

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

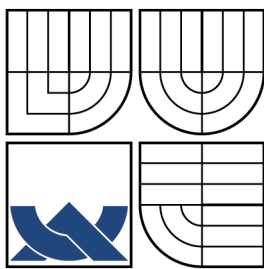
IMPLEMENTACE OFDM V OBVODU FPGA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

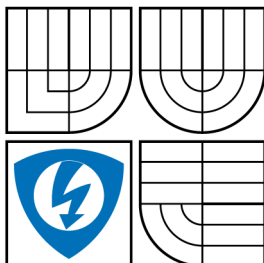
AUTOR PRÁCE
AUTHOR

Bc. Martin HORÁK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

IMPLEMENTACE OFDM V OBVODU FPGA IMPLEMENTATION OFDM INTO FPGA CIRCUIT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Martin HORÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Roman Maršálek, Ph.D.

BRNO 2007

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Martin HORÁK
Bytem: A. Provazníka 1340, Rychnov nad Kněžnou, 516 01
Narozen/a (datum a místo): 5.11. 1983 v Rychnově n. Kn.

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- .. disertační práce
 - ☒ diplomová práce
 - .. bakalářská práce
 - .. jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Implementace OFDM v obvodu FPGA

Vedoucí/ školitel VŠKP: Ing. Roman Maršálek, PhD.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- ☒ v tištěné formě – počet exemplářů: 2
- ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 30. května 2008

.....
Nabyvatel

.....
Autor

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace OFDM v obvodu FPGA“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....
(podpis autora)

PODĚKOVÁNÍ

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

V Brně dne

.....
(podpis autora)

ABSTRAKT

Diplomová práce je zaměřena na implementaci OFDM modemu do programovatelného obvodu FPGA. Výhody využití OFDM jsou zřejmé, vysoká spektrální účinnost a robustnost vůči mnohacestnému šíření. Základní principy OFDM signálů jsou proto zmíněny na několika prvních stranách. V tomto případě byla využita konstrukce OFDM signálu pomocí inverzní Fourierovy transformace s vnitřní modulací nosných 256-QAM pro zvýšení kapacity kanálu. Jelikož je algoritmus výpočtu diskrétní Fourierovy transformace poměrně složitý, je využit efektivní algoritmus IFFT a jeho algoritmem využitým v obvodech DSP se také budeme zabývat. Dále jsou analyzovány možné problémy spojené s realizací takto zvoleného modemu a poskytnuta vhodná řešení pro jejich eliminaci. Před započítím implementace návrhu do vlastního obvodu je však nutno provést nezbytné teoretické simulace v programu Matlab. Analyzovaný OFDM signál je podroben statistické i spektrální analýze. Několik stran je dále věnováno samotnému HW, pro který je návrh určen a dále pokračuje kapitola věnovaná simulacím v obvodovém simulátoru ModelSim. Pro vlastní programování v HDL kódu je použito programu ISE Xilinx. V závěru práce jsou pak uvedeny praktická měření potvrzující funkčnost daného modemu. Všechny skripty a data použité v rámci práce jsou distribuovány na příloženém médiu.

KLÍČOVÁ SLOVA

modem, OFDM, FPGA, VHDL, Virtex

ABSTRACT

The thesis is focused on designing OFDM modem, which should be implemented into the FPGA device. The advantages of using OFDM signals in order to provide high baud rates together with high multipath immunity has provoked a mass expansion into media systems such as DSL, DVB, Wi-Fi, WLAN, etc.. Thanks to this technology we can guarantee high modulation rates with minimal negative disturbance effects. The first part is dedicated to characterise OFDM signals, their generation and the algorithm producing the OFDM which is implemented in DSP devices. For the purpose of using the fastest algorithm, the Fast Fourier Transform using Cooley-Tukey algorithm was shown. Before we can implement OFDM modem into the FPGA device, we have to simulate its correct function. Because there is no OFDM analyser available at our department, it's necessary to prove its correct function by simulations. Therefore a large part of this thesis is focused on simulations using Matlab and ModelSim, in order to show comparison between the theoretical, and simulated results. Between the theoretical and practical simulations there is a part which shows the brief characteristics of available FPGA devices. Detailed view is presented just for the Virtex II device, which the implementation is made for. As a suitable FPGA device, we have chosen Virtex II XC2V1000 which is available for students. In the last part the measured results were shown to prove the correct function of the modem. Programming the FPGA using VHDL language is realized in the software ISE Xilinx (distributed in Xilinx software support). All programmed scripts and data used in this thesis are included on distributed media.

KEYWORDS

modem, OFDM, FPGA, VHDL, Virtex

OBSAH

1	Úvod	10
1.1	OFDM	10
1.2	FPGA	10
1.3	VHLD	10
2	OFDM	11
2.1	Stručná charakteristika OFDM	11
2.2	Ortogonalita v časové oblasti	12
2.3	Ortogonalita ve frekvenční oblasti	13
2.4	Ortogonalita pomocí DFT	15
2.5	Algoritmus výpočtu FFT	15
2.5.1	Cooley-Tukey	16
2.5.2	Decimace v časové oblasti (DIT)	16
2.5.3	Decimace ve frekvenční oblasti (DIF)	17
2.6	Problémy spojené s aplikací OFDM	19
2.6.1	Frekvenční odchylka oscilátoru	19
2.6.2	Fázová odchylka oscilátoru	19
2.6.3	Nepřesná pozice FFT okna	19
2.6.4	Nepřesná vzorkovací frekvence	20
2.7	COFDM	20
3	Formáty vyjádření binárních čísel	21
3.1	Integer fixed-point representation	21
3.1.1	Offset binary	21
3.1.2	Two's complement	21
3.2	Fraction fixed-point representation	22
4	Simulace OFDM modemu - Matlab	23
4.1	Statistická simulace OFDM modemu	23
4.2	Simulace OFDM modemu ve spektrální oblasti	25
5	Obvody FPGA	28
5.1	Řada Virtex II TM	28
5.2	Řada Virtex 4 TM	28
6	Virtex II XC2V1000	29
6.1	V2MB1000	29
6.2	P160	29
6.3	Implementace	29
6.3.1	AD převodník	30
6.3.2	DA převodník	30
7	Simulace OFDM modemu - implementace	34
7.1	Modulátor 256-QAM	34
7.2	Blok IFFT	36
7.2.1	Knihovna IFFT	36
7.2.2	Simulace bloku IFFT	37
7.3	RF modulátor	39
7.3.1	DDS	39
7.3.2	Násobička	39
7.4	OFDM modem - celková simulace	40
7.5	Celková simulace spektra	43
7.5.1	Simulované spektrum	43

8 Praktické měření	45
8.1 Změřená spektra	45
8.2 Tréninková sekvence	47
8.3 Shrnutí	47
9 Závěr	48
Literatura	49
A Přílohy	50
A.1 Závěrečná zpráva z programátoru (Design summary)	50
A.2 Simulace ModelSim - časové průběhy bloků	51
A.3 Simulace ModelSim - časové průběhy bloků 2	52
A.4 Simulace ModelSim - časové průběhy bloků 3	53

SEZNAM OBRÁZKŮ

2.1	Princip digitální OFDM	11
2.2	Analogový modulátor OFDM	11
2.3	Příklad vzájemně ortogonálních funkcí v časové oblasti	14
2.4	Blokové schéma OFDM systému	16
2.5	Zjednodušená struktura algoritmu FFT metodou DIT	17
2.6	Zjednodušená struktura algoritmu FFT metodou DIF	18
2.7	Úplný signálový diagram 8-bodové FFT metodou DIT	18
4.1	Principiální schéma OFDM modemu	23
4.2	Časový průběh signálu 256-QAM	23
4.3	Konstelační diagram signálu 256-QAM (Grayovo kódování)	24
4.4	Charakteristiky výstupního signálu po IFFT - část 1	24
4.5	Charakteristiky výstupního signálu po IFFT - část 2	25
4.6	Použité blokové schéma	25
4.7	Vkládání nul na subnosné vlny	26
4.8	Teoretické spektrum OFDM modemu - část 1	26
4.9	Teoretické spektrum OFDM modemu - část 2	27
6.1	Virtex 2 TM Development board - V2MB1000	29
6.2	Analogový modul P160	30
6.3	Blokové schéma vstupní části P160	31
6.4	Blokové schéma výstupní části P160	32
6.5	Přiřazení pinů XC2V1000 FG456	33
7.1	Funční blokové schéma OFDM modemu	34
7.2	Blokové schéma - QAM	34
7.3	Konstelační diagram 256-QAM	35
7.4	Blokové schéma - IFFT	36
7.5	Nastavení jádra Fast Fourier Transform v3.2	37
7.6	Blokové schéma - RF modulátor	39
7.7	Nastavení jádra DDS v5.0	39
7.8	Blokové schéma - kompletní OFDM modem	40
7.9	Simulace spektra OFDM modemu - část 1	43
7.10	Simulace spektra OFDM modemu - část 2	44
8.1	Spektrum signálu $\cos(x)$ generovaného blokem DDS	45
8.2	Spektrum signálu $I \cdot \cos(x)$	45
8.3	Spektrum signálu $Q \cdot \sin(x)$	46
8.4	Spektrum výstupního signálu $I \cdot \cos(x) - Q \cdot \sin(x)$	46
8.5	Popis výstupních signálů (viz. přílohy)	47

1 ÚVOD

V této práci je proveden softwarový návrh OFDM modemu určeného pro různé aplikace. Implementace je určena pro vývojové desky firmy Xilinx typového označení **Virtex-IITM V2MB1000**. Zdrojový kód v jazyce VHDL však může být překompilován a využit pro jiné vývojové desky obsahující čip Xilinx FPGA. V současnosti se různé varianty OFDM používají v bezdrátových i kabelových spojích digitálních datových systémů. Vznik této nové digitální technologie OFDM, umožnil rozvoj rychlých signálových procesorů DSP. Metodu OFDM v řadě různých podob dnes používá řada komunikačních systémů např. xDSL, IEEE 802.11a, WLAN, DAB(Digital Audio Broadcasting), DVB(Digital Audio Broad), mobilní sítě 3G, flash OFDM apod.. Výsledný OFDM modem tak má za účel zjednodušenou formou ukázat charakteristické vlastnosti těchto systémů založených na OFDM. Parametry námi navrženého systému budou uvedeny dále v kap. 4.

1.1 OFDM

(Orthogonal Frequency Division Multiplex)

Tato metoda multiplexování je založena na technice MCM (Multi Carrier Modulation). U systémů využívajících FDM je celé přenosové pásmo rozděleno na velké množství nezávislých kanálů definovaných hodnotou subnosné frekvence. Jednotlivé kanály jsou nejprve omezeny pomocí anti-aliasingových filtrů typu DP a následně namodulovány na patřičnou subnosnou frekvenci danou kmitočtovým syntezátorem. Vzdálenost kmitočtů těchto kanálů je určena kmitočtovým krokem syntezátoru. Od systému FDM se však OFDM liší použitím ortogonálních subnosných vln, nahrazujících nutnost použití frekvenčních rozestupů mezi subkanály vhodným algoritmem tvorby subnosných vln. Tyto subnosné vlny jsou poté vzájemně ortogonální a mohou se tak vhodně překrývat. Hlavní přínos této techniky spočívá v prodloužení symbolové periody a eliminaci ISI. Princip OFDM byl patentován Bellovými laboratořemi již v roce 1966, ale trvalo dalších 5 let, než došlo k využití efektivního algoritmu FFT při tvorbě OFDM symbolu. Další informace v kap. 2.

1.2 FPGA

(Field Programmable Gate Arrays)

Obvody FPGA se v současnosti staly klíčovými komponenty při implementaci výkonných systémů digitálního zpracování (DSP) v odvětví komunikačních technologií, sítí, videa a zpracování obrazu. Jejich výkonnost dovoluje realizovat složité matematické operace jako jsou digitální filtrace, FFT apod. bez potřeby dalšího hardwaru. Další podstatnou výhodou je jejich snadná rekonfigurovatelnost pomocí PC rozhraní. Podrobné informace se nachází v kap. 5.

1.3 VHDL

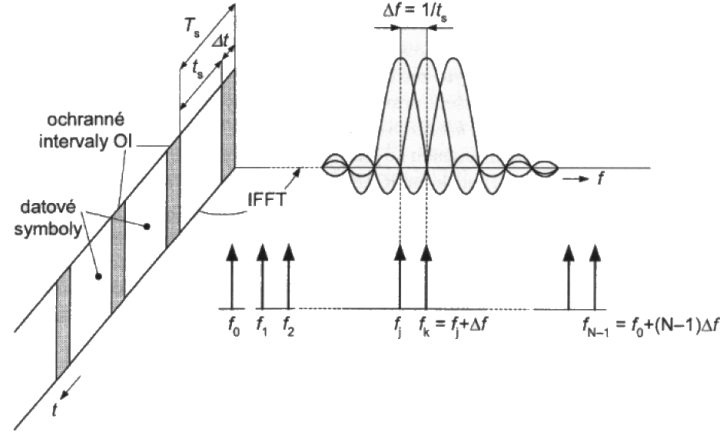
(Very high speed integrated circuits Hardware Description Language)

Jazyk VHDL je jazykem vysoké úrovně navržený pro účely návrhu a simulace velmi rozsáhlých číslicových systémů. Výhodami tohoto jazyka jsou bohaté vyjadřovací schopnosti a značná nezávislost číslicového systému popsaného pomocí VHDL na cílové technologii jeho implementace. „VHDL je pouze programovací jazyk, uživatel si zvolí sám obvod aplikace a metodiku řešení“ [16]. Zkonstruovaný návrh systému ve VHDL je syntetizován a výsledkem je zapojení hradel (klopných obvodů) určené pro daný obvod. Výjimku tvoří soubor testovacích vektorů *testbench* a řada dalších VHDL příkazů, které jsou využity pouze pro simulaci, nikoli však pro syntézu do cílového hardwaru.

Mezinárodní normu jazyka VHDL najdeme pod označením **IEEE 1076** v opravené verzi z roku 1993 [6]. Poslední novelizace byla vydána roku 2004. Syntaxe VHDL je plně definována v manuálu LRM (*Language Reference Manual*) (viz. IEEE Std 1076 část 1-1), čímž se jazyk odlišuje od ostatních hardwarových programovacích jazyků, které jsou až na výjimky přizpůsobeny chování obvodu a postrádají tak univerzálnost aplikace.

2 OFDM

2.1 Stručná charakteristika OFDM



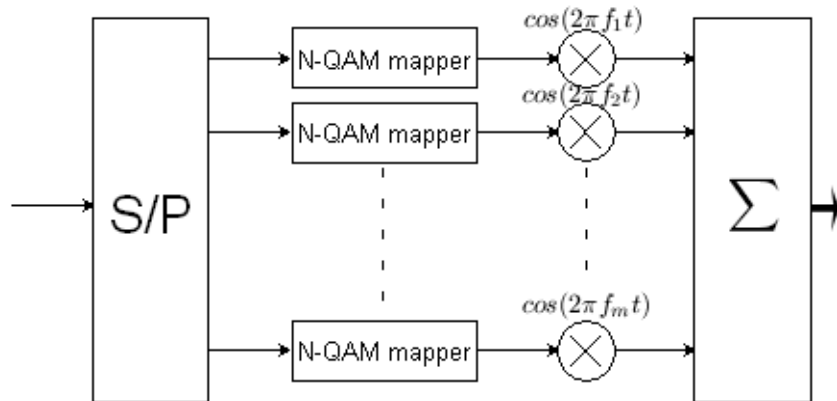
Obr. 2.1: Princip digitální OFDM - časová i frekvenční reprezentace [11]

Matematický popis OFDM signálu [1] :

$$s(t) = \sum_{n=-\infty}^{\infty} \sum_{m=0}^{M-1} s_{m,n} \text{Rect}_{T_s}(t - nT) e^{j \frac{2\pi m}{T} t}, \quad (2.1)$$

kde s je n -tý symbol vyslaný na m -té subnosné vlně popsané kmitočtem $\omega_m = \frac{2\pi m}{T}$ a Rect_T vyjadřuje pravoúhlé okno s trváním T určujícím periodu OFDM symbolu.

Pro další ilustraci tvorby OFDM signálu je na obr. 2.2 znázorněno blokové schéma analogového modulátoru OFDM.



Obr. 2.2: Analogový modulátor OFDM

Výhody OFDM :

- Vysoká spektrální účinnost i bez použití strmých filtů vysokého řádu.
- Velká odolnost vůči mnohacestnému šíření signálu (Multipath propagation)
- Velká odolnost vůči interferencím typu ISI (Inter-Symbol Interference)
- Velká odolnost vůči interferencím typu ICI (Inter-Channel Interference)
- Případné interference postihují jen malé úseky dlouhých symbolů
- Velká odolnost vůči úzkopásmovému rušení
- Vysoce efektivní implementace FFT
- Podpora SFN (Single Frequency Networks)

Nedostatky OFDM :

- Citlivost na offset subnosných vln (Dopplerův jev)
- Citlivost na frekvenční synchronizaci
- Nekonstantní obálka signálu, vysoké PAPR a nutné lineární zesilovače

2.2 Ortogonalita v časové oblasti

Vycházíme z předpokladu existence M bodové reprezentace vstupního signálu v konstalačním diagramu. Poté těchto M bodů můžeme zakódovat pomocí k bitů, kde $M = 2^k$. Vzájemnou *ortogonalitou* mezi těmito body vytváříme *M -dimenzionální prostor bázových funkcí*, které tvoří lineárně nezávislé vektory (viz. Gramm-Schmidtův ortonormalizační proces)[1] :

$$\begin{aligned} v_1 &= (\sqrt{\epsilon_s}, 0, 0, \dots, 0) \\ v_2 &= (0, \sqrt{\epsilon_s}, 0, \dots, 0) \\ v_3 &= (0, 0, \sqrt{\epsilon_s}, \dots, 0) \\ &\vdots \\ v_M &= (0, 0, 0, \dots, \sqrt{\epsilon_s}), \end{aligned} \tag{2.2}$$

$$\epsilon_s = \frac{1}{2} \int_0^T s^2(t) dt = A_m^2 \int_0^T g_T^2(t) dt = A_m^2 \cdot \epsilon_s, \tag{2.3}$$

kde ϵ_s vyjadřuje energii v pulzu g_t amplitudy A_m signálu s vytvořeného z těchto bází.

Z výše uvedených vzorců vyplývá, že ortogonální signály **mají shodnou energii** periody T . Dalším studiem zjistíme, že **skalární součin** dvou ortogonálních signálů s_m, s_n **musí být roven nule**.

$$\int_0^T s_m(t) \cdot s_n(t) dt = 0 \tag{2.4}$$

Pro naše využití ověříme ortogonalitu signálů modulovaných pomocí $\sin(x)$, $\cos(x)$ [1] :

$$u_m = s_m(t) \cos 2\pi f_c t, \quad m = 0, 1, 2, \dots, M, \quad 0 \leq t \leq T \tag{2.5}$$

Energie takového signálu je popsána [1] :

$$\epsilon_s = \int_0^T u_m^2(t) dt = \frac{1}{2} \int_0^T s_m^2(t) dt + \frac{1}{2} \int_0^T s_m^2(t) \cdot \cos 4\pi f_c t dt \tag{2.6}$$

Pro hledané ortogonální signály o stejné energii vytvářející M-bodový prostor tedy platí [1] :

$$u_m = \sqrt{\frac{2\epsilon_s}{T}} \cos(2\pi f_c t + \frac{2\pi m}{M}), \quad m = 0, 1, 2, \dots, M-1, \quad 0 \leq t \leq T \quad (2.7)$$

Výsledek nám říká, že při využití vhodně fázově posunutých signálů budou vykazovat vzájemnou ortogonalitu (PM modulace). Zjistíme, že toto vhodné posunutí je právě dáno funkcemi $\sin(x), \cos(x)$. Toto tvrzení podložíme výpočtem [1] :

$$\begin{aligned} \int_0^T u_m(t) \cdot u_n(t) dt &= \int_0^T s_m(t) s_n(t) \cos^2 2\pi f_c t dt \\ &= \frac{1}{2} \int_0^T s_m(t) s_n(t) dt + \frac{1}{2} \int_0^T s_m(t) s_n(t) \cos 4\pi f_c t dt = 0 \end{aligned} \quad (2.8)$$

Dalším rozбором zjistíme vzájemnou ortogonalitu i při použití vhodných funkcí lišících se násobkem kmitočtu (FM modulace) nebo vytvořených pomocí speciálních algoritmů (kap. 2.4). Použijeme-li tedy opět funkce $\sin(x), \cos(x)$, zjistíme vzájemnou ortogonalitu všech **harmonických** průběhů těchto funkcí (kap. 2.3). Další možné varianty ortogonálních průběhů jsou uvedeny na následující straně, kde je zřejmá vlastnost stejné signálové energie.

2.3 Ortogonalita ve frekvenční oblasti

Způsob dosažení ortogonalit ve spektru si ukážeme na nejjednodušším způsobu modulace FSK (Frequency-Shift Keying). Máme dva signály popsané [1] :

$$\begin{aligned} s_m(t) &= \sqrt{2\frac{\epsilon_b}{T_b}} \cos(2\pi f_m t), \\ s_n(t) &= \sqrt{2\frac{\epsilon_b}{T_b}} \cos(2\pi f_n t), \end{aligned} \quad (2.9)$$

kde $0 \leq t \leq T_b$ a ϵ_b je energie signálu v jednom bitu.

K dosažení ortogonalit je nutné splnit korelační podmínku [1] :

$$\begin{aligned} R_{mn} &= \frac{1}{\epsilon_s} \int_0^T s_m(t) s_n(t) dt = 0 \\ R_{mn} &= \frac{2\epsilon_s}{T} \cos(2\pi f_c t + 2\pi m \Delta f t) \cos(2\pi f_c t + 2\pi n \Delta f t) dt \\ &= \frac{\sin(2\pi(m-n)\Delta f T)}{2\pi(m-n)\Delta f T} = 0, \end{aligned} \quad (2.10)$$

kde ϵ_s je energie signálu v jednom symbolu a T je perioda symbolu

$$a \quad \epsilon_s = k \cdot \epsilon_b; \quad T = k \cdot T_b; \quad k = \log_2 M. \quad (2.11)$$

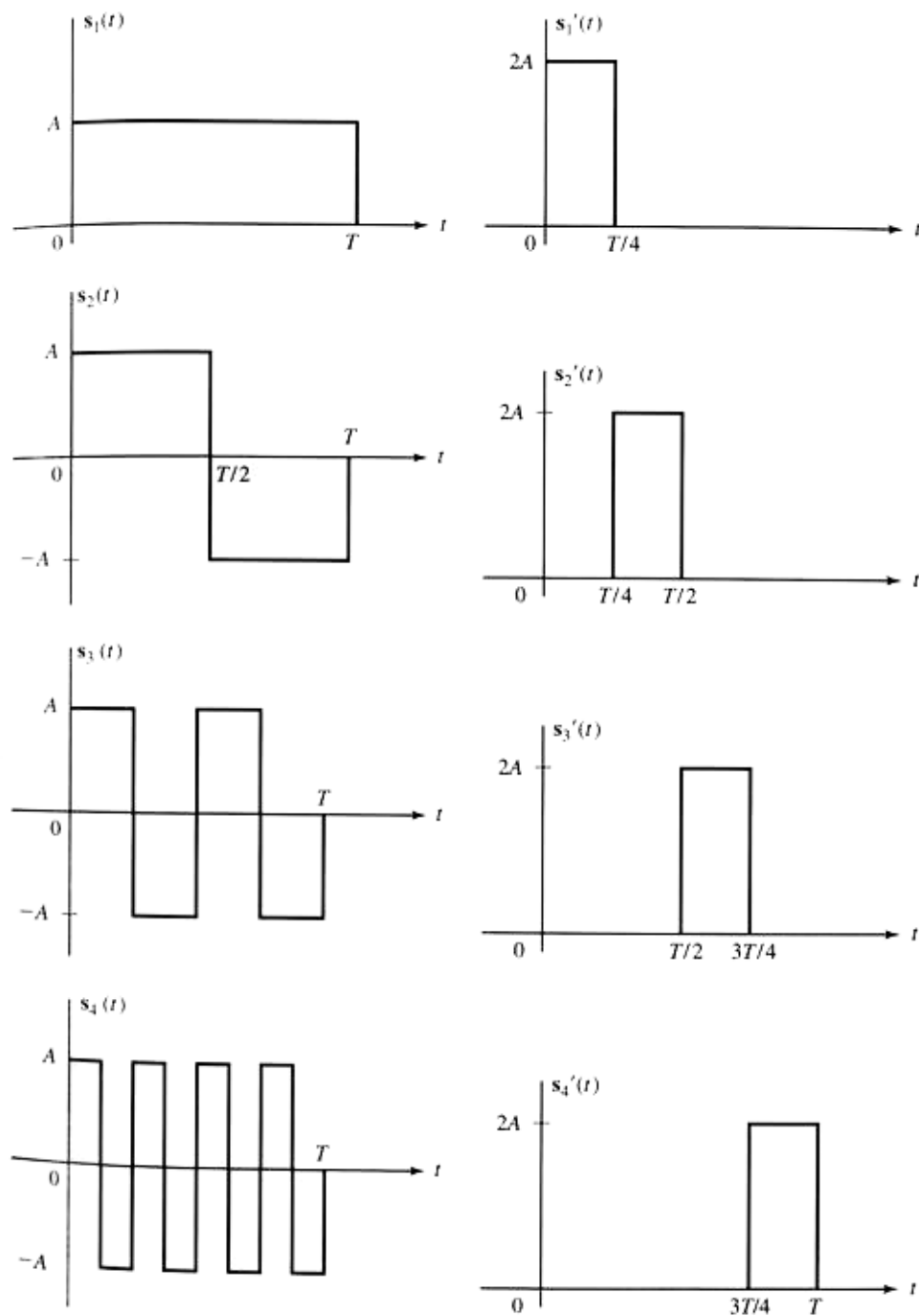
Z (2.10) vyplývá, že korelace R_{mn} mezi dvěma signály m, n je nulová pokud [1] :

$$\Delta f = f_n - f_{n-1} = \frac{1}{2T} \quad (2.12)$$

Rovnice (2.12) říká, že k dosažení ortogonálních funkcí je třeba docílit frekvenčního odstupu subnosných vln rovnému inverzní hodnotě dvou symbolových period. Jako korelaci vstupního signálu s *bázovou funkcí* (patříčná subnosná vlna) můžeme chápat použití DFT (Discrete Fourier Transform). V digitálních systémech se pro vyšší počet subkanálů používá FFT (kap. 2.5), která je výpočetně snažší a rychlejší. Vycházíme z faktu, že jakýkoli signál může být reprezentován v časové oblasti lineární kombinací ortogonálních funkcí, které jsou vzájemně nekorelované. Poté využijeme DFT a společně se souborem těchto ortogonálních bázových funkcí aplikujeme na signál.

Na frekvencích, kde je přenášena největší energie signálu se objeví maximální hodnota korelace tohoto signálu s bázovou funkcí. Jelikož jsou bázové funkce nekorelovány, korelaci v DFT pro danou subnosnou vlnu dostáváme pouze energii signálu na dané frekvenci. Z tohoto důvodu se spektra jednotlivých subkanálů mohou překrývat, aniž bychom ztratili potřebnou informaci. Ortogonalita v rámci OFDM tak říká, že díky konstantní amplitudě a fázi všech subnosných vln v rámci periody FFT a díky vhodnému rozmístění těchto subnosných kmitočtů je zaručena nezávislost jednotlivých symbolů a tudíž eliminace ICI.

Některé vzájemně ortogonální signály si můžeme pomocí obdélníkového průběhu ilustrovat v následujícím obrázku 2.3 [1].



Obr. 2.3: Příklad vzájemně ortogonálních funkcí v časové oblasti [1]

2.4 Ortogonalita pomocí DFT

Dosažení ortogonality pomocí metody DFT probíhá přibližně takto [1] :

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \quad x(t) = \int_{-\infty}^{\infty} X(\omega)e^{j\omega t}dt, \quad (2.13)$$

kde $x(t)$ nazýváme **originálem** a $X(\omega)$ **obrazem vstupního signálu**.

Podmínkou existence transformace původní funkce $x(t)$ je, aby transformovaná funkce byla po částech spojitá s konečným počtem bodů nespojitosti a aby integrál funkce $x(t)$ byl spojitý. Diskrétní Fourierovu transformaci pro k -tý vzorek definujeme :

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn} \quad \text{pro } k = 0, \dots, N-1. \quad (2.14)$$

Obdobně pro **inverzní diskrétní** Fourierovu transformaci :

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j\frac{2\pi}{N}kn} \quad \text{pro } n = 0, \dots, N-1. \quad (2.15)$$

Z uvedených vzorců je zřejmé, že přímá i zpětná Fourierova transformace využívá stejného algoritmu a dají se vzájemně přepočítat pomocí znalosti komplexně sdružených hodnot signálu :

$$x_n = (DFT\{X_k^*\})^*. \quad (2.16)$$

Po úpravě vztahů (2.14) a (2.15) lze dokázat, že vektory $e^{j\frac{2\pi}{N}kn}$ tvoří ortogonální báze funkce N -dimenzionálního prostoru [1].

$$\sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}k_1n})(e^{-j\frac{2\pi}{N}k_2n}) = N \cdot \delta_{k_1k_2}, \quad (2.17)$$

kde $\delta_{k_1k_2}$ je Kroneckerovo delta pro které platí :

$$\delta_{k_1k_2} = \begin{cases} 0 & \text{pro } k_1 \neq k_2 \\ 1 & \text{pro } k_1 = k_2 \end{cases} \quad (2.18)$$

Je zřejmé, že pro $k_1 \neq k_2$, tudíž pro 2 různé frekvence, splňuje rovnice (2.17) nutnou podmínku ortogonalit (2.4). Výsledné spektrum signálu je tedy tvořeno vzájemně ortogonálními funkcemi.

2.5 Algoritmus výpočtu FFT

Reprezentaci komplexní funkce $\exp(-j2\pi nk/N)$ chápeme jako koncový bod vektoru rotujícího po jednotkové kružnici, přičemž jeho pohyb bude nabývat diskretních hodnot. Každý následující bod tak bude pootočen o úhel daný poměrem $\frac{2\pi}{N}$. Tento parametr *faktor pootočení* (twiddle factor) je definován [13] :

$$W_N = e^{-j\frac{2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right) \quad (2.19)$$

Pro výpočet N -bodové Fourierovy transformace tudíž využijeme vzorce (2.14), (2.19) a výslednou hodnotu stanovíme jako součet hodnot vzorků vážených mocninou faktoru pootočení [13] :

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi nk}{N}}, \quad x_n = \sum_{k=0}^{N-1} X_k \cdot e^{j\frac{2\pi nk}{N}} \quad (2.20)$$

Při výpočtu N -bodové FFT se každý součet X_k vzorků signálu násobí mocninou W_N^{nk} . Pro výpočet všech koeficientů je zapotřebí stejný počet operací násobení jako sčítání. Teoreticky je třeba

vypočítá všechny tyto koeficienty, tedy N^2 mocnin faktoru pootočení. Zjednodušení výpočtu spočívá ve využití vlastností *symetrie a periodicity* této funkce. Mocniny faktoru pootočení se opakují s periodou N , což je ekvivalentní otočení vektoru W_N o úhel 2π a dokazuje periodicitu funkce [13].

$$W_N^k = W_N^{k+N}. \quad (2.21)$$

Symetrie je omezená na sudý počet bodů N a pro $\frac{N}{2}$ -násobné pootočení vektoru W_N o úhel π .

$$W_N^k = -W_N^{k+\frac{N}{2}}. \quad (2.22)$$

Pro N sudé stačí tedy vypočítat pouze $\frac{N}{2}$ hodnot funkcí *sinus* a *cosinus* argumentu $-2\pi \frac{n}{N}$ přičemž pracujeme s hodnotami $n = 0, 1, \dots, \frac{N}{2} - 1$.

Jelikož je nemožné jednoduše realizovat tento výpočet pomocí hardwaru obsaženého v FPGA, musíme operaci provést pomocí jednoho z mnoha možných výpočetních algoritmů. Jmenujme například algoritmy Cooley-Tukey, Nordic, Rader, dělení kořenů apod.. Pro náš účel použijeme nejčastěji používaného algoritmu Cooley-Tukey pro FFT, který je navíc přímo implementován do předprogramované knihovny bloku Fast Fourier Transform v3.2 v programátoru ISE Xilinx.

2.5.1 Cooley-Tukey

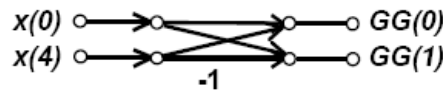
Algoritmus spočívá ve využití 2^N vzorků (Radix-2) popřípadě 4^N vzorků (Radix-4). Nejprve jsou rozděleny podle pořadí na liché a sudé; následně je nad těmito skupinami provedena dílčí FT. Přičemž výběr těchto vzorků může být realizován buď decimací v čase (DIT) nebo decimací ve frekvenční oblasti (DIF).

2.5.2 Decimace v časové oblasti (DIT)

Tato metoda pracuje s posloupností časových vzorků [13] :

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \cdot \exp(-j \frac{2\pi nk}{N}) \\ &= \sum_{v=0}^{N/2-1} x_{2v} \cdot \exp(-j \frac{2\pi vk}{N/2}) + \exp(-j \frac{2\pi k}{N}) \sum_{v=0}^{N/2-1} x_{2v+1} \cdot \exp(-j \frac{2\pi vk}{N/2}) \\ &= G_k + \exp(-j \frac{2\pi k}{N}) H_k. \end{aligned} \quad (2.23)$$

Při aplikaci FFT využíváme speciální struktury (2.4), tzv. **motýlek** na jehož 2 vstupy (radix-2) je přiveden komplexní signál x_n .



Obr. 2.4: Struktura motýlku radix-2 pro $N = 2$ [13]

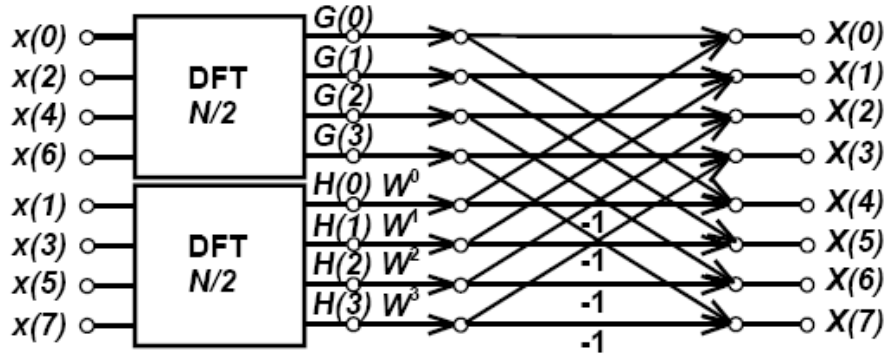
Výstupní signál je popsán [13] :

$$\begin{aligned} GG_0 &= x_0 + \exp\left(-2\pi \frac{0}{N}\right) x_4 = x_0 + x_4 \\ GG_1 &= x_0 - \exp\left(-2\pi \left(\frac{N}{2}\right) / N\right) x_4 = x_0 - x_4. \end{aligned} \quad (2.24)$$

Tyto vzorky x_n však musí být ve struktuře FFT seřazené podle jejich reverzované bitové reprezentace čísla n vzhledem k prostřednímu bitu. Názorné pořadí vzorků v motýlkové struktuře pro případ $N = 8$ je uveden v tab. 2.1.

Tab. 2.1: Řazení vzorků ve struktuře FFT

Výchozí pořadí desítkově	Dvojkové číslo	Reverzní pořadí bitů	Konečné pořadí desítkově
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7



Obr. 2.5: Zjednodušená struktura algoritmu FFT metodou DIT [13]

2.5.3 Decimace ve frekvenční oblasti (DIF)

Výpočet je založen na rozdělení **obrazové** posloupnosti vzorků X_k na liché a sudé $X_{2\mu}$ členy nad kterými je provedena dílčí FT [13].

$$\begin{aligned}
 X_{2\mu} &= \sum_{n=0}^{N-1} x_n \cdot \exp(-j \frac{2\pi\mu n}{N/2}) \\
 &= \sum_{n=0}^{N/2-1} x_n \cdot \exp(-j \frac{2\pi\mu n}{N/2}) + \sum_{n=0}^{N/2-1} x_{n+N/2} \cdot \exp(-j \frac{2\pi\mu(n+N/2)}{N/2}) \\
 &= \sum_{n=0}^{N/2-1} (x_n + x_{n+N/2}) \exp(-j \frac{2\pi\mu n}{N/2}).
 \end{aligned} \tag{2.25}$$

$$\begin{aligned}
 X_{2\mu+1} &= \sum_{n=0}^{N-1} x_n \cdot \exp(-j \frac{2\pi(2\mu+1)n}{N}) \\
 &= \sum_{n=0}^{N/2-1} (x_n - x_{n+N/2}) \cdot \exp(-j \frac{2\pi n}{N}) \exp(-j \frac{2\pi\mu n}{N/2})
 \end{aligned} \tag{2.26}$$

Provedeme-li následující substituci [13] :

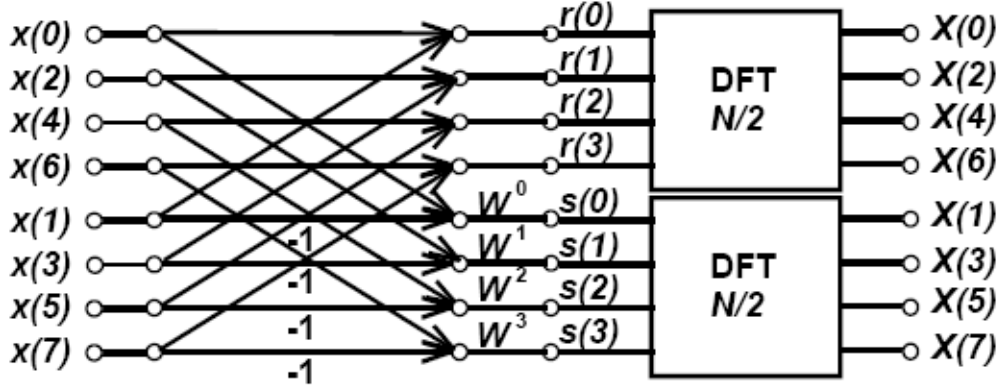
$$\begin{aligned}
 x_n + x_{n+N/2} &= r_n \\
 (x_n - x_{n+N/2})W^n &= s_n \quad \text{pro } n = 0, 1, \dots, N/2 - 1,
 \end{aligned} \tag{2.27}$$

můžeme vzorce (2.25) a (2.26) upravit do tvaru [13] :

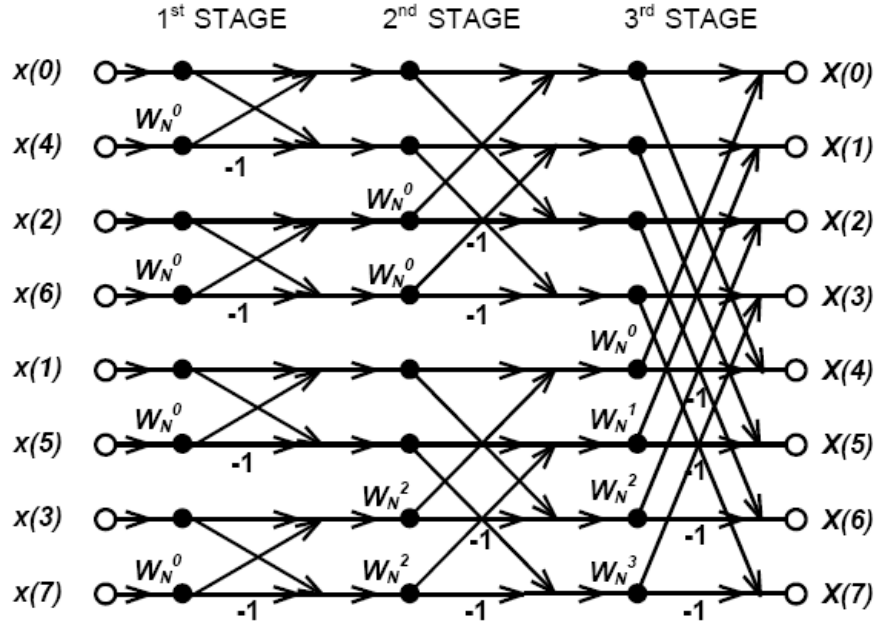
$$X_{2\mu} = \sum_{n=0}^{N/2-1} r_n \cdot \exp(-j \frac{2\pi\mu n}{N/2}) \quad (2.28)$$

$$X_{2\mu+1} = \sum_{n=0}^{N/2-1} s_n \cdot \exp(-j \frac{2\pi\mu n}{N/2}).$$

Principiální schéma výpočtu je uvedeno na obr. 2.6 a struktura motýlkové struktury radix-2 realizující výpočet prostřednictvím FFT je na obr. 2.7 [13].



Obr. 2.6: Zjednodušená struktura algoritmu FFT metodou DIF [13]



Obr. 2.7: Úplný signálový diagram 8-bodové FFT metodou DIT [13]

2.6 Problémy spojené s aplikací OFDM

V této kapitole byly uvedeny některé základní problémy, které se mohou při výsledném návrhu vyskytnout a bylo navrženo jejich možné řešení. Podrobné informace viz. [8].

2.6.1 Frekvenční odchylka oscilátoru

Na přijímací straně dochází ke zpracování signálu pomocí DFT (FFT), provádíme tedy konvoluci získaného signálu se souborem známých subnosných vln. Tyto známé subnosné kmitočty musí být v přijímači zpětně obnoveny. Zde nastává problém spočívající v přesné synchronizaci kmitočtů na vysílací a na přijímací straně.

Předpokládejme, že dochází k odchylce mezi těmito kmitočty [8] :

$$\Delta\omega = 2\pi\Delta f. \quad (2.29)$$

Poté můžeme zkreslení signálu $s(t)$ popsat [8] :

$$s(t) \cdot e^{j\Delta\omega t} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega - \Delta\omega) \quad (2.30)$$

Z toho vyplývá, že v důsledku této odchylky dochází k posunutí spektra přijímaného signálu. To však zapříčiní nesplnění podmínky ortogonalit (2.10) a dochází ke *ztrátě ortogonalit*. Následkem toho vzniká zkreslení signálu v důsledku tzv. interference IBI (Inter-Bin Interference), která vyplývá z následujícího mechanismu DFT. Na vstupu DFT je banka N filtrů, které vykonávají konvoluci signálu se subnosnou vlnou. V důsledku nepřesně nastavené nosné frekvence dochází ve spektru k přerozdělení části energie signálu i na okolní nosné kmitočty s patřičnými filtry. Tento jev se také nazývá DFT *únik* (leakage) a projevuje se jako aditivní Gaussovský šum v signálu, který způsobuje celkové *zhoršení SNR*.

Řešení :

Odchylku od správného kmitočtu můžeme eliminovat vynásobením signálu *korekčním faktorem*, kterým může být funkce sinus o frekvenci dané odchylky kmitočtů. Dalším řešením je použití jiné metody synchronizace nosných kmitočtů.

2.6.2 Fázová odchylka oscilátoru

Při obnově subnosných kmitočtů dochází k fázové odchylce mezi vysílačem a přijímačem. Zkreslení tedy můžeme vyjádřit v časové oblasti [8] :

$$s'(t) = s(t) \cdot e^{j\Delta\Phi}. \quad (2.31)$$

Výsledkem je otočení fáze ve frekvenční oblasti pro každý nosný kmitočet (dochází k pootočení konstalačního diagramu o daný úhel).

Řešení :

Pokud dochází jen k malé fázové odchylce, je možné zkreslení odstranit pomocí ekvalizéru, který vynásobí signál subnosné vlny komplexním koeficientem upravujícím fázi signálu. Toto řešení však může být použito jen pro malé fázové odchylky, kde nedochází k překryvu frekvenčních oblastí vymezených pro jednotlivé body konstalačního diagramu. Větší fázové odchylky musí být řešeny zvolením jiné metodiky stanovení fáze původní nosné vlny.

2.6.3 Nepřesná pozice FFT okna

Signál vstupující do bloku FFT je současně zpracováván po úsecích N vzorků. Ideálně je těchto N vzorků shodných se vzorky jednoho přijmutého OFDM symbolu. Nesprávná synchronizace se projeví jako časový posun FFT okna, tudíž se v krajním případě mezi N zpracovávanými vzorky FFT mohou objevit vzorky ze sousedního OFDM symbolu. Tento jev způsobuje vážné zkreslení konstalačního diagramu, nastává však pouze při vážných nedostatcích synchronizačního algoritmu. Častěji dochází k posunu těchto N vzorků mezi bankou filtrů v rámci pouze jednoho OFDM symbolu díky přítomnosti ochranného intervalu mezi těmito symboly. Tento časový posun vzorků způsobuje ve frekvenční oblasti lineární závislost fázového otočení konstalačního diagramu na velikosti posunu. Vzorky na nízkých frekvencích jsou tedy zkresleny menší odchylkou fáze než vzorky na vysokých frekvencích.

Řešení :

Eliminace tohoto zkreslení spočívá v přesné synchronizaci, při které se využívá korelace s tzv. *tréninkovou sekvencí* obsaženou taktéž v prvních znacích signálu. Tato sekvence dat je na přijímací straně předem známá a při korelaci se objevuje špička v místě začátku vysílání signálu. Znalost pozice této špičky tak umožní přijímači přesnou časovou synchronizaci.

2.6.4 Nepřesná vzorkovací frekvence

Analogový signál využitý při přenosu dat kanálem je nutné získat pomocí D/A převodníku (DAC). Vzorkovací frekvence, která je zde použita musí splňovat Shannon-Kotělníkův teorém $f_{max} = \frac{f_s}{2}$, kde f_s je vzorkovací frekvence. Převzorkováním dosáhneme stlačení spektra a naopak podvzorkováním dochází k rozšíření spektra, které může způsobit překryv sousedních spekter (tzv. Aliasing). Zároveň obě tyto varianty způsobují již uvedenou interferenci IBI v bloku FFT (2.6.2)

Řešení :

Toto zkreslení odstraníme korekcí vzorkovací frekvence, které se provádí generováním chybového slova, které se použije k úpravě vzorkovací frekvence.

2.7 COFDM

(Coded Orthogonal Frequency Division Multiplex)

V současnosti se prakticky nesetkáme s metodami OFDM, které nejsou založeny na COFDM, a proto tyto pojmy často splývají. Při použití velkého množství subnosných kmitočtů v OFDM dochází k nežádoucímu kmitočtovému úniku ve spektru. Pro širokopásmové vysílání se jedná o kmitočtově selektivní únik (selective fading) a pro úzkopásmové vysílání o tzv. plochý únik (flat fading), přičemž šířka pásma je určena trváním symbolové periody T_s . Dochází tak při dekódování k velké chybovosti získaného signálu vzhledem k originálu. Proto je narozdíl od OFDM u COFDM navíc využito *kanálového kódování a prokládání*. Technika COFDM se využívá např. pro DAB a DVB-T. Další podrobnosti viz. [12].

3 FORMÁTY VYJÁDŘENÍ BINÁRNÍCH ČÍSEL

Při práci s digitálními signály můžeme narazit na různé formy reprezentace binárních čísel, které jsou pro různé aplikace často preferovány. Jistou formou různé reprezentace jsou i různé kódy - Johnsonův, Grayův apod., kterými se však zabývat nebudeme. My se zaměříme na různé reprezentace binárních dat používané při práci s FPGA. Jelikož jsou tyto procesory používány v aplikacích DSP, bylo nutné vyvinout kód usnadňující operace násobení a dělení. Podrobnější informace [19].

3.1 Integer fixed-point representation

Jedná se o vyjádření N bitů ve formátu *integer* s pevnou desetinnou čárkou, která se však neuplatní, jelikož se jedná o celá čísla. Vyskytuje se ve dvou formách reprezentujících rozsah hodnot 2^N , a to pouze pro kladná čísla ve formě *unsigned integer* nebo pro záporná čísla v rozsahu -2^{N-1} až $2^{N-1} - 1$ ve formě *signed integer*.

3.1.1 Offset binary

Poslední uvedené reprezentace čísel se občas vyskytují pod názvem **signed (unsigned) offset binary**. Při použití *unsigned offset* reprezentace máme k dispozici rozsah hodnot 0 až $2^N - 1$ popsaný N bity, přičemž první bit nazýváme MSB (most significant bit) a poslední LSB (least significant bit). Rozsah hodnot *signed offset* je oproti tomu -2^{N-1} až $2^{N-1} - 1$ a první bit čísla je tzv. znaménkový bit (tab. 3.1). Při operacích sčítání čísel s opačnými znaménky však dochází k chybám. Není možné jednoduše čísla sečíst (problém znaménkového bitu) a je nutné respektovat pořadí odčítaných prvků.

3.1.2 Two's complement

Za účelem možného odečítání čísel byla vyvinuta *two's complement* reprezentace, která nedostatky původního formátu částečně odstraňuje. Odlišnost od normální reprezentace se týká pouze pořadí bitů u záporných čísel (tab.3.1). Bohužel ani tento formát není univerzální a nelze použít při operacích násobení potřebných v DSP. Možné řešení spočívá evidentně v použití speciálního formátu. V tabulce 3.1 si ještě stručně shrneme zmíněné reprezentace a jejich přepočet do dekadické podoby.

Tab. 3.1: Binární reprezentace čísel pro 4 bity

dekadicky	-8	-7	-6	-5	-4	-3	-2	-1	0
Offset	1111	1110	1101	1100	1011	1010	1001	1000	0000
Two's	1000	1001	1010	1011	1100	1101	1110	1111	0000

1	2	3	4	5	6	7
0001	0010	0011	0100	0101	0110	0111
0001	0010	0011	0100	0101	0110	0111

3.2 Fraction fixed-point representation

Jako univerzální formát pro použití v DSP byl proto vytvořen nový formát reprezentace čísel. Tento formát, nazývaný též **formát Qn.m**, vyjadřuje reprezentaci desetinného čísla, kde **n** je počet bitů pro hodnoty čísla před desetinnou čárkou (integer portion) a **m** je počet bitů pro reprezentaci hodnoty čísla za desetinnou čárkou (fraction portion). Zapisuje se ve formě **Qn.m**, nejčastěji však $n = 0$, proto lze zapsat zkráceně jako **Qm**. Obecný převod tohoto formátu do dekadické reprezentace je uveden v tab.3.2. Pro naší aplikaci bude využita 16-ti bitová architektura, tedy kód Q15. Kód Q15 je specifický pro 16-ti bitové DSP procesory, kde dosahuje přesnosti 2^{-15} a dynamického rozsahu 90dB. Bližší informace viz. [19].

Tab. 3.2: Algoritmus převodu kódu Q15 do dekadické formy

Pořadové číslo bitu	1	2	3	4	5	6	7	8	9	...	n
Váha daných bitů	-1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	...	2^{-n+1}

Posloupnost bitů	1	1	1	1	1	1	1	1	1	...	1
Dekadická reprezentace	0										

Posloupnost bitů	1	0	0	0	0	0	0	0	0	...	0
Dekadická reprezentace	-1										

Posloupnost bitů	0	1	1	1	1	1	1	1	1	...	1
Dekadická reprezentace	1										

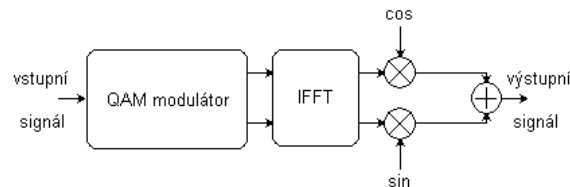
Posloupnost bitů	0	1	1	0	0	0	0	0	0	...	0
Dekadická reprezentace	0.75										

Posloupnost bitů	0	0	1	1	0	0	0	0	0	...	0
Dekadická reprezentace	0.275										

atd ...

4 SIMULACE OFDM MODEMU - MATLAB

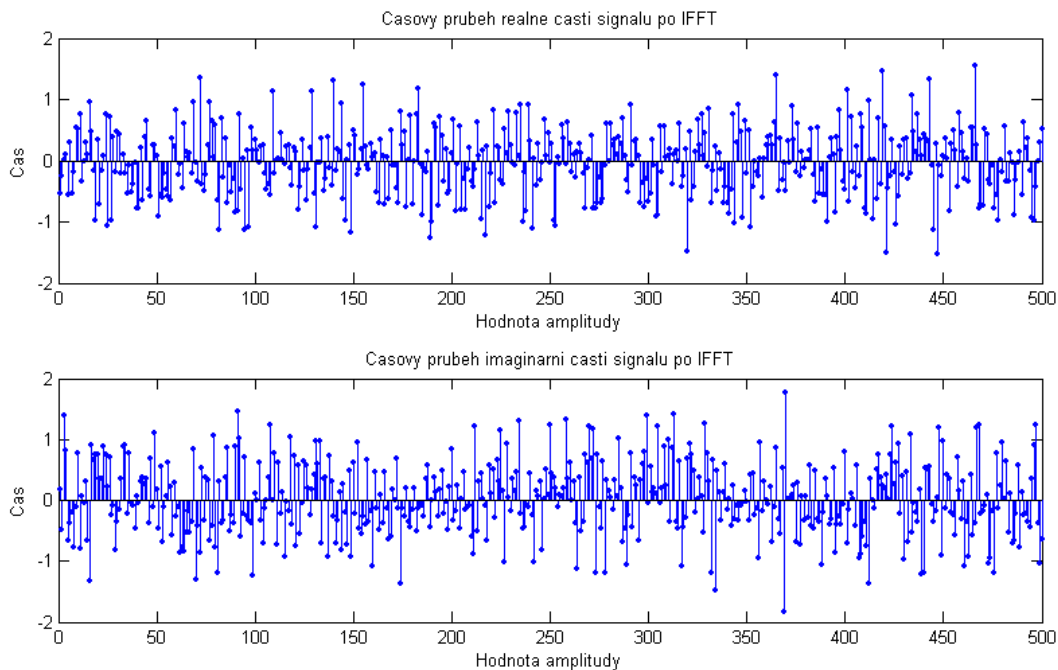
Úkolem dané simulace bude ověření činnosti funkce OFDM modemu, jenž bude implementován do obvodu Xilinx Virtex II. Abychom mohli výsledný návrh analyzovat, popřípadě optimalizovat, je při analýze dat využít program **Matlab 7.1**. V této části kapitoly se tedy zaměříme na analýzu dat získaných simulacemi v Matlabu a následující část věnujeme samotné implementaci a získaným hodnotám. Nakonec provedeme srovnání těchto dvou částí, abychom zhodnotili korektní funkci.



Obr. 4.1: Principiální schéma OFDM modemu

4.1 Statistická simulace OFDM modemu

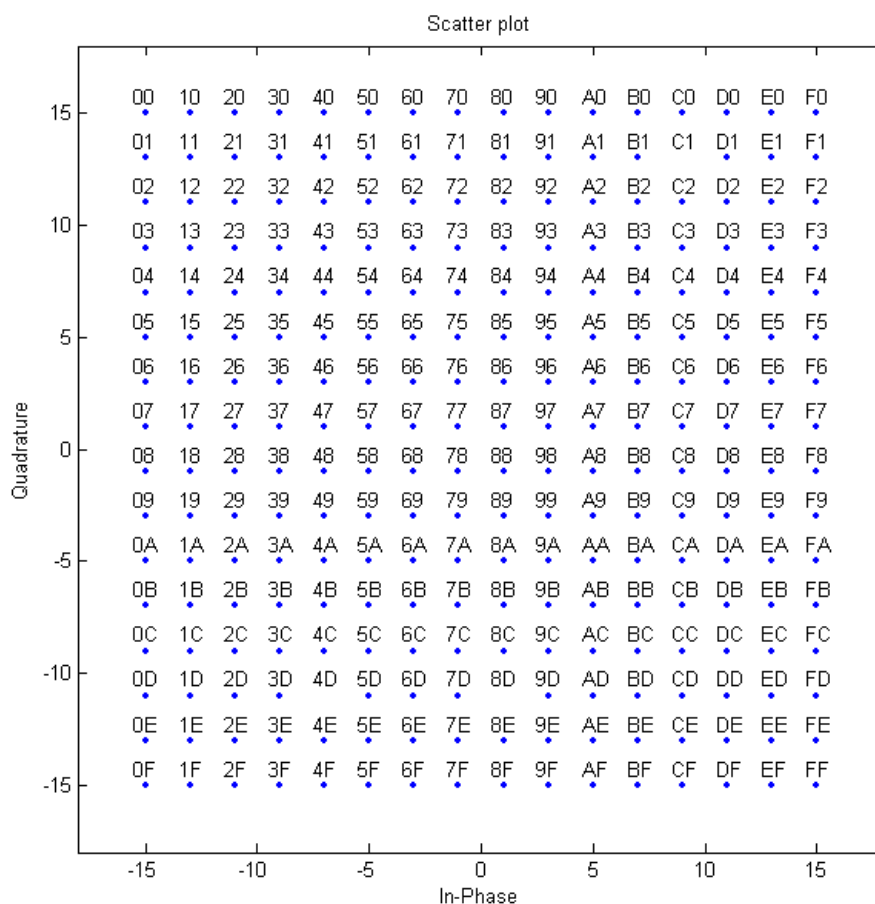
Nejprve provedeme statistickou analýzu náhodného signálu 256-QAM, který bude při implementaci použit. Budou nás zajímat zejména vlastnosti takového komplexního signálu po průchodu blokem realizujícím inverzní FFT. Vytvoříme si proto náhodný signál s modulací 256-QAM.



Obr. 4.2: Časový průběh signálu 256-QAM

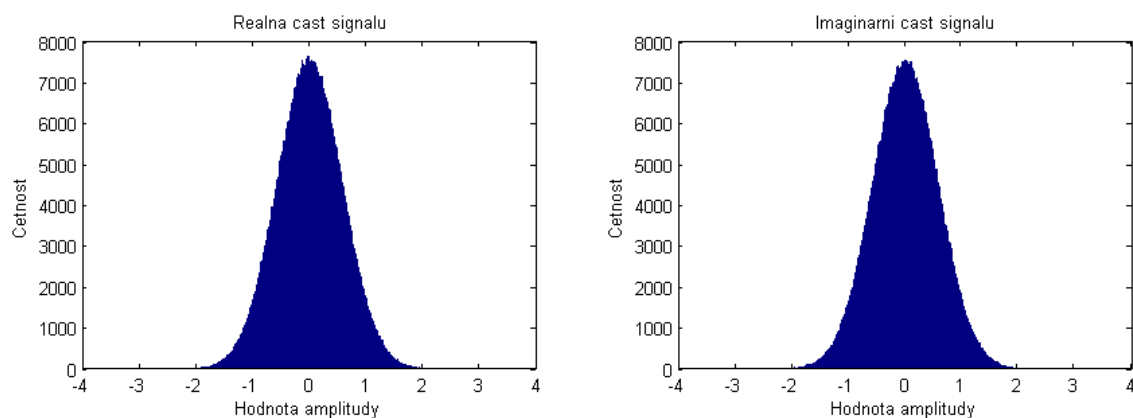
Takto modulovaný signál je komplexní a lze jej znázornit pomocí konstelačního diagramu uvedeného na obr. 4.3. Jednotlivé body reprezentují 8-mi bitové symboly, ale pro přehlednost jsou v diagramu zapsány v hexadecimálním formátu. Tento konstelační diagram popisuje normální mapování 256-QAM. Další možnou variantou je Grayovo mapování (obr. 7.3), jenž má výhodné vlastnosti při praktických aplikacích a korekci chyb. Tento způsob mapování je použit i při následné

implementaci, avšak pro analýzu statistických charakteristik nemá vliv na výsledky.

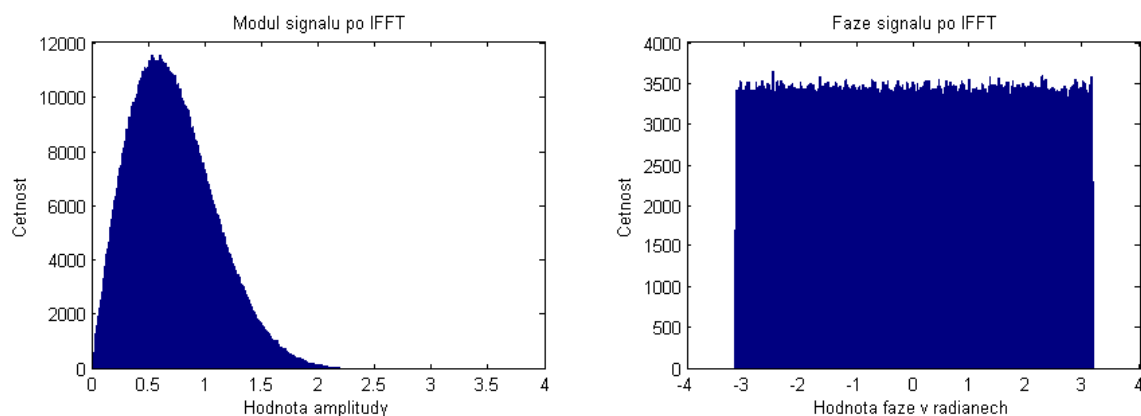


Obr. 4.3: Konstelační diagram signálu 256-QAM (Grayovo kódování)

Takto modulovaný signál byl transformován pomocí 16-ti bodové IFFT a z výstupních charakteristik byly vyvozeny obecné vlastnosti takto zpracovaného signálu (obr. 4.4 a 4.5).



Obr. 4.4: Charakteristiky výstupního signálu po IFFT - část 1



Obr. 4.5: Charakteristiky výstupního signálu po IFFT - část 2

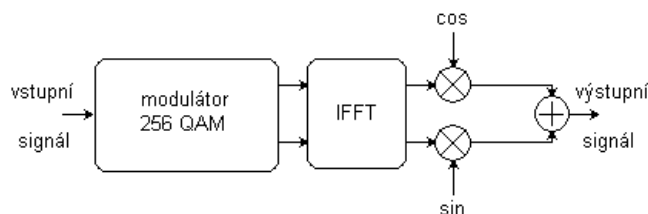
Závěr :

Z prvních průběhů vyjadřujících četnost výskytu hodnot amplitudy v signálu transformovaném pomocí IFFT (obr. 4.4) je patrné, že *reálná* a *imaginární* složka výstupního signálu vykazují Gaussovo rozdělení amplitudy. Z následujících průběhů (obr. 4.5) vyplývá, že *modul amplitudy* je popsán Rayleighovo rozdělením amplitudy a *fáze amplitudy* je konstantní v celém rozsahu hodnot.

Předpokládané výsledky se tak shodují s odsimulovanými a vyplívají z obecné teorie náhodných signálů. Použitý zdrojový kód je dostupný na datovém médiu `./matlab/qam_sim.m`.

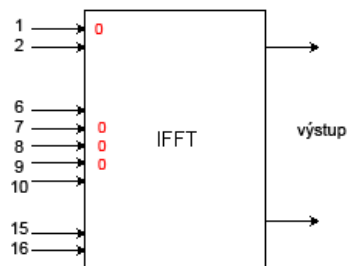
4.2 Simulace OFDM modemu ve spektrální oblasti

Jelikož bude při implementaci jediným možným prostředkem analýzy funkce OFDM modemu spektrální analyzátor, budeme se soustředit na spektra jednotlivých signálů. Za účelem porovnání předpokládaného spektra s reálně naměřeným, byla funkce OFDM modemu nejprve odsimulována programem Matlab. Na vstupu obvodového modelu byl generován náhodný signál, následně proběhlo 256-QAM mapování, 16-ti bodová transformace IFFT a výstupní kvadrurní modulace. Použité blokové schéma je uvedeno na obr. 4.6.

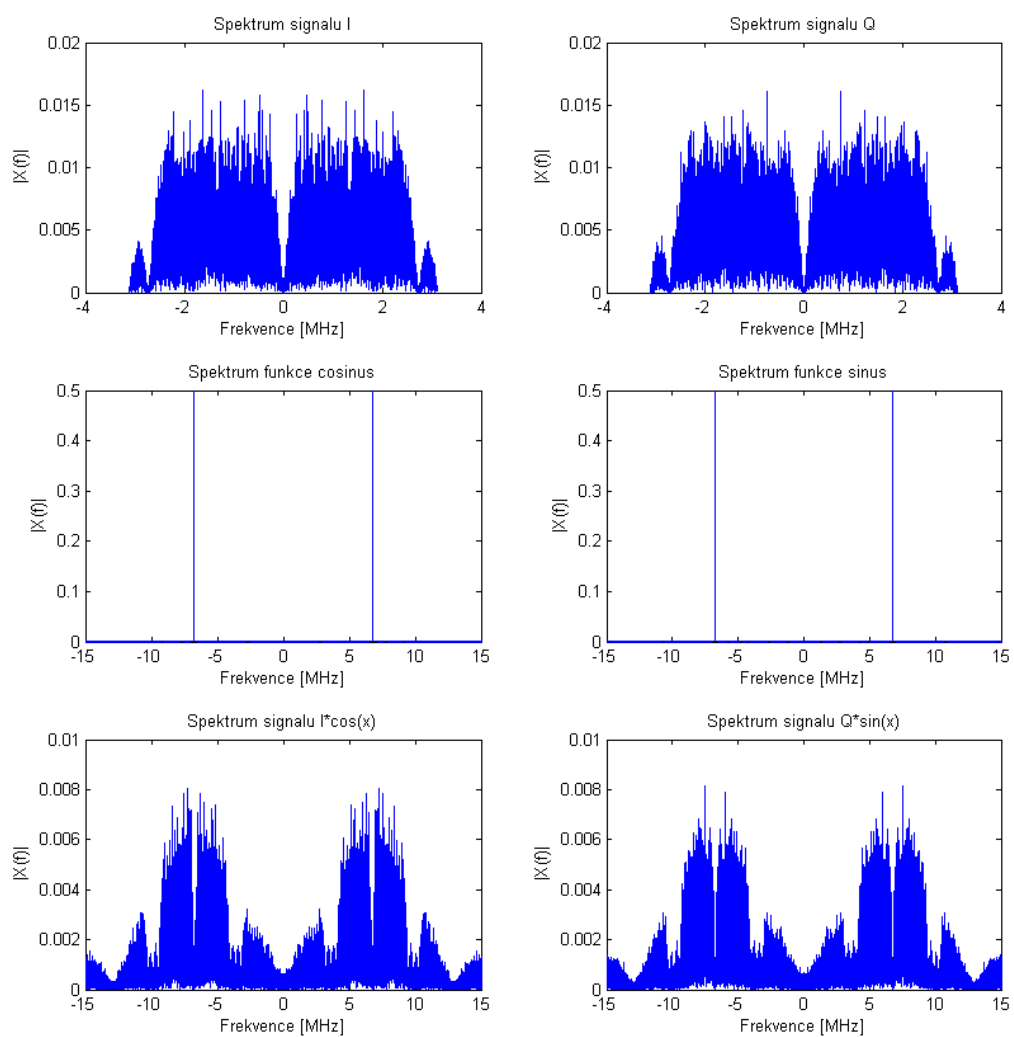


Obr. 4.6: Použité blokové schéma

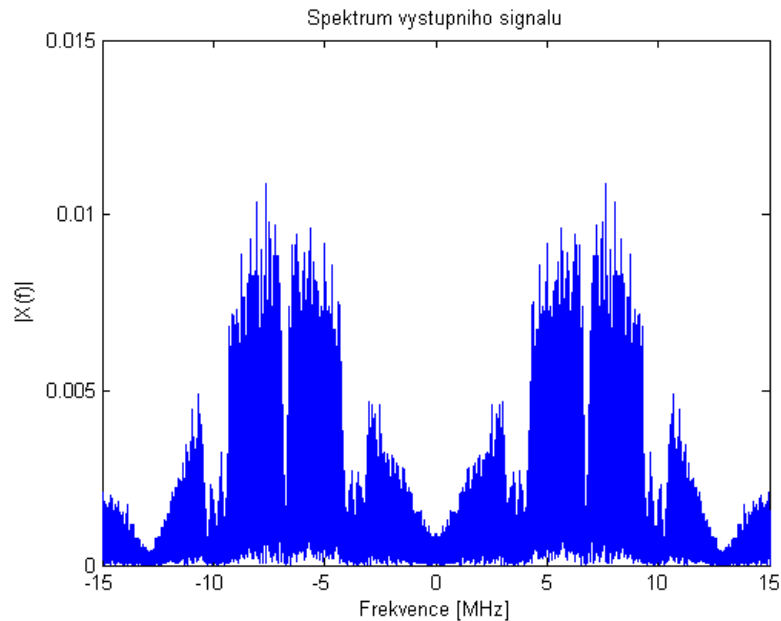
V bloku IFFT bylo použito vkládání nulových symbolů na specifické subnosné vlny (obr. 4.7). Vložení třech nulových symbolů doprostřed vstupní posloupnosti (subnosné vlny č.7,8,9) způsobuje potlačení bočních ramen spektra OFDM. Dále vložení nulového symbolu na první pozici IFFT (subnosná č. 0) dosáhneme vynulování stejnosměrné složky signálu, která je pro náš účel nepoužitelná z hlediska kapacitních vazeb výstupních filtrů obvodu. Spektra výstupních signálů bloků IFFT a rf modulátoru byly následně analyzovány a výsledky zaneseny do grafů na obr. 4.8 a 4.9.



Obr. 4.7: Vkládání nul na subnosné vlny



Obr. 4.8: Teoretické spektrum OFDM modemu - část 1



Obr. 4.9: Teoretické spektrum OFDM modemu - část 2

Parametry simulace :

Náhodný vstupní signál je generován bitovou rychlostí odpovídající $f_b = 50\text{ Mb/s}$ a modulován pomocí 256-QAM modulace, symbolová rychlost je tedy rovna $f_s = \frac{f_b}{\log_2(256)} = 6,25\text{ MS/s}$. Takto namodulovaný signál je přiveden na blok 16-ti bodové *IFFT*. Frekvenci OFDM symbolů vypočteme jako $f_{OFDM} = \frac{f_s}{N} = 390,6\text{ kHz}$. Šířka pásma výsledného signálu je tedy dána $B = N \cdot f_{OFDM} = 6,25\text{ MHz}$. Následná QAM modulace v sekci RF modulátoru prování násobení signálu s nosnými vlnami $\cos(x)$ popř. $\sin(x)$, které jsou na kmitočtu $f_0 = 6,75\text{ MHz}$ a jejich sečtení (respektive odečtení).

Poznámky

Při simulaci bylo použito 160 000 bitů náhodného signálu, což odpovídá počtu 625 OFDM symbolů. Tento počet symbolů byl pro simulaci postačující a výsledky odpovídali požadované funkci. Zdrojový kód použité simulace je dostupný na datovém médiu (`./matlab/spectra.m`).

5 OBVODY FPGA

Nabídka firem distribuuujících obvodu FPGA je vcelku pestrá - Altera, Atmel, Actel, Lattice, Xilinx. Nabízený sortiment je rozsáhlý a řada z nich poskytuje i rozsáhlou softwarovou základnu k dodávaným obvodům. My se zaměříme pouze na firmu **Xilinx**, jejíž software (vývojové prostředí ISE Xilinx 9.1i, simulátor ModelSim III 6.2g) a hardware (Virtex II) bude v této práci použit pro realizaci zadaného tématu. Firma Xilinx jako první na světě vyvinula programovatelná hradlová pole a v současnosti vyrábí více než polovinu světové produkce těchto součástek. Poskytuje bezkonkurenčně nejširší sortiment PLD v širokém spektru pouzder, pracovních podmínek, rychlostí apod. Obecně všechny obvody FPGA Xilinx používají konfiguraci pomocí statické paměti RAM. To znamená, že po připojení napájení je nutné vždy znovu nahrát konfiguraci obvodu. Výhodou tohoto řešení je téměř nekonečná reprogramovatelnost FPGA a také rychlost. Konfigurační propojky pracující na principu paměti RAM jsou tak rychlejší než přepínače založené na principu například EEPROM.

5.1 Řada Virtex IITM

Jelikož je výsledný návrh OFDM modemu určen pro implementaci do obvodu Xilinx Virtex II XC2V1000, budeme se nyní zabývat touto řadou podrobněji. Obvody byly vyvinuty pro kompletní HW design obvodů určených pro telekomunikace, sítě, video technologie a DSP. FPGA modul má HW podporu pro rozhraní PCI, LVDS a DDR. Výčet obvodů řady Virtex II je uveden v následující tabulce 5.1.

Tab. 5.1: Nabídka obvodů Virtex II [17]

Typ	Počet logických buněk	Počet bloků násobiček	Počet DCM ¹ bloků	Distribuovaná kapacita RAM [kbit]	Celková kapacita RAM [kbit]
XC2V40	40 000	4	4	8	72
XC2V80	80 000	8	4	16	144
XC2V250	250 000	24	8	48	432
XC2V500	500 000	32	8	96	576
XC2V1000	1 mil.	40	8	160	720
XC2V1500	1,5 mil.	48	8	240	864
XC2V2000	2 mil.	56	8	336	1008
XC2V3000	3 mil.	96	12	448	1728
XC2V4000	4 mil.	120	12	720	2160
XC2V6000	6 mil.	144	12	1056	1104
XC2v8000	8 mil.	168	12	1456	3024

5.2 Řada Virtex 4TM

Pro informaci je zde uveden i výčet poněkud novější řady Virtex 4. Tento výčet je zde uveden z důvodu možného využití obvodu XC4VSX35 na půdě fakulty FEKT.

Tab. 5.2: Nabídka FPGA obvodů Xilinx Virtex-4 [17]

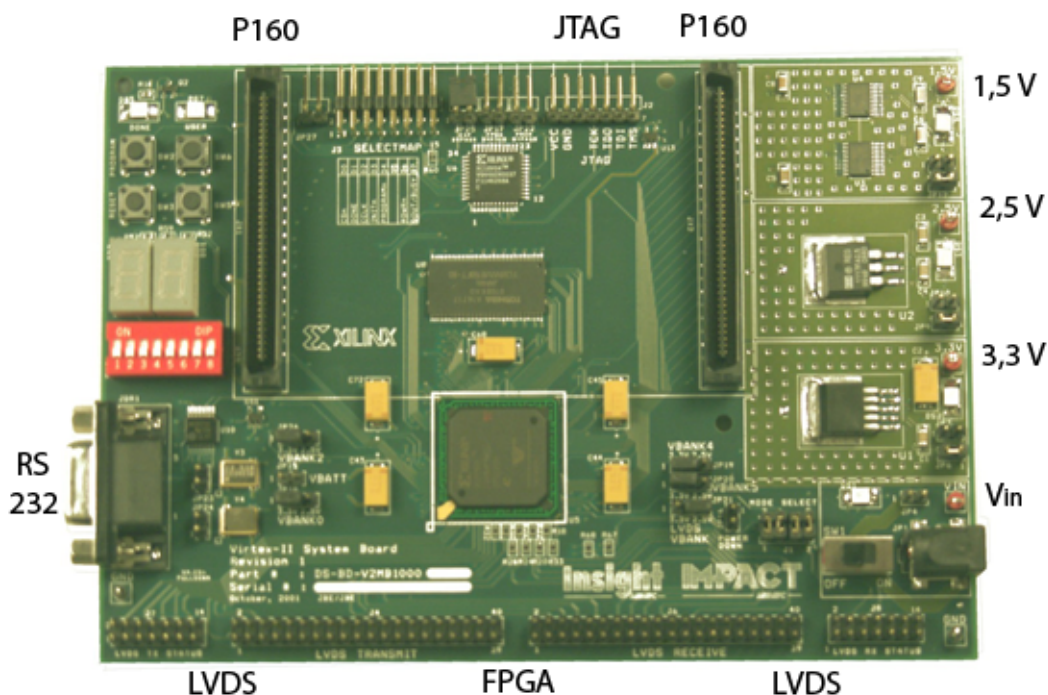
Typ	Počet pinů	Počet logických buněk	Syst. hradel [tis.]	Distrib. RAM [kbit]	Bloková RAM [kbit]
XC4VLX15	240/320	13,824	12,288	98,304	1,296
XC4VLX100	768/960	110,592	98,304	786,432	4,320
XC4VLX200	960	200,448	178,176	1,425,408	6,048
XC4VSX25	320	23,040	20,480	163,840	2,304
XC4VSX35	448	34,560	30,720	245,760	3,456
XC4VSX55	640	55,296	49,152	393,216	5,760
XC4VFX40	352/576	41,904	37,248	297,984	2,592

6 VIRTEX II XC2V1000

Implementace obvodu byla provedena v obvodu *XC2V1000* pouzdra FG456 a pro rychlostní stupeň -4. Tento obvod je součástí vývojové desky V2MB1000, která je dostupná studentům FEKT.

6.1 V2MB1000

Deska je napájena z externího stabilizovaného zdroje a naprogramována přes rozhraní JTAG připojeného přes konvertor do USB. Obrázek desky se stručným popisem je na obr. 6.1



Obr. 6.1: Virtex 2TM Development board - V2MB1000

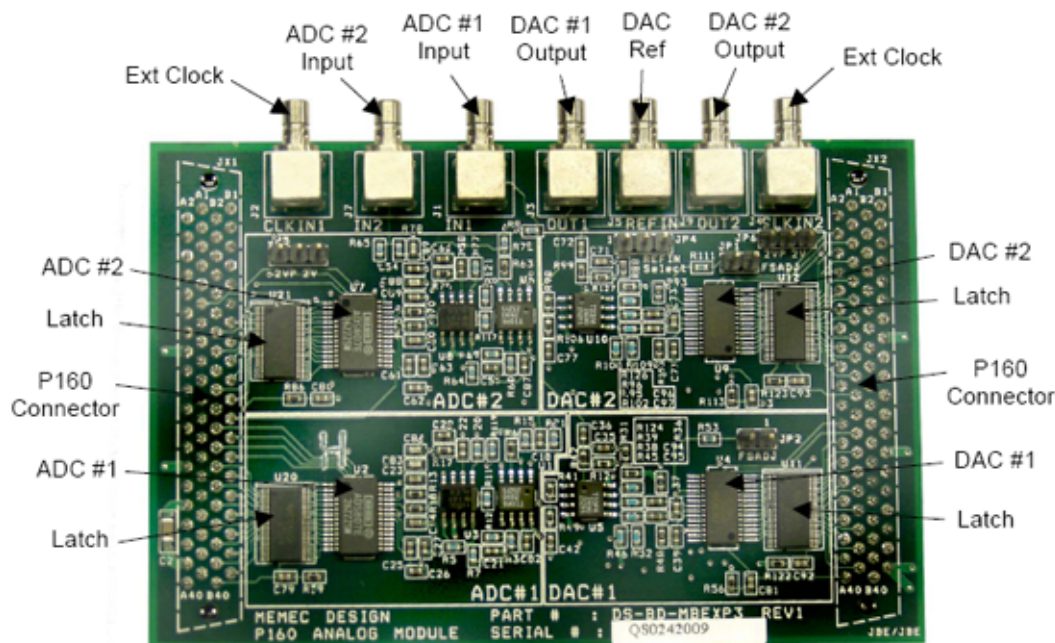
Samotná vývojová deska je však pro analogový signál prakticky nepoužitelná. Pro vstupní a výstupní signály používá sběrnici LVDS, která je pro náš účel nevhodná. Proto použijeme analogový modul P160 (obr. 6.2), který analogový signál zpracovává a v digitální formě předává procesoru prostřednictvím 160 pinové patice.

6.2 P160

Tento modul obsahuje zejména AD převodníky *TI ADS807* a DA převodník *TI DAC902*. Převodníky pracují s rozsahem 12 bitů a číselnou reprezentací pomocí *unsigned offset binary* (kap. 3). Při práci s nimi je nutné respektovat katalogové parametry těchto převodníků [21], přičemž amplitudy vstupních signálů ADC mohou být signálem *FSSEL* nastaveny až na hodnotu $3V_{pp}$. Výstupní signály DAC se budou pohybovat mezi 1,5V až 3,5V (tab. 6.4).

6.3 Implementace

Při implementaci se v programátoru ISE nejprve vytvoří přiřazení pinů FPGA čipů k desce VMB1000. To znamená připojit interní krystalový oscilátor desky a poté asociovat požadovaný



Obr. 6.2: Analogový modul P160

výstupní signál na s piny DA převodníku. Toto přiřazení je provedeno pomocí rozhraní PACE integrovaného v programátoru ISE. Návrh pouzdra FG456 v prostředí PACE je uveden na obr. 6.5. Význam jednotlivých pinů může být dohledán v datasheetu [20] na str. 111 až 124. Pro hodinový signál můžeme využít hned 3 možné signály (tab. 6.1). Pokud není nastaven globální hodinový

Tab. 6.1: Hodinové signály desky V2MB1000 a jejich přiřazení

Signál	číslo pinu	typ	popis
CLK.CAN2	B11	vstupní	interní oscilátor 100 MHz
CLK.CAN1	A11	vstupní	interní oscilátor 24 MHz
CLK.CAN3	F12	vstupní	externí hodinový signál (2.5V)

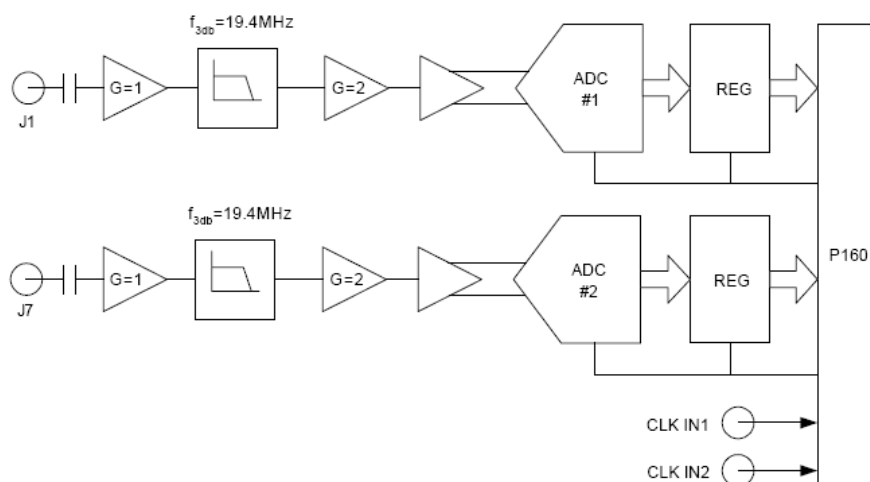
signál, je možné přivést 2 externí hodinové signály na JP2 a JP4 modulu P160, které jsou konfigurovatelné pomocí jumperů JP5 a JP6.

6.3.1 AD převodník

Pro další využití je možné nastavit parametry AD převodníku a asociovat jeho výstupy s FPGA (tab. 6.2). Pro náš účel však nebude AD převodník využit, proto je možné nastavení uvedeno pouze jako informativní. Převodník ADS807 vzorkuje vstupní na vzestupnou hranu hodinového signálu. Výstupní data se na výstupu objeví se zpožděním 6-ti hodinových cyklů. Signál o amplitudě $1V_{pp}$ nebo $1,5V_{pp}$ přivedeme na vstup J1 (ADC1). Nastavený rozsah je možné změnit nastavením zisku ADC pomocí odporů vstupního filtru. Tento rozsah je navíc možné změnit signálem FSSEL. Vstupní blok ADC obsahuje konfigurovatelný filtr DP s mezním kmitočtem $f = 19,4MHz$ (obr. 6.3).

6.3.2 DA převodník

Integrovaný převodník *TIDAC902* pracuje s 12-ti bitovým rozlišením a vzorkováním $165Ms^{-1}$. Výstupní signál je zpožděn oproti signálu FPGA o 1 periodu díky registru. Navíc DAC pracuje s dvěma hodinovými signály. První je odvozen z hodinového signálu FPGA a druhý musí být



Obr. 6.3: Blokové schéma vstupní části P160

Tab. 6.2: Signály ADC a jejich přiřazení na V2MB1000

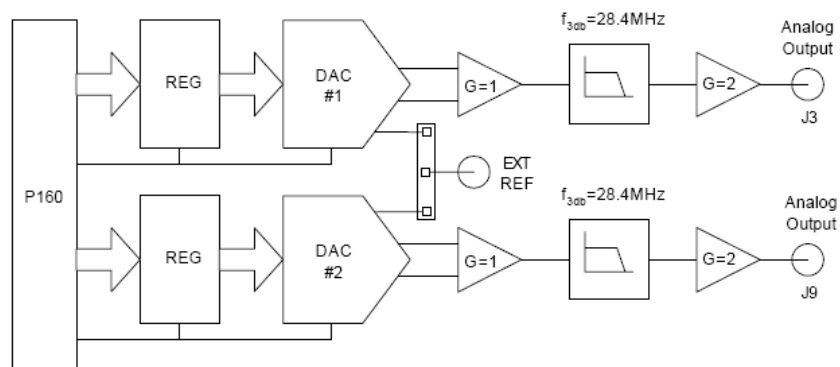
Konektor P160	signál	popis	pin V2MB1000
B24	ADC1.OEn	output enable (Low=povolen, High = tří stavový)	J20
B25	ADC1.REFSEL	reference select (Low = Interní, High = Externí)	J19
B26	ADC1.OTR	over-range indicator	H19
B27	ADC1.FSSEL	full scale select (Low = 2V _{pp} , High = 3V _{pp})	G20
B28	ADC1.DB1	MSB výstupní bit 0	E19
B29	ADC1.DB2	výstupní bit 1	E20
B30	ADC1.DB3	výstupní bit 2	L17
B31	ADC1.DB4	výstupní bit 3	K17
B32	ADC1.DB5	výstupní bit 4	J17
B33	ADC1.DB6	výstupní bit 5	J18
B34	ADC1.DB7	výstupní bit 6	H18
B35	ADC1.DB8	výstupní bit 7	G18
B36	ADC1.DB9	výstupní bit 8	F18
B37	ADC1.DB10	výstupní bit 9	E18
B38	ADC1.DB11	výstupní bit 10	E11
B39	ADC1.DB12	LSB výstupní bit 11	A10
B40	ADC1.CLK	hodiny ADC (nástupná hrana)	B10

opožděn vůči prvnímu o polovinu periody, aby nedošlo k chybnému zpracování. Výstupní část DAC obsahuje filtr DP s mezním kmitočtem $f = 28,4MHz$, s kterým je při návrhu také nutno počítat (obr. 6.4).

Výstupní signál vyčítaný na nástupnou hranu signálu DAC.CLK nabývá hodnot napětí daných tab. 6.4. Z tabulky je zřejmé, že formát binárních dat neodpovídá formátu výstupního signálu OFDM modemu, který je ve formátu $Q15$. Je tak nutné provést jednoduchou korekci výstupních dat OFDM modemu formou inverze MSB bitu. Tím upravíme data do požadované podoby. Výstupní signál DAC1 se poté objeví na výstupu J3, přičemž ještě musíme provést přiřazení signálů FPGA k tomuto převodníku podle tab. 6.5. Pro ilustraci přiřazení pinů přes rozhraní PACE je ještě uveden obr. 6.5, kde je patrný náčrtek pouzdra s uvedenými souřadnicemi odpovídajícími přiřazovaným pinům. Pro další podrobnosti odkazují na datasheety [20], [21] a [22].

Tab. 6.3: Výstupní signály ADC pro $1V_{pp}$

Přivedený signál	Výstupní signál
+0.5V	1111 1111 1111
0V	1000 0000 0000
-0.5V	0000 0000 0000



Obr. 6.4: Blokové schéma výstupní části P160

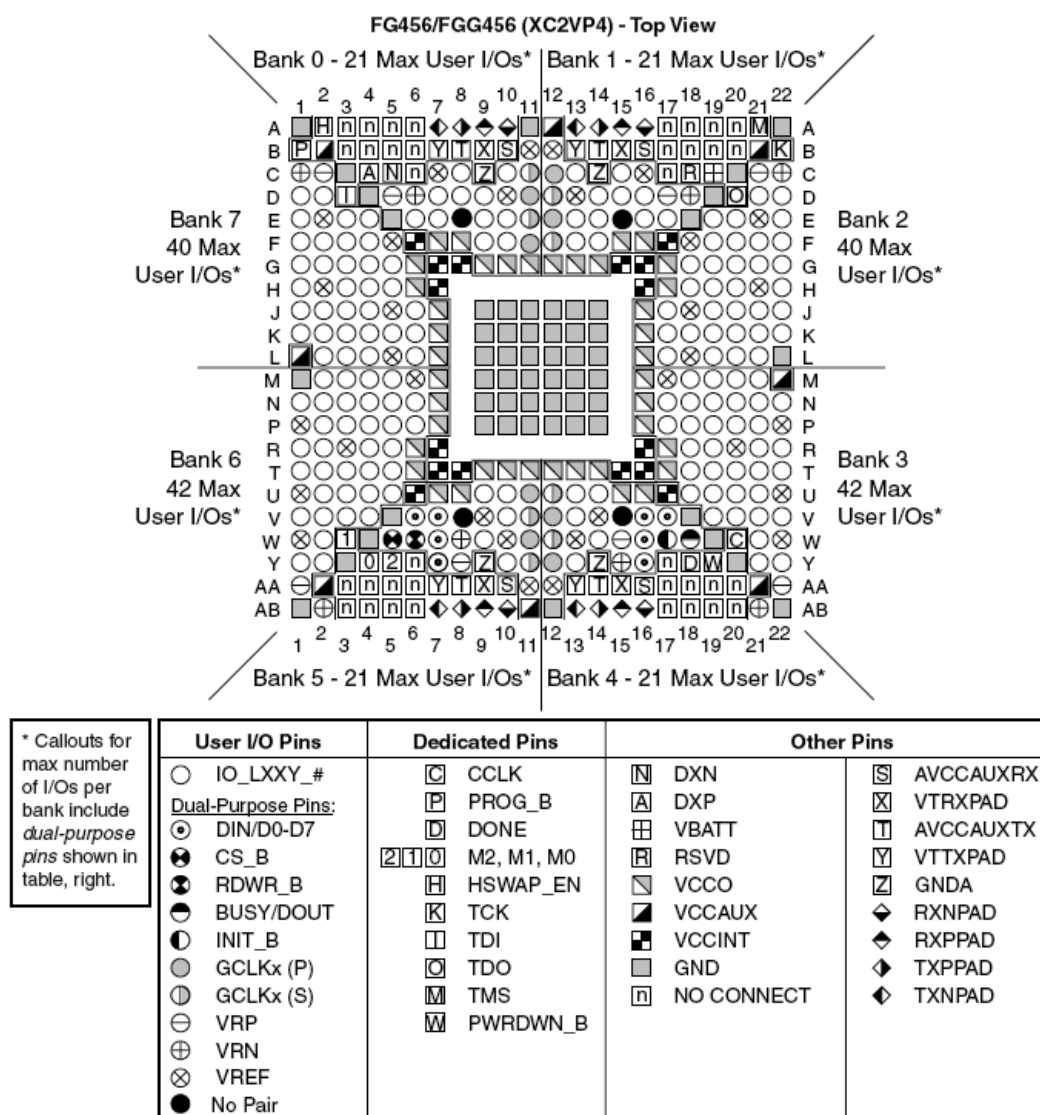
Tab. 6.4: Výstupní signály DAC1

Přivedený signál DB1 až DB12	Výstupní signál (J3)
1111 1111 1111	3.5V
1000 0000 0000	2.5V
0000 0000 0000	1.5V

Tab. 6.5: Signály DAC desky V2MB1000 a jejich přiřazení

Konektor P160	signál	popis	pin V2MB1000
A23	DAC1.CLK2	hodiny registru	U13
A24	DAC1.PD	power down (Low=Normální, High=Power down)	AA9
A25	DAC1.REFSELECT	reference select (Low=Interní, High=Externí)	U12
A26	DAC1.DB14	nepoužit	AB8
A27	DAC1.DB13	nepoužit	U11
A28	DAC1.DB12	LSB vstupní bit 0	AA8
A29	DAC1.DB11	vstupní bit 1	U10
A30	DAC1.DB10	vstupní bit 2	AB7
A31	DAC1.DB9	vstupní bit 3	U9
A32	DAC1.DB8	vstupní bit 4	AA7
A33	DAC1.DB7	vstupní bit 5	V9
A34	DAC1.DB6	vstupní bit 6	AB6
A35	DAC1.DB5	vstupní bit 7	V8
A36	DAC1.DB4	vstupní bit 8	AA6
A37	DAC1.DB3	vstupní bit 9	V7
A38	DAC1.DB2	vstupní bit 10	AB5
A39	DAC1.DB1	MSB vstupní bit 11	V6
A40	DAC1.CLK	hodiny DAC (nástupná hrana)	AA5

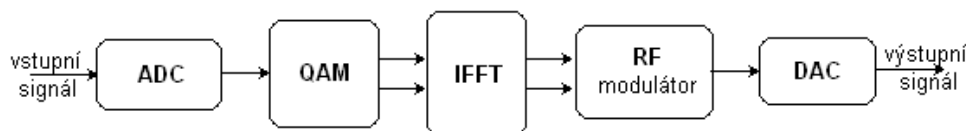
FG456/FGG456 Composite Pinout Diagram (XC2VP4)



Obr. 6.5: Přiřazení pinů XC2V1000 FG456

7 SIMULACE OFDM MODEMU - IMPLEMENTACE

V této části kapitoly se budeme věnovat analýze dat získaných při návrhu OFDM modemu pomocí softwarového vybavení distribuovaného společně s obvodem. Samotné programování probíhá prostřednictvím jazyku VHDL a programátoru **ISE Xilinx 9.1i SP3**. K funkčním simulacím je použit obvodový simulátor **ModelSim SE 6.3c**, který na základě naprogramovaných vstupních vektorů simuluje výstupní signály. Stěžejním článkem celé konstrukce realizující funkci *OFDM* je blok *IFFT*, který je realizován pomocí předprogramované knihovny Fast Fourier Transform v3.2. Tato knihovna spadá do sekce *IP cores*, jenž obsahuje některé předprogramované funkční bloky a usnadňuje tak práci při návrhu. Blokové schéma navrženého modemu je uvedeno na obr. 7.1.

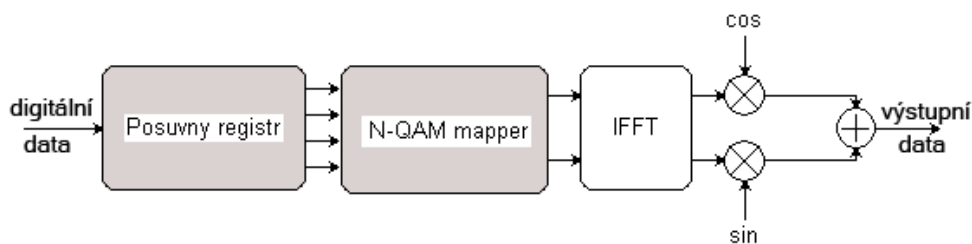


Obr. 7.1: Funkční blokové schéma OFDM modemu

AD a DA převodníky jsou integrovány na přídavném modulu P160, který je prostřednictvím 160-ti pinové patice připojen na vývojovou desku V2MB1000 (kap. 6). Tyto převodníky však nemají vliv na výsledky teoretických simulací, proto jsou v následujících blokových schématech již vypuštěny. Cílem tedy bude naprogramovat a odsimulovat zbylé funkční bloky modemu. Budou tedy provedeny simulace sloužící k ověření funkce jednotlivých bloků a na závěr simulace kompletního návrhu formou spektrální analýzy realizovaného modemu.

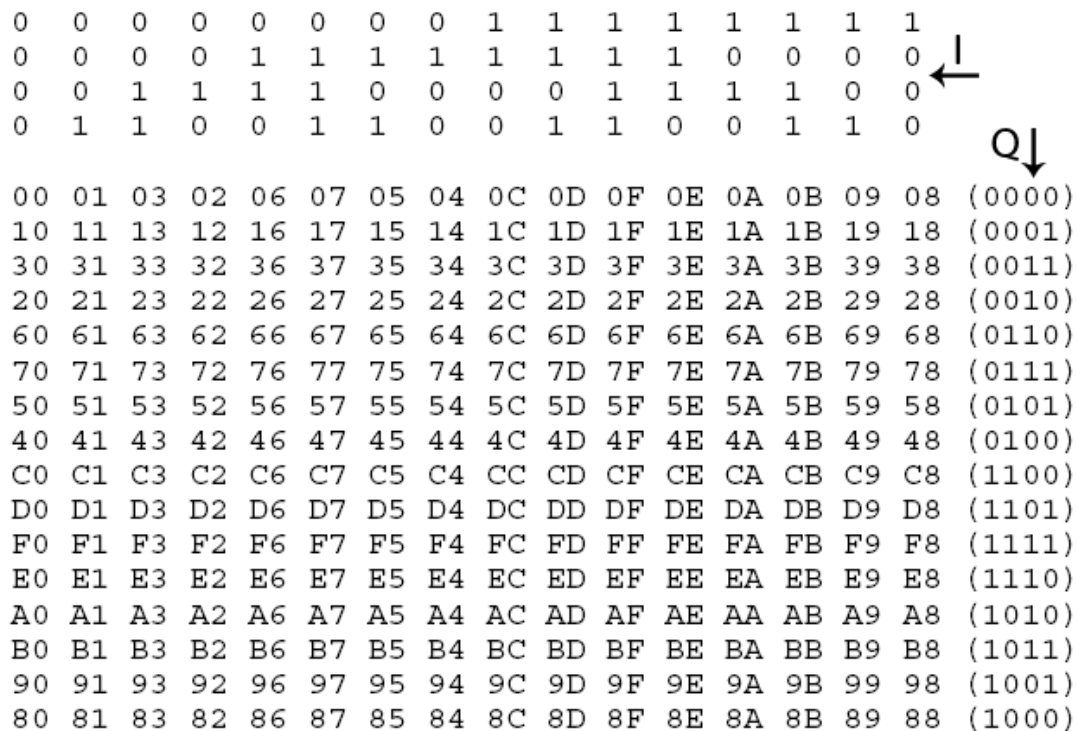
7.1 Modulátor 256-QAM

Pro implementaci byl zvolen vstupní signál modulovaný pomocí *256-QAM* s Grayovým mapováním. Konstalační diagram je v hexadecimálním formátu uveden na obr. 7.3. Realizace takového modulace uvnitř obvodu *FPGA* byla naprogramována pomocí dvou funkčních bloků - posuvného registru a vlastního QAM-mapování.



Obr. 7.2: Blokové schéma - QAM

Blok posuvného registru zde plní roli generátoru symbolů pro QAM-mapper. Signálový vektor zpracovaný v posuvném registru tedy vytvoří posloupnost bitů reprezentující symbol, který je přiveden na blok *QAM-mapperu* realizujícího Grayovo mapování *256-QAM*. Jelikož je v další fázi použito knihoven *IP core*, je nutné respektovat formát čísel DSP a výstupní signály převést do formátu Q15 (kap. 3.2).



Obr. 7.3: Konstalační diagram 256-QAM

Pro ilustraci funkce QAM mapperu je zde uvedena část zdrojového kódu `qam256.vhd` :

```
entity qam256 is
    Port ( clk : in  STD_LOGIC;
          data : in  STD_LOGIC;
          xn_real : out  STD_LOGIC_VECTOR (15 downto 0);
          xn_imag : out  STD_LOGIC_VECTOR (15 downto 0);
          clk4 : in  STD_LOGIC;
          clk16 : in  STD_LOGIC);
end qam256;

...
architecture Behavioral of qam_mapper is
    constant st_8 : STD_LOGIC_VECTOR (15 downto 0) := "1000000000000000";
    constant st_7 : STD_LOGIC_VECTOR (15 downto 0) := "1001000100010001";
    ...
    process (clk_b) begin
        if (clk_b'event and clk_b = '1') then
            vstup(0) <= Q0;
            vstup(1) <= Q1;
            vstup(2) <= Q2;
            ...
        end if;
    end process;

    process (clk_m) begin
        if (clk_m'event and clk_m = '1') then
            case vstup (7 downto 4) is
                when "0000" => out_Q <= (st8);
                when "0001" => out_Q <= (st7);
            end case;
        end if;
    end process;
```

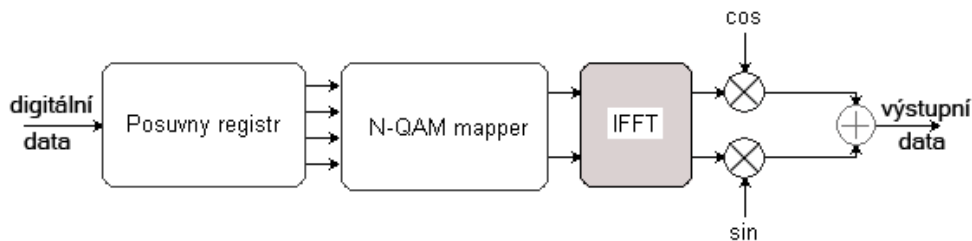
```

when "0011" => out_Q <= (st6);
...
when others => out_Q <= (st8);
end case;
case vstup (3 downto 0) is
when "0000" => out_I <= (st_8);
when "0001" => out_I <= (st_7);
when "0011" => out_I <= (st_6);
...
when others => out_I <= (st_8);
end case;
xn_real <= x_re;
xn_imag <= x_im;
...

```

7.2 Blok IFFT

Pro realizaci *OFDM* byla použita 16-ti bodová *IFFT*. Na vstup přivedeme komplexní signál formátu Q15. 16-ti bodová *IFFT* byla záměrně zvolena pro snadné pochopení a ozřejmění funkce tohoto bloku. Výstupem bloku *IFFT* je komplexní signál modulovaný na 16 subnosných básových funkcí (str. 15).



Obr. 7.4: Blokové schéma - IFFT

7.2.1 Knihovna IFFT

Jako blok *IFFT* je v programu ISE využita předprogramovaná knihovna *Fast Fourier Transform v3.2*, jež je součástí programové sady nástrojů *IP cores*.

Základní vlastnosti knihovny *Fast Fourier Transform v3.2* [18] :

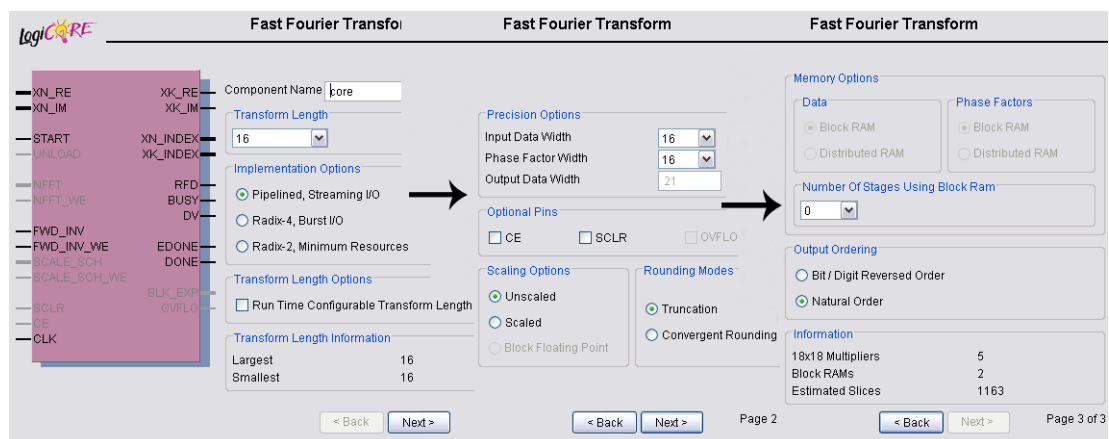
- Využit algoritmus výpočtu Cooley-Tukey (2.5.1)
- Možnost přímé i inverzní FFT
- Velikost transformace 2^3 až 2^{16}
- Výstupní hodnoty možno korigovat scale faktorem
- Možnost výběru vnitřní architektury - pipelined, radix-2, radix-4
- Možnost kontinuálního provozu - akvizice i transformace
- Možno využít pro Virtex II, Virtex II Pro, Spartan 3, Virtex 4

Pro náš účel využijeme 16-ti bodovou *IFFT* využívající architektury pipelined, výstupní hodnoty zůstanou neváženy (unscaled) a v normálním pořadí. Zde je nutno podotknout, že jiné nastavení než právě zmíněné nefungovalo korektně a odsimulované hodnoty programem Matlab se tak lišily od očekávaných hodnot.

Při výpočtu FFT je zřejmě využita metoda DIT (kap. 2.5.2). Bližší prozkoumání této blokové struktury až na úroveň hradel bohužel nebylo možné, patně z důvodu utajení použitých firemních algoritmů.

7.2.2 Simulace bloku IFFT

Simulace navrženého systému v časové oblasti vyžaduje využití programu ModelSim distribuovaného jako softwarová podpora programátoru ISE. Na základě vstupních hodnot signálů zadaných pomocí naprogramovaného testovacího vektoru jsou vyhodnoceny výstupní signály. Nezbytné knihovny by měly být do simulátoru implementovány automaticky (zejména Xilinxcore a Unisim [17]); u starších verzí nutno knihovny doplnit. Blok *IFFT* požaduje nastavení i některých inicializačních signálů, které jsou nezbytné pro jeho funkci. Tato inicializace byla provedena v rámci VHDL kódu, jelikož externí ovládání těchto signálů je téměř nemožné. V našem případě je použit formát Q15 neboli Q0.15 a tudíž reprezentace čísel v intervalu $< -1; 1 >$ (kap. 3.2). Na výstupu se však objeví signál formátu Q20, což je způsobeno nastavením jádra na neváženou logiku výpočtu. Nyní přistoupíme k samotné simulaci. Výstupní signály bloku *IFFT* jsou znázorněny v přílohách (obr. A.2 a A.3). Jelikož průběh transformace trvá řádově $100\mu s$ je nemožné rozumně zobrazit celý průběh. Na přiloženém datovém médiu je dostupný podrobnější detail `./img/celkovy_pohled.bmp` popřípadě `./img/simulace-wb.png`.



Obr. 7.5: Nastavení jádra Fast Fourier Transform v3.2

Výsledky simulace

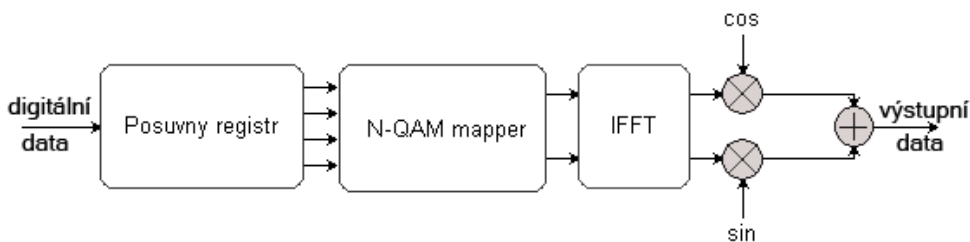
Při simulaci bylo zjištěno, že v důsledku časového zpoždění daného inicializačního bloku *IFFT* je nutné data přivést také časově opožděné. Za vstupní data, která jsou zahrnuta do transformace tak můžeme považovat všechny symboly přivedené s nástupnou hranou a signalizačním vektorem `xn_index` s hodnotou "0011" včetně. Následující symboly již budou načteny automaticky, jelikož je architektura jádra nastavena na kontinuální provoz (pipelined). Analýzou výsledků (tab.7.1) je zřejmé, že blok *IFFT* pracuje korektně a výsledky simulací se s očekávanými hodnotami shodují. Předpokladem správné funkce je správné nastavení parametrů jádra *IFFT* (str. 37).

Tab. 7.1: Výsledky simulace jádra IFFT

Vstupní sekvence dat									
111011011110010111010100110000011101100101010000011001000010000									
Vstupní data ISE				Vystup bloku IFFT - ModelSim				Matlab	
real part	imaginary part	decimal		real part	imaginary part	decimal		zpětně po FFT	
0111111111111111	0010101010101010	0,99997 + 0,33331i		00001111111111110111	000000101010101001101	0,12498 + 0,041651i		0,99997 + 0,33331i	
0111111111111111	1101010101010101	0,99997 - 0,33334i		00110000111010111000	0001000011100110101111	0,38219 + 0,27817i		0,99994 - 0,33334i	
0111111111111111	1000000000000000	0,99997 - 1i		00100010001101010101	111110011000111111101	0,26725 - 0,10059i		0,99995 - 1i	
0010101010101010	0111111111111111	0,33331 + 0,99997i		00001000111110011100	111100001100010111000	0,070122 - 0,23793i		0,33332 + 0,99997i	
0010101010101010	0010101010101010	0,33331 + 0,33331i		00001111111111111111	000011010101010101001	0,125 + 0,20833i		0,33331 + 0,33331i	
0010101010101010	1101010101010101	0,33331 - 0,33334i		11100010000001011000	11111011110000000000001	-0,23421 - 0,066404i		0,33334 - 0,33334i	
0010101010101010	1000000000000000	0,33331 - 1i		11100100000010100101	00000110011100000000001	-0,21844 + 0,10059i		0,33332 - 0,99997i	
1101010101010101	0111111111111111	-0,33334 + 0,99997i		1111001011110000000001	0000110100111000010001	-0,10254 + 0,20656i		-0,33336 + 0,99999i	
1101010101010101	0010101010101010	-0,33334 + 0,33331i		00001111111111111111	11111000000000000000001	0,125 - 0,125i		-0,33334 + 0,33331i	
1101010101010101	1101010101010101	-0,33334 - 0,33334i		00010001101010101000	000001111111001010111	0,13802 + 0,12419i		-0,33335 - 0,33334i	
1101010101010101	1000000000000000	-0,33334 - 1i		000100110001111111101	000000010001101011001	0,14941 + 0,01726i		-0,33332 - 1i	
1000000000000000	0111111111111111	-1 + 0,99997i		0000100010110111000	11111010000010011010	0,096542 - 0,046581i		-1 + 0,99995i	
1000000000000000	0010101010101010	-1 + 0,33331i		11010000000000000011	0000110101010101001	-0,37499 + 0,20833i		-1 + 0,33331i	
1000000000000000	1101010101010101	-1 - 0,33334i		000001100000011110000	111111111101010100001	0,047333 - 0,0026226i		-0,99999 - 0,33334i	
0111111111111111	0111111111111111	0,99997 + 0,99997i		00010001010010110001	111111101110010101001	0,1351 - 0,017256i		0,99996 + 0,99994i	
0111111111111111	1101010101010101	0,99997 - 0,33334i		00100010011101010011	111011111010011111101	0,2692 - 0,25538i		0,99997 - 0,33335i	

7.3 RF modulátor

Tento blok realizuje vnější modulaci na nosné funkce $\sin(x)$ a $\cos(x)$. RF modulátor je sestaven z bloků 18-ti bitových binárních násobiček a jedné binární sčítačky.

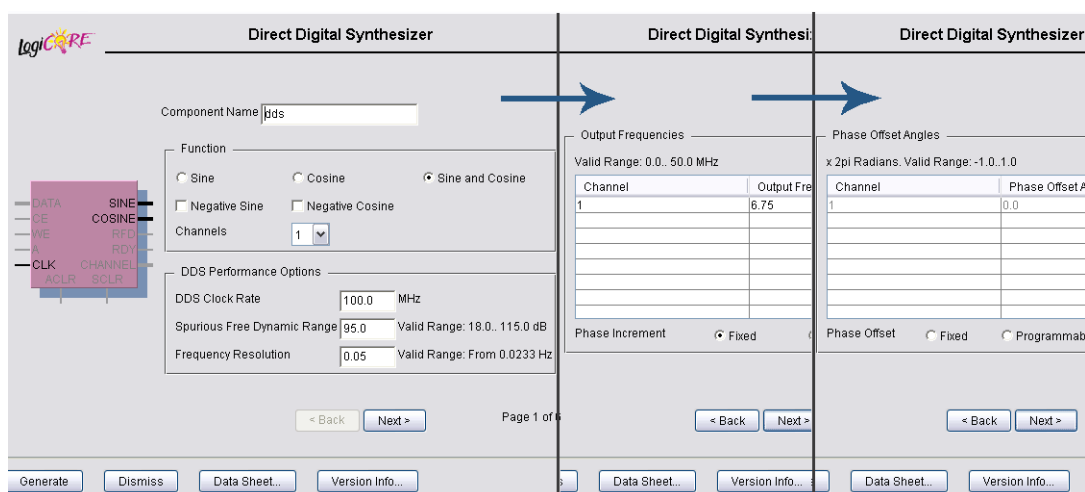


Obr. 7.6: Blokové schéma - RF modulátor

Perioda použitých nosných vln byla zvolena tak, aby nedocházelo k ořezu na straně výstupních filtrů typu DP modulu P-160.

7.3.1 DDS

Pro blok generátoru funkcí $\sin(x)$ a $\cos(x)$ bylo použito jádro *DDS ver.5* (Direct Digital Synthesizer) z knihoven *IP core*. Pro vhodné nastavení doporučuji nastavení na obr. 7.7, které bylo experimentálně ověřeno, aby vyhovělo zadaným požadavkům. Jedná se zejména o vhodnou frekvenci respektující výstupní filtry typu DP a dále o vhodnou hodnotu inkrementace a offsetu fází. Blok násobiček realizuje násobení těchto vygenerovaných funkcí s výstupním signálem bloku *IFFT*.



Obr. 7.7: Nastavení jádra DDS v5.0

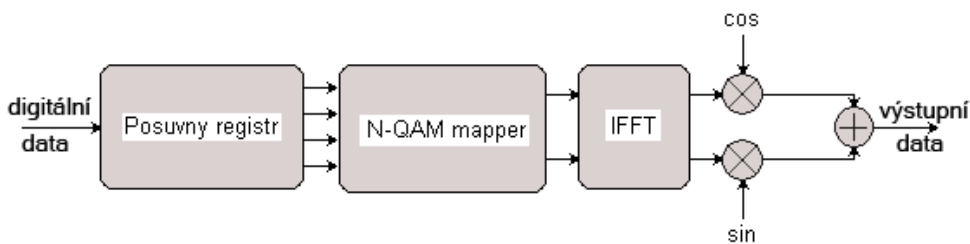
7.3.2 Násobička

Při nastavování násobičky použijeme délku symbolů 16-ti bitů a ponecháme znaménkový MSB bit (signed). Po vynásobení tak obdržíme vektor délky 32 bitů, který však obsahuje dva znaménkové MSB bity, které jsou totožné. Ořízneme proto redundantní první znaménkový bit posloupnosti a dále zkrátíme vektor na původní délku 16-ti bitů. Experimentálně bylo ověřeno, že oříznutím

posledních 15-ti bitů se poměr SNR zhorší maximálně o 30dB. Následuje sečtení těchto orthogonálních signálů v bloku sčítačky. Sčítačka je realizována pomocí knihovny *IP core*. Délku symbolů nastavíme na 16 bitů a použijeme MSB znaménkový bit (signed). Takto zpracovaný signál namodulovaný na frekvenci orthogonálních nosných vln je připraven na přenos komunikačním kanálem (popř. možno dále modulovat na vyšší frekvenci).

7.4 OFDM modem - celková simulace

Abychom ověřili funkci navrženého modemu, musíme ověřit korektní funkci všech použitých bloků před závěrečnou implementací. Funkce jádra *IFFT* byla ověřena v kap. 7.2.2, nyní tedy přistoupíme ke kompletní simulaci celé části obvodu. Při počítačových simulacích nebudeme ověřovat pouze činnosti AD a DA převodníků.



Obr. 7.8: Blokové schéma - kompletní OFDM modem

Do simulace jsou zahrnuty bloky : *posuvný registr*, *DCM*, *QAM mapper*, *IFFT DDS*, *násobičky* a *sčítačka*. Na vstup je zaveden unipolární linkový signál NRZ, ten je v bloku posuvného registru vhodně zpožděn a rozdělen na bity reprezentující symbol. Plní zde funkci sériově-paralelního převodníku. Pro názornost je zde použita modulace *16-QAM* a simulace všech 16-ti signálových stavů. Jako dělič kmitočtu z původního f_{clk} je zde použit blok *DCM* (Digital Clock Manager), který poskytuje veškeré potřebné hodinové takty. Výstupem *QAM – mapperu* jsou komplexní signály *I* a *Q* délky 16 bitů reprezentující symbol (viz. 7.3). Abychom tyto signály mohli dále použít, musíme provést transformaci do kódu *Q15* požadovaného blokem *IFFT* (viz. tab. 3.2). Princip transformace *IFFT* byl popsán v kap. 2.4. Výstupem *IFFT* jsou 2 komplexní ortogonální signály o délce 21 bitů, přičemž první bit je redundantní. Následně je provedeno násobení s funkcemi $\sin(x)$ a $\cos(x)$. Po vynásobení dvou 20-ti bitových signálů vznikne signál délky 40-ti bitů, kde první bit je opět redundantní. Dostatečnou přesnost však zaručí i prvních platných 20 bitů, které odečteme s druhou větví v bloku sčítačky. Popsaná činnost je zřejmá z obrázků uvedených v příloze (A.2, A.3 a A.4) popřípadě z dále uvedených tabulek 7.2 a 7.3. V těchto tabulkách jsou vyexportována data ze simulací provedených v ModelSimu. Tyto data jsou transformována do dekadické podoby, abychom mohli jednoduše ověřit jejich správnost. V posledním sloupci je provedeno porovnání od-simulované hodnoty v ModelSimu a referenční hodnoty získané výpočtem v Matlabu. Chyby bloků modemu získané srovnáním s daty získanými programem Matlab formou matematických výpočtů použitých funkcí je uvedeno v poslední tabulce 7.3 a je uvedena získaná přesnost výpočtu. Tato přesnost udává v případě simulace *IFFT* nepřesnosti použitého algoritmu v DSP, zejména chyby zaokrouhlování. Jako referenční hodnoty jsou považovány hodnoty získané 16-ti bodovou transformací vstupních hodnot příkazem $fft(x,16)$. Následující chyby bloků násobičky a sčítačky jsou způsobeny zejména ztrátou několika posledních bitů v důsledku praktické realizace modemu používajícího 12-ti bitové DA převodníky. Výsledná chyba OFDM modemu je tak způsobena zejména chybou ořezu celkového počtu bitů na požadovaných 12 bitů a pohybuje se v řádu tisícín (tab. 7.3). V tabulkách jsou výstupní signály jednotlivých větví uvedeny jako komplexní signál, kde reálná část reprezentuje větev *I* a násobení cosinem a imaginární část reprezentuje větev *Q* a násobení sinem. Toto vyjádření bylo zvoleno pouze za účelem snadného zápisu a může být trochu matoucí.

Tab. 7.2: Výsledky simulace 2

Vstupní data	
1111111011011100101110101010001110110010101000011001000010000	

QAM				IFFT	
realna část I	imaginarní část Q	dekadicky I+j*Q	realna část	imaginarní část	dekadicky
011111111111111	011111111111111	1.0000 + 1.0000i	11111111111111000	111111111111111000	0,0000+0,0000i
011111111111111	00101010101010	1.0000 + 0.3333i	000010101010101001	001010110011111011	0,0833+0,4189i
011111111111111	1101010101010101	1.0000 - 0.3333i	000010101010101001	0001100111000000010	0,0833+0,2011i
011111111111111	1000000000000000	1.0000 - 1.0000i	000010101010101001	0000111111101100111	0,0833+0,1247i
0010101010101010	011111111111111	0.3333 + 1.0000i	1101010101010101000	001010101010101000	-0,3333+0,3333i
0010101010101010	0010101010101010	0.3333 + 0.3333i	00001010101010101010	0000011001000001010	0,0833+0,0556i
0010101010101010	1101010101010101	0.3333 - 0.3333i	00001010101010101010	0000010001101011111	0,0833+0,0345i
0010101010101010	1000000000000000	0.3333 - 1.0000i	00001010101010101010	0000001000011110110	0,0833+0,0165i
1101010101010101	011111111111111	-0.3333 + 1.0000i	0000000000000000000	00101010101010101000	0,0000+0,3333i
1101010101010101	0010101010101010	-0.3333 + 0.3333i	00001010101010101011	1111101111000001011	0,0833-0,0165i
1101010101010101	1101010101010101	-0.3333 - 0.3333i	00001010101010101011	11111011100101010010	0,0833-0,0345i
1000000000000000	011111111111111	-1.0000 + 1.0000i	00001010101010101011	1111100011011110111	0,0833-0,0556i
1000000000000000	0010101010101010	-1.0000 + 0.3333i	00101010101010101000	00101010101010101000	0,3333+0,3333i
1000000000000000	0010101010101010	-1.0000 + 0.3333i	00001010101010101010	1111000000010011000	0,0833-0,1247i
1000000000000000	1101010101010101	-1.0000 - 0.3333i	00001010101010101010	1110011000111111101	0,0833-0,2011i
1000000000000000	1000000000000000	-1.0000 - 1.0000i	00001010101010101010	11001010011000000100	0,0833-0,4189i

DDS				násoбіčka	
sinus	cosinus	dekadicky	I*cosinus	Q*sinus	dekadicky I*cos+j*sin
01110010000011000111	00111010000111000111	0,8910+0,4539i	111111111111111000	111111111111111100	0,0000+0,0000i
10111000000011011101	0110100111011011110	-0,5621+0,8271i	11111010000000010011	00101100010110011111	-0,0468+0,3464i
100111111111000101	101010101010100100	-0,7500-0,6613i	1111011111111111100	1110111011110001001	-0,0625-0,1330i
011000000000011011	10101011010110100010	0,7501-0,6613i	00001000000000000011	11110101011100011001	0,0625-0,0824i
0100011111100100111	0110100111011011111	0,5621+0,8271i	11101000000001001001	0010001010010011101	-0,1873+0,275i
1000110111100110111	00111010000111000111	-0,8910+0,4539i	11110110011111101111	00000011001111000101	-0,0742+0,0252i
11010100101001000101	10000111100100010011	-0,3387-0,9408i	11111100011000110000	11110111110101111110	-0,0282-0,0324i
0111110011101010101	1110100001001111111	0,9759-0,2181i	00001010011010001101	11111111100010010111	0,0813-0,0036i
00001100000010111010	0111111101101101011	0,0941+0,9955i	00000000000000000000	00101010011110100001	0,0000+0,3318i
1000000000100000100	1111101111110101100	-0,9995-0,0314i	111101010101010101010	00000000000100010001	-0,0832+0,0005i
0001010000001100000	10000001100100110110	0,1564-0,9876i	00000001101010110010	00000100010111001111	0,0130+0,0340i
01111010111010110001	00100011101101011110	0,9603+0,2789i	00001010001111100100	11111110000001011111	0,0800-0,0155i
11001101001010100100	0111010101110010000	-0,3971+0,9177i	111101111000011100010	00100111001010000010	-0,1323+0,3059i
10010001110100110001	10111101101010111110	-0,8607-0,5090i	11110110110100011001	00001000001000000010	-0,0717+0,0634i
0100111001110011011	10011010110111000011	0,6129-0,7901i	00000110100010011001	00010100010110001101	0,0510+0,1589i
01011010100000101000	01011010100000101001	0,7071+0,7071i	0000011111000101010101	1101101000001010011111	0,0589-0,2962i

Tab. 7.3: Výsledky simulace 2 - ověření funkce IFFT

QAM přesně	po provedení IFFT	zpětně po provedení FFT	rozdíl $\cdot 10^{-4}$
1.000000000000 + 1.000000000000i	-0.000015258789 - 0.000015258789i	0.999969482421 + 0.999969482421i	0.30517578125 + 0.30517578125i
1.000000000000 + 0.333333333333i	0.083330154418 + 0.418935775756i	0.999942800968 + 0.333312988281i	0.57199031975 + 0.20345052083i
1.000000000000 - 0.333333333333i	0.083330154418 + 0.201175689697i	0.999948005304 - 0.333343505859i	0.51994695879 + 0.10172526041i
1.000000000000 - 1.000000000000i	0.083330154418 + 0.124708175659i	0.999971603315 - 0.999990476204i	0.28396684423 - 0.09523795176i
0.333333333333 + 1.000000000000i	-0.333328247070 + 0.333328247070i	0.333312988281 + 0.999969482421i	0.20345052083 + 0.30517578125i
0.333333333333 + 0.333333333333i	0.083332061767 + 0.055683135986i	0.333337852090 + 0.333312988281i	-0.04518757415 + 0.20345052083i
0.333333333333 - 0.333333333333i	0.083332061767 + 0.034513473510i	0.333291511163 - 0.333343505859i	0.41822169837 + 0.10172526041i
0.333333333333 - 1.000000000000i	0.083332061767 + 0.016582489013i	0.333323147026 - 0.999995426512i	0.10186307230 - 0.04573487729i
-0.333333333333 + 1.000000000000i	0 + 0.333328247070i	-0.333343505859 + 0.999969482421i	0.10172526041 + 0.30517578125i
-0.333333333333 + 0.333333333333i	0.083333969116 - 0.016580581665i	-0.333343505859 + 0.999969482421i	0.15757783164 + 0.20345052083i
-0.333333333333 - 0.333333333333i	0.083333969116 - 0.034511566162i	-0.333322028741 - 0.333343505859i	-0.11304591712 + 0.10172526041i
-0.333333333333 - 1.000000000000i	0.083333969116 - 0.055681228637i	-0.333367103870 - 0.99999734201i	0.33770537497 - 0.01265798042i
-1.000000000000 + 1.000000000000i	0.333328247070 + 0.333328247070i	-1.000000000000 + 0.999969482421i	0 + 0.30517578125000i
-1.000000000000 + 0.333333333333i	0.083332061767 - 0.124710083007i	-0.99992597098 + 0.333312988281i	-0.07402901474 + 0.20345052083i
-1.000000000000 - 0.333333333333i	0.083332061767 - 0.201177597045i	-0.999978522882 - 0.333343505859i	-0.21477117754 + 0.10172526041i
-1.000000000000 - 1.000000000000i	0.083332061767 - 0.418937683105i	-0.999988681627 - 1.000015363080i	-0.11318372901 + 0.15363080948i

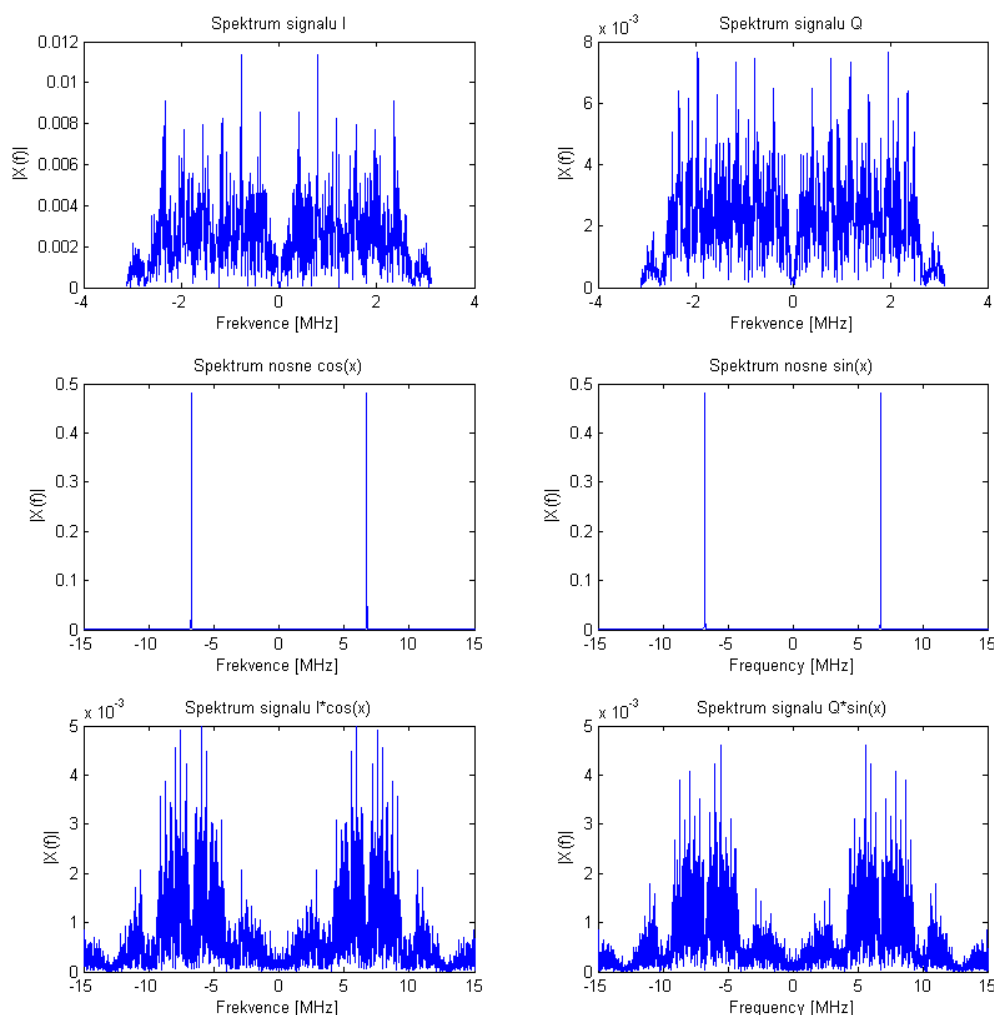
výstupy IFFT (20 bitů)	výstupy násobiček (40 bitů)	výstupy násobiček (12 bitů)	rozdíl $\cdot 10^{-3}$
-0.00001525878906 - 0.00001525878906i	-0.00048828125000 - 0.00048828125000i	-0.00001359565067 - 0.00000692738104i	0.47468559932895 + 0.48135386896320i
0.08333015441895 + 0.41893577575684i	-0.04687500000000 + 0.34619140625000i	-0.04683808619666 + 0.34649490440643i	0.03691380334203 + 0.30349815642694i
0.08333015441895 + 0.20117568969727i	-0.06298828125000 - 0.13330078125000i	-0.06250699325392 - 0.13303915105644i	0.48128799608094 + 0.26163019356318i
0.08333015441895 + 0.12470817565918i	0.06250000000000 - 0.08251953125000i	0.06250699325392 - 0.08247102594032i	0.00699325391906 + 0.04850530967815i
-0.33332824707031 + 0.33332824707031i	-0.18750000000000 + 0.27539062500000i	-0.18735917619779 + 0.27569095822400i	0.14082380221225 + 0.30033322400413i
0.08333206176758 + 0.05568313598633i	-0.07470703125000 + 0.02490234375000i	-0.07424956511386 + 0.02527974524855i	0.45746613614028 + 0.3774014985538i
0.08333206176758 + 0.03451347351074i	-0.02832031250000 - 0.03271484375000i	-0.02822753431246 - 0.03247309663493i	0.09277818753617 + 0.24174711506930i
0.08333206176758 + 0.01658248901367i	0.08105468750000 - 0.00390625000000i	0.08132508479321 - 0.00361732995225i	0.27039729320677 + 0.28892004775116i
0 + 0.33332824707031i	0 + 0.33154296875000i	0 + 0.33185007443535i	0 + 0.30710568535142i
0.08333396911621 - 0.01658058166504i	-0.08349609375000 + 0.00048828125000i	-0.08329264291388 + 0.00052079967281i	0.20345083612483 + 0.03251842281315i
0.08333396911621 - 0.03451156616211i	0.01269531250000 + 0.03369140625000i	0.01303619157989 + 0.03408672781370i	0.34087907988578 + 0.39532156370115i
0.08333396911621 - 0.05568122863770i	0.07958984375000 - 0.01562500000000i	0.080025448872537 - 0.01533438818519i	0.43564497536863 + 0.09061181481229i
0.33332824707031 + 0.33332824707031i	-0.13281250000000 + 0.30566406250000i	-0.13238069077488 + 0.30591370770708i	0.43180922511965 + 0.24964520707726i
0.08333206176758 - 0.12471008300781i	-0.0717734375000 + 0.06347056250000i	-0.07172777080996 + 0.06348204874666i	0.04957294004271 + 0.00548624666408i
0.08333206176758 - 0.20117759704590i	0.05078125000000 + 0.15869140625000i	0.05107467917696 + 0.15896156683084i	0.29342917696340 + 0.27016058083973i
0.08333206176758 - 0.41893768310547i	0.05859375000000 - 0.29638671875000i	0.05892472952837 - 0.2962347952706i	0.33097952837124 + 0.15192352293525i

7.5 Celková simulace spektra

Jelikož nebyl v průběhu návrhu dostupný OFDM demodulátor, zůstává spektrální analýza jedinou možnou variantou ověření korektní činnosti implementovaného obvodu. Za účelem možného srovnání byly proto simulovány jednotlivá spektra a vyhodnoceny výsledky.

7.5.1 Simulované spektrum

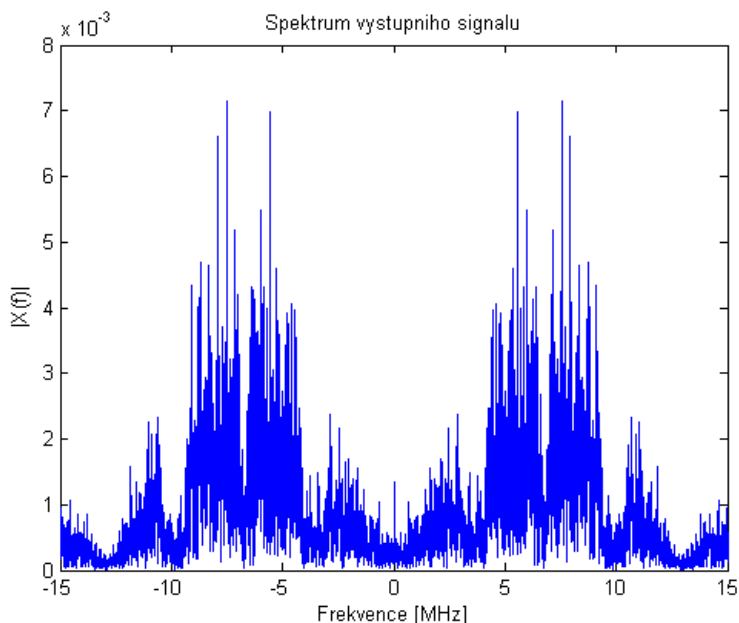
V této části provedeme spektrální analýzu relativně dlouhého úseku dat délky téměř $10ms$ vytvořených generátorem pseudonáhodných posloupností *LFSR* (Linear Feedback Shift Register), a odsimulovaných programem ModelSim. Tento úsek je relativně dlouhý vzhledem k době, kterou simulace trvá v reálném čase. Parametry jednotlivých bloků zůstaly stejné jako v předchozí simulaci, včetně nulových subnosných č. 0, 7, 8, 9 (obr. 4.7). Data získaná simulací je nutné exportovat v ModelSimu do textové podoby a pomocí naprogramovaných skriptů převést do dekadické podoby a analyzovat. Proces zpracování těchto dat již probíhá v programu Matlab. Do simulace jsou zahrnuty bloky *LFSR*, posuvný registr, *DCM*, *QAM mapper*, *IFFT*, *DDS*, násobička a sčítačka. Odsimulovaná spektra jsou uvedena na obr. 7.9 a obr. 7.10.



Obr. 7.9: Simulace spektra OFDM modemu - část 1

Parametry simulace :

Generátor náhodných čísel je nastaven na generování Fibonacciho posloupností maximální délky 168 bitů (generující polynom $x^{168} + x^{17} + x^{15} + x^2 + x^0$). Hodinový signál generátoru odpovídá frekvenci $f_b = 50\text{MHz}$, která byla získána pomocí bloku DCM využívajícího referenční frekvenci $f = 100\text{MHz}$. Tento blok dále generuje hodinové kmitočty potřebné pro další části obvodu, zejména symbolovou frekvenci $f_s = 6,25\text{MHz}$ a hodinové signály pro převodníky modulu P160.



Obr. 7.10: Simulace spektra OFDM modemu - část 2

Poznámky

Simulace byla realizována programem Modelsim v délce téměř 10ms a poskytla zhruba 55 000 výstupních hodnot, tudíž zhruba 3 500 OFDM symbolů. Pro další analýzu byly signálová data exportována do textové podoby a dále transformována pomocí naprogramovaných skriptů pro převod do dekadické podoby. Takto zpracovaná data již mohla být jednoduše podrobena spektrální analýze. Veškerá použitá data jsou dostupná na datovém médiu `./vhdl/lfsr/`, včetně použitých skriptů pro převod Q-formátu do dekadické podoby `./matlab/q2vect.m` a pro načtení dat ze souboru `*.txt` `./matlab/load.m`. Popis funkce skriptů je uveden v hlavičce m-souboru.

Závěry vyplívající ze simulací :

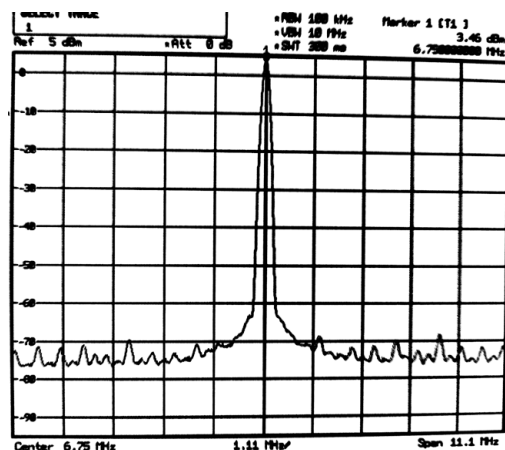
Paralelně byla odsimulována náhodná data dostatečného rozsahu v programech Modelsim a Matlab, aby bylo možné vyvodit obecné spektrální charakteristiky. Simulace potvrdily, že obě metody vykazují spektrum charakteristické pro OFDM modem s šířkou pásma přibližně 6MHz. V naprogramovaném VHDL kódu nebylo použito eliminace ISI formou cyklického prefixu, jelikož jeho doprogramování by bylo neúměrně složité a jeho funkce je již implementována v nové knihovně FFT core v5.0, která je dostupná v programátoru ISE 9.2i SP3 a vyšším. Jelikož však bylo při realizaci této práce využito programátoru 9.1i, nebyla zmíněná knihovna dostupná.

8 PRAKTICKÉ MĚŘENÍ

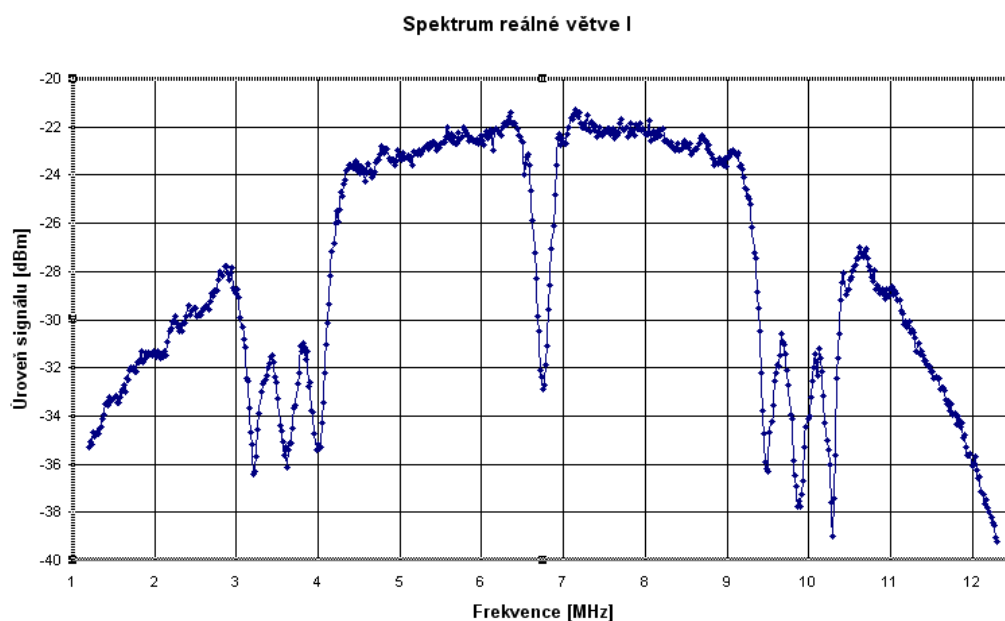
Navržený obvod byl implementován do vývojové desky V2MB1000 a bylo pozorováno výstupní spektrum jednotlivých funkčních komponentů. Spektrální analýza OFDM modemu byla zvolena za účelem ověření funkce záměrně, jelikož nebyl dostupný speciální analyzátor OFDM signálu.

8.1 Změřená spektra

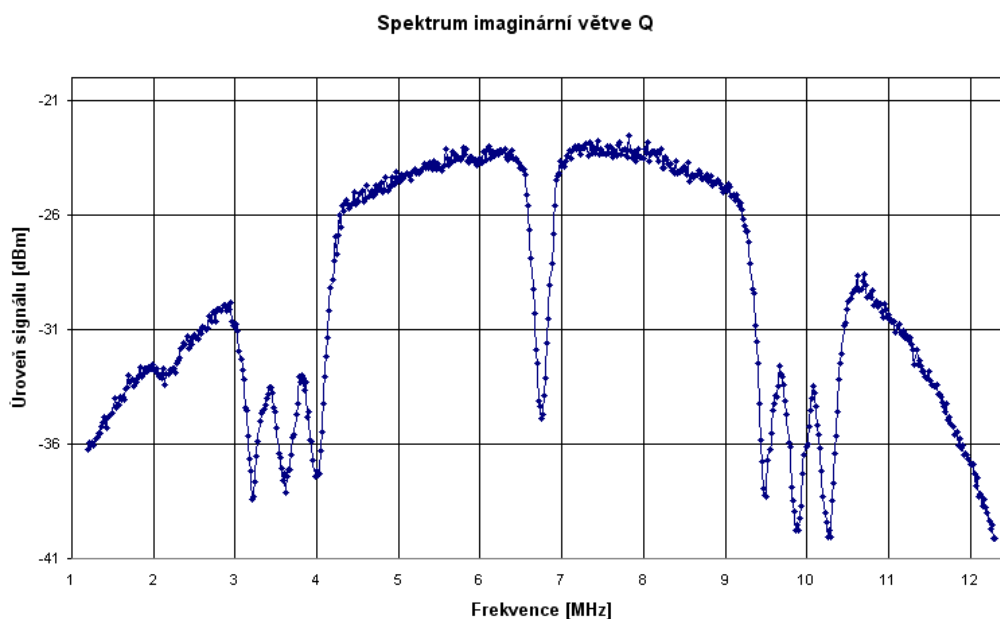
Obvod OFDM modemu byl analyzován na spektrální analyzátoru a jednotlivá spektra jsou uvedena na několika následujících obrázcích 8.1, 8.2, 8.3 a 8.4.



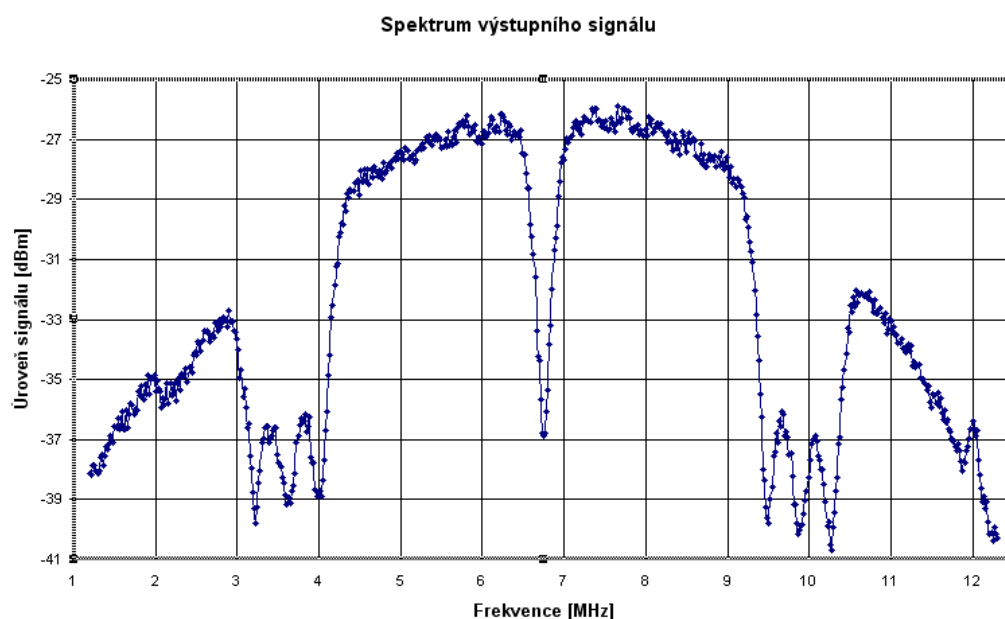
Obr. 8.1: Spektrum signálu $\cos(x)$ generovaného blokem DDS



Obr. 8.2: Spektrum signálu $I \cdot \cos(x)$



Obr. 8.3: Spektrum signálu $Q \cdot \sin(x)$



Obr. 8.4: Spektrum výstupního signálu $I \cdot \cos(x) - Q \cdot \sin(x)$

Závěr:

Průběhy spekter zobrazené na analyzátoru odpovídali předpokládaným průběhům a tedy i oběma předešlým simulacím. Je zde patrný velký vliv výstupních filtrů DP modulu P160, který charakteristiku na vyšších kmitočtech potlačuje ačkoliv je jeho mezní kmitočet $f_{-3dB} = 28,4 MHz$. Ve všech průbězích spekter jsou patrné vložené nuly na daných subnosných kmitočtech. Uprostřed spektra tak můžeme vidět nulovou stejnosměrnou složku, kterou bychom ani nedokázali přenést vzhledem k parametrům výstupních filtrů a další nuly sloužící k potlačení ramen spektra OFDM. Jako základní referenční kmitočet byl použit interní krystalový oscilátor desky V2MB1000 pra-

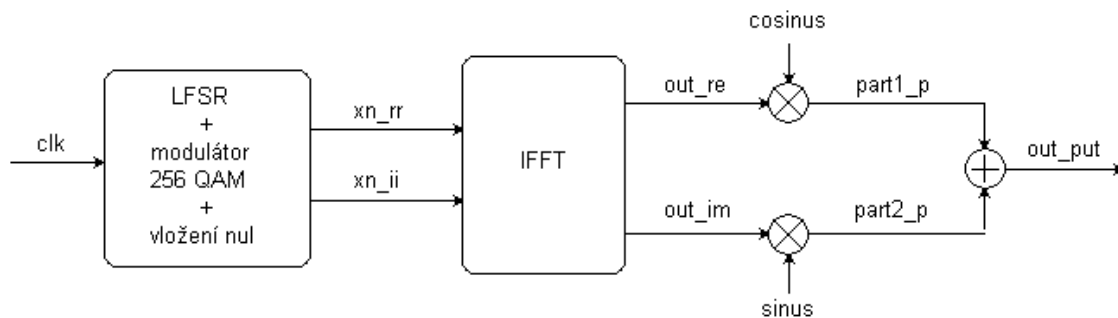
cující na frekvenci 100MHz. Zapojení jednotlivých pinů je vysvětleno v kap. 6.2. Pro případnou implementaci doporučuji opětovnou rekompilaci VHDL kódu, jelikož při pouhém naprogramování pomocí již dříve vygenerovaných souborů obvod neplnil požadovanou funkci. Zdrojové kódy mohou být nalezeny na datovém médiu ve složce `./vhdl/lfsr/`

8.2 Tréninková sekvence

Pro budoucí konstrukci OFDM receiveru byl za účelem ověření jeho správné funkce naprogramován OFDM modem do funkce generátoru tréninkové sekvence. Funkci generátoru přesně definované sekvence dat tu přebírá blok 2-portové adresované paměti, do níž jsou data načtena ze souboru `qam.coe`. Jako generátor adres poslouží signál bloku IFFT udávající index subnosných kmitočtů. Tato tréninková sekvence obsahuje bity $(00 - 91 - B3 - A2 - E6 - F7 - D5 - 00 - 00 - 00 - 5D - 7F - 6E - 2A - 3B - 19)_{hex}$, které se neustále opakují s periodou OFDM symbolu. Tyto bity jsou samozřejmě zakódovány modulací *256-QAM*, převedeny do kódu *Q15*, přivedeny na blok *IFFT* a nakonec v bloku *rf modulátoru* kvadraturně namodulovány na nosnou vlnu s kmitočtem 6,75 MHz. Takto vytvořené OFDM symboly se poté objeví na výstupu DA převodníku modulu P160. Zdrojový kód dané funkce může být nalezen na datovém médiu ve složce `./vhdl/training/`.

8.3 Shrnutí

Závěrem bych uvedl některé komplikace, které v průběhu návrhu nemusí být zcela zřejmé a způsobují nefunkčnost výsledné implementace. Jedná se zejména o nutnou konverzi formátů mezi jednotlivými funkčními bloky. Některé bloky IP core totiž pracují s číselnými formáty *Q15* vhodnými pro DSP a jiné bloky s nimi pracovat neumí. Je proto nutné tyto formáty vzájemně převádět a to jednoduše inverzí prvního MSB bitu. Další komplikací může být rozsah platných hodnot bitů na výstupu některých bloků. Zejména bloky IFFT a násobičky pracující se znaménkovým bitem v důsledku matematických úprav generují výstupní signál se dvěma znaménkovými bity, přičemž první bit nesmí být uvažován, aby nedocházelo k interpretaci nesmyslných výsledků. Nakonec je ještě třeba dbát na možné přetečení hodnoty v důsledku sečtení velkých čísel a toto přetečení ošetřit dělením vstupních hodnoty sčítačky na polovinu. Pro ilustraci funkce ještě uvedu několik časových průběhů získaných při simulaci OFDM modemu v ModelSimu. Z těchto průběhů A.2, A.3 a A.4 jsou patrné výstupní signály bloků QAM mapperu, IFFT, DDS a násobiček. Popsané signály jsou uvedeny v obr. 8.5.



Obr. 8.5: Popis výstupních signálů (viz. přílohy)

9 ZÁVĚR

Cílem této diplomové práce byl návrh funkčního modulátoru OFDM, který využívá HW prostředky FPGA obvodu vývojové desky V2MB1000 firmy Xilinx.

První část práce je věnována teoretické analýze funkce navrhovaného modemu. Je zde vysvětlena problematika modulací OFDM a jejich možné realizace. Větší pozornost je věnována algoritmu realizovanému pomocí FFT, který má podporu v DSP obvodech, včetně použitého FPGA. Dále byla provedena teoretická analýza možných problémů se kterými se můžeme při realizaci setkat a bylo naznačeno i možné řešení těchto komplikací. Jelikož jsou některé formáty binárních čísel specifické pro danou aplikaci, je další kapitola věnována těmto formátům a jejich konverzi do dekadické podoby. Největší úsilí a také největší rozsah je věnován kapitole simulací navrhovaného obvodu. Jelikož nebyl dostupný OFDM demodulátor, je tato forma jedinou možnou variantou ověření správné funkce OFDM modemu. Tento obvod byl podroben nejprve teoretickým analýzám OFDM signálu. Následně byly uvedeny některé charakteristiky obvodů FPGA distribuovaných firmou Xilinx. Jedná se však pouze o velmi stručný přehled dostupných FPGA obvodů a jejich parametrů, jelikož jsou neustále vyvíjeny obvody nové a tudíž se přehled rychle stává zastaralým. Další kapitola je věnována přímo použitému HW, tedy vývojové desce V2MB1000 a potřebným informacím, které slouží ke správné konfiguraci této desky. V této práci bylo snahou uvést všechny informace potřebné k úspěšné implementaci OFDM modemu a je doporučeno se těmito informacemi řídit. Následně byly provedeny simulace v simulačním programu určenému pro dané obvody a nakonec praktická analýza implementovaného obvodu na spektrálním analyzátoru. Tyto simulace tak mohou být vzájemně porovnány a průběhy dokazují správnost funkcí realizovaného modemu. Pro další použití realizovaného obvodu jsou na přiloženém médiu dostupné veškeré použité skripty umožňující obvod naprogramovat. Jsou dostupné 2 varianty modemu umožňující buď náhodné generování binárních čísel bez nutnosti zavádění jakýchkoli vstupních hodnot nebo zavedení vstupních hodnot formou datového souboru do paměťového bloku. Tento soubor je při inicializaci jádra zaveden paměťového registru a slouží jako generátor stále se opakující známé posloupnosti hodnot (tréninkové sekvence). Tato funkce je praktická pro následnou realizaci OFDM demodulátoru. Požadavky zadání práce tak byly splněny a výsledky praktického měření potvrdili správnost teoretických analýz.

LITERATURA

- [1] John G. Proakis, Masoud SALEHI: *Communication systems engineering (second edition)*, Prentice Hall, 2002
- [2] Ahmad R. Bahai, Burton R. Saltzberg : *Multi-Carrier Digital Communications - Theory and Applications of Ofdm*, Norwell, MA Kluwer, 1999
- [3] Ramjee Prasad : *OFDM for Wireless Communications Systems*, Artech House, 2004
- [4] Joaquin Garcia, Rene Cumplido : *Modulátor OFDM pro 802.11a v obvodu FPGA*, stručná zpráva z konference IEEE, 2005
- [5] Pinker Jiří, Poupa Martin : *Číslicové systémy a jazyk VHDL*, nakladatelství BEN, 2006
- [6] Standard pro jazyk VHDL : IEEE Std 1076 - 1993
- [7] Novelizace standardu VHDL : IEEE Std 1076 - 2000

Internetové odkazy :

- [8] The principles of OFDM [online] – URL : <<http://www.cs.vu.nl>>
- [9] Principles of OFDM - IEEE Potentials [online] – URL : <<http://ieeexplore.ieee.org>>
- [10] Modulace DMT [online] – URL : <<http://www.elektrorevue.cz>>
- [11] OFDM implementace v FPGA [online] – URL : <<http://hw.cz>>
- [12] Principles of COFDM [online] – URL : <<http://www.digitalradiotech.co.uk>>
- [13] Fourierova transformace [online] – URL : <<http://fs1.vsb.cz>>
- [14] Konvoluční kódér [online] – URL : <<http://fel.zcu.cz>>
- [15] Obvody PLD [online] – URL : <<http://hw.cz>>
- [16] The designers guide to VHDL [online] – URL : <<http://www.doulos.com/knowhow/vhdl>>
- [17] Stránky výrobce Xilinx Inc. [online] – URL : <<http://www.xilinx.com>>
- [18] Datasheety IP cores [online] – URL : <<http://www.xilinx.com>>
- [19] Fixed Point Arithmetic [online] – URL : <<http://www.fulton.asu.edu>>
- [20] Datasheet Virtex II – URL : <./podklady/virtex2.pdf>
- [21] Datasheet modulu P160 – URL : <./podklady/p160.pdf>
- [22] Datasheet vývojové desky V2MB1000 – URL : <./podklady/v2mb.pdf>

A PŘÍLOHY

A.1 Závěrečná zpráva z programátoru (Design summary)

DDSZK Project Status			
Project File:	ddszk.ise	Current State:	Programming File Generated
Module Name:	modul	• Errors:	No Errors
Target Device:	xc2v1000-4fg456	• Warnings:	256 Warnings
Product Version:	ISE 9.1.03i	• Updated:	st 21. V 10:07:29 2008

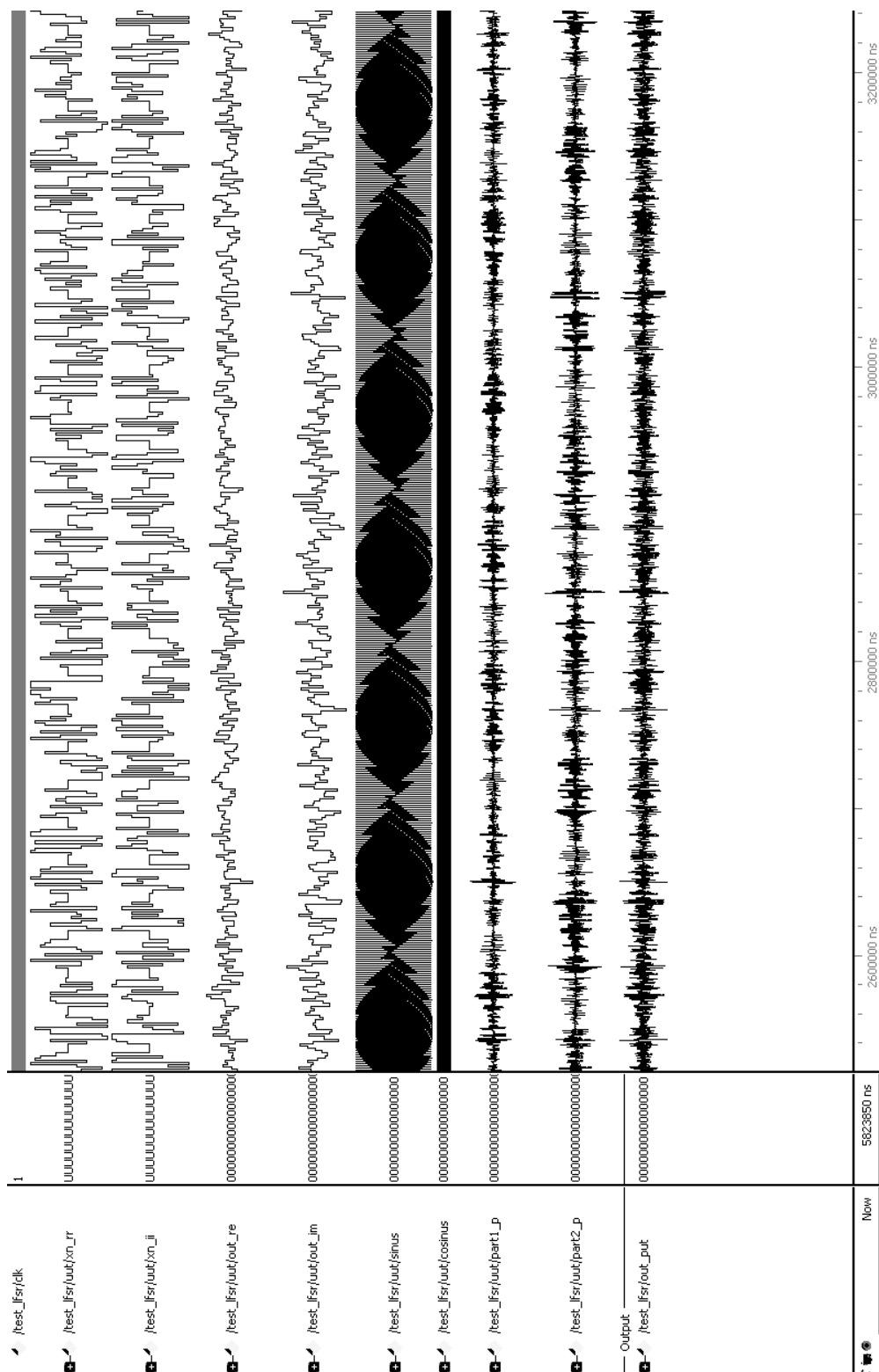
DDSZK Partition Summary
No partition information was found.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,882	10,240	18%	
Number of 4 input LUTs	1,576	10,240	15%	
Logic Distribution				
Number of occupied Slices	1,496	5,120	29%	
Number of Slices containing only related logic	1,496	1,496	100%	
Number of Slices containing unrelated logic	0	1,496	0%	
Total Number of 4 input LUTs	2,039	10,240	19%	
Number used as logic	1,576			
Number used as a route-thru	41			
Number used for Dual Port RAMs	128			
Number used as Shift registers	294			
Number of bonded IOBs	15	324	4%	
Number of Block RAMs	4	40	10%	
Number of MULT18X18s	4	40	10%	
Number of GCLKs	4	16	25%	
Number of DCMs	1	8	12%	
Number of RPM macros	1			
Total equivalent gate count for design	351,494			
Additional JTAG gate count for IOBs	720			

Performance Summary			
Final Timing Score:	0	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints Met		

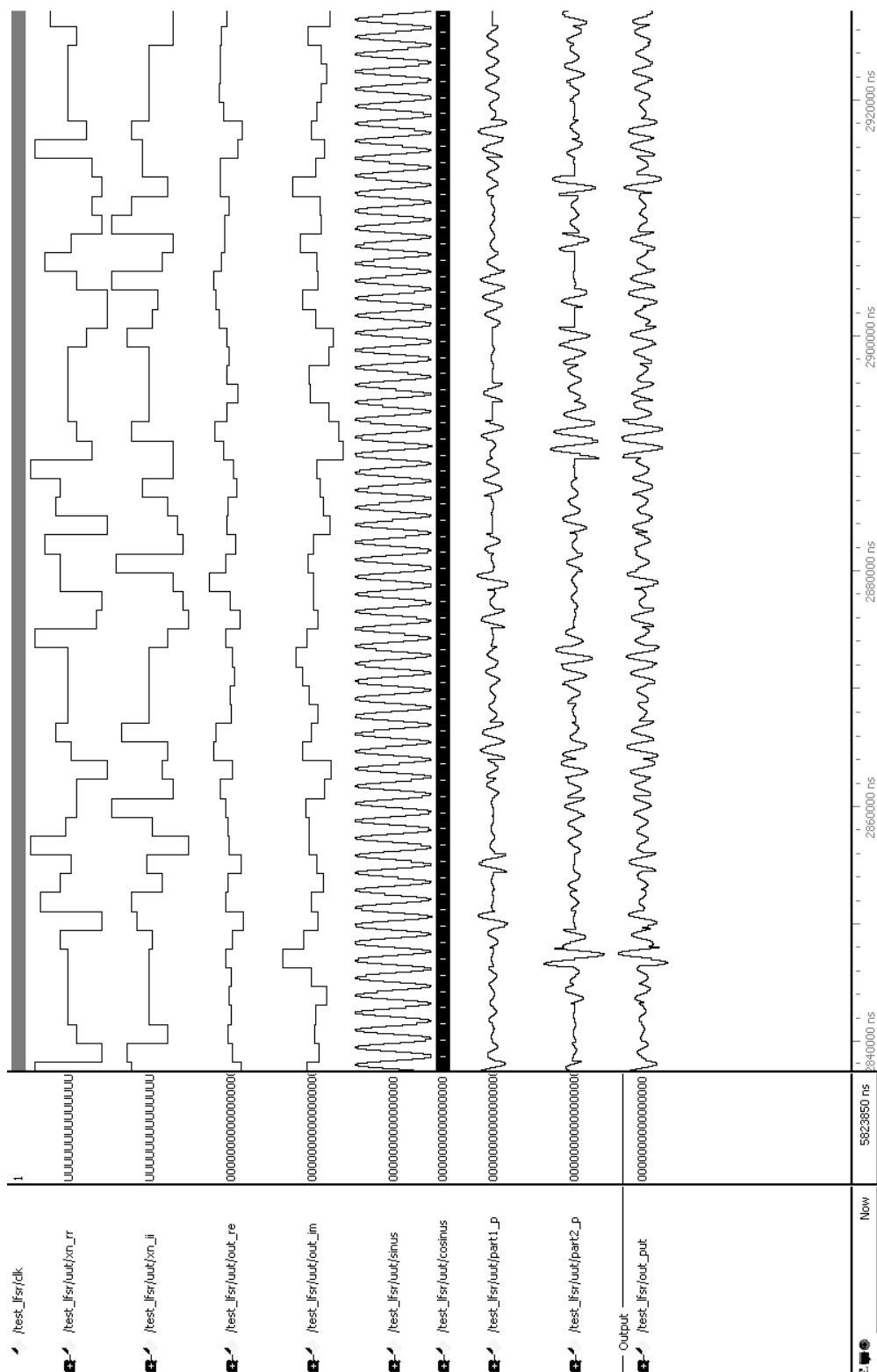
Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	út 20. V 12:24:37 2008	0	144 Warnings	66 Infos
Translation Report	Current	út 20. V 12:25:26 2008	0	107 Warnings	0
Map Report	Current	út 20. V 12:25:58 2008	0	3 Warnings	4 Infos
Place and Route Report	Current	út 20. V 12:29:09 2008	0	1 Warning	2 Infos
Static Timing Report	Current	út 20. V 12:29:31 2008	0	0	3 Infos
Bitgen Report	Current	út 20. V 12:30:19 2008	0	1 Warning	1 Info

A.2 Simulace ModelSim - časové průběhy bloků



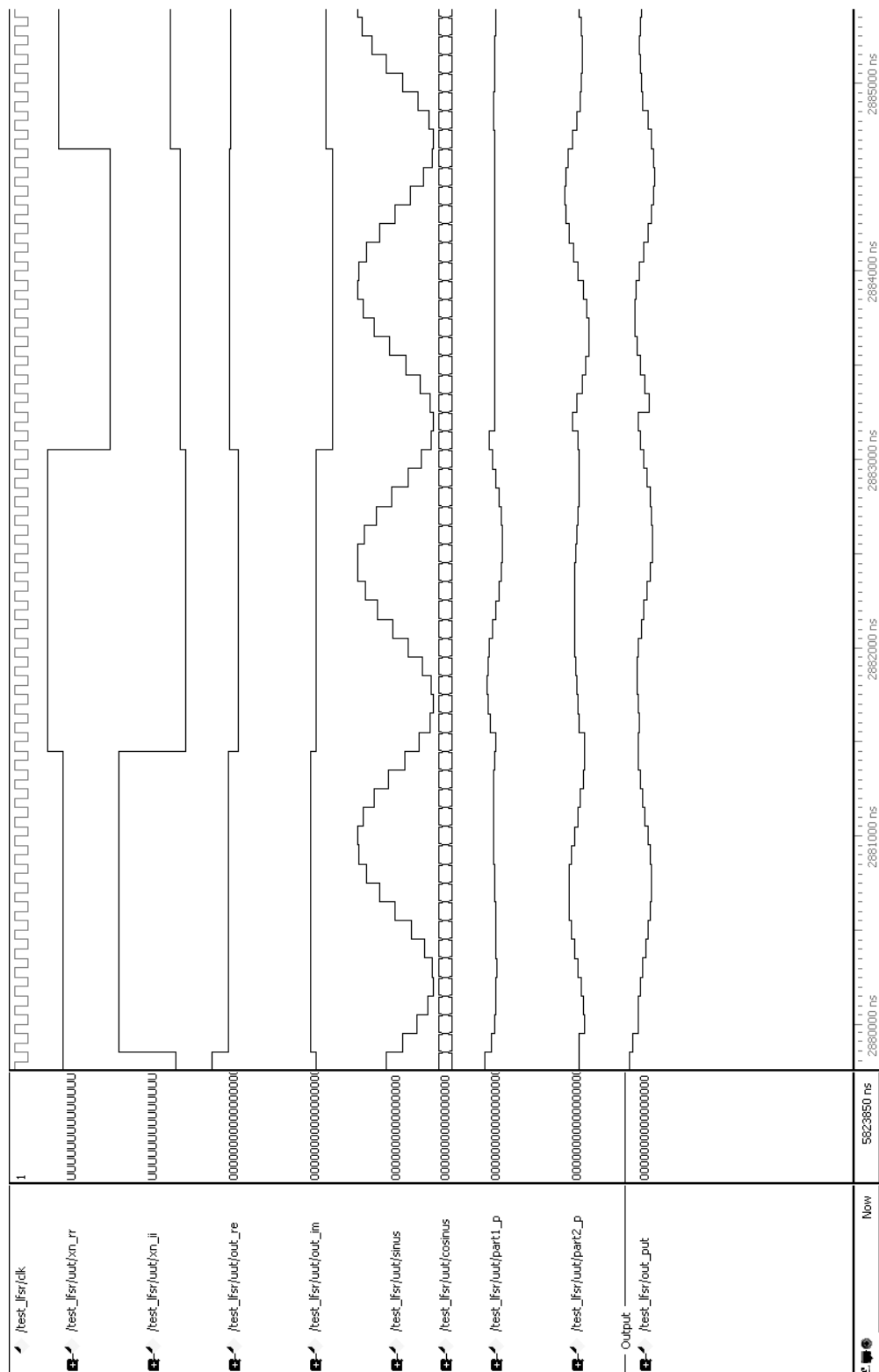
Průběhy výstupních signálů funkčních bloků v rozpětí 0,6ms. Pro ilustraci funkcí jsou znázorněny v ModelSimu jako průběhy odpovídající napětí v čase. Znázornění v časovém průběhu dovoluje rozumné zobrazení i delších úseků signálu (implicitní zobrazení v bitové reprezentaci).

A.3 Simulace ModelSim - časové průběhy bloků 2



Časové průběhy výstupních signálů funkčních bloků v rozpětí 0,1ms. Průběhy jsou totožné s předešlými, pouze je použito zvětšené měřítko.

A.4 Simulace ModelSim - časové průběhy bloků 3



Poslední časové průběhy výstupních signálů funkčních bloků jsou v rozpětí $5\mu s$. Opět se jedná o totožné průběhy a ještě více zvětšené měřítko.