

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘEVODNÍKOVÝ MODUL S RYCHLOU SÉRIOVOU KOMUNIKAČNÍ LINKOU

THE AD CONVERTER MODULE WITH HIGH SPEED COMMUNICATION LINE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Václav Kabátník

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Soběslav Valach

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Václav Kabátník

ID: 154754

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Převodníkový modul s rychlou sériovou komunikační linkou

POKYNY PRO VYPRACOVÁNÍ:

Cílem projektu je realizovat obvodové a firmwarové řešení převodníku (AD nebo DA) pro rychle sériové rozhraní – uvažujeme komunikační rychlosti vyšší než 1 Gbit/s. Systém by měl disponovat definovanou latencí (obecně minimální) a definovaným konstantním fázovým posunem. Cílová platforma bude FPGA (Xilinx nebo Altera).

1. Prostudujte techniky sériového přenosu dat.
2. Prostudujte vhodné standardy a jejich varianty.
3. Navrhněte obvodové řešení.
4. Vytvořte vhodný firmware v jazyku VHDL nebo Verilog.
5. Ověřte funkci na reálném systému (analýza jitteru, latence a dalších časových charakteristik).

DOPORUČENÁ LITERATURA:

www.xilinx.com

www.altera.com

Standard JESD204

Termín zadání: 8.2.2016

Termín odevzdání: 1.8.2016

Vedoucí práce: Ing. Soběslav Valach

Konzultant bakalářské práce:

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

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá návrhem a implementací standardu JESD204B. V první řadě práce seznamuje se samotným standardem, a poté pojednává o jeho revizích. Standard JESD204B funguje na bázi vrstevné specifikace, podobně jako ostatní protokoly. Práce dále seznamuje s vlastní implementací samotného firmwaru, podle kterého probíhá přenos dat na bázi standardu JESD204B. Firmware je psán pomocí jazyku VHDL. Součástí práce je obvodové řešení, kde je vytvořeno schéma našeho obvodu. Výsledkem práce je realizované obvodové a firmwarové řešení převodníku pro rychlé sériové rozhraní.

KLÍČOVÁ SLOVA

FPGA, JESD204, AD/DA převod, VHDL, Xilinx

ABSTRACT

This thesis deals with the standard JESD204B. First of all, it introduces the work standard itself and then describe its revision. Standard itself works on the basis of layer specification, like other protocols. Work also introduces firmware custom implementation by which data transfer is based on standard JESD204B. Firmware is written using language VHDL. Part of work is circuit solution where is created scheme of our cuircit. The result of the work is implemented circuit and firmware solutions transmitter for fast serial interface.

KEYWORDS

FPGA, JESD204, AD/DA conversion, VHDL, Xilinx

BIBLIOGRAFICKÁ CITACE

KABÁTNÍK, V. Převodníkový modul s rychlou sériovou komunikační linkou. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 63 s. Vedoucí bakalářské práce Ing. Soběslav Valach.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Převodníkový modul s rychlou sériovou komunikační linkou“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s využitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků za porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplívajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 1.8.2016

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Soběslavu Valachovi za odbornou pomoc, kritické připomínky a cenné rady při zpracování mé bakalářské práce. Děkuji také mé rodině a blízkým přátelům za podporu během celého studia i psaní této bakalářské práce.

V Brně dne 1.8.2016

.....

(podpis autora)

OBSAH

1	Úvod	9
2	Porovnání standardu JESD204B s LVDS.....	10
3	Standard JESD204.....	12
3.1	Co je to standard JESD204?.....	12
3.2	JESD204A.....	14
3.3	JESD204B	15
3.3.1	Deterministická latence.....	17
3.4	Klíčové aspekty standardů JESD204	19
3.4.1	8b/10b Interní hodiny.....	19
3.4.2	Zarovnání sériových diferenciálních párů	19
3.4.3	Údržba/Monitorování.....	19
3.5	Klíčové vrstvy v JESD204 standardech	20
3.5.1	Transportní vrstva	21
3.5.2	Aplikační vrstva.....	23
3.5.3	Datová vrstva	24
3.5.4	Fyzická vrstva	27
3.5.5	Kódující (Scrambling) vrstva – volitelná.....	28
3.6	JESD linkové parametry/znaky	29
3.6.1	Parametry linky.....	29
3.7	JESD204 Terminologie	30
4	Měření jitteru	32
4.1	Jitter.....	33
5	FPGA.....	35
5.1	Historie	35
5.2	Struktura	36
5.3	Xilinx.....	38
5.3.1	Spartan 6	39
5.3.2	Virtex 7	39
5.3.3	Kintex 7.....	40
5.3.4	Artix 7	41
5.4	Vývojové prostředky	42
5.4.1	Software	42
5.4.2	Hardware.....	42
5.5	Hodiny	43
6	Nástroj pro popis hardware.....	43
7	Návrh obvodového řešení.....	44
4.1	Převodník ADC34J25	44

4.2	Schéma	44
4.3	Moje práce.....	44
8	Vlastní Implementace firmwaru.....	48
8.1	Nastavení parametrů linky	49
8.2	Vztahy hodinových frekvencí	52
8.3	Implementované moduly a ověření funkčnosti	53
8.3.1	CGS.....	53
8.3.2	Synchronizace počátečního rámce.....	55
8.3.3	Znakové nahrazení.....	57
8.4	Transportní vrstva	58
9	Ověření na reálném systému.....	59
9.1	Zjištění latence IP coru.....	59
10	Závěr	60
	Literatura	61
	Seznam obrázků.....	62
	Seznam tabulek.....	63
	Seznam příloh	63

1 ÚVOD

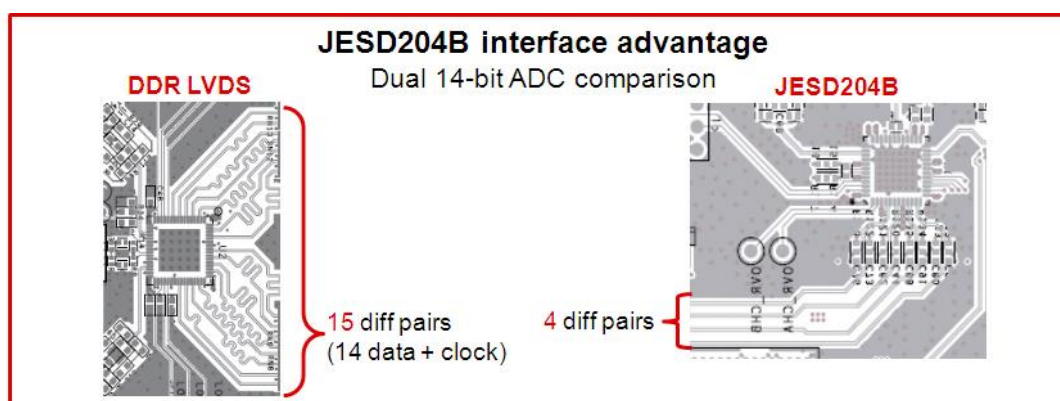
Moderní AD převodníky se dnes vyrábějí s rychlostí gigasample-za-vteřinu (GSPS). K těmto převodníkům je potřeba rozhraní, které tyto rychlosti podporují. Aby bylo možné zachytit RF (rádio frekvenční) spektra ve vyšším kmitočtovém rozsahu, je potřeba širokopásmových RF AD převodníků. Moderní převodníky zachycují širší šířku pásma a umožňují více konfigurovatelné SDR (software defined radio) platformy. Pro tyto převodníky je nutné vysoce rychlostní sériové rozhraní - JESD204B. JESD204B standard popisuje přenos dat, které jsou vysílány z vysílače (AD převodník) do přijímače (FPGA nebo vlastních ASIC).

JESD204B je využíváno v mnoha různých aplikacích a oborech, ke kterým patří například bezdrátové a drátové komunikace, kde designéři mohou navrhnout softwarově definované rádio. Mezi další obory, kde se JESD204B standard uplatní, se řadí armáda, která standard může použít pro naváděcí střely. V neposlední řadě se standard využívá v měřicím a testovacím průmyslu.

2 POROVNÁNÍ STANDARDU JESD204B S LVDS

Na jednoduchém příkladu si ukážeme hlavní nevýhodu LVDS standardu. ADC12D1600 je 12-bitový dual 1600 MSPS AD převodník. Ten má 48 LVDS spárovaných dvojic. Směrování 96 přenosových cest (vyjma hodin) může být náročné, protože si všechny musí elektricky na délku odpovídat (zpoždění).

Cílem výrobců bylo zabudovat datové hodiny do obvodu, což pomohlo ke snadnějšímu návrhu desky.



Obrázek 1. Porovnání obvodového řešení standardu LVDS se standardem JESD204B [1]

Sériové rozhraní JESD204B významně snižuje počet diferenciálních párů a interní výstupní datové hodiny eliminují potřebu sběrnice, což vede k zjednodušení digitálního datového rozhraní a k redukci místa na desce.

spojení	LVDS (DDR)	JESD204B(ADS42JB69)
Cesty Datových hodin	2 (1 pár)	interní
Datové cesty	32 (16 párů)	2 až 4 (1 až 2 páry) závisí na módu
Potřeba shody	ano	ne
Fázově zarovnané vzorkování pro několik AD převodníků	Možné, ale složité	ano
kódování	100% (bez kódování)	84% (8b/10b kódování)

Tabulka 1. Srovnání LVDS s JESD204B [1]

V **Tabulka 1.** Srovnání LVDS s JESD204B [1] 1 byly použity parametry převodníku ADS42JB69 dual 16-bit, 250 MSPS.

Návrh obvodového řešení pro rozhraní JESD204B je o trochu jednodušší, než tomu bylo u LVDS a poskytuje mnohem více funkcí, např. přechodem z 14-bitového AD převodníku na 16-bitový by se použilo stejného počtu diferenciálních párů – komponenty ve

stejných rodinách mohou mít i stejně definované piny. Fázové zarovnání několika AD převodníků je také mnohem jednodušší. Navíc celková velikost pouzdra (package) bývá zpravidla o cca 25 % menší než u LVDS. Oproti LVDS standardu je u JESD204B větší linková latence. JESD204B je škálovatelné na větší frekvence než LVDS, za pomoci použití SERDES a CDR technik. [1]

3 STANDARD JESD204

Standard JESD204 je rozhraní, které se používá pro moderní převodníky. Toto rozhraní bylo původně navrženo před několika lety, avšak po čase byly nároky na kvalitu rozhraní větší, proto JESD204 prošlo dvěma revizemi, které jej dělají mnohem atraktivnější a efektivnější.

JESD204 rozhraní přináší efektivitu a nabízí několik výhod oproti svým CMOS a LVDS předchůdcům, pokud jde o rychlost, velikost a cenu. Návrháři JESD204 využívají výhod rychlejšího rozhraní, aby udrželi krok s rychlejším vzorkováním převodníků. Nejen že došlo k redukci počtu pinů, což vede k menší velikosti pouzdra, ale toto obvodové řešení též obsahuje nižší počet cest, které tvoří návrh desky mnohem jednodušší a nabízí nižší celkové náklady na systém. Standard je také snadno škálovatelný tzn., že může být přizpůsoben budoucím potřebám.

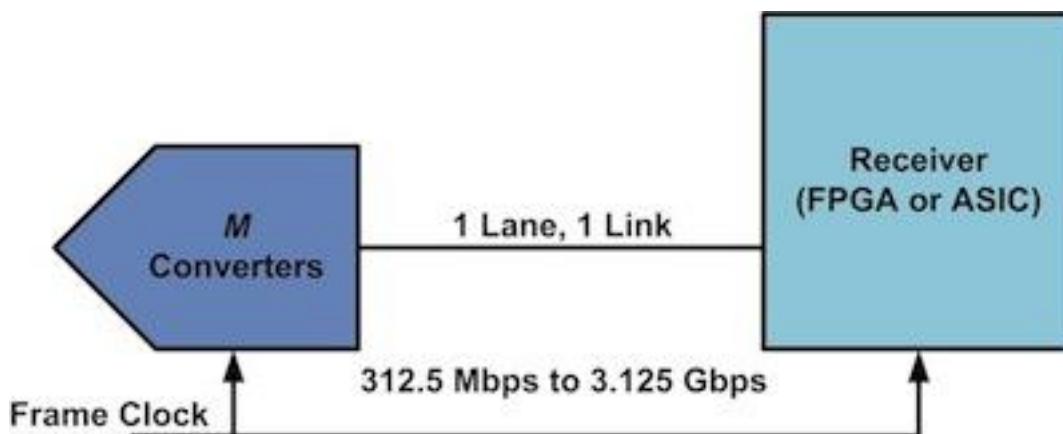
Standard JESD204 prošel dvěma revizemi od svého zavedení v roce 2006. V současné době je aktuální revize B, ve které byly přidány nové funkce.

Tento standard se vztahuje jak na analogově-číslicové převodníky (ADC), tak i na digitálně analogové převodníky (DAC) a je primárně určen jako rozhraní pro FPGA (mohou být také použity s ASIC). [2]

3.1 Co je to standard JESD204?

V dubnu roku 2006 byla vydána první verze standardu JESD204. Tento standard popisuje multi-gigabitové sériové datové spojení mezi převodníky a přijímačem, což jsou obvykle zařízení, jako například FPGA nebo ASIC. V této původní verzi JESD204 byla sériová datová linka definována pro jeden sériový diferenciální pár mezi převodníkem nebo více převodníky a přijímačem. Grafické znázornění je uvedeno na **Obrázek 2**. Originální standard JESD204

Znázorněný diferenciální pár (Lane) představuje fyzické rozhraní mezi M počtem převodníků (Converters) a přijímačem (Receiver). Skládá se z diferenciální dvojice vodičů využívající CML (Current Mode Logic). Zobrazená linka (Link) je sériová datová linka, která je zavedena mezi převodníky a přijímačem. Hodinový rámec, neboli frame clock, je směřován, jak do převodníků, tak do přijímače a poskytuje hodinový signál pro link JESD204 mezi zařízeními. [2]



Obrázek 2. Originální standard JESD204 [2]

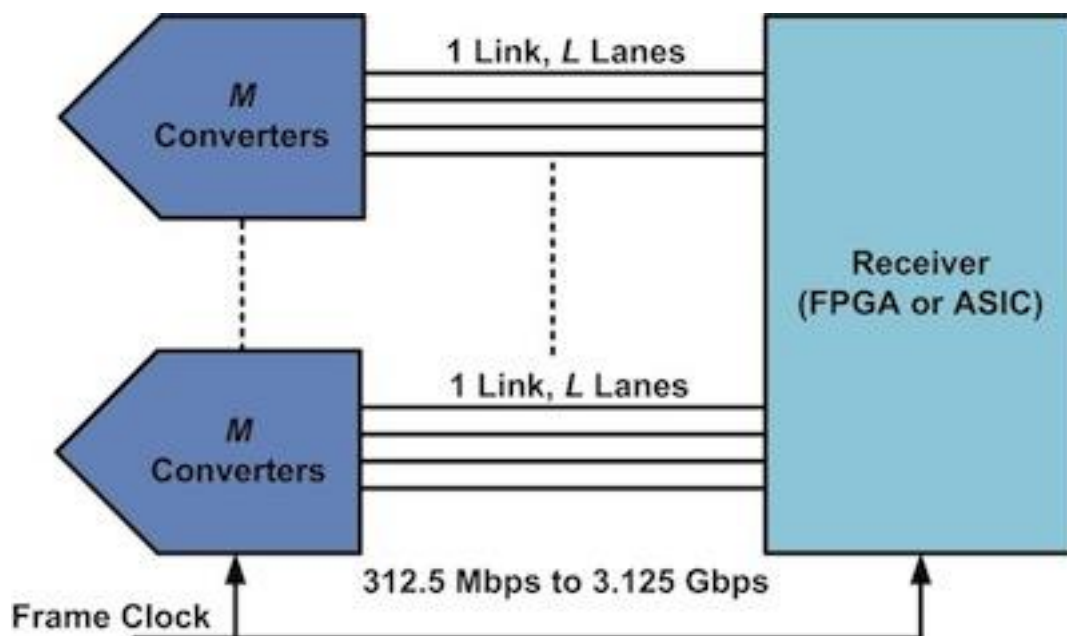
Rychlost dat je definována od 312,5 megabitů za sekundu (Mbps) do 3,125 gigabitů za sekundu (Gbps) s oběma zdroji a impedance zátěže definované jako $100 \Omega \pm 20\%$. Úroveň diferenciálního napětí je definováno formálně jako 800 mV špička-špička v běžném režimu rozsahu úrovně napětí z 0,72 V až 1,23 V. Linka využívá kódování 8b / 10b, které zahrnuje interní hodiny a odstraní nutnost pro spojení dodatečné hodinové linie a související složitost při zarovnání dodatečného hodinového signálu s přenesenými daty. S růstem rychlosti a rozlišení převodníků bylo potřeba standard podrobit revizi, která začlení podporu pro více sériových diferenciálních párů s více převodníky.

Toto poznání vedlo k první revizi standardu JESD204 v dubnu 2008 s názvem JESD204A. Tato revize standardu přidává více sériových diferenciálních párů s více převodníky. Datové rychlosti linky podporující od 312,5 Mbps až do 3,125 GB zůstaly nezměněny stejně jako hodinový rámec a specifikace elektrického rozhraní. Zvýšení schopnosti standardu pro podporu více sériových diferenciálních párů, bylo možné pro převodníky s vysokou vzorkovací frekvencí a s vysokým rozlišením ke splnění maximální podporované přenosové rychlosti 3,125 Gbps.

Obrázek 3 ukazuje grafické znázornění dalších možností přidané v revizi JESD204A pro podporu více diferenciálních párů. [2]

3.2

JESD204A



Obrázek 3. První revize - JESD204A [2]

Klíčové signály

- **Rámcové hodiny (Frame clock)** – dodává hodinový signál do systému.
- **SYNC~** - synchronní signál s hodinovým rámcem. Označuje stav synchronizace. Jestliže je signál na hodnotě LOW, tak poukazuje na to, že přijímač s vysílačem komunikují.
- **Lane 0, ...,L-1** – tvořen diferenciálním párem vodičů (Typicky vysoce rychlostní CML - current mode logic)

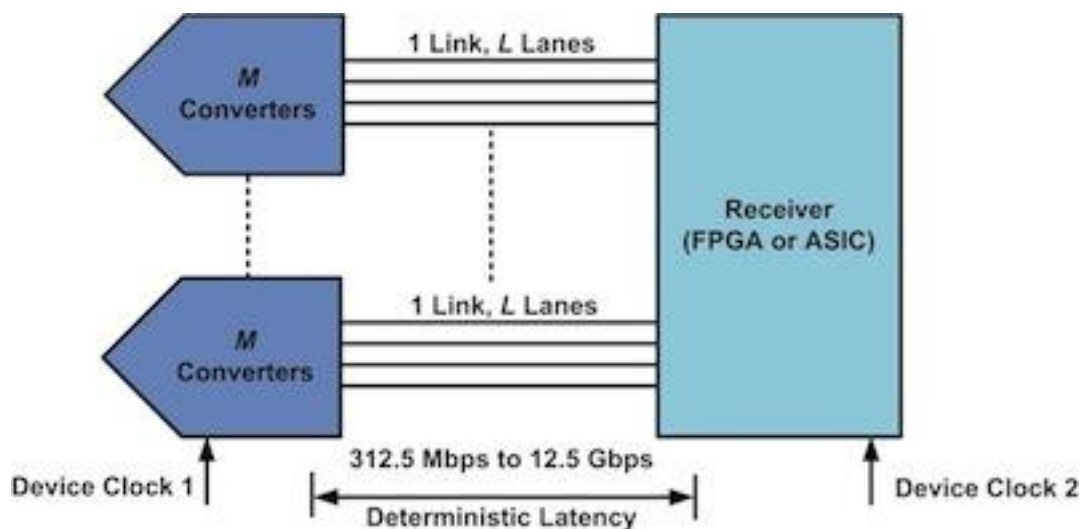
Kódované 8b/10b skupiny jsou přenášeny následujícím způsobem:

MSB (Most significant bit – Nejvíce významný bit) první / **LSB** (Least significant bit – Nejméně významný bit) poslední.

Ačkoliv JESD204 standard a revidovaný standard JESD204A měly větší výkon než předchozí rozhraní, stále postrádali klíčový element. Tento chybějící element byla deterministická latence.

Kromě toho převodníky pokračovaly ve zvýšení rychlosti i rozlišení. Tyto faktory vedly k druhé revizi standardu. V červenci 2011 byla vydána druhá a zároveň aktuální revize standardu pod názvem JESD204B. Jedním z klíčových komponentů revidovaného standardu bylo přidání opatření pro dosažení deterministické latence. Kromě toho, že podporované datové rychlosti byly posunuty na 12.5 Gbps, bylo potřeba je rozdělit na různé rychlostní stupně zařízení. Tato revize standardu vyzývá k přechodu od používání hodinového rámce, jako hlavního zdroje hodin, k systémovým hodin jako hlavního zdroje hodin. **Obrázek 4.** Druhá (Aktuální) revize JESD204B [2] - poskytuje reprezentaci dalších možností přidané revizí JESD204B. [2]

3.3 JESD204B



Obrázek 4. Druhá (Aktuální) revize JESD204B [2]

Klíčové signály

- **Systémové hodiny (Device clock)** – dodává hodinový signál do systému. Může mít odlišnou periodu než je rámcová, či multirámcová perioda. Jestliže ano, tak je zařízení odpovědné za generování rámcových, či multirámcových hodin do systémových hodin.
- **SYNC~** - stejné jako u JESD204A, jen je synchronní s lokálními multirámcovými hodinami, místo hodinového rámce
- **Lane 0, ..., L-1** – stejné jako u JESD204A
- **SYSREF (volitelný)** – odpovědný za reset rozdělovače systémových hodin (včetně lokálních multirámcových hodin) k zajištění deterministické latence.

Kromě deterministické latence, verze JESD204B zvyšuje podporované přenosové rychlosti linky na 12.5 Gbps a rozděluje zařízení do tří různých rychlostních stupňů. Zdrojová a zátěžová impedance je stejná pro všechny tři rychlostní stupně a je definována jako $100 \Omega \pm 20\%$.

První rychlostní stupeň se vyrovnává s přenosovou rychlostí linky z JESD204 a JESD204A verzí standardu a definuje elektrické rozhraní pro přenosové rychlosti linky až na 3.125 Gbps.

Druhý rychlostní stupeň u JESD204B definuje elektrické rozhraní pro přenosové rychlosti linky až na 6.375 Gbps. Tento rychlostní stupeň snižuje minimální úroveň diferenčního napětí na 400 mV špička-špička, z 500 mV špička-špička pro první rychlostní stupeň.

Třetí rychlostní stupeň u JESD204B definuje elektrické rozhraní pro přenosové rychlosti linky až na 12.5 Gbps. Tento rychlostní stupeň snižuje minimální úroveň diferenčního napětí potřebný pro elektrické rozhraní na 360 mV špička-špička. Jako diferenciální pár rychlostního přenosu dat se zvyšuje s rychlostním stupněm, tak se minimální požadovaná hodnota diferenčního napětí snižuje, aby se fyzická realizace zjednodušila za pomoci snížení potřebných rychlostí přeběhu (změna napětí za jednotku času) v ovladačích. [2]

Revize JESD204B přechází z hodinového rámce na systémové hodiny, neboli device clock, což vede k větší flexibilitě systému. Hodinový rámec a vzorkovací hodiny převodníků jsou obvykle stejné. Tento způsob však nenabízí příliš flexibility, naopak může způsobovat nežádoucí složitost u návrhu systému při pokusu o vedení tohoto stejného signálu do více zařízení. V JESD204B jsou systémové hodiny časovým odkazem pro každý element v systému. Každý převodník a přijímač přijímá jejich příslušné hodiny z generátoru hodinového

obvodu, který je zodpovědný za generování všech hodin ze společného zdroje. To umožňuje větší flexibilitu při návrhu systému, ale vyžaduje to, aby vztah mezi hodinovým rámcem a hodinovým zařízením byl specifikován pro dané zařízení. [2]

3.3.1 Deterministická latence

Latence je označována jako deterministická, když čas ze vstupu JESD204 vysílače do výstupu JESD204 přijímače má trvale stejný počet hodinových cyklů.

V paralelních implementacích je tato latence jednodušší - hodiny jsou prováděny s daty. V sériových implementacích existují vícehodinové domény a mohou způsobit nedeterminismus. JESD204 a JESD204A neobsahují podporu pro zajištění deterministické latence. JESD204B specifikuje 3 podtřídy zařízení:

- **podtřída 0** - žádná podpora pro deterministickou latenci,
- **podtřída 1** – deterministická latence při použití SYSREF (nad 500 MSPS),
- **podtřída 2** – deterministická latence při použití SYNC~ (až do 500 MSPS).

[3]

3.3.1.1 Deterministická latence při použití SYNC~

Pro převodníky pracující s rychlostí menší než 500 MSPS, může být signál SYNC~ použit k zaručení deterministické latence. Jakmile se dosáhne synchronizace linky, přijímač deaktivuje SYNC~ do vysílače. Vysílač tuto událost zachytí pomocí rámcových hodin. Latence ve vysílači od doby, kdy zachycuje signál SYNC~, až do začátku ILAS, musí být konstantní. Se známou latencí přijímač nyní ví, kdy se deaktivuje SYNC~, poté je známa latence přední cesty a může být opravena.

Důležité je podotknout, že SYNC~ signalizace je systémově synchronní (časování systému může záležet na interakci více zařízení) a musí splňovat nastavení a držet čas vzhledem k rámcovému vzorku hrany ve vysílači nebo se vyskytnou problémy. [3]

3.3.1.2 Deterministická latence při použití SYSREF

Signál SYSREF je použit pro převodníky pracující s rychlostí vyšší než je 500 MSPS. SYSREF je volitelný a zdrojově synchronní (časování systému záleží na interakci dvou zařízení) signál, který je používán v JESD204B specifikaci k:

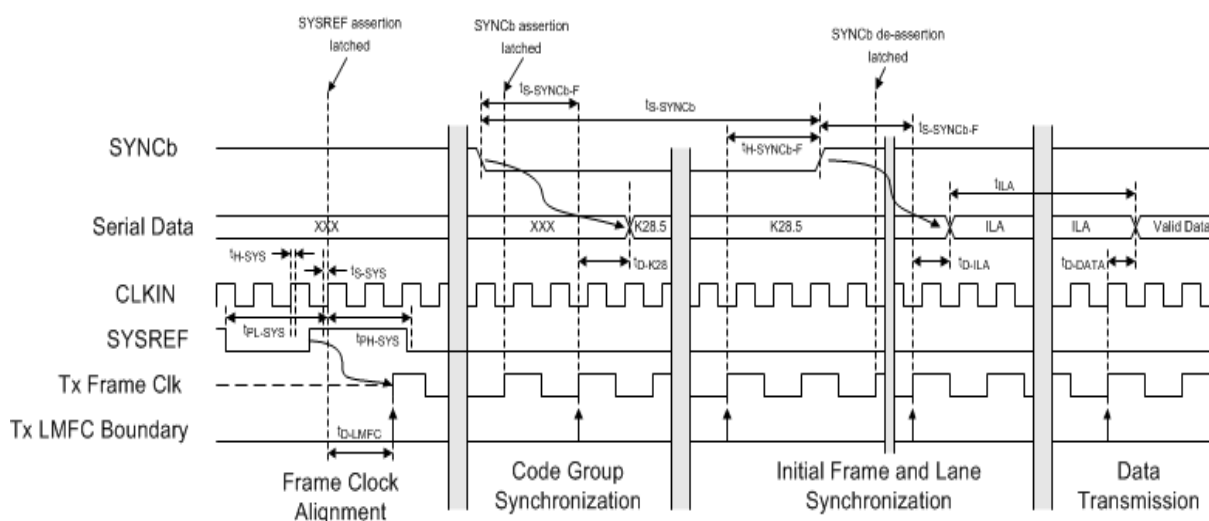
- budoucí důkaz specifikace pro převodníky s vysokým vzorkovacím kmitočtem,
- poskytuje nejvyšší úroveň přesnosti s nejnižší složitostí.

V JESD204B bude SYSREF:

- zarovnávat vnitřní děliče od systémových hodin (device clock) u převodníků,
- zarovnávat všechny lokální multirámcové hodin (LMFC). Tyto LMFC zdroje jsou použity v systému k zajištění deterministické latence,
- signál může být periodický při nízké rychlosti, mezerový či one-shot.

V přijímači je SYNC~ vysílán LMFC hranou a ne hranou rámcových hodin.

Ve vysílači je ILAS (a tím i /A/ znak) také vysílán LMFC hranou, oproti hraně rámcových hodin. SYNC~ je nyní synchronní k rámcovým hodinám a systémově synchronní k LMFC. Při použití SYSREF je systém 100% zdrojově synchronní. [3]



Obrázek 5. Ukázka JESD204B rozhraní linky - inicializovaný časový diagram [4]

3.4 Klíčové aspekty standardů JESD204

3.4.1 8b/10b Interní hodiny

Interní hodiny využívají stejnosměrně vyvážené (střední hodnota signálu rovna nule) kódování, což zaručuje značný přechod frekvence pro použití Clock and Data Recovery (CDR – Hodiny a datové obnovení) modelů. Přenos dat je kódován 8 na 10. Hodiny se následně odvozují.

Kódování povoluje využití obou datových i řídicích znaků (řídicí znaky mohou být použity k uspořádání linky, údržby, dohlížení, atp.). Toto kódování dále detekuje jednobitové chybové události, které se na lince objevují.

3.4.2 Zarovnání sériových diferenciálních párů

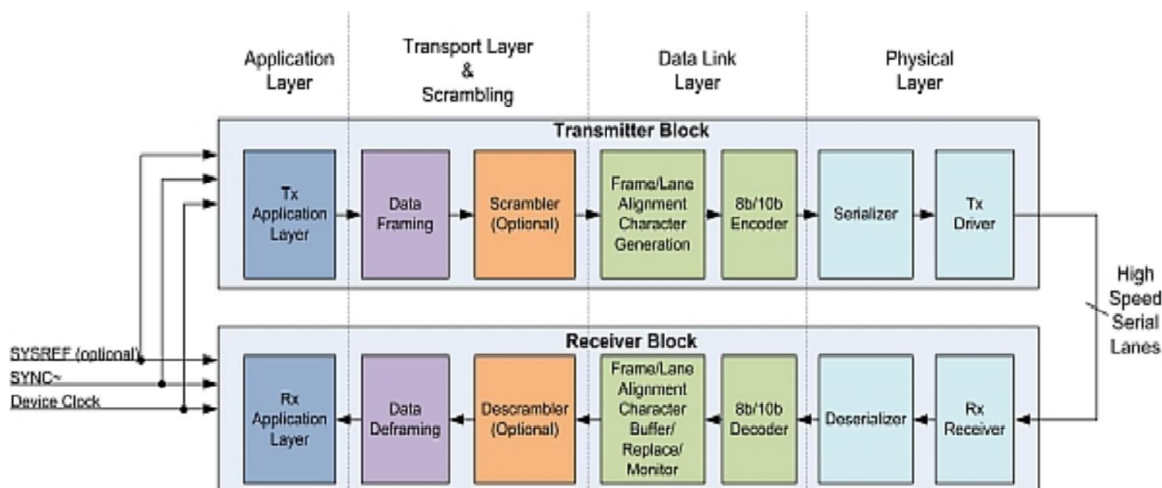
Zarovnání sériových diferenciálních párů je dosaženo pomocí speciálních vzorů (patterns) s řídicími znaky. V definovaném okamžiku všechny vysílače vyšlou zarovnávací znak diferenciálního páru /A/. Vzhledem k různým časovým zpožděním diferenciálních párů mohou tyto zarovnávací znaky být přijímány přijímačem v různých časech. Když všechny přijímače zachytí /A/ znaky, začnou přenášet přijaté data do následného zpracování dat ve stejném okamžiku. Hlavní prací zarovnání je tedy korekce dat v protokolu.

3.4.3 Údržba/Monitorování

K zajištění zarovnání je využito specifických zarovnávacích znaků. Kvalita linky je sledována přijímačem na diferenciálním páru. Linka je korigována přijímačem na základě „error tresholds“.

3.5 Klíčové vrstvy v JESD204 standardech

Je důležité vědět, že JESD204B standard je tzv. **vrstvová specifikace**. Každá vrstva ve specifikaci vykonává určitou činnost. Aplikační vrstva umožňuje konfiguraci a mapování dat na JESD204B lince. Transportní vrstva mapuje konverzní vzorky do a z rámcových nezakódovaných oktet. Kódovací vrstva může volitelně vzít tyto oktety a zakóduje je, aby se snížily účinky EMI (nežádoucí rušení ovlivňující funkce elektrického zařízení vyzařovanou elektromagnetickou energií). Kódování se provádí ve vysílači a dekódování se provádí v přijímači. Datová vrstva je místo, kde jsou případně zakódované oktety zakódovány do 10-bitových znaků. V této vrstvě se také děje generace či detekce kontrolních znaků, pro monitorování, zarovnání a údržbu diferenciálního páru. Fyzická vrstva obsahuje serializér/deserializér, neboli (SERDES). Vrstva odpovídá za vysílání či přijímání znaků při zvolené rychlosti linky. Vrstva dále obsahuje serializér, přijímače, CDR, atd. **Obrázek 6** ukazuje na zjednodušené schéma toku dat v JESD204 systémech.



Obrázek 6. Zjednodušené schéma toku dat v JESD204 systémech [5]

Na **Obrázek 6** si můžeme povšimnout, že tok dat jde z vysílače (Transmitter Block) aplikační vrstvy do transportní vrstvy, kde můžeme volitelně využít kódovacího bloku (Scrambler). Dále tok pokračuje přes datovou vrstvu, kde kóduje případně zakódované oktety do 10 bitových znaků. Poslední vrstvou, kterou tok musí projít, aby se dostal do přijímače, je vrstva fyzická, po které se přes vysokorychlostní sériové diferenciální páry dostáváme do přijímače, kde probíhá opačný proces (volitelně je možné využít Dekódovacího bloku). [5]

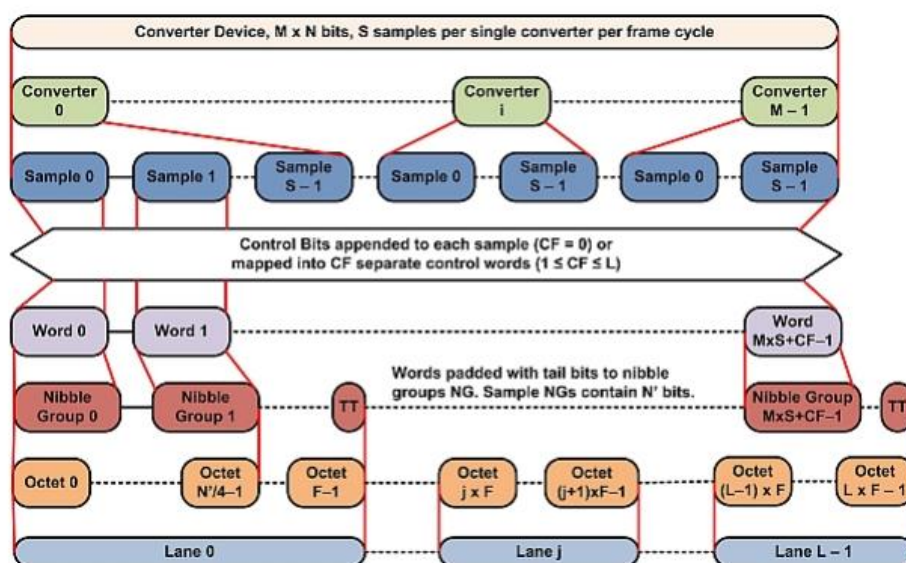
3.5.1 Transportní vrstva

Transportní vrstva bere vzorky z AD převodníku a přidává informaci k vytvoření nibble skupin (obvykle 4 - bitová hranice). Tato informace je ve formě koncových bitů nebo řídicích bitů, které mohou poskytnout dodatečné informace o přenášených datech. Transportní vrstva uspořádá tyto nibble skupiny do rámců. Je důležité poznamenat, že transportní vrstva dodává vzorky do datové vrstvy, jako paralelní data. Šířka paralelní datové sběrnice je určena pomocí rámcových architektur (jeden byte = 8 bitu, dva byty = 16 bitů, atp.). V tento moment se data ještě nedostaly do serializéru.

4 způsoby mapování:

- jeden převodník může být mapován na linku s jedním diferenciálním párem,
- jeden převodník může být mapován na více linek s více diferenciálními páry,
- více převodníků ve stejném zařízení může být mapováno na jednu linku s jedním diferenciálním párem,
- více převodníků ve stejném zařízení může být mapováno na více linek s více diferenciálními páry.

[5]



Obrázek 7. Mapování konverzních vzorků do sériových diferenciálních párů [5]

Obrázek 7 poukazuje na bližší seznámení s mapováním vzorků AD převodníku do sériových diferenciálních párů.

Parametr N' udává velikost JESD204B slova. Rozlišení převodníku se člení na 4 - bitové nibbly. 14-bitový převodník, stejně jako 16-bitový převodník, má 4 nibbly, zatímco 12-bitový převodník má nibbly 3. Jestliže je parametr N' nastaven na hodnotu 12 pro AD9625, počet požadovaných diferenciálních párů může být snížen na 2 tím, že 6 diferenciálních párů je nutných k údržbě rychlosti linky méně než 6.5 Gbps. Doporučuje se, aby konverzní vzorky (S) byly mapovány do JESD204B slov na 4 - bitové nibble hranice.

Transportní vrstva určuje, jak přijmout data z AD převodníku, na základě konfigurace parametrů linky, které jsou definovány pro dané zařízení. Tyto parametry jsou přenášeny z AD převodníku do FPGA během Initial Lane Alignment Sequence (ILAS). Tato nastavení jsou konfigurována přes Serial Port Interface (SPI), který by nastavil hodnoty registrů AD převodníku a FPGA k definování konfigurace parametrů linky. Kontrolní součet je generován z parametrů a přenášen tak, aby přijímač (FPGA) mohl ověřit, zda byla konfigurace parametrů linky korektně přijata. Parametry odeslané přes linku nejsou použity ke konfiguraci přijímače, nýbrž jsou použity pouze k ověření, že parametry linky odpovídají. Je-li zjištěna chyba, přijímač

nám tuto chybu nahlásí pomocí přerušení, které je definováno v chybách JESD204B specifikace.

Parametr N' se zjistí vynásobením hodnotou nibblu s čílem 4. To může být výhodné, jak pro vysílač, tak pro přijímač k nastavení N' na 16, pro převodníky s rozlišením v rozmezí od 8 bitů do 16 bitů. To umožňuje, že stejný vysílač a přijímač je použit pro více převodníků, což vede k zjednodušení celkového designu systému. Neúplný nibble pojme buď řídicí bity (CS) nebo koncové bity (TT), jak je definováno v JESD204B standardu. Rovnice $N' = N + CS + T$ musí být splněna. Řídicí bity (pokud jsou) jsou připojeny po LSB (nejméně významném bitu) do každého konverzního vzorku. K výpočtu počtu oktet přenášených za rámec F potřebujeme znát tyto následující parametry - počet převodníků, počet vzorků za rámec, velikost JESD204B slova a maximální rychlost linky. Aby bylo možné určit tento parametr, může být použita tato rovnice:

$$F = (M \times S \times N') / (8 \times L) \quad (3.5.1) [5]$$

3.5.2 Aplikační vrstva

Aplikační vrstva umožňuje speciální uživatelské konfigurace a mapování datových vzorků mimo typickou specifikaci JESD204B. Tímto můžeme dosáhnout více efektivnějšího využití rozhraní k snížení energie a dalších výhod.

Vysílač (AD převodník) a přijímač (FPGA) musí být nakonfigurovány na tyto speciální konfigurace. Toto je důležité mít na paměti. Přijímač a vysílač musí být nakonfigurovány identicky, aby data byla přenášena a interpretována správně.

Konfigurace aplikační vrstvy unikátním způsobem může být přínosné pro AD převodníky, které potřebují přenášet data ve velikostech vzorku, které mohou být odlišné od parametru N' (počet přenášených bitů po vzorku).

To by mohlo umožnit více vzorkům být znovu „zabalena“, takovým způsobem, že se rychlost linky sníží a celková účinnost linky zvýší. [5]

3.5.3 Datová vrstva

Datová vrstva bere paralelní rámcová data (zahrnuje AD převodníkové vzorky, řídicí bity a konečné bity) a výstup 8B/10B slov, které jsou uspořádány ve fyzické vrstvě a mohou být volitelně zakódovány pomocí scrambleru (používá se pouze anglická terminologie). 8B/10B systém poskytuje stejnosměrně vyvážené výstupní data a kontrolu chyb. Datová vrstva synchronizuje JESD204B linku přes proces sestavení linky.

Proces sestavení linky se skládá ze tří fází:

- synchronizace kódových skupin (CGS),
- sekvence zarovnání počátečního diferenciálního páru (ILAS),
- uživatelská data.

Během CGS, každý přijímač (FPGA) musí lokalizovat K28.5 znaky ve svém vstupním datovém toku, který je přenášen z AD převodníku, pomocí Clock and Data recovery (CDR) technik. Jakmile je detekován určitý počet po sobě jdoucích K28.5 znaků, na všech linkových diferenciálních párech, blok přijímače deaktivuje signál SYNC~ do vysílacího bloku. V JESD204B vysílací blok zachytí změny v SYNC~ a spustí ILAS na dalších lokálních multirámcových hodinách (LMFC).

Hlavním účelem ILAS je zarovnat všechny diferenciální páry na lince. Dalším účelem je ověřit, že parametry linky jsou správné a stanovit, kde jsou rámcové a multirámcové hranice v příchozím datovém toku. Během ILAS jsou parametry linky posílány do přijímače (FPGA), aby určily, jak budou data odeslána. ILAS se skládá ze 4 nebo více multirámců. Poslední znak nebo každý multirámec, je multirámcový zarovnávací znak /A/. První, třetí a čtvrtý multirámec začíná se znakem /R/ a končí znakem /A/. Data mezi těmito znaky jsou tzv. Ramp data. Přijímač používá konečný znak /A/ každého diferenciálního páru k zarovnání konců multirámců v přijímači. [4]

Druhý multirámec obsahuje znak /R/ a znak /Q/ následovaný parametry linky. /Q/ znak indukuje to, že v následujících datech je konfigurace parametrů linky. Dodatečné multirámce mohou být přidány do ILAS v případě potřeby přijímače. Po posledním /A/ znaku, posledního ILAS multirámce, se spustí uživatelská data.

Po dokončení CGS a ILAS fází, vysílač může začít rozesílat uživatelská data (což jsou vzorky z AD převodníku). V této fázi, uživatelská data proudí z vysílače do přijímače, podle parametrů linky, které byly definovány ve vysílači (AD převodník) a vysílány do přijímače (FPGA). Odtud, všechny šířky pásma z vysokofrekvenčního spektra, která byla digitalizována GSPS AD převodníkem, jsou nyní přenášena pro zpracování. Přijímací blok zpracovává a monitoruje data, které obdrží z hlášených chyb. Tyto chyby zahrnují nesprávnou disparitu (8B/10B chyba), nečekaný znak (8B/10B chyba), nečekaný řídicí znak, nesprávný ILAS

a časový posuv. (poznámka: 8B/10B je navržen tak, aby disparita byla udržována tak, že výstupní data jsou stejnosměrně vyvážena, zatímco udržuje dostatečné výstupní přechody pro CDR v přijímači). Pokud se objeví některá z těchto chyb, tak je hlášena zpět do vysílače dvěma způsoby:

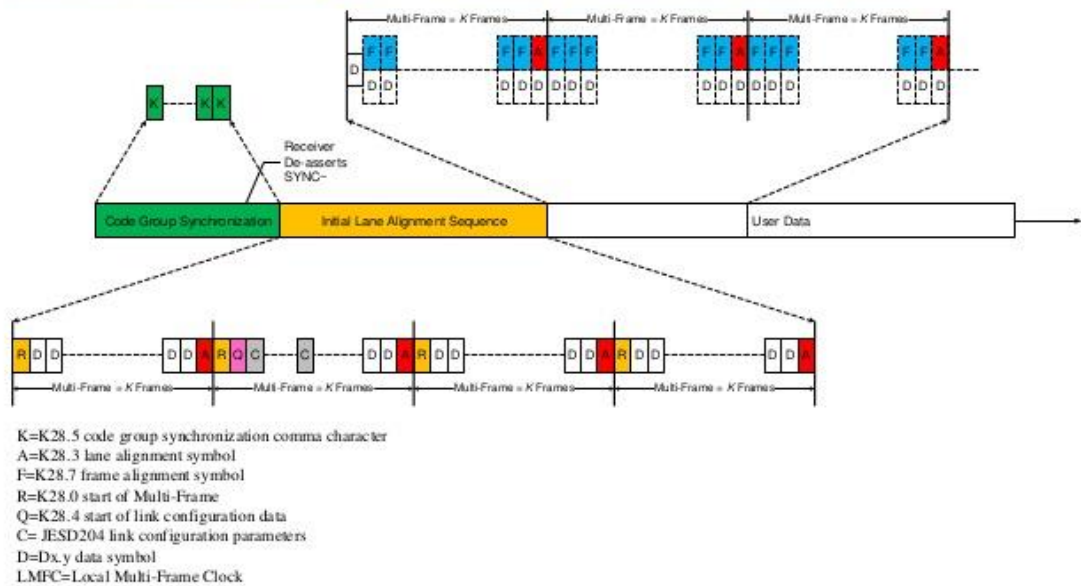
- SYNC~ aktivace – resynchronizace (SYNC~ staženo na low) je volána při každé chybě,
- SYNC~ hlášení - SYNC~ je změněno na logickou hodnotu high pro periodu rámcových hodin, jestliže se objeví chyba.

Během ILAS, datové vrstvy jsou odpovědné za zarovnání diferenciálních párů v přijímači. Umístění znaků /A/ se používá k zarovnání diferenciálních párů v přijímači. JESD204 A a B specifikace vyžadují, aby znaky /A/ byly od sebe alespoň 17 oktet. To vysokým způsobem zmírňuje účinky časového posunutí. V systémech JESD204 A a B je časové posunutí definováno třemi možnými scénáři:

- 1 vysílací blok a 1 přijímací blok,
- více vysílacích bloků a 1 přijímací blok,
- 1 vysílací blok a více přijímacích bloků.

Po dosažení uživatelské datové fáze, znakové nahrazení v datové vrstvě umožňuje rámcové a párové zarovnání, aby bylo monitorováno a opraveno v případě potřeby. Znakové nahrazení se provádí, jak na rámcových, tak i na multirámcových hranicích. Existují dva případy, jeden pro rámcově založené znakové nahrazení a druhý pro multirámcově založené znakové nahrazení. V rámcově založeném znakovém nahrazení, v případě, že poslední znak z rámce, je identický s posledním znakem z předchozího rámce na daném diferenciálním páru, pak vysílač nahradí tento znak /F/ znakem. To se také provádí v případě, že poslední znak z předchozího rámce je 0xFC, za podmínky povoleného kódování (scrambling). U multirámcově založeném znakovém nahrazení, v případě, že poslední znak z multirámcce, je identický s posledním znakem předchozího rámce na daném diferenciálním páru, poté vysílač nahradí znak /A/ znakem. Znakové nahrazení, je rovněž provedeno, jestliže poslední znak z předchozího multirámcce je 0x7C a kódování (scrambling) je povoleno. Ilustrace CGS, ILAS a uživatelských datových fází, společně se znakovým nahrazení je uvedeno na **Obrázek 8**. [5]

Data Link Layer – Link Establishment



Obrázek 8. Datová vrstva, CGS, ILAS, Datová sekvence [5]

Při znakovém nahrazení, přijímač musí dělat přesný opak toho, co se děje ve vysílači. Je-li detekován znak /F/, tak je nahrazen posledním znakem předchozího rámce. Když je detekován znak /A/, tak je nahrazen posledním znakem předchozího multirámce. Když je kódování (scrambling) povoleno, tak znak /F/ je nahrazen 0xFC a znak /A/ je nahrazen 0x7C. Jestliže přijímač detekuje dvě po sobě jdoucí chyby, můžou se znovu zarovnat diferenciální páry. Nicméně, data budou poškozena, zatímco se tato operace provádí. Stručný seznam všech řídicích znaků JESD204 je poskytnut v **Tabulka 2.** [5]

Control Character	Control Symbol	8-Bit Value	10-Bit Value, RD = -1	10-Bit Value, RD = +1	Description
/R/	K28.0	000 11100	001111 0100	110000 1011	Start of multiframe
/A/	K28.3	011 11100	001111 0011	110000 1100	Lane alignment
/Q/	K28.4	100 11100	001111 0010	110000 1101	Start of link configuration data
/K/	K28.5	101 11100	001111 1010	110000 0101	Group synchronization
/F/	K28.7	111 11100	001111 1000	110000 0111	Frame alignment

Tabulka 2. Tabulka řídicích znaků JESD204 [5]

3.5.4 Fyzická vrstva

Ve fyzické vrstvě jsou uspořádána data a jsou zde 8B/10B kódované data vysílány a přijímány při dané rychlosti. Fyzická vrstva zahrnuje serializér/deserializér (SERDES) bloky, přijímače a CDR. Jak již víme, JESD204B specifikace se dělí na 3 rychlostní stupně. **Tabulka 3** ukazuje na tyto rychlosti, diferenční napětí, náběžnou a sestupnou dobu, celkový jitter pro signály ve fyzické vrstvě JESD204B standardu, podle každého rychlostního stupně.

Parameter	OIF-Sx15-01.0	CEI-6G-SR	CEI-11G-SR
Line Rate (Gbps)	≤3.125	≤6.375	≤12.5
Output Differential Voltage (mVppd)	500 (min) 1000 (max)	400 (min) 750 (max)	360 (min) 770 (max)
Output Rise/Fall Time (ps)	>50	>30	>24
Output Total Jitter (pp UI)	0.35	0.30	0.30

Tabulka 3. JESD204B specifikace fyzické vrstvy [5]

Vyšší rychlostní stupně mají sníženou amplitudu signálu, aby byla vyšší rychlost přeběhu (rychlost změny napětí za jednotku času), a tak udržovat data eye diagramu (okno času, během kterého předpokládáme, že datové bity jsou platné) pro správný přenos signálu. Tyto vysokorychlostní signály s rychle stoupajícími a klesajícími hranami určují přísná omezení při návrhu obvodového řešení. [5]

3.5.5 Kódující (Scrambling) vrstva – volitelná

Mezi Transportní a Datovou vrstvou můžeme využít kódovacího/dekódovacího bloku. Při přenosu dat se také používá pseudonáhodných sekvencí, generovaných LFSR (lineární zpětnovazební posuvný registr), jako scrambleru (v publikacích se používá pouze anglická terminologie), pro randomizaci datového proudu. Tato technika je využívána kvůli dlouhým sekvencím nul a jedniček, například pro přenos digitálního signálu.

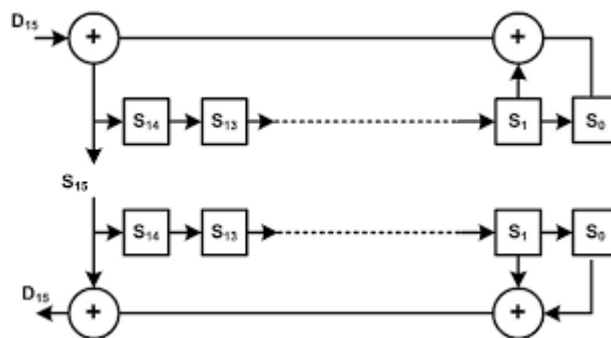
Datový scrambler je implementován kvůli snížení spektrálních špiček na vysokorychlostních sériových diferenciálních párech mezi vysílacím blokem a přijímacím blokem.

Scrambler využívá synchronního kódovacího vzorku s polynomem

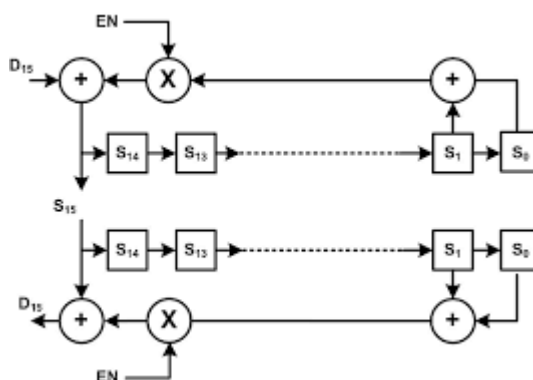
$$1 + x^{14} + x^{15}$$

Data jsou kódovány prioritně podle 8B/10B kodéru a dále jsou dekódovány. 2 posuvné registry na vstupu a výstupu musí být nastaveny na stejný počáteční stav, aby scrambler fungoval.

Důležité je upozornit na to, že kódování nezačne, dokud první oktet následuje ILAS. To znamená, že CGS a ILAS nejsou zakódovány. Desrambler je implementován tak, aby se sám synchronizoval ke scrambleru, po dvou oktety z dat. [5]



Obrázek 9. Blokové schéma scrambleru [5]



Obrázek 10. Blokové schéma descrambleru [5]

3.6 JESD linkové parametry/znaky

3.6.1 Parametry linky

L – počet diferenciálních párů v převodníkovém zařízení

M – počet převodníků v převodníkovém zařízení

S – počet přenášených vzorků v převodníku za rámeček

N' – počet přenášených bitů za vzorek (velikost JESD204 slova)

N – počet konverzních bitů v převodníku

F – počet oktět za rámeček

HD – vysoká hustota zápisu uživatelských dat

3.7 JESD204 Terminologie

Oktet – skupina 8 bitů, sloužící pro vstup do 8B/10B kodéru a pro výstup z dekodéru

Nibble – sada 4 bitů, která je základní pracovní jednotkou pro JESD204 specifikaci

Znak – 10 - bitový symbol generovaný podle schéma kódování

Znakové hodiny – hodiny běží na znakové rychlosti používané pro zpracování znaků v systému

Kódová skupina – sada 10 bitů v sériovém datovém proudu používaného pro zprostředkování oktět

disparita – měřítko stejnosměrného proudu nevyváženosti na lince používaná ke kódování a dekódování 8B/10B kódových skupin

Rámeček – sada po sobě jdoucích oktět, u kterých pozice každé oktety může být určena odkazem na zarovnaném rámcovém signálu

Rámcové hodiny (Frame clock) – systémové hodiny v JESD204A, které běží na rámcové frekvenci. V systémech JESD204 A jsou rámcové hodiny absolutním časovým odkazem v systému

Multirámeček – sada po sobě jdoucích rámců ve kterém pozice rámců může být určena ve vztahu k zarovnání multirámcového znaku

Multirámcové hodiny nebo Local Multi-Frame Clock (LMFC) – hodiny v systému, které běží na multirámcové rychlosti a určuje lokaci více multirámcových znaků, které jsou používány pro zarovnání diferenciálních párů. Typicky jsou tyto hodiny jinou verzí hodinového rámce, který je blízko vysílači/přijímači. V systémech JESD204B jsou tyto hodiny absolutním časovým odkazem.

diferenciální pár – diferenciální pár vodičů mezi vysílačem a přijímačem

datová linka neboli linka – sada dat diferenciálního páru, které poskytují kanál pro data posílané z vysílače do přijímače

multi-point linka – datová komunikační linka se třemi nebo více zařízeními. Tyto tři zařízení se skládají, buď ze dvou vysílačů a jednoho přijímače nebo z dvou přijímačů a jednoho vysílače.

Vysílač – místo pro kódování, uspořádání a řízení dat

Vysílací blok – všechny vysílače na lince

Přijímač – místo pro zachycení a dekodování dat

Přijímací blok – všechny přijímače na lince

Převodník – samotný převodník v pouzdru (package)

Převodníkové zařízení – zařízení s jedním nebo více převodníky

[3]

4 MĚŘENÍ JITTERU

Měření jitteru obsaženého v signálu dnes představuje jedno ze základních měření v oblasti vysokorychlostních signálů a lze jej využít pro odhad chybovosti (BER; Bit Error Rate) datového spoje, kdy může nahradit pro nízké hodnoty chybovosti časově náročné přímé měření BER. Možnost měřit jitter přímo v systému je ve většině reálných případů omezena na použití osciloskopu, spektrálního analyzátoru nebo podobného specializovaného přístroje, který je nutné do obvodu připojit pomocí sondy.

Sondou je měřený obvod do jisté míry ovlivněn a snižuje se tak přesnost měření. V některých případech je navíc připojení samotné sondy problematické. V případě vysokorychlostních diferenčních signálů často vyžaduje i pájení a použití přídavných zakončovacích rezistorů. Proto jsou zkoumány možnosti implementace obvodu pro měření jitteru přímo v cílovém obvodu (typicky FPGA). Takový měřič by byl součástí samotného zařízení a prakticky by neovlivňoval zkoumaný signál.

Jitter by bylo možné měřit přímo v cílové aplikaci, bez nutnosti připojení měřicích přístrojů (například osciloskopu) a bez jakýchkoliv zásahů do zapojení. Samotný zkoumaný spoj tak nemusí být fyzicky vůbec přístupný (například signál ve vnitřní vrstvě plošného spoje). Tuto metodou by navíc bylo možné použít v již existujících systémech s obvody FPGA bez nutnosti jejich fyzické modifikace.

Pro obvody FPGA známe několik metod měření jitteru. Metoda Follow-me zakládá na sledování jitteru jemným fázovým posunem hodinového signálu s využitím obvodu pro úpravu hodinového signálu v FPGA (DCM; Digital Clock Manager). Takto měříme jitter na nižších kmitočtech. Vlivem omezené šířky pásma zpětnovazební smyčky, obvod nedovoluje zachytit vysokofrekvenční jitter. Metoda v literatuře není moc popsána, což se ukázalo být překážkou pro její implementaci do obvodu a při ověření jejich vlastností. Autoři zde uvádějí jen výsledky behaviorální simulace obvodu a výsledky reálných měření chybí.

Nová technika pro měření jitteru v FPGA je založena na principu TDC (Time to Digital Conversion) měření periody datového signálu a umožňuje měřit vysokofrekvenční jitter v synchronních systémech. Její nevýhodou je omezený pracovní kmitočet, malé rozlišení (v uvedené implementaci pouze 1 ns) a poměrně vysoké nároky na hardware (velké množství použitých hodinových signálů).

Obvody BO-CDR mají obvykle dostatečnou šířku pásma, která jim umožňuje bezpečně sledovat jitter na nižších frekvencích, který tak pro jejich analýzu není podstatný. Jejich chybovost je způsobena především vysokofrekvenčním jitterem, který by měl být při jejich analýze hlavní sledovanou veličinou. Vysokofrekvenční jitter metodou Follow-me nelze zachytit, proto nemůže být pro tato měření použita. Použití metody TDC není vhodné z hlediska malého rozlišení a vysokých nároků na hardware. Pro odhad vlastností kanálu byla proto vyvinuta nová metoda měření jitteru v FPGA založená na **vzorkování diagramu oka**. [6]

4.1 Jitter

Digitální (binární) signál je průchodem přenosovou soustavou vždy zkreslen. Z hlediska bloku CDR se zkreslení projevuje jako časová změna polohy průchodu signálu rozhodovací úrovní komparátoru přijímače. Časový posun je způsoben šumem, rušením, omezenou šířkou pásma kanálu a dalšími neideálními vlastnostmi kanálu. Posun je tím větší, čím menší je strmost hran a čím menší je poměr signálu k šumu.

Relativní posun hran vzhledem k jejich ideální poloze lze definovat pomocí jitteru. Jitter je ve standardu JEDEC definován jako časová odchylka hrany generované fázovým závěsem (PLL) od své nominální pozice. Standard ITU jitter definuje jako krátkodobou nekumulativní odchylku významných událostí digitálního signálu od ideální jejich pozice v čase. Touto významnou událostí je v kontextu obnovy bitové synchronizace hrana datového a hodinového signálu.

Tzv. celkový jitter (TJ; Total Jitter), který je v signálu přítomen, je složen ze dvou základních složek: náhodného jitteru (RJ; Random Jitter) a deterministického jitteru (DJ; Deterministic Jitter). Náhodný jitter je teoreticky neomezený (odchylka může nabývat libovolně velkých hodnot) a je popsán normálním (Gaussovým) rozložením pravděpodobnosti. Je plně charakterizován směrodatnou odchylkou a střední hodnotou. Jeho zdrojem je elektrický šum v systému, který ovlivňuje okamžik průchodu signálu rozhodovací úrovní komparátoru přijímače a vytváří tak časové odchylky (chyby) v okamžicích přechodů.

Deterministický jitter není způsoben náhodnými jevy a rozložení hustoty pravděpodobnosti není normální. DJ je vždy omezen a jeho velikost tak lze udávat jako špičkovou hodnotu p-p (peak-to-peak). Zdrojem deterministického jitteru jsou nedokonalosti přenosové trasy, vysílače a přijímače. Pro snadnější analýzu je deterministický jitter dále dělen na jednotlivé komponenty, které mají specifický charakter. [6]

- **DCD (Duty Cycle Distortion) jitter,**
 - je dán nestejnou dobou trvání bitu 1 a 0. Bývá způsoben nestejnou dobou náběžné a sestupné hrany signálu nebo posunutím rozhodovací úrovně komparátoru přijímače.
- **DDJ (Data Dependent Jitter) jitter,**
 - je závislý na přenášené datové sekvenci (konkrétní posloupnosti bitů). Je zapříčiněn omezenou šířkou pásma systému a jeho jednotlivých komponent.
- **PJ (Periodic Jitter) jitter,**
 - má periodický charakter (například sinusový) a je vázán na přenášený datový signál. Způsoben je nedokonalými vlastnostmi systému (například zvlnění zdrojů napájení) a zpravidla je poměrně malý.

- **UBJ (Uncorrelated Bounded Jitter) jitter.**

- je do signálu vnesen vazbami s okolními systémy a není korelován s datovým signálem.

Jeho zdroji jsou rušení, kapacitní a induktivní vazby a šum spínaných napájecích zdrojů. Komponenty jitteru, které se mění jen velmi pomalu (pro standard SONET jsou to komponenty s frekvencí pod 10 Hz), se zpravidla mezi jitter nezařazují a bývají označovány jako drift frekvence (frequency wander). Jeho příčinou může být například teplotní závislost parametrů obvodu.

Reálný signál obsahuje náhodný i deterministický jitter, celkový jitter je pak roven konvoluci obou složek. Jednotka jitteru je v obou případech (pro RJ i DJ) čas [s]. Tuto hodnotu je ale vždy nutné uvádět s periodou zkoumaného signálu, bez níž postrádá význam. Aby bylo možné srovnávat systémy s různými přenosovými rychlostmi, byla zavedena poměrná jednotka UI (Unit Interval), vztažená k délce bitu datového signálu vztahem: [6]

$$J_{[UI]} = \frac{J_{[s]}}{T_{bit}}, \quad (4.1) [6]$$

kde $J_{[s]}$ je velikost jitteru [s] a T_{bit} je perioda bitu [s]. Tato jednotka je také používána pro vyjádření fáze signálu:

$$\varphi_{[UI]} = \frac{\varphi_{[rad]}}{2\pi} = \frac{\varphi_{[deg]}}{360^\circ}, \quad (4.2) [6]$$

kde $\varphi_{[rad]}$ je velikost fáze v radiánech a $\varphi_{[deg]}$ je velikost fáze ve stupních.

Vlastnosti systému z hlediska jitteru se hodnotí ze tří základních hledisek:

- generování jitteru (Jitter Generation),
- přenos jitteru (Jitter Transfer),
- tolerance jitteru (Jitter Tolerance).

Generování jitteru lze definovat jako množství jitteru přidané k původnímu signálu. Tento parametr je používán pro charakterizaci komponent vysílače, jako jsou laserové budiče, paralelně-sériové převodníky, budiče, zesilovače, obvody CDR a limitující zesilovače.

Přenos jitteru vyjadřuje množství jitteru ze vstupního signálu, které je přeneseno ze vstupu na výstupní signál. Toto množství je obvykle vyjadřováno poměrem vstupní a výstupní

hodnoty jitteru pro konkrétní frekvence jitteru. Tato veličina je používána u komponent typu regenerátoru dat, tedy například obvodů CDR.

Schopnost zařízení korektně přijímat signál obsahující jitter se nazývá tolerance jitteru. Lze ji vyjádřit jako amplitudu jitteru na dané frekvenci ve vstupním signálu, kterou je přijímač schopen zpracovat, aniž by byla překročena stanovená chybovost. Zakreslením hodnot tolerance pro různé frekvence jitteru do grafu lze získat charakteristiku tolerance jitteru přijímače. Tímto údajem bývají charakterizovány přijímací obvody, jako obvody CDR a paralelně-sériové převodníky. [6]

5 FPGA

Programovatelná hradlová pole (FPGA, Field Programmable Gate Array) jsou speciální číslicové integrované obvody obsahující různě složité programovatelné bloky propojené konfigurovatelnou maticí spojů. Field Programmable v názvu je to, čím se FPGA odlišují od zákaznických integrovaných obvodů – obvod je totiž nakonfigurován u zákazníka.

FPGA obvody dnes nacházejí uplatnění v široké škále aplikací díky své programovatelnosti, snadnému návrhu, flexibilitě, neustále klesajícím cenám a zvolna se snižující spotřebě energie vlastním čipem. Typické použití je v oblasti menších sérií navrhovaných zařízení, kdy se nevyplatí návrh zákaznického integrovaného obvodu a současně konvenční řešení systému s procesorem už není vhodné. Další aplikace můžeme nalézt například v oblasti prototypů složitějších zákaznických integrovaných obvodů. Velká programovatelná hradlová pole dnes umožňují i implementaci komplikovaných procesorů.

Kromě FPGA obvodů se lze běžně setkat i s tzv. CPLD obvody (Complex Programmable Logic Devices). CPLD použijeme spíše pro návrh jednoduché propojovací logiky při integraci složitějších prvků (glue logic). [7]

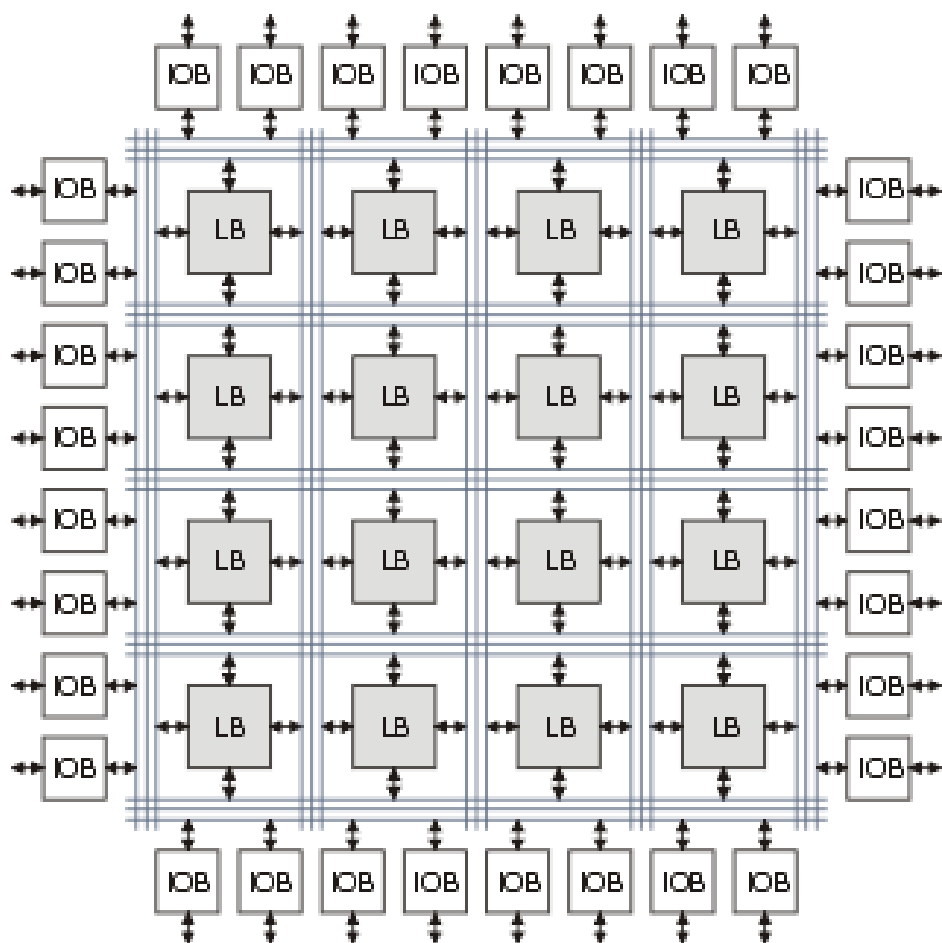
5.1 Historie

Historické kořeny moderních programovatelných polí jsou v prvních programovatelných pamětech typu PROM (firma Radiation, 1970) a jejich zákaznický programovatelných verzím EPROM (Intel, 1971) a EEPROM (Intel, 1978). Permanentní paměti jako takové ale neumožňovaly úspornou realizaci logické funkce. Logickým vývojem

krokem proto byl vznik tzv. FPLA obvodů, ty ale byly drahé a pomalé. Aby bylo možné dosáhnout vyšší rychlosti logiky, byla součtová matice realizována jako fixní – vznikla koncepce PAL obvodů (Programmable Array Logic, Monolithic Memories, 1978) s programovatelnou maticí AND a pevnou maticí OR. S postupně se zlepšujícími technologiemi výroby integrovaných obvodů bylo možné vyrábět programovatelné obvody s vyšší kapacitou a realizující složitější logické funkce. Kvůli rostoucí velikosti obvodů se začalo později místo rozšiřování logických funkcí užívat spíše skládání více matic PLD obvodů do jednoho pouzdra. Vznikly tak obvody, které dnes nazýváme CPLD. Od CPLD byl už pak jen malý krok k prvním FPGA obvodům (Xilinx, 1984). Dnes dostupná FPGA se ovšem od architektur z poloviny osmdesátých let významně odlišují. Trendem jsou obvody, které kromě elementárních programovatelných logických bloků obsahují i další komplexní podpůrné bloky. [7]

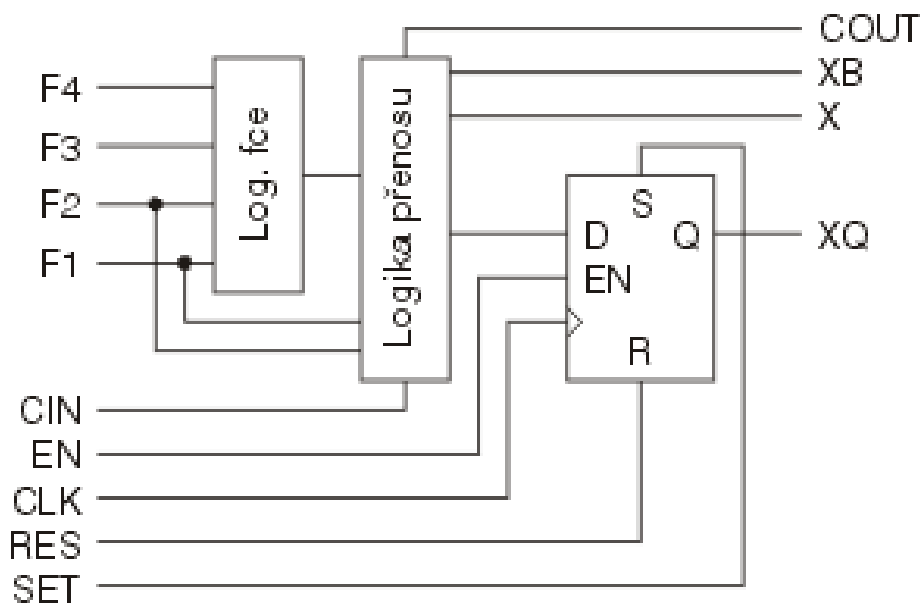
5.2 Struktura

Obvody typu FPGA mají z programovatelných obvodů nejobecnější strukturu a obsahují nejvíce logiky. Současné největší obvody FPGA obsahují až 6 milionů ekvivalentních hradel (typické dvouvstupové hradlo NAND). Typickou strukturu obvodu FPGA znázorňuje **Obrázek 11**.



Obrázek 11. Typická struktura obvodu FPGA [8]

Bloky označené IOB (Input/Output Block) představují vstupně-výstupní obvody pro každý v-v pin FPGA. Tyto bloky obvykle obsahují registr, budič, multiplexer a ochranné obvody. Bloky LB (Logic Block) představují vlastní programovatelné logické bloky. Všechny bloky mohou být různě propojeny globální propojovací maticí. Nejpoužívanější struktura konfigurovatelného logického bloku je znázorněna na **Obrázek 12**.



Obrázek 12. Struktura logického bloku [8]

FPGA obvykle umožňují propojit některé signály logických bloků přímo se sousedním bez nutnosti využívat globální propojovací matici. Takovéto spoje mají mnohem menší zpoždění a umožňují tak realizovat například rychlé obvody šíření přenosu, což je nezbytné pro sčítačky nebo násobičky.

Kromě bloků znázorněných na předchozích obrázcích integrují výrobci do FPGA další prvky. Většina moderních FPGA obsahuje několik bloků rychlé synchronní statické paměti RAM. Velmi často obvody FPGA obsahují PLL (Phase Locked Loop) nebo DLL (Delay Locked Loop) pro obnovení charakteristik hodinového signálu, případně pro násobení nebo dělení jeho frekvence. [8]

5.3 Xilinx

Firma Xilinx patří k jedním ze dvou největších výrobců FPGA. Jednotlivé typové řady programovatelných hradlových polí Xilinx jsou porovnány v následujících řádcích.

5.3.1 Spartan 6

Vyrobeny na 45 nm technologii.

- Programovatelná systémová integrace

Nejvyšší počet pinů logického poměru pro vstup/výstup připojení
Přes 40 vstup/výstup standardů pro zjednodušení systémového designu
PCI Express s integrovaným koncovým blokem

- Zvýšený systémový výkon

Až 8 nízkospotřebných 3.2 GB/s sériových transceiverů
800 MB/s DD3 kontrolerem integrované paměti

- Celkové snížení spotřeby

1.2 V nebo 1.0 V napětí jádra
Žádná spotřeba energie s režimem spánku

- Výhody

Jsou nejvíce cenově výhodnými FPGA. Nabízí průmyslové přední propojení, jako je vysoká logika u pinů. Poskytuje vysoké číslo podporujících vstup/výstup protokolů. Zařízení jsou ideální pro celou řadu pokročilých aplikací u automobilového zpravodajství. [9]

5.3.2 Virtex 7

Vyrobeny na 28 nm technologii.

- Programovatelná systémová integrace

Až 2M logických buněk, VCXO komponent, AXI IP a AMS integrace

- Zvýšený systémový výkon

Až 2.8 TB/s s celkovou šířkou pásma až 96 x 13.1G GTS, až 16 x 28.05G GTS, 5.335 GMACs, 68 MB BRAM, DDR3-1866

- Celkové snížení spotřeby
Až 70 % nižší spotřeby než multi-chipové řešení
- Výhody

Jsou optimalizované pro systémový výkon a integraci na 28 nm.

Nejlepší ve své třídě v poměru výkon/watt, DSP výkonu a v vstup/výstup šířky pásma v našem designu. Tato rodina je používána v řadě aplikacích, jako například u 10G do 100G sítí, přenosného radaru a u ASIC prototypů. [9]

5.3.3 Kintex 7

Vyrobeny na 28 nm technologii

- Programovatelná systémová integrace
Až 478 tisíc logických buněk, VCXO komponent, AXI IP a AMS integrace
- Zvýšený systémový výkon
Až 32 x 12.5G GTs, 2.845 GMACs, 34 MB BRAM, DDR3-1866
- Celkové snížení spotřeby
50 % menší spotřeba než u předešlých 40 nm zařízení
- Výhody

Poskytuje nejlepší poměr cena/výkon/watt na 28 nm technologii a zároveň nám dává vysoké DSP poměry. Kintex-7 rodina je vhodná pro aplikace zahrnující bezdrátové 3G a 4G a ploché displeje. [9]

5.3.4 Artix 7

Další z 28 nm technologie.

- Programovatelná systémová integrace

Až 251 tisíc logických buněk, AXI IP a Analog Mixed Signal integrace

- Zvýšený systémový výkon

Až 16 x 16.6G GTs, 930 GMACs, 13 MB BRAM, 1.2 GB/s LVDS, DDR3-1066

- Celkové snížení spotřeby

65 % méně statický a 50 % menší spotřeba než u 45nm generace

- Výhody

Dodávají nejnižší spotřebu a cenu u 28 nm generace a jsou optimalizované na nejvyšší výkon/watt, AMS integraci. Rodina se nejvíce uplatňuje u aplikací, které jsou citlivé na spotřebu, třeba softwarově definované radio. [9]

Všechny FPGA Xilinx používají konfiguraci pomocí statické paměti RAM. To znamená, že po připojení napájení je nutné vždy nahrát znovu konfiguraci. Výhodou tohoto řešení je téměř nekonečná pře-programovatelnost FPGA a také rychlost. Konfigurační propojky pracující na principu paměti RAM jsou totiž rychlejší než přepínače na principu např. EEPROM.

Pro využití obvodů FPGA v našich podmínkách je důležitý i údaj o podpoře ve vývojovém systému WebPACK. Firma Xilinx totiž nabízí svůj vývojový systém ISE v několika verzích. Cena roční licence na plný systém se pohybuje v tisících dolarů. Jako alternativu však Xilinx nabízí ISE WebPACK, který lze zdarma stáhnout z webových stránek firmy. Tento systém však samozřejmě nepodporuje všechny typové řady FPGA. [8]

5.4 Vývojové prostředky

5.4.1 Software

Pro vývoj aplikací s FPGA existuje několik návrhových systémů. Pro vývoj je nutné použít minimálně dvou nástrojů. Prvním je nástroj pro syntézu, který převede většinou textový popis návrhu v některém HDL jazyce na netlist využívající obecné logické bloky. Druhý nástroj zajistí konverzi obecného netlistu na netlist využívající prostředky konkrétního FPGA a zajistí jejich "optimální" rozmístění a propojení. Nástroje pro rozmístění a propojení obvykle nabízejí pouze výrobci programovatelných hradlových polí. Prostředky pro syntézu nabízejí i jiné firmy.

Pokud chceme začít pracovat s obvody FPGA, musíme si obstarat základní programové vybavení od výrobce obvodů a případně další software od třetí strany. Kromě nástrojů pro syntézu je velmi výhodné používat ještě simulátor, čímž se může předejít chybám již v průběhu návrhu. Ceny vývojových prostředků jsou však velmi vysoké (ceny licencí na jeden rok se obvykle pohybují od 1000 do 2000 dolarů). Firma Xilinx nabízí pro FPGA s menší hustotou logiky mnohem levnější alternativu. Vývojový systém ISE WebPACK je totiž zadarmo.

Tento vývojový systém pro FPGA firmy Xilinx je omezenou verzí jejich kompletního systému. Omezení se však týká pouze velikosti hradlových polí, pro které je možno prostředí použít. Navíc neobsahuje některé rozšířené součásti jako například plnohodnotný editor výsledného propojení. WebPACK je i přesto plně funkční a plnohodnotný návrhový systém. Kromě vlastního prostředí WebPACK je možné zdarma získat i omezenou verzi HDL simulátoru ModelSim XE, což je verze s předkompilovanými knihovnamy primitiv pro FPGA Xilinx. Omezení simulátoru spočívá ve zpomalení jeho funkce pro velké návrhy. Simulátor je tedy opět plně funkční, pouze doba simulace složitěho návrhu je několikanásobná oproti plné verzi. [8]

5.4.2 Hardware

Pro bližší seznámení s programovatelnými hradlovými poli je výhodné použít již hotovou vývojovou desku. Na trhu jich je několik a jejich ceny se pohybují řádově od \$200. Pokud nechceme za desku utratit tolik peněz, ale přesto bychom pro začátek použili ověřenou funkční desku, můžeme si sami vyrobit, příp. nechat vyrobit vývojovou desku podle zveřejněné open-source dokumentace. Takovou deskou je například FPGA-evb-S2. [8]

5.5 Hodiny

Většina z obvodů FPGA se vyrábí se synchronním obvodem, který vyžaduje hodinový signál. FPGA obsahují globální a regionální sítě pro vstupy hodin a reset, takže mohou být dodány s minimálním časovým posuvem. FPGA obvykle obsahují i analogové PLL a/nebo DLL komponenty k syntéze nových hodinových frekvencí, stejně jako k zmírnění jitteru. U složitějších návrhů lze použít více hodin s různými frekvencemi a fázové vztahy, z nichž každý tvoří samostatné hodinové domény. Tyto hodinové signály mohou být lokálně generovány oscilátorem, nebo mohou být získány z vysokorychlostního sériového datového proudu. FPGA obvykle obsahují bloky RAM, které jsou schopny pracovat jako duální port s různými hodinami, pomáhající při stavbě FIFO a u paměti duálního portu, který připojuje lišící se hodinové domény. [10]

6 NÁSTROJ PRO POPIS HARDWARE

VHDL je programovací jazyk sloužící pro popis hardware. Používá se pro návrh a simulaci digitálních integrovaných obvodů, například programovatelných hradlových polí (CPLD, FPGA, ...), nebo různých zákaznických obvodů (ASIC).

VHDL je standardem IEEE od r. 1987, byl revidován v roce 1997 a je použitelný i pro návrh analogových obvodů. Jedná se o typovaný programovací jazyk. VHDL má prostředky pro popis paralelismu, konektivity a explicitní vyjádření času. Jazyk VHDL se používá jak pro simulaci obvodů, tak i pro popis integrovaných obvodů, které se mají vyrábět.

Zkratka VHDL znamená VHSIC Hardware Description Language (česky jazyk pro popis hardware), kde VHSIC je zkratka z Very-High-Speed Integrated Circuit (česky velmi rychlé integrované obvody). [11]

7 NÁVRH OBVODOVÉHO ŘEŠENÍ

4.1 Převodník ADC34J25

Pro můj projekt byl vybrán převodník ADC34J25. Tento převodník byl vybrán vedoucím práce, panem inženýrem Soběslavem Valachem.

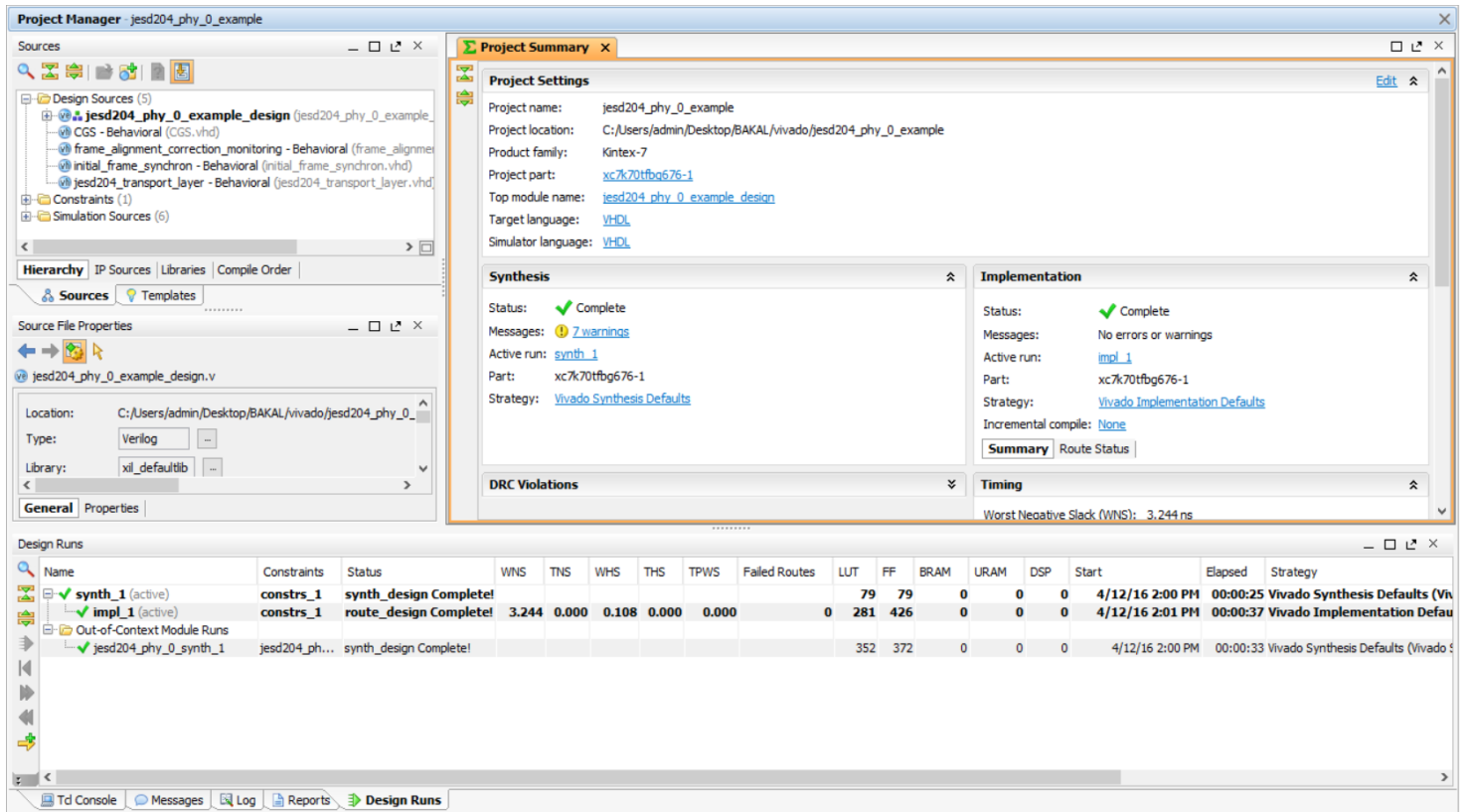
Je to analogově digitální převodník s JESD204B rozhraním. Tento převodník se vyznačuje vysokou linearitou a nízkou spotřebou. Jeho maximální rychlost je 160 MSPS. Obsahuje fázový závěs, který násobí příchozí vzorkovací hodiny dvaceti k odvození bitových hodin, které jsou používány k serializaci 12-bitových dat z každého kanálu. Zařízení podporuje podtřídou 1 s rychlostí rozhraní maximálně 3.2 Gbps. Vstupní signál SYSREF umožňuje kompletní synchronizaci systému. Navržené schéma převodníku ADC34J25 je zobrazeno na **Obrázek 14**.

4.2 Schéma

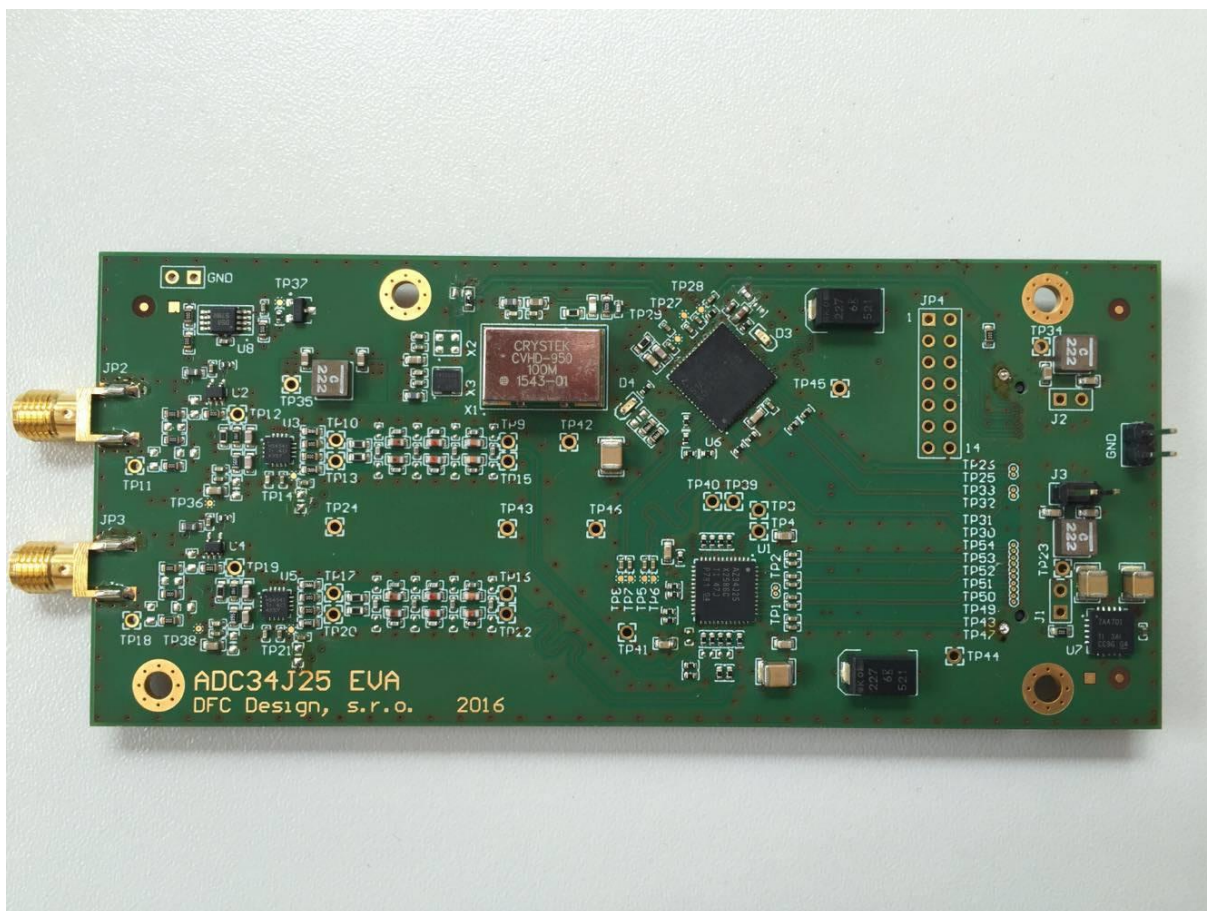
Převodník je napájen napětím 1.8 V. Kondenzátory ve vrchní části schématu slouží ke stabilizaci napájení. Kondenzátory ve střední části převodníku jsou tzv. vazební kondenzátory. Tyto kondenzátory oddělují stejnosměrnou složku a propouští pouze střídavou složku. Integrovaný obvod SN65CML100, ve schématu označený jako U9, je vysokorychlostní překladač/opakovač (translator/repeater). Jedná se o 1.5 Gbps - LVDS/LVPECL/CML-TO-CML překladač/opakovač a je napájen napětím 3.3 V. Kompletní obvodové řešení je zobrazeno na **Obrázek 15** a na **Obrázek 16**.

4.3 Moje práce

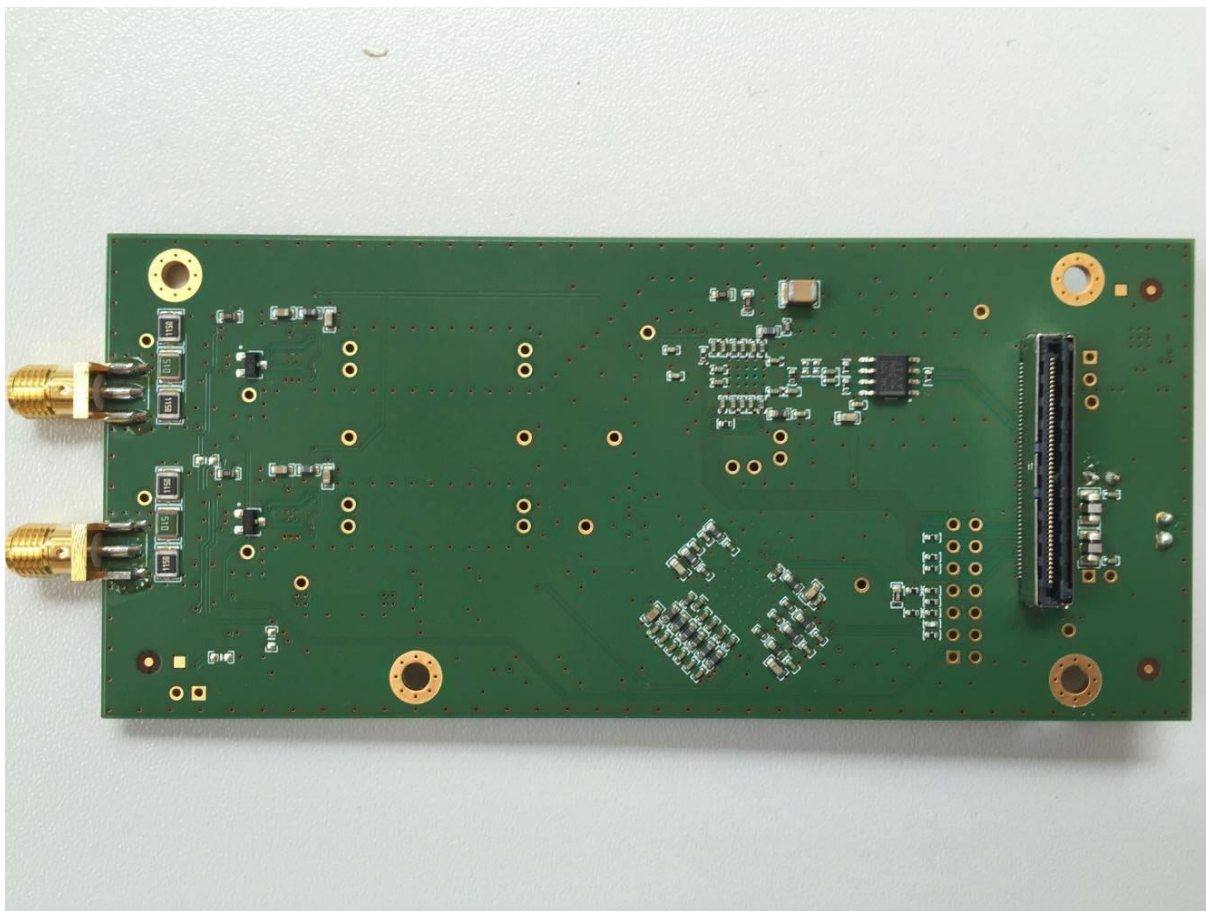
Můj podíl na obvodovém řešení spočíval přiřazením vstupně/výstupních pinů (tzv. Mapování pinů). Hradlové pole (FPGA) obsahuje transceivery, které vyžadují svůj vlastní hodinový signál. Bylo tedy potřeba zauvažovat nad pravidly, které tento problém přináší. Veškeré pravidla jsou dány dokumentacemi pro hradlová pole. Mapování pinů se provádí pomocí UCF (user constraint file) souboru, ve kterém se definují všechny používané piny na místa dané pravidly a požadavky na hradlové pole. Úspěšným přeložením celého implementovaného projektu jsem ověřil, že moje úloha vyhovuje všem pravidlům a požadavkům.



Obrázek 13. Úspěšné přeložení designu



Obrázek 15. Kompletní obvodové řešení – horní strana desky



Obrázek 16. Kompletní obvodové řešení - spodní strana desky

8 VLASTNÍ IMPLEMENTACE FIRMWAREU

Pro vytvoření firmwaru je vhodné mít k dispozici prostředí, ve kterém budeme pracovat. Vybral jsem si prostředí Vivado Design Suite, které poskytuje firma Xilinx na jejich stránkách. Toto prostředí nabízí LogiCORE™ IP JESD204 PHY core, které implementuje JESD204B funkce fyzické vrstvy. Podporuje sdílení transceiverů mezi vysílač a přijímač. Při generování tohoto core jsem nastavil následující parametry:

- typ transceiveru – GTXE2,
- počet diferenciálních párů na lince – 4,

- rychlost linky – 3.125 Gbps,
- typ fázového závěsu – QPLL.

8.1 Nastavení parametrů linky

Všechny potřebné informace k nastavení parametrů linky získáme z příslušné dokumentace, kterou poskytuje výrobce daných převodníků. V našem obvodu je použito zařízení od firmy Texas Instruments ADC34J2X.

1. Kolik převodníků naše zařízení obsahuje (M)?

ADC34J2X je Quad channel, tudíž zařízení obsahuje 4 převodníky.

$$M = 4$$

2. Jaké je vzorkovací rozlišení převodníku v zařízení (N)?

12 – bitové rozlišení

$$N = 12$$

3. Jaká je velikost JESD204 slova (N')?

Vzorkovací rozlišení převodníku se dělí na nibbly. 12 -bitové převodníky mají 3 nibbly. K určení hodnoty N', vynásobme počet nibblů, číslem 4. Může být výhodné mít N'=16, aby 8 až 16 -bitové převodníky mohli být řešeny stejným vysílačem/přijímačem.

Neúplný nibble je doplněn, buď řídicími bity (CS) nebo koncovými bity (T) a musí odpovídat:

$$N' = N + CS + T \quad (8.1.1) [3]$$

12 -bitový převodník s 16 -bitovou velikostí JESD204 slova ponechává 4 bity, buď pro řídicí bity a/nebo pro koncové bity.

4. Kolik se do rámce vejde vzorků (S)?

V našem případě je výrobcem nastavena hodnota S na 1.

$$S = 1$$

Pro jiné převodníky, lze hodnotu S vypočítat, dle nadcházejícího postupu.

- Určeno pomocí rámcové rychlosti a vysílací/přijímací vnitřní charakteristikou.
- U jednoduchých nízko rychlostních převodníků je často hodnota S nastavena na 1. U složitějších vysoko rychlostních převodníků hodnota S, závisí na:

$$\text{JESD204 rychlosti dat} = (\text{M} \times \text{vzorkovací frekvence AD převodníku}) / (\text{faktor decimace AD převodníku}) \quad (8.1.2) [3]$$

Požadované rychlosti hodinového rámce

- Hodnota S je určena pomocí :

$$S = (\text{JESD rychlost dat}) / \text{rychlost frame clocku (FC)} \quad (8.1.3) [3]$$

Příklad 1 : Jednoduchý AD převodník (bez decimace) s vzorkovací frekvencí = 156MHz (JESD204 rychlost dat = 156.25 MHz), požadovaná rychlost hodinového rámce = 156.25 MHz nám dává S = 1

Příklad 2 : Dvojitý AD převodník (s decimací = 2) se vzorkovací frekvencí = 625 MHz, požadovanou rychlostí hodinového rámce = 156.25 MHz nám dává:

$$S = (2 \times 625 \text{ MHz}) / 2 / 156.25 \text{ MHz} = 4 \quad (8.1.4) [3]$$

5. Kolik je potřeba diferenciálních párů na lince (L)?

Opět hodnota, která je dána výrobcem našeho převodníku.

$$L = 4$$

Pro jiné převodníky, lze také vypočítat, dle následujícího postupu.

- určeno pomocí požadovaného maxima rychlosti linky za diferenciální pár. Rychlost diferenciálního páru na lince, může být vypočítána, pomocí následujícího vzorce (FC = frame clock rate) :

$$\text{Lane Line Rate} = (M \times S \times N' \times 10/8 \times \text{FC}) / L \quad (8.1.5) [3]$$

Příklad 1 : samotný 14 -bitový AD převodník s 1 vzorkem @ 312.5 MHz (M = 1, S = 1, N' = 16, FC = 312.5 MHz)

L = 1 -> rychlost diferenciálního páru na lince = 6250 Mbps

L = 2 -> rychlost diferenciálního páru na lince = 3125 Mbps

Příklad 2 : Dva 12 - bitové AD převodníky s dvěma vzorky za hodinový rámec (I / Q) @ 625 MHz (M = 2, S = 2, fs = 625 MHz)

N = 12, L = 4 -> rychlost diferenciálního páru na lince = 9375 Mbps

N = 16, L = 4 -> rychlost diferenciálního páru na lince = 12500 Mbps

N = 12, L = 8 -> rychlost diferenciálního páru na lince = 4687.5 Mbps

N = 16, L = 8 -> rychlost diferenciálního páru na lince = 6250 Mbps

6. Kolik je přenášeno oktet za rámeček (F)?

Vypočítáno pomocí následující rovnice:

$$F = (M \times S \times N') / (8 \times L) \quad (8.1.6) [3]$$

$$F = (4 \times 1 \times 16) / (8 \times 4) = 2 \quad (8.1.7) [3]$$

$$\mathbf{F = 2}$$

7. Je použit mód Vysoké Hustoty (HD)?

Jestliže jsou vzorky rozděleny mezi diferenciální páry, poté $HD = 1$

$$\mathbf{HD = 1}$$

8.2 Vztahy hodinových frekvencí

Jsou-li parametry linky nastaveny, je potřeba dále stanovit určité vztahy frekvencí

- Rychlost linky (Bitová rychlost) – efektivní rychlost dat na sériové lince

Vypočítáno u kroku 5

- Znakové hodiny

Vypočítáme pomocí: Rychlost linky / 10

- Hodinový rámeček

Vypočítáno u kroků 3 a 4

- Multirámčové hodiny (MFC)

Vypočítáme pomocí: Hodinový rámeček / K

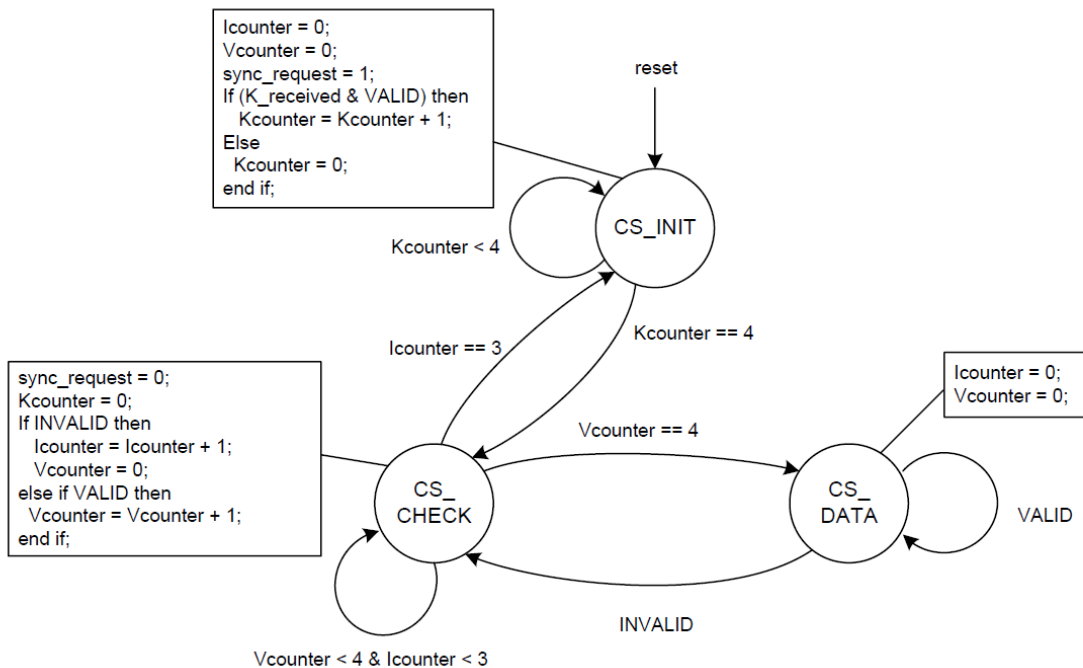
Kde K je celé číslo rámečků.

[3]

8.3 Implementované moduly a ověření funkčnosti

8.3.1 CGS

Jakmile vysílač zachytí změnu signálu **sync_request** (~SYNC), začne poté vysílat řídicí symboly do přijímače. Začíná tedy proces zvaný synchronizace kódových skupin (Code Group Synchronization). Jak již víme, CGS končí, když přijdou po sobě čtyři K28.5 znaky. Pro tento proces byl vytvořen stavový automat s třemi definovanými stavy. Na **Obrázek 17** je přiblížena sekvence stavového automatu.

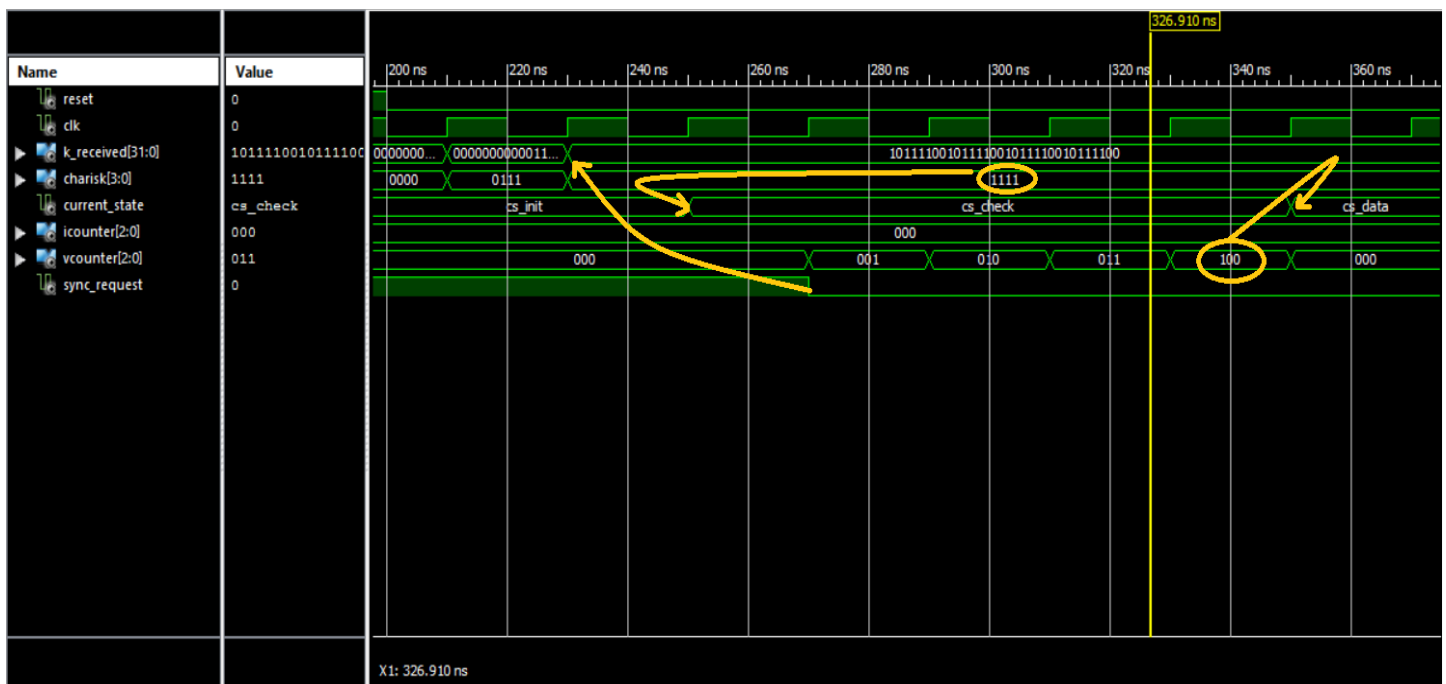


Obrázek 17. Stavový automat pro CGS [12]

Na **Obrázek 18** je ověřena funkčnost modulu pro CGS. Nejdříve vysvětlím funkčnost čítačů **vcounter** a **icounter**. **Vcounter** je čítač, který čítá K28.5 znaky v našem příchozím datovém toku. Jestliže tento čítač dopočítá do 4, vynuluje se a přepne stav do **cs_data**. **Icounter** počítá jiné řídicí znaky od K28.5 (/F/,/A/,/R/, atd.) a jestliže přijímač zaznamená jiný řídicí znak, než K28.5, zvýší čítač **icounter** o hodnotu 1 a vynuluje čítač **vcounter**. Jestliže **icounter** na počítá do 3, vrátí stav do **cs_init**.

Current_state nám dává informaci, v jakém stavu se nacházíme. (**cs_init**,**cs_check**,**cs_data**). Vstupy tohoto modulu jsou: **reset**, hodinový signál **clk**, příchozí tok dat **k_received** a vstup **charisk**, který značí, jestli v aktuální okteti je řídicí znak. Tento vstup koresponduje přímo s datovým tokem, a jestliže zaznamená řídicí znak, tak nastaví příslušný bit na 1. Bity přímo korespondují s oktety ze vstupu dat. Tedy první bit (zleva), ze vstupu **charisk**, koresponduje s první osmicí bitů z příchozího toku dat (zleva).

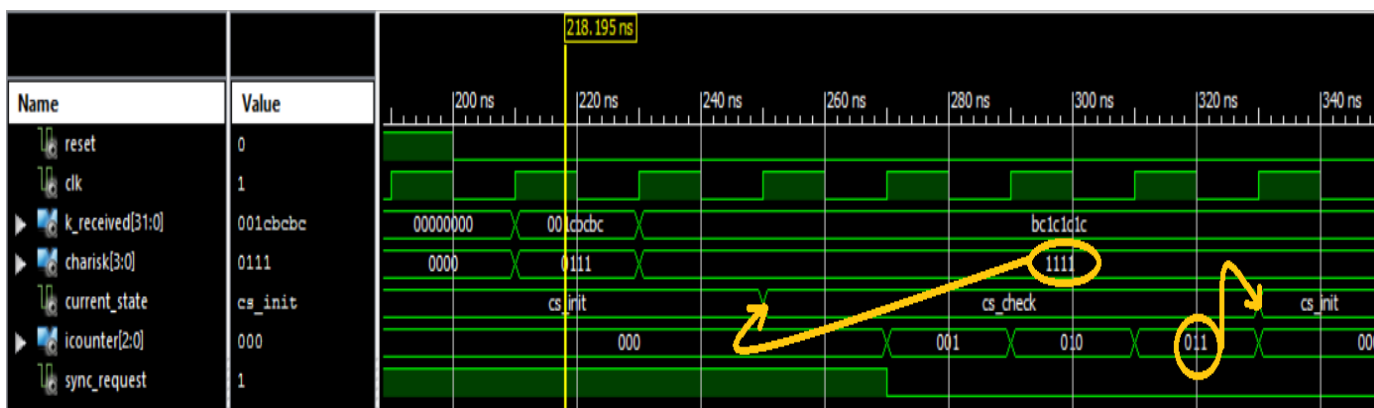
Výstupem tohoto modulu je **sync_request** který, když je aktivní, vysílá řídicí znaky do datového toku.



Obrázek 18. Ověření funkčnosti CGS modulu

Proces na **Obrázek 18** je krátce po resetu a nacházíme se ve stavu **cs_init**. Z prvního příchozího datového toku nám vstup **charisk** dává informaci o tom, že v datovém toku jsou 3 řídicí znaky, tedy není splněna podmínka pro přechod do stavu **cs_check**. V dalším příchozím toku už je podmínka pro přechod do stavu **cs_check** splněna, kdy nám vstup **charisk** označil, že v datovém toku jsou 4 řídicí znaky. Jakmile se **current_state** do tohoto stavu, nastaví se **sync_request** na 0, to znamená, že přestává vysílat řídicí znaky. **Vcounter** zvýší svou hodnotu o 1 pokaždé, kdy zaznamená v datovém toku K28.5 znak, tedy hodnotu 10111100. Jak je na obrázku zřejmé, v datovém toku jsou takové znaky 4, je teda splněna podmínka pro přechod do stavu **cs_data**. Pokud by přijímač zaznamenal v příchozím toku nový řídicí znak (**sync_request** by musel být aktivní), vrátil by **current_state** do stavu **cs_check**.

Na **Obrázek 19** je demonstrována CGS funkce stavového automatu při třech příchozích řídicích znacích od znaku K28.5. Podmínku přechodu do stavu **cs_check** splňuje hodnota vstupu **charisk** 1111, tedy příchozí 4 řídicí znaky. Hexadecimální hodnota znaku K28.5 je hodnota bc a hexadecimální hodnota 1c je tedy hodnota znaku odlišného od znaku K28.5. **Icounter** správně napočítal 3 odlišné řídicí znaky od K28.5, což splňuje podmínku pro přechod do stavu **cs_init**.

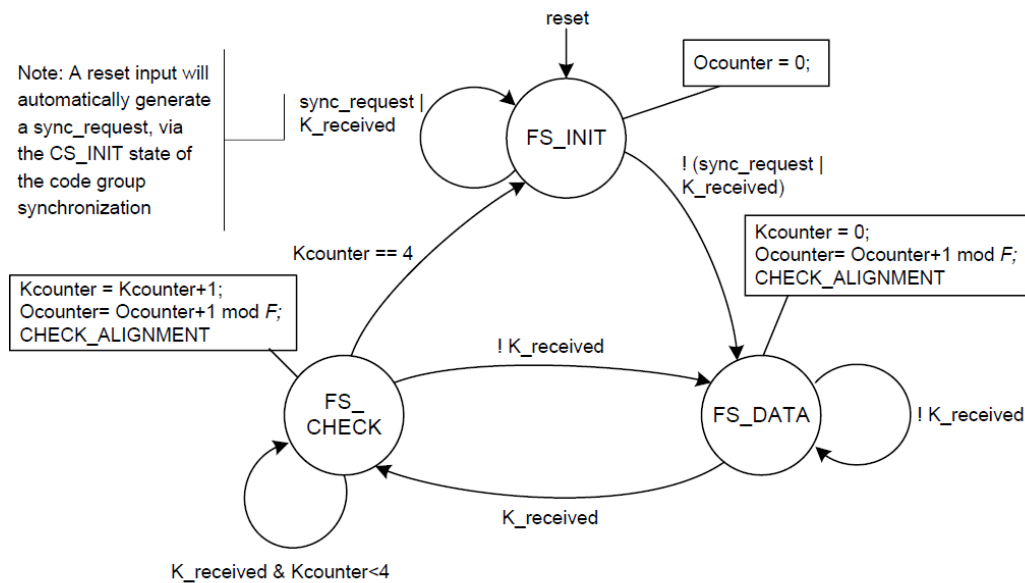


Obrázek 19. Ověření funkce modulu CGS

8.3.2 Synchronizace počátečního rámce

Synchronizace počátečního rámce je proces řešený pomocí stavového automatu. CGS aktivuje signál **sync_request**, který drží proces synchronizace rámce ve svém počátečním stavu po celou dobu CGS. Po dokončení CGS se deaktivuje signál **sync_request** a vysílač přestává posílat /K/ symboly. Tím **current_state** změní stav na **FS_DATA**.

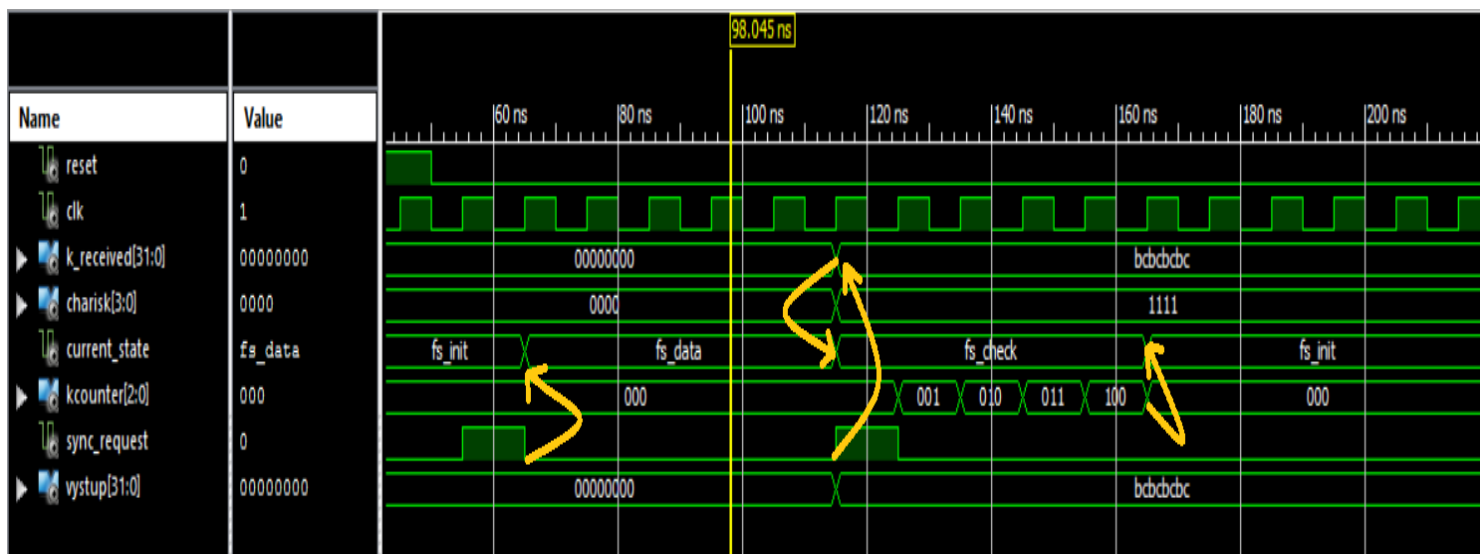
Oktetový čítač (Ocounter) počítá pozici přijímaných oktet v rámci mezi 0 a F-1. Jestliže přijímač vyšle informaci o tom, že není splněná CGS, nastaví se **sync_request** na aktivní úroveň a vyšle /K/ symboly. Jestli obdržíme příslušný K28.5 znak, **current_state** změní stav na **FS_CHECK**, ve kterém oktetový čítač stále počítá. Jestliže obdržíme čtyři po sobě jdoucí K28.5 znaky, **current_state** změní stav na prvotní stav, kterým je **FS_INIT**.



Obrázek 20. Stavový automat pro synchronizaci počátečního rámce [12]

Na **Obrázek 21** je ověřena funkce modulu synchronizace počátečního rámce.

Po úspěšném dokončení CGS se deaktivuje signál **sync_request** na hodnotu 0, čímž je splněna podmínka pro přechod ze stavu **fs_init** do stavu **fs_data**. Nastala situace, kdy přijímač řádně nedokončil CGS. Aktivuje se tedy signál **sync_request** a vysílají se další /K/ symboly. Po úspěšném přijetí K28.5 znaku je splněna podmínka pro změnu stavu **current_state** na stav **fs_check**. Čítač **Kcounter** počítá příchozí K28.5 znaky (hexadecimální hodnota bc). Jakmile zaznamená 4 příchozí K28.5 znaky je splněna podmínka pro přechod **current_state** do původního stavu, čímž je **fs_init**.



Obrázek 21. Ověření funkce modulu synchronizace počátečního rámce

8.3.3 Znakové nahrazení

Nahrazení příslušných oktét probíhá podle pravidel pro rámcové zarovnání a pro pravidla zarovnání diferenciálního páru. Jestliže vysílač vyšle řídicí znak /F/ (do vstupního toku dat), přijímač má za úkol nahradit tento znak hodnotou poslední oktety z předchozího rámce.

To stejné se provede, jestliže vysílač vyšle řídicí znak /A/. Již víme, že řídicí znak /F/ značí konec rámce a řídicí znak /A/ značí konec multirámce. Rámec tvoří 2 oktety a multirámec tvoří 8 rámců (pro náš převodník). Je tedy jisté, že řídicí znak /F/ může být jen v druhé oktetě z příchozího rámce a řídicí znak /A/ jen v druhé oktetě posledního (osmého) rámce z multirámce. Jestliže jsou tyto řídicí znaky na jiné pozici, než se předpokládá, přijímač to vyhodnotí jako chybu a je vhodné proces resetovat. V tomto modulu používáme několik pomocných signálů. Jmenovitě jde o tyto signály:

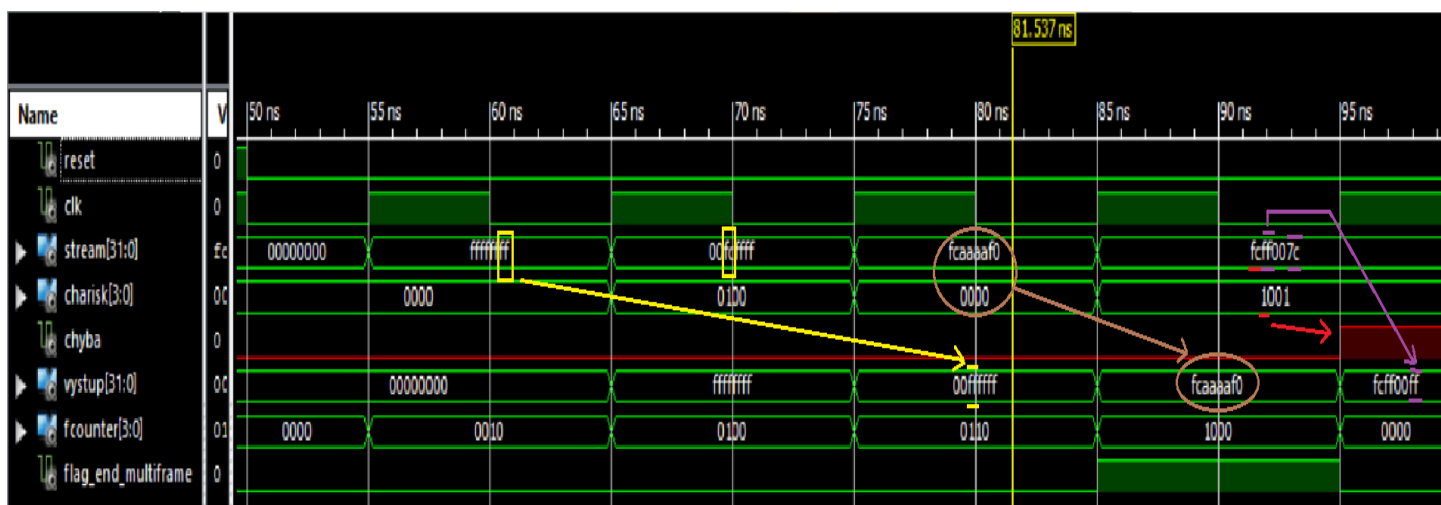
Flag_end_multiframe – je pomocný signál, který nám značí, že příslušný rámec je posledním (osmým) v celém multirámci.

previous_F – slouží k zapamatování poslední oktety z předchozího toku dat.

Fcounter – je čítač, který počítá příchozí rámce

Vstupy tohoto modulu jsou: **reset**, hodinový signál **clk**, příchozí tok dat **stream** a vstup **charisk**, který značí, jestli v příchozí oktetě je daný řídicí znak (v tomto případě řídicí znak /F/ či /A/). Tento vstup koresponduje přímo se vstupním tokem, a jestliže zaznamená řídicí znak, tak nastaví příslušný bit na 1. Bity přímo korespondují s oktety ze vstupu dat. Tedy první bit (zleva) ze vstupu **charisk** koresponduje s prvními dvěma hexadecimálními znaky (zleva) z příchozího toku dat, které zaplní první oktetu.

Výstupem tohoto modulu je zarovnaný tok dat dle pravidel pro znakové nahrazení a **Chyba**, která přijímač upozorní, že jeden ze symbolů je na nečekané pozici.



Obrázek 22. Ověření funkčnosti modulu znakového nahrazení

Ověření funkce implementovaného modulu je popsána na **Obrázek 22**.

Proces začíná krátce po resetu, kdy na první náběžnou hranu hodinového signálu (**clk**) přijímač přijme první příchozí tok dat (**stream**). Tok dat je vždy 32-bitový a obsahuje tedy 2 rámece. Vstup **charisk** nezaznamenal žádný řídicí znak (ani /F/ ani /A/) a zjistil, že příchozí tok dat obsahuje jen platná data (v tomto případě data ffffffff), které pošle na další náběžnou hranu na výstup. S touto náběžnou hranou přijímač současně přijmul další 32-bitový tok dat (00fcffff). Tento tok dat obsahuje hexadecimální hodnotu fc, který souhlasí s hodnotou symbolu /F/. Vstup **charisk** tento znak zaznamenal a nastavil příslušný bit na 1. Jeho hodnota je tedy 0100. Řídicí znak /F/ je na správné pozici (druhá okteta z rámce) a zarovná rámeček. Tuto oktetu přijímač nahradí poslední (druhou) oktetou z předchozího rámce (ff). K tomu využijeme náš signál `previous_F`, který pamatuje poslední oktetu z předchozího toku dat. Na další náběžnou hranu přijímač pošle na výstup náš tok dat s nahrazeným řídicím znakem /F/ - (00ffffff). Další datový tok je následující – fcaaaaf0 – vstup **charisk** vyhodnotil tento tok jako platné data (0000), tudíž nám dal informaci o tom, že tyto data může přijímač poslat na výstup. V první okteti současného toku můžeme vidět hexadecimální hodnotu fc, která je svou hodnotou stejná jako hodnota řídicího /F/, nicméně ji vstup **charisk** označil jako 0 (tedy platné data). Je tedy tato okteta společně s ostatními z celého toku poslána na výstup. Posledním datovým tokem na této ukázce je tok fcff007c. Vstup **charisk** nám dává informaci o tom, že zaznamenal řídicí znaky v první a poslední okteti tohoto toku, tedy 1001. Hexadecimální hodnoty (fc a 7c) těchto oktět korespondují s řídicím znakem /F/ a řídicím znakem /A/. Víme, že řídicí znak /F/ značí konec rámce, může proto být pouze v druhé okteti rámce. Přijímač tuto chybu vyhodnotil nastavením výstupu **chyba** na logickou hodnotu 1. Také již víme, že řídicí znak /A/ značí konec multirámce (osmý rámeček). Čítač **fcounter** již napočítal do osmi. Signál **Flag_end_multiframe** je tedy nastaven na logickou hodnotu a v poslední okteti tohoto toku můžeme tedy hledat hodnotu 7c. Řídicí znak /A/ je potom nahrazen druhou oktetou z předchozího rámce, tedy hodnotou ff a následně tento tok poslán na výstup. Pro bližší vysvětlení posledního případu. Daný tok fcff007c je tvořen ze 2 rámečků. První rámeček tvoří hexadecimální hodnota fcff (z toho fc je první okteta a ff je okteta druhá) a druhý tvoří hodnota 007c (00 je první okteta, 7c je okteta druhá).

8.4 Transportní vrstva

Jde o proces, kde z daných příchozích platných dat po diferenciálních párech, získáváme zpátky vzorky z našeho AD převodníku.

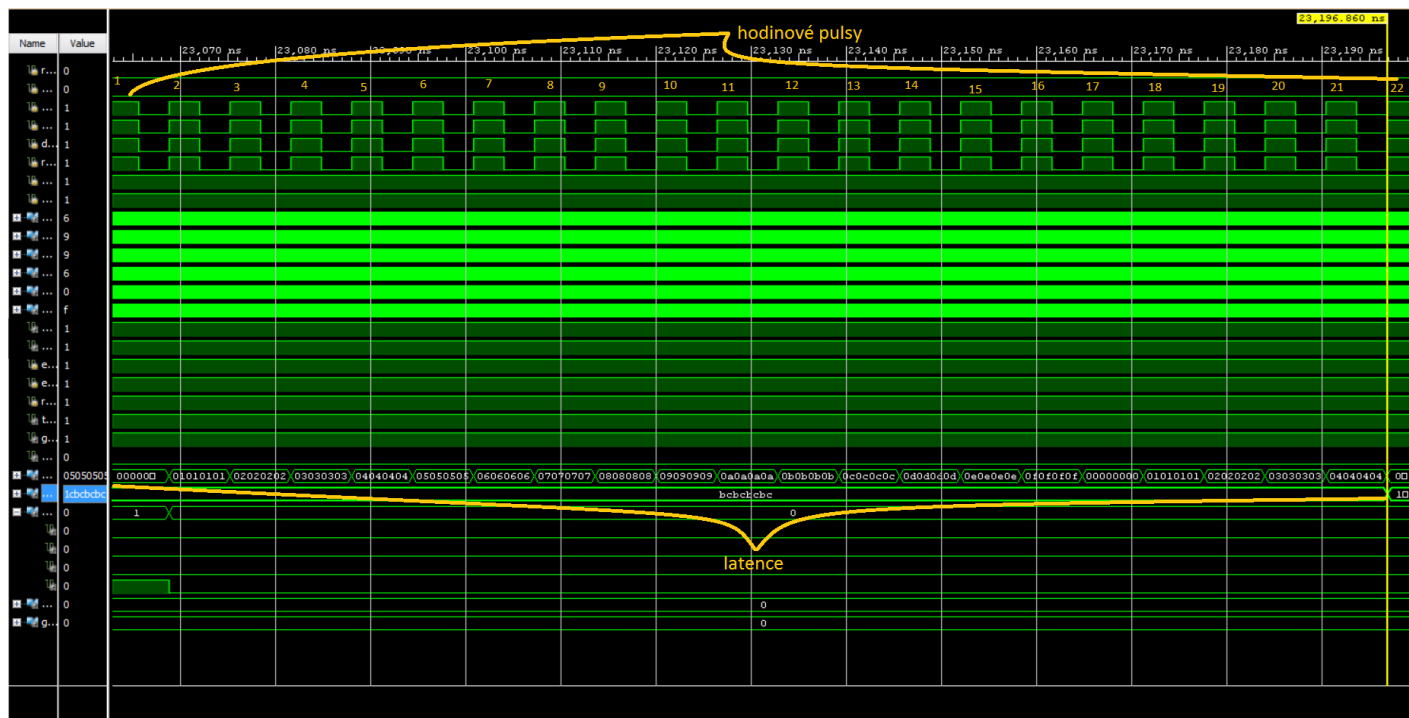
9 OVĚŘENÍ NA REÁLNÉM SYSTÉMU

Ověření funkčnosti implementovaných modulů bylo doporučeno vedoucím práce, panem inženýrem Soběslavem Valachem, provést pomocí simulací v prostředí ISE Project Navigator, ve kterém bylo vytvořeno hned několik test benchů, které zajišťují bezpečný chod zmíněné simulace. Tento test bench si uživatel naprogramuje sám podle potřeby a požadavků na simulaci. Test bench se píše v programovacím jazyce VHDL, stejně jako můj naprogramovaný firmware. Výsledky simulace jsou tytéž samé výsledky, které by byly získané pomocí ověření na reálném systému.

Navržené obvodové řešení neboli náš reálný systém, je klíčový element pro ověření funkčnosti na reálném systému. Nicméně, samotné obvodové řešení mi bylo předáno do ruky v době, kdy už jsme se s panem inženýrem Soběslavem Valachem dohodli, že bude vhodné, když výsledky ověřím na simulacích. Ověření funkčnosti na reálném systému je proces, který trvá minimálně 6 měsíců. Bylo by opět potřeba prostudovat obrovské množství materiálů jak pro PCI-E kartu, tak i dokumentaci k naprogramování našeho převodníku. Nicméně je to proces, ve kterém bych rád pokračoval v mém navazujícím studiu.

9.1 Zjištění latence IP coru

Na **Obrázek 23** jsou vyznačeny hodinové pulsy a celková latence IP coru.



Obrázek 23. Zjištění latence IP coru

U žluté čáry poukazující na latenci IP coru vidíme data z vysílače a hned pod nimi zpožděná data přijímače. Latence IP coru je zpoždění, za které se data z vysílače dostanou do přijímače. Jak je vidno ze simulace, latence IP coru činí 22 hodinových cyklů.

Latence AD převodníku je dlouhá 17 hodinových cyklů. Tato hodnota je zjištěna z dokumentace onoho převodníku.

10 ZÁVĚR

Obsahem této bakalářské práce je návrh obvodového a firmwarového řešení pro dané rozhraní. Vše bylo navrženo podle standardu JESD204B. Samostatný popis standardu a porovnání s předešlým standardem LVDS je vysvětlen v úvodních kapitolách této práce. Poté následuje návrh obvodového řešení a návrh samostatného firmwaru.

Dle navrženého obvodového řešení vznikl systém. Ověřil jsem celkovou latenci (zpoždění) vygenerovaného IP coru, která je dlouhá 22 hodinových cyklů. Latence AD převodníku je dlouhá 17 hodinových cyklů.

Součástí práce je vygenerované IP core od firmy Xilinx, které implementuje fyzickou vrstvu standardu JESD204B. Dále byly vytvořeny a implementovány moduly, které plní funkci datové a transportní vrstvy standardu JESD204B. Součástí práce je ověření jejich funkce.

JESD204 je v současné době stále vyvíjen. Stal se z něj plnohodnotný standard, který plní novější potřeby rozhraní. Veškerá vylepšení se zaměřují na to, aby sériová linka byla více rozšiřitelná.

Začít se podrobněji zabývat programovatelnými hradlovými poli označuji za vysoký přínos, nejen pro rozšíření mé bakalářské práce, ale také pro mé budoucí magisterské studium, kde již získané znalosti o VHDL jistě využiji.

Jako vylepšení do budoucna by bylo vhodné implementovat paralelní scrambler. S výhledem na diplomovou práci je možné rozšířit zadání na zhotovení vlastního softwarového rádia, pomocí právě standardu JESD204B.

LITERATURA

[1] ZARR, Richard. JESD204B vs. LVDS in high speed data converters. In: *Texas Instruments* [online]. 2013 [cit. 2016-05-05].

Dostupné z: http://e2e.ti.com/blogs_/b/analogwire/archive/2013/09/20/jesd204b-vs-lvds-in-high-speed-data-converters

[2] ANALOG DEVICES. JESD204B Survival Guide [online]., 1-78 [cit. 2016-01-07]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/technical-articles/JESD204B-Survival-Guide.pdf>

[3] ANALOG DEVICES: Demystifying the JESD204B High-speed Data Converter-to-FPGA interface [online]. 2013 [cit. 2016-05-05].

Dostupné z: <http://www.slideshare.net/AnalogDevicesInc/jesd204-april2012>

[4] TEXAS INSTRUMENTS: ADC16DX370 Dual 16-Bit 370 MSPS ADC With 7.4 Gb/s JESD204B Outputs [online]. In: . 2014 [cit. 2016-05-12]. Dostupné z: <http://www.ti.com/lit/ds/symlink/adc16dx370.pdf>

[5] JONATHAN HARRIS. Understanding layers in the JESD204B Specification [online]. 2014, 6 [cit. 2016-02-19]. Dostupné z: http://www.analog.com/media/en/technical-documentation/technical-articles/Understanding-the-Layers-in-the-JESD204B-Specification-A_Hide-Speed-ADC-Perspective-Part-1-MS-2714.pdf

[6] KUBÍČEK, Michal. MODELOVÁNÍ A IMPLEMENTACE SUBSYSTÉMŮ KOMUNIKAČNÍHO ŘETĚZCE V OBVODECH FPGA [online]. Brno, 2009 [cit. 2016-05-05].

Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=19776. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Prof. Dr. Ing. Zdeněk Kolka.

[7] ŠŤASTNÝ, Jakub. FPGA prakticky: realizace číslicových systémů pro programovatelná hradlová pole. Praha: BEN - technická literatura, 2010. ISBN 978-80-7300-261-9.

[8] Nebojte se FPGA [online]. 2002 [cit. 2016-01-02]. Dostupné z: <http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/nebojte-se-fpga.html>

[9] PROGRAMMABLE DEVICES: FPGA. *Xilinx: All Programmable* [online]. 2015 [cit. 2016-05-05]. Dostupné z: <http://www.xilinx.com/products/silicon-devices/fpga.html>

[10] BHATIA, Sagar, Tania BAJAJ a Shikha SHARMA. FIELD PROGRAMMABLE GATE ARRAY- A BRIEF STUDY [online]. In: Gurgaon: Dronacharya College of Engineering, 2015 [cit. 2016-05-05]. ISSN 2349-6002. Dostupné z: http://www.ijirt.org/vol1/paperpublished/IJIRT102140_PAPER.pdf

[11] PINKER, Jiří a Martin POUPA. Číslicové systémy a jazyk VHDL. Praha: BEN - technická literatura, 2006. ISBN 80-7300-198-5

[12] JEDEC: SERIAL INTERFACE FOR DATA CONVERTERS [online]. In: 2012 [cit. 2016-05-12]. Dostupné z: <https://www.jedec.org/standards-documents/docs/jesd-204a>

SEZNAM OBRÁZKŮ

Obrázek 1. Porovnání obvodového řešení standardu LVDS se standardem JESD204B [1]	10
Obrázek 2. Originální standard JESD204 [2]	13
Obrázek 3. První revize - JESD204A [2]	14
Obrázek 4. Druhá (Aktuální) revize JESD204B [2]	15
Obrázek 5. Ukázka JESD204B rozhraní linky - inicializovaný časový diagram [4] ...	18
Obrázek 6. Zjednodušené schéma toku dat v JESD204 systémech [5]	20
Obrázek 7. Mapování konverzních vzorků do sériových diferenciálních párů [5]	22
Obrázek 8. Datová vrstva, CGS, ILAS, Datová sekvence [5]	26
Obrázek 9. Blokové schéma scrambleru [5]	29
Obrázek 10. Blokové schéma descrambleru [5]	29
Obrázek 11. Typická struktura obvodu FPGA [8]	37
Obrázek 12. Struktura logického bloku [8]	38
Obrázek 13. Úspěšné přeložení designu	45
Obrázek 14. Schéma obvodového řešení převodníku ADC34J25	46

Obrázek 15. Kompletní obvodové řešení – horní strana desky	47
Obrázek 16. Kompletní obvodové řešení - spodní strana desky	48
Obrázek 17. Stavový automat pro CGS [12].....	53
Obrázek 18. Ověření funkčnosti CGS modulu.....	54
Obrázek 19. Ověření funkce modulu CGS.....	55
Obrázek 20. Stavový automat pro synchronizaci počátečního rámce [12]	56
Obrázek 21. Ověření funkce modulu synchronizace počátečního rámce.....	56
Obrázek 22. Ověření funkčnosti modulu znakového nahrazení.....	57
Obrázek 23. Zjištění latence IP coru	59

SEZNAM TABULEK

Tabulka 1. Srovnání LVDS s JESD204B [1]	10
Tabulka 2. Tabulka řídicích znaků JESD204 [5]	27
Tabulka 3. JESD204B specifikace fyzické vrstvy [5].....	28

SEZNAM PŘÍLOH

Příloha 1 : DVD

- Vivado – navržený firmware
- ISE Project Navigator – navržený firmware + test benche k simulacím + samotné simulace
- Dokumentace – bakalářská práce v elektronické podobě
- Foto – foto výsledku práce
- Schéma navrženého obvodového řešení