesto, že zařízení nejsou zapojena ve stejné síti [10]. Je tak možné přenést paket mezi místy, jež jsou od sebe vzdálené tisíce kilometrů.

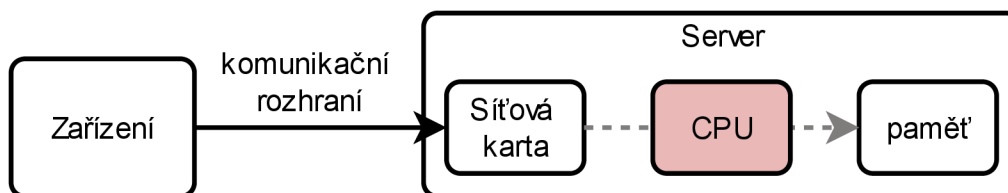
UDP protokol je pak nadstavbou IP protokolu, který umožňuje přenos dat mezi zdrojovým a cílovým UDP portem [11]. Negarantuje spolehlivé doručení paketu, neboť nepodává odesilateli žádnou zpětnou vazbu o úspěšném či neúspěšném přijetí. Je používán především v aplikacích, ve kterých není ztráta jednoho paketu problém nebo není možné transakci z různých důvodů opakovat, jako například stream dat (video, data z AD převodníku atd.).

### 4.2 RDMA

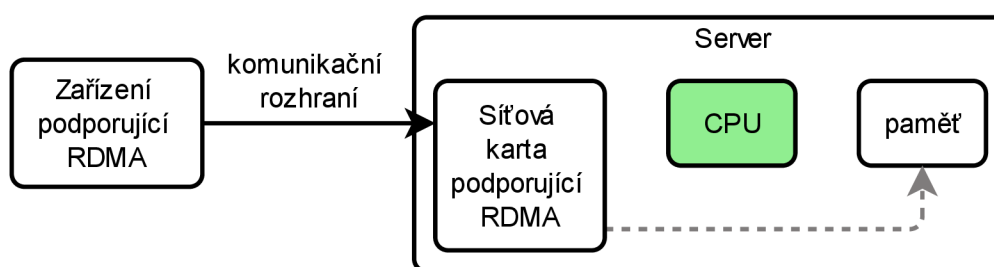
Přímý přístup do paměti, zkráceně DMA (Direct Memory Access), umožňuje zápis nebo čtení z nebo do paměti počítače bez zbytečného zatěžování procesoru. Princip je takový, že procesor určí kam, odkud a kolik dat chce přesunout, DMA řadič operaci vykoná a následně procesoru oznámí, že byl přenos dokončen.

Na podobném principu funguje také RDMA, avšak data již nejsou přenášena v rámci jednoho počítačového systému, ale mohou pocházet z externích zařízení [12]. Aby bylo možné RDMA použít, musí být tato funkce podporována oběma systémy. V této práci bude implementováno RDMA, a to použitím dále popsaného protokolu RoCE. Na straně PC nebo serveru bude RoCE podporováno síťovou

kartou Connect X5 od firmy Mellanox. Na obrázcích 30 a 31 je demonstrován rozdíl komunikace mezi zařízením a severem bez a s podporou RDMA.



Obrázek 30. Komunikace se serverem bez RDMA



Obrázek 31. Komunikace se serverem s RDMA

### 4.3 RoCE

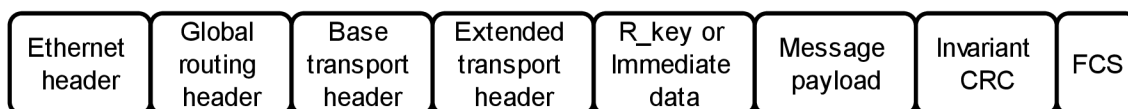
Základem RoCE je protokol komunikačního rozhraní Infiniband [13], jež nabízí několik typů přenosu dat s využitím RDMA. Tyto typy se dělí na spojově orientované a neorientované a dále na spolehlivé a nespolehlivé. Infiniband implementuje čtyři spodní vrstvy OSI (Open System Interconnection) modelu a umožňuje tak směrování paketů v síti. Složení paketu je možné vidět na obrázku 32. Jeho základní struktura se nápadně podobá struktuře IP paketu, neboť Local Routing Header (LRH) a Global Routing Header (GRH) jsou ekvivalenty Ethernetové a IP hlavičky. Další hlavičky v paketu pak určují typ přenosu a jeho parametry.



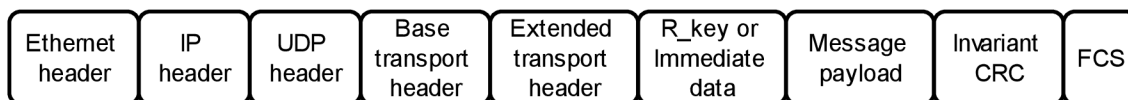
Obrázek 32. Struktura Infiniband paketu

Cílem RoCE je zachovat protokol a architekturu Infinibandu a zároveň využít rozšířenosti ethernetu. Existuje ve dvou verzích. První z nich nahrazuje LRH

Ethernetovou hlavičkou a dovoluje tak použití pouze v lokální Ethernetové síti. Struktura takového paketu je demonstrována obrázkem 33. Druhá zabaluje vyšší vrstvy Infiniband protokolu do UDP/IP protokolu a umožňuje směrování paketu i mimo lokální síť. Z paketu tak mizí i GRH, které je nahrazeno IP a UDP hlavičkami. Tato verze RoCE je nazývána RoCEv2 a jeho strukturu je možné vidět na obrázku 34.



Obrázek 33. Struktura RoCE paketu



Obrázek 34. Struktura RoCEv2 paketu

Pro detailnější popis toho, jak RDMA funguje, je zapotřebí definovat několik pojmů. Pojmy se týkají hardwarové implementace RDMA a nezaměřují se, stejně jako tato práce, na softwarovou část.

**SQ** (Send Queue) je fronta obsahující požadavky čekající na odeslání dat. Nachází se v hardwaru a bývá zpravidla plněna softwarem, jenž požadavky na transakce generuje.

**RQ** (Receive Queue) je fronta obsahující požadavky na přijetí dat. Stejně jako SQ se nachází v hardwarové části s je plněna softwarem, který tyto požadavky generuje a zajišťuje vyhrazení paměti pro přicházející data.

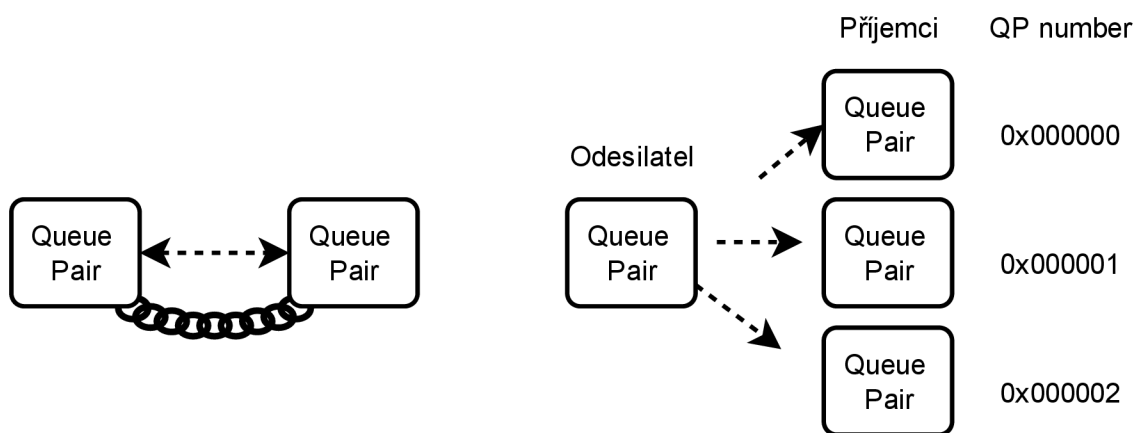
**CQ** (Completion Queue) je fronta obsahující zpracované požadavky, jež byly předtím umístěny v SQ nebo RQ. Jednotlivé položky jsou postupně odebírány softwarem pro signalizaci dokončení požadované transakce.

**QP** (Queue Pair) je pak sdružení SQ a RQ, které bývá označováno číslem nazývaným QP number. V cílové destinaci paketu může být vytvořených více QP, a proto je jeho účelem specifikovat, který je pro danou transakci cílový.

Jak již bylo zmíněno dříve, Infiniband nabízí několik typů přenosů. V případě spolehlivých přenosů příjemce dat jejich obdržení vždy potvrdí odesláním tzv. ACK (Acknowledge) a při jeho neobdržení nebo obdržení NACK (Not Acknowledge) přenos zopakuje. U nespolehlivých přenosů k tomuto nedochází a není tak zaručeno doručení odeslaných dat. Přenosy mohou být dále spojově orientované nebo neorientované. V případě spojově orientovaných je před započítáním přenosu dat nejprve nutno navázat spojení s protistranou. Dojde tak ke spárování dvou QP, které mohou přenášet data pouze mezi sebou. U spojově



neorientovaných přenosů není třeba navazovat spojení a data mohou být zaslána jakémukoliv QP. Oba typy jsou ilustrovány na obrázku 35.



a) Spojově orientovaný přenos

b) Spojově neorientovaný přenos

Obrázek 35. Infiniband – spojově orientované/neorientované přenosy

V této práci bude použito nejjednoduššího, spojově neorientovaného a nespolehlivého typu přenosu. Ten se nazývá UD (Unreliable Datagram) a bude zde nyní detailněji popsán.

Jak již bylo ilustrováno výše, RoCE paket obsahuje Ethernetovou, IP a UDP hlavičku, následovanou Infinibandí BTH (Base Transport Header) a dalšími hlavičkami danými jeho typem. V případě UD bude BTH (obrázek 36) následována DETH (Datagram Extended Transport Header). Po DETH (obrázek 37) následují data, a nakonec CRC počítané od IP hlavičky až po poslední byte dat.



Obrázek 36. Infiniband Base Transport Header

Jednotlivá pole BTH definují:

- OpCode – interpretaci následujících hlaviček a nákladu
- SE (Solicited Event) – poslední paket transakce
- PAD – počet padding bytů na konci paketu (náklad je poslán v násobcích 4)
- Tver (Transport header Version) – verzi BTH = 0x0
- Partition Key – oddíl, jehož je cílový QP součástí
- Destination QP – cílový QP (QP number)
- A (Acknowledge) – žádost o zaslání potvrzovacího paketu

- PSN (Packet Sequence Number) – sekvenční číslo paketu



**Obrázek 37. Infiniband Datagram Extended Transport Header**

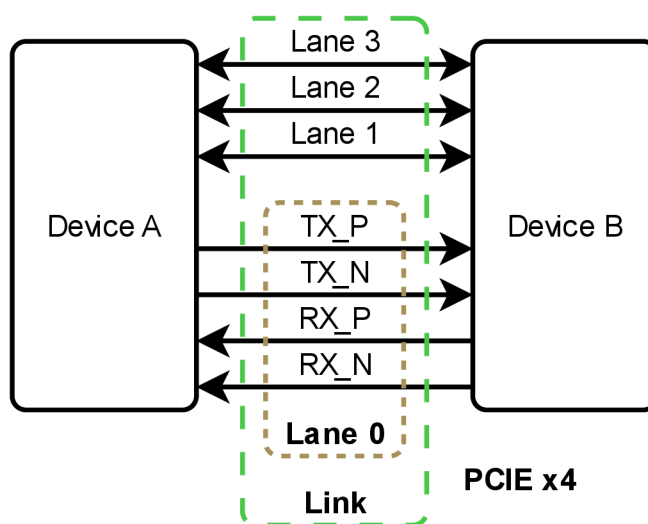
Queue Key je klíč kontrolovaný příjemcem paketu, sloužící k umožnění přístupu k cílovému QP. K identifikace odesilatele paketu slouží pole Source QP obsahující jeho QP number.

## 5 SBĚRNICE PCI EXPRESS

PCIe vychází z dřívější sběrnice PCI. Ta přenášela data v paralelním uspořádání a obsahovala kromě datových také množství dalších řídicích a statusových signálů. Velké množství signálů zvyšovalo složitost DPS (Deska Plošných Spojů). Neméně tomu přispíval fakt, že paralelní uspořádání sběrnice vyžadovalo, aby byly z důvodu minimalizace zpoždění mezi jednotlivými signály všechny spoje stejné délky. Velký důraz byl také kladen na rozvod hodinového signálu, který byl pro všechna zařízení společný. Nevyhovující maximální rychlost sběrnice a složitost návrhu hardwaru byly motivací k vytvoření nového standardu PCI Express [14] [15].

PCI Express přešel z paralelního přenosu dat na sériový. Řídicí a statusové signály, které byly v PCI vedeny samostatnými vodiči, jsou v PCIe zabudovány v protokolu a přenášeny zprávami. To vede ke snížení počtu potřebných vodičů a tím i zjednodušení návrhu a provedení DPS. V průběhu let byly představeny další tři generace standardu, přičemž nejaktuálnější je PCIe Gen 4. Jednotlivé generace jsou zpětně kompatibilní a liší se především maximální přenosovou rychlostí.

V tomto bodě je potřeba definovat pojmy lane a link, jejichž ilustrace jsou na obrázku 38. Lane je složený ze dvou diferenciálních párů, jeden pro směr od zařízení A k zařízení B a druhý opačně. Link je pak seskupení lanes tvořící spoj mezi dvěma zařízeními. Počet lanes v jednom linku může být 1, 2, 4, 8, 16 a 32. *Tabulka 1* ukazuje maximální přenosové rychlosti při použití různého počtu lanes.



Obrázek 38. PCI Express link a lane v konfiguraci x4

**Tabulka 1. Porovnání rychlostí jednotlivých generací PCIe**

	PCIe Gen 1	PCIe Gen 2	PCIe Gen 3	PCIe Gen 4
x1	2,5 Gbps	5 Gbps	8 Gbps	16 Gbps
x2	5 Gbps	10 Gbps	16 Gbps	32 Gbps
x4	10 Gbps	20 Gbps	32 Gbps	64 Gbps
x8	20 Gbps	40 Gbps	64 Gbps	128 Gbps
x16	40 Gbps	80 Gbps	128 Gbps	256 Gbps
X32	80 Gbps	160 Gbps	256 Gbps	512 Gbps

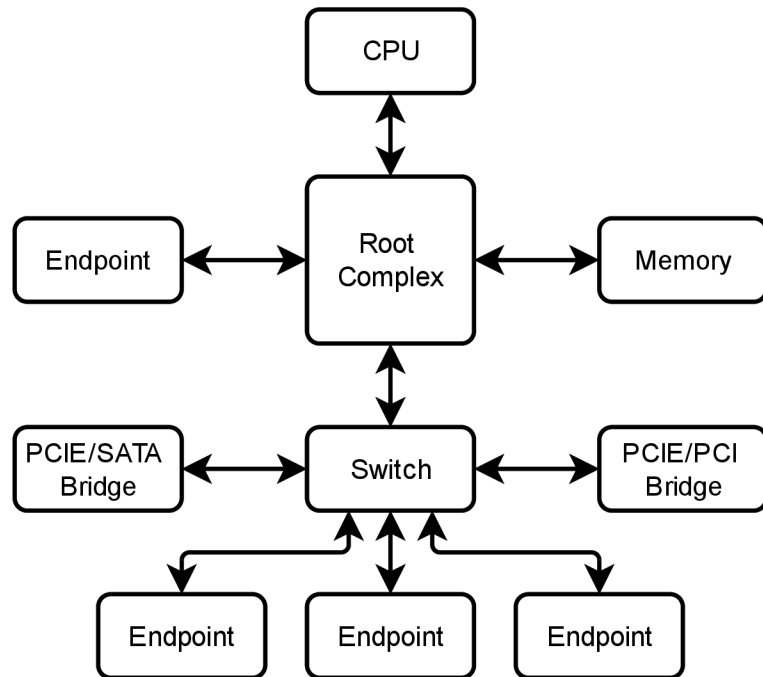
## 5.1 Topologie

PCI používal jednu sběrnici sdílenou vícero zařízeními. To se v případě PCIe změnilo a začalo se užívat architektury point-to-point. Dvě zařízení tak vždy spojuje jediný link, který není sdílený s žádným jiným zařízením. Topologie PCIe je demonstrována obrázkem 39. PCI Express definuje čtyři typy zařízení, kterými jsou:

- Root complex – propojuje CPU a paměť s PCIe sběrnici. S CPU je spojen lokální sběrnici a může v něm být i obsažen. Může obsahovat jeden nebo více PCIe portů.
- Switch – stejně jako v případě Ethernetu přijímá pakety a přeposílá je dále na správný port podle cílové adresy.
- Endpoint – je koncovým zařízením propojeným s root complexem buď přímo, nebo přes switch. Endpointem může být široká škála zařízení, jako například grafická karta, síťová karta, komunikační karta, měřicí karta a jiné.
- Bridge – propojuje PCIe s jiným typem sběrnice (PCIe, SATA a další)

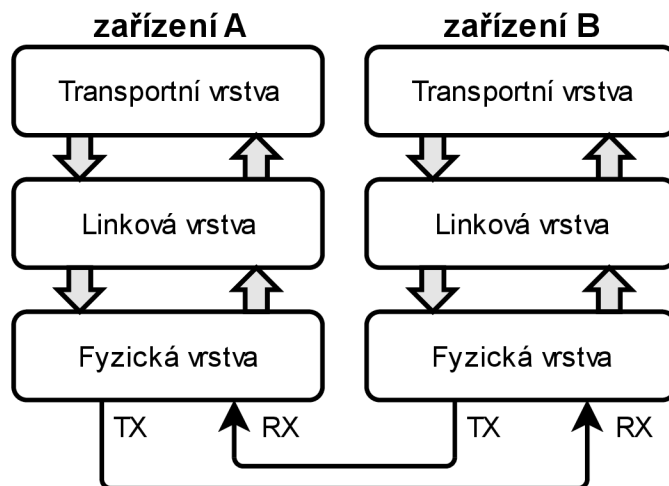
## 5.2 Vrstvy

Architektura PCIe se skládá ze tří na sebe navazujících vrstev. Tyto vrstvy jsou fyzická, linková a transakční. Každá z nich vykonává jinou funkci a umožňuje vyšší vrstvě fungovat správně. Ukázka propojení dvou zařízení a jejich vrstev je ilustrována na obrázku 40.

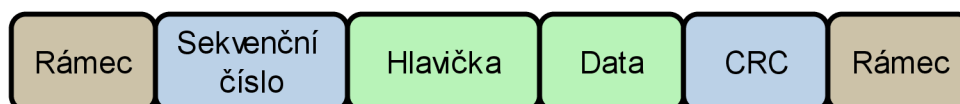


Obrázek 39. Topologie sběrnice PCI Express

Každá z vrstev se projeví při tvorbě paketu, neboť informace z vrstvy vyšší je vždy zabalena v informaci z vrstvy nižší. Co jednotlivé vrstvy dělají bude popsáno v následujících kapitolách, avšak pro přehled je zde obrázkem 41 ilustrováno složení PCIe paketu.



Obrázek 40. Propojení vrstev v PCIe



Transportní vrstva

Linková vrstva

Fyzická vrstva

Obrázek 41. Složení PCIe paketu

## 5.2.1 Fyzická

Jedná se o nejnižší vrstvu v modelu PCIe, jejíž primárním účelem je data fyzicky přenést mezi odesilatelem a příjemcem. Na straně odesilatele přejímá data z linkové vrstvy, volitelně je skrambluje, kóduje a přidává znaky označující začátek a konec rámce. Následně vše serializuje a odešle příjemci, který naopak provede deserializaci. Analogicky rozezná počátek rámce, data dekóduje, deskrambluje a předá vyšší vrstvě. Fyzická vrstva může být rozdělena do dvou bloků, logického a elektrického.

- Logický – přejímá data z vyšší vrstvy a připravuje je pro přenos. Provádí tedy skramblování, kódování 128b/130b (předchozí verze 8b/10b) a zabalení do rámce.
- Elektrický – převádí data z paralelní podoby do sériové a naopak. Na straně příjemce také provádí fázové zarovnání hodinového signálu.

## 5.2.2 Linková

Je prostřední vrstvou v modelu PCIe. Jejím úkolem je zajistit, že data přijdou k příjemci neporušená a ve správném pořadí. Toho je docíleno použitím CRC (Cyclic Redundancy Check) a sekvenčního čísla paketu, jež jsou při odesílání do paketu vloženy a při přijetí kontrolovány. Pokud není nalezena chyba, jsou data z paketu předána dále transakční vrstvě. V opačném případě pošle zpět linková vrstva zprávu o chybě a pokus je zopakován.

Jak bylo naznačeno, linková vrstva je schopná sama o sobě generovat pakety, označované jako DLLP (Data Link Layer Packet). Jejich účelem je řízení nižších funkcí, vztahujících se k danému linku. Těmito funkcemi jsou řízení toku dat, potvrzování transakcí a řízení napájení. Jsou generovány i konzumovány linkovou vrstvou.

### 5.2.3 Transakční

Je nejvyšší vrstvou v modelu PCIe. Účelem transakční vrstvy je pracovat s daty, tedy je odesílat a přijímat. Komunikuje s linkovou vrstvou, které předává data k odeslání a přejímá přijatá data. Veškerá přijatá data předaná transakční vrstvě by měla, jak již bylo dříve zmíněno, být ve správném pořadí a bez chyb. Pakety generované touto vrstvou se nazývají TLP (Transaction Layer Packets). Skládají se z hlavičky, dat a volitelně také CRC. Hlavička obsahuje informace určující vlastnosti paketu. První dvojslovo (4 byty) je pro každý typ paketu stejné a definuje především typ prováděné transakce, délku paketu a další informace o tom, jak s paketem nakládat. Jeden z atributů v hlavičce také udává, zda se na za daty nachází CRC, které bývá označované jako ECRC (End-to-end CRC).

Paket přijatý z nižší vrstvy by měl být sice bezchybný, avšak CRC generované linkovou vrstvou nijak nepokrývá možnost, že se v TLP objeví chyba vlivem průchodu transportní vrstvou. K tomu by mohlo dojít například ve switchi, kde projdou data z fyzické do transakční vrstvy a pak zpět, avšak v rámci rozdílných linků. ECRC tak dokáže zaručit, že data dojdou v pořádku až k cílovému příjemci.

## 5.3 Typy transakcí

PCI Express definuje čtyři druhy transakcí, a to paměťové, IO (Input Output), konfigurační a zprávy. Každý z těchto typů umožňuje přístup k jinému adresovému prostoru.

**Paměťové** transakce jsou využívány pro přenosy do a z paměťově mapovaných oblastí. Existuje několik typů paměťových transakcí, kterými jsou žádost o zápis, žádost o čtení a dokončení čtení. Paměťové transakce mohou používat 32bitové i 64bitové adresování.

**IO** transakce cílí IO mapovaný prostor. PCIe podporuje tento typ transakcí z důvodu zpětné kompatibility se zařízeními, jež IO prostor používají. IO transakce může využívat pouze 32bitové adresování. Podporuje čtyři druhy transakcí, a to žádosti o zápis, dokončení zápisu, žádosti o čtení a dokončení čtení.

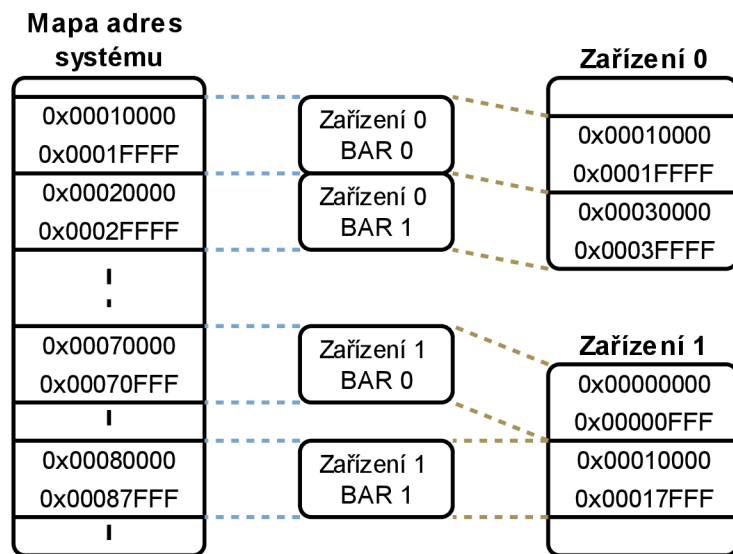
**Konfigurační** transakce umožňují přístup ke konfiguračním registrům zařízení. PCI Express nabízí oproti PCI větší konfigurační prostor o velikosti 4 kB.

**Zprávy** nahrazují kontrolní a statusové signály používané PCI. Umožňují odesílat přerušování, signalizovat chyby nebo řídit napájení.

## 5.4 Přenos dat

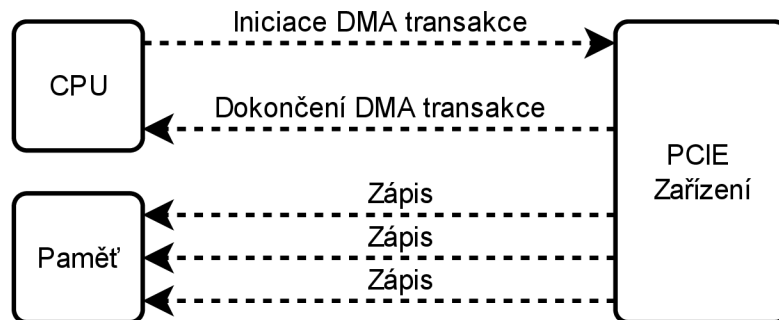
Existují dvě možnosti, jak v rámci PCIe architektury přistupovat k přenosu dat z jednoho zařízení do druhého. Těmito možnostmi jsou BARy (Base Address Register) nebo DMA (Direct Memory Access) přepnutím zařízení do režimu bus master. U obou přístupů závisí na tom, zda je daným zařízením podporován.

Zařízení mohou obsahovat až 6 BARů. Každý BAR má svou velikost a konzumuje místo v adresovém prostoru systému. Jeho účelem je převod adresy z adresového prostoru systému do adresového prostoru daného zařízení. Obrázek 42 demonstruje funkci BARů.



Obrázek 42. Princip funkce BARů

Další možností přenosu dat mezi zařízeními je DMA. Principem DMA je přenos dat bez nutnosti zapojit do procesu samotný procesor. Ten přenos pouze inicializuje vytvořením DMA deskriptoru, jež obsahuje zdrojovou a cílovou adresu, délku dat a další potřebné parametry. Zařízení se tak stává tak zvaným bus masterem, neboť přebírá řízení přenosu. DMA je demonstrováno obrázkem 43.



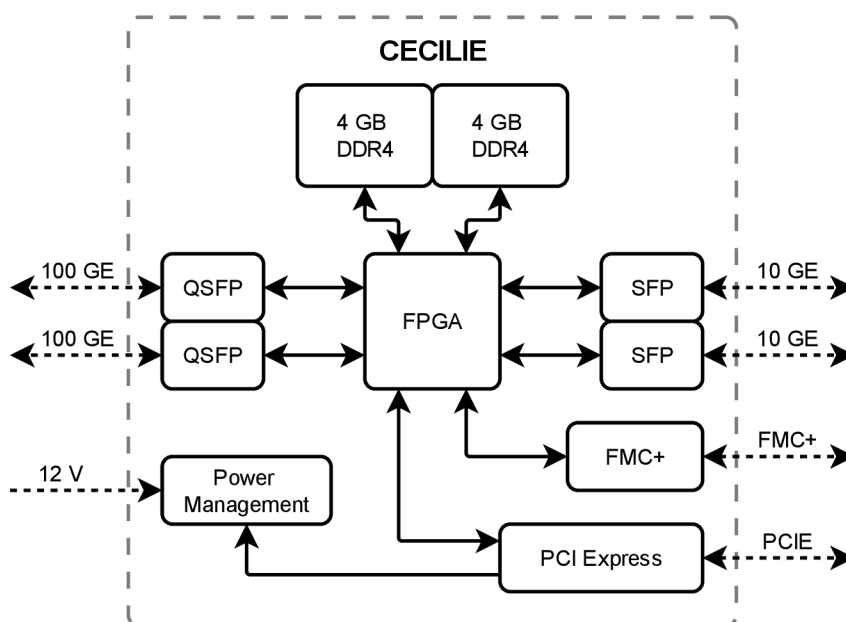
Obrázek 43. Princip DMA přenosu



## 6 HARDWARE

Použitou hardwarovou platformou je PCIe karta Cecilie od firmy DFC Design. Její blokové schéma je na obrázku 44. Karta disponuje hradlovým polem řady Kintex Ultrascale Plus, jež obsahuje hardwarové bloky realizující PCI Express a nejnižší vrstvy 100G Ethernetu. Oba tyto bloky budou v práci využity. Karta nabízí:

- FPGA Kintex Ultrascale Plus (XCKU11P)
- 2 x 4 GB DDR4
- 2 x QSFP (Quad Small Form-factor Pluggable) pro 100 Gbps Ethernet
- 2 x SFP (Small Form-factor Pluggable) pro 10 Gbps Ethernet
- PCI Express pro vložení do PC nebo Serveru
- FMC+ pro připojení rozšiřující karty (v případě této práce převodníkové)

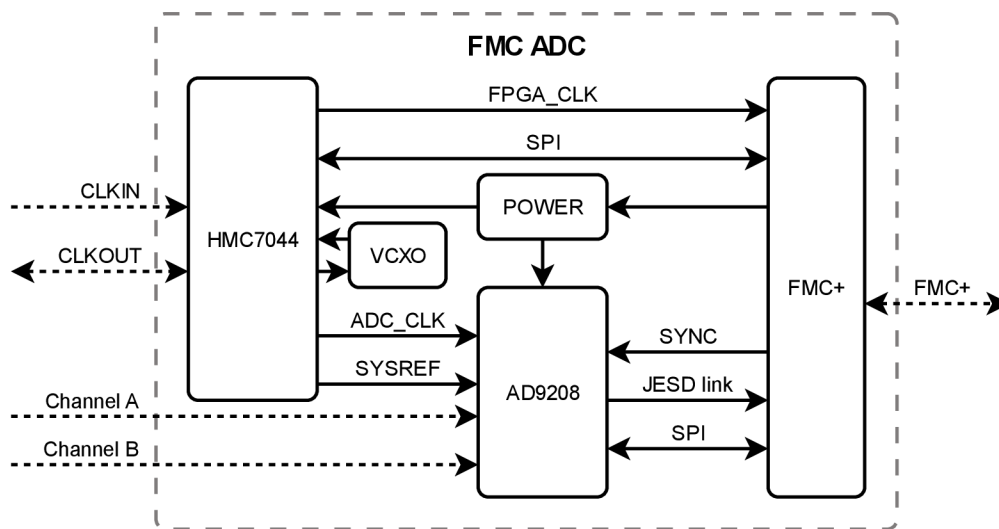


Obrázek 44. Blokové schéma CECILIE

Karta bude použita pro získávání dat z AD převodníku, který se nachází na rozšiřující FMC+ převodníkové kartě. Data budou v hradlovém poli zpracována a následně pomocí 100G Ethernetu nebo PCIe přenesena do PC nebo serveru.

Rozšiřující karta obsahuje AD převodník AD9208, obvody pro distribuci napájení a generátor referenčních hodinových signálů HMC7044. Převodník je s FPGA spojen osmi JESD204B linkami o maximálních rychlostech 16 Gbps. Vzorkovací frekvence převodníku je dána hodinovým signálem, generovaným obvodem HMC7044. Ten je určen primárně pro převodníkové a rádiové aplikace a vyznačuje se nízkým jitterem v řádech desítek fs. Blokové schéma rozšiřující karty je na obrázku 45.

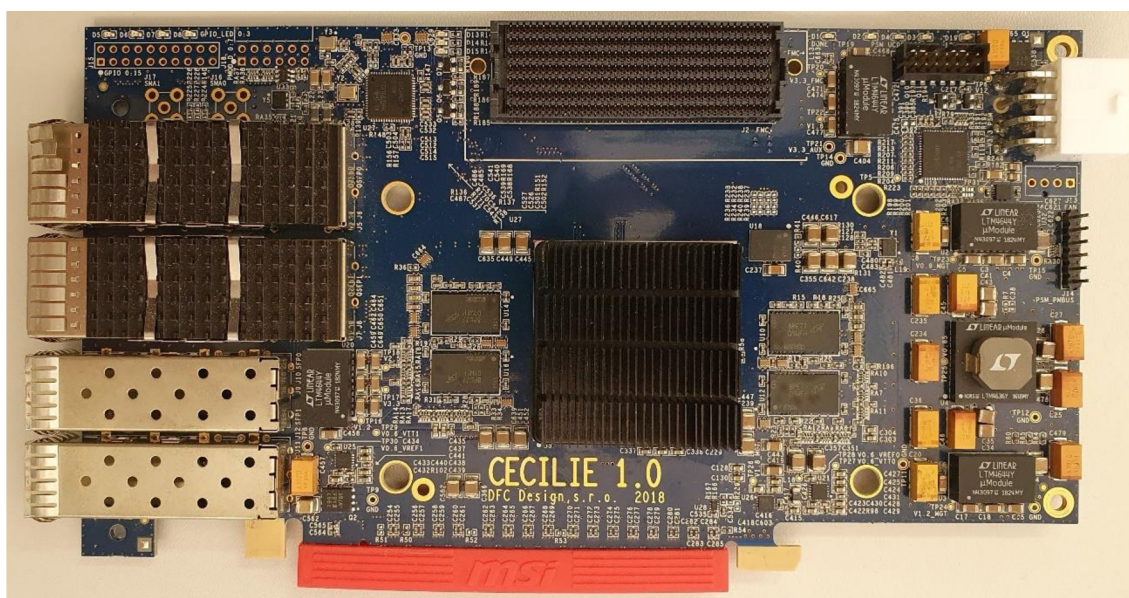
PCIe karta Cecilie a rozšiřující převodníková karta jsou na obrázcích 47 a 48. Celý řetězec použitých zařízení je pak ilustrován obrázkem 46.



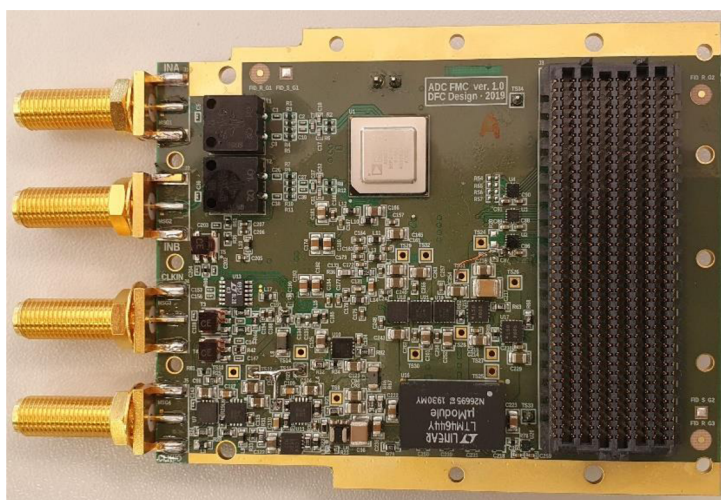
Obrázek 45. Blokové schéma rozšiřující převodníkové karty



Obrázek 46. Blokové schéma řetězce s PCIe kartou Cecilie a rozšiřující převodníkovou kartou



Obrázek 47. PCIe karta Cecilie

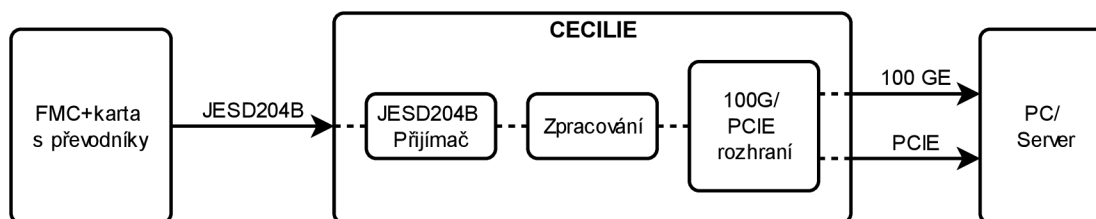


**Obrázek 48. Rozšiřující převodníková karta**

## 7 ROZBOR ŘEŠENÍ

Cílem práce je akvizice dat AD převodníkem, jejich přenos do FPGA, zpracování a následné přenesení do PC přes rozhraní PCIe nebo Ethernet. Bude zapotřebí vytvořit HDL design a firmware pro FPGA a software pro přijímací část v PC. Řešení designu lze rozdělit na tři části, a to získání dat z převodníku, jejich zpracování a jejich přenesení. Všechny části mohou být řešeny samostatně a na závěr sloučeny dohromady. Řešení je demonstrováno blokovým schématem na obrázku 49.

Vrstvy rozhraní JESD204B budou tvořeny modulem *JESD204 PHY* od společnosti Xilinx v kombinaci s modulem *JESD204B RX*, jenž mi byl dán k dispozici firmou DFC Design. Výstupem bude *AXI4-Stream (AXIS)*, obsahující data z obou kanálů převodníku. Bude zapotřebí vytvořit několik modulů, jejichž účelem bude přicházející data transformovat a následně odeslat rozhraním PCIe nebo 100G Ethernet. Moduly pro odesílání budou interagovat s bloky CMAC a XDMA od společnosti Xilinx.

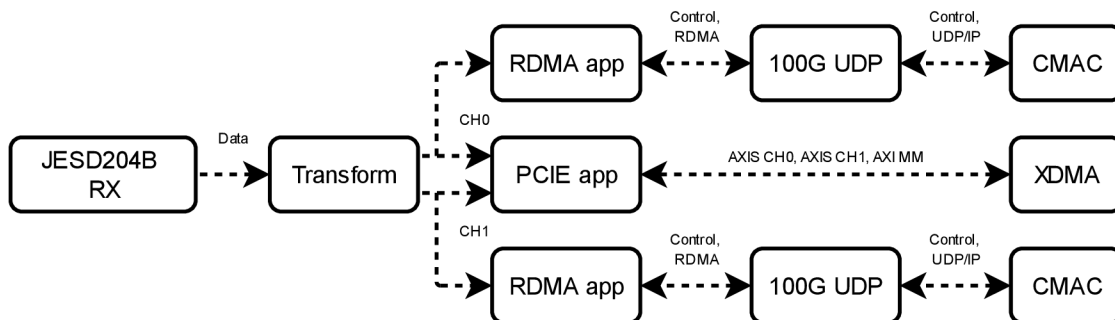


Obrázek 49. Blokové schéma řešení

Pro PC budou napsány dvě aplikace, jedna využívající RDMA a protokol RoCE a druhá sběrnici PCIe. Použití RoCE vyžaduje síťovou kartu, podporující tuto technologii a API (Application Programming Interface) pro realizaci RDMA. Použitou kartou bude ConnectX-5 od společnosti Mellanox. Jako vhodné API bylo vybráno libibverbs, které je open source a určené pro operační systém linux. Na rozdíl od Network Direct SPI (Service Provider Interface), jenž je jeho ekvivalentem pro operační systém Windows, je k libibverbs dostupné velké množství dokumentace a demo aplikací.

PCIe kartu nelze používat bez jejího ovladače. Cílem práce není jeho vytvoření, a proto bude využit již existující a volně dostupný ovladač od společnosti Xilinx. Ten je určen pro použití s demo designem k XDMA, avšak pro potřeby testování je postačující.

Spojením všech zmíněných částí vznikne kompletní řešení pro akvizici a ukládání dat z AD převodníku. Výsledné blokové schéma FPGA designu s detailnějším popisem toku dat je na obrázku 50.



Obrázek 50. Tok dat v aplikaci pro FPGA

## 7.1 Získání dat z AD převodníku

Převodník použitý pro převod měřeného analogového signálu na digitální je AD9208 od firmy Analog Devices. Jeho maximální vzorkovací frekvence je 3,1 GHz a maximální rozlišení 14 bitů. Obsahuje dva kanály s šířkou pásma 9 GHz. Jeho výstupem je rozhraní JESD204B, skládající se z osmi sériových linek, z nichž každá může přenést až 16 Gbps [14]. Vzhledem ke znalosti maximální přenosové rychlosti rozhraní JESD204B můžeme pomocí rovnice 11 vypočítat výslednou propustnost tohoto rozhraní.

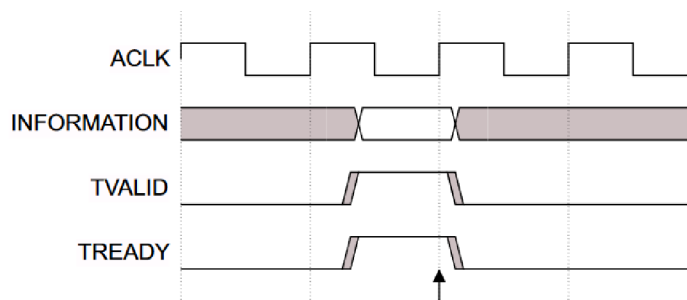
$$Propustnost = počet\ linek \cdot rychlost \cdot \frac{8}{10} = 8 \cdot 16 \cdot \frac{8}{10} = 99,2\ Gbps \quad (11)$$

Referenční hodinové signály pro převodník a FPGA jsou vytvořeny generátorem HMC7044. Tento integrovaný obvod umožňuje také přivedení externích hodinových referencí, na které se lze vnitřním PLL zavěsit. V kombinaci s napěťově řízeným oscilátorem pak může být výstupní hodinový signál odvozen ze vstupního signálu [15]. To následně převodníku umožňuje vzorkovat signál koherentně.

AD převodník i generátor mohou být konfigurovány přes rozhraní ADI-SPI (Serial Control Interface). Toto rozhraní je poměrně nestandardní. V podstatě se jedná o klasické SPI (Serial Peripheral Interface), avšak místo jednosměrných signálů MOSI a MISO obsahuje obousměrný signál SDIO. Společnost Xilinx nabízí pouze klasické SPI a bude tedy zapotřebí vytvořit komponentu, která provede konverzi na ADI-SPI.

JESD linky z AD převodníku budou připojeny k IP core *JESD204 PHY* od společnosti Xilinx, který zajistí fyzickou vrstvu protokolu JESD204B. Vyšší vrstvy protokolu pak zajistí *JESD204B RX* core od firmy DFC Design, jenž provede inicializační CGS a ILA fáze a jehož výstupem budou data z AD převodníku ve formě AXI4-Streamu o šířce 256 bitů a rychlosti 375 MHz (při vzorkovací frekvenci 3 GHz).

AXI4-Stream je rozhraní pro přenos dat mezi komponentami, udržované společností ARM. Jeho základem jsou signály TVALID, TREADY a TDATA, které jsou vztaženy k referenčnímu hodinovému signálu. Strana odesílající data je označována jako master, který ovládá signály TVALID a TDATA, a přijímající strana, ovládající signál TREADY, jako slave. Přenos probíhá tak, že master vystaví na TDATA validní data a nastaví TVALID. Slave potvrdí přečtení dat nastavením signálu TREADY. Poté může master buď pokračovat vystavením nových dat, nebo resetovat signál TVALID a označit je tak za neplatná. Ukázka AXI4-Streamu je na obrázku 51. Kromě zmíněných může rozhraní obsahovat také další signály, rozšiřující možnosti jeho použití [16].



Obrázek 51. AXI4 Stream [16]

AD převodník, HMC7044 a JESD204B RX budou konfigurovány soft-core procesorem Microblaze, jenž je součástí designu. Kromě konfigurace se bude starat také o správné resetování jednotlivých částí.

## 7.2 Transformace dat

Cílem této práce je pouze přenesení dat z AD převodníku do PC a jejich následná analýza. Nebude se tedy provádět žádná decimace, komprese nebo hledání příznaků. Data z převodníku budou pouze rozdělena na dva kanály, převedena do hodinové domény navazujícího rozhraní a přenesena tak, jak jsou.

## 7.3 Přenos dat do PC

Data mohou být přenesena do PC dvěma způsoby, a to pomocí rozhraní PCI Express nebo 100G Ethernet. FPGA obsahuje pro obě tato rozhraní hardwarové bloky, jež značně usnadňují jejich použití.

Hardwarový blok pro 100G Ethernet implementuje MAC a fyzickou vrstvu [17]. Umožňuje použití pokročilejších funkcí, jako například FEC (Forward Error Correction) nebo řízení toku dat. Vstupem do něj jsou data k odeslání, ze kterých spočítá FCS (Frame Check Sum) a tím přispívá ke zjednodušení aplikace.

Hardwarový blok pro PCIe implementuje všechny jeho vrstvy a kompletně se stará o řízení přenosu [18]. Nabízí dvě možnosti přenosu dat, a to PCIe AXI (Advanced eXtensible Interface) Bridge a PCIe DMA. První z nich tvoří most mezi PCIe a AXI nebo AXIS rozhraním, zatímco druhá implementuje DMA a umožňuje tak zařízení stát se bus masterem. Vzhledem k tomu, že množství přenášených dat bude velké a není vhodné zatěžovat procesor PC, zvolíme DMA. Přístup k datům může být buď přes paměťově mapované rozhraní AXI [19], nebo streamovací rozhraní AXI Stream. Vzhledem k charakteru aplikace, tedy kontinuálnímu toku dat z převodníků, bude vhodnější zvolit rozhraní AXI Stream.

Můžeme vypočítat teoretickou propustnost Ethernetu a PCI Expressu a zjistit tak, jaké množství dat jsme schopni těmito rozhraními přenést. Propustnost Ethernetu je již známa z jeho samotného názvu, tj. 100 Gbps. Propustnost PCIe může být vypočítána *rovnici 12*.

$$\text{Propustnost PCIe} = \text{počet linek} \cdot \text{rychlost} \cdot \frac{128}{130} = 16 \cdot 8 \cdot \frac{128}{130} \doteq 126 \text{ Gbps} \quad (12)$$

$$\text{Propustnost Ethernetu} = 100 \text{ Gbps} \quad (13)$$

Reálné propustnosti obou rozhraní budou ve skutečnosti nižší, neboť zde není počítáno s použitými protokoly, které představují neužitečná data. Vyšší poměr užitečných dat k neužitečným vede k vyšší propustnosti rozhraní. Z toho důvodu je zapotřebí pracovat s pakety maximálních velikostí.

Na výše zjištěných hodnotách propustností lze vidět, že by veškerá data z převodníků bez jejich předchozího zpracování nemohla být přenesena jediným 100G Ethernetovým rozhraním. Bylo by možné použít dvou rozhraní, avšak využití šířky jejich pásma by bylo pouze padesát procent. Oproti tomu PCI Express disponuje teoretickou propustností dostatečně vysokou na to, aby byla data přenesena bez předchozích úprav. Reálně může být tato propustnost omezena ještě komunikací nižších vrstev.

Část, jejímž úkolem bude přenos dat z PCIe karty do nadřazeného systému, bude s jistou nadstavbou využívat oba výše zmíněné hardwarové bloky pro 100G Ethernet a PCI Express. Přenos a jeho parametry budou nastavovány registry, přístupnými buď přes definovaný UDP port v případě Ethernetu, nebo v případě PCIe přes BAR mapovaný do uživatelského adresového prostor. Obě rozhraní budou použitelná současně.

## 8 DESIGN A FIRMWARE

Tato kapitola popisuje FPGA design a firmware. Je rozdělena do několika částí, které se věnují AD převodníku spolu s HMC7044 a JESD204B, transformaci dat, aplikaci využívající 100G Ethernet a aplikaci využívající PCIe.

Použité vývojové prostředí pro tvorbu HDL designu je Vivado 2019.1. Veškeré navržené moduly jsou napsány v jazyce VHDL. Firmware pro softcore procesorem Microblaze byl vytvořen ve vývojovém prostředí Xilinx SDK v jazyce C.

Pro získání dat z AD převodníku byly použity moduly *JESD204 PHY* od společnosti Xilinx a *JESD204B RX*, který jsem dostal k dispozici od firmy DFC Design a který realizuje funkce vrstev JESD204B. Modul *JESD204B RX* umožňuje pracovat ve všech podtřídách rozhraní, nabízí možnost volitelného deskramblování a konfiguraci přes AXI registry. Jeho výstupem je AXI4-Stream o šířce 256 bitů a rychlosti 375 MHz (při vzorkovací frekvenci 3 GHz).

Na *JESD204B RX* navazuje modul *TRANSFORM ADC STREAM*, jehož úkolem je připravit data pro odeslání rozhraním PCIe nebo 100G Ethernet. V modulu dochází k rozdělení dat na dva kanály, jejich převodu do hodinové domény výstupního rozhraní a rozšíření streamu na požadovanou šířku 512 bitů.

**Tabulka 2. Sumarizace použitých HDL modulů a firmwaru (zelená – mnou vytvořeno, oranžová – nakonfigurováno, modrá – dostal jsem k dispozici)**

Modul	Počet řádků kódu	Počet použitých CLB LUT/FF
JESD204 PHY	-	249/333
JESD204B RX	2600	4872/4112
TRANSFORM ADC STREAM	200	465/3432
RDMA APP	3100	21729/7002
UDP 100G	3500	3796/9283
CMAC	-	1533/2758
PCIE APP	1600	2447/1531
XDMA	-	69401/66434
MICROBLAZE	-	1366/1708
AXI QSPI	-	455/668
AXI IIC	-	415/383
AXI GPIO	-	82/238
SPI TO 3WIRE SPI	170	37/23
Microblaze firmware	1400	-

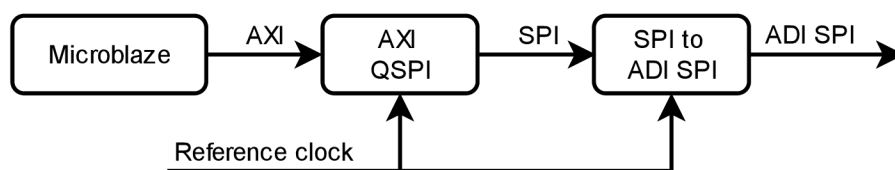
Data z modulu *TRANSFORM ADC STREAM* jsou přivedena do bloků realizujících PCIe a 100G Ethernet. Aplikace využívající PCIe se skládá ze vstupního



FIFa, AXI registrů, logiky umožňující použití triggeru a synchronizace kanálů a bloku pro předávání dat XDMA. Detailnější popis je v kapitole 8.4. Aplikace využívající 100G Ethernet se skládá ze dvou částí, a to z RDMA aplikace a modulu *100G UDP*. RDMA aplikace obsahuje vstupní FIFO, UDP registry a blok předávající data modulu *RoCE to 100G UDP*, který vytváří RoCE paket. Na výstupu aplikace je modul *UDP port switch*, jenž umožňuje slučovat streamy z dílčích částí. *100G UDP* kompletně realizuje UDP protokol, do kterého jsou zabaleny jak RoCE pakety, tak pakety z UDP registrů. Tabulka 2 sumarizuje použité moduly, přičemž jsou barevně odděleny nově vytvořené (zelená), nakonfigurované (oranžová) a ty, jenž jsem dostal k dispozici (modrá).

## 8.1 AD převodník, HMC7044 a JESD204B

Základním předpokladem pro korektní fungování celého řešení je správná konfigurace HMC7044 a AD převodníku. Ta se provádí pomocí rozhraní ADI SPI, které, jak již bylo výše zmíněno, je velice podobné rozhraní SPI [20]. Jejich rozdíl spočívá v tom, že jsou v rozhraní ADI-SPI signály MOSI a MISO nahrazeny jediným obousměrným signálem SDIO. Je tedy zapotřebí provést konverzi mezi těmito dvěma rozhraními. První bit přenosu vždy udává, zda se jedná o operaci zápisu či čtení. Následuje 15 adresových bitů a data. Velmi jednoduše lze zjistit, zda se bude po prvních 16 bitech číst či zapisovat a na základě této informace nastavit po odeslání adresy port SDIO jako vstupní, nebo výstupní. Aby tento princip fungoval, musí mít referenční hodiny pro *AXI QSPI* [21] a *SPI to ADI SPI* několikanásobně vyšší frekvenci, než je frekvence samotného SPI. To je v tomto případě splněno, neboť rychlost SPI bude šestnáctina referenčních hodin (dáno nastavením AXI QSPI). Na obrázku 52 lze vidět zjednodušený blokový diagram zapojení výše zmíněných komponent.



Obrázek 52. Blokové schéma zapojení komponent AXI QSPI a SPI to ADI SPI

Vstupem do FPGA je osm diferenciálních párů tvořících JESD204B link a referenční hodinový signál generovaný HMC7044. V případě JESD204B, pracujícího v podtřídě 1, by do FPGA vstupoval také signál SYSREF, avšak ten není z FMC karty vyveden. Rozhraní tedy bude použito v podtřídě 0.

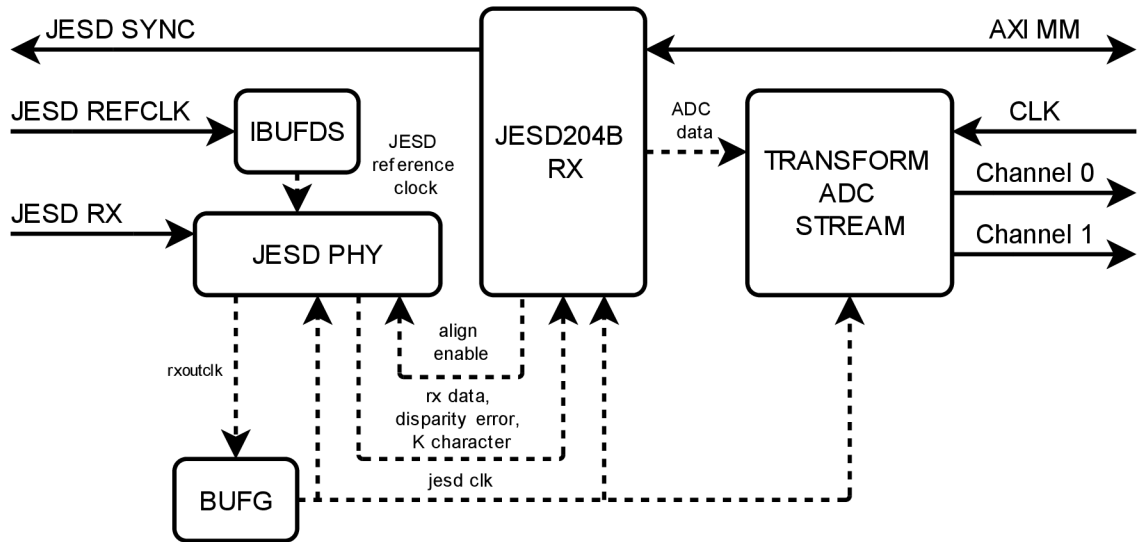
Referenční hodinový signál je diferenciální, a proto musí před přivedením do JESD204 PHY projít nejprve diferenciálním bufferem, který jej převede na single ended. Uvnitř komponenty je signál zapojen na vstup QPLL (Quad PLL), které jeho vynásobením vytvoří hodiny pro multigigabitové transceivery. Ty obsahují SERDESy, jež provedou vzorkování signálu přicházejícího JESD linkami. Jejich výstupem jsou paralelní data s frekvencí  $\frac{f_{lane}}{40}$ , kde  $f_{lane}$  je přenosová rychlost linky. Z JESD204 PHY pak vycházejí data spolu se signály *CHARISK* a *DISPERR* a *RXOUTCLK*. Aktivní bit signálu *CHARISK* značí, že byte v datech, odpovídající pozici tohoto bitu, obsahuje kontrolní K znak. Toho je využíváno v inicializační fázi přenosu a později také ke sledování stavu linky. Aktivní bit signálu *DISPERR* značí, že byte v datech, odpovídající pozici tohoto bitu, není validní z důvodu chyby disparity. *RXOUTCLK* je hodinový signál z přijímací části transceiverů. Je přiveden do *BUFG* bufferu, čímž se dostane do globálního hodinového rozvodu. Následně je zaveden zpět do *JESD204 PHY* a jsou k němu vztaženy všechny výše uvedené signály. *JESD204B RX* povoluje signálem *rxencommaligh* zarovnání vzorkovacích hodin, k čemuž dochází v průběhu CGS fáze [22].

Úkolem *JESD204B RX* je provést inicializační CGS a ILA fázi a následně na svůj výstup přivést data z AD převodníku. Komponentu je možné nastavit a resetovat zápisem do registrů, které jsou přes sběrnici AXI dostupné z procesoru Microblaze [23]. Nastavení se týká především parametrů přenosu, tedy počtu oktetů na rámec, počtu multirámců a volitelného deskramblování. Komponenta není mým vlastním dílem, avšak pro použití s řadou Kintex Ultrascale Plus musela být upravena. Jednalo se o *Input Double Data Rate* buffer (*IDDRE1*), který se v této řadě liší od řady, pro kterou byla komponenta vytvořena. Dále byla v komponentě nalezena chyba při provádění CGS fáze, kterou jsem opravil. Konkrétně šlo o příliš brzké uvolňování kontrolního signálu *SYNC*. Obrázek 53 ilustruje zapojení *JESD204 PHY* a *JESD204B RX* spolu s *TRANSFORM ADC STREAM*, kde dochází k transformaci vstupního streamu dat.

Pro korektní fungování HMC7044 a převodníku je zapotřebí jejich správná konfigurace. Toho docílíme zápisem do jejich registrů, které jsou přístupné přes ADI-SPI, jak již bylo zmíněno výše. Konfigurace je prováděna soft-core procesorem Microblaze, který je přes AXI rozhraní připojen k AXI QSPI, jak je zobrazeno na obrázku 52. Nejprve je nakonfigurováno HMC7044, které generuje referenční hodinové signály pro převodník a FPGA, a poté samotný AD převodník.

HMC7044 obsahuje dvě na sebe navazující PLL. Účelem prvního z nich je vyčistit vstupní hodinový signál od šumu a jitteru. K tomu se používá externí VCXO (napětově řízený oscilátor), který díky PLL uchová průměrnou frekvenci vstupu (většinou tu správnou). První PLL nemusí být využito a místo VCXO může být osazen oscilátor se známou a přesnou frekvencí. Druhé PLL pak slouží jako

násobička frekvence a používá pro svou činnost buď interní VCO nebo externí VCXO. V našem případě se jedná o interní VCO. Výsledný hodinový signál je přiveden do distribuční sítě, ze které je připojen ke každému z výstupů HMC7044. Každý z těchto výstupů obsahuje další děličky, pomocí kterých je možné vytvořit požadované frekvence.



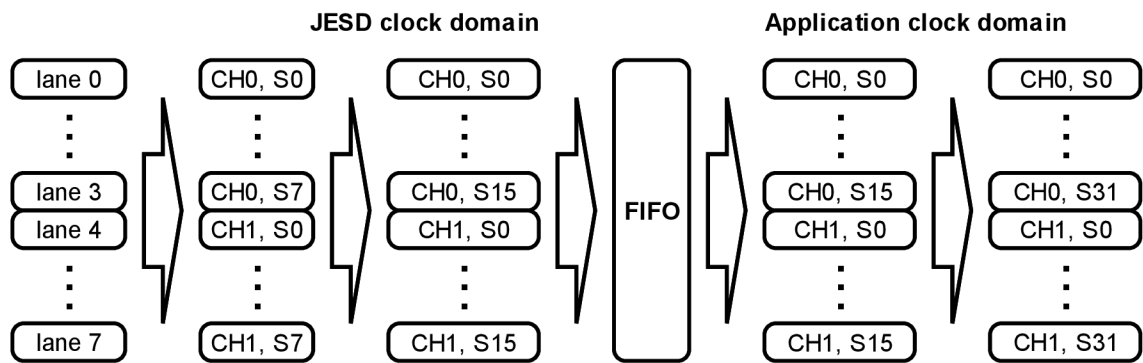
Obrázek 53. Blokové schéma zapojení JESD komponent v FPGA

Nejdříve je potřeba HMC7044 resetovat, a to buď hardwarově, nebo softwarově pomocí soft-reset registru. Poté do něj může být zapsána konfigurace, která byla vytvořena programem HMC704X\_GUI. Ten slouží k programování vývojového hardwaru k tomuto čipu, avšak je možné jej použít i bez něj a vytvořenou konfiguraci exportovat. Za účelem konverze dat z exportovaného souboru do .c zdrojového souboru, s konfigurací ve formě pole, byl napsán jednoduchý python skript *hmc\_cfg\_convert*. Takto vytvořená konfigurace je do HMC7044 zapsána a následně je proveden reset děliček a obou PLL, což je postup doporučený v dokumentaci k tomuto čipu [24].

Po úspěšné konfiguraci HMC7044 přichází na řadu AD převodník. AD9208 obsahuje dva kanály a nabízí velké množství aditivních funkcí, jako jsou FIR filtry nebo obvody určené k decimaci. Jejich použití vyžaduje hlubší znalost dané problematiky, a proto nebudou v rámci této práce použity.

Stejně jako HMC7044 je zapotřebí AD převodník nejprve resetovat, aby bylo zaručeno, že jsou všechny registry nastaveny na výchozí hodnoty. Před počátkem konfigurace je doporučeno deaktivovat všechny JESD linky. Následně se nastaví parametry JESD, jako je počet použitých linek, počet oktetů na rámec, počet rámců na multirámec, počet tail bitů a další. Kromě parametrů linky je možné nastavit

také formát výstupních hodnot, a to buď jako dvojkový doplněk, nebo čistě binární, a rozmezí přenosových rychlostí, ve kterém se budeme pohybovat. To je nutné z důvodu obsaženého PLL, které z referenčního hodinového signálu vytváří rychlejší hodinový signál pro SERDES, nacházející se ve výstupních obvodech JESD. Poté, co jsou všechny výše uvedené parametry nastavené, aktivují se JESD linky a provede se postkonfigurační sekvence, doporučená v dokumentaci k AD9208 [25]. Za účelem testování JESD jsou k dispozici testovací vzory, které slouží jak ke kontrole rozhraní, tak ke kontrole správnosti příchozích dat generovaných převodníkem. Testovací vzory byly použity k ověření integrity spojení mezi AD převodníkem a FPGA a následně celého řetězce tvořeného AD převodníkem, FPGA a PC.



Obrázek 54. Transformace dat z převodníku

## 8.2 Transformace dat

Jak již bylo zmíněno výše, není v rámci této práce plánována žádná složitější transformace dat, jako je jejich komprese či decimace. Z *JESD204 PHY* vychází kontinuální stream dat o šířce 256 bitů, přičemž první polovina jsou data z prvního kanálu a druhá z druhého kanálu převodníku.

Cílem transformace tohoto streamu je separovat data z obou kanálů a převést je z původní hodinové domény do domény obvodů sloužících k jejich odeslání. Prvním krokem je tedy rozdělení příchozích dat do dvou kanálů. V použité konfiguraci obsahují linky 0 až 3 data z prvního kanálu převodníku a linky 4 až 7 data z druhého kanálu, přičemž je nastavena velikost rámce jeden oktet. To v praxi znamená, že 4 současně přijaté oktety z prvních 4 linek obsahují dva vzorky, každý o velikosti 16 bitů (14 bitů dat + 2 kontrolní bity). Stream vycházející z *JESD204 PHY* má šířku 32 bitů  $\times$  počet linek, tedy 4 po sobě jdoucí rámce. Současně tedy dostaneme 8 vzorků z každého kanálu. Vzorky poskládáme tak, aby jedna polovina šířky streamu odpovídala vzorkům z jednoho kanálu a

druhá polovina vzorkům z druhého kanálu. Následně stream rozšíříme na 512 bitů tak, že data v jednom cyklu uložíme a v dalším je sloučíme s aktuálními daty. Takto je zapíšeme do FIFa, které slouží k přechodu mezi hodinovými doménami. Na výstupu FIFa data opět rozdělíme podle kanálů a vytvoříme dva streamy o šířce 512 bitů. Tato šířka odpovídá šířkám rozhraní použitých dále. Ilustrace manipulace s daty je zobrazena na obrázku 54.

### 8.3 Aplikace využívající 100G Ethernet a RoCE

Jednou z možností přenosu dat rychlostí v řádu desítek gigabitů za sekundu je 100G Ethernet. Jak již bylo zmíněno dříve, řada FPGA Kintex Ultrascale Plus obsahuje hardwarový blok CMAC, realizující fyzickou a MAC vrstvu 100G Ethernetu [17]. V rámci práce byla vytvořena komponenta *100G UDP*, jež plní funkci UDP/IP stacku a obsahuje logiku pro kontrolu CMAC. Kromě toho nabízí také možnost přijímat ARP žádosti a náležitě na ně odpovídat. Není tak zapotřebí přidávat do ARP tabulky cílové zařízení manuálně.

Hodinový signál, ke kterému jsou vztažena data vstupující a vystupující z CMACu, má frekvenci 322,66 MHz. Tato frekvence je poměrně vysoká a vzhledem ke krátké periodě znesnadňuje implementaci a časování signálů. Z tohoto důvodu funguje komponenta *100G UDP* v jiné hodinové doméně a je od CMACu oddělena FIFy. Minimální požadovaná propustnost streamu mezi CMACem a *100G UDP* je 100 gigabitů při šířce 512 bitů. Minimální frekvence, kterou může mít hodinový signál řídicí komponentu *100G UDP*, tak bude 200 MHz.

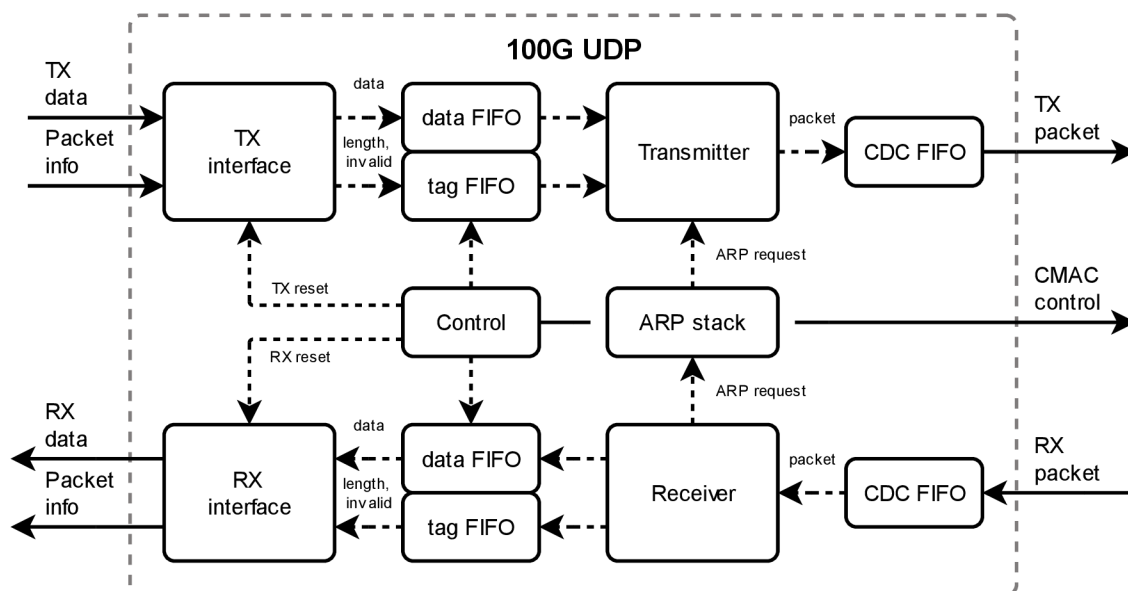
Vstupem do *100G UDP* na straně vysílací jsou uživatelská data ve formě AXI Streamu a aditivní informace nutné k sestavení UDP/IP paketu, jako jsou MAC adresy, IP adresy a UDP porty. Data i informace vstupují do TX rozhraní, jehož smyslem je kontrolovat, zda je příchozí stream validní, a zjistit délku zprávy a případně ji rozdělit tak, aby vyhovovala MTU (Maximum Transmission Unit). Z TX rozhraní jsou data zapisována do datového FIFa. S každým paketem v datovém FIFu je spojen jeden záznam ve FIFu s tagy. Tento záznam se skládá z cílové MAC adresy, cílové IP adresy, cílového a zdrojového UDP portu, délky paketu a informace o tom, zda jsou data v datovém FIFu validní či ne. V případě, že data validní nejsou, budou pouze vyčtena z FIFa a zahozena. V opačném případě jsou před data vložena MAC, IP a UDP hlavičky a vše je předáno výstupnímu CDC FIFu, ze kterého jsou následně vyčtena CMACem.

Opačným směrem pak vede stream z CMACu do CDC FIFa, ze kterého jsou data posláze vyčtena receiverem. Zde proběhne výpočet kontrolního součtu IP hlavičky a porovnání s přijatou hodnotou. Pokud kontrolní součet neodpovídá, je paket zahozen. V opačném případě jsou porovnány cílové MAC a IP adresy s přiřazenými adresami a pokud souhlasí, jsou data zapsána dále do datového FIFa.

Současně s daty je také přidán záznam do FIFa s tagy, opět jeden záznam pro každý paket v datovém FIFu. Tagy v tomto případě obsahují zdrojové IP a MAC adresy, zdrojové i cílové UDP porty, délku paketu a informaci o tom, zda jsou data v datovém FIFu validní. Na výstupu komponenty je RX rozhraní, které předává data a informace o paketu v případě, že jsou validní, uživateli.

Kromě UDP paketů detekuje přijímací část také ARP žádosti. Ty neprocházejí v RX řetězci dále, ale jsou zapsány, pokud jsou povoleny, do ARP stacku. Odsud mohou být v okamžicích mezi dvěma odesílanými pakety vyčteny a zpracovány. Zjednodušené blokové schéma komponenty *100G UDP* je na obrázku 55.

Pro bezproblémové použití je zapotřebí CMAC správně resetovat a následně jej provést inicializační fázi, při které se naváže spojení s protější stranou. Do doby, než je spojení dosaženo, jsou TX i RX části *100G UDP* drženy v resetu. O resetování i inicializaci se stará blok Control.

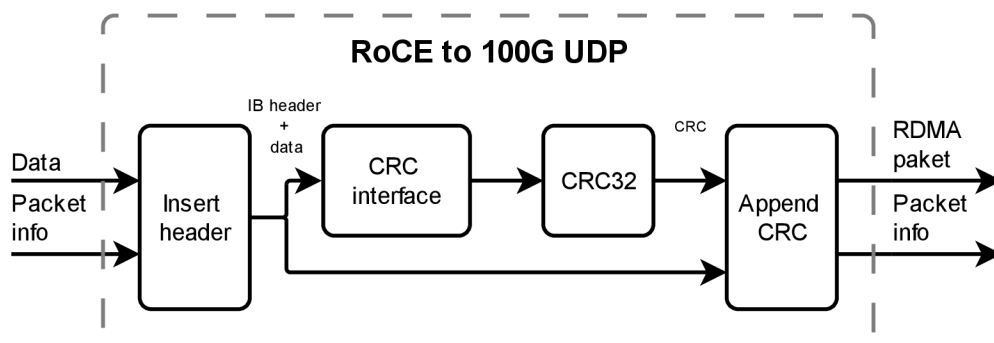


Obrázek 55. Blokové schéma komponenty *100G UDP*

Jak již bylo zmíněno dříve, pro přenos dat z FPGA do PC bude použit protokol RoCE, konkrétně jeho druhá verze. Ta zapouzdřuje část Infiniband protokolu v UDP/IP paketu a umožňuje tak využít již stávající infrastrukturu. Komponenta odesílající data zabalená v UDP/IP protokolu byla popsána výše. Je tedy zapotřebí další komponenty, která vezme data z převodníku a vloží je do RoCE paketu.

Tato komponenta se nazývá *RoCE to 100G UDP*. Vstupem jsou jí data k odeslání, spolu s požadovanou velikostí paketu a zdrojovými a cílovými adresami. Tyto informace jsou potřeba z důvodu nutnosti spočítat CRC celého IP paketu. Na vstupu komponenty se nachází modul, jenž před data vloží Infiniband

BTH a DETH hlavičky, které obsahují veškeré potřebné informace pro použití RoCE v režimu UD [8]. Tento režim byl zvolen z důvodu nejjednodušší implementace. Data s hlavičkami jsou následně předána CRC rozhraní a modulu vkládajícímu vypočtené CRC na konec paketu. CRC rozhraní vloží před data s RoCE hlavičkami IP a UDP hlavičky a vše předá CRC modulu. Modul připínající CRC nejprve data propouští dále a zastaví se až na konci paketu, kde čeká na dokončení výpočtu. Poté je za poslední data v paketu vloženo vypočtené CRC a paket je předán dále. IP a UDP hlavičky vytvořené v CRC rozhraní musí odpovídat hlavičkám vytvořeným v následující komponentě *100G UDP*. V opačném případě by příjemce po vypočtení CRC označil paket za poškozený a zahodil jej. Výpočet CRC je rozložen do několika kroků, a proto vkládá do řetězce zpoždění několika taktů. Jak bylo výše zmíněno, pro zaručení maximální propustnosti 100G rozhraní je při šířce streamu 512 bitů zapotřebí frekvence nejméně 200 MHz. Vzhledem k tomuto faktu může být zpoždění několika kroků výpočtem CRC kompenzováno použitím vyšší frekvence, v našem případě 250 MHz. Kromě datového streamu vycházejí z komponenty také informace o paketu, tedy MAC a IP adresy a UDP porty, jež jsou převzaty z jejího vstupu a jsou s paketem svázané. Blokové schéma na obrázku 56 demonstruje zapojení komponenty *RoCE to 100G UDP*.

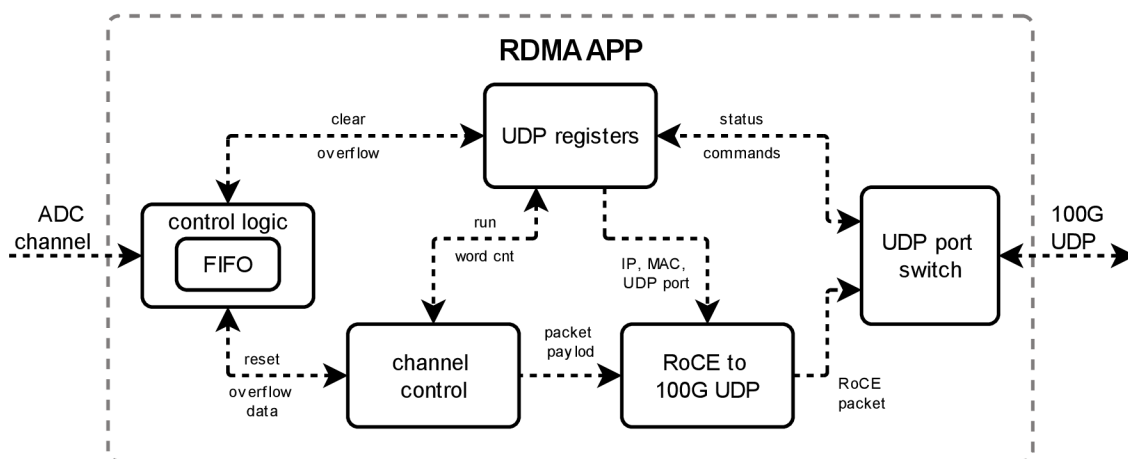


Obrázek 56. Blokové schéma komponenty *RoCE to 100G UDP*

Cílem je vytvoření aplikace, která umožňuje přenos dat z AD převodníku do PC. Část řetězce již byla představena, konkrétně tedy *100G UDP* a *RoCE to 100G UDP*. Dále je zapotřebí aplikace, jež využívá obou výše popsaných komponent a je zodpovědná za řízení toku dat.

Ta je reprezentována komponentou *RDMA ADC APP*, přijímá data z převodníků, která zabalí do RoCE paketu a předá *100G UDP* modulu. Vše je řízeno skrze UDP registry, umožňující řízení přenosu a nastavení destinace paketů. Registry jsou popsány v tabulce 3. Na vstupu komponenty je FIFO, jež je obklopeno řídicí logikou a v případě potřeby může sloužit kromě jiného také k přechodu mezi hodinovými doménami. Při vypnutém přenosu je FIFO drženo v resetu a data

z převodníku jsou zahazována. Po zapnutí přenosu zápisem do patřičného registru je FIFO vyvedeno z resetu a jsou do něj ukládána příchozí data. Ta jsou poté blokem *channel control* z FIFO vyčtena a předána *RoCE to 100G UDP*, jež bylo popsáno dříve. *Channel control* začne vyčítat data až poté, kdy je FIFO zaplněno nad určitý práh. Před každý soubor dat předaný k odeslání je vloženo sekvenční číslo, aby se posléze dalo určit, zda na sebe přijatá data navazují, nebo byl některý paket po cestě ztracen. Každý právě zpracovávaný paket je odeslán i při vypnutí přenosu. Toto opatření bylo přijato z důvodu nutnosti odesílat pakety stejné velikosti. V případě přetečení FIFO, například z důvodu nedostatečné propustnosti, je odeslán UDP paket s upozorněním a přenos je zastaven. Blokové schéma ilustrující zapojení komponenty je na obrázku 57.



Obrázek 57. Blokové schéma aplikace RDMA APP

Tabulka 3. Seznam a popis registrů aplikace RDMA APP

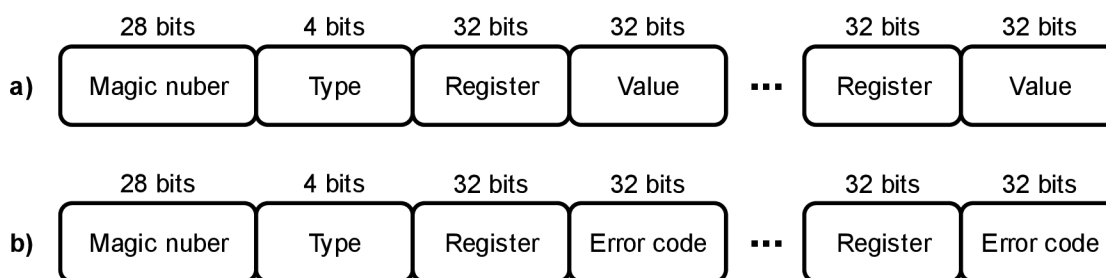
Address	Name	Description	
0x00000000	Destination IP	bit 31:0	IP address of destination node
0x00000001	Destination MAC LSB	bit 31:0	MAC address of destination node
0x00000002	Destination MAC MSB	bit 31:0	MAC address of destination node
0x00000003	Destination UDP	bit 15:0	RoCE UDP port of destination node
0x00000004	Destination QP	bit 31:0	QP number of destination Queue pair
0x00000005	First PSN	bit 31:0	First Packet Sequence Number
0x00000006	Q key	bit 31:0	Q key of destination application
0x00000007	Control register	bit 0	RUN - start sending data
		bit 1	Clear overflow
0x00000008	Status register	bit 0	Overflow - has to be cleared before asserting RUN
0x00000009	Word counter LSB	bit 31:0	64byte word counter LSB
0x0000000A	Word counter MSB	bit 31:0	64byte word counter MSB
0x0000000B	ID register	bit 31:0	Constant value 0xDFC0CEC0



Za účelem komunikace PC s Cecilíí musel být vytvořen jednoduchý protokol, umožňující základní práci s registry. Protokol se skládá ze tří částí, kterými je Magic number, příkaz a data. Magic number je konstantní 28bitové číslo s hodnotou 0xDfC1DFC, umožňující filtrovat pakety, jež nejsou určeny pro tuto aplikaci. Typ zprávy a interpretace navazujících dat je dána polem message type. Protokol podporuje 5 typů zpráv a to write, read, response, info a error. Jejich shrnutí a hodnoty type polí jsou v tabulce 4. Jak již napovídá název, write je určen pro zápis do registrů a read pro čtení z registrů. Typ info je žádostí o informace, tedy MAC a IP adresy dané karty, a může být použit v kombinaci s broadcastovým paketem pro nalezení všech připojených Cecilíí. Veškeré zprávy, jež jsou odpověďmi na jiné zprávy, jsou označeny typem response. Poslední z uvedených, tedy error, je určen pro signalizaci chyb. V současné době je jedinou signalizovanou chybou přetečení FIFO a ukončení přenosu. Obrázky 58, 59, 60 a 61 ilustrují formát jednotlivých typů zpráv.

**Tabulka 4. Typy zpráv podporovaných UDP registry**

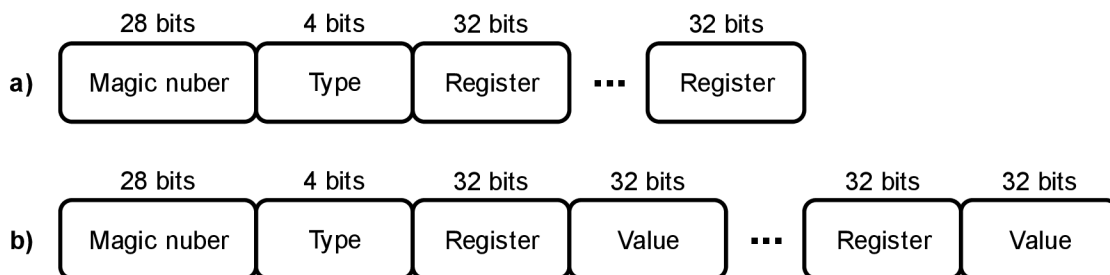
Type	Code	Description
Write	0x1	Write to UDP registers
Read	0x2	Read from UDP registers
Info	0x3	Get board informations request
Response	0x4	Response to write, read and info types
Error	0x8	Signaling error - overflow occurred



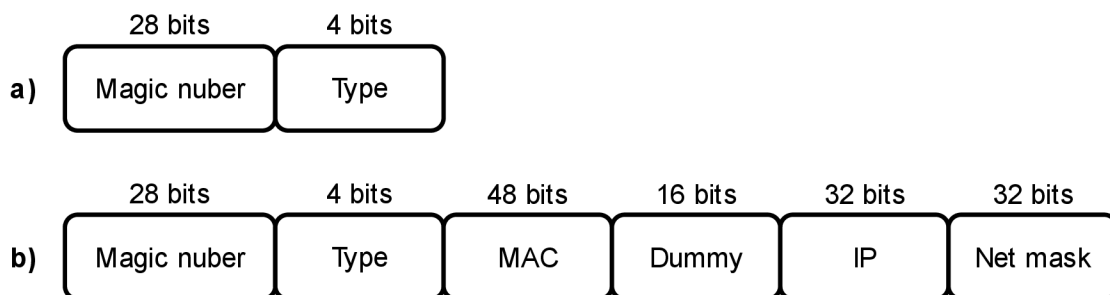
**Obrázek 58. Typ write**

Jak je možné vidět na obrázku 58, odpověď na zprávu typu write obsahuje adresy registrů, které byly cílem zápisu, a chybový kód. Tento kód může signalizovat, že se jedná o read-only registr, že daný registr neexistuje, anebo že zápis proběhl v pořádku. Na obrázku 59 je ilustrován formát paketu při čtení z registru. Zpráva typu read obsahuje pouze adresy registrů, jejichž hodnoty jsou v odpovědi požadovány. Odpověď pak obsahuje adresu tohoto registru následovanou jeho hodnotou. V případě neexistujícího registru je jeho adresa

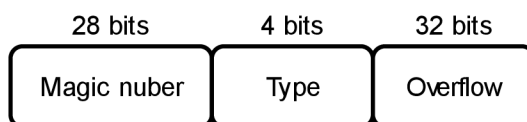
nahrazena hodnotou 0xFFFFFFFF. Odpověď na paket typu info se skládá z MAC adresy, IP adresy a masky sítě. Formát paketu je na obrázku 60. V případě, že v průběhu přenosu dat dojde k přetečení vstupního FIFO, je odeslán error paket s kódem overflow. Přenos je následně ukončen a před dalším použitím je nutné příznak overflow resetovat nastavením bitu clear overflow.



Obrázek 59. Typ read



Obrázek 60. Typ info



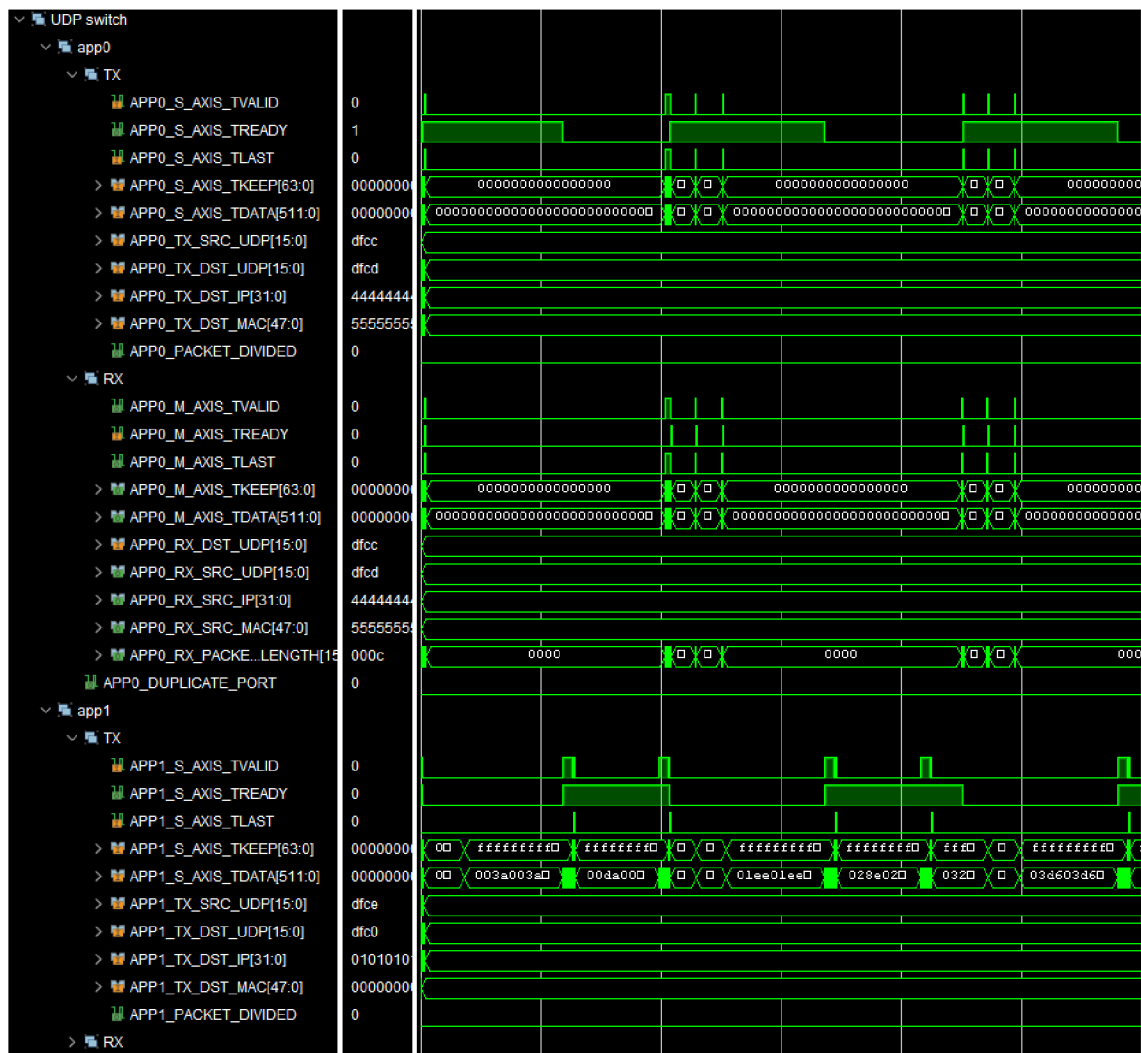
Obrázek 61. Typ error

Pro fungování RoCE je zapotřebí na straně přijímače hardware, tedy síťovou kartu, jež tuto technologii podporuje. Spolu s tím je také potřebné softwarové řešení, které bude popsáno dále v kapitole RDMA aplikace.

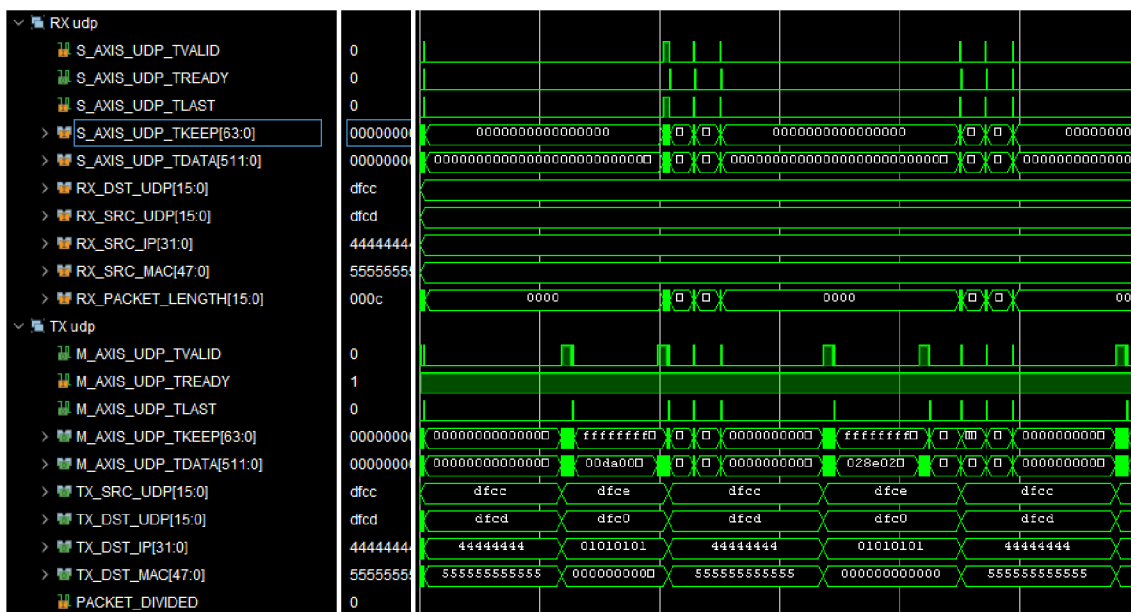
Aby bylo možné používat CMAc, je zapotřebí referenční hodinový signál vstupující do bloku transceiverů. Za tímto účelem je karta osazena hodinovým syntetizérem Si5332, který je nutno nastavit zápisem do jeho registrů, přístupných přes sběrnici I2C. Výrobce tohoto čipu nabízí konfigurační software ClockBuilder PRO, ve kterém je možné nastavit požadované parametry a konfiguraci ve formě .c zdrojového souboru následně exportovat. Stejně jako HMC7044 a AD převodník,

syntetizér je konfigurován pomocí soft-core procesoru Microblaze, ke kterému je přes sběrnici AXI připojen AXI IIC IP core od společnosti Xilinx [26].

Modul *RDMA APP* byl průběžně testován v simulátoru vývojového prostředí Vivado. Na obrázcích 62 a 63 jsou záznamy simulace, ve které jsou opakovaně získávány informace o kartě a je zapisováno do jejich registrů, včetně zapínání přenosu. Zachyceny jsou rozhraní bloku *UDP port switch*, přičemž je zde viditelné přepínání mezi aplikacemi app0 a app1, které reprezentují bloky *UDP registers* a *RoCE to 100G UDP*.



Obrázek 62. Simulace RDMA app s opakovaným zapínáním a vypínáním přenosu – strana aplikace



Obrázek 63. Simulace RDMA app s opakovaným zapínáním a vypínáním přenosu – strana 100G UDP modulu

## 8.4 Aplikace využívající PCI Express

Základem aplikace, přenášející data z AD převodníků do PC přes sběrnici PCI Express, je *DMA PCIE Subsystem* IP core od Xilinxu. Tento blok realizuje veškeré nižší vrstvy PCIe protokolu a umožňuje uživateli relativně snadné použití této sběrnice [18]. V rámci DMA nabízí možnost výběru ze dvou rozhraní, a to paměťově mapované AXI, nebo streamově orientovaný AXI Stream. Aplikace předpokládá kontinuální stream dat, a proto je vhodnou volbou AXI Stream. Kromě DMA rozhraní je možné použít i paměťově mapované rozhraní, které bude moci skrze určitý BAR přistupovat do adresového prostoru FPGA. Toho bude využito pro přístup k AXI registrům, pomocí kterých budou řízeny některé z funkcí této aplikace.

AXI registry nabízejí možnost synchronizace obou kanálů, nastavení triggeru a informace o stavu přenosu, jako je čítač přenesených dat nebo přetečení vstupního FIFO. Registry a jejich popis jsou uvedeny v tabulce 5. Nastavení triggeru umožňuje zvolit kanál, na kterém se má provést, vzestupnou či sestupnou hranu a úroveň signálu, při kterém má k triggeru dojít. Není zde zapotřebí RUN bitu jako v případě RDMA aplikace, neboť tato informace je propagována přes DMA Subsystem.

Stejně jako v případě RDMA aplikace jsou na vstupu komponenty FIFO, a to jedno pro každý z kanálů, která mohou sloužit v případě rozdílné datové a PCIe hodinové domény k jejímu překročení. Před jejich vstupy se nachází logika, jejímž úkolem je rozhodnout, za jakých okolností povolit zápis. FIFO jsou držena bloky

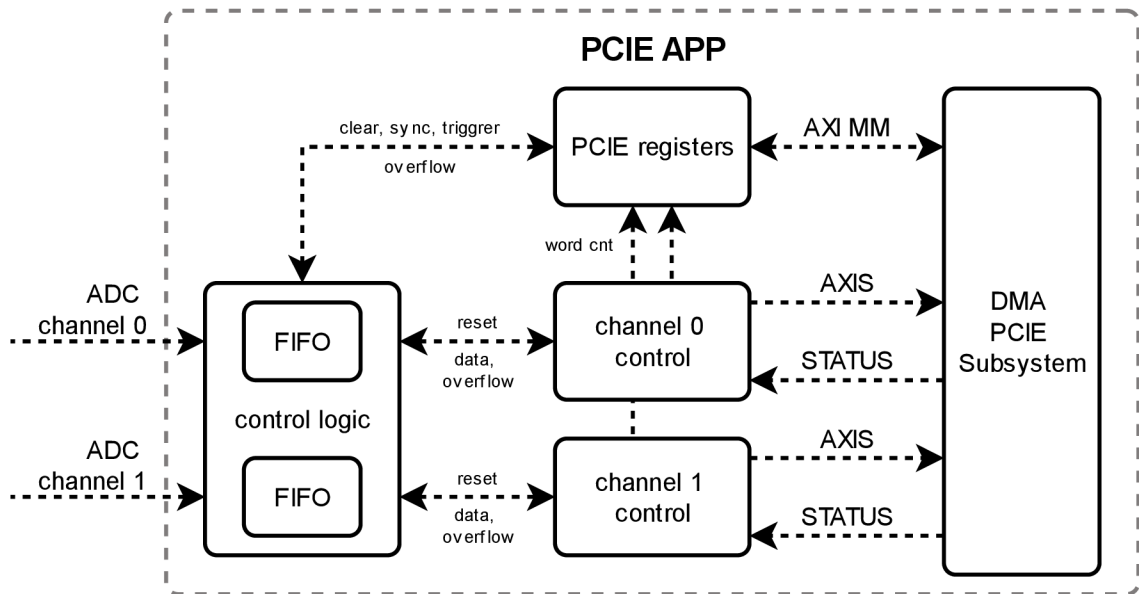
channel control v resetu až do chvíle, než přijde z DMA Subsystemu informace o započetí přenosu. Jakmile jsou FIFa vyvedena z resetu, dají logice na jejich vstupech signál, že jsou připravena, a v závislosti na nastavení registrů může být povolen zápis. Bez zapnutého triggeru a synchronizace povolí zápis okamžitě. V případě zapnuté synchronizace čeká na to, až jsou připravena obě FIFa, aby do nich mohla být zapisována současně přijatá data. Pokud je zapnutý trigger, čeká se kromě připravenosti FIF také na překročení nastavené úrovně.

**Tabulka 5. Seznam a popis PCIe registrů**

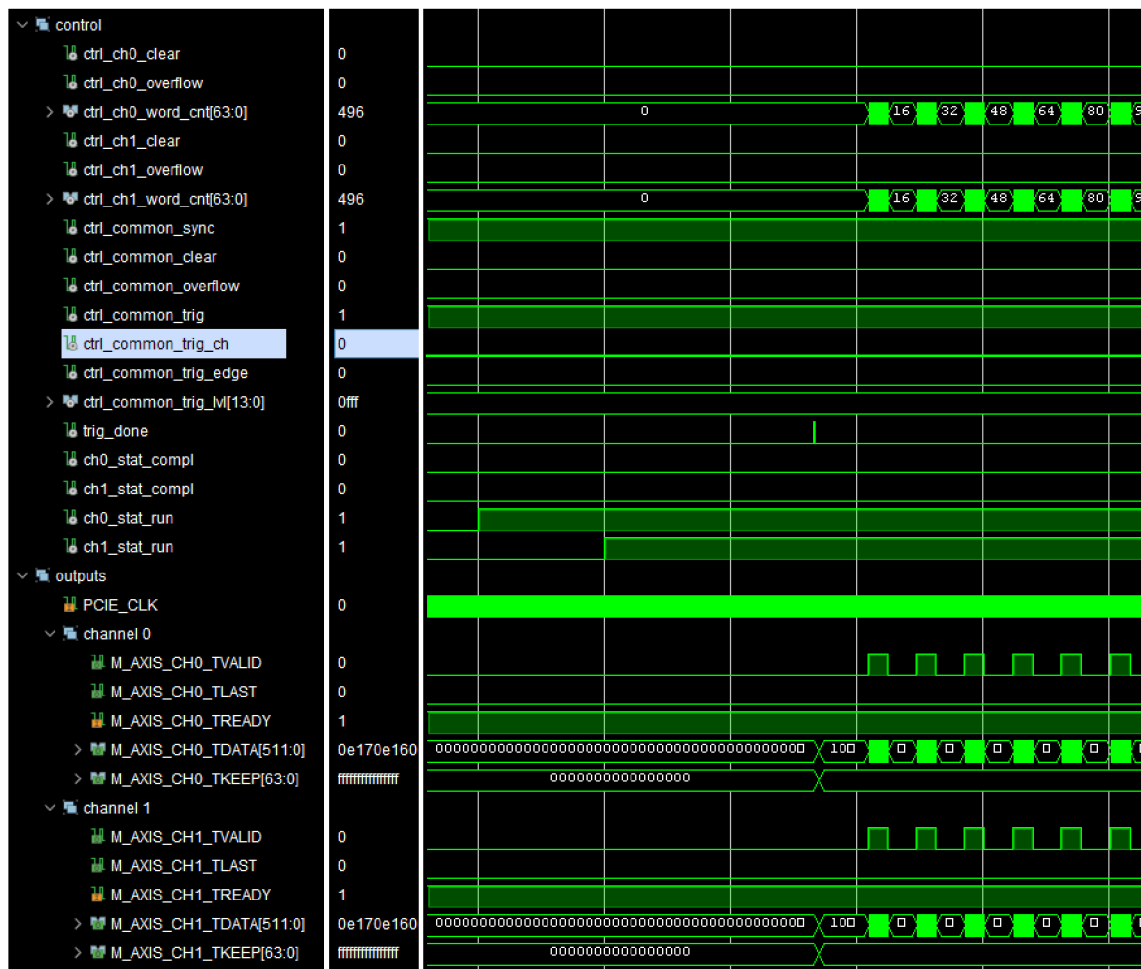
Offset	Name	Description	
0x00000000	Channel 0 control register	bit 1	Clear channel 0 overflow
0x00000004	Channel 1 control register	bit 1	Clear channel 1 overflow
0x00000008	Common status register	bit 0	Synchronization enable
		bit 1	Trigger enable
		bit 2	Clear both channels
		bit 3	Trigger channel
		bit 4	Trigger edge
		bit 18:5	Trigger level
0x0000000C	Channel 0 status register 0	bit 0	Channel 0 overflow
0x00000010	Channel 0 status register 1	bit 31:0	Channel 0 64byte word counter LSB
0x00000014	Channel 0 status register 2	bit 31:0	Channel 0 64byte word counter LSB
0x00000018	Channel 1 status register 0	bit 0	Channel 1 64byte overflow
0x0000001C	Channel 1 status register 1	bit 31:0	Channel 1 64byte word counter LSB
0x00000020	Channel 1 status register 2	bit 31:0	Channel 1 word counter LSB
0x00000024	Common status register	bit 0	Channel 0 or Channel 1 overflow
0x00000028	ID register	bit 31:0	Constant value 0xDFC0CECO

Data jsou z obou FIF vyčítána bloky channel control, které inkrementují počet odeslaných 64bytových slov a předávají je AXI registrům. Konec přenosu je opět indikován signálem z DMA Subsystemu, jež oznamuje, že byl použit poslední z DMA deskriptorů. Na obrázku 64 je blokové schéma komponenty *PCIE ADC APP*.

Modul byl během vývoje testován v simulátoru vývojového prostředí Vivado. Obrázek 65 demonstruje simulaci při zapnutí synchronizace (signál *ctrl\_common\_sync*) a triggeru (signál *ctrl\_common\_trig*). Pro iniciaci přenosu je zapotřebí, aby byly nastaveny signály *ch0\_stat\_run* a *ch1\_stat\_run* a zároveň byla překročena úroveň triggeru, což je reprezentováno signálem *trig\_done*. Po splnění těchto podmínek začne na obou kanálech přenos dat.



Obrázek 64. Blokové schéma aplikace PCIE APP



Obrázek 65. Simulace PCIe app při aktivované synchronizaci a triggeru

## 9 SOFTWARE

Pro přenos dat z karty do PC je nezbytný software, který se bude starat o jeho inicializaci, řízení, a ukládání přijatých dat. Bylo zapotřebí vytvořit dvě aplikace, jednu pro RDMA a druhou pro PCI Express, obě pod operačním systémem Linux. Ten byl vybrán jako vhodnější, a to především z důvodu snadnější implementace RDMA. Tabulka 6 sumarizuje použitý software, přičemž barevně odděluje mnou vytvořené aplikace (zelená) od ostatních (oranžová).

**Tabulka 6. Sumarizace použitého softwaru (zelená - mnou vytvořeno, oranžová - ostatní)**

SW + FW	Počet řádků kódu
RDMA app	2100
PCIE app	900
XDMA driver	-
checker	80
converter	70
hmc_cfg_convert	70

### 9.1 RDMA aplikace

RDMA je technologie, která musí být podporovaná samotným hardwarem, tedy síťovou kartou. Použitou síťovou kartou je ConnectX-5 od firmy Mellanox, nabízející dvě 100gigabitová rozhraní, podporující Ethernet a Infiniband. Dodávány jsou také ovladače, jež umožňují práci s kartou a přístup ke konfiguračním a debugovacím nástrojům.

Pro použití RDMA ve vlastní aplikaci je zapotřebí API, komunikující s nižšími vrstvami systému a případně i ovladačem k hardware. K dispozici jsou NetworkDirect [27] pro platformu Windows a libibverbs [28] nebo Kernel verbs pro unixové systémy. Vzhledem k dostupnosti velkého množství materiálů k libibverbs a naopak absenci návodů a dokumentace k NetworkDirect bylo rozhodnuto pro použití první zmíněné možnosti. Mimo jiné je libibverbs také open source a jsou tak k dispozici veškeré zdrojové soubory.

API lze rozdělit do dvou částí, a to kontrolní a datovou. Kontrolní část obsahuje vytvoření protekční domény, alokaci paměti a její registraci, vytvoření CQ, QP a nastavení a modifikaci jejich parametrů. Provádění těchto operací v průběhu přenosu není doporučeno, neboť jsou časově náročné a nevhodné při nutnosti rychlé odezvy na události. Datová část obsahuje přidávání žádostí do send queue a receive queue a vyčítání položek z CQ. V rámci práce vytvořená aplikace je inspirovaná příklady, dostupnými z vývojového repositáře na GitHubu.

Aplikace je konzolová a nabízí jednoduché menu. Umožňuje nalézt všechny připojené karty a zapsat či číst hodnoty jejich registrů. Nejdůležitější je však možnost přenést a uložit data, a to v množství omezeném především velikostí operační paměti. Celá aplikace zde bude představena ve formě jednoduchých vývojových diagramů.

Na obrázku 66 je diagram základního menu. Aplikace nabízí 9 možností, kterými jsou:

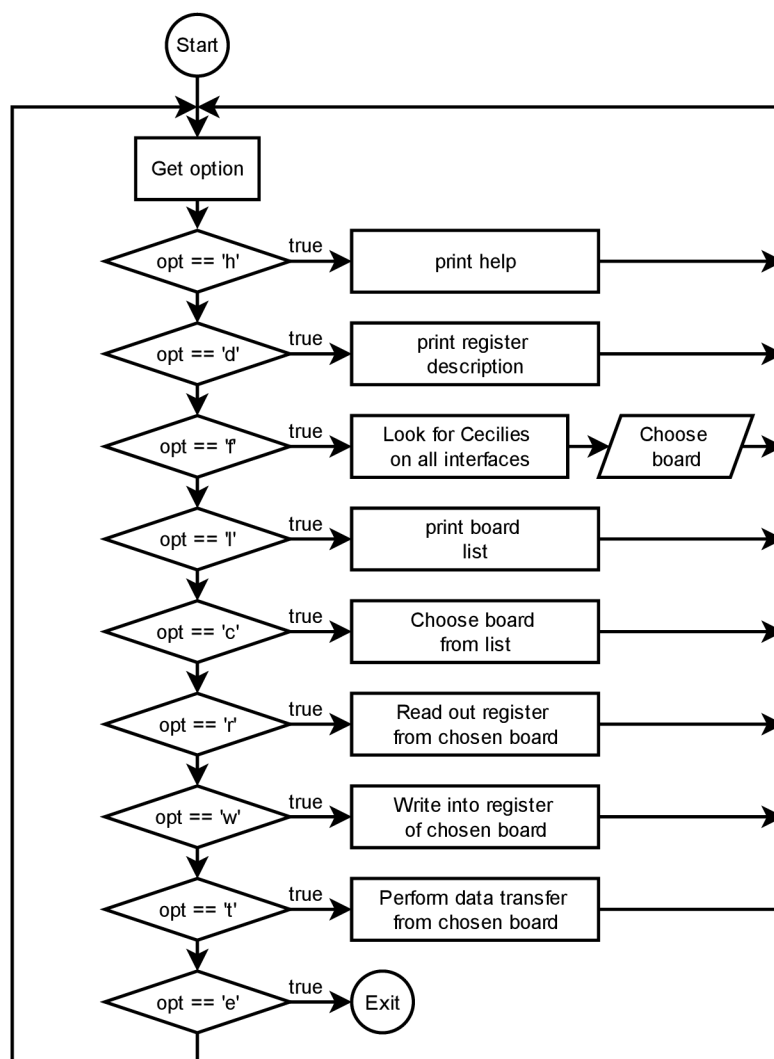
- h – Zobrazí nápovědu, tedy seznam zde uváděných možností.
- d – Zobrazí seznam registrů dostupných v kartě Cecilie a jejich popis.
- f – Projde všechny dostupné ethernetové porty a odešle na ně Discovery paket. Následně vypíše seznam všech nalezených karet a umožní vybrat jednu, se kterou se bude následně pracovat.
- l – Vypíše seznam dříve nalezených karet.
- c – Umožní vybrat jednu z dříve nalezených karet.
- r – Vyčte hodnotu registru dříve zvolené karty. Adresa registru je interaktivně získána od uživatele.
- w – Zapiše hodnotu do registrů dříve zvolené karty. Adresa a hodnota registru jsou interaktivně získány od uživatele.
- t – Provede RDMA přenos dat z dříve zvolené karty. Množství dat a název souboru pro zápis jsou interaktivně získány od uživatele.
- e – Ukončení aplikace.

Způsob zápisu do registrů zde nebude popisován, neboť se jedná pouze o odeslání UDP paketu. Méně známý, a tudíž zajímavější, je však způsob použití RDMA. Po zvolení možnosti *t* je uživatel dotázán na požadované množství přenesených paketů (jeden paket má velikost 4 kB) a na název souboru, do kterého budou data zapsána.

Zařízení podporující RDMA je ovladačem vytvořeno jako IB (Infiniband) zařízení. Nejprve je potřeba spárovat známé ethernetové rozhraní se správným IB zařízením. Poté je možné s tímto zařízením dále pracovat. Veškerá potřebná paměť musí být z důvodu časové náročnosti alokována a registrována před započítím přenosu. Registrace paměti spočívá v asociaci příslušného paměťového regionu s protekční doménou. Při registraci je nastaven požadovaný přístup k paměti, kterým může být zápis nebo čtení, a to v obou případech buď lokální, nebo vzdálené. V případě, že jsou použity události, které šetří výpočetní zdroje systému, je v tomto bodě možné vytvořit completion channel. Následně je možné vytvořit a inicializovat CQ a QP, kterému je potřeba nastavit hloubku work request fronty a Qkey. Tímto končí kontrolní fáze a QP čeká na přechod do stavu ready.



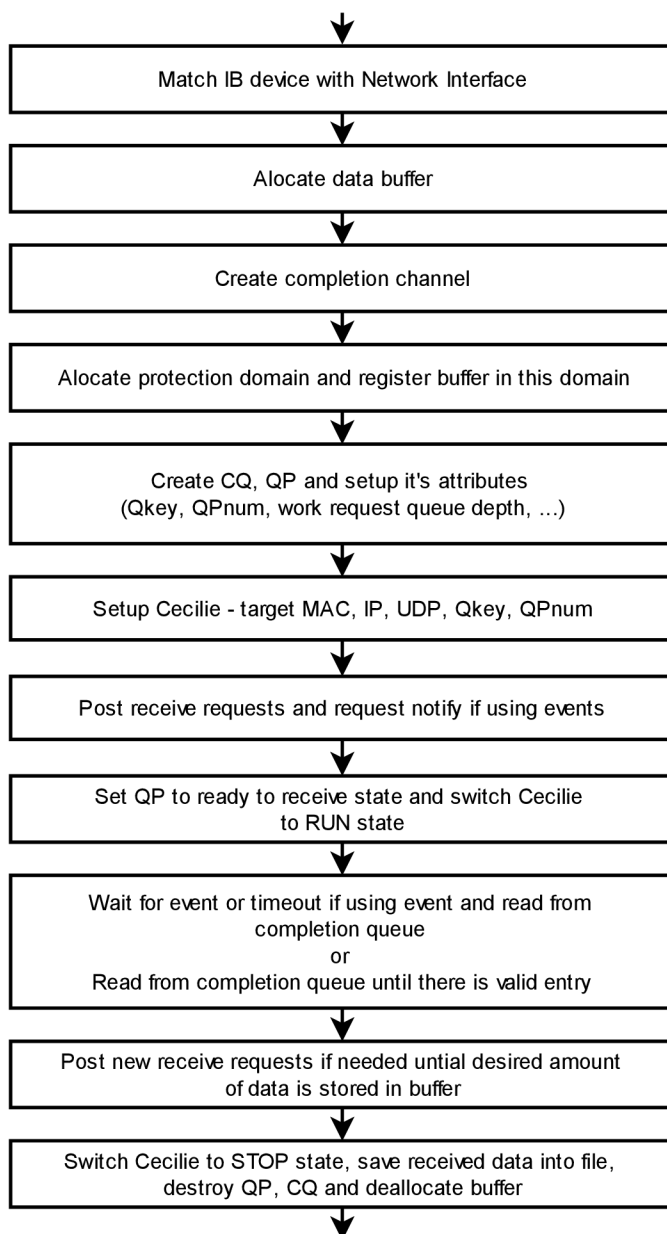
Poté jsou na základě parametrů vytvořeného RDMA serveru nastaveny registry karty, je naplněna RQ a zapnut požadavek na notifikaci událostí. Přenos dat může začít a QP je tak přepnut do stavu ready a karta do stavu run. Každý přijatý RoCE paket spotřebuje jednu žádost v RQ. Je proto potřeba držet si přehled o tom, kolik nespotřebovaných žádostí zbývá a v případě nutnosti je doplňovat, dokud nebylo přijato požadované množství dat a zaplněna celá alokovaná paměť. Přehled o již zpracovaných žádostech je držen díky záznamům v CQ, které jsou postupně vyčítány.



Obrázek 66. Blokové schéma hlavního menu RDMA aplikace

Jakmile je přijato požadované množství dat, je karta přepnuta do stavu stop a data jsou uložena do souboru. Následně je dealokována použitá paměť, spolu s instancemi QP a CQ. Tím je ukončen přenos a aplikace se vrací zpět do hlavního menu.

Každý RoCE paket obsahuje sekvenční číslo pro kontrolu správnosti pořadí. Při použití UD bohužel není na tento údaj brán ohled a není ani nikterak propagován k uživateli. Z toho důvodu každý paket obsahující data nese na svém začátku 32bitové sekvenční číslo, aby bylo možné odhalit chybějící paket a zamezit tak chybám v dalším zpracování. Na obrázku 67 je diagram znázorňující výše popsanou činnost aplikace při přenosu dat.



**Obrázek 67. Blokové schéma inicializace a průběhu RDMA přenosu**

Ukázka aplikace je na obrázcích 68, 69 a 70. První z uvedených demonstruje její chování po zapnutí, kdy je uživateli zobrazena nápověda, popis registrů a dostupná rozhraní. Pro přenos dat je nejprve potřeba najít všechny připojené karty a zvolit

jednu, se kterou se bude dále pracovat. To lze vidět na obrázku 69. Následně může být přenos iniciován, přičemž množství přenesených dat je omezeno pouze množstvím dostupné paměti RAM. Na obrázku 70 je nastavení jeho parametrů a výsledná statistika s dobou trvání a dosaženou propustností. Přijatá data jsou uložena v souboru, který má v tomto případě vzhledem k množství přenesených dat velikost 8 GB. Soubor a jeho velikost je vidět na obrázku 71.

```
linux@linux:~/projects/cecilie/cecilie_app$ ./cecilie_app
*** Cecilie control ***

Cecilie control help:
'h'   Print this help
'd'   Register description
'f'   Find cecilie
'l'   Print cecilie list
'c'   Choose board
'w'   Write register(s)
'r'   Read register(s)
't'   Start RDMA transfer
'e'   Exit

Description of Cecilie registers:
DST_IP_REG      0x00000000 Destination IP of RoCE packet
DST_MAC_HIGH    0x00000001 Destination MAC of RoCE packet - higher 32 bits
DST_MAC_LOW     0x00000002 Destination MAC of RoCE packet - lower 32 bits
DST_UDP         0x00000003 Destination UDP port of RoCE packet
DST_QP          0x00000004 Destination QP number of RoCE packet
FIRST_PSN      0x00000005 Sequence number of first RoCE packet
Q_KEY          0x00000006 Q Key of RoCE packet
CONTROL        0x00000007 Control register (0 bit = run/stop; 1 bit = clear)
STATUS         0x00000008 Status register (0 bit = overflow)
WORD_CNT_HIGH  0x00000009 Counter counting already sent 64 byte words - higher 32 bits
WORD_CNT_LOW   0x0000000a Counter counting already sent 64 byte words - lower 32 bits
ID             0x0000000b Identification register = 0xDFC0CEC0

Number of available interfaces: 4
eno2          9f:f9:13:d2:d9:c8   192.168.137.32
eno1          9e:f9:13:d2:d9:c8   210.19.249.158
enp33s0f0     56:ad:a9:9b:03:98   192.168.135.203
enp33s0f1     57:ad:a9:9b:03:98   192.168.136.204

Choose option: 
```

Obrázek 68. Aplikace cecilie\_app po spuštění

```
Choose option: f
Looking for Cecilies ...
  Found 0 boards on interface eno2
  Found 0 boards on interface eno1
  Found 0 boards on interface enp33s0f0
  Found 1 boards on interface enp33s0f1
Cecilie board list:
Cecilie 0:      enp33s0f1      8b:7b:67:ec:10:54      192.168.136.205
Choose board: 0
Cecilie:        enp33s0f1      8b:7b:67:ec:10:54      192.168.136.205

Choose option: 
```

Obrázek 69. Aplikace cecilie\_app při hledání připojených karet

```

Choose option: t
Start RDMA transfer
Packet size: default 4096
Packet count: 1000000
Results file name: capture
Setting cecilie registers ...
Register      7: write OK
Writing configuration to cecilie
Starting RDMA server (pkt size [4096], pkt_cnt [1000000], rx_depth [30000]) ...
Register      7: write OK
Register      7: write OK
Transfer time: 682 ms, Transferred: 4.136000 GB, Throughput: 6.064516 GBps / 48.516129 Gbps
Choose option: 

```

Obrázek 70. Aplikace cecilie\_app při přenosu dat z karty do PC

```

linux@linux:~/projects/cecilie/cecilie_app/results$ ls -l
total 8000044
-rw-r--r-- 1 linux linux 8192000000 May 25 07:36 capture
-rwxr-xr-x 1 linux linux 8928 Apr 7 04:42 checker
-rw-r--r-- 1 linux linux 1652 Apr 7 04:42 checker.c
-rwxr-xr-x 1 linux linux 13112 May 25 07:38 convert
-rw-r--r-- 1 linux linux 1281 Apr 7 06:36 convert.c
-rw-r--r-- 1 linux linux 142 Apr 6 05:04 makefile

```

Obrázek 71. Výsledek přenosu dat z karty do PC rozhraním 100G Ethernet

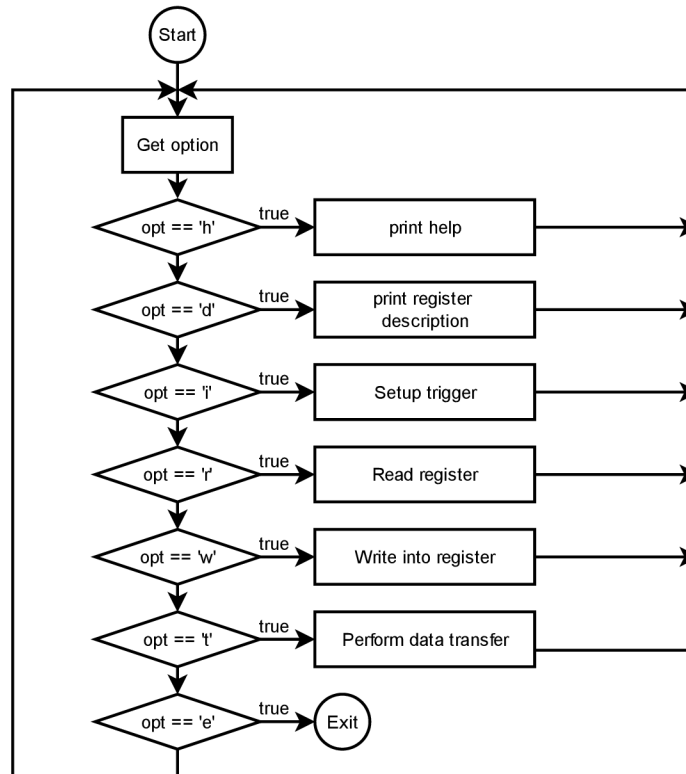
## 9.2 PCI Express aplikace

Prvkem, bez kterého nelze PCIe kartu používat, je její ovladač. Ten funguje jako modul kernelu, tedy v nižší vrstvě systému, a interaguje přímo s hardwarem. Vytvoření vlastního ovladače je relativně složitá věc, která si žádá hlubší pochopení jeho struktury a další studium. Z toho důvodu byl použit již hotový ovladač, jenž slouží jako příklad výrobce a má svá omezení, která však nezabraňují jeho použití k ověření funkčnosti řešení.

Aplikace využívající sběrnice PCIe bude stejně jako aplikace využívající RDMA popsána sérií vývojových diagramů, jež v jednoduchosti popisují její funkce. Aplikace je také konzolová s prostým menu, umožňující zápis do registrů karty a čtení dat z AD převodníku. Kromě zmíněného je podporována synchronizace kanálů a trigger. Ten může být nastaven na jednom ze dvou kanálů, a to na vzestupnou či sestupnou hranu a volitelnou úroveň signálu. Obrázek 72 ilustruje základní menu aplikace. V nabídce jsou tyto možnosti:

- h – Zobrazí nápovědu, tedy seznam zde uváděných možností.
- d – Zobrazí adresy a popis registrů karty.
- w – Umožňuje zápis hodnoty do registru karty. Adresa registru i jeho hodnota jsou interaktivně získány od uživatele.
- r – Umožňuje čtení hodnoty registru karty. Adresa registru je interaktivně získána od uživatele.

- i – Umožňuje zapnutí a nastavení triggeru. Od uživatele jsou interaktivně získány informace o použitém kanálu, hraně a úrovni signálu.
- t – Provede přenos dat z karty do paměti PC.

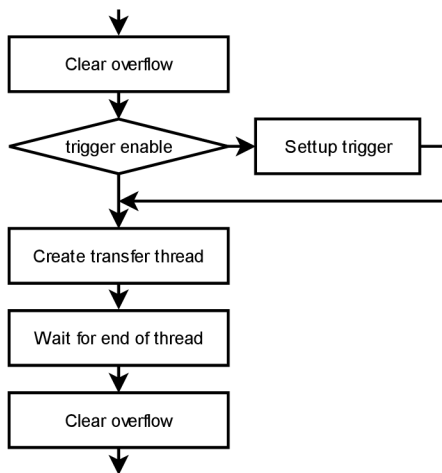


Obrázek 72. Blokové schéma hlavního menu PCIe aplikace

Operace zápisu a čtení z registrů fungují na principu mapování paměťového prostoru karty do paměťového prostoru aplikace. K registrům je pak možné přistupovat přímo, stejně jako k jiným způsobem alokované paměti.

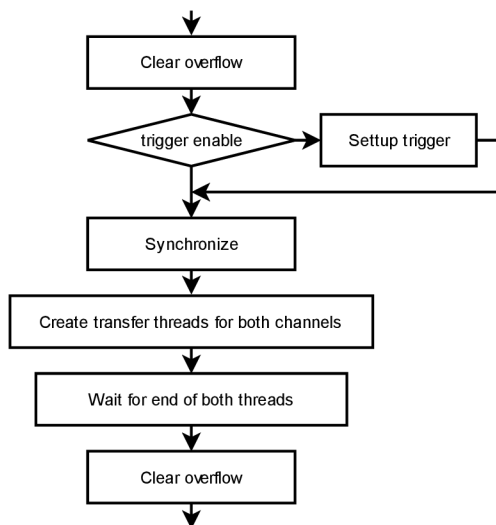
Přenos dat je započat inicializací DMA přenosu, který má kompletně na starosti ovladač. Z pohledu uživatele se jedná o čtení ze souboru, kterým je zařízení vytvořené pro daný DMA kanál. Po zvolení možnosti *t* je uživatel dotázán na kanál, ze kterého mají být data čtena. Na výběr je buďto jeden z kanálů samostatně, nebo oba kanály současně. V obou případech je před započítím přenosu nastaven a následně vymazán *clear* bit, aby bylo označeno případné předchozí přetečení FIFO za akceptované. Poté je zkontrolováno, zda je povolen trigger a případně je nastaven podle zvolených parametrů. V případě přenosu z obou kanálů současně je také nastaven *sync* bit. To je zapotřebí, aby bylo zaručeno, že data z jednotlivých kanálů na stejné pozici byla vzorkována ve stejný okamžik. Poté, co jsou všechny potřebné registry správně nastaveny, jsou vytvořena vlákna, ve kterých proběhne čtení ze zařízení a následné uložení přijatých dat do souboru. Důvodem pro použití vláken je současný přenos dat z obou kanálů. Po dokončení přenosu a uložení dat

jsou vlákna ukončena. Na závěr je opět akceptováno případné přetečení. Obrázky 73 a 74 demonstrují inicializaci přenosu jak pro jednotlivé kanály, tak pro oba kanály současně.



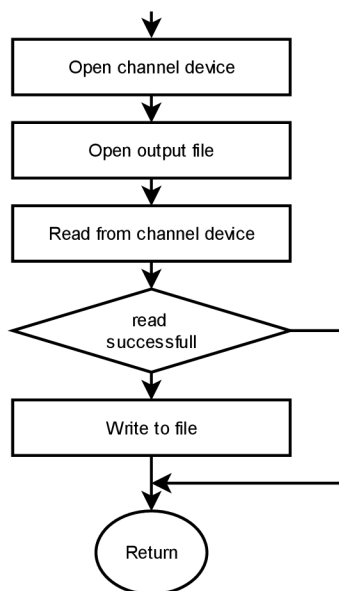
**Obrázek 73. Blokové schéma inicializace PCIe přenosu jednoho kanálu**

Vláknu použitému pro přenos je jako parametr předána struktura, jež obsahuje informace o přenosu a návratovou hodnotu vlákna. Informace o přenosu obsahují název souboru, do kterého mají být data zapsána, číslo použitého kanálu a požadované množství dat. Ve vlákne je nejprve alokována potřebná paměť a následně otevřeno patřičné zařízení spolu s výstupním souborem. Poté je z otevřeného zařízení čteno, dokud není zaplněn alokovaný buffer. V případě, že přenos proběhl bez chyby, jsou data zapsána do výstupního souboru. Následně je nastavena návratová hodnota a vlákno je ukončeno. Vývojový diagram na obrázku 75 popisuje funkci vlákna.



**Obrázek 74. Blokové schéma inicializace PCIe přenosu obou kanálů**

Maximální množství dat, jež ovladač umožňuje zároveň přenést, je 255krát velikost paměťové stránky. To je v praxi značně nevyhovující, avšak pro účel otestování funkčnosti designu a ověření integrity dat postačující.



Obrázek 75. Blokové schéma vlákna pro přenos a uložení dat

Ukázka aplikace je na obrázcích 76, 77, 78 a 79. Obrázek 77 demonstruje její chování po zapnutí, kdy jsou zobrazeny dostupné registry s jejich popisem a nápověda. Kromě toho je zkontrolováno, že je karta připojena a ovladač vytvořil potřebná zařízení. Na obrázku 78 je znázorněno nastavení triggeru, v tomto případě na vzestupnou hranu na kanálu 1 s úrovní 9000. Iniciace přenosu z obou kanálů, tedy synchronně, je ilustrováno obrázkem 79. Na obrázku 76 zobrazeny soubory s přenesenými daty, která si navzájem časově odpovídají.

```

linux@linux:~/projects/cecilie/cecilie_pcie/results$ ls -l
total 240
-rw-r--r-- 1 linux linux 102400 May 25 08:54 capture_ch0
-rw-r--r-- 1 linux linux 102400 May 25 08:54 capture_ch1
-rwxr-xr-x 1 linux linux 8928 Apr 24 09:30 checker
-rw-r--r-- 1 linux linux 1259 Apr 24 09:30 checker.c
-rwxr-xr-x 1 linux linux 125 Apr 28 08:12 conv_cpy.sh
-rwxr-xr-x 1 linux linux 9016 Apr 27 05:35 convert
-rw-r--r-- 1 linux linux 1210 Apr 27 05:35 convert.c
-rw-r--r-- 1 linux linux 134 Apr 27 05:28 makefile
  
```

Obrázek 76. Výsledky přenosu dat z karty do PC přes sběrnici PCIe

```
linux@linux:~/projects/cecilie/cecilie_pcie$ ./pcie_app
*** Cecilie PCI Express application ***

Description of Cecilie PCIE registers:
CH0_CONTROL 0x00000000 0 - run, 1 - clear
CH1_CONTROL 0x00000004 0 - run, 1 - clear
COMMON_CONTROL 0x00000008 0 - sync, 1 - trigger, 2 - clear, 3 - trigger channel, 4 - trigger edge, 18:5 - trigger level
CH0_STATUS0 0x0000000c 0 - overflow
CH0_STATUS1 0x00000010 31:0 - word counter lsb
CH0_STATUS2 0x00000014 31:0 - word counter msb
CH1_STATUS0 0x00000018 0 - overflow
CH1_STATUS1 0x0000001c 31:0 - word counter lsb
CH1_STATUS2 0x00000020 31:0 - word counter msb
COMMON_STATUS 0x00000024 0 - overflow
REG_ID 0x00000028 31:0 - 0xDFC0CEC0

Application Usage
-----
'h' - Show this help
's' - Show registers
'w' - Write to register
'r' - Read from register
't' - Transfer data
'i' - Trigger settings
'e' - Exit

Checking if devices exists ...
OK
Choose option ('h' for help):
```

Obrázek 77. Aplikace pcie\_app po spuštění.

```
Choose option ('h' for help): i
Enable trigger? (0 off / 1 on): 1
Enabling trigger
Trigger channel (0 or 1 - default 0): 1
Trigger edge ('r' or 'f' - default 'r'): r
Trigger level (0 to 16383 - default 8191): 9000
Trigger settings:
  enable: 1
  channel: 1
  edge: rising
  level: 9000
Choose option ('h' for help):
```

Obrázek 78. Aplikace pcie\_app při nastavování triggeru

```
Choose option ('h' for help): t
Transfer size (modulo 1024, max 1044480): 102400
Results file name: capture
Choose channel (0 - fist, 1 - second, b - both): b
rw_register reg_addr: 00000008 write_val: 00000005 read_val: (nil)
rw_register reg_addr: 00000008 write_val: 00000001 read_val: (nil)
transfer_both_channel Transfer on channel 0 SUCCEED
transfer_both_channel Transfer on channel 1 SUCCEED
rw_register reg_addr: 00000008 write_val: 00000000 read_val: (nil)
Choose option ('h' for help):
```

Obrázek 79. Aplikace pcie\_app při přenosu dat z karty do PC



## 10 VÝSLEDKY

V této kapitole budou prezentovány výsledky měření a ověření integrity všech rozhraní.

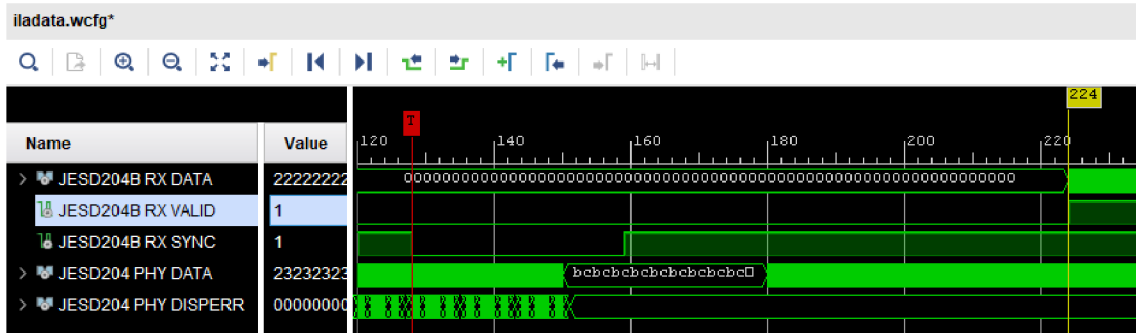
### 10.1 Ověření integrity dat rozhraní JESD204B

Integrita dat přenášených rozhraním JESD204B se dá ověřit ve dvou bodech. Prvním bodem je integrita samotných linek. Z důvodu špatného vedení linky, její nadměrné kapacity, špatného zakončení, přeslechů, nebo jiných problémů může být signál nekvalitní. Převodník umožňuje generování několika různých testovacích vzorů, jejichž účelem je kontrola bezchybnosti přenosu. Testovací vzory použité pro věření integrity rozhraní jsou PRBS (Pseudo Random Bit Sequence), rampa a uživatelem nastavitelný vzor. PRBS o délce 32 bitů zajistí co největší podobnost náhodnému signálu a bylo použito k otestování stavu linky. Na straně FPGA byl pro kontrolu správnosti použit IBERT (Integrated Bit Error Ration Test). IBERT je nástrojem pro kontrolu transceiverů, umožňující monitorování stavu linek a jejich nastavování přes JTAG – tedy z vývojového prostředí Vivado. Tento test proběhl v pořádku a neodhalil na linkách žádné chyby. Obrázek 80 demonstruje výsledky testování.

Link Group	Link	Rate	Error Rate	BER	Reset	PRBS	Reset	Status
Link Group 0 (8)	Link 0	15.022 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 1	15.000 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 2	15.000 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 3	15.008 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 4	15.000 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 5	15.000 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 6	15.000 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked
	Link 7	15.021 Gbps	1.915E13	0E0	5.222E-14	Reset	PRBS 31-bit	Locked

Obrázek 80. Kontrola PRBS IBERTem na JESD linkách

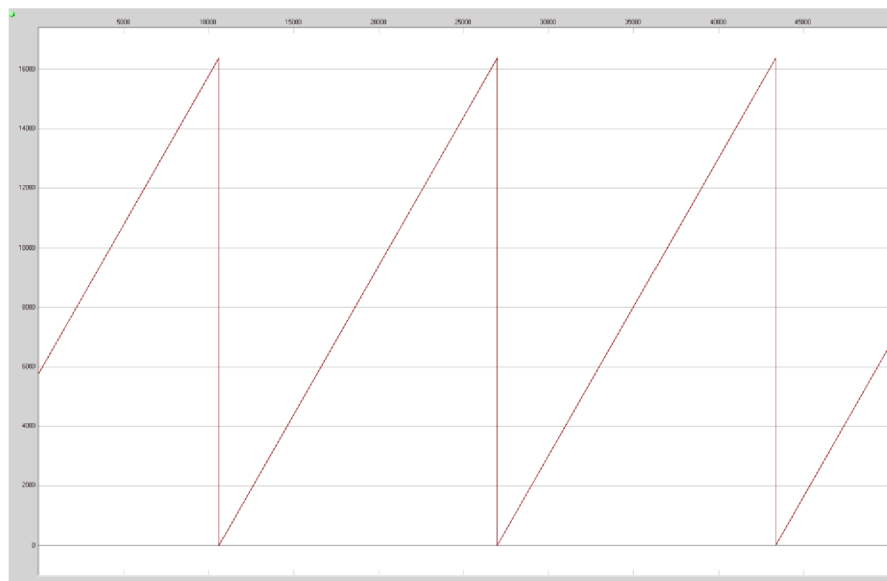
Druhým bodem testování je ověření správnosti samotných dat, přijatých rozhraním JESD204B po dokončení všech inicializačních fází. Na obrázku 81 lze vidět průběh od počátku inicializace až po vystavení validních dat na výstupu komponenty *JESD204B RX*. Z obrázku je patrné, že byl SYNC deaktivován až poté, co byly všechny linky bez chyby disparity a detekovaly K28.5 znaky (mající hodnotu 0xBC). Správnost dat zde byla zkontrolována vizuálně pomocí nástroje ILA (Integrated Logic Analyzer) za použití uživatelsky definovaného testovacího vzoru.



Obrázek 81. Inicializační fáze JESD přenosu zachycená logickým analyzátozem

## 10.2 Ověření integrity dat rozhraní 100G Ethernet a PCI Express

Převodník nabízí kromě testovacích vzorů pro ověření linek také testovací vzory pro ověření správnosti dat. Vhodným vzorem pro kontrolu správnosti dat je rampa. V podstatě se jedná o 14bitový čítač, který svou hodnotu inkrementuje o 1 s každým vzorkem. Přijatá data je tedy možné jednoduše zkontrolovat a odhalit tak chyby v transformaci nebo způsobu přenosu. Za tímto účelem byl napsán program *checker*, který data z přijatého souboru vyčte a zkontroluje. Průběhy byly zároveň vizualizovány programem VisualAnalog od společnosti Analog Devices, viz obrázek 82. Data však musela být nejprve konvertována z bitového formátu do textového, k čemuž byla vytvořena aplikace *converter*. Tímto způsobem byla ověřena integrity dat přenášených rozhraními 100G Ethernet i PCI Express.



Obrázek 82. Pilovitý průběh testovacího vzoru převodníku

## 10.3 Měření

Cílem měření bylo zjistit chování převodníkové karty a ověřit korektnost převedeného signálu. V průběhu bylo zjištěno několik problémů, které však byly buďto opraveny, nebo byla alespoň nalezena jejich příčina.

Při použití referenčního 10MHz hodinové signálu pro PLL1 v HMC7044 nebylo možné dosáhnout zavěšení. Po měření osciloskopem bylo zjištěno, že napěťový rozsah kontrolního napětí CP (Charge Pump) je nedostačující a z toho důvodu není možné správně naladit VCXO. Proto byl místo referenčního hodinového signálu z generátoru použit jeho výstup a postupným snižováním frekvence z 10 MHz bylo nakonec dosaženo zavěšení. V průběhu bylo zjištěno, že je VCXO nadměrně citlivé na teplotu a její výkyvy. S největší pravděpodobností se jedná o poškození součástky špatným pájením, neboť po její výměně už vše fungovalo správně a bylo možné se na zavěsit 10 MHz referenci generátoru.

Po zjištění příčiny nezavěšení PLL1 a opravě chyby se jako referenční použil výstupní signál z generátoru. Při vstupní frekvenci 10 MHz byla, vzhledem k nastavení HMC7044, vzorkovací frekvence 3 GHz. Ve chvíli, kdy je přesně známa vzorkovací frekvence, je možné vzorkovat signál koherentně. Koherentní vzorkování zaručuje, že dochází k menšímu až nulovému zkreslení frekvenčního spektra. Aby bylo vzorkování koherentní, musí být vztah vzorkovací frekvence ( $f_s$ ) a frekvence vstupního signálu ( $f_{in}$ ) podle vztahu 14 [29]. Počet period  $M$  by měl být celé číslo a počet vzorků  $N$  mocnina dvou. Tím je zaručeno, že je na periodu vzorkovaného signálu konstantní a celočíselný počet vzorků. Pro zjištění kvality vstupního signálu, který před vstupem do převodníku prochází úpravami, byla vypočtena frekvence pro zajištění koherentního vzorkování. Výsledné spektrum signálu mohlo být poté porovnáno s hodnotami naměřenými spektrálním analyzátozem.

$$f_s = 3000 \text{ MHz}$$

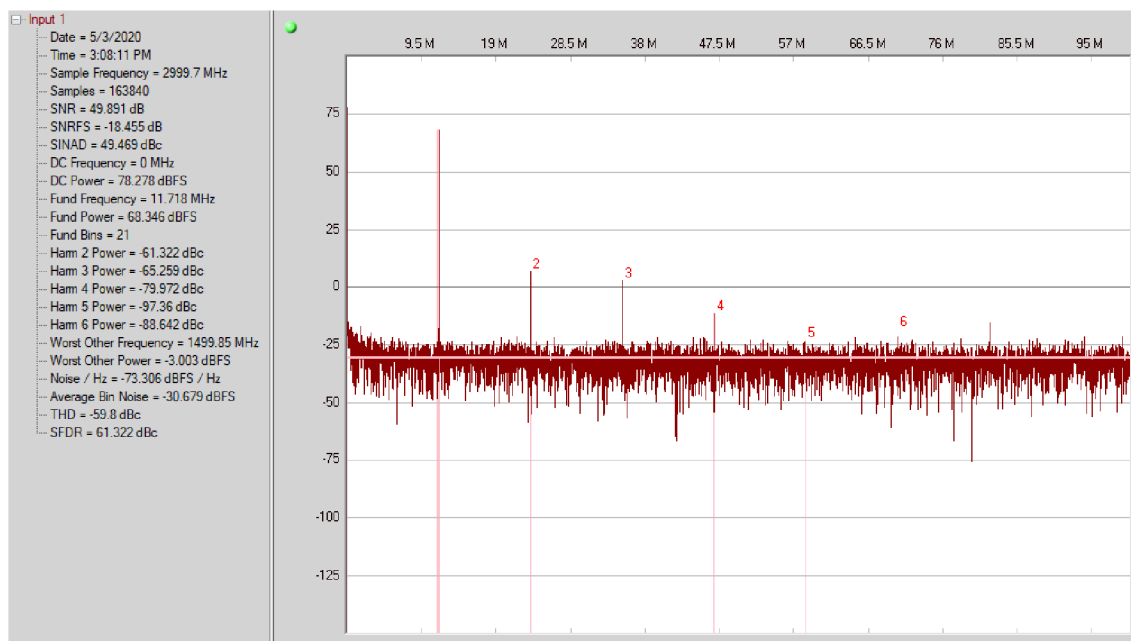
$$M = 512$$

$$N = 131072$$

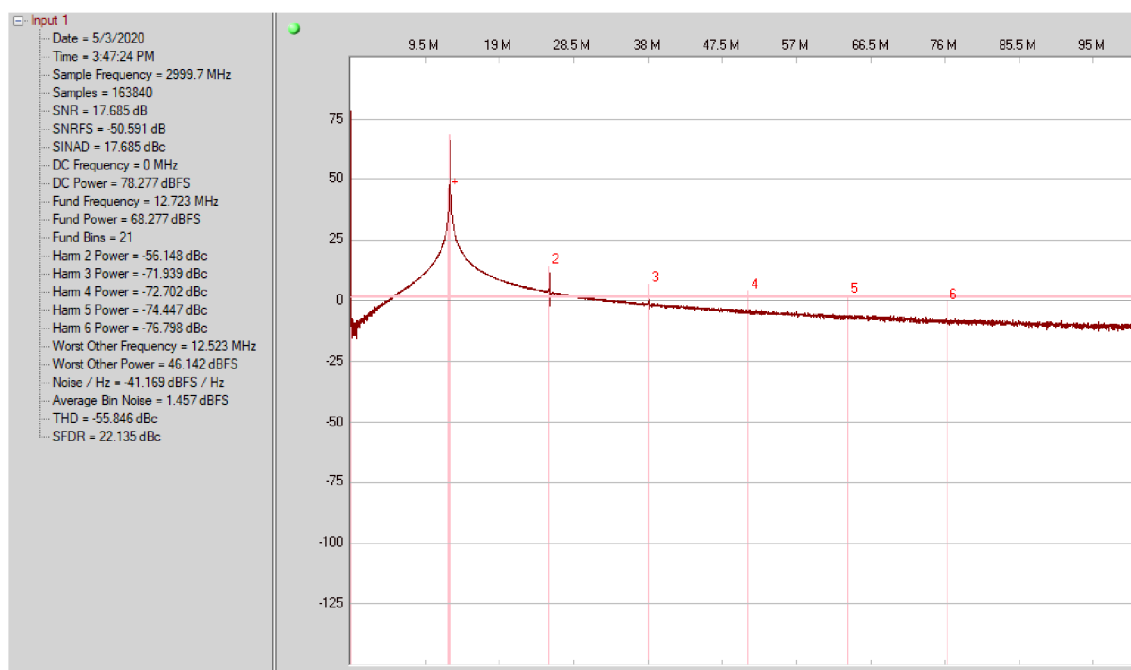
$$\frac{f_{in}}{f_s} = \frac{M}{N} \Rightarrow f_{in} = \frac{f_s M}{N} = \frac{3000 \cdot 512}{131072} = 11,71875 \text{ MHz} \quad (14)$$

Vzorkovací frekvence je již známá. Čím větší bude počet vzorků  $N$ , tím větší bude následně rozlišení ve frekvenčním spektru počítaném DFT (Diskrétní Fourierova Transformace). Zvolená hodnota  $N$  je 131 072. Následně se zvolí  $M$ , a to nejlépe iterativním postupem podle toho, v jakých mezích se má frekvence měřeného signálu  $f_{in}$  nacházet. Pro měření v rámci této práce je použita frekvence

$f_{in}$  11,71875 MHz, čemuž odpovídá 512 period na zvolný počet vzorků. Na následujícím obrázku 83 lze vidět frekvenční spektrum takto změřeného signálu. Pro porovnání koherentního a nekoherentního vzorkování bylo provedeno měření i s rozdílnými vstupními frekvencemi. To je demonstrováno obrázkem 84, při odchylce 1 MHz od výše zvolené frekvence.

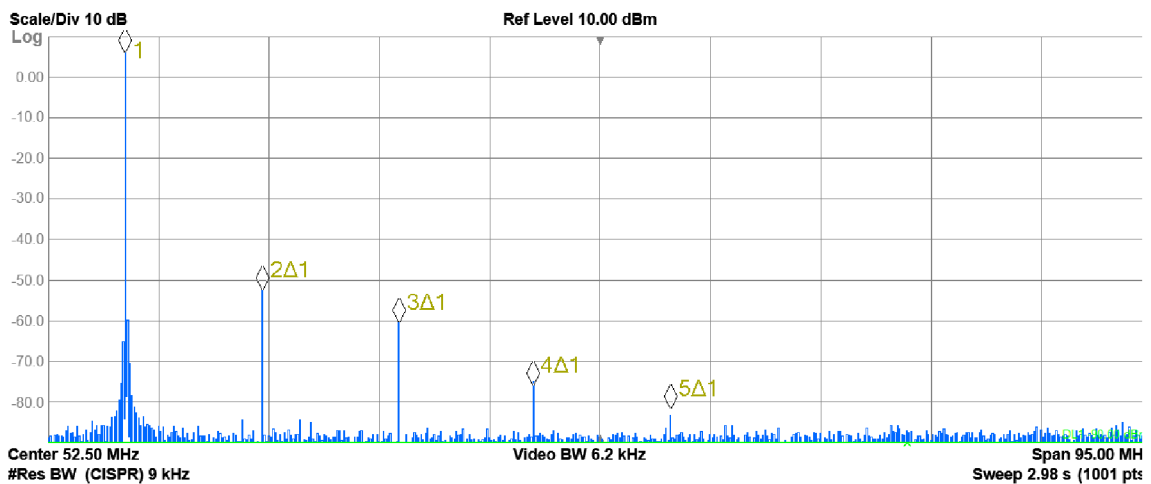


Obrázek 83. Frekvenční spektrum signálu při koherentním vzorkování



Obrázek 84. Frekvenční spektrum signálu při nekoherentním vzorkování

Frekvenční spektrum signálu získaného AD převodníkem může být porovnáno s frekvenčním spektrem získaným spektrálním analyzátozem. Z porovnání obrázků 83 a 85 lze vidět, že se získaná spektra zásadně neliší a vstupní signál tedy není vstupními obvody převodníkové karty významně změněn. Výčet nejvýraznějších harmonických frekvencí nalezených spektrálním analyzátozem je na obrázku 86.



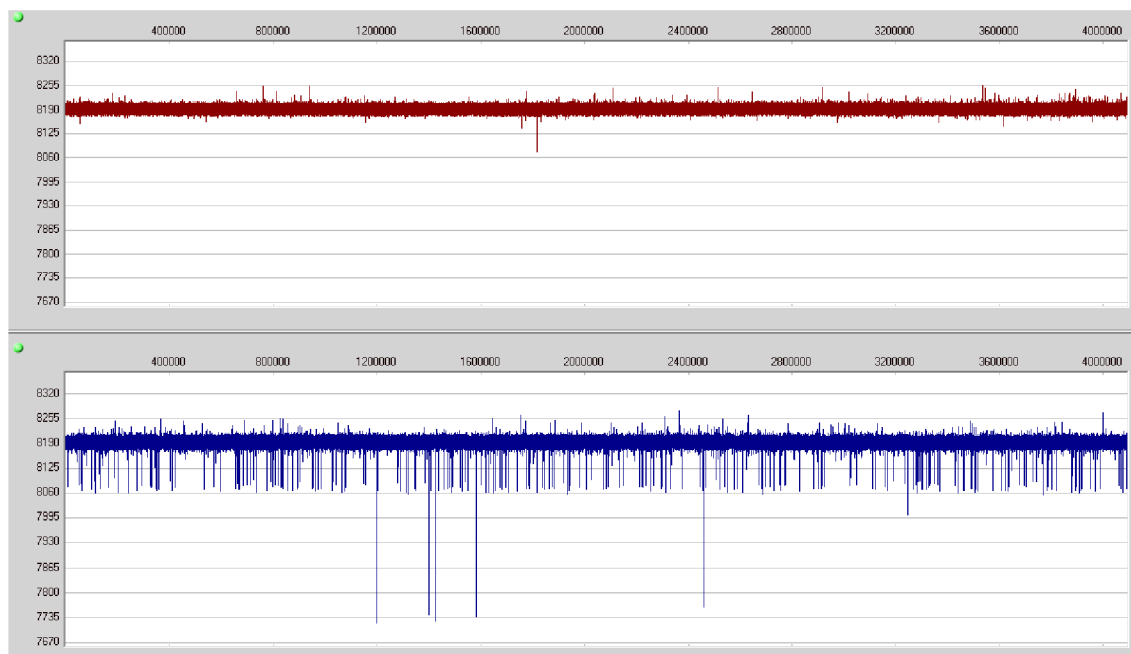
Obrázek 85. Frekvenční spektrum signálu získané spektrálním analyzátozem

	Mode	Trace	Scale	X	Y
1	N	1	f	11.650 MHz	5.926 dBm
2	Δ1	1	f (Δ)	11.780 MHz (Δ)	-58.58 dB
3	Δ1	1	f (Δ)	23.560 MHz (Δ)	-66.41 dB
4	Δ1	1	f (Δ)	35.150 MHz (Δ)	-81.15 dB
5	Δ1	1	f (Δ)	46.930 MHz (Δ)	-88.83 dB
6					
7					

Obrázek 86. Velikosti významných složek frekvenčního spektra získané spektrálním analyzátozem

Pro nalezení aditivního šumu převodníkové karty byl její vstup zakončen 50 Ω. Následným měřením byly objeveny velké odchylky od jinak konstantní úrovně očekávaného šumu. Tyto odchylky byly řádově o velikosti až 9 bitů rozsahu a každá měla trvání přesně jeden vzorek. Šum takového charakteru je fyzikálně nereálný. Nakonec bylo zjištěno, že se jedná o vlastnost převodníku. Chyba se nazývá sparkle code a její příčinou je meta stabilita mezi jednotlivými úrovněmi převodníku [30]. Četnost a velikost těchto chyb může být snížena použitím nižší

vzorkovací frekvence a strmějšího vzorkovaného signálu (větší amplituda nebo frekvence). Na obrázku 87 je pro srovnání signál ze zakončeného vstupu, vzorkovaný s frekvencí 2,5 GHz a 3 GHz.

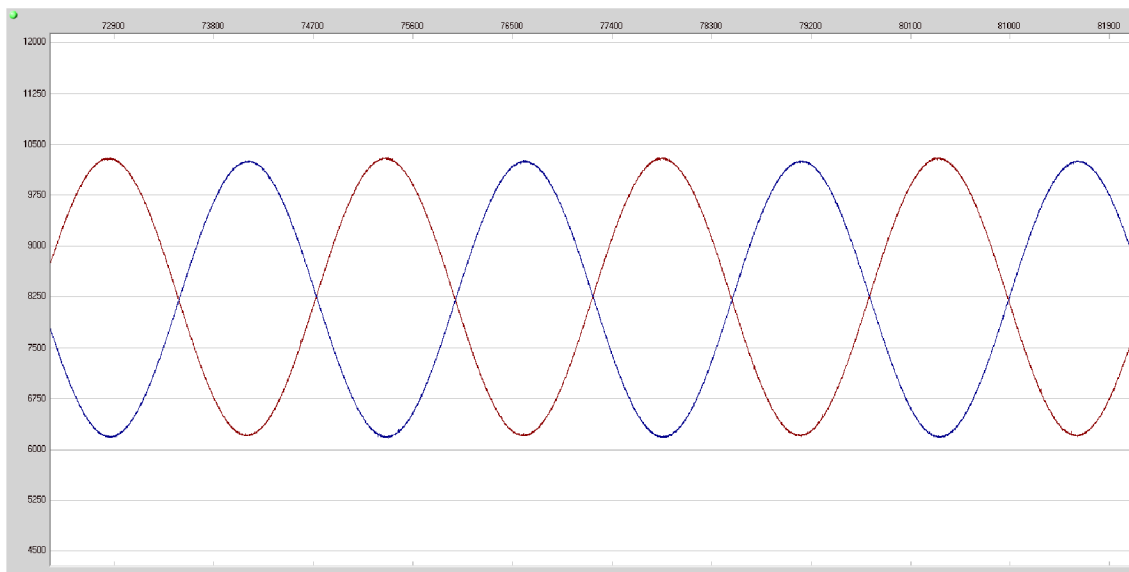


**Obrázek 87. Sparkle code error**

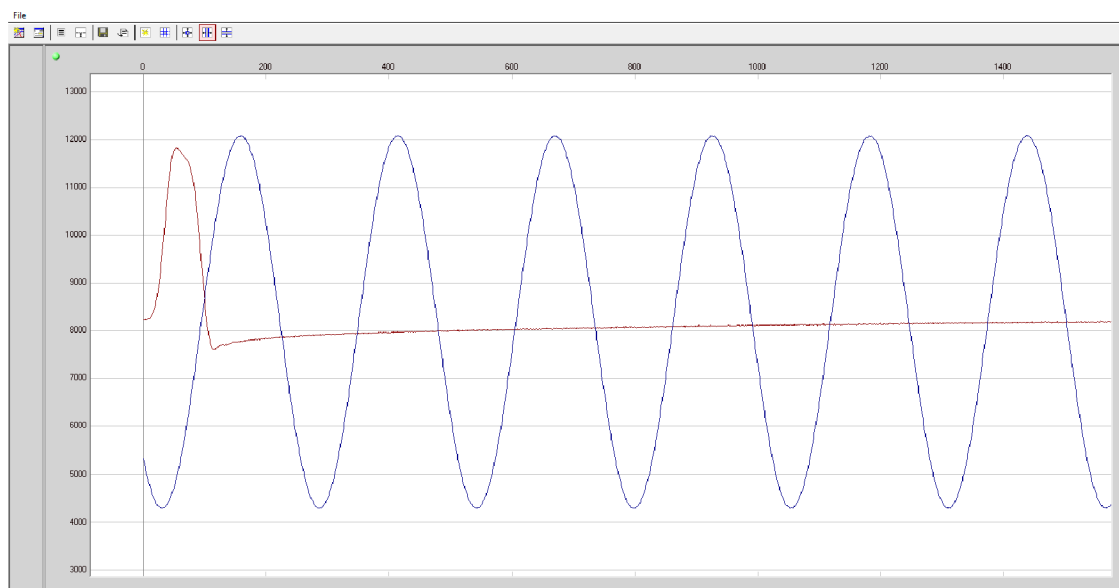
Následující měření se týká především ověření funkčnosti některých prvků designu, jako je synchronizace dvou kanálů a trigger, a zajímavostí jako aliasing nebo multitónový signál.

Synchronizace je nutná, aby bylo zaručeno, že dva vzorky uložené v paměti na stejných pozicích odpovídají stavu signálu ve stejný okamžik. Obrázek 88 zobrazuje průběh stejného signálu změřeného oběma kanály. Kanály jsou vůči sobě invertovány z důvodu opačného zapojení transformátoru, jež ze vstupního single ended signálu dělá signál diferenciální. Důvodem je zjednodušení vedení signálu na DPS.

Velmi užitečným a často používaným nástrojem při měření je trigger. Ten může být nastaven na jakoukoliv hodnotu v rozmezí 14 bitů na libovolný z kanálů. Je tak možné získat pouze žádanou část průběhu měřeného signálu a není potřeba měřit na příliš dlouhém časovém intervalu a ukládat obrovské množství nepotřebných dat. Ukázka triggerování na nástupnou hranu pulzu je na obrázku 89.



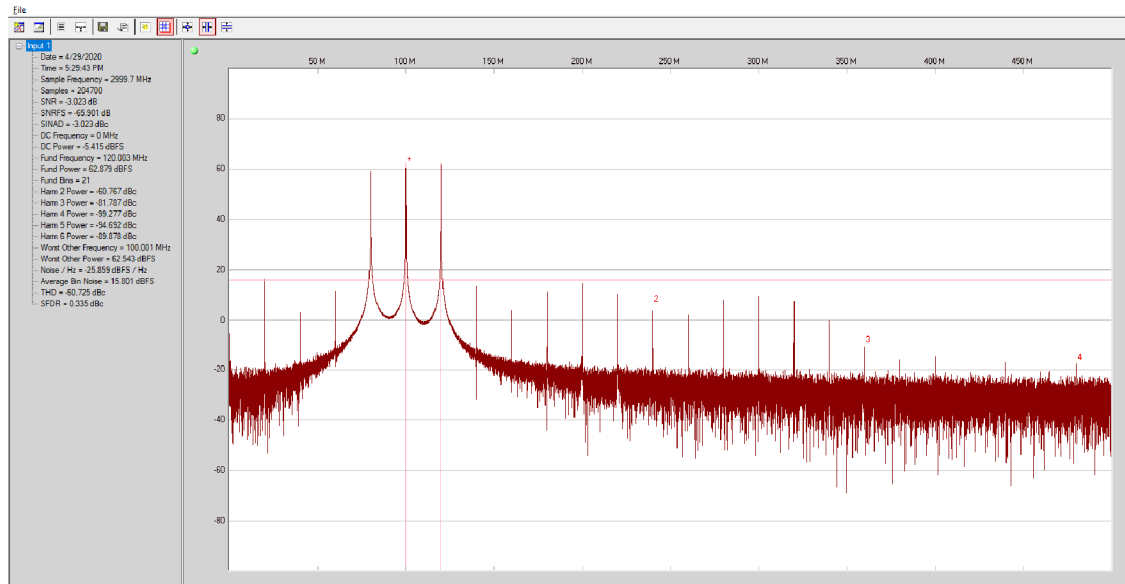
**Obrázek 88. Současné měření na obou kanálech**



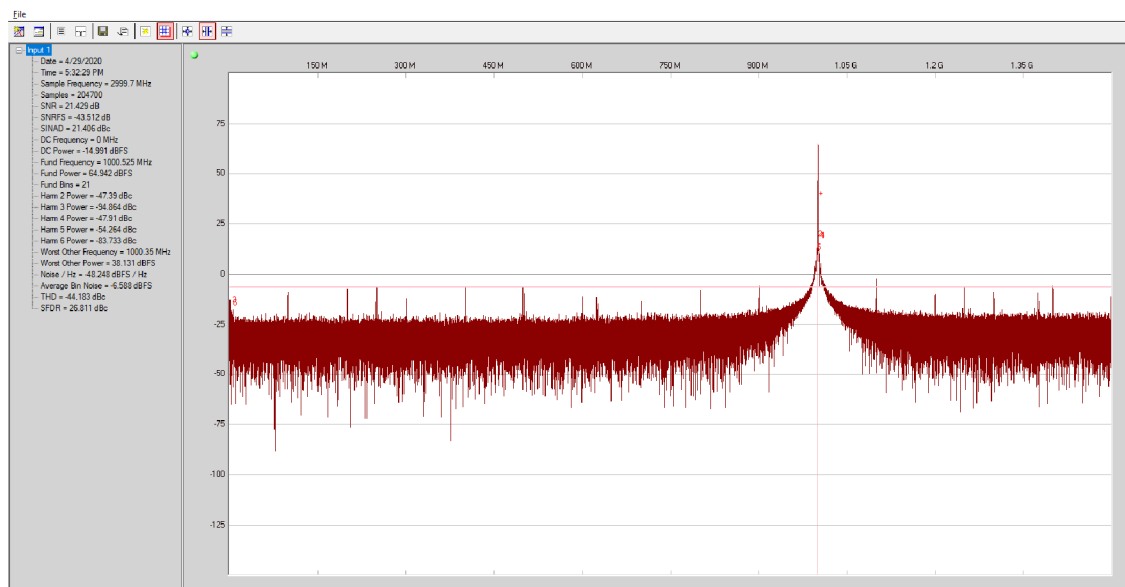
**Obrázek 89. Demonstrace použití triggeru**

K dispozici je generátor schopný generovat frekvence do 6 GHz a vytvářet modulovaný multitónový signál. Toho bylo využito pro vygenerování signálu skládajícího se z 80, 100 a 120MHz složek. Po změření a následném výpočtu frekvenčního spektra tohoto signálu byl pro ilustraci vytvořen obrázek 90. Zde je možné vidět tři špičky patřící jednotlivým složkám a jejich harmonické frekvence. Mimo multitón byl také pro zajímavost vyzkoušen aliasing, a to při měření signálu o frekvenci 4 GHz. Výsledek takového měření je na obrázku 91. Frekvence 4 GHz se zde vlivem aliasingu jeví jako 1 GHz. Takovým způsobem lze měřit signály

s frekvencí vyšší, než je frekvence vzorkování. Vždy je však zapotřebí vědět, ve které Nyquistově zóně se nacházíme. Výsledné spektrum by v opačném případě mohlo patřit také signálu o frekvenci  $f_{in} = kf_s \pm f$ , kde  $k$  je celé číslo,  $f_s$  frekvence vzorkování a  $f$  ve spektru nalezená frekvence.



Obrázek 90. Frekvenční spektrum multi tónového signálu

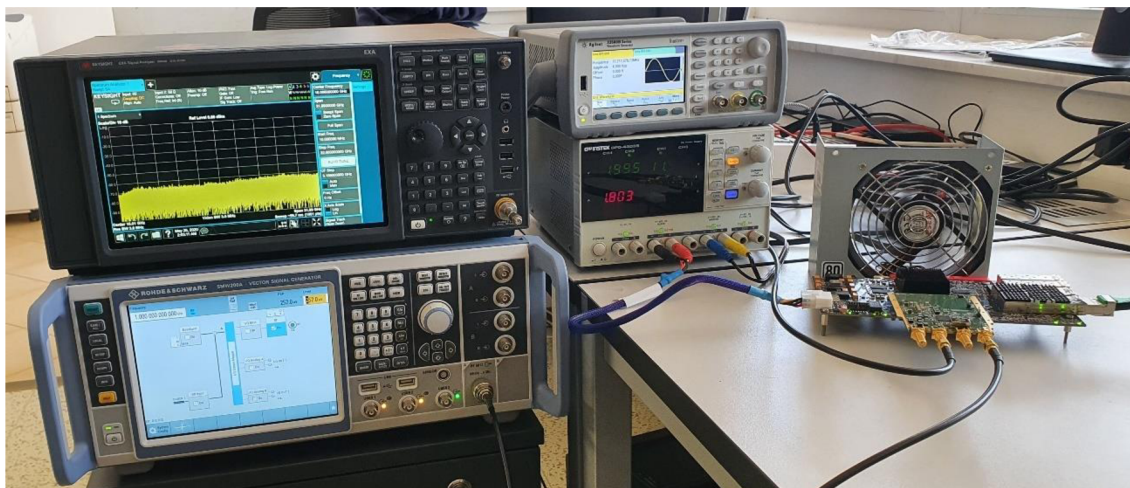


Obrázek 91. Frekvenční spektrum při aliasingu

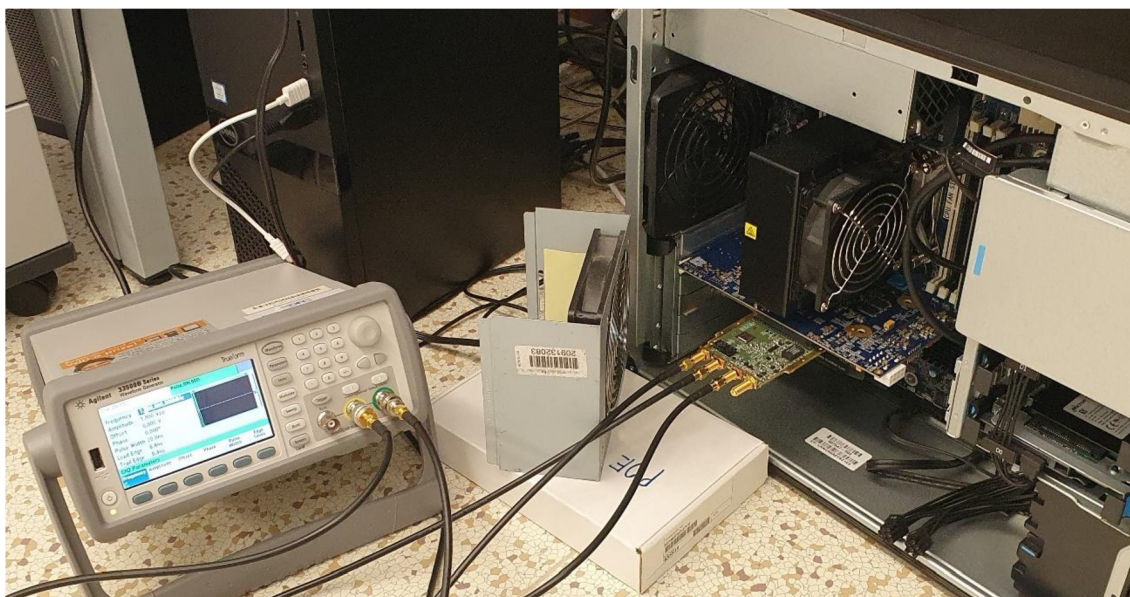


Soustava zařízení použitá za účelem generování vstupního signálu a jeho analýzy je ilustrována obrázkem 92. Na obrázku 93 je demonstrováno zapojení karty při měření s triggerem. Použitými přístroji jsou:

- Spektrální analyzátor Keysight N9010B EXA
- Vektorový generátor Rohde & Schwarz SMW200A
- Generátor signálu Agilent 33500B
- Napěťový DC zdroj GWInstek GPD-4303S



**Obrázek 92. Soustava použitá pro měření a analýzu signálu**



**Obrázek 93. Soustava použitá pro měření s triggerem**

## 11 ZÁVĚR

Teoretická část práce se věnuje představení použitých technologií. První kapitola je věnována AD převodníkům, jejich parametrům, chybám a různým technikám jejich použití. Následuje popis rozhraní JESD204B, kterým jsou data z AD převodníku přenášena do FPGA. Více pozornosti je zde věnováno principu synchronizace zařízení a dosažení deterministické latence. Poté jsou zde rozebrána dvě rozhraní, která jsou podporována použitým hardwarem a umožňují přenos dat s rychlostí 100 Gbps a vyšší. Prvním z těchto dvou rozhraní je 100gigabitová verze Ethernetu. Nejvhodnějším pro přenos takto velkého množství dat je protokol RoCE, který při použití kompatibilní síťové karty umožňuje zápis přímo do paměti cílového zařízení bez zbytečného přetěžování procesoru. Druhým rozhraním je PCI Express, které taktéž umožňuje zápis přímo do paměti a zároveň nabízí o čtvrtinu vyšší maximální přenosovou rychlost.

V úvodu praktické části je představen hardware, kterým je PCIe karta Cecile a aditivní převodníkový modul od firmy DFC Design. PCIe karta nabízí, jak je výše zmíněno, dvě rozhraní pro přenos dat rychlostí 100 Gbps, a to 100G Ethernet a PCIe. Dále obsahuje také 10G Ethernetové rozhraní, 8 GB DDR4 paměti a FPGA řady Kintex Ultrascale Plus. Následně je proveden rozbor řešení, kde jsou vypočítány propustnosti rozhraní JESD204B, 100G Ethernet a PCI Express. Bylo zjištěno, že veškerá data z převodníku mohou být bez úpravy přenesena do PC pouze pomocí rozhraní PCIe. Rozhraním 100G Ethernet by mohla být přenesena data jen z jednoho ze dvou kanálů převodníku. Pro přenos dat z obou kanálů by musela být tato rozhraní použita dvě.

Do FPGA byla implementována rozhraní JESD204B, 100G Ethernet a PCI Express. Pro implementaci rozhraní JESD204B je použito JESD204 PHY od společnosti Xilinx, tvořící fyzickou vrstvu standardu, a JESD204B RX od firmy DFC Design, tvořící jeho vyšší vrstvy. JESD204B RX bylo zapotřebí před použitím z důvodu drobné chyby upravit.

Pro přenos dat rozhraním 100G Ethernet byly pro FPGA navrženy a napsány moduly realizující UDP protokol, RoCE protokol a ovládání pomocí UDP registrů. Použití RDMA vyžadovalo vytvoření softwaru, jehož účelem je inicializace přenosu a uložení přijatých dat. Pro rozhraní PCIe byl vytvořen modul, který kromě přenosu dat umožňuje použít funkce jako trigger či synchronizace kanálů. Tyto funkce mohou být konfigurovány zápisem do registrů přístupných přes BAR. Jako ovladač k zařízení byl využit volně dostupný příklad výrobce. Stejně jako v případě RDMA bylo zapotřebí vytvořit software, jehož prostřednictvím se provede konfigurace a inicializace přenosu. V rámci této práce bylo napsáno

přibližně 4 500 řádků kódu v jazyce C a 10 000 řádků kódu v jazyce VHDL, jak je demonstrováno tabulkami 2 a 6.

Byla ověřena integrita dat všech použitých rozhraní. V průběhu testování byla zjištěna abnormální citlivost VCXO na okolní vlivy, zejména na teplotu. Po výměně součástky toto nestandardní chování již neprojevovalo, přičemž nejpravděpodobnější příčinou bylo poškození, způsobené nevhodným pájecím profilem. Kromě kontroly integrity proběhla také řada dalších měření, testujících parametry převodníkové karty. Bylo provedeno koherentní měření signálu a následné porovnání jeho spektra se spektrem získaným spektrálním analyzátozem. Při měření se vstupem zakončeným  $50 \Omega$  byly objeveny velké špičky o hodnotách až 9 bitů z rozsahu. Po zkoumání tohoto jevu bylo zjištěno, že se jedná o tzv. sparkle code error, který se projevuje především při vyšších vzorkovacích frekvencích a malé strmosti měřeného signálu. Po snížení vzorkovací frekvence z 3 GHz na 2,5 GHz došlo k velkému zlepšení, které může být pozorováno na obrázku 87. Dále byly také otestovány funkce synchronizace a triggeru, jejichž funkčnost je demonstrována obrázky 88 a 89.

Převodníková karta byla netestovaným prototypem, a z toho důvodu se na ni nacházely chyby v zapojení. Konkrétně se jednalo o špatně zapojené převodníky napěťových úrovní. Chyby byly díky vhodně navrženému testovacímu designu brzy identifikovány a opraveny.

Práce by mohla pokračovat implementací některé formy komprese nebo decimace dat. To by vedlo ke snížení množství přenášených dat a menšímu zatížení celého systému. Mimo decimaci nebo kompresi by také mohlo být v FPGA v závislosti na cílové aplikaci vykonáváno zpracování a vyhodnocení vzorkovaného signálu.

## 12 LITERATURA

- [1] Atmel, „AVR127: Understanding ADC Parameters,“ 2016. [Online]. Dostupné z: [http://ww1.microchip.com/downloads/en/appnotes/atmel-8456-8-and-32-bit-avr-microcontrollers-avr127-understanding-adc-parameters\\_application-note.pdf](http://ww1.microchip.com/downloads/en/appnotes/atmel-8456-8-and-32-bit-avr-microcontrollers-avr127-understanding-adc-parameters_application-note.pdf).
- [2] Texas Instruments, „High-Speed, Analog-to-Digital Converter Basics,“ 2011. [Online]. Dostupné z: <http://www.ti.com/lit/an/slaa523a/slaa523a.pdf?ts=1589797921133>.
- [3] Texas Instruments, „Jitter vs SNR for ADCs,“ 2017. [Online]. Dostupné z: <https://training.ti.com/sites/default/files/docs/TIPL%204704%20-%20Jitter%20vs%20SNR.pdf>. [Přístup získán 2020].
- [4] W. Kester, „ADC Input Noise: The Good, The Bad, and The Ugly. Is No Noise Good Noise?,“ 2006. [Online]. Dostupné z: <https://www.analog.com/media/en/analog-dialogue/volume-40/number-1/articles/adc-input-noise.pdf>. [Přístup získán 2020].
- [5] Analog Devices, „JESD204 Survival Guide,“ 2014. [Online]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/technical-articles/JESD204B-Survival-Guide.pdf>.
- [6] JEDEC, „JESD204B,“ 2011.
- [7] Xilinx, „UltraScale Architecture GTY Transceivers,“ 2017. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/user\\_guides/ug578-ultrascale-gty-transceivers.pdf](https://www.xilinx.com/support/documentation/user_guides/ug578-ultrascale-gty-transceivers.pdf). [Přístup získán 2020].
- [8] R. Ivry, „FEC and Line Coding for EFM,“ 2001. [Online]. Dostupné z: [http://www.ieee802.org/3/efm/public/sep01/ivry\\_1\\_0901.pdf](http://www.ieee802.org/3/efm/public/sep01/ivry_1_0901.pdf). [Přístup získán 2020].
- [9] JEDEC, „JESD204C,“ 2017.
- [10] Information Sciences Institute University of Southern California, „Internet Protocol - RFC 791,“ 1981. [Online]. Dostupné z: <https://tools.ietf.org/html/rfc791>.
- [11] Postel J., „User Datagram Protocol - RFC 768,“ 1980. [Online]. Dostupné z: <https://tools.ietf.org/html/rfc768>.
- [12] Infiniband Trade Association, „RDMA over Converged Ethernet (RoCE),“ 2010. [Online]. Dostupné z: <https://cw.infinibandta.org/document/dl/7781>.
- [13] Infiniband Trade Association, „Infiniband Architecture Specification volume 1,“ 2015. [Online]. Dostupné z:

<https://cw.infinibandta.org/document/dl/8567>.

- [14] Intel, „Introduction to PCI Express,“ 2003.
- [15] PCI-SIG, „PCI Express Base Specification Revision 3.0,“ 2010. [Online]. Dostupné z: <http://www.lttconn.com/res/lttconn/pdres/201402/20140218105502619.pdf>.
- [16] ARM, „AMBA® 4 AXI4-Stream Protocol,“ 2010. [Online]. Dostupné z: [https://static.docs.arm.com/ihi0051/a/IHI0051A\\_amba4\\_axi4\\_stream\\_v1\\_0\\_protocol\\_spec.pdf](https://static.docs.arm.com/ihi0051/a/IHI0051A_amba4_axi4_stream_v1_0_protocol_spec.pdf). [Přístup získán 2020].
- [17] Xilinx, „Integrated 100G Ethernet Subsystem,“ 2018. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/cmac\\_usplus/v2\\_4/pg203-cmac-usplus.pdf](https://www.xilinx.com/support/documentation/ip_documentation/cmac_usplus/v2_4/pg203-cmac-usplus.pdf).
- [18] Xilinx, „DMA/Bridge Subsystem for PCI Express,“ 2019. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/xdma/v4\\_1/pg195-pcie-dma.pdf](https://www.xilinx.com/support/documentation/ip_documentation/xdma/v4_1/pg195-pcie-dma.pdf).
- [19] Xilinx, „AXI Reference Guide,“ 2017. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_ref\\_guide/latest/ug1037-vivado-axi-reference-guide.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/latest/ug1037-vivado-axi-reference-guide.pdf). [Přístup získán 2020].
- [20] Analog Devices, „ADI-SPI Serial Control Interface Standard,“ [Online]. Dostupné z: [https://wiki.analog.com/\\_media/resources/technical-guides/adispi\\_rev\\_1p0\\_customer.pdf](https://wiki.analog.com/_media/resources/technical-guides/adispi_rev_1p0_customer.pdf).
- [21] Xilinx, „AXI Quad SPI,“ 2019. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_quad\\_spi/v3\\_2/pg153-axi-quad-spi.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_quad_spi/v3_2/pg153-axi-quad-spi.pdf).
- [22] Xilinx, „JESD204 PHY,“ 2017. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/jesd204\\_phy/v4\\_0/pg198-jesd204-phy.pdf](https://www.xilinx.com/support/documentation/ip_documentation/jesd204_phy/v4_0/pg198-jesd204-phy.pdf).
- [23] Xilinx, „MicroBlaze Micro Controller System,“ 2019. [Online]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/microblaze\\_mcs/v3\\_0/pg116-microblaze-mcs.pdf](https://www.xilinx.com/support/documentation/ip_documentation/microblaze_mcs/v3_0/pg116-microblaze-mcs.pdf).
- [24] Analog Devices, „HMC7044 datasheet,“ 2016. [Online]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/HMC7044.pdf>.
- [25] Analog Devices, „AD9208 datasheet,“ 2017. [Online]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9208.pdf>.
- [26] Xilinx, „AXI IIC Bus Interface,“ 2016. [Online]. Dostupné z:

[https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_iic/v2\\_0/pg090-axi-iic.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_iic/v2_0/pg090-axi-iic.pdf).

- [27] Microsoft, „Microsoft Network Direct SPI documentation,“ 2020. [Online]. Dostupné z: [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/cc904397\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/cc904397(v=vs.85)). [Přístup získán 2020].
- [28] D. Barak, 2020. [Online]. Dostupné z: <https://www.rdmamojo.com/>. [Přístup získán 2020].
- [29] Analog Devices, „Coherent sampling calculator,“ [Online]. Dostupné z: <https://www.maximintegrated.com/en/design/technical-documents/app-notes/3/3190.html>. [Přístup získán 2020].
- [30] Kester, Walt, „Find Those Elusive ADC Sparkle Codes and Metastable States,“ 2008. [Online]. Dostupné z: <https://www.analog.com/media/en/training-seminars/tutorials/MT-011.pdf>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

FPGA	Field Programmable Gate Array
ASIC	Application Specific Integrated Circuit
AD	Analog Digital
SPI	Serial Peripheral Interface
PCIe	Peripheral Component Interconnect Express
MAC	Medium Access Control
CMAC	(C = 100) Medium Access Control
GSPS	Giga Samples Per Second
LSB	Least Significant Bit
SNR	Signal Noise Ration
THD	Total Harmonic Distortion
SINAD	Signal To Noise And Distortion
ENOB	Effective Number Of Bits
DNL	Differential Non Linearity
INL	Integral Non Linearity
CDR	Clock Data Recovery
LMFC	Local Multi Frame Clock
ILAS	Initial Lane Alignment Sequence
CGS	Code Group Synchronization
RoCE	RDMA over Convergent Ethernet
DMA	Direct Memory Access
RDMA	Remote Direct Memory Access
PC	Personal Computer
IP	Internet Protocol
UDP	User Datagram Protocol
LRH	Local Routing Header
GRH	Global Routing Header
BTH	Base Transport Header
DETH	Datagram Extended Header
ACK	ACKnowledge
NACK	Not ACKnowledge
WQ	Write Queue
RQ	Read Queue
CQ	Completion Queue
QP	Queue Pair
DLLP	Data Link Layer Packet
TLP	Transaction Layer Packet

CRC	Cyclic Redundancy Check
ECRC	End-to-end CRC
BAR	Base Address Register
DDR	Double Data Rate
QSFP	Quad Small Form-factor Pluggable
SFP	Small Form-factor Pluggable
FMC	Connector
FCS	Frame Check Sum
AXI	Advanced eXtensible Interface
AXIS	Advanced eXtensible Interface Stream



# **PŘÍLOHY**

Příloha 1. Příložené DVD se zdrojovými kódy