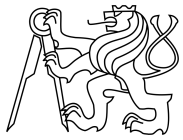
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Přenos zvukových signálů počítačovou sítí na platformě Zynq
Student:	Bc. Michal Majerčík
Vedoucí:	Dr. Ing. Sven Ubik
Studijní program:	Informatika
Studijní obor:	Návrh a programování vestavných systémů
Katedra:	Katedra číslicového návrhu
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Navrhněte hardwarový systém s obvodem FPGA pro přenos zvukových signálů přes počítačovou síť. Návrh vytvořte v jazyce VHDL a realizujte na vývojové desce s obvodem řady Zynq poskytnuté zadavatelem. Systém musí podporovat vstup a výstup zvuku přes digitální zvuková rozhraní S/PDIF a ADAT a jako vnořený zvukový signál v rozhraní 3G-SDI podle standardu SMPTE 299. Pro implementaci síťové části návrhu využijte modul poskytnutý zadavatelem.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 27. března 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra číslicového návrhu

Diplomová práca

Přenos zvukových signálů počítačovou sítí na platformě Zynq

Bc. Michal Majerčík

Vedúci práce: Dr. Ing. Sven Ubik

7. mája 2018

Poďakovanie

Ďakujem všetkým, ktorí ma pri tvorbe tejto práce podporovali. Zvláštna vďaka patrí vedúcemu práce Dr. Ing. Svenovi Ubikovi a Ing. Matějovi Bartíkovi za ich pomoc a rady.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 7. mája 2018 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2018 Michal Majerčík. Všetky práva vyhradené.
Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Majerčík, Michal. *Přenos zvukových signálů počítačovou sítí na platformě Zynq*. Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Predmetom tejto diplomovej práce je implementácia audio kodekov pre platformu Audio MVTP založenú na platforme Xilinx Zynq-7000. Jej výsledkom sú moduly vďaka ktorým je na platforme možný prevod audia medzi analógovou formou, digitálnymi audio štandardmi ADAT a S/PDIF a embedded audiom do 3G-SDI videa vo formáte AES3 podľa štandardu SMPTE299m. Umožňujú tiež audio prijímať a posielať po IP sieťach v RTP paketoch. Moduly sú implementované v jazyku VHDL.

Kľúčové slova Prenos audia, Zynq, S/PDIF, AES3, I^2S, ADAT, RTP, SMPTE299m

Abstract

Subject of this diploma thesis is implementation of audio codecs for Audio MVTP platform based on Xilinx Zynq-7000 system-on-chip. Results are modules that allow audio to be converted between analog form, digital audio standards ADAT and S/PDIF and embedded AES3 audio into 3G-SDI video stream according to SMPTE299m. Modules also allow the audio to be transmitted as RTP packets across IP networks. The modules are implemented in VHDL.

Keywords Audio transfer, Zynq, S/PDIF, AES3, I^2S, ADAT, RTP, SMPTE299m

Obsah

Úvod **1**

1 Analýza a návrh **3**
1.1 Pulzná kódová modulácia 3
1.2 Štandardy pre prenos zvuku 4
1.3 Real-time transport protokol 10
1.4 SMPTE 299m - 24-bitový digitálny audio formát pre SMPTE 292 13
1.5 Hardware použitý na vývoj 23
1.6 Návrh architektúry modulov 32

2 Realizácia **35**
2.1 Popis modulov pre prenos zvuku počítačovou sieťou 35
2.2 Popis modulov pracujúcich s formátom ADAT 46
2.3 Popis modulov na prenos audia spoločne s videom 53

3 Testovanie **59**
3.1 Simulácia modulov 59
3.2 Testovanie modulov na HW prototype 62
3.3 Výsledky testovania 68

Záver **71**

Literatúra **73**

A Zoznam použitých skratiek **75**

B **77**

C Obsah priloženého CD **79**

Zoznam obrázkov

1.1 Vizualizácia PCM. (Prevzaté z [1]) 4
1.2 AES3 kódovanie. (Prevzaté z [2]) 5
1.3 AES3 preambula X, ktorá nasleduje po bite s hodnotou 0. (Prevzaté z [2]) 5
1.4 AES3 Sub-frame s 24b audio vzorkou. (Prevzaté z [2]) 6
1.5 AES3 Sub-frame s 20b audio vzorkou. (Prevzaté z [2]) 6
1.6 AES3 Frame. (Prevzaté z [2]) 6
1.7 Channel status formát S/PDIF. (Prevzaté z [3]) 7
1.8 I^2S systémová konfigurácia a časovanie. (Prevzaté z [4]) 8
1.9 Kódovanie NRZI. (Prevzaté z [5]) 9
1.10 Hlavička RTP paketu. (Prevzaté z [6]) 10
1.11 Vzťah audio dáta paketu a AES3 (32-48 kHz). (Prevzaté z [7]) . . 14
1.12 Štruktúra audio paketu. (Prevzaté z [7]) 15
1.13 Pozícia audio paketov vo video streame. (Prevzaté z [7]) 16
1.14 Vzťah medzi video riadkami a audio vzorkami. (Prevzaté z [7]) . . 16
1.15 Rozloženie hodinových bitov. (Prevzaté z [7]) 17
1.16 Rozloženie AES3 bitov v CHn časti audio paketu. (Prevzaté z [7]) 17
1.17 Obvod používaný pre generovanie BCH kódu. (Prevzaté z [7]) . . . 18
1.18 Rozloženie bitov v ECC časti audio paketu. (Prevzaté z [7]) 18
1.19 Štruktúra audio kontrol paketu. (Prevzaté z [7]) 19
1.20 Počet video framov na celočíselný počet audio vzoriek. (Prevzaté z [7]) 20
1.21 Rozloženie DEL bitov. (Prevzaté z [7]) 20
1.22 Vkladanie viac ako 4 audio skupín. (Prevzaté z [8]) 21
1.23 Použitie audio multiplexeru so synchrónnym audiom. (Prevzaté z [8]) 21
1.24 Vyberanie viac ako 4 audio skupín. (Prevzaté z [8]) 22
1.25 Použitie audio demultiplexeru pre synchrónne audio. (Prevzaté z [8]) 22
1.26 Zynq-7000 (Prevzaté z [9]) 23
1.27 Štruktúra PL časti Zynq. (Prevzaté z [10]) 26
1.28 Štruktúra CLB. (Prevzaté z [10]) 27
1.29 Vývojová doska Trenz Electronic. (Prevzaté z [11]) 29

1.30 Blokový diagram WM8804 čipu. (Prevzaté z [12]) 30
1.31 Blokový diagram CS4272 čipu. (Prevzaté z [13]) 31
1.32 Príklad zapojenia výsledných modulov do funkčného systému. . . . 33

2.1 RTP Packet Payload sender. 36
2.2 Prenos jedného bloku dát. 38
2.3 Prenos dvoch blokov dát. 38
2.4 RTP packet payload sender - bloková schéma. 39
2.5 RTP packet payload receiver. 41
2.6 RTP Packet Payload Receiver - bloková schéma. 43
2.7 ADAT to I^2S konvertor. 46
2.8 ADAT to I^2S konvertor - bloková schéma. 49
2.9 I^2S to ADAT konvertor. 50
2.10 I^2S to ADAT konvertor - bloková schéma. 52
2.11 AES3 receiver. 53
2.12 AES3 receiver - boková schéma. 55
2.13 AES3 sender. 56
2.14 AES3 sender - bloková schéma. 58

3.1 Simulačné prostredie pre simuláciu RTP modulov v kooperácii. . . 60
3.2 Snímok zo simulácie kooperujúcich modulov. 61
3.3 Bloková schéma audio časti prototypu. 62
3.4 Prechod signálov cez izolátor. 63
3.5 Riešenie oneskorenia na izolátore. 64
3.6 Blokový diagram testovania modulov na prenos zvuku počítačovou sieťou. 65
3.7 Blokový diagram testovania modulov na prevod medzi ADAT a I^2S. 66
3.8 Blokový diagram testovania modulov na prenos audia spoločne s videom. 67
3.9 Vstup a výstup testov modulov na prenos zvuku počítačovou sieťou a prenos audia spoločne s videom. 68
3.10 Vstup a výstup ADAT modulov v loopbacku. 70
3.11 Prerušovaný výstup z RME ADI-2 zariadenia. 70

B.1 Channel status formát AES3. (Prevzaté z [2]) 77
B.2 Štruktúra ADAT paketu. (Prevzaté z [14]) 78

Zoznam tabuliek

1.1 Možné AES3 preambule. 5
1.2 Poradie kanálov v RTP pakete v závislosti od ich počtu. 12
1.3 Hodnoty RATE pre vzorkovacie frekvencie. 20
1.4 Zoznam I/O periférií Zynqu. 25

2.1 RTP packet payload sender popis rozhraní. 37
2.2 RTP packet payload receiver popis rozhraní. 42
2.3 Minimálna hodinová frekvencia pre moduly pracujúce s I^2S v závislosti od vzorkovacej frekvencie audia. 45
2.4 ADAT-I^2S modul popis rozhraní. 47
2.5 ADAT dekódovacia tabuľka. 48
2.6 I^2S-ADAT modul popis rozhraní. 51
2.7 AES3 receiver modul popis rozhraní. 54
2.8 AES3 sender modul popis rozhraní. 57

Úvod

Rozvoj vysokorýchlostného internetu priniesol možnosti prenosu veľkého množstva dát s nízkou latenciou. To nám umožňuje prenášať audiovizuálny obsah vysokej kvality po celom svete a v reálnom čase. Avšak so zvyšujúcimi sa možnosťami, rastú aj nároky užívateľov na kvalitu tohoto obsahu. Preto sa neustále zvyšujú nielen požiadavky na kvalitu a rozlíšenie videa, ale aj zvuku. Ten je rovnako dôležitým faktorom, ak má byť estetický zážitok z audiovizuálneho diela kvalitný.

Zvuk a obraz sú väčšinou neoddeliteľne späté. Je dôležitá ich synchronizácia, inak by dochádzalo k degradácii diela. Preto je vhodné pri streamovaní videa pridávať audio priamo doň a prenášať obe súčasne.

Avšak v niektorých prípadoch je zvuková stránka dôležitejšia a užívateľ má záujem najmä o ňu. Vhodným príkladom je streamovanie živých akcií klasickej hudby, kedy sú hudobníci viac-menej statickí a divák má záujem najmä o hudbu. Ďalším príkladom je prípad, kedy sa hudobníci nachádzajú na rôznych miestach na Zemi a chceli by spoločne hrať. Vtedy je potrebná okrem kvality aj nízka latencia audio streamu kvôli ich synchronizácii. Takéto spracovávanie a prenos audia má umožniť Audio MVTP.

Zariadenie Audio MVTP, vyvíjané združením CESNET, je navrhnuté na záznam audia v analógovej alebo digitálnej forme podľa štandardov ADAT a S/PDIF. Audio má byť následne možné prenášať po IP sieťach, alebo vkladať do video streamu ako embedded audio v reálnom čase. Taktiež má umožňovať prijímať audio dáta zo siete alebo ako embedded audio a konvertovať ho na analógový alebo digitálny audio signál s minimálnou latenciou. Na prenos po sieti je používaný real-time transport protokol. Podporovaný formát videa je 3G-SDI.

Výsledkom tejto práce je niekoľko modulov pre platformu Audio MVTP, ktoré umožňujú prijímať digitalizované audio, vkladať ho do paketov a posielať ich po sieti alebo ho vkladať do pomocnej časti videa ako embedded audio.

Štruktúra práce

V kapitole 1 je popísaný princíp prevodu analógového audio signálu na digitálne audio dáta. Ďalej sú tu popísané audio formáty, s ktorými MVTP platforma pracuje, formát paketov používaných na prenos audia v reálnom čase a štandard definujúci mapovanie audia do pomocnej časti videa. Na záver je popísané hardwarové vybavenie audio časti platformy a krátky návrh vyvíjaných modulov a spôsobu prenášania audia medzi nimi.

V kapitole 2 je presnejší popis modulov pre prenos zvuku počítačovou sieťou, modulov pracujúcich s formátom ADAT a modulov na prenos audia spoločne s videom spoločne s ich architektúrou a popisom rozhraní.

V kapitole 3 je popísaná softwarová simulácia modulov počas vývoja, integrácia a konečné testovanie na prototype Audio MVTP.

Kapitola 1

Analýza a návrh

Kapitola sa zaoberá princípom digitálnej reprezentácie analógového audio signálu a štandardami pre prenos zvuku, či už medzi rôznymi zariadeniami alebo medzi integrovanými obvodmi. Ďalej popisuje prenos real-time dát (najmä audia) po IP sieťach pomocou real-time transport protokolu a vkladanie zvuku do videa ako embedded audio. Popisuje použitý hardware na ktorom bude výsledný produkt implementovaný. Na záver sa zaoberá návrhom jednotlivých blokov pre spracovanie audia.

1.1 Pulzná kódová modulácia

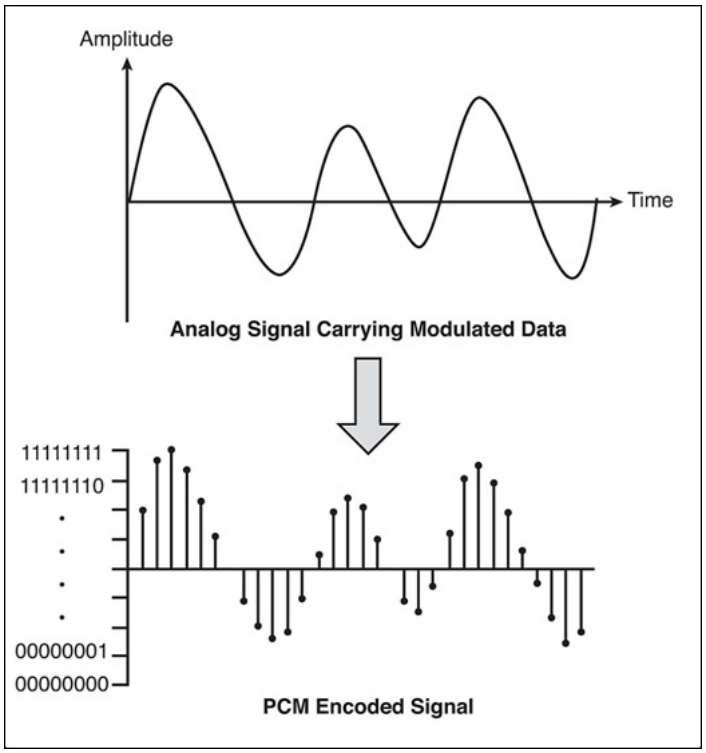
Pulzná kódová modulácia [1] (Pulse-Code Modulation, PCM) je metóda digitálnej reprezentácie analógového signálu.

Princíp PCM spočíva v pravidelnom vzorkovaní hodnôt signálu pomocou A/D prevodníku a ich zázname v binárnej podobe. Vizualizácia A/D prevodu je na obrázku 1.1. Pri reprodukcii sa k prevodu z digitálneho signálu na analógový používa D/A prevodník. Metóda má dva základné parametre, ktoré zásadne ovplyvňujú vernosť záznamu voči pôvodnému signálu. Sú to vzorkovacia frekvencia a bitové rozlíšenie (bitová šírka).

Vzorkovacia frekvencia určuje ako často je vstupný analógový signál odčítaný. Podľa Nyquist-Shannon teorému je presná rekonštrukcia spojitého, frekvenčne obmedzeného signálu z jeho vzoriek možná len za predpokladu, že vzorkovacia frekvencia je vyššia ako dvojnásobok najvyššej harmonické zložky vzorkovaného signálu. Priemerné ľudské ucho je schopné zaznamenať frekvencie zvuku medzi 20 Hz a 20 kHz. Ak by sme chceli verne pokryť celé toto spektrum, musí byť vzorkovacia frekvencia aspoň 40 kHz.

Bitové rozlíšenie určuje počet rôznych digitálnych hodnôt, ktorými je možné reprezentovať každú analógovú vzorku. Z toho vyplýva, že pri použití 24 bitového rozlíšenia existuje 2^{24} možných reprezentácii analógového signálu. Určenie hodnoty je nazývané kvantovanie. Každé kvantovanie pridáva do signálu kvantizačný šum, ktorého úroveň s rastúcim rozlíšením klesá.

Najčastejšie sa môžeme stretnúť s PCM s lineárnym kvantovaním (LPCM). Táto metóda sa používa najmä pri zázname hudby, napríklad audio CD funguje s dvoma audio kanálmi so vzorkovacou frekvenciou 41,1 kHz a 16b rozlíšením. Pri nelineárnom kvantovaní sa úroveň kvantizácie mení ako funkcia amplitúdy signálu (A-law, μ-law). Táto metóda sa vyskytuje najmä v telekomunikáciách.



Obr. 1.1: Vizualizácia PCM. (Prevzaté z [1])

1.2 Štandardy pre prenos zvuku

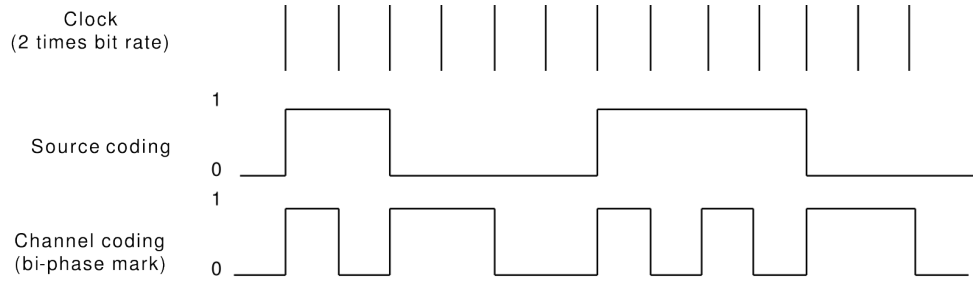
1.2.1 AES3

AES3 [2] je štandard pre výmenu digitálneho audio záznamu medzi profesionálnymi audio zariadeniami. Umožňuje prenos dvoch audio kanálov s PCM audio dátami po metalických a optických prenosových médiách. Štandard presne nedefinuje vzorkovaciu frekvenciu avšak odporúča frekvenciu 48 kHz. Možné bitové rozlíšenia sú 20b a 24b.

1.2.1.1 Protokol

AES3 prenáša dáta a hodiny jedným kanálom. Používa na to diferenciálne kódovanie Manchester (*Differential Manchester encoding, biphase mark code*).

Každý bit je reprezentovaný dvoma po sebe idúcimi binárnymi stavmi. Prvý stav súčasného bitu je vždy opačný ako druhý stav predchádzajúceho bitu. Druhý stav súčasného bitu je rovnaký, ako prvý stav ak je bit rovný 0. V prípade, že je bit rovný 1 druhý stav sa líši od prvého (obr.1.2).



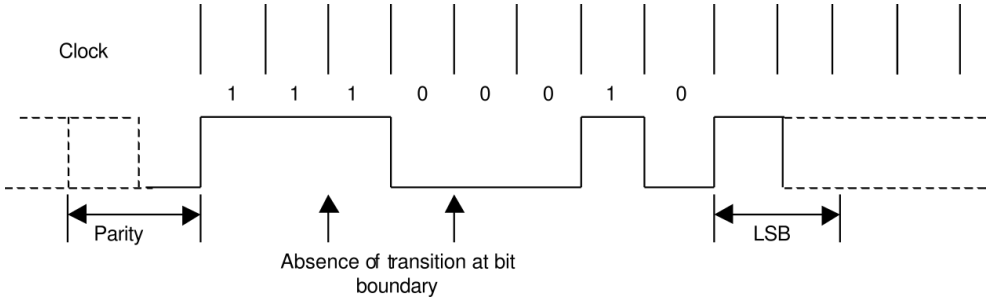
Obr. 1.2: AES3 kódovanie. (Prevzaté z [2])

Preambula

Preambula je špecifický vzor slúžiaci na synchronizáciu a identifikáciu sub-framov a blokov. Aby bola jednoznačne rozlíšiteľná od audio dát porušuje diferenciálne kódovanie (obr.1.3). Tvar preambule závisí od bitu, po ktorom nasleduje. Možné varianty AES3 preambúl sú v tabuľke 1.1.

Preambula	Umiestnenie preambuly	Predchádzajúci bit	
		0	1
X	Sub-frame 1	11100010	00011101
Y	Sub-frame 2	11100100	00011011
Z	Sub-frame 1 + block start	11101000	00010111

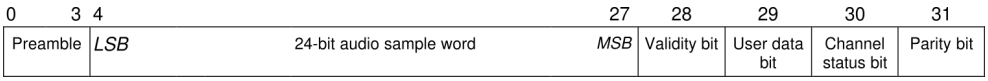
Tabuľka 1.1: Možné AES3 preambule.



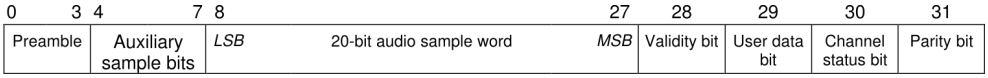
Obr. 1.3: AES3 preambula X, ktorá nasleduje po bite s hodnotou 0. (Prevzaté z [2])

Sub-frame

Každý sub-frame je rozdelený na 32 rovnakých časových úsekov (obr.1.4). Prvé 4 časové úseky obsahujú preambulu. V prípade 20b audio vzorku nasledujú 4 časové úseky pre pomocné bity (obr.1.5). Po preambule (resp. pomocných bitoch) nasleduje 24 (resp. 20) časových úsekov pre bity audio vzorky v poradí od najmenej významného (LSB) po najviac významný (MSB). Posledné 4 časové úseky slúžia pre validity bit, určujúci validitu audio vzorky, po ňom nasleduje user data bit, ktorého účel určuje užívateľ. Channel status bit nesie informácie o audio kanáli. Posledný parity bit je nastavený tak, aby v sub-frame bol párny počet bitov s hodnotou 1 (párna parita).



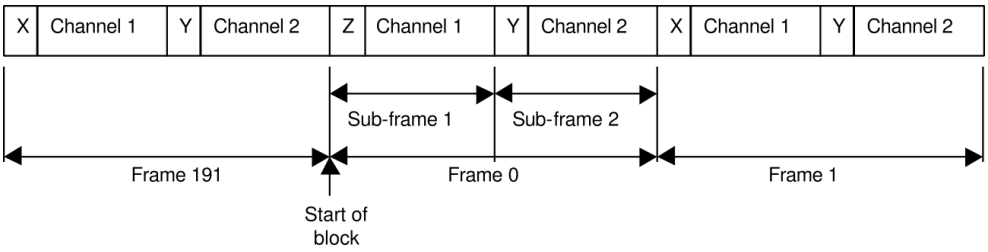
Obr. 1.4: AES3 Sub-frame s 24b audio vzorkou. (Prevzaté z [2])



Obr. 1.5: AES3 Sub-frame s 20b audio vzorkou. (Prevzaté z [2])

Frame

Frame je zložený z 2 sub-framov. Prvý frame v bloku začína preambulou Z a každý ďalší začína preambulou X. Druhý sub-frame začína vždy preambulou Y (obr.1.6). Frekvencia vysielania framov je rovná vzorkovacej frekvencii.



Obr. 1.6: AES3 Frame. (Prevzaté z [2])

Blok

192 po sebe idúcich framov začínajúcich Z preambulou sa nazýva blok. Je používaný na organizáciu informácií o audio kanáli. 192 bitov prenesených v channel status bit časti subframu vytvára informácie o kanáli (obr.B.1 v prílohe).

Fyzická vrstva

AES3 štandard taktiež určuje fyzickú vrstvu prepojenia. Odporúča symetrické tienené vedenie s charakteristickou impedanciou 110Ω (krútená dvojlinka) a XLR konektorom.

1.2.2 S/PDIF

S/PDIF [3] (*Sony/Philips Digital Interface*) je štandard pre výmenu digitálneho audio záznamu medzi spotrebiteľskými audio zariadeniami. Je založený na AES3 a rovnako umožňuje prenos dvoch audio kanálov s PCM audio dátami po metalických a optických prenosových médiách.

Jediný rozdiel medzi AES3 a S/PDIF je v použití channel status bitu (obr.1.7) a vo fyzickej vrstve, kde S/PDIF používa koaxiálny kábel s RCA konektormi. Inak sú oba štandardy totožné.

bit	meaning
0-1	channel status bits: bit 0 / bit 1 0 / 0 — IEC 60958-3 (Consumer) 1 / 0 — IEC 60958-4 (Professional) 0 / 1 — IEC 61397 (MPEG/AC-3/DTS/AAC/ATRAC), IEC 62105 and others 1 / 1 — SMPTE 337M and others
2	copy-protection. Copying is allowed when this bit is set.
3	is set when pre-emphasis is used.
4-7	0 (reserved)
9-15	category-code: ▪ 0 = common 2-channel format ▪ 1 = 2-channel CD-format (set by a CD-player when a subcode is transmitted) ▪ 2 = 2-channel PCM-encoder-decoder format ▪ others are not used
19-191	0 (reserved)

Obr. 1.7: Channel status formát S/PDIF. (Prevzaté z [3])

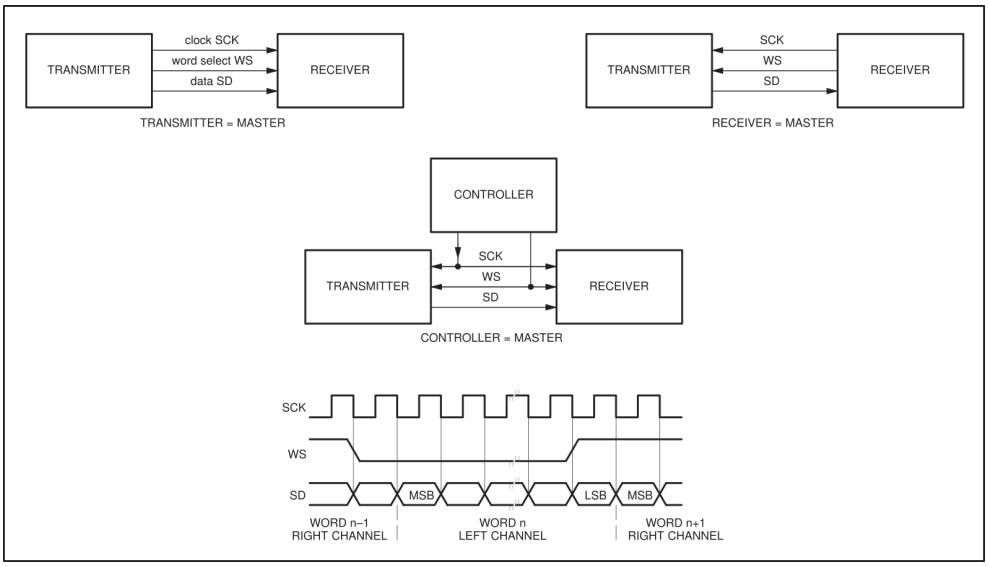
1.2.3 I^2S

I^2S [4] (Inter-IC Sound) je sériová zbernica určená na prenos PCM audio dát medzi integrovanými obvodmi, ale môže byť použitá aj medzi elektronickými zariadeniami. I^2S zbernica je schopná preniesť 2 časovo multiplexované audio kanály a pozostáva z 3 liniek:

- *Continuous serial clock* (SCK, BCLK) - hodiny jednotlivých dátových bitov.
- *Word select* (WS, LRCLK) - signál určujúci kanál (0 - ľavý, 1 - pravý).
- *Serial data* (SD, SDIN, SDOUT) - časovo multiplexované dáta.

Často sa používa ďalšia linka *master clock* (MCLK) ako hodinový signál pre slave jednotky. Tento signál však nie je štandardom definovaný.

Za vysielanie hodinového signálu (BCLK) a word select signálu (LRCLK) je zodpovedný master. V prípade, že obaja master aj slave potrebujú vysielať audio dáta naraz, musia byť použité dve serial data linky. K zbernici môže byť pripojený vždy len jeden master alebo kontrolér generujúci BCLK a LRCLK signály. Časovanie zbernice a možné konfigurácie sú na obrázku 1.8.



Obr. 1.8: I^{2}S systémová konfigurácia a časovanie. (Prevzaté z [4])

Sériové dáta

Dáta sú po zbernici posielané v poradí od najvýznamnejšieho bitu (MSB) po najmenej významný (LSB). Je to preto, že vysielač a prijímač nemusia pracovať s audio vzorkami rovnakej dĺžky. Nie je teda potrebné, aby vysielač a prijímač vedeli s akou dĺžkou audio slova pracuje ten druhý. V prípade, že vysielač používa väčšiu dĺžku slova ako prijímač, najnižšie bity sú zahodené. V opačnom prípade, kedy používa vysielač menšiu dĺžku slova ako prijímač, je audio vzorka prijímačom doplnená interne nulami na požadovanú dĺžku. Podstatné je, že najvýznamnejší bit (MSB) má presne určené miesto, kedy je vysielaný.

Vysielač vyšle MSB audio vzorky jednu hodinovú periódu (BCLK) po zmene word select (LRCLK) signálu. Ďalšie bity sú po zbernici posielané každý hodinový cyklus (BCLK).

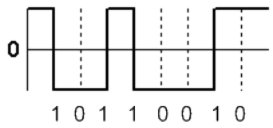
Podľa štandardu môže byť posledný (LSB) bit vysielaný až po zmene word select signálu tesne pred MSB ďalšieho vzorku. V praxi sa to však väčšinou nestáva a pre posielanie 16b, 20b a 24b audio vzoriek sa používa 32 hodinových cyklov na vzorku.

1.2.4 ADAT Lightpipe

ADAT Lightpipe [14] je štandard pre výmenu audio záznamu medzi profesionálnymi audio zariadeniami. Jedná sa o proprietárny štandard vyvinutý spoločnosťou *Alesis*. Originálny ADAT protokol umožňuje prenos 8 PCM audio kanálov so vzorkovacou frekvenciou 48 kHz a 24b rozlíšením. Avšak rozšírený protokol o techniku S/MUX (sample multiplexing) dovoľuje prenos 4 audio kanálov pri vzorkovacej frekvencii 96 kHz a 24b rozlíšením alebo 2 audio kanálov so vzorkovacou frekvenciou 192 kHz a 24b rozlíšením. Na prenos sa používa optické rozhranie (odtiaľ názov *lightpipe*).

Prenos dát

Na rozdiel od S/PDIF alebo AES3 nepoužíva ADAT diferenciálne kódovanie Manchester, ale NRZI (*Non-return-to-zero inverted*), ktoré umožňuje preniesť dvojnásobný objem dát oproti kódovaniu Manchester. V NRZI je bit 1 zakódovaný ako časový úsek so zmenou hodnoty signálu a bit 0 je zakódovaný ako časový úsek bez zmeny hodnoty signálu (obr.1.9).



Obr. 1.9: Kódovanie NRZI. (Prevzaté z [5])

ADAT spojenie je rozdelené do paketov veľkosti 256b (obr.B.2 v prílohe). Paket obsahuje 8 audio vzoriek s rozlíšením 24b rozdelených po 4b preskladaných bitmi hodnoty 1 kvôli dostatku zmien signálu ($24b * 8 * \frac{5}{4} = 240b$). Takže v najhoršom prípade dôjde každých 5 bitov ku zmene hodnoty signálu. Toto však neplatí pre synchronizačnú sekvenciu, ktorá pozostáva z 10 bitov hodnoty 0, teda bez zmeny hodnoty signálu, nasledovaných bitom hodnoty 1, teda zmenou hodnoty signálu ($11b$). Za synchronizačnou sekvenciou nasledujú 4 user bity ukončené bitom hodnoty 1 ($5b$). User bity sú používané nasledovne:

- User bit 0 je určený pre Timecode transport
- User bit 1 je určený pre MIDI data transport
- User bit 2 je určený pre S/Mux indikáciu (96 kHz mode)
- User bit 3 je rezervovaný a nastavený na 0

Fyzické rozhranie

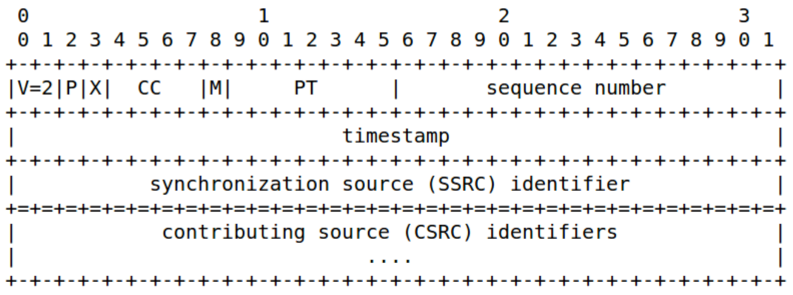
Na prenos ADAT sa používa optická linka s *Toslink* konektormi, umožňujúca prenos signálu na vzdialenosť 10 m.

1.3 Real-time transport protokol

Real-time Transport Protokol (RTP) je sieťový protokol štandardizujúci prenos zvukových a video dát po IP sieťach v reálnom čase. Využíva sa najmä na streamovanie, telefonovanie alebo videokonferencie po internete. Typicky RTP používa na transportnej vrstve user datagram protocol (UDP). S RTP protokolom súvisí aj RTP control protokol (RTCP). Zatiaľ, čo úloha RTP je prenos dát, RTCP monitoruje prenosové štatistiky, quality of service (QoS) a synchronizáciu viacerých dátových streamov.

Protokol bol vyvinutý Audio-Video Transport Working Group pri IETF, publikovaný ako RFC1889 [15] a neskôr nahradený RFC3550 [6]

1.3.1 Hlavičky paketov



Obr. 1.10: Hlavička RTP paketu. (Prevzaté z [6])

Hlavička RTP paketu je vyobrazená na obrázku 1.10. Jej minimálna veľkosť je 12B. Po základnej hlavičke môže byť vložené rozšírenie. Po hlavičke nasleduje obsah (payload), ktorého formát závisí na aplikácii. Jednotlivé časti hlavičky sú:

- *Version* (V): 2b. Určuje verziu RTP paketu. Súčasná verzia definovaná RFC3550 je 2.
- *Padding* (P): 1b. Ak je padding bit nastavený na 1, že paket obsahuje dodatočnú výplň. Posledný byte obsahu určuje veľkosť výplne (vrátane tohoto bytu), ktorá nie je súčasťou obsahu.
- *Extension* (X): 1b. Indikuje prítomnosť rozširujúcej hlavičky, ktorá sa nachádza medzi štandardnou hlavičkou a obsahom. Táto časť je špecifická pre niektoré formáty prenášaných dát.
- *CSRC count* (CC): 4b. Počet CSRC identifikátorov, ktoré nasledujú za hlavičkou.
- *Marker* (M): 1b. Použitie markeru je definované v profile (formáte) prenášaných dát.

- *Payload type* (PT): 7b. Identifikátor formátu RTP obsahu a určuje interpretáciu dát.
- *Sequence number*: 16b. Sekvenčné číslo je inkrementované o 1 s každým poslaným RTP paketom a je užitočné pri rekonštrukcii dát a detekcii stratených paketov.
- *Timestamp*: 32b. Timestamp je určený na identifikáciu okamihu, kedy má prijímač prijaté dáta prehrať. Granularita závisí na charaktere prenášaných dát. V súvislosti s prenosom dát, vďaka sekvenčným číslam a timestampu je jednoduché rozlíšiť časti prenosu, kde boli pakety stratené od častí prenosu, kde neboli žiadne pakety odoslané z dôvodu ticha na strane odosielateľa.
- *SSRC*: 32b. Identifikátor synchronizačného zdroja.
- *CSRC list*: 0 až 15 položiek, 32b každý. Zoznam identifikátorov prispievajúcich zdrojov k obsahu daného RTP paketu. Počet je určený CSRC count (CC) časťou.

1.3.2 Profily a formáty prenášaných dát

Design RTP paketu umožňuje prenášať množstvo multimediálnych formátov a umožňuje prenos novovzniknutých formátov bez potreby zmeny RTP štandardu. Toto je možné vďaka profilom a špecifikáciám payload formátov. Pre každý druh aplikácie (či už audio alebo video) RTP definuje profil a špecifikáciu jedného alebo viacerých payload formátov.

Profil definuje kodek použitý na kódovanie dát v obsahu a payload type (PT) časť hlavičky RTP paketu. Každý profil má jednu alebo viac špecifikácii payload formátov, ktoré špecifikujú prenos daného typu dát.

Z hľadiska tejto práce sú dôležité štandardy RFC3551 [16], popisujúci profil a RFC3190 [17] špecifikujúci payload formát.

1.3.3 RFC3551: RTP Profil pre audio a video konferencie s minimálnou potrebou riadenia

Tento profil [16] je určený pre použitie v rámci audio a video konferencií s minimálnou potrebou riadenia. Z hľadiska prenosu audia udáva profil poradie kanálov v obsahu RTP paketu. V prípade, ak sú použité viac ako 2 audio kanály sú očíslované zľava-doprava začínajúc číslom 1. V obsahu RTP paketu sú následne vzorky z týchto kanálov ukladané podľa ich čísla vzostupne.

Je použité nasledujúce značenie:

l	ľavý kanál (left)
r	pravý kanál (right)
c	stredný kanál (center)
S	surround
F	predný kanál (front)

Príklad radenia kanálov je v tabuľke 1.2.

Počet	Kanál					
kanálov	1	2	3	4	5	6
2	l	r				
3	l	r	c			
4	l	c	r	S		
5	Fl	Fr	Fc	Sl	Sr	
6	l	lc	c	r	rc	S

Tabuľka 1.2: Poradie kanálov v RTP pakete v závislosti od ich počtu.

V prípade štyroch kanálov by obsah v pakete vyzeral nasledovne:

1.vzorka ľavý k.	1.vzorka stredný k.	1.vzorka pravý k.
1.vzorka surround k.	2.vzorka ľavý k.	2.vzorka stredný k.
2.vzorka pravý k.	2.vzorka surround k.	...

Profil taktiež špecifikuje, že vzorky zo všetkých kanálov, zaznamenané v jeden okamih, musia byť prenášané v jednom RTP pakete.

1.3.4 RFC3190: RTP payload formát pre 12b DAT audio, 20b a 24b lineárne vzorkované audio

Tento payload formát [17] špecifikuje vkladanie 12b nelineárnych, 20b a 24b lineárnych audio dát do obsahu RTP paketu. Nelineárne 12b a nekomprimované 20b a 24b audio dáta reprezentované v dvojkovom doplnkovom kóde sú súvisle, jeden za druhým, vkladané do obsahu RTP paketu, začínajúc najvýznamnejším bytom (MSB). V prípade 12b a 20b vzorkou, ak obsah obsahuje nepárny počet vzoriek, nie sú posledné 4b posledného bytu použité. Vzorky vkladané do paketu začínajú najstaršou vzorkou.

1.4 SMPTE 299m - 24-bitový digitálny audio formát pre SMPTE 292

SMPTE 299m [7] štandard definuje mapovanie 24b audio dát a pridružených riadiacich dát do pomocného priestoru sériového digitálneho videa. Audio dáta sú do paketu vkladané vo formáte AES3.

Preferovaná realizácia tohoto štandardu je audio vzorkované s frekvenciou 48 kHz alebo 96 kHz synchrónne s videom. Je však možné použiť aj synchrónne alebo asynchrónne audio a vzorkovaciu frekvenciou 32 kHz a 41,1 kHz.

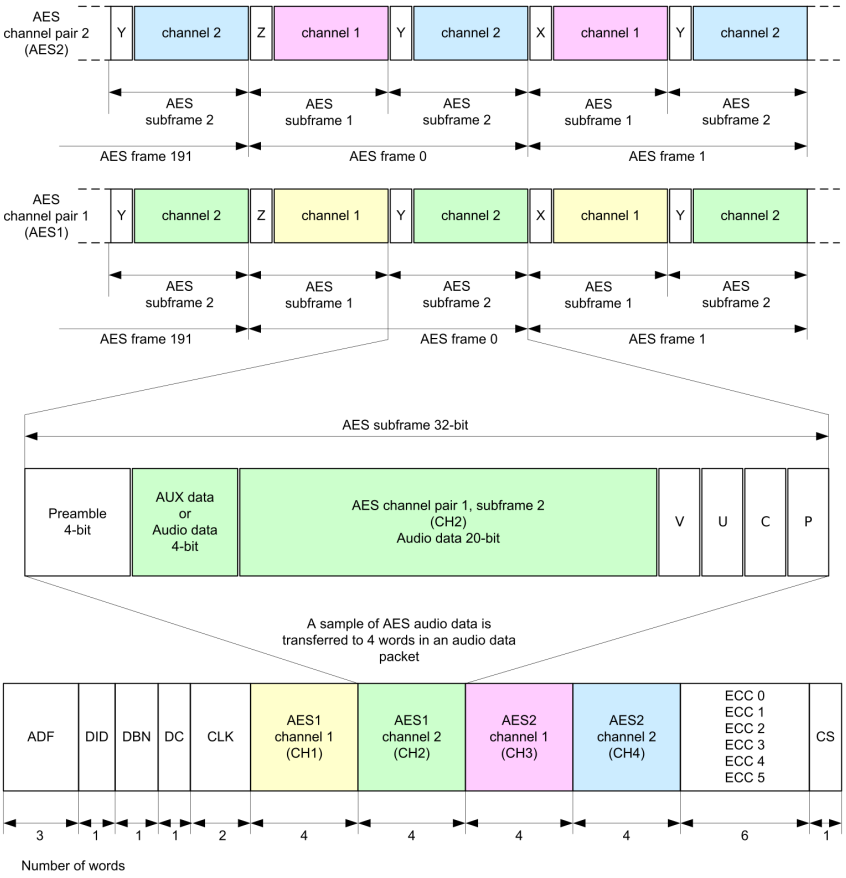
Audio kanály sú prenášané v skupinách. Skupín je maximálne 4 a každá má unikátne ID. Pri vzorkovacej frekvencii 32 kHz až 48 kHz sú v jednom pakete 4 kanály a je možne prenášať 16 kanálov. Pri vzorkovacej frekvencii 96 kHz sú v pakete 2 kanály a prenášať je možné 8 kanálov.

Kanály 1 až 4 (resp. 1 a 2) sú v skupine 1, kanály 5 až 8 (resp. 3 a 4) v skupine 2 a tak ďalej. Všetky audio kanály v danej skupine musia mať rovnakú vzorkovaciu frekvenciu, fázu a musia byť buď synchrónne alebo asynchrónne s videom.

Audio dáta pakety sú vkladané do horizontálneho pomocného priestoru C_b/C_r dáta streamu (HANC) a riadiace dáta sú vkladané do horizontálneho pomocného priestoru Y dáta streamu.

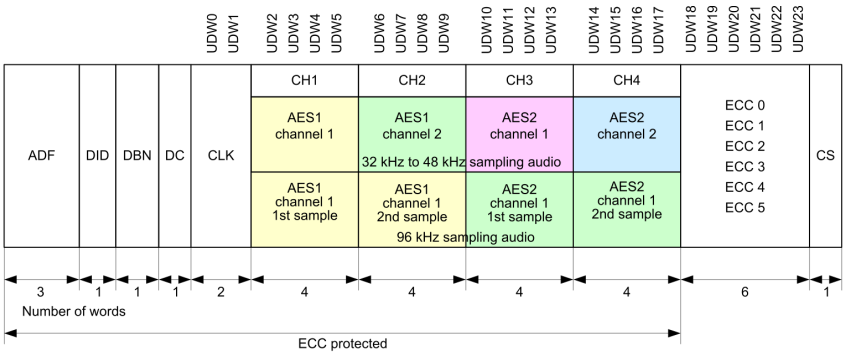
1.4.1 Audio dáta pakety

Audio dáta so vzorkovacou frekvenciou 32k Hz až 48 kHz získané zo 4 audio kanálov (dvoch párov AES3 zdrojov) sú vkladané do audio paketu podla obrázku 1.11 (*jedno dátové slovo - word - má 10 bitov*). V každom audio dáta pakete je prenášaná práve jedna audio vzorka na kanál. Počet audio dáta paketov v jednej skupine je maximálne N_a v horizontálnom pomocnom dáta bloku, kde N_a závisí na vzorkovacej frekvencii audia, frekvencii riadkov videa a na iných parametroch videa (presný výpočet je uvedený v štandarde).



Obr. 1.11: Vzťah audio dáta paketu a AES3 (32-48 kHz). (Prevzaté z [7])

V prípade audia so vzorkovacou frekvenciou 96 kHz sú AES3 subframe-y 1 a 2 (CH1 a CH2 na obr.1.11) dve po sebe idúce vzorky z jedného audio kanálu a to isté platí aj pre subframe-y 3 a 4. Z toho vyplýva, že vzorky z jedného kanálu na jeden paket sú dve (obr.1.12). Počet audio paketov v jednej skupine je maximálne $\frac{N_a}{2}$ v horizontálnom pomocnom dáta bloku.



Obr. 1.12: Štruktúra audio paketu. (Prevzaté z [7])

Štruktúra audio paketu (obr.1.12):

- *Ancillary data flag* (ADF) - Značka pomocných dát. Obsahuje hodnotu 000_{h} $3FF_{\mathrm{h}}$ $3FF_{\mathrm{h}}$.
- *Data identification* (DID) - Identifikátor dát. Jeho hodnota je $2E7_{\mathrm{h}}$ pre audio skupinu 1 (kanál 1-4), $1E6_{\mathrm{h}}$ pre audio skupinu 2 (kanál 5-8), $1E5_{\mathrm{h}}$ pre audio skupinu 3 (kanál 9-12) a $2E4_{\mathrm{h}}$ pre audio skupinu 4 (kanál 13-16).
- *Data block number* (DBN) - Číslo bloku. Hodnota DBN je inkrementovaná o 1 s každým ďalším paketom s rovnakým DID. (Hodnota DBN je 8 bitová, 9. bit je párna parita a 10. bit (MSB) inverzia bitu 9.)
- *Data count* (DC) - Počet slov v pakete. Musí byť nastavený na hodnotu 218_{h}. (8 najmenej významných bitov má hodnotu $24_{\mathrm{d}} = 18_{\mathrm{h}}$, bit 9. obsahuje párnu paritu a 10. bit (MSB) inverziu párnej parity.)
- *User data words* (UDW) - Užívateľské dáta. Paket obsahuje vždy 24 užívateľských slov (UDW0, UDW1 ... UDW22, UDW23).
- *Checksum* (CS) - Kontrolný súčet. Je 9 bitový s bitom číslo 10 rovným inverzii bitu 9. Hodnota tohoto súčtu sa počíta z DID, DBN, DC a všetkých UDW slov.

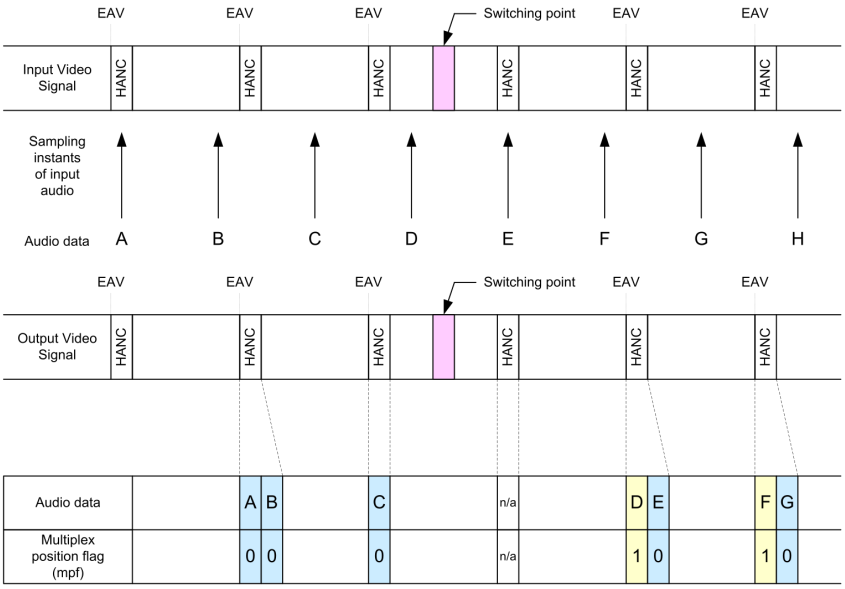
1.4.2 User data words - UDW

UDW sa skladá z troch typov dát - CLK, CHn, ECC.

CLK - audio clock phase data

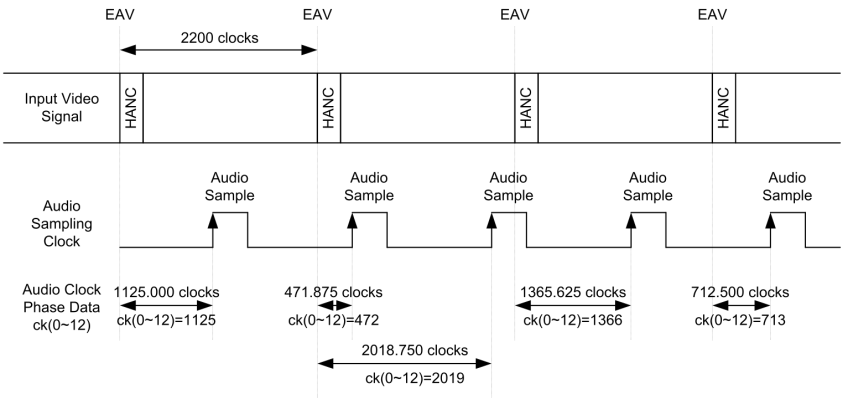
Audio dáta pakety sú vkladané do video riadku, ktorý nasleduje po momente, v ktorom boli audio vzorky v danom pakete zaznamenané. Avšak paket môže byť vložený až do druhého riadka, po zachytení vzorky a to v prípade, že je

zaznamenaný po prepínacom bode (*switching point*). MPF (*multiplex position flag*) bit určuje pozíciu audio dát vzhľadom k video riadkom. V prípade, že je MPF = 0, audio paket je umiestnený hneď v nasledujúcom video riadku. Ak je MPF = 1 tak sa audio paket nachádza až v druhom riadku. Príklad je na obrázku 1.13.



Obr. 1.13: Pozícia audio paketov vo video streame. (Prevzaté z [7])

Bity ck0 až ck12 udávajú čas (počet video hodín) medzi prvým slovom EAV (*End of Active Video*) a video vzorkou v čase, kedy sa na zvukovom výstupe má objaviť daná vzorka. V prípade 96 kHz audia je to čas druhej vzorky v poradí. Príklad časovania je na obrázku1.14.



Obr. 1.14: Vzťah medzi video riadkami a audio vzorkami. (Prevzaté z [7])

Rozloženie 12 bitových časovacích dát do dvoch 10b slov je uvedené na obrázku 1.15.

Bit number	UDW0	UDW1
b9 (MSB)	Not b8	Not b8
b8	Even parity [1)]	Even parity [1)]
b7	ck7 audio clock phase data	Reserved (set to 0)
b6	ck6 audio clock phase data	Reserved (set to 0)
b5	ck5 audio clock phase data	ck12 audio clock phase data (MSB)
b4	ck4 audio clock phase data	mpf multiplex position flag
b3	ck3 audio clock phase data	ck11 audio clock phase data
b2	ck2 audio clock phase data	ck10 audio clock phase data
b1	ck1 audio clock phase data	ck9 audio clock phase data
b0 (LSB)	ck0 audio clock phase data (LSB)	ck8 audio clock phase data

[1)] Even parity for b0 through b7.

Obr. 1.15: Rozloženie hodinových bitov. (Prevzaté z [7])

CHn - audio data

Rozloženie AES3 audio bitov v časti CHn paketu pre kanál 1 je na obrázku 1.16. V audio dáta pakete sú UDW2 až UDW17 použité pre prenos AES3 audio dát. V prípade, že sú použité menej ako 4 kanály v danom audio pakete, sú hodnoty V, U, C a P nastavené na 0. Bit Z je odvodený z preambuly AES3 a teda je nastavený na 1 v prípade, ak sa v AES3 streame nachádza preambula Z. Je spoločný pre dvojicu kanálov (CH1, CH2 alebo CH3, CH4).

	Bit number	UDW2	UDW3	UDW4	UDW5
CH1	b9 (MSB)	Not b8	Not b8	Not b8	Not b8
	b8	Even parity [1]	Even parity [1]	Even parity [1]	Even parity [1]
	b7	aud_1 3	aud_1 11	aud_1 19	P_1
	b6	aud_1 2	aud_1 10	aud_1 18	C_1
	b5	aud_1 1	aud_1 9	aud_1 17	U_1
	b4	aud_1 0 (LSB)	aud_1 8	aud_1 16	V_1
	b3	Z	aud_1 7	aud_1 15	aud_1 23(MSB)
	b2	0	aud_1 6	aud_1 14	aud_1 22
	b1	0	aud_1 5	aud_1 13	aud_1 21
	b0 (LSB)	0	aud_1 4	aud_1 12	aud_1 20

1 - Párna parita pre bity 0 až 7.

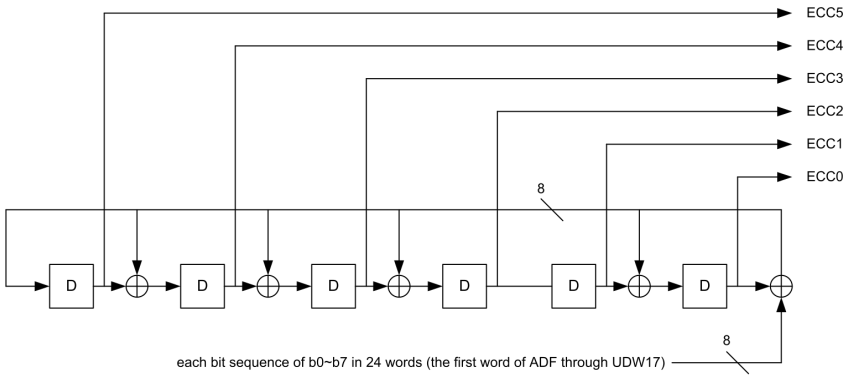
Obr. 1.16: Rozloženie AES3 bitov v CHn časti audio paketu. (Prevzaté z [7])

ECC - error-correcting code

Pre detekciu alebo opravu chýb od začiatku paketu (ADF) po koniec UDW17 je používaný cyklický samoopravný kód BCH(31, 25) s generujúcim polynómom:

$$ECC(X) = (X + 1)(X^5 + X^2 + 1) = X^6 + X^5 + X^3 + X^2 + X + 1$$

Obvod generujúci BCH kód je na obrázku 1.17 a rozloženie bitov v audio pakete na 1.18. Počiatočná hodnota registrov je 0. Výpočet prebieha pre každý bit b0 až b7, pre každé slovo od ADF až po UDW17 a výsledná hodnota v registroch je použitá ako ECCn.



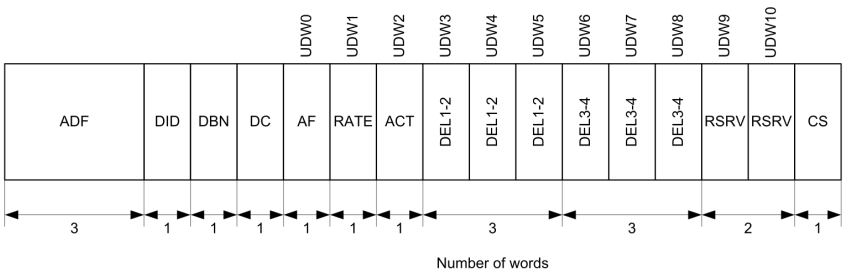
Obr. 1.17: Obvod používaný pre generovanie BCH kódu. (Prevzaté z [7])

	UDW18	UDW19	UDW20	UDW21	UDW22	UDW23
Bit number	ECC0	ECC1	ECC2	ECC3	ECC4	ECC5
b9 (MSB)	Not b8	Not b8	Not b8	Not b8	Not b8	Not b8
b8	Even parity [1)]	Even parity [1)]	Even parity [1)]	Even parity [1)]	Even parity [1)]	Even parity [1)]
b7	ecc0 7	ecc1 7	ecc2 7	ecc3 7	ecc4 7	ecc5 7
b6	ecc0 6	ecc1 6	ecc2 6	ecc3 6	ecc4 6	ecc5 6
b5	ecc0 5	ecc1 5	ecc2 5	ecc3 5	ecc4 5	ecc5 5
b4	ecc0 4	ecc1 4	ecc2 4	ecc3 4	ecc4 4	ecc5 4
b3	ecc0 3	ecc1 3	ecc2 3	ecc3 3	ecc4 3	ecc5 3
b2	ecc0 2	ecc1 2	ecc2 2	ecc3 2	ecc4 2	ecc5 2
b1	ecc0 1	ecc1 1	ecc2 1	ecc3 1	ecc4 1	ecc5 1
b0 (LSB)	ecc0 0	ecc1 0	ecc2 0	ecc3 0	ecc4 0	ecc5 0

1 - Párna parita pre bity 0 až 7.

Obr. 1.18: Rozloženie bitov v ECC časti audio paketu. (Prevzaté z [7])

1.4.3 Audio kontrol paket



Obr. 1.19: Štruktúra audio kontrol paketu. (Prevzaté z [7])

Štruktúra audio kontrol paketu (obr.1.19):

- *Ancillary data flag* (ADF) - Značka pomocných dát. Obsahuje hodnotu 000_h $3FF_h$ $3FF_h$.
- *Data identification* (DID) - Identifikátor dát. Jeho hodnota je $1E3_h$ pre audio skupinu 1 (kanál 1-4), $2E2_h$ pre audio skupinu 2 (kanál 5-8), $2E1_h$ pre audio skupinu 3 (kanál 9-12) a $1E0_h$ pre audio skupinu 4 (kanál 13-16).
- *Data block number* (DBN) - Číslo bloku. Hodnota DBN je vždy 200_h.
- *Data count* (DC) - Počet slov v pakete. Hodnota je vždy $10B_h$.
- *User data words* (UDW) - Užívateľské dáta. Paket obsahuje vždy 11 užívateľských slov (UDW0, UDW1 ... UDW9, UDW10).
- *Checksum* (CS) - Kontrolný súčet. Súčet je 9 bitový s bitom číslo 10 rovným inverzii bitu 9. Hodnota tohoto súčtu je počítaná z DID, DBN, DC a všetkých UDW.

1.4.4 User data words - UDW

UDW sa skladá z piatich typov dát - AF, RATE, ACT, DELm-n a rezervovanej časti (RSRV).

AF - audio frame number data

Číslo audio framu je poradové číslo v sérii audio framov. Séria audio framov je počet video framov potrebných na celočíselný počet audio vzoriek. Dĺžka série audio framov je na obrázku 1.20 (číslo za „/“). Hodnota AF je 9 bitová.

	Samples/frame				
Audio sampling rate	30.00 frame/s	30.00/1.001 frame/s	25.00 frames/s	24.00 frames/s	24.00/1.001 frames/s
96.0 kHz	3200/1	16016/5	3840/1	4000/1	4004/1
48.0 kHz	1600/1	8008/5	1920/1	2000/1	2002/1
44.1 kHz	1470/1	147147/100	1764/1	3675/2	147147/80
32.0 kHz	3200/3	16016/15	1280/1	4000/3	4004/3

Obr. 1.20: Počet video framov na celočíselný počet audio vzoriek. (Prevzaté z [7])

RATE

Vzorkovacia frekvencia všetkých audio kanálov. Bit 0 určuje, či sa jedná o synchrónne alebo asynchrónne dáta (0 - synch., 1 - asynch). Bity 1 až 3 určujú vzorkovaciu frekvenciu (tab.1.3). Viac bitov nie je použitých.

b2	b1	b0	vzorkovacia frekvencia
0	0	0	48.0 kHz
0	0	1	41.1 kHz
0	1	0	32.0 kHz
1	0	0	96.0 kHz

Tabuľka 1.3: Hodnoty RATE pre vzorkovacie frekvencie.

ACT

Aktívne audio kanály 1 až 4 sú určené bitmi 0 až 3 v ACT slove, iné bity nie sú použité. V prípade, že je kanál aktívny, je bit nastavený na 1, inak je 0.

DELm-n

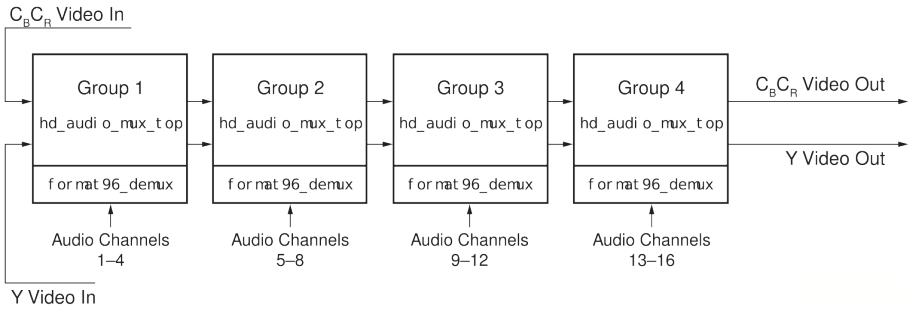
DELm-n udáva oneskorenie (delay) audia k videu spôsobené spracovaním a je udávané v vzorkovacích periódach. Rozloženie DEL bitov je na obrázku 1.21. Bit *e* určuje validitu audio oneskorenia.

	UDW3	UDW4	UDW5	UDW6	UDW7	UDW8
Bit number	DEL1-2			DEL3-4		
b9 (MSB)	Not b8	Not b8	Not b8	Not b8	Not b8	Not b8
b8	del 7	del 16	del 25 (±)	del 7	del 16	del 25 (±)
b7	del 6	del 15	del 24 (MSB)	del 6	del 15	del 24 (MSB)
b6	del 5	del 14	del 23	del 5	del 14	del 23
b5	del 4	del 13	del 22	del 4	del 13	del 22
b4	del 3	del 12	del 21	del 3	del 12	del 21
b3	del 2	del 11	del 20	del 2	del 11	del 20
b2	del 1	del 10	del 19	del 1	del 10	del 19
b1	del 0 (LSB)	del 9	del 18	del 0 (LSB)	del 9	del 18
b0 (LSB)	*e*	del 8	del 17	*e*	del 8	del 17

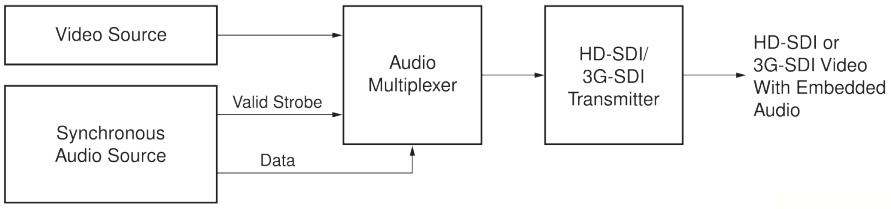
Obr. 1.21: Rozloženie DEL bitov. (Prevzaté z [7])

1.4.5 Audio multiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video

Audio multiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video [8] je verifikovaný referenčný design od spoločnosti Xilinx, ktorý implementuje vkladanie audia do video streamu podľa štandardov HD-SDI, 3G-SDI a dual-link HD-SDI. Dokáže generovať a vkladať audio dáta pakety i audio control pakety. Avšak jedna inštancia multiplexeru dokáže vkladať pakety len s jednou audio skupinou, teda štyrmi respektíve s dvomi kanálmi v prípade vzorkovacej frekvencie 96 kHz. Číslo danej skupiny je konfiguračným vstupom pre multiplexer a je teda možné použiť viac inštancii za sebou pre vkladanie viacerých audio skupín (obr.1.22). Podporované sú vzorkovacie frekvencie od 32 kHz po 96 kHz. Pri frekvencii 96 kHz je nutné použiť formátovač pre túto vzorkovaciu frekvenciu. Audio je do multiplexeru privádzané ako 24b audio vzorky spoločne so Z, V, U a C bitmi, potrebnými pre AES3 formát. Multiplexer taktiež dokáže prepisovať audio dáta rovnakej audio skupiny, vložné do video streamu v prípade, ak je to povolené. Príklad použitia multiplexeru je na obrázku 1.23.



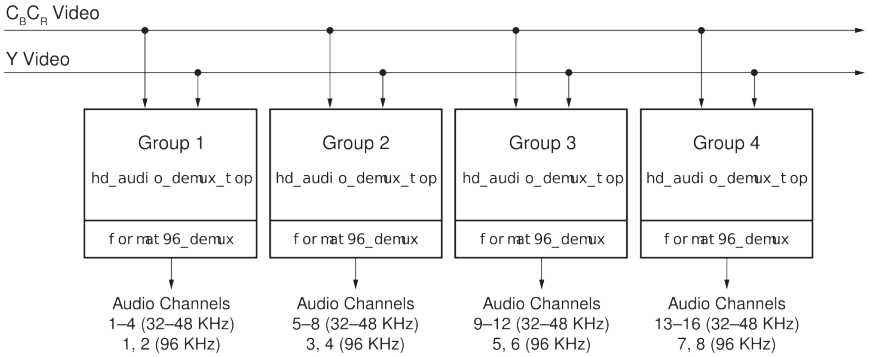
Obr. 1.22: Vkladanie viac ako 4 audio skupín. (Prevzaté z [8])



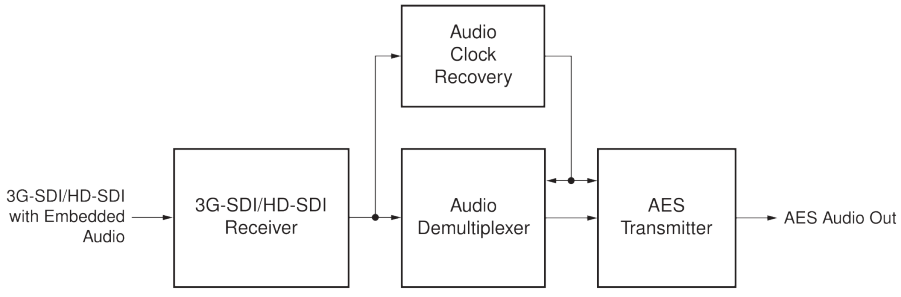
Obr. 1.23: Použitie audio multiplexeru so synchrónnym audiom. (Prevzaté z [8])

1.4.6 Audio demultiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video

Podobne existuje aj audio demultiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video [8]. Je to verifikovaný referenčný design od spoločnosti Xilinx, ktorý implementuje získavanie embedded audia z video streamu podľa štandardov HD-SDI, 3G-SDI, a dual-link HD-SDI. Jedna inštancia audio demultiplexeru dokáže z videa získať iba jednu AES3 audio skupinu, teda štyri kanály (respektíve dva v prípade 96 kHz vzorkovacej frekvencie). Taktiež dokáže získať dáta z control audio paketov a indikovať chyby vzniknuté pri prenose. Podporuje vzorkovacie frekvencie od 32 kHz po 96 kHz. Pri 96 kHz je však nutné použiť formátovač pre túto vzorkovaciu frekvenciu. Číslo danej skupiny je konfiguračným vstupom pre demultiplexer a je možné použiť viac inštancii za sebou pre získavanie viacerých audio skupín (obr.1.24). Na výstupe sú 24b audio vzorky a Z, V, U a C bity. Validita týchto dát je indikovaná vzostupnou hranou valid signálu. Príklad použitia multiplexeru je na obrázku 1.25.



Obr. 1.24: Vyberanie viac ako 4 audio skupín. (Prevzaté z [8])



Obr. 1.25: Použitie audio demultiplexeru pre synchrónne audio. (Prevzaté z [8])

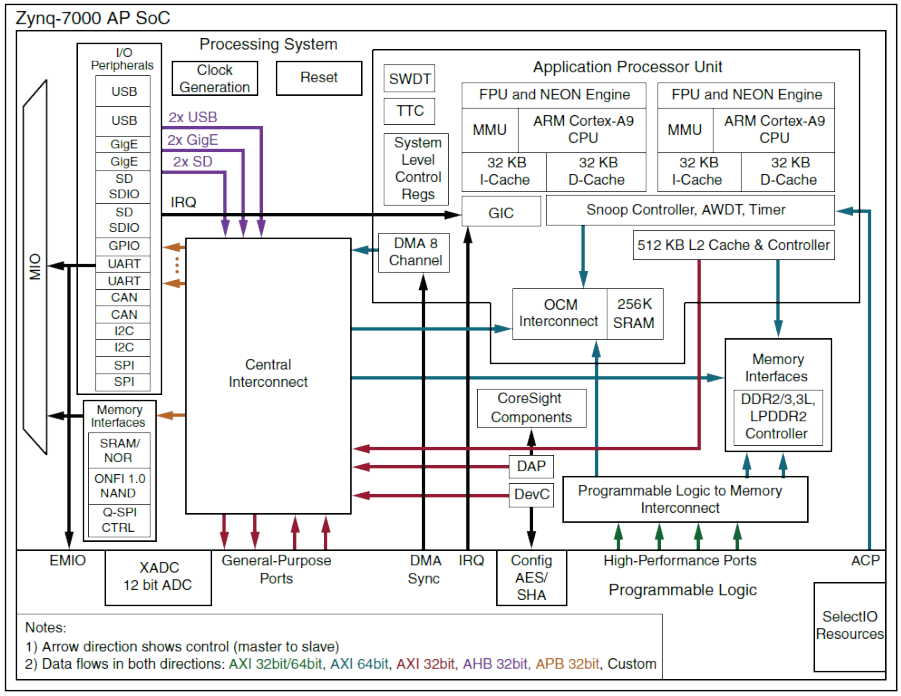
1.5 Hardware použitý na vývoj

Zariadenie Audio MVTP je založené na platforme Zynq. Konkrétne je použitý modul od firmy Trenz Electronic [11]. Okrem Zynqu sú na Audio MVTP použité aj kodeky spracúvajúce audio. V nasledujúcej kapitole sa nachádza popis samotného Zynqu, použitého modulu a popis použitých kodekov.

1.5.1 Zynq-7000 All Programmable SoC

Zynq-7000 [9], [10] je rodina systémov na čipe (System on chip - SoC) určených pre high-end vstavané aplikácie. Integruje v jednom zariadení aplikačný procesor a FPGA, vďaka čomu nachádza široké spektrum uplatnení.

Zynq sa skladá z dvoch hlavných častí, processing systemu (PS) a programmable logic (PL). *Processing system* je založený na single-core alebo dual-core procesore ARM Cortex-A9 vyrobenom 28 nm technológiou, na ktorom je možné spustiť operačný systém Linux. *Programmable logic* je postavená na siedmej sérii FPGA od spoločnosti Xilinx. Tieto dve časti môžu byť používané súčasne alebo každá zvlášť. Napájanie systému je navrhnuté tak, že ak je používaná len jedna časť, či už PS alebo PL, môže byť tá druhá odpojená od napájania. Architektúra celého systému je prepojená *Advanced eXtensible Interface* (AXI) zbernicou. Na obrázku 1.26 je bloková schéma Zynqu.



Obr. 1.26: Zynq-7000 (Prevzaté z [9])

1.5.1.1 Processing system (PS)

Zynq processing system obsahuje „hard"ARM Cortex-A9, schopný bežať na taktovacej frekvencii až 1 GHz (záleží však na konkrétnej rade Zynq zariadení). Výhodou oproti „soft"procesorom, ktoré môžu byť implementované v FPGA, je vyšší výkon.

PS neobsahuje len ARM procesor, ale aj ďalšie jednotky, ako je vidieť na obrázku 1.26 - Application processor unit (APU) - pamäť cache, rozhrania k pamätiam, generátor hodín a iné jednotky a prepojenia.

Application processor unit sa skladá z dvoch ARM jadier s prídavnými akceleračnými jednotkami, dedikovanými pre každé jadro:

- *NEON Media Processing Engine* (MPE) - pre akceleráciu práce s multimédiami (napr. kódovanie videa).
- *Floating Point Unit* (FPU) - jednotka na prácu s číslami s pohyblivou rádovou čiarkou.
- *Memory Management Unit* (MMU)
- *Level 1 cache memory* - každé jadro má 32KB inštrukčnej cache a 32KB dátovej cache.

APU obsahuje taktiež 512KB level 2 cache pre inštrukcie a dáta a 256KB on chip memory (OCM) spoločnú pre obe ARM jadrá. Jeho funkčnými blokmi sú aj časovače a kontroléri prerušení.

Snoop Control Unit (SCU) prepája ARM jadrá, level 2 cache a OCM pamäť. Má taktiež za úlohu starať sa o koherenciu dát medzi level 1 cache a zdielanou level 2 cache. SCU sprostredkováva aj transakcie medzi PS a PL cez *Accelerator Coherency Port* (ACP).

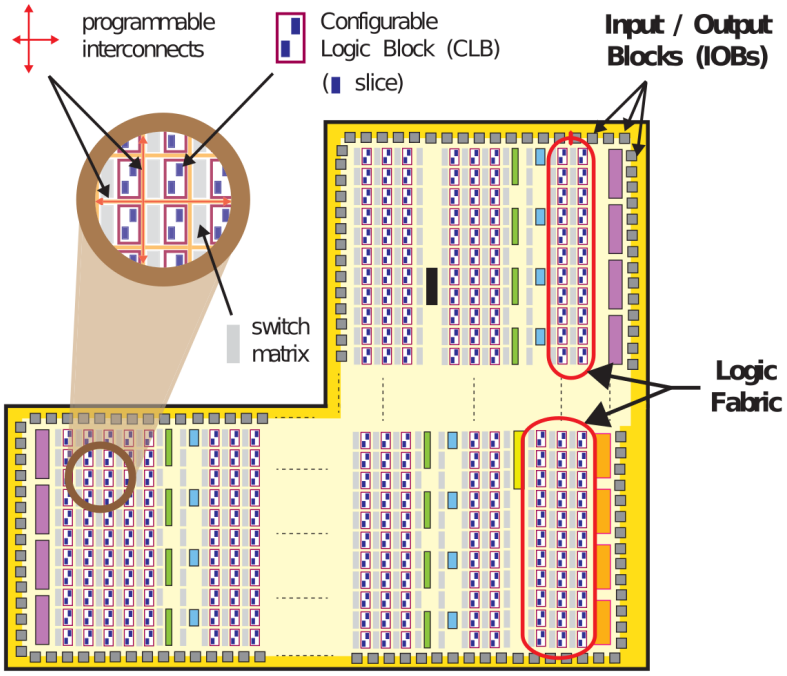
Aby bola zaistená flexibilita platformy okrem APU obsahuje Zynq aj radu periférnych zberníc, ktoré zabezpečujú prepojenie a komunikáciu s externými komponentami a zariadeniami. Táto komunikácia je zabezpečená pomocou *Multiplexed Input/Output* (MIO). Jedná sa o 54 konfigurovateľných pinov, ktoré je možné mapovať priamo na fyzické piny Zynqu. V prípade ak je potrebné pripojiť viac periférii je možné použiť *Extended MIO* (EMIO), ktoré sprostredkovávajú spojenie cez zdieľané I/O zdroje v PL časti Zynqu. Naviac umožňuje EMIO prepojenie PS časti s IP blokmi implementovaným v PL časti Zynqu. Zynq poskytuje periférne, zbernice a rozhrania uvedené v tabuľke 1.4.

I/O periféria	Popis
SPI (x2)	*Serial Peripheral Interface*. Štandard pre sériovú komunikáciu založenú na 4 pinovom rozhraní. Je podporovaný ako master tak i slave mode.
I^2C (x2)	*Inter-Integrated Circuit*. Nízkorýchlostná zbernica podľa štandardu I^2C, verzie 2. Podporovaný je master aj slave mode.
CAN (x2)	*Controller Area Network*. Sériová zbernica používaná hlavne v automobilovom priemysle.
UART (x2)	*Universal Asynchronous Receiver Transmitter*. Nízkorýchlostná sériová zbernica používaná často na pripojenie terminálu k PC.
GPIO	*General Purpose Input/Output*. Univerzálne vstupy a výstupy. Používa sa na LED, tlačítka, prepínače a iné. Zynq obsahuje 4 banky po 32 bitoch.
SD (x2)	Rozhranie pre *Secure Digital* kartu.
USB (x2)	*Universal Serial Bus*. Podpora USB 2.0. Môže byť použitý ako host, zariadenie alebo kombinovane („on-the-go").
GigE (x2)	Ethernet rozhranie s podporou 10Mbps, 100Mbps a 1Gbps modov.

Tabuľka 1.4: Zoznam I/O periférií Zynqu.

1.5.1.2 Programmable logic (PL)

Druhá časť architektúry Zynqu je programovateľná logika, ktorá je založená na Xilinx Artic-7 alebo Kintex-7 v závislosti na type modelu Zynqu. PL sa skladá z matice Configurable Logic Blocks (CLB), konfigurovateľných vstupno/výstupných blokov (I/O Blocks – IOBs), DSP48E1 blokov, block RAM pamätí, prepojovacej matice a ďalších komponentov. Štruktúra PL je na obrázku 1.27.



Obr. 1.27: Štruktúra PL časti Zynq. (Prevzaté z [10])

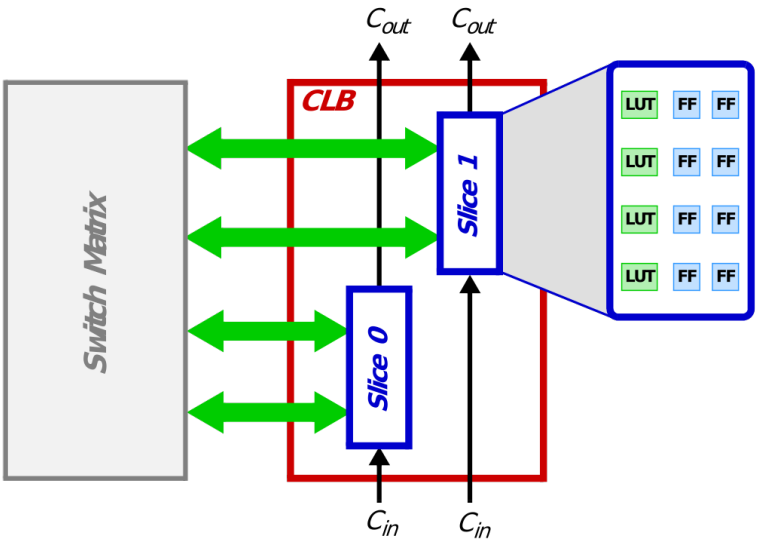
Configurable Logic Blocks - CLB

Konfigurovateľné logické bloky sú malé zoskupenia logických elementov. Tieto bloky tvoria maticu prepojenú programovateľnými spojmi. Štruktúra takéhoto CLB je na obrázku 1.28.

Každý blok CLB je tvorený dvoma slice. *Slice* je jednotka obsahujúca 4 *lookup tables* (LUT) a 8 jedno bitových pamäťových buniek typu *flip-flop* (FF). Tieto dva slice nie sú v CLB navzájom prepojené, ale sú priamo spojené s prepojovacou maticou. Každý slice môže byť zapojený do carry reťazca pre realizáciu rýchlych aritmetických výpočtov. LUT môžu byť použité ako:

- Logická funkcia so 6 vstupmi alebo dve logické funkcie s 5 spoločnými vstupmi a separátnymi výstupmi
- Malá ROM

- Malá RAM
- Posuvný register
- Zložitý multiplexer



Obr. 1.28: Štruktúra CLB. (Prevzaté z [10])

Block RAM - BRAM

V PL sú rozmiestnené block RAM pamäte. Každá z nich je dvoj-portová s veľkosťou 36Kb, ale je možné ju použiť ako dve nezávislé pamäte s veľkosťou 18Kb. Tieto pamäte je možné využiť ako RAM, ROM alebo FIFO. Štandardne je veľkosť slova 18b a v takejto konfigurácii BRAM obsahuje 2048 pamäťových elementov. Je možné ju rekonfigurovať, aby obsahovala viac menších elementov (napr. 4096 x 9b) alebo menej väčších (napr. 512 x 72b).

DSP48E1

DSP sú zložené z predsčítačky/odčítačky (pre-adder/subtractor), násobičky (multiplier) a post-sčítačky/odčítačky (post-adder/subtractor). Dokážu tiež vykonávať logické funkcie NOT, AND, OR, NAND, NOR, XOR a XNOR.

DSP sú špeciálne navrhnuté pre vysoký výkon a efektivitu. Sú vhodné na implementáciu komplexných aritmetických výpočtov, výpočtov s pohyblivou rádovou čiarkou, či na implementáciu rýchlych aritmetických operácií so strednou až dlhou dĺžkou slova. Pre krátke operácie sa používajú logické funkcie v LUT.

Input/Output blocks - IOBs

Vstupy a výstupy sú organizované do baniek po 50 IOB. Každý IOB obsahuje port, ktorý umožňuje spojenie s okolitým svetom pre jeden vstup alebo výstup. I/O banky sú kategorizované buď ako *high performance* (HP) alebo *high range* (HR) a podporujú rôzne I/O štandardy a napätia.

HP rozhrania sú limitované na 1,8 V a sú určené pre vysoko rýchlostne rozhrania (napr. pamäte). HR rozhrania podporujú napätia až do 3,3 V a podporujú viac I/O štandardov. Podporovaná je signalizácia so spoločnou zemou (potrebný 1 IOB), ale aj diferenciálna signalizácia (potrebné 2 IOB).

Každý IOB obsahuje IOSERDES pre programovú konverziu medzi paralelnými a sériovými formátmi s dĺžkou 2 až 8 bitov.

1.5.1.3 Rozhranie PS-PL

Silou Zynqu je najmä jeho schopnosť používať PS a PL časti v tandeme ako komplexný systém. Dôležitým aspektom je komunikácia týchto dvoch častí, ktorá je založená na štandarde *Advanced eXtensible Interface* (AXI). Súčasná verzia AXI4 je časťou štandardu ARM AMBA. Tento štandard bol vyvinutý pre komunikáciu v rámci mikrokontroléru, ale dnes sa používa aj v systémoch na čipe. AXI4 obsahuje viac protokolov, ktorých výber záleží na požadovaných vlastnostiach komunikácie.

- *AXI4* - používa sa na komunikáciu mapovanú do pamäte a poskytuje najvyšší výkon. Po poslaní adresy nasleduje dávka až 256 dátových slov („data beats").
- *AXI4-Lite* - je zjednodušená linka mapovaná do pamäte, kde po adrese nasleduje len jedno dátové slovo.
- *AXI4-Stream* - realizuje vysokorýchlostné streamovanie dát, podporuje dávkový prenos neobmedzeného množstva dát a je bez adresovacieho mechanizmu. Dáta plynú priamo medzi zdrojom a cieľom.

Rozhranie medzi PS a PL je tvorené 9 AXI rozhraniami, z toho každé je tvorené viacerými kanálmi. Tieto kanály vytvárajú spojenie medzi PS a PL. Rozhranie je point-to-point spojenie pre prenos adries, dát a synchronizačných signálov medzi master a slave zariadeniami. Toto prepojenie je tvorené:

- 2x 32-bit AXI master rozhranie (PS je master)
- 2x 32-bit AXI slave rozhranie (PL je master)
- 4x 64-bit High performance ports
- 1x 64-bit AXI Accelerator Coherency Port (ACP) rozhranie

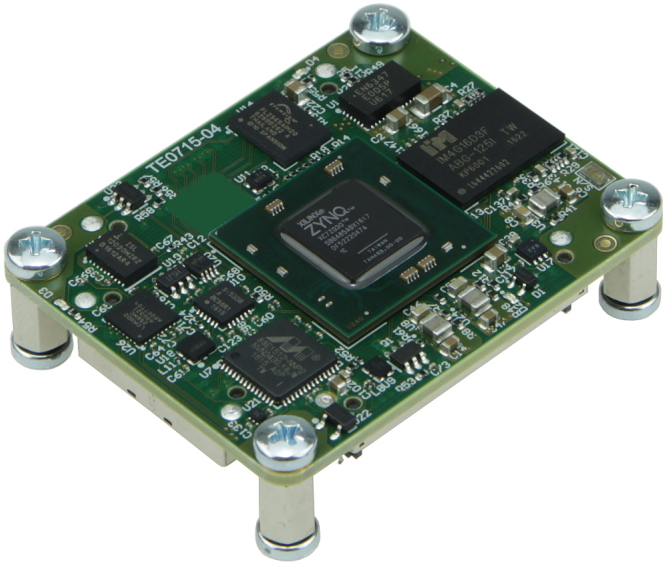
Ďalším rozhraním medzi PS a PL je EMIO, ktoré rozširuje možnosť pripojenia externých zariadení.

1.5.1.4 Vývojová doska s Xilinx Zynq XC7Z030-1SBG485C

Firma Trenz Electronic ponúka vývojárom niekoľko vývojových dosiek. Jednou z nich je TE0715-04-30-1C (obr.1.29) [11] osadená čipom Xilinx Zynq Z-7030. Na doske sú taktiež 4 MGT linky (gigabitový ethernet vysielač-prijímač), 1GB DDR3L SDRAM s 32b šírkou, 32MB SPI Flash pamäť, USB 2.0, programovateľný generátor hodín, teplotne kompenzované real-time hodiny, užívateľská LEDka a výkonný spínaný zdroj pre všetky potrebné napätia. Poskytuje 132 FPGA vstupov/výstupov a 14 PS-MIO.

Osadený Zynq čip obsahuje FPGA rady Kintex-7 s 78600 LUT, 157200 FF, 9.3Mb BRAM (265 blokov) a 400 DSP blokov.

Doska má veľkosť 4 x 5 cm a môže byť pripojená k ďalším zariadeniam pomocou dvoch 100 pinových a jedného 60 pinového konektoru.



Obr. 1.29: Vývojová doska Trenz Electronic. (Prevzaté z [11])

1.5.2 WM8804

WM8804 [12] je výkonný S/PDIF prijímač-vysielač, určený pre spotrebnú elektroniku. Podporuje jeden prijímaný a jeden odosielaný S/PDIF kanál.

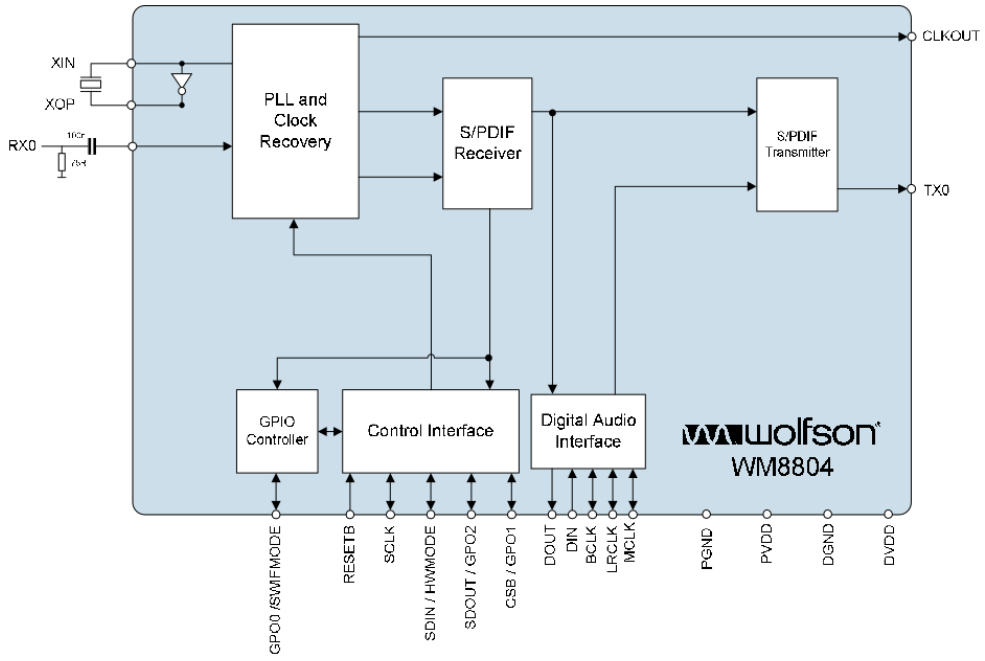
Ako hodinový signál potrebuje externý oscilátor alebo externe generované master hodiny. Dokáže generovať všetky štandardné audio hodiny pomocou interného fázového závesu (PLL - Phase-locked loop). Tieto hodiny môže distribuovať pomocou CLKOUT výstupu.

Zariadenie môže byť používane v hardware alebo v software control móde. V software control móde je možné zariadenie konfigurovať cez I^2C alebo SPI rozhranie.

Dáta zo vstavaného status a error monitoringu je možné získať pomocou kontrolného rozhrania (I^2C, SPI), pomocou GPO (general purpose output) pinov alebo zo streamu audio dát v móde „With Flags". Mód „With Flags"vysiela dáta spoločne s pripojenými status signálmi.

Rozhranie pre audio dáta podporuje I^2S, audio dáta zarovnané doprava alebo doľava a DSP audio formát s 16 až 24 bitovým rozlíšením so vzorkovacou frekvenciou 32 kHz až 192 kHz.

Blokový diagram WM8804 je na obrázku 1.30.



Obr. 1.30: Blokový diagram WM8804 čipu. (Prevzaté z [12])

1.5.3 CS4272

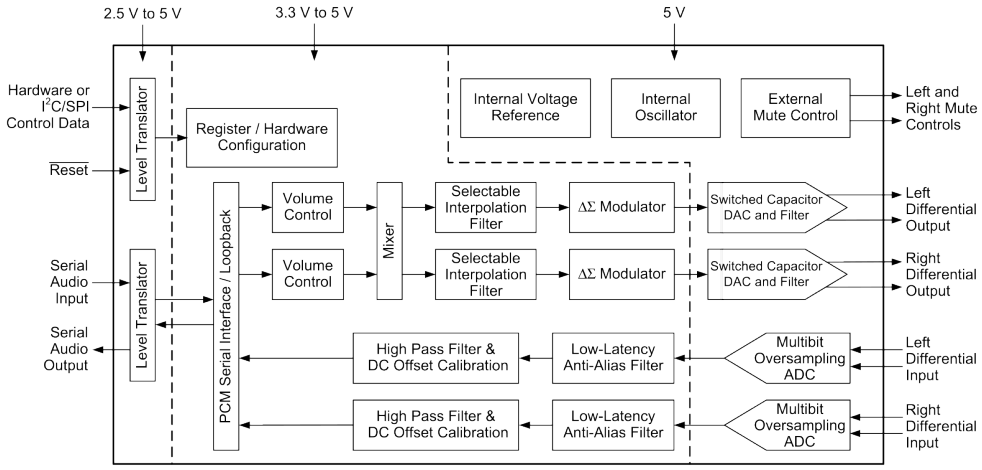
CS4272 [13] je výkonný integrovaný audio kodek, ktorý konvertuje analógový audio signál na digitálny a naopak pri vzorkovacej frekvencii až 192 kHz s audio šírkou až 24 bitov.

CS4272 využíva diferenčnú analógovú architektúru. Obsahuje interný oscilátor, ovládanie výstupnej hlasitosti, voliteľnú odpoveď digitálneho filtra a integrovaný prevodník logických úrovní, vďaka čomu je možné čip pripojiť k logickým úrovniam vstupného signálu od 2,5 V až po 5 V. V prípade absencie master hodín je automaticky použitý interný oscilátor.

Umožňuje výber zo sériových audio formátov I^2S, audio dáta zarovnané doprava alebo doľava a šíriek audio vzoriek z 16b, 18b, 20b a 24b. Softwarová konfigurácia je možná pomocou I^2C alebo SPI rozhrania. Čip môže byť nakonfigurovaný ako master alebo slave. V master mode sú LRCLK and BCLK výstupmi a signál na nich je generovaný na čipe. V slave móde sú tieto dva piny vstupmi a LRCLK a BCLK musí byť generované externe.

CS4272 je vhodný pre použitie v digitálnych mixpultoch, efektových procesoroch, DVD rekordéroch, set-top boxoch alebo v automobilových audio systémoch.

Blokový diagram CS4272 je na obrázku 1.31.



Obr. 1.31: Blokový diagram CS4272 čipu. (Prevzaté z [13])

1.6 Návrh architektúry modulov

Úlohou každého modulu je konvertovať audio z jedného štandardu na iný. Okrem toho musí byť možné posielať dáta medzi modulmi navzájom. Preto je vhodné, najmä kvôli ušetreniu práce, na prenos dát medzi modulmi použiť jeden zo štandardov, s ktorými má MVTP platforma pracovať, alebo ním posielajú dáta použité čipy WM8804 a CS4272.

Pretože na FPGA nie je problém to, že zbernica má viac vodičov, bola vybraná zbernica I^2S. Jej výhodou je najmä jednouchšia implementácia oproti ADAT, S/PDIF alebo AES3 a taktiež nízky jitter, pretože odpadá nutnosť zlučovať a oddeľovať hodinový signál od dát.

Výsledkom tejto práce je 6 modulov implementovaných v jazyku VHDL. Každý z modulov môže byť spojený s ostatnými modulmi pomocou skupiny I^2S zberníc.

- *RTP packet payload sender* má za úlohu prijať audio vzorky po I^2S zberniciach, vytvoriť z nich obsah RTP paketu a výsledok poslať na výstup. Na výstupe ich prevezme ďalší modul, ktorý obsah ďalej zapúzdri a pošle tieto dáta po sieti. Modul preberajúci dáta má zadávateľ práce už pripravený a nie je nutné ho vyvíjať.

- *RTP paket payload receiver* je opakom RTP packet payload senderu, jeho úloha je rozbaliť dáta prijaté v RTP pakete a poslať ich po odpovedajúcej I^2S zbernici. Modul prijímajúci dáta po sieti a extrahujúci obsah z RTP paketu je taktiež vyvinutý.

- *Embedded audio receiver* vyjme z horizontálneho pomocného priestoru C_b/C_r dáta streamu audio pakety definované SMPTE 299m. Rozbalí ich na samostatné audio vzorky a pošle po I^2S. Tento modul používa design audio multiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video popísaný v časti 1.4.5. Na to, aby mohol tento design správne pracovať je nutné doimplementovať konvertor medzi I^2S a AES3 štandardami.

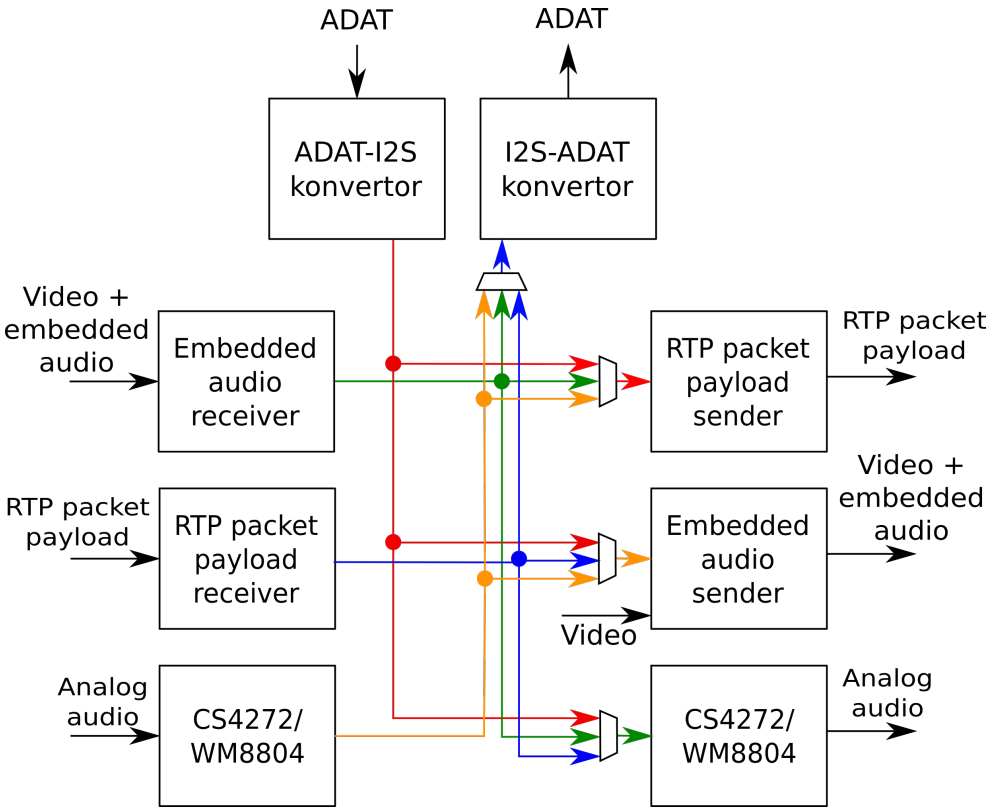
- *Embedded audio sender* zabalí audio vzorky prijaté po I^2S do audio paketov definovaných SMPTE 299m a vloží ich do horizontálneho pomocného priestoru C_b/C_r video dáta streamu. Tento modul používa design audio demultiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video popísaný v časti 1.4.6. Na to, aby mohol tento design správne pracovať je nutné doimplementovať konvertor medzi AES3 a I^2S štandardami.

- *ADAT-I^2S konvertor* dekóduje ADAT stream a prevádza ADAT pakety na I^2S.

- *I^2S-ADAT konvertor* prevádza I^2S na ADAT pakety a následne ich kóduje a vysiela ako ADAT stream.

Moduly je možné rozdeliť na dve skupiny tie, ktoré prijímajú dáta po I^2S a tie, ktoré ich vysielajú. Ktorýkoľvek modul z prvej skupiny môže byť pripojený na modul z druhej a prijímať od neho dáta. Keďže aj čipy CS4272 a WM8804 prenášajú dáta pomocou I^2S, je ich taktiež možné pripojiť priamo k modulom.

Napríklad dáta pochádzajúce z embedded audio receiveru je možné prehrať pomocou CS4272, alebo poslať po sieti ako RTP paket alebo oboje naraz. Rovnako audio, prijaté zo siete ako RTP paket, môže byť prehrané a vložené do videa. Audio vzorky z CS4272 je možné poslať po sieti, vložiť do videa, ale aj obe súčasne. Presun dát závisí len od konfigurácie modulov a multiplexerov medzi nimi. Príklad takého zapojenia a konfigurácie je na obrázku 1.32.



Obr. 1.32: Príklad zapojenia výsledných modulov do funkčného systému.

Kapitola 2

Realizácia

Kapitola podrobne popisuje každý z implementovaných modulov. Popisuje ich použitie, implementačné detaily, rozhrania a bloky, z ktorých sa skladajú.

2.1 Popis modulov pre prenos zvuku počítačovou sieťou

2.1.1 RTP packet payload sender

RTP packet payload sender prijíma audio vzorky po I^2S zberniciach. Tie usporiada v poradí definovanom štandardmi RFC 3551 [16] a RFC 3190 [17] a vytvorí tak obsah RTP paketu. Takto vytvorené dáta pošle ďalšiemu modulu, ktorý im pridá RTP hlavičku a celý RTP paket posunie na spracovanie nižším vrstvám UDP/IP.

Modul prijíma audio vzorky maximálne z 10 audio kanálov. Keďže I^2S podporuje prenos maximálne 2 kanálov, obsahuje 5 rozhraní pre pripojenie I^2S zberníc. Každému DIN vstupu pre audio dáta pripadá jeden LRCLK a BCLK vstup. Takže každá I^2S zbernica je úplne samostatná, nie je potrebná ich synchronizácia. Rozhrania sú v slave mode, BCLK a LRCLK sú vstupmi pre externé signály. Prijímanie dát na danom rozhraní je možne zakázať. Počet prijímaných audio bitov je 16 alebo 24, avšak je jednotný pre všetky audio vzorky zo všetkých vstupov. Vzorky sú do paketu ukladané podľa čísla rozhrania a kanálu vzostupne (ľavý kanál prvý). Možná vzorkovacia frekvencia je 32 kHz až 192 kHz.

Podľa RFC 3551 je odporúčaná dĺžka audio záznamu v jednom pakete 20 ms čo znamená, že len obsah paketu, pri posielaní 10 kanálov a maximálnej vzorkovacej frekvencii (192 kHz), má veľkosť:

Vzorkovacia frekvencia $*$ Čas $*$ Bitová šírka vzorky $*$ Počet kanálov:

$$192000\ Hz * 0,02\ s * 24b * 10 = 921\ 600b = 115\ 200B$$

Takýto obsah je príliš veľký vzhľadom na *maximum transmission unit* (MTU) ethernet framu, ktorá je 1500B. Preto je dĺžka audio záznamu v jednom pakete 40 vzoriek z každého kanálu. Maximálna možná veľkosť v prípade 24b audio vzoriek a 10 kanálov tým pádom bude:

$$40 * 24b * 10 = 9600b = 1200B$$

Pri vzorkovacej frekvencii 192 kHz to predstavuje $0,2083$ ms, pri 48 kHz - $0,8\bar{3}$ ms a pri 32 kHz - $1,25$ ms. V tomto prípade zostane ešte dosť miesta pre RTP, UDP a IP hlavičky a nebude dochádzať k fragmentácii paketov.

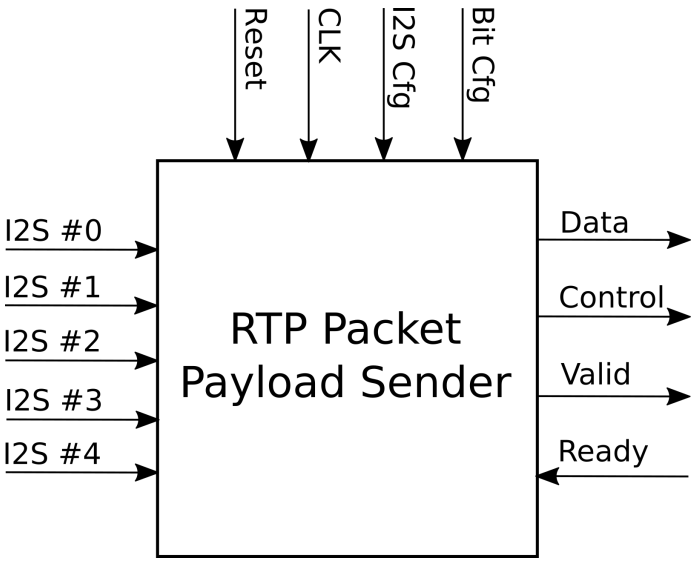
RTP packet payload sender nečaká na dostatok dát na vytvorenie celého paketu, ale v okamihu, keď má 64b audio dát ich pošle na výstup. V prípade poslednej časti audio dát je možné poslať menej ako 64b.

Modul taktiež generuje 32b timestamp.

2.1.1.1 Interface

Interface RTP packet senderu je vyobrazený na obrázku 2.1 a popísaný v tabuľke 2.1. Má tri časti:

- *I^2S interface* - rozhrania pre prijímanie dát po I^2S zberniciach.
- *Konfiguračný interface* (I2S Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku a aktívne I^2S rozhrania.
- *Paket interface* (Data, Control, Valid, Ready) - po tomto rozhraní posiela modul obsah RTP paketov na ďalšie spracovanie a poslanie po sieti.



Obr. 2.1: RTP Packet Payload sender.

Interface modulu				
#	Názov	Počet bitov	Smer	Popis
1	I2S #0	3	in	I^2S vstup (DIN, BCLK, LRCLK)
2	I2S #1	3	in	I^2S vstup (DIN, BCLK, LRCLK)
3	I2S #2	3	in	I^2S vstup (DIN, BCLK, LRCLK)
4	I2S #3	3	in	I^2S vstup (DIN, BCLK, LRCLK)
5	I2S #4	3	in	I^2S vstup (DIN, BCLK, LRCLK)
6	Data	64	out	Výstup pre obsah RTP paketu
7	Control	8	out	Riadiace signály bit 0 - Packet data bit 1 - Packet end bit 2-4 - Packet alignment bit 5 - PHY error bit 6 - Timestamp bit 7 - CRC error
8	Valid	1	out	Dáta pripravené na odoslanie
9	Ready	1	in	Cieľ pripravený prevziať dáta
10	I2S Cfg	5	in	Aktívne I^2S rozhrania bit 0 - I2S #0 bit 1 - I2S #1 bit 2 - I2S #2 bit 3 - I2S #3 bit 4 - I2S #4
11	Bit Cfg	1	in	Počet prijímaných audio bitov 0 - 16b 1 - 24b
12	CLK	1	in	Hodinový vstup
13	Reset	1	in	Vstup pre synchrónny reset aktívny v 1

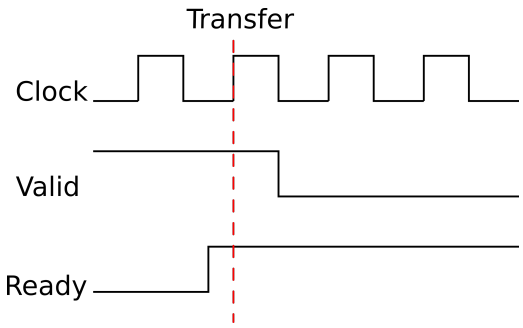
Tabuľka 2.1: RTP packet payload sender popis rozhraní.

Riadiace signály

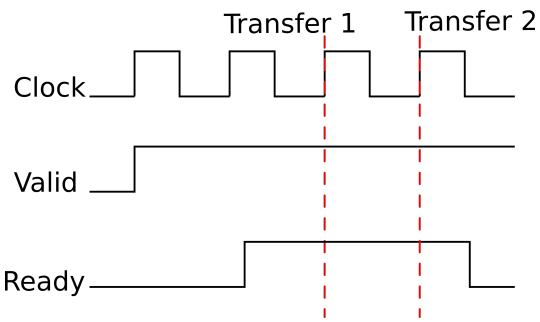
- Packet data - Indikuje validitu dát.
- Packet end - Signalizuje posledný cyklus validných dát.
- Packet alignment - Počet bytov v cykle validných dát (0 = 8b, 1 = 1b, 2 = 2b, atď).
- PHY error - Error generovaný fyzickou vrstvou (nastavený na trvalú 0).
- Timestamp - Signalizuje timestamp dáta.
- CRC error - Indikuje paket so zlým CRC (nastavený na trvalú 0).

Ready/Valid handshake

Zdroj nastaví valid signál na 1, ak je pripravený poslať dáta cieľu. Cieľ nastaví ready signál na 1, ak je pripravený dáta spracovať. Dáta sa prenášajú zo zdroja k cieľu, iba ak sú oba signály ready a valid nastavené na 1 súčasne. Po nastavení ready a valid na 1 budú dáta zo zdroja na nasledujúcu vzostupnú hranu hodín prijaté cieľom (Obr.2.2). K prenosu dát dochádza na každú vzostupnú hranu hodín, počas ktorej sú ready a valid v 1 (Obr.2.3).



Obr. 2.2: Prenos jedného bloku dát.

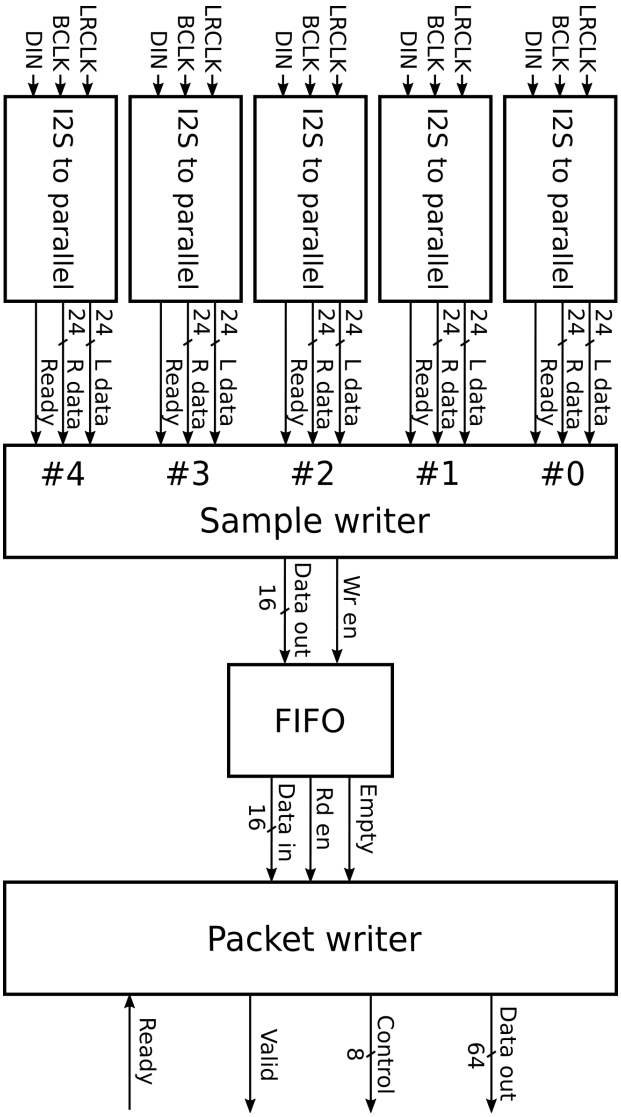


Obr. 2.3: Prenos dvoch blokov dát.

2.1.1.2 Architektúra modulu

Počas vývoja modulu boli vyskúšané dva prístupy. Najskôr boli použité dva bloky, z ktorých jeden prijímal vzorky po I^2S a druhý ich vkladal do paketu. Tento prístup však bol nedostatočný. Ukázalo sa, že ak nastane oneskorenie na výstupe modulu, môže dôjsť k strate dát. Preto bola do návrhu pridaná fronta, ktorá tomu zabraňuje.

Bloková schéma výsledného modulu je na obrázku 2.4. Všetky bloky majú hodinový vstup, synchrónny reset a všetky okrem fronty (FIFO) majú konfiguračný vstup na šírku audio slov (bit cfg) a na konfiguráciu aktívnych I^2S rozhraní (I2S cfg). Tieto vstupy na obrázku kvôli prehľadnosti nie sú.



Obr. 2.4: RTP packet payload sender - bloková schéma.

RTP packet sender sa skladá zo 4 častí:

- *I^2S to parallel blok* - návrh obsahuje 5 takýchto blokov. Ich úlohou je prijímať audio vzorky po I^2S zbernici. Blok prijíma dve audio vzorky (ľavý a pravý kanál) so šírkou 24b každú vzorkovaciu periódu. Prijaté slová následne zapíše do registrov bloku, ku ktorému je pripojený. V tomto prípade do sample writer bloku. Dáta sa prenášajú z bloku na vzostupnú hranu hodín v prípade, ak je signál *ready* nastavený na 1. V prípade 16b vzoriek je najmenej významných 8 bitov v 24b slove nastavených na 0.

  Každý z I^2S to parallel blokov je možne deaktivovať, resetovať a nastaviť bitovú šírku vzoriek. Šírka však musí byť jednotná pre všetky bloky, ak má modul správne fungovať.

- *Sample writer blok* - úlohou tohoto bloku je prechádzať sekvenčne registre, do ktorých boli zapísané dáta z aktívnych I^2S to parallel blokov a zapisovať prijaté audio vzorky po slovách do fronty. Sample writer vždy čaká pokým nie sú audio vzorky zapísané do registra pripadajúceho danému I^2S to parallel bloku. Napríklad, ak sú aktívne vstupy 1 a 4, bude blok čakať až do chvíle, kedy sú audio vzorky zapísané do registrov pripadajúcich bloku 1. Následne dáta z nich zapíše do fronty a prejde na registre bloku 4. Ak sú už audio vzorky zapísané v týchto registroch zapíše ich do fronty, inak čaká na zápis. Následne prejde opäť na registre číslo 1.

  Slová zapisované do fronty majú dĺžku 16b. V prípade 16b audio vzoriek sú každú vzorkovaciu periódu zapísané do fronty dve slová z každej I^2S zbernice. Ak je bitová šírka vzoriek 24b, do fronty sú zapísané tri 16b slová (1. slovo = najvýznamnejších 16b ľavého kanálu, 2. slovo = najmenej významných 8b ľavého kanálu + najvýznamnejších 8b pravého kanálu, 3. slovo = najmenej významných 16b pravého audio kanálu).

- *FIFO* - synchrónna fronta z knižnice *unimacro* od firmy Xilinx. Je realizovaná v pamäti BRAM s veľkosťou 18Kb s 16b šírkou slova. Hĺbka fronty je 1024 slov.

- *Packet writer blok* - vyberá dáta z fronty, vytvára z nich 64b slová a zapisuje ich do ďalšieho modulu, ktorý z nich pridaním hlavičky vytvára pakety a posiela ich po sieti. Na prenos dát sa používa ready-valid handshake. Blok taktiež generuje timestamp a nastavuje riadiace signály.

  V jednom pakete je 40 audio vzoriek z každého kanála. Preto ich packet writer počíta a s posledným slovom nastaví *packet end* signál. Na to musí poznať počet aktívnych kanálov a bitovú šírku jednej vzorky.

  Timestamp je posielaný prvý. Po resete je nastavený na nulu a s každým ďalším odoslaným paketom sa zvýši o počet 40 (počet vzoriek na kanál).

2.1.2 RTP packet payload receiver

RTP Packet Payload receiver je opakom RTP packet payload senderu. Modul prijíma od nižších vrstiev UDP/IP obsah RTP paketu. Získa z neho audio vzorky, ktoré následne rozosiela po I^{2}S zberniciach.

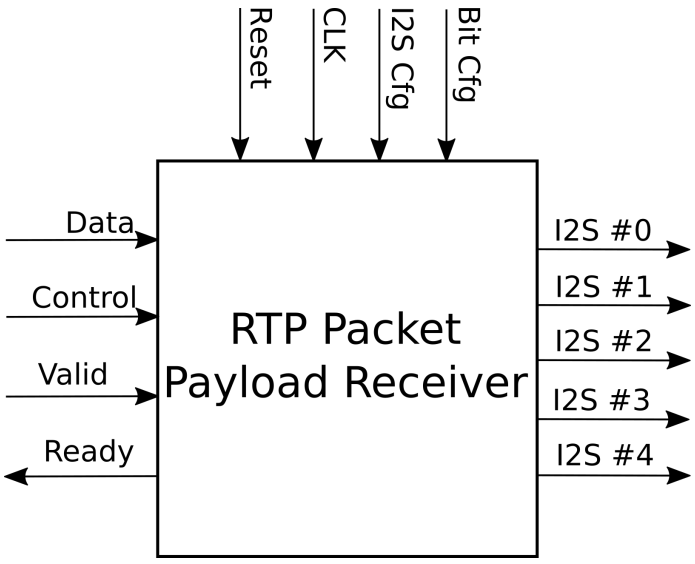
Modul môže vysielať audio vzorky so šírkou 16 alebo 24 bitov po maximálne piatich I^{2}S zberniciach, teda po 10 audio kanáloch. Bitová šírka je jednotná pre všetky vysielané audio vzorky a I^{2}S výstupy. Odosielanie dát na danom výstupe je možné vypnúť. Každému DOUT výstupu pre audio dáta pripadá jeden LRCLK a BCLK vstup. Každá I^{2}S zbernica je úplne samostatná a nie je potrebná ich synchronizácia. I^{2}S rozhrania sú v slave mode, v ktorom sú BCLK a LRCLK vstupmi pre externé signály.

V prípade výskytu CRC erroru alebo PHY erroru nebude paket prijatý. Miesto takéhoto paketu je po I2S poslaná audio vzorka s hodnotou 0.

2.1.2.1 Interface

Interface RTP packet receiveru je vyobrazený na obrázku 2.5 a popísaný v tabuľke 2.2. Má tri časti

- *I^{2}S interface* - rozhrania pre posielanie dát po I^{2}S zberniciach.
- *Konfiguračný interface* (I2S Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku a aktívne I^{2}S rozhrania.
- *Paket interface* (Data, Control, Valid, Ready) - po tomto rozhraní prijíma modul obsah RTP paketov.



Obr. 2.5: RTP packet payload receiver.

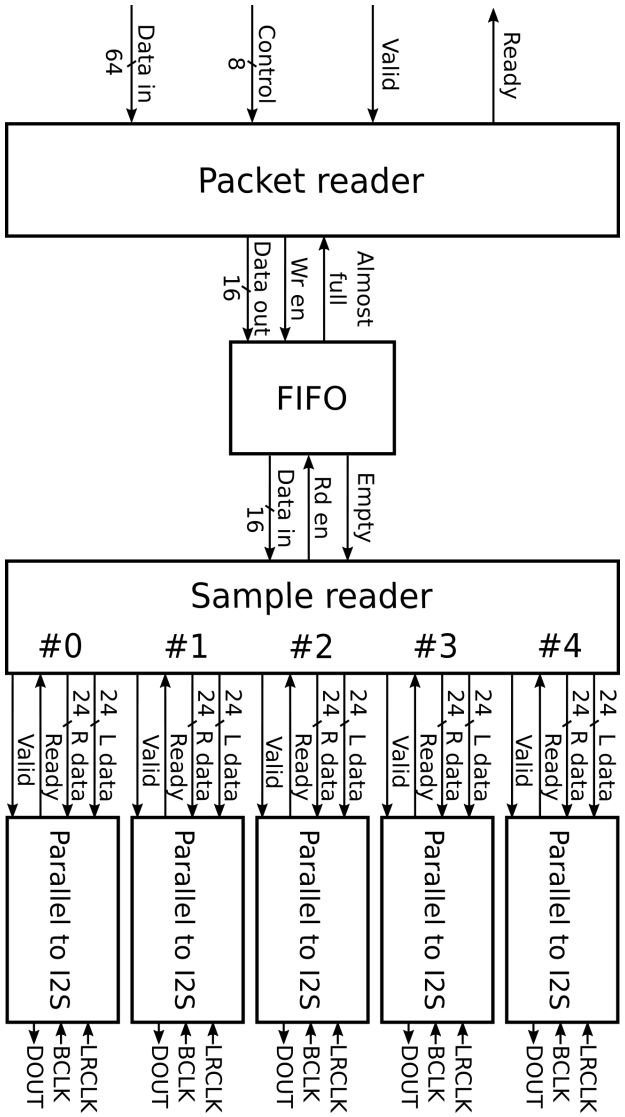
Interface modulu				
#	Názov	Počet bitov	Smer	Popis
1	I2S #0	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
2	I2S #1	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
3	I2S #2	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
4	I2S #3	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
5	I2S #4	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
6	Data	64	in	Vstup pre obsah RTP paketu
7	Control	8	in	Riadiace signály bit 0 - Packet data bit 1 - Packet end bit 2-4 - Packet alignment bit 5 - PHY error bit 6 - Timestamp bit 7 - CRC error
8	Valid	1	in	Dáta pripravené na odoslanie
9	Ready	1	out	Modul pripravený prevziať dáta
10	I2S Cfg	5	in	Aktívne I^2S rozhrania bit 0 - I2S #0 bit 1 - I2S #1 bit 2 - I2S #2 bit 3 - I2S #3 bit 4 - I2S #4
11	Bit Cfg	1	in	Počet prijímaných audio bitov 0 - 16b 1 - 24b
12	CLK	1	in	Hodinový vstup
13	Reset	1	in	Vstup pre reset aktívny v 1

Tabuľka 2.2: RTP packet payload receiver popis rozhraní.

2.1.2.2 Architektúra modulu

Rovnako ako modul RTP packet payload sender prešiel receiver vývojom. Prvý návrh bol bez fronty. Obsahoval iba časť, ktorá prijímala obsah paketov a vzorky rozposielala ďalším blokom, ktoré ich posielali po I^2S zberniciach. Tento prístup sa opäť neosvedčil a fungoval len do chvíle, kedy pakety prichádzali pravidelne.

Bloková schéma modulu je na obrázku 2.6. Všetky bloky majú hodinový vstup synchrónny reset a všetky (okrem fronty) majú aj konfiguračný vstup na šírku audio slov (bit cfg) a na konfiguráciu aktívnych I^2S rozhraní (I2S cfg). Tieto vstupy na obrázku kvôli prehľadnosti nie sú.



Obr. 2.6: RTP Packet Payload Receiver - bloková schéma.

RTP packet receiver sa skladá zo 4 častí:

- *Packet reader blok* - úlohou tohoto bloku je čítať 64b slová z modulu, ktorý zaobstaráva sieťovú komunikáciu. Tieto slová následne rozdelí na 16b slová a zapíše do fronty. Na príjem slov používa ready-valid handshake. Slová do fronty zapisuje až po kým nie je plná alebo neprestanú prichádzať. V prípade, ak je nastavený ktorýkoľvek z error flagov, dáta nie sú prijaté. Taktiež blok nerieši timestamp, takže modul od ktorého RTP packet receiver prijíma dáta, musí riešiť to, či sú pakety v správnom poradí.

- *FIFO* - synchrónna fronta z knižnice *unimacro* od firmy Xilinx. Je realizovaná v pamäti BRAM s veľkosťou 18Kb so šírkou slova 16b. HĺBka fronty je 1024 slov.

- *Sample reader blok* - úlohou tohoto bloku je čítať dáta po 16b slovách z fronty a zapisovať ich do jednotlivých parallel to I^2S blokov. Blok pripraví najskôr vzorky na *L data* a *R data* vstupy všetkých aktívnych blokov parallel to I^2S a až vo chvíli, keď sú všetky dáta pripravené, je nastavený *valid* signál pre všetky bloky naraz. Po nastavení *ready* signálov a prenesení audio vzoriek do parallel to I^2S blokov sú čítané a spracovávané dáta.

  Aby boli dáta správne rozdelené, musí byť nastavená požadovaná bitová šírka a aktivované rozhrania k parallel to I^2S blokom.

  Ak je bitová šírka audio vzoriek 16b, pripadajú na jedno I^2S rozhranie dve slová, v prípade 24b šírky sú to tri slová (1. slovo = najvýznamnejších 16b ľavého kanálu, 2. slovo = najmenej významných 8b ľavého kanálu + najvýznamnejších 8b pravého kanálu, 3. slovo = najmenej významných 16b pravého audio kanálu).

- *Parallel to I^2S blok* - návrh obsahuje 5 takýchto blokov. Ich úlohou je vysielať prijaté 16b alebo 24b audio vzorky po I^2S zbernici. Vzorky sú prijímané od sample reader bloku. Prenos dát prebieha pomocou ready-valid handshaku. Ak by z nejakého dôvodu (napr. stratený alebo poškodený paket) dáta od sample reader bloku neboli prijaté včas, po I^2S zbernici bude poslaná nulová vzorka.

  Každý z parallel to I^2S blokov je možné deaktivovať a resetovať. Je možné nastaviť bitovú šírku vzoriek. Šírka však musí byť jednotná pre všetky bloky, ak má modul správne fungovať.

2.1.2.3 Minimálna hodinová frekvencia modulov s I²S zbernicou

Minimálna hodinová frekvencia modulu musí byť dostatočná, aby sa medzi prijatím alebo odoslaním dvoch po sebe idúcich audio vzoriek po I²S zbernici stihli spracovať a nedochádzalo ku stratám dát. Taktiež hodinová frekvencia musí byť dostatočná na to, aby bolo možné správne prijímať a odosielať dáta po I²S zbernici.

Samotné spracovávanie a pripravovanie dát nie je tak náročné na hodinovú frekvenciu ako I²S zbernica. Pri používaní všetkých I²S zberníc a audio šírke 24b potrebuje RTP packet payload sender 41 hodinových taktov od prijatia dát blokom sample writer, až kým sú všetky dáta odoslané ako obsah paketu ďalšiemu modulu. Teda bloky sample writer, fronta a packet writer by dokázali pracovať pri 7,872 MHz, ak by spracovávali audio dáta so vzorkovacou frekvenciou 192 kHz. RTP packet payload receiver potrebuje 44 hodinových cyklov na rozbalenie obsahu paketu blokom packet reader, prenos frontou a sample reader blokom až na vstupné rozhrania parallel to I²S blokov. Táto časť by dokázala pracovať pri 8,448 MHz, ak bude prenášať 192 kHz audio dáta.

Pri prijímaní dát po I²S sú detekované vzostupné hrany signálu BCLK, pretože vtedy je zaručene platný posielaný dátový bit. Pri odosielaní sú naopak detekované zostupné hrany BCLK a počas nich sú posielané na výstup ďalšie dátové bity. Vstupný signál (BCLK) je zaznamenávaný na vzostupnú hranu hodinového signálu a teda na zachytenie všetkých zmien vstupného signálu musí mať hodinový signál minimálne dvojnásobnú frekvenciu signálu vstupného.

Pri I²S prenose sa zvyčajne používa 32 BCLK cyklov na prenos jedného kanálu. Teda frekvencia BCLK je 64 násobkom vzorkovacej frekvencie audio signálu. V tabuľke 2.3 sú uvedené minimálne hodnoty hodinovej frekvencie pre správne spracovávanie dát na I²S v závislosti od vzorkovacej frekvencie, spoločne s frekvenciou BCLK.

Vzorkovacia frekvencia (kHz)	Frekvencia BCLK (MHz)	Hodinová frekvencia (MHz)
32	2,048	4,096
41,1	2,6304	5,2608
48	3,072	6,144
96	6,144	12,288
192	12,288	24,576

Tabuľka 2.3: Minimálna hodinová frekvencia pre moduly pracujúce s I²S v závislosti od vzorkovacej frekvencie audia.

2.2 Popis modulov pracujúcich s formátom ADAT

2.2.1 ADAT-I²S konvertor

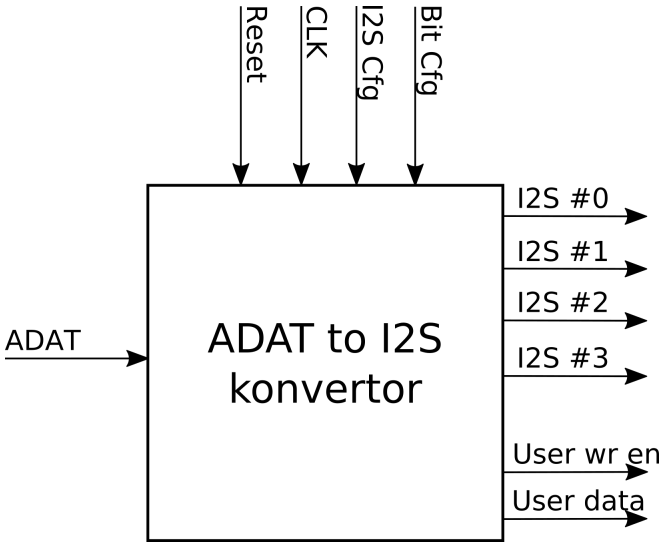
ADAT-I²S konvertor prijíma ADAT stream z fyzického rozhrania, dekóduje z neho ADAT pakety a z nich vyberá audio vzorky, ktoré následne posiela po I²S zberniciach.

Keďže ADAT podporuje prenos 8 audio kanálov ADAT-I2S konvertor má štyri I²S výstupy, ktoré majú vlastnosti totožné s výstupmi v RTP packet payload receiveri.

2.2.1.1 Interface

Interface ADAT-I²S konvertoru je vyobrazený na obrázku 2.7 a popísaný v tabuľke 2.4. Má štyri časti:

- *ADAT vstup* - rozhranie pre prijímanie ADAT audio dát.
- *Konfiguračný interface* (I2S Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku a aktívne I²S rozhrania.
- *I²S interface* - rozhrania pre pripojenie I²S zberníc.
- *User data* - výstup pre user dáta z ADAT paketu.



Obr. 2.7: ADAT to I²S konvertor.

Interface modulu				
#	Účel	Počet bitov	Smer	Popis
1	I2S #0	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
2	I2S #1	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
3	I2S #2	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
4	I2S #3	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
5	I2S Cfg	4	in	Aktívne I2S rozhrania bit 0 - I2S #0 bit 1 - I2S #1 bit 2 - I2S #2 bit 3 - I2S #3
6	Bit Cfg	1	in	Počet prijímaných audio bitov 0 - 16b 1 - 24b
7	ADAT	1	in	Vstup ADAT streamu
8	User data	4	out	Výstup pre ADAT user dáta
9	User data wr en	1	out	Write enable pre user dáta
10	CLK	1	in	Hodinový vstup
11	Reset	1	in	Vstup pre reset aktívny v 1

Tabuľka 2.4: ADAT-I^2S modul popis rozhraní.

2.2.1.2 Architektúra modulu

Bloková schéma modulu je na obrázku 2.8. Všetky bloky majú hodinový vstup, synchrónny reset. Parallel to I^2S bloky majú navyše konfiguračný vstup na šírku audio slov (bit cfg) a povoľovací (I2S cfg) vstup. Tieto vstupy na obrázku kvôli prehľadnosti nie sú.

ADAT-I^2S konvertor sa skladá z 3 častí:

- *ADAT to parallel blok* - úlohou tohoto bloku je dekódovať ADAT pakety, vyberať z nich audio vzorky a posielať ich na výstup v paralelnom formáte, spoločne s číslom kanálu odkiaľ pochádzajú (adresa).

  Blok sa sám synchronizuje s ADAT streamom detekovaním synchronizačnej sekvencie. Funguje to tak, že čas medzi každými dvoma zmenami

hodnoty signálu je zaznamenávaný a porovnávaný s doposiaľ nameraným maximom.

Synchronizačná sekvencia je v pakete len jedna. Skladá sa z desať po sebe idúcich bitov s hodnotou 0, kedy nedochádza k zmenám hodnoty signálu. Mimo nej dochádza aspoň každých 5 bitov k zmene hodnoty signálu a teda táto sekvencia je ľahko detekovateľná. Vďaka tomu je jednoduché zmerať čas vysielania jedného paketu. Čas vysielania jedného bitu je $\frac{1}{256}$ času paketu.

Ďalším krokom je detekovanie samotných bitov. Využije sa pri tom vypočítaný čas jedného bitu a to, že po každých štyroch dátových bitoch nasleduje synchronizačný bit s hodnotou 1. Čas medzi dvoma zmenami signálu je zmeraný a následne podľa toho, do ktorého intervalu jeho dĺžka spadá, sú bity paketu dekódované podľa tabuľky 2.5.

Časový interval (čas / čas bitu)	Počet bitov	Hodnota bitov
0.5 - 1.5	1	1
1.5 - 2.5	2	01
2.5 - 3.5	3	001
3.5 - 4.5	4	0001
4.5 - 5.5	5	00001

Tabuľka 2.5: ADAT dekódovacia tabuľka.

Taktiež je potrebné zaznamenávať, v akom časovom úseku paketu sa dekódované dáta nachádzali, aby ich bolo možné správne interpretovať. Prvých 5 bitov po synchronizačnej sekvencii budú dáta interpretované ako user dáta, nasledujúcich 30 bitov ako vzorka z kanála 0, ďalších 30 bitov ako vzorka z kanála 1 atď.

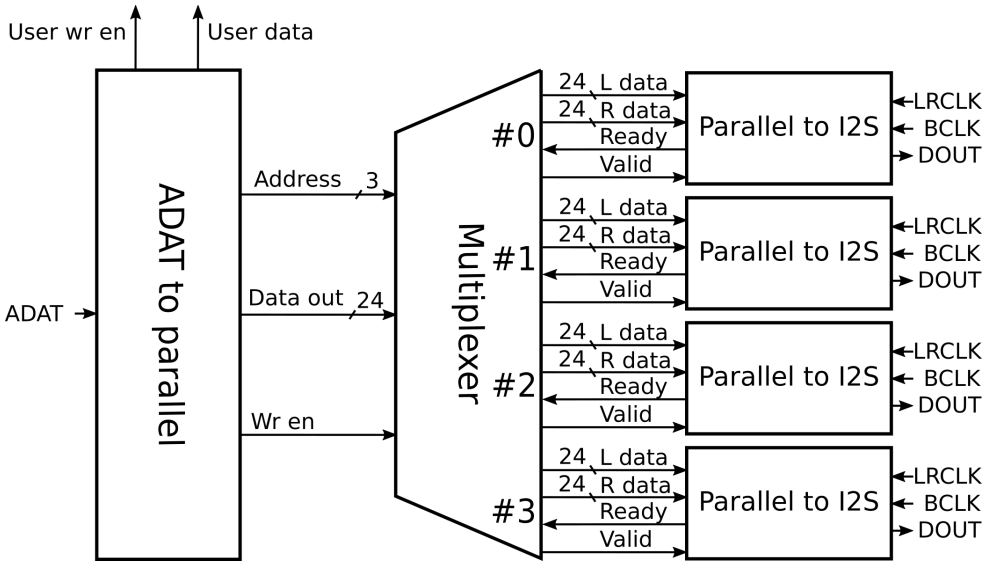
Najväčšou výzvou pri tomto bloku bolo dosiahnuť to, aby sa modul dokázal sám synchronizovať s prijímanými dátami aj v prípade, že na vstup dlhšiu dobu dáta neprichádzajú. Ak by to nedokázal, bolo by nevyhnutné resetovať zariadenie pri každom pripojení optického kábla.

Ako už bolo spomenuté, synchronizácia je založená na detekcii najdlhšej doby, počas ktorej nedochádza k zmene signálu. Ak istú dobu nemá modul pripojený na vstup zdroj dát, bude táto doba vyhodnotená ako synchronizačná sekvencia a ďalšie dekódovanie ADAT streamu bude chybné.

Prvý nápad ako to vyriešiť bol v pravidelných intervaloch zahodiť nameranú synchronizačnú hodnotu. Toto však spôsobí, že až do najbližšej synchronizačnej sekvencie budú dáta nesprávne dekódované. Ďalšia

možnosť je staticky určiť túto dobu v závislosti na hodinovej frekvencii. Tento prístup funguje dobre, ale je neprenosný a nedovoľuje meniť hodinovú frekvenciu. Ako najlepšie sa osvedčilo riešenie, kedy sa nameraná maximálna doba medzi dvoma zmenami signálu znižuje o 1, ak 2^{16} po sebe idúcich dôb medzi zmenami nedosiahne túto hodnotu. Modul sa takto synchronizuje behom niekoľkých milisekúnd a následne už dekóduje dáta správne.

- *Multiplexer* - v závislosti na adrese, udávanej ADAT to parallel blokom zapisuje 24b dáta do jednotlivých parallel to I^{2}S blokov.
- *Parallel to I^{2}S blok* - návrh obsahuje 4 takéto bloky, ktoré sú totožné s tými popísanými v časti 2.1.2.2. Ich úlohou je posielať audio vzorky po I^{2}S zbernici. Vzorky sú prijímané od ADAT to parallel bloku.



Obr. 2.8: ADAT to I^{2}S konvertor - bloková schéma.

2.2.1.3 Minimálna hodinová frekvencia modulu

Minimálna hodinová frekvencia je zdola obmedzená rýchlosťou vysielania ADAT dát, čo je 12,288 Mbps. Na to, aby modul správne fungoval a dekódoval dáta z ADAT, je nutné, aby jeho hodinová frekvencia bola aspoň 8 násobná. Teda 96,304 MHz.

2.2.2 I²S-ADAT konvertor

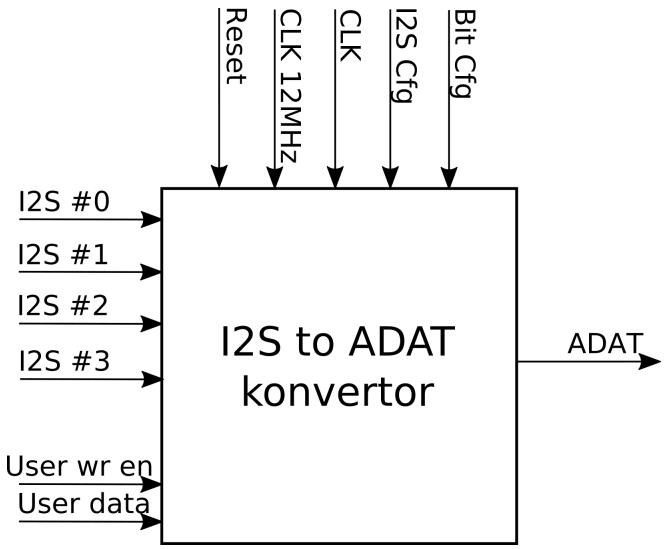
I²S-ADAT konvertor prijíma audio vzorky po I²S zberniciach, vytvára z nich ADAT pakety, kóduje ich do NRZI a posiela ako ADAT steam.

Keďže ADAT podporuje prenos 8 kanálov ADAT-I2S, má konvertor štyri I²S vstupy, ktoré majú vlastnosti totožné so vstupmi v RTP packet payload senderi.

2.2.2.1 Interface

Interface I²S-ADAT konvertoru je vyobrazený na obrázku 2.9 a popísaný v tabuľke 2.6. Má štyri časti:

- *I²S interface* - rozhrania pre pripojenie I²S zberníc.
- *Konfiguračný interface* (I2S Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku a aktívne I²S rozhrania.
- *User data* - vstup pre user dáta v ADAT pakete.
- *ADAT výstup* - rozhranie pre vysielanie ADAT audio dát.



Obr. 2.9: I²S to ADAT konvertor.

Interface modulu				
#	Názov	Počet bitov	Smer	Popis
1	I2S #0	3	in	I^{2}S vstup (DIN, BCLK, LRCLK)
2	I2S #1	3	in	I^{2}S vstup (DIN, BCLK, LRCLK)
3	I2S #2	3	in	I^{2}S vstup (DIN, BCLK, LRCLK)
4	I2S #3	3	in	I^{2}S vstup (DIN, BCLK, LRCLK)
5	I2S Cfg	4	in	Aktívne I2S rozhrania bit 0 - I2S #0 bit 1 - I2S #1 bit 2 - I2S #2 bit 3 - I2S #3
6	Bit Cfg	1	in	Počet prijímaných audio bitov 0 - 16b 1 - 24b
7	ADAT	1	out	Výstup ADAT streamu
8	User data	4	in	Vstup pre ADAT user dáta
9	User data wr en	1	in	Write enable pre user dáta
10	Reset	1	in	Vstup pre reset aktívny v 1
11	CLK	1	in	Hodinový vstup
12	CLK 12MHz	1	in	Riadiaci signál s frekvenciou 12,288 MHz

Tabuľka 2.6: I^{2}S-ADAT modul popis rozhraní.

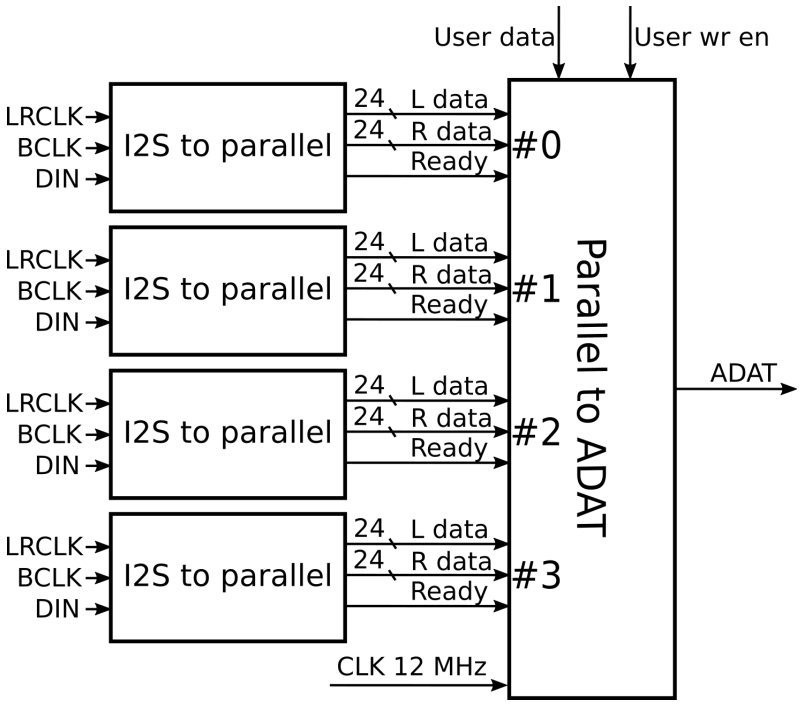
2.2.2.2 Architektúra modulu

Bloková schéma modulu je na obrázku 2.10. Všetky bloky majú hodinový vstup a synchrónny reset. I^{2}S to parallel bloky majú navyše konfiguračný vstup na šírku audio slov (bit cfg) a povoľovací (I2S cfg) vstup. Tieto vstupy na obrázku pre prehľadnosť nie sú.

- *I^{2}S to parallel blok* - návrh obsahuje 4 takéto bloky, ktoré sú totožné s tými popísanými v časti 2.1.1.2. Ich úlohou je prijímať audio vzorky po I^{2}S zbernici a zapisovať ich do parallel to ADAT bloku.

- *Parallel to ADAT blok* - úlohou tohoto bloku je prijímať 24b audio vzorky a vytvárať z nich ADAT pakety. Tie následne zakódovať a posielať ich na ADAT výstup. Blok je značne jednoduchší ako *ADAT to parallel*, potrebuje však presný riadiaci signál s frekvenciou 12,288 MHz.

  Parallel to ADAT blok zoradí prijaté audio vzorky jednu za druhou, pridá user bity, synchronizačné bity a synchronizačnú sekvenciu. Takto vytvorený paket vysiela po jednom bite pri vzostupnej hrane 12,288 MHz riadiaceho signálu.



Obr. 2.10: I^2S to ADAT konvertor - bloková schéma.

2.2.2.3 Minimálna hodinová frekvencia modulu

Minimálna hodinová frekvencia je zdola obmedzená rýchlosťou vysielania ADAT dát, čo je 12,288 Mbps. Pri vysielaní sa detekuje vzostupná hrana riadiaceho 12,288 MHz signálu, počas čoho sa vysielanie posunie o jeden bit ďalej. Minimálna hodinová frekvencia v tomto prípade musí byť aspoň dvojnásobná, a to je 24,576 MHz.

2.3 Popis modulov na prenos audia spoločne s videom

Časť z modulov pre prenos audia spoločne s videom je už implementovaná. Jedná sa o audio multiplexer a demultiplexer pre 3G-SDI, dual-link HD-SDI a HD-SDI video od spoločnosti Xilinx popísané v časti 1.4.5 a 1.4.6. Moduly sú voľne dostupné a na to, aby ich bolo možné použiť s modulmi z tejto práce boli doimplementované adaptéry konvertujúce I^{2}S na AES3 a opačne.

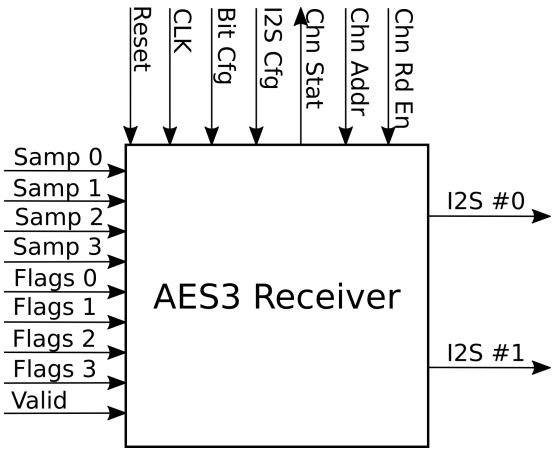
2.3.1 Embedded audio receiver

Audio vo formáte AES3 je z 3G-SDI videa získané pomocou audio demultiplexeru. Získané audio vzorky následne *AES3 receiver* posiela po I^{2}S zberniciach. Demultiplexer podporuje prenos 4 audio kanálov, preto má AES3 receiver dva I^{2}S výstupy, ktoré majú vlastnosti totožné s výstupmi v RTP packet payload receiveri. Ďalej je popisovaný len *AES3 receiver*.

2.3.1.1 Interface

Interface AES3 receiveru je vyobrazený na obrázku 2.11 a popísaný v tabuľke 2.7. Má štyri časti:

- *I^{2}S interface* - rozhrania pre pripojenie I^{2}S zberníc.
- *Konfiguračný interface* (I2S Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku a aktívne I^{2}S rozhrania.
- *Channel status rozhranie* - výstup pre channel status prijímaný spoločne s AES3 framami od demultiplexeru.
- *AES3 interface* - rozhranie pre prijímanie audio dát od demultiplexeru.



Obr. 2.11: AES3 receiver.

Interface modulu				
#	Názov	Počet bitov	Smer	Popis
1	I2S #0	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
2	I2S #1	3	out	I^2S výstup (DOUT(out), BCLK(in), LRCLK(in))
3	I2S Cfg	2	in	Aktívne I2S rozhrania bit 0 - I2S #0 bit 1 - I2S #1
4	Bit Cfg	1	in	Počet odosielaných audio bitov 0 - 16b 1 - 24b
5	Chn Stat	192	out	Channel status register výstup
6	Chn Addr	2	in	Adresa channel status registru
7	Chn Rd En	1	in	Channel status read enable vstup
8	Samp 0	24	in	Vstup pre audio vzorky 0
9	Samp 1	24	in	Vstup pre audio vzorky 1
10	Samp 2	24	in	Vstup pre audio vzorky 2
11	Samp 3	24	in	Vstup pre audio vzorky 3
12	Flags 0	4	in	Vstup pre flagy vzorky 0 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit bit 3 - Z - Z bit pre kanály 0 a 1
13	Flags 1	4	in	Vstup pre flagy vzorky 1 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit
14	Flags 2	4	in	Vstup pre flagy vzorky 2 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit bit 3 - Z - Z bit pre kanály 2 a 3
15	Flags 3	4	in	Vstup pre flagy vzorky 3 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit
16	Valid	1	in	Valid vstup. Ak sú dáta na Samp a Flags aktuálne je nastavený na 1 na jeden hodinový cyklus.
17	Reset	1	in	Vstup pre reset aktívny v 1
18	CLK	1	in	Hodinový vstup

Tabuľka 2.7: AES3 receiver modul popis rozhraní.

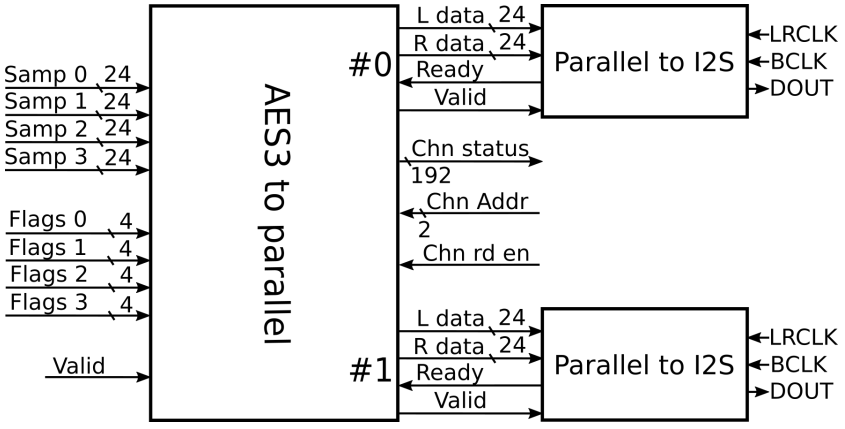
2.3.1.2 Architektúra modulu

Bloková schéma modulu je na obrázku 2.12. Všetky bloky majú hodinový vstup a synchrónny reset. Parallel to I^{2}S bloky majú navyše konfiguračný vstup na šírku audio slov (bit cfg) a povoľovací (I2S cfg) vstup. Tieto vstupy na obrázku pre prehľadnosť nie sú.

- *AES3 to parallel blok* - prijíma AES3 framy pre 4 audio kanály od audio demultipelexeru. Z framu vyjme audio vzorky a zapíše ich do parallel to I^{2}S blokov.

  AES3 frame prijímaný od audio demultiplexeru si so sebou nesie validity bit (V), user data bit (U), channel status bit (C) a Z bit. V prípade, že je validity bit nastavený na 0, nie je vzorka z príslušného framu prijatá. User data bit sa v bloku nepoužíva. Bit Z nastavený na 1 určuje pozíciu prvého framu v AES3 bloku a označuje prvý bit channel statusu. Channel status bity sú ukladané a 192b channel status je možné z modulu prečítať po channel status rozhraní nastavením adresy kanálu a read enable bitu. Prvý prijatý status nemusí byť správny, keďže modul potrebuje prijať aspoň jeden Z bit hodnoty 1 pre správnu synchronizáciu.

- *Parallel to I^{2}S blok* - návrh obsahuje 2 takéto bloky, ktoré sú totožné s tými popísanými v časti 2.1.2.2. Ich úlohou je vysielať prijaté audio vzorky po I^{2}S zbernici.



Obr. 2.12: AES3 receiver - boková schéma.

2.3.1.3 Minimálna hodinová frekvencia modulu AES3 receiver

Minimálna hodinová frekvencia je obmedzená I^{2}S zbernicou. Odôvodnenie je v časti 2.1.2.3.

2.3.2 Embedded audio sender

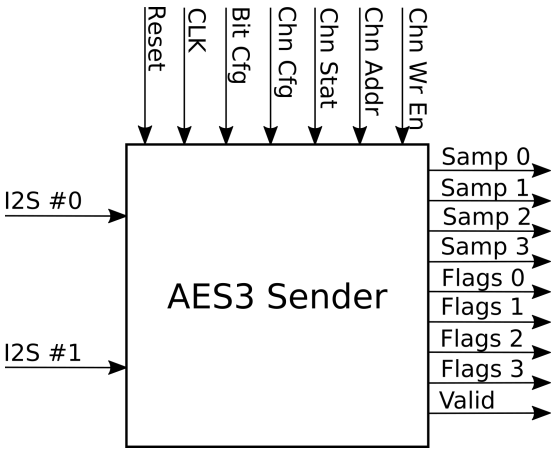
AES3 sender získava audio vzorky po I^{2}S zberniciach, pridáva im AES3 flagy a posiela ich audio multiplexeru. Ten prijíma 3G-SDI video, do ktorého ich vkladá.

Multiplexer podporuje prenos 4 audio kanálov, preto má AES3 sender dva I^{2}S vstupy, ktoré majú vlastnosti totožné so vstupmi v RTP packet payload senderi. Ďalej je popisovaný len *AES3 receiver*.

2.3.2.1 Interface

Interface AES3 senderu je vyobrazený na obrázku 2.13 a popísaný v tabuľke 2.8. Má štyri časti:

- *I^{2}S interface* - rozhrania pre pripojenie I^{2}S zberníc.
- *Konfiguračný interface* (Chn Cfg, Bit Cfg) - toto rozhranie definuje bitovú šírku I^{2}S rozhraní a aktívne AES3 kanály.
- *Channel status rozhranie* - vstup pre channel status, ktorý bude odoslaný po jednom bite do multiplexeru.
- *AES3 interface* - rozhranie pre odosielanie AES3 audio framov do multiplexeru.



Obr. 2.13: AES3 sender.

Interface modulu				
#	Názov	Počet bitov	Smer	Popis
1	I2S #0	3	in	I^2S vstup (DIN, BCLK, LRCLK)
2	I2S #1	3	in	I^2S vstup (DIN, BCLK, LRCLK)
3	Chn Cfg	4	in	Aktívne AES3 kanály bit 0 - Samp 0 bit 1 - Samp 1 bit 2 - Samp 2 bit 3 - Samp 3
4	Bit Cfg	1	in	Počet odosielaných audio bitov 0 - 16b 1 - 24b
5	Chn Stat	192	in	Channel status register výstup
6	Chn Addr	2	in	Adresa channel status registru
7	Chn Wr En	1	in	Write enable vstup
8	Samp 0	24	out	Výstup pre audio vzorky 0
9	Samp 1	24	out	Výstup pre audio vzorky 1
10	Samp 2	24	out	Výstup pre audio vzorky 2
11	Samp 3	24	out	Výstup pre audio vzorky 3
12	Flags 0	4	out	Výstup pre flagy vzorky 0 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit bit 3 - Z - Z bit pre kanály 0 a 1
13	Flags 1	4	out	Výstup pre flagy vzorky 1 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit
14	Flags 2	4	out	Výstup pre flagy vzorky 2 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit bit 3 - Z - Z bit pre kanály 2 a 3
15	Flags 3	4	out	Výstup pre flagy vzorky 3 bit 0 - V - Valid bit bit 1 - U - User bit bit 2 - C - Channel status bit
16	Valid	1	out	Valid výstup. Ak sú dáta na Samp a Flags aktuálne je nastavený na 1 na jeden hodinový cyklus.
17	Reset	1	in	Vstup pre reset aktívny v 1
18	CLK	1	in	Hodinový vstup

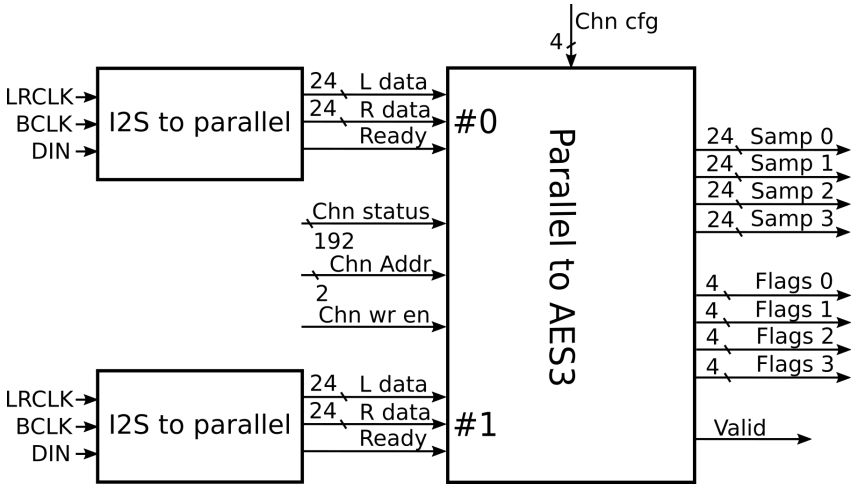
Tabuľka 2.8: AES3 sender modul popis rozhraní.

2.3.2.2 Architektúra modulu

Bloková schéma modulu je na obrázku 2.14. Všetky bloky majú hodinový vstup a synchrónny reset. I^{2}S to parallel bloky majú navyše konfiguračný vstup na šírku audio slov (bit cfg) a povoľovací (I2S cfg) vstup, ktorý zodpovedá Chn cfg vstupu AES3 to sender modulu. Tieto vstupy na obrázku pre prehľadnosť nie sú.

- *I^{2}S to parallel blok* - návrh obsahuje 2 takéto bloky, ktoré sú totožné s tými popísanými v časti 2.1.1.2. Ich úlohou je prijímať audio vzorky po I^{2}S zbernici.
- *Parallel to AES3 blok* - prijíma audio vzorky od I^{2}S to parallel blokov a vkladá ich do AES3 framov. Tieto framy nastaví na výstup pre audio multiplexer. Valid flag je nastavený na 1, počas jednej hodinovej periódy až po prijatí dát od všetkých aktívnych I^{2}S to parallel blokov. Takto sú zapísané všetky AES3 framy do audio multiplexeru naraz.

  AES3 frame odosielaný do audio multiplexeru si so sebou nesie aj validity bit (V), user data bit (U), channel status bit (C) a Z bit. V prípade, že je kanál povolený, je validity bit nastavený na 1. User data bit sa v bloku nepoužíva a je nastavený na 0. Bit Z je nastavený na 1 počas prvého framu v AES3 bloku. C bit je postupne počas 192 framov v bloku nastavovaný podľa bitov channel status registru.



Obr. 2.14: AES3 sender - bloková schéma.

2.3.2.3 Minimálna hodinová frekvencia modulu AES3 sender

Minimálna hodinová frekvencia je obmedzená I^{2}S zbernicou. Odôvodnenie je v časti 2.1.2.3.

Kapitola 3

Testovanie

Pri vývoji aplikácii pre FPGA patrí testovanie k náročným úlohám. Je to najmä z dôvodu, že syntéza a implementácia obvodu sú časovo náročné operácie. Taktiež debugging hardwaru nie je tak jednoduchý ako pri software. Preto je testovanie a ladenie vytvoreného designu na vývojovej doske neefektívne. Využíva sa na to softwareová simulácia, ktorá umožňuje získať výsledky jednoduchšie a v neporovnateľne kratšom čase, najmä v začiatkoch vývoja. Následne je takto odladený design otestovaný na vývojovej doske alebo prototype.

Táto kapitola popisuje simulácie modulov, ich integráciu a následné testovanie na hardwarovom prototype.

3.1 Simulácia modulov

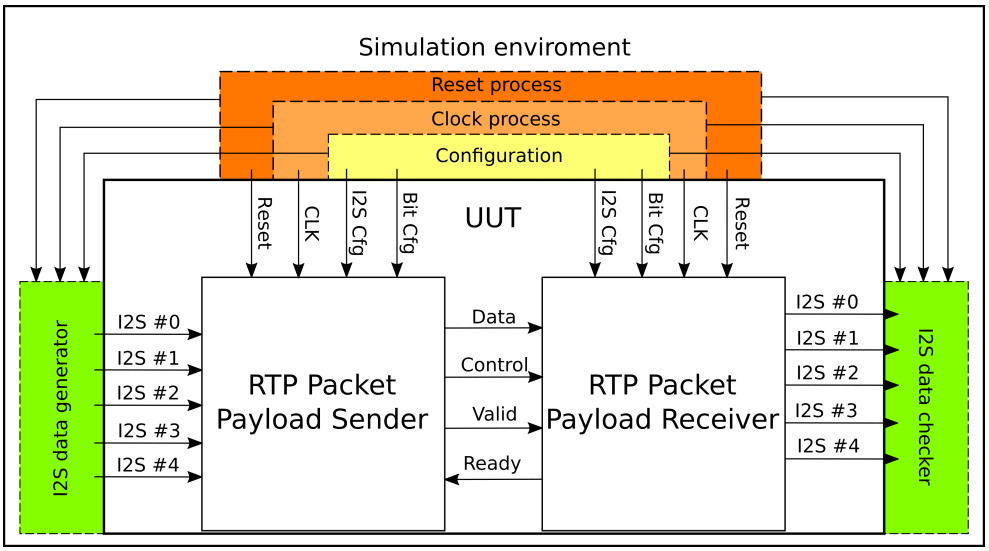
V priebehu vytvárania modulov bol každý simulovaný na viacerých úrovniach. Každý blok každého modulu bol simulovaný samostatne a výsledky boli vyhodnocované ručne. Keďže sa jedná zväčša o jednoduché bloky, vývoj automatického testovania by zabral zbytočne mnoho času. Následne bol modul zložený z blokov simulovaný ako celok a na koniec v kooperácii s inými modulmi. Pre samotné moduly a moduly v kooperácii s inými boli napísané automatické testy. Simulácia VHDL modulov bola vykonávaná v simulátore, ktorý je súčasťou vývojového prostredia Vivado Design Suite od spoločnosti Xilinx.

3.1.1 Príklad simulácie modulov RTP payload sender a receiver v kooperácii

Pre simuláciu bolo navrhnuté verifikačné prostredie (*testbench*) implementované v jazyku VHDL. Architektúra simulačného prostredia je na obrázku 3.1.

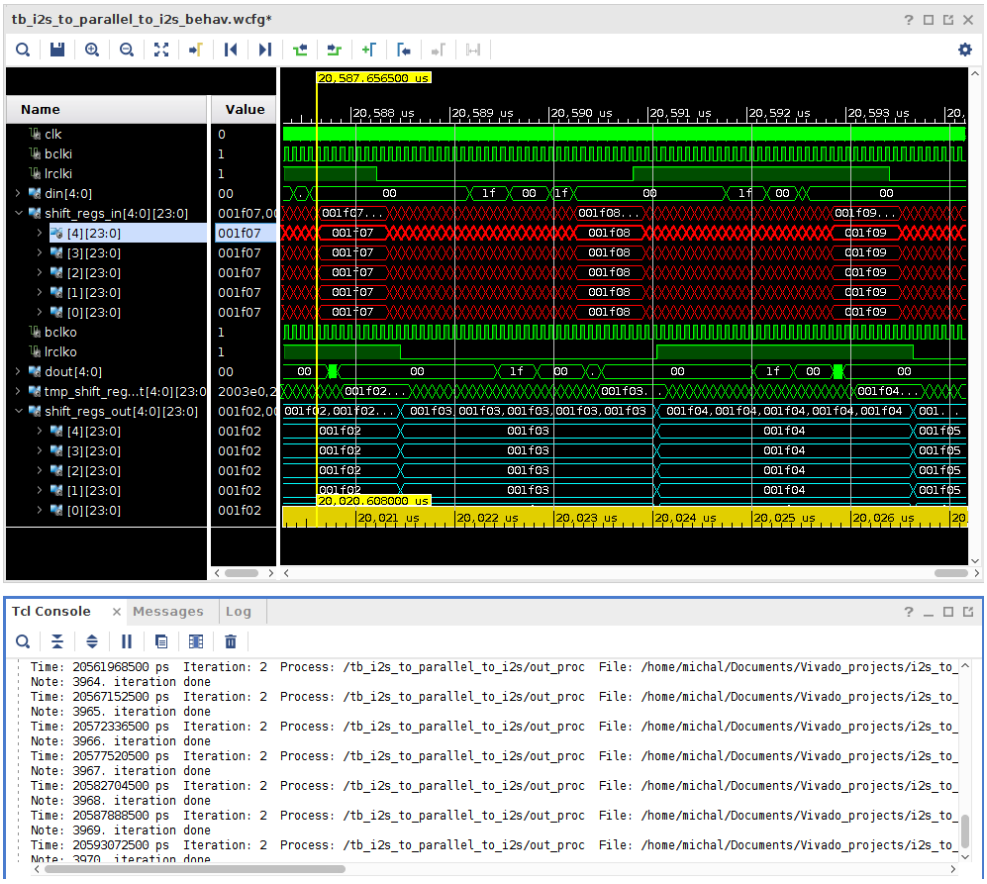
Testbench sa skladá z piatich častí a inštancie testovaných kooperujúcich modulov:

- *Unit under test (UUT)* - Testovaná jednotka. V tomto príklade sa jedná o jednu inštanciu RTP payload senderu a RTP payload receiveru. Dáta teda vstupujú do RTP payload senderu po I^2S, sú zabalené ako obsah RTP paketu, poslané RTP payload receiveru, kde sú rozbalené a poslané po I^2S zberniciach von z modulu. Ak je všetko v poriadku objavia sa dáta ktoré boli na vstupe na výstupe po maximálne dvoch LRCLK periódach.
- *Reset process* - generuje reset aktívny v 1 na začiatku testu po dobu 10 hodinových cyklov. Následne nastaví reset na 0 a do konca testu je nečinný.
- *Clock process* - generuje hodinový signál s frekvenciou 125 MHz.
- *Configuration* - nastavenie aktívnych I^2S rozhraní a bitovej šírky prijímaných audio vzoriek.
- *I^2S data generator* - Simuluje I^2S protokol. Skladá sa z troch procesov. Prvý generuje BCLK hodiny s frekvenciou 12,288 MHz. Druhý LRCLK hodiny s frekvenciou 192 kHz. Posledný vytvára dáta, ktoré následne po jednom bite vkladá na všetky aktívne I^2S zbernice.
- *I^2S data checker* - Kontroluje dáta na výstupe UUT. Skladá sa z troch procesov. Prvé dva rovnako ako v I^2S data generatore generujú BCLK a LRCLK s rovnakou frekvenciou avšak posunuté vzhľadom na tie, ktoré generuje I^2S data generator. Posledný proces po jednom bite prijíma dáta posielané I^2S zbernicami a po prijatí celej audio vzorky ju porovná s očakávanou. Ak sa nezhodujú ukončí test s chybou. Všetky kontroly správnosti sú vykonávané pomocou VHDL príkazu *assert*.



Obr. 3.1: Simulačné prostredie pre simuláciu RTP modulov v kooperácii.

Výsledok tejto simulácie je na obrázku 3.2, ktorý zobrazuje jednu periódu LRCLK signálu. Vstupné dáta sú vkladané na DIN I^{2}S zberníc (červené) a výstupné dáta očakávané na DOUT aktívnych I^{2}S zberníc (modré). V spodnej časti obrázku v Tcl konzoly je vidieť počet úspešne prenesených audio vzoriek. V prípade, že došlo k chybe sa simulácia preruší a v Tcl konzoly sa objaví čas chyby a krátky popis.



Obr. 3.2: Snímok zo simulácie kooperujúcich modulov.

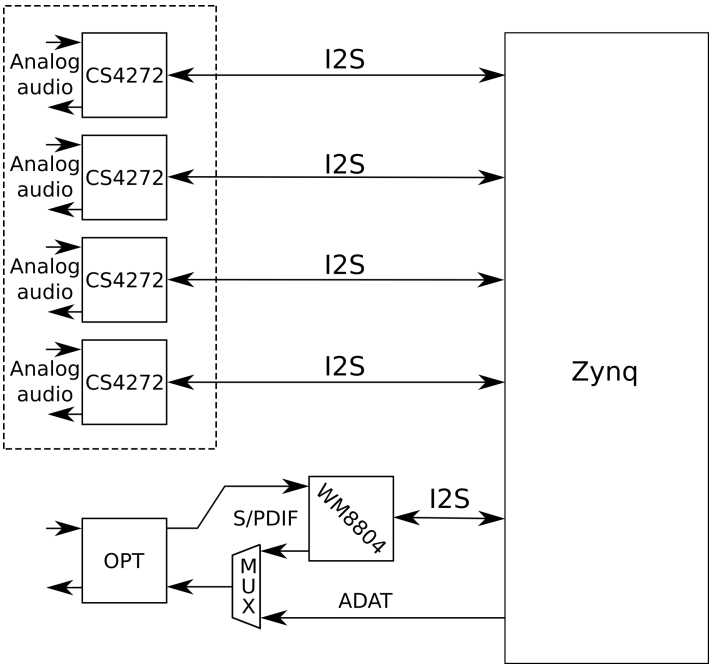
Simulácie všetkých ostatných modulov sú podobné. Líšia sa len protokolom na vstupe/výstupe a rozhraním. Preto ich v tejto práci ďalej popisovať nebudem.

3.2 Testovanie modulov na HW prototype

Produkt Audio MVTP, na ktorom má byť implementovaný výsledok tejto práce, je stále vo vývoji. Po dokončení má prijímať analógové audio, digitálne audio ako S/PDIF alebo ADAT, po ethernete v RTP paketoch alebo v pomocnej časti 3G-SDI videa. Audio umožňuje rovnakými kanálmi aj odosielať.

3.2.1 Popis audio časti prototypu

Z hľadiska audia má prototyp dve časti - analógovú a digitálnu. Obe časti sú od seba galvanicky oddelené. Na obrázku 3.3 predstavuje galvanické oddelenie prerušovaná čiara s analógovou časťou vo vnúti tejto oblasti.



Obr. 3.3: Bloková schéma audio časti prototypu.

V analógovej časti sa nachádzajú štyri CS4272 audio kodeky a ďalšie analógové obvody. Súčasťou tejto časti je fázový záves (PLL - Phase-Locked Loop). Audio dáta sú získavané priamo vzorkovaním analógového signálu čipmi CS4272 a následne poslané po I^2S zberniciach Zynqu. Pri vysielaní sú dáta zo Zynqu poslané po I^2S čipom CS4272 a prevedené na analógový signál.

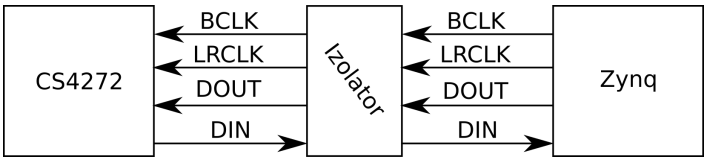
Čipy CS4272 sú nakonfigurované v slave móde a potrebujú externe generované LRCLK, BCLK a MCLK. Na tento účel je použitý fázový záves, ktorý generuje hodinový signál (MCLK) pre CS4272 čipy s frekvenciou 49,152 MHz. Signál frekvencie 24,576 Mhz je privádzaný z PLL do Zynqu, kde sú z neho deličkami frekvencie vytvárané I^2S signály LRCLK a BCLK. Frekvencie oboch

signálov závisia od požadovanej vzorkovacej frekvencie. LRCLK má vždy frekvenciu rovnú vzorkovacej (f_{samp}) a BCLK má frekvenciu $64 * f_{samp}$. Konfigurácia kodekov i PLL prebieha po I^2C zbernici.

V digitálnej časti sa nachádza Zynq a kodek WM8804. Digitálne dáta v tejto časti sú prijímané a vysielané po optickom rozhraní. Vstup vysielacej časti optického rozhrania je pripojený pomocou multiplexeru buď k WM8804 alebo k Zynqu. Výstup prijímacej časti optického rozhrania je pripojený len k WM8804. Toto umožňuje prijímať a vysielať S/PDIF kodekom WM8804 a prijaté dáta posielať po I^2S Zynqu alebo vysielať dáta po optickom rozhraní priamo zo Zynqu. Takto je možné implementovať ADAT kodek priamo v FPGA logike. Na to, aby bolo možné prijímať ADAT, však budú nutné hardwarové zmeny na zariadení. V súčasnom stave je možné vysielať S/PDIF aj ADAT, ale prijímať je možné len S/PDIF. WM8804 je v master mode a externe generované BCLK a LRCLK nepotrebuje, pretože vytvára vlastné. Konfigurácia WM8804 prebieha taktiež po I^2C zbernici.

3.2.1.1 Galvanické oddelenie analógovej a digitálnej časti

Na galvanické oddelenie analógovej a digitálnej audio časti boli použité izolátory, ktoré nezanedbateľne oneskorujú prechádzajúci signál (podľa datasheetu o čas (t_p) 12 ns až 18 ns). Cez izolátor prechádzajú signály I^2S zbernice LRCLK, BCLK, DOUT a DIN (obr.3.4). Signály LRCLK, BCLK a DOUT, generované Zynqom, s oneskorením prichádzajú k CS4272 čipom, ktoré ako odpoveď na zmeny týchto signálov posielajú Zynqu bity audio vzoriek po DIN. Celkovo teda prichádza DIN signál Zynqu s oneskorením viac ako $2*t_p$ (do oneskorenia prispieva aj prechod signálu cez metastabilný filter na strane Zynqu a čas, ktorý potrebuje CS4272 na detekciu zmeny BCLK a LRCLK a vyslanie ďalšieho bitu audio vzorky).

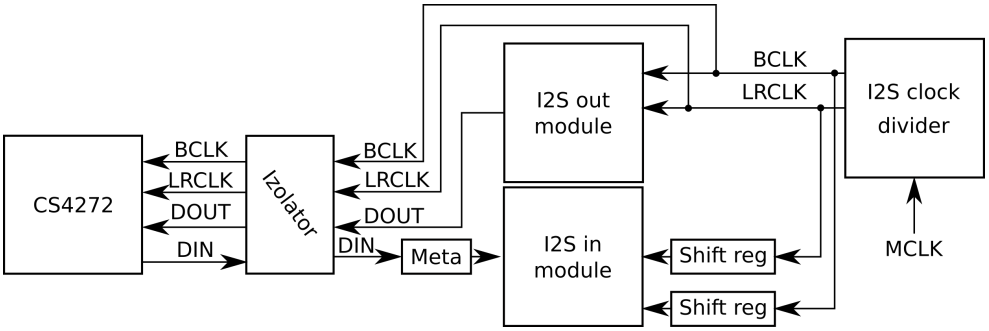


Obr. 3.4: Prechod signálov cez izolátor.

Testom sa zistilo, že celkové oneskorenie DIN signálu nie je zanedbateľné a hodnoty vzoriek prijímané na strane Zynqu neodpovedajú vysielaným hodnotám. Preto musí byť toto oneskorenie kompenzované.

Keďže všetky vytvorené bloky používajú I^2S zbernicu v slave mode a moduly vždy len posielajú alebo len prijímajú dáta je možné mať oddelené LRCLK a BCLK signály pre moduly prijímajúce dáta a moduly posielajúce dáta. Pre prijímajúci modul je možné oneskoriť signály tak, aby sa vyrovnali oneskoreniu signálu DIN. Sú pri tom použité posuvné registre (*shift registers*).

Výpočtom a následným testom sa zistilo, že pre správne oneskorenie je vhodné použiť posuvné registre s 9 klopnými obvodmi za sebou. Pri tomto prístupe nezávisí dĺžka posuvného registru od vzorkovacej frekvencie audio signálu, ale len od oneskorenia na izolátore. Bloková schéma riešenia tohoto problému je na obrázku 3.5.



Obr. 3.5: Riešenie oneskorenia na izolátore.

Druhou možnosťou je oneskorovať DIN o dobu jednej periódy signálu LRCLK zmenšenú o $2 * t_{\mathrm{p}}$ a teda vkladať dáta na vstup modulu o jednu vzorkovaciu periódu neskôr. Toto riešenie však vyžaduje omnoho dlhšie posuvné registre. Navyše počet klopných obvodov v nich závisí na vzorkovacej frekvencii (v prípade 192 kHz vzorkovacej frekvencie je to približne 620 klopných obvodov na jeden posuvný register a so znižujúcou sa vzorkovacou frekvenciou ich počet úmerne rastie).

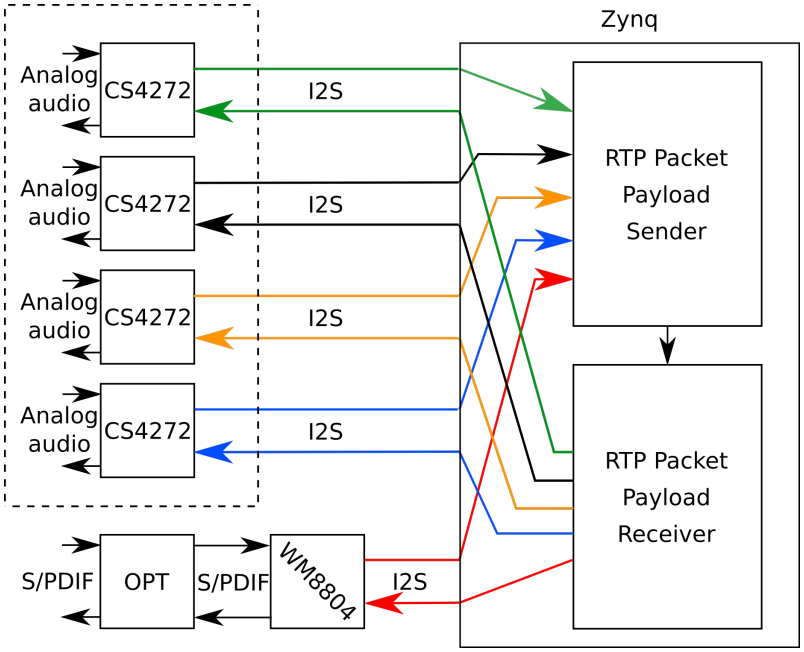
3.2.2 Vstupy použité pri testovaní

Ľudské ucho je dostatočne dokonalé na to, aby odhalilo chyby v prehrávanom zvuku. Ale len málo ľudí dokáže z počutia zistiť, čo so zvukom nie je v poriadku. Preto bol ako testovací vstup použitý harmonický signál generovaný signálovým generátorom a výstup bol pozorovaný na osciloskope. Takto sa dá ľahko zistiť, či je signál degenerovaný alebo nie. Až po testovaní harmonickou funkciou vytváranou generátorom bola na testovanie použitá hudba.

3.2.3 Popis testov

3.2.3.1 Testovanie modulov na prenos zvuku počítačovou sieťou

Test pracuje s oboma modulmi RTP packet payload sender i receiver naraz. Blokový diagram je na obrázku 3.6. Pri teste sú audio dáta prijímané od všetkých CS4272 čipov i WM8804. Sú zabalené RTP packet payload senderom ako obsah RTP paketu a poslané na vstup RTP packet payload receiveru, kde sú rozbalené a opäť poslané čipom, od ktorých boli prijaté. Na diagrame je to znázornené farbou I^2S zberníc. Z toho vyplýva, že dáta prijaté od prvého čipu po zelenej zbernici, sú opäť posielané tomuto čipu naspäť. Vzorkovacia frekvencia všetkých čipov je 48 kHz.

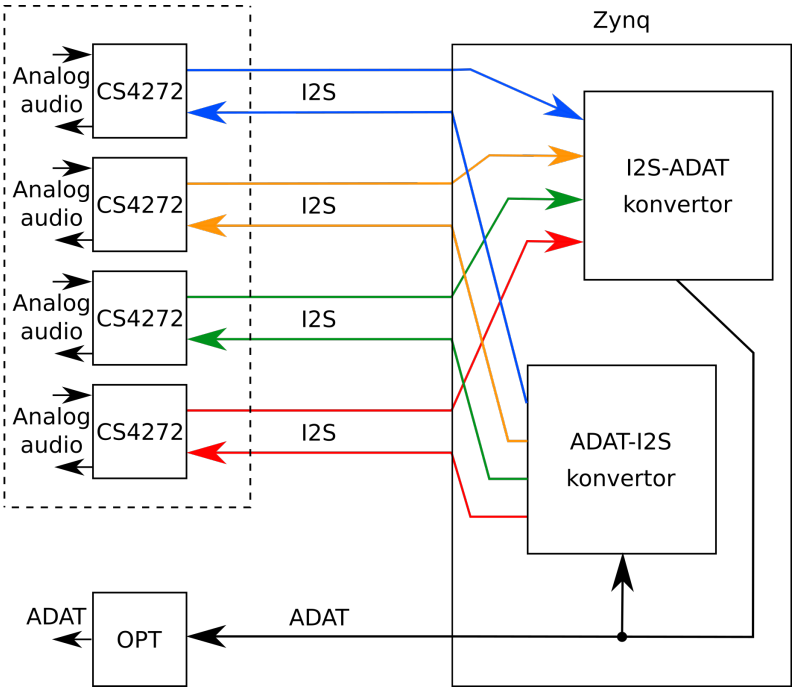


Obr. 3.6: Blokový diagram testovania modulov na prenos zvuku počítačovou sieťou.

Testovaný bol aj prípad, kedy boli zbernice cyklicky posunuté, teda na vstup prvého CS4272 čipu bol privedený výstup z WM8804, na vstup druhého CS4272 bol privedený výstup z prvého atď. Pri ďalšom teste boli použité len čipy CS4272 a vzorkovacia frekvencia bola 192 kHz. WM8804 nebol použitý pretože využívané externé A/D a D/A prevodníky na S/PDIF dokázali pracovať len pri vzorkovacej frekvencii 48 kHz.

3.2.3.2 Testovanie modulov pracujúcich s formátom ADAT

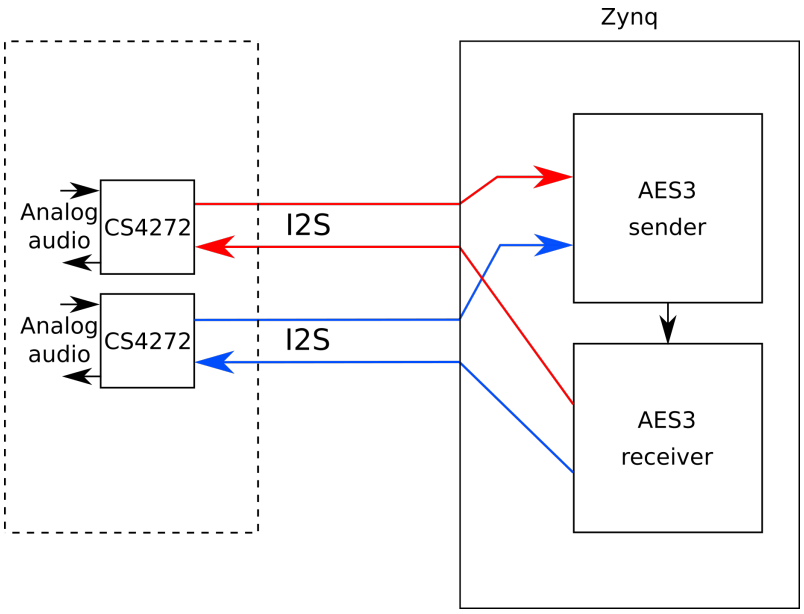
Test testuje oba konvertory I^{2}S-ADAT i ADAT-I^{2}S naraz. Blokový diagram je na obrázku 3.7. Audio dáta sú prijímané od všetkých CS4272 čipov, sú konvertované na ADAT stream I^{2}S-ADAT konvertorom a poslané po optickom rozhraní na výstup zariadenia a zároveň do ADAT-I^{2}S konvertoru. ADAT-I^{2}S konvertor je pripojený k CS4272 čipom a po I^{2}S im posiela prijaté dáta. K optickému výstupu je pripojené externé zariadenie, schopné pracovať s audiom vo formáte ADAT. Konkrétne sa jedná o zariadenie RME ADI-2. V súčasnom stave nie je možné dáta prijímať z externého zdroja (dôvod je popísaný v časti 3.2.1). Vzorkovacia frekvencia je 48 kHz.



Obr. 3.7: Blokový diagram testovania modulov na prevod medzi ADAT a I^{2}S.

3.2.3.3 Testovanie modulov na prenos audia spoločne s videom

Test pracuje s modulmi AES3 sender a receiver. Blokový diagram je na obrázku 3.8. Keďže moduly podporujú len prenos 4 audio kanálov, sú aktívne len 2 audio čipy. Dáta sú posielané po I^2S AES3 senderu, ktorý ich pošle AES3 receiveru spoločne s AES3 flagmi. AES3 receiver dáta pošle opäť po I^2S zbernici audio čipom. Vzorkovacia frekvencia je 48 kHz.



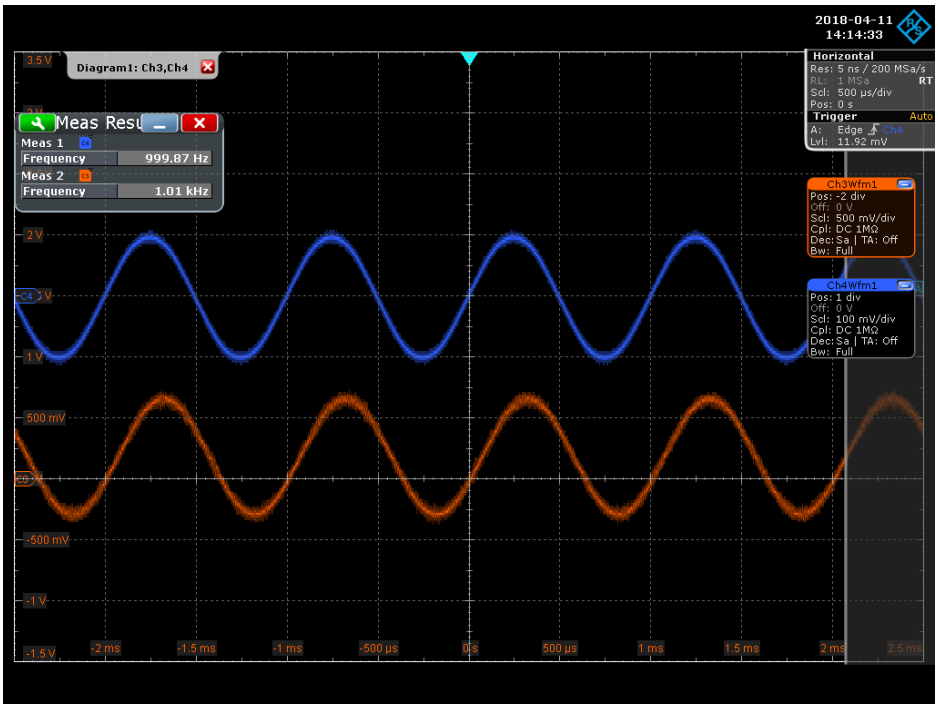
Obr. 3.8: Blokový diagram testovania modulov na prenos audia spoločne s videom.

3.3 Výsledky testovania

Všetky bloky modulov a aj samotné moduly boli najskôr simulované vo vývojovom prostredí Vivado Design Suite. Výpis priebehov testov ani manuálna kontrola signálov neukázali chybu vo VHDL návrhu modulov.

Následne boli moduly integrované a testované na prototype poskytnutom zadávateľom práce, podľa scenárov popísaných vyššie. Vďaka tomu boli odhalené a opravené menšie chyby v implementácií, ako napríklad nesprávne implementované hranové detektory. Napokon sa signálové priebehy na osciloskope, pri teste modulov na prenos zvuku počítačovou sieťou a modulov na prenos audia spoločne s videom, javili ako bezchybné. Taktiež pri prehrávaní hudby nebol zvuk nijak skresľovaný.

Ukážka vstupu a výstupu testu je na obrázku 3.9. Vstup je na kanále 4 (modrá farba) a výstup je na kanále 3 (oranžová farba). Snímok je z testu modulov na prenos zvuku počítačovou sieťou, ale výstup testu modulov na prenos audia spoločne s videom je totožný.



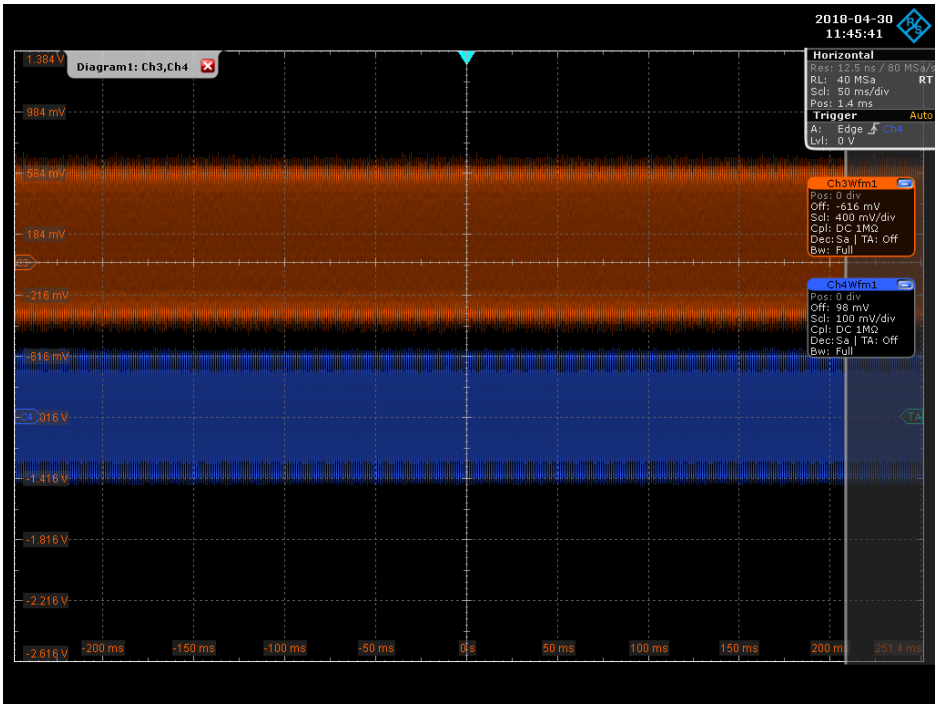
Obr. 3.9: Vstup a výstup testov modulov na prenos zvuku počítačovou sieťou a prenos audia spoločne s videom.

3.3.1 Výsledky testovania modulov pracujúcich s formátom ADAT

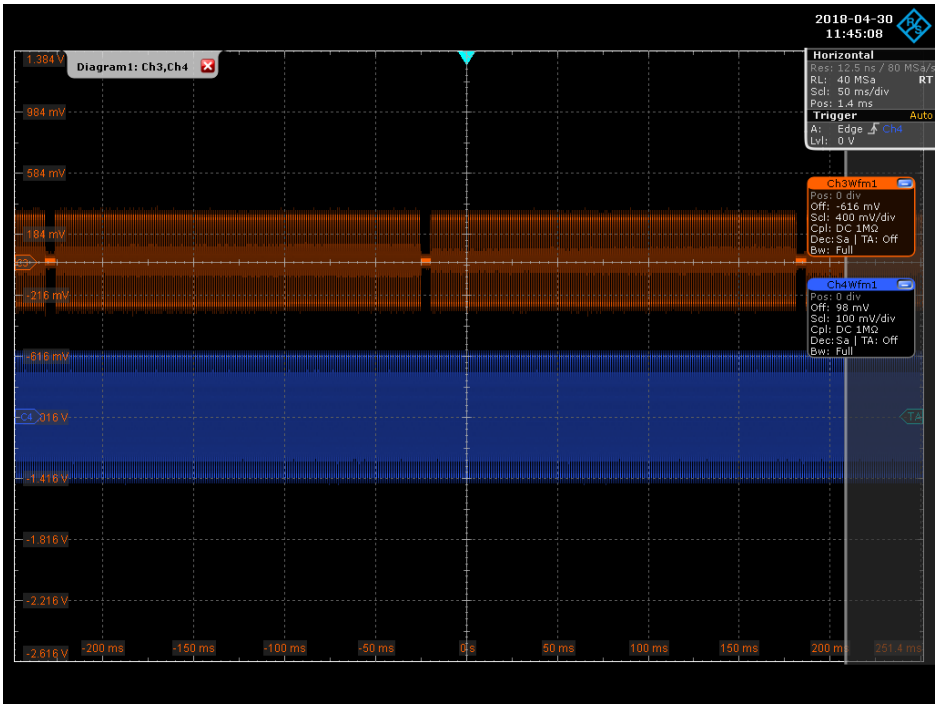
Pri testovaní ADAT modulov v loopbacku, teda pri posielaní dát z I^2S-ADAT konvertoru do ADAT-I^2S konvertoru, sa výstup javil bezchybne. Výstup vyzeral totožne s tým na obrázku 3.9. Pri zachytení dlhšieho časového úseku bol signál neprerušovaný, presne podľa očakávaní. Na obrázku 3.10 je zachytených približne 500 ms prenášaného signálu. Vstup je na kanály 4 (modrá) a výstup na kanály 3 (oranžová).

Po pripojení externého zariadenia RME ADI-2 pracujúceho s ADAT formátom na optický výstup Audio MVTP prototypu sa však na výstupnom signále RME ADI-2 začali objavovať chyby. Výstupný signál je každých 200 ms prerušený na približne 10 ms. Na obrázku 3.11 je tento výstupný signál na kanále 3 (oranžová). Výstupný signál na obrázku 3.11 má inú amplitúdu oproti tomu na obrázku 3.10, spôsobenú rozličným zosilnením výstupných zosilňovačov Audio MVTP a RME ADI-2.

Moduly boli naimplementované podľa popisu *ADAT project* [14]. Daný zdroj nie je oficiálna špecifikácia ADAT Lightpipe štandardu, ale len jeho voľne dostupný popis získaný reverznou analýzou. Týmto testom bolo dokázané, že implementácia podľa tohoto popisu je funkčná a dokáže prenášať audio dáta, ale nejedná sa o oficiálny ADAT Lightpipe štandard a nie je ním možné prenášať audio k zariadeniam podporujúcim tento štandard. Jedná sa však o jedinú voľne dostupnú špecifikáciu ADAT štandardu.



Obr. 3.10: Vstup a výstup ADAT modulov v loopbacku.



Obr. 3.11: Prerušovaný výstup z RME ADI-2 zariadenia.

Záver

V rámci tejto práce sa podarilo navrhnúť a v jazyku VHDL implementovať šesť modulov na prenos audia pre platformu Audio MVTP. Moduly umožňujú prijímať audio vzorky po I^2S zberniciach, vkladať ich ako obsah do RTP paketov a vytvárať z nich AES3 framy, ktoré môžu byť následne vkladané do pomocnej časti videa ako embedded audio. Okrem toho môžu vytvárať a vysielať digitálne audio vo formáte ADAT po optickom rozhraní. Taktiež dokážu prijímať audio vzorky z obsahu RTP paketu, AES3 framy alebo dekódovať ADAT stream a následne posielať prijaté vzorky po I^2S zberniciach.

Správanie modulov bolo simulované najskôr po častiach, potom ako celok a nakoniec v kooperácií modulov. Následne boli moduly otestované na prototype Audio MVTP. Ako vstup pri tomto testovaní slúžil generátor funkcii a výstup bol kontrolovaný použitím digitálneho osciloskopu. Na záver boli moduly otestované prehrávaním hudby.

Pri testovaní všetkých modulov, okrem tých, ktoré pracujú so štandardom ADAT, neboli objavené chyby. Moduly pracujúce so štandardom ADAT neboli implementované podľa oficiálneho štandardu, ale len podľa voľne dostupných informácii, keďže sa jedná o uzavretý štandard. Pri posielaní dát medzi modulom vytvárajúcim ADAT stream a modulom, ktorý ho prijíma, prenos funguje bez chýb. Avšak pri posielaní dát zariadeniu, ktoré podporuje príjem ADAT, jeho výstupný audio signál v pravidelných intervaloch vypadával. Prijímanie ADAT dát z externého zdroja nebolo na prototype otestované kvôli chybe v návrhu prototypu, pre ktorú nebolo možné pripojiť výstup optického rozhrania priamo k platforme Zynq.

Ďalšími rozšíreniami tejto práce a krokmi vo vývoji Audio MVTP platformy by mohla byť integrácia vytvorených modulov s modulmi pre implementáciu sieťovej časti návrhu, ktoré má zadávateľ už naimplementované, integrácia modulov vytvárajúcich AES3 s modulmi vkladajúcimi embedded audio do videa od spoločnosti Xilinx, prípadne dôkladná reverzná analýza alebo kúpa oficiálneho ADAT štandardu a následná úprava ADAT modulov.

Literatúra

[1] S.Gowdham; Batcha, S. A.; Kumar, V.: Pulse Code Modulation. [cit. 11.3.2018]. Dostupné z: https://www.kprblog.in/cse/sem3/pulse-code-modulation/

[2] European Broadcasting Union: *SPECIFICATION OF THE DIGITAL AUDIO INTERFACE (The AES/EBU interface)*. [cit. 11.3.2018]. Dostupné z: https://tech.ebu.ch/docs/tech/tech3250.pdf

[3] Ögren, J.: S/PDIF. [cit. 14.3.2018]. Dostupné z: http://www.hardwarebook.info/S/PDIF

[4] Philips Semiconductors: *I2S bus specification*. [cit. 12.3.2018]. Dostupné z: https://www.sparkfun.com/datasheets/BreakoutBoards/I2SBUS.pdf

[5] Haq, E. U.: Example NRZI encoding. [cit. 12.3.2018]. Dostupné z: https://commons.wikimedia.org/wiki/File:NRZI_example.png

[6] Schulzrinne, H.; Casner, S.; Frederick, R.; aj.: *RTP: A Transport Protocol for Real-Time Applications[online]*. Network Working Group, [cit. 13.3.2018]. Dostupné z: https://tools.ietf.org/html/rfc3550

[7] THE SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS: *ST 299-1:2009 - SMPTE Standard - 24-Bit Digital Audio Format for SMPTE 292 Bit-Serial Interface*. 2009, doi:10.5594/SMPTE.ST299-1.2009, cit. 25.3.2018. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7291761&isnumber=7291760

[8] Xilinx, Inc.: *Audio/Video Connectivity Solutions for Virtex-5 FPGAs, Reference Designs for the Broadcast Industry: Volume 2*. 2009. Dostupné z: https://www.xilinx.com/support/documentation/application_notes/xapp1014.pdf

[9] Zynq-7000 All Programmable SoC Product Advantages. [cit. 14.3.2018]. Dostupné z: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

[10] Crockett, L. H.; Elliot, R. A.; Enderwitz, M. A.; aj.: *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*. Strathclyde Academic Media, 2014, ISBN 9780992978709, [cit. 15.5.2018]. Dostupné z: http://www.zynqbook.com/

[11] SoC Micromodule with Xilinx Zynq XC7Z030-1SBG485C. [cit. 16.3.2018]. Dostupné z: https://shop.trenz-electronic.de/en/TE0715-04-30-1C-SoC-Micromodule-with-Xilinx-Zynq-XC7Z030-1SBG485C-com.-temp.-range

[12] Wolfson Microelectronics plc: *WM8804 - 1:1 Digital Interface Transceiver with PLL*. [cit. 17.3.2018]. Dostupné z: https://d3uzseaevmutz1.cloudfront.net/pubs/proDatasheet/WM8804_v4.5.pdf

[13] Cirrus Logic, Inc.: *CS4272 - 24-Bit, 192 kHz Stereo Audio CODEC*. [cit. 17.3.2018]. Dostupné z: https://www.cirrus.com/products/cs4272/

[14] Witberg, D.: ADAT project. 2016. Dostupné z: https://ackspace.nl/wiki/ADAT_project

[15] Schulzrinne, H.; Casner, S.; Frederick, R.; aj.: *RTP: A Transport Protocol for Real-Time Applications[online]*. Network Working Group, [cit. 13.3.2018]. Dostupné z: https://tools.ietf.org/html/rfc1889

[16] Schulzrinne, H.; Casner, S.: *RTP Profile for Audio and Video Conferences with Minimal Control[online]*. Network Working Group, [cit. 28.2.2018]. Dostupné z: https://tools.ietf.org/html/rfc3551

[17] Kobayashi, K.; Ogawa, A.; Casner, S.; aj.: *RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio[online]*. Network Working Group, [cit. 28.2.2018]. Dostupné z: https://tools.ietf.org/rfc/rfc3190.txt

Dodatok **A**

Zoznam použitých skratiek

ACP Accelerator Coherency Port

ADC Analog-to-Digital converter

A/D Analog to digital

APU Application processor unit

AXI Advanced eXtensible Interface

BCLK Bit clock

CLB Configurable Logic Blocks

CSRC Contributing source

DAC Digital-to-analog converter

D/A Digital to analog

EAV End of Active Video

EMIO Extended MIO

FPGA Field Programmable Gate Array

FPU Floating Point Unit

HANC Horizontal Ancillary Data Space

IC Integrated Circuit

I2C Inter-Integrated Circuit

IP Internet protocol

I^{2}S Inter-IC Sound

LPCM Linear pulse-code modulation

LRCLK Left-right clock

LSB Least significant bit

MIO Multiplexed Input/Output

MMU Memory Management Unit

MPE NEON Media Processing Engine

MPF Multiplex Position Flag

MSB Most significant bit

MVTP Modular Video Transfer Platform

OCM On chip memory

PCM Pulse-Code Modulation

PL Programmable logic

PLL Phase-locked loop

PS Processing system

RTP Real-time packet

SAV Start of Active Video

SCK Continuous serial clock

SDIN/OUT Serial data in/out

SCU Snoop Control Unit

SMPTE Society of Motion Picture and Television Engineers

SoC System on chip

SSRC Synchronization source

UDP User datagram protocol

UUT Unit under test

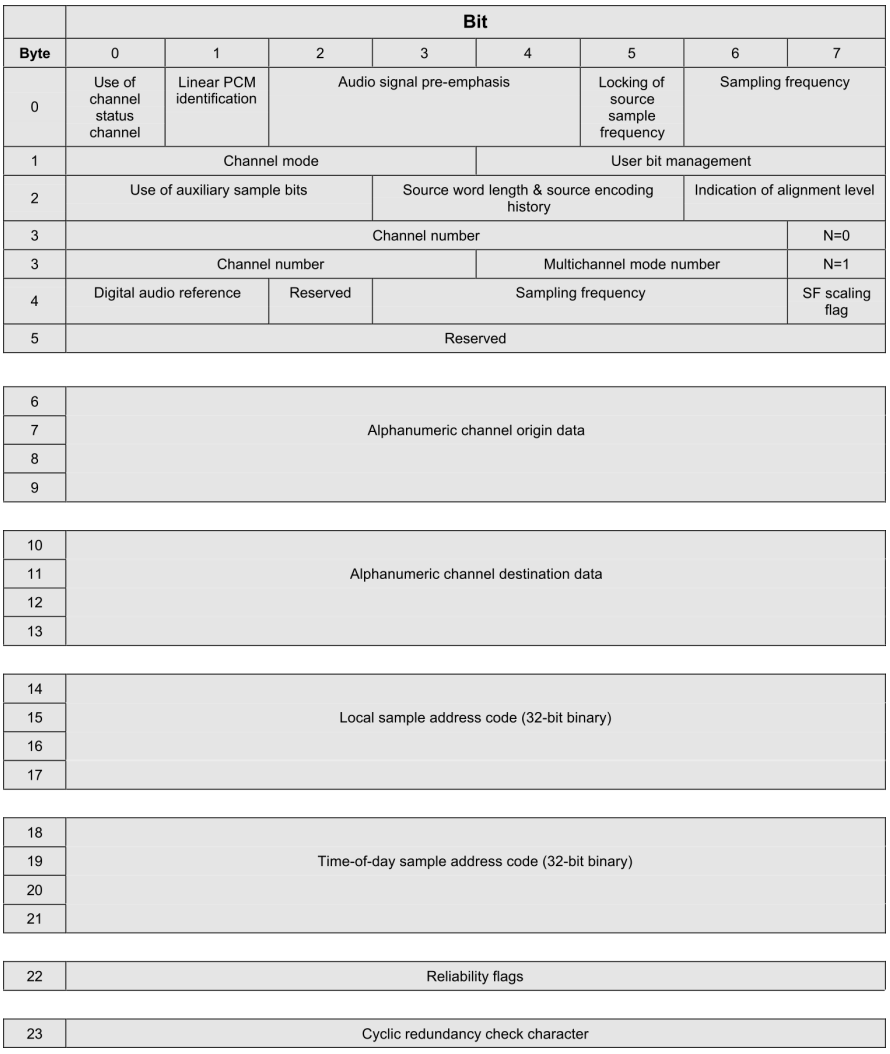
VANC Vertical Ancillary Data Space

VHDL VHSIC Hardware Description Language

VHSIC Very High Speed Integrated Circuit

WS Word select

Dodatok B



	Bit							
Byte	0	1	2	3	4	5	6	7
0	Use of channel status channel	Linear PCM identification	Audio signal pre-emphasis			Locking of source sample frequency	Sampling frequency	
1	Channel mode				User bit management			
2	Use of auxiliary sample bits			Source word length & source encoding history			Indication of alignment level	
3	Channel number							N=0
3	Channel number				Multichannel mode number			N=1
4	Digital audio reference		Reserved	Sampling frequency				SF scaling flag
5	Reserved							
6	Alphanumeric channel origin data							
7								
8								
9								
10	Alphanumeric channel destination data							
11								
12								
13								
14	Local sample address code (32-bit binary)							
15								
16								
17								
18	Time-of-day sample address code (32-bit binary)							
19								
20								
21								
22	Reliability flags							
23	Cyclic redundancy check character							

Obr. B.1: Channel status formát AES3. (Prevzaté z [2])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit 0 - 15	1	Ch0D0	Ch0D1	Ch0D2	Ch0D3	1	Ch0D4	Ch0D5	Ch0D6	Ch0D7	1	Ch0D8	Ch0D9	Ch0D10	Ch0D11	1
Bit 16 - 31	Ch0D12	Ch0D13	Ch0D14	Ch0D15	1	Ch0D16	Ch0D17	Ch0D18	Ch0D19	1	Ch0D20	Ch0D21	Ch0D22	Ch0D23	1	Ch1D0
Bit 32 - 47	Ch1D1	Ch1D2	Ch1D3	1	Ch1D4	Ch1D5	Ch1D6	Ch1D7	1	Ch1D8	Ch1D9	Ch1D10	Ch1D11	1	Ch1D12	Ch1D13
Bit 48 - 63	Ch1D14	Ch1D15	1	Ch1D16	Ch1D17	Ch1D18	Ch1D19	1	Ch1D20	Ch1D21	Ch1D22	Ch1D23	1	Ch2D0	Ch2D1	Ch2D2
Bit 64 - 79	Ch2D3	1	Ch2D4	Ch2D5	Ch2D6	Ch2D7	1	Ch2D8	Ch2D9	Ch2D10	Ch2D11	1	Ch2D12	Ch2D13	Ch2D14	Ch2D15
Bit 80 - 95	1	Ch2D16	Ch2D17	Ch2D18	Ch2D19	1	Ch2D20	Ch2D21	Ch2D22	Ch2D23	1	Ch3D0	Ch3D1	Ch3D2	Ch3D3	1
Bit 96 - 111	Ch3D4	Ch3D5	Ch3D6	Ch3D7	1	Ch3D8	Ch3D9	Ch3D10	Ch3D11	1	Ch3D12	Ch3D13	Ch3D14	Ch3D15	1	Ch3D16
Bit 112 - 127	Ch3D17	Ch3D18	Ch3D19	1	Ch3D20	Ch3D21	Ch3D22	Ch3D23	1	Ch4D0	Ch4D1	Ch4D2	Ch4D3	1	Ch4D4	Ch4D5
Bit 128 - 143	Ch4D6	Ch4D7	1	Ch4D8	Ch4D9	Ch4D10	Ch4D11	1	Ch4D12	Ch4D13	Ch4D14	Ch4D15	1	Ch4D16	Ch4D17	Ch4D18
Bit 144 - 159	Ch4D19	1	Ch4D20	Ch4D21	Ch4D22	Ch4D23	1	Ch5D0	Ch5D1	Ch5D2	Ch5D3	1	Ch5D4	Ch5D5	Ch5D6	Ch5D7
Bit 160 - 175	1	Ch5D8	Ch5D9	Ch5D10	Ch5D11	1	Ch5D12	Ch5D13	Ch5D14	Ch5D15	1	Ch5D16	Ch5D17	Ch5D18	Ch5D19	1
Bit 176 - 191	Ch5D20	Ch5D21	Ch5D22	Ch5D23	1	Ch6D0	Ch6D1	Ch6D2	Ch6D3	1	Ch6D4	Ch6D5	Ch6D6	Ch6D7	1	Ch6D8
Bit 192 - 207	Ch6D9	Ch6D10	Ch6D11	1	Ch6D12	Ch6D13	Ch6D14	Ch6D15	1	Ch6D16	Ch6D17	Ch6D18	Ch6D19	1	Ch6D20	Ch6D21
Bit 208 - 223	Ch6D22	Ch6D23	1	Ch7D0	Ch7D1	Ch7D2	Ch7D3	1	Ch7D4	Ch7D5	Ch7D6	Ch7D7	1	Ch7D8	Ch7D9	Ch7D10
Bit 224 - 239	CH7D11	1	Ch7D12	Ch7D13	Ch7D14	Ch7D15	1	Ch7D16	Ch7D17	Ch7D18	Ch7D19	1	CH7D20	CH7D21	CH7D22	CH7D23
Bit 240 - 255	1	0	0	0	0	0	0	0	0	0	0	1	User0	User1	User2	User3

Data Integrity (NRZI)
Synchronisation
User bits

Ch4D13 = Channel 4 databit 13

Obr. B.2: Štruktúra ADAT paketu. (Prevzaté z [14])

Dodatok C

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy implementácie
    - `VHDL`
      - `RTP` ..... zdrojové a simulačné súbory pre RTP moduly
      - `I2S` ..... zdrojové a simulačné súbory pre I2S moduly
      - `ADAT` ..... zdrojové a simulačné súbory pre ADAT moduly
      - `AES3` ..... zdrojové a simulačné súbory pre AES3 moduly
      - `TEST` ..... top moduly pre testy na Audio MVTP platforme
      - `CONST` ..... použité konštanty
    - `C` ..... konfigurácia platformi Audio MVTP
  - `thesis` ..... zdrojová forma práce vo formáte LaTeX
- `text` ..... text práce
  - `DP_Majercik_Michal_2018.pdf` ..... text práce vo formáte PDF