

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

OVLÁDÁNÍ PROGRAMOVATELNÉHO ANALOGOVÉHO OBVODU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JOSEF SOBOTKA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# OVLÁDÁNÍ PROGRAMOVATELNÉHO ANALOGOVÉHO OBVODU

CONTROLLING OF PROGRAMMABLE ANALOGUE CIRCUIT

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JOSEF SOBOTKA

VEDOUCÍ PRÁCE  
SUPERVISOR

prof. Ing. KAMIL VRBA, CSc.

BRNO 2013



**VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky  
a komunikačních technologií**

**Ústav telekomunikací**

# Bakalářská práce

bakalářský studijní obor  
**Teleinformatika**

**Student:** Josef Sobotka

**ID:** 136584

**Ročník:** 3

**Akademický rok:** 2012/2013

## NÁZEV TÉMATU:

**Ovládání programovatelného analogového obvodu**

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte zapojení nového diferenčního analogového programovatelného obvodu QTA, který obsahuje transkonduktační, transadmitanční zesilovač, říditelný dvoustupňový napěťový zesilovač a dvoustupňový napěťový zesilovač s fixním zesílením. Pro nastavení parametrů tohoto obvodu a pro propojení dílčích částí navrhnete a otestujete ovládání prostřednictvím počítače. Navrhnete zkušební desku plošného spoje, ověříte její kompletní funkčnost a změříte vybrané základní parametry obvodu.

## DOPORUČENÁ LITERATURA:

- [1] Koudar, I.: Low Noise Variable Gain Analog Front. Technická zpráva ON Semiconductor, 2012
- [2] Atmel: Microcontrollers. Atmel Data Sheet, 2012

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 5.6.2013

**Vedoucí práce:** prof. Ing. Kamil Vrba, CSc.

**Konzultanti bakalářské práce:**

**prof. Ing. Kamil Vrba, CSc.**  
*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## **ABSTRAKT**

Tato práce se zabývá návrhem ovládacího softwaru pro programovatelný analogový obvod nastavitelný prostřednictvím počítače přes sériové rozhraní a to jak pro počítač, tak pro programovací mikrokontrolér. Dále se zabývá analýzou vnitřní struktury programovatelného analogového obvodu, na kterou je dále ovládání navrženo, návrhem desky plošného spoje pro výsledné programovací a zkušební zařízení a měřením základních parametrů obvodu (přenosové a fázové frekvenční charakteristiky dílčích částí i celků).

## **KLÍČOVÁ SLOVA**

bota, btia, fdca, vga, usart, nastavení, programovatelný, analogový, obvod, mikrokontrolér, převodník, sériové, rozhraní, USB, C#, program, firmware, návrh, deska

## **ABSTRACT**

This work deals with the design of the control software for the programmable analog circuit adjustable by means of a computer via the serial interface for the computer, as well as for programming the microcontroller. In addition, analyses the internal structure of programmable analog circuit on which it is proposed to further control, design of printed circuit boards for final programming and test equipment and measurement of the basic parameters of the circuit (transmission and phase frequency characteristics of parts and units).

## **KEYWORDS**

bota, btia, fdca, vga, usart, setting, programmable, analog, circuit, microcontroller, converter, serial, interface, USB, C#, program, firmware, design, board

SOBOTKA, Josef *Ovládání programovatelného analogového obvodu*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 95 s. Vedoucí práce byl prof. Ing. Kamil Vrba, CSc.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Ovládání programovatelného analogového obvodu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce prof. Ing. Kamilu Vrbovi, CSc. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Dále bych chtěl poděkovat Ing. Pavlu Hanákovi, Ph.D. za odborné konzultace k návrhu desky plošného spoje a konstrukci samotné, Ing. Janu Jeřábkovi, Ph.D. za odborné konzultace ohledně oživování a měření a Ing. Ivanu Koudarovi, Ph.D. za odborné konzultace ohledně teorie nastavovaného obvodu.

Brno .....

.....

(podpis autora)

# OBSAH

<b>Úvod</b>	<b>11</b>
<b>1 Teoretický základ</b>	<b>12</b>
1.1 Diferenční varianty analogových obvodů . . . . .	12
1.2 Operační transkonduktanční zesilovač . . . . .	12
1.3 Operační transimpedanční zesilovač . . . . .	15
1.4 Plně diferenciální proudový zesilovač . . . . .	18
1.5 $\Sigma - \Delta$ převodník . . . . .	18
<b>2 Analýza programovatelného analogového obvodu</b>	<b>20</b>
2.1 Programovatelný analogový obvod . . . . .	20
2.1.1 Piny obvodu . . . . .	20
2.1.2 Popis bloků obvodu . . . . .	20
<b>3 Návrh softwaru</b>	<b>31</b>
3.1 Ovládací program pro PC . . . . .	31
3.1.1 Úvahy nad řešením ovládací aplikace . . . . .	31
3.1.2 Okénko do objektově orientovaného programování . . . . .	31
3.1.3 Architektura ovládací aplikace . . . . .	32
3.1.4 Třída PAOProperties . . . . .	32
3.1.5 Třída UsbCom . . . . .	34
3.1.6 Třída Profiler . . . . .	43
3.1.7 Třída InfoConsole . . . . .	44
3.1.8 Třída Form1 . . . . .	45
3.1.9 Popis rozvržení okna aplikace . . . . .	46
3.2 Firmware pro mikrokontrolér . . . . .	48
3.2.1 Důležité funkce . . . . .	48
3.2.2 Zhodnocení návrhu softwaru . . . . .	50
<b>4 Konstrukce desek plošných spojů</b>	<b>51</b>
4.1 První verze zařízení . . . . .	51
4.1.1 Návrh zkušební desky plošného spoje . . . . .	51
4.1.2 Návrh měřicí desky plošného spoje . . . . .	54
4.1.3 Nevýhody první verze zařízení . . . . .	56
4.2 Druhá verze zařízení . . . . .	57
4.3 Zhodnocení druhé verze zařízení . . . . .	57
<b>5 Měření na programovatelném analogovém obvodu</b>	<b>62</b>
5.1 Úvodní měření obvodu . . . . .	62
5.1.1 Blok LNA . . . . .	62

5.1.2	Blok VGA . . . . .	63
5.1.3	Blok FGA . . . . .	65
5.1.4	Celý LNVGA . . . . .	66
5.2	Předřadný blok $R_{IN}$ . . . . .	68
5.2.1	Zhodnocení úvodních měření . . . . .	69
5.3	Analýza obvodovým analyzátozem . . . . .	69
5.3.1	Zhodnocení měření na obvodovém analyzátoru . . . . .	70
<b>6</b>	<b>Závěr</b>	<b>72</b>
	<b>Literatura</b>	<b>73</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>75</b>
	<b>Seznam příloh</b>	<b>77</b>
<b>A</b>	<b>Odkazy do fotogalerie pozorování</b>	<b>78</b>
<b>B</b>	<b>Výsledky nediferenčního měření na obvodovém analyzátoru</b>	<b>83</b>
<b>C</b>	<b>Výkresy schémat desek plošných spojů</b>	<b>95</b>
C.1	Popis obsahu přiloženého CD-ROM . . . . .	95
C.2	Výkresy schémat . . . . .	95



# SEZNAM OBRÁZKŮ

1.1	Operační transkonduktanční zesilovač – a) ideální a b) reálný model (upraveno podle [5]) . . . . .	13
1.2	Demostrační bipolární OTA (upraveno podle [5]) . . . . .	14
1.3	Transimpedanční zesilovač – a) ideální a b) reálný model (upraveno podle [3, 4]) . . . . .	15
1.4	Demostrace zesilovače TIA (principiální zapojení, převzato z [1]) . . . . .	17
1.5	Blokové schéma $\Sigma - \Delta$ převodníku 1. řádu (upraveno podle [9]) . . . . .	18
2.1	Programovatelný analogový obvod – Pinout . . . . .	20
2.2	Architektura BOTA uvnitř programovatelného analogového obvodu (převzato z [10]) . . . . .	21
2.3	Architektura BTIA uvnitř programovatelného analogového obvodu (převzato z [10]) . . . . .	22
2.4	Redukce vstupní impedance před blokem jedna (upraveno podle [10]) . . . . .	22
2.5	Architektura LNA bloku jedna (upraveno podle [10]) . . . . .	22
2.6	Architektura vnitřního signálového multiplexu (upraveno podle [10]) . . . . .	25
2.7	Ukázka funkce implementovaného $\Sigma - \Delta$ modulátoru . . . . .	27
3.1	Sériový rámec odesílaný do programovacího mikrokontroléru . . . . .	35
3.2	Vzhled ovládací aplikace . . . . .	47
4.1	Horní vrstva zkušební DPS (první verze) . . . . .	52
4.2	Spodní vrstva zkušební DPS (první verze) . . . . .	52
4.3	Napájecí mezi-vrstva zkušební DPS (první verze) . . . . .	53
4.4	Mezi-vrstva zemí zkušební DPS (první verze) . . . . .	53
4.5	Horní vrstva měřicí DPS (první verze) . . . . .	55
4.6	Spodní vrstva měřicí DPS (první verze) . . . . .	55
4.7	Zákmity na výstupním signálu celého řetězce (první verze zařízení) . . . . .	56
4.8	Zákmity na výstupním signálu bloku LNA (první verze zařízení) . . . . .	57
4.9	Výstupní signál celého řetězce (druhá verze zařízení) . . . . .	58
4.10	Horní vrstva programovací DPS (druhá verze) . . . . .	59
4.11	Spodní vrstva programovací DPS (druhá verze) . . . . .	59
4.12	Horní vrstva zkušební DPS (druhá verze) . . . . .	60
4.13	Napájecí mezi-vrstva zkušební DPS (druhá verze) . . . . .	60
4.14	Mezi-vrstva zemí zkušební DPS (druhá verze) . . . . .	61
4.15	Spodní vrstva zkušební DPS (druhá verze) . . . . .	61
5.1	Auto-oscilace celkového řetězce pro vstup „ve vzduchu“ u první verze zařízení . . . . .	67
5.2	Grafická závislost $U_{VST\_LNA}(U_{DAMP\_CTRL})$ pro $U_{BT1\_IN} = \text{konst.}$ . . . . .	69
B.1	Frekvenční závislost zesílení $A_U(f)$ pro blok LNA (větší oblast spektra) . . . . .	83
B.2	Fázová charakteristika $\varphi(f)$ pro blok LNA (větší oblast spektra) . . . . .	83
B.3	Frekvenční závislost zesílení $A_U(f)$ pro blok VGA (větší oblast spektra) . . . . .	84
B.4	Fázová charakteristika $\varphi(f)$ pro blok VGA (větší oblast spektra) . . . . .	84

B.5	Frekvenční závislost zesílení $A_U(f)$ pro podblok VGA2 bloku VGA (větší oblast spektra) . . . . .	85
B.6	Fázová charakteristika $\varphi(f)$ pro podblok VGA2 bloku VGA (větší oblast spektra) . . . . .	85
B.7	Frekvenční závislost zesílení $A_U(f)$ pro podblok A2 bloku FGA (větší oblast spektra) . . . . .	86
B.8	Fázová charakteristika $\varphi(f)$ pro podblok A2 bloku FGA (větší oblast spektra)	86
B.9	Frekvenční závislost zesílení $A_U(f)$ pro podblok A3 bloku FGA (větší oblast spektra) . . . . .	87
B.10	Fázová charakteristika $\varphi(f)$ pro podblok A3 bloku FGA (větší oblast spektra)	87
B.11	Frekvenční závislost zesílení $A_U(f)$ pro blok FGA (větší oblast spektra) . .	88
B.12	Fázová charakteristika $\varphi(f)$ pro blok FGA (větší oblast spektra) . . . . .	88
B.13	Frekvenční závislost zesílení $A_U(f)$ pro řetězec LNVGA (větší oblast spektra)	89
B.14	Fázová charakteristika $\varphi(f)$ pro řetězec LNVGA (větší oblast spektra) . . .	89
B.15	Frekvenční závislost zesílení $A_U(f)$ pro blok VGA (detail v pracovní oblasti)	90
B.16	Fázová charakteristika $\varphi(f)$ pro blok VGA (detail v pracovní oblasti) . . .	90
B.17	Frekvenční závislost zesílení $A_U(f)$ pro podblok VGA2 bloku VGA (detail v pracovní oblasti) . . . . .	91
B.18	Fázová charakteristika $\varphi(f)$ pro podblok VGA2 bloku VGA (detail v pracovní oblasti) . . . . .	91
B.19	Frekvenční závislost zesílení $A_U(f)$ pro podblok A3 bloku FGA (detail v pracovní oblasti) . . . . .	92
B.20	Fázová charakteristika $\varphi(f)$ pro podblok A3 bloku FGA (detail v pracovní oblasti) . . . . .	92
B.21	Frekvenční závislost zesílení $A_U(f)$ pro blok FGA (detail v pracovní oblasti)	93
B.22	Fázová charakteristika $\varphi(f)$ pro blok FGA (detail v pracovní oblasti) . . . .	93
B.23	Frekvenční závislost zesílení $A_U(f)$ pro řetězec LNVGA (detail v pracovní oblasti) . . . . .	94
B.24	Fázová charakteristika $\varphi(f)$ pro řetězec LNVGA (detail v pracovní oblasti)	94

## SEZNAM TABULEK

2.1	Popis pinů a jejich funkcí pro obvod . . . . .	26
2.2	Vlastnosti bloku LNA . . . . .	27
2.3	Vlastnosti bloku VGA . . . . .	27
2.4	Kód zisku VGA podle hodnoty v kontrolním registru . . . . .	28
2.5	Vlastnosti podbloku A2 bloku FGA . . . . .	28
2.6	Vlastnosti podbloku A3 bloku FGA . . . . .	29
2.7	Vlastnosti $\Sigma - \Delta$ modulátoru . . . . .	29
2.8	Kombinace meziblokových sad přepínačů pro řídicí hodnoty . . . . .	29
2.9	Význam užitečných hodnot v poli kontrolních registrů . . . . .	30
3.1	Využití parametrů informační struktury TransRes . . . . .	34
4.1	Rozpis součástek na zkušební DPS (první verze) . . . . .	54
4.2	Rozpis součástek na měřicí DPS . . . . .	56
4.3	Rozpis součástek na novou desku programovacího zařízení . . . . .	58
4.4	Rozpis součástek na zkušební desku (druhá verze) . . . . .	61
5.1	Pozorování podbloku VGA2 bloku VGA . . . . .	63
5.2	Pozorování bloku VGA . . . . .	64
5.3	Pozorování řetězce LNVGA . . . . .	68
5.4	Pohled na jednotlivé bloky (podbloky) a celky na širší části spektra . . . . .	71
5.5	Pohled na jednotlivé bloky (podbloky) a celky na pracovní části spektra . . . . .	71
A.1	Odkazy do fotogalerie – pozorování podbloku VGA2 bloku VGA . . . . .	78
A.2	Odkazy do fotogalerie – pozorování FGA . . . . .	78
A.3	Odkazy do fotogalerie – pozorování podbloku A2 bloku FGA . . . . .	78
A.4	Odkazy do fotogalerie – pozorování podbloku A3 bloku FGA . . . . .	78
A.5	Odkazy do fotogalerie – pozorování LNA . . . . .	78
A.6	Odkazy do fotogalerie – pozorování VGA . . . . .	79
A.7	Odkazy do fotogalerie – pozorování LNVGA (část 1) . . . . .	80
A.8	Odkazy do fotogalerie – pozorování LNVGA (část 2) . . . . .	81
A.9	Odkazy do fotogalerie – pozorování LNVGA (část 3) . . . . .	82

## ÚVOD

Analogová technika je odvětví elektrotechniky, které se jako jednou z mnoha disciplín zabývá filtrováním určitých frekvencí, případně jejich skupin či pásem. Obor filtrování se rozdělil na dva celky, z nichž v jednom má smysl filtrovat signál reprezentovaný proudovou veličinou (za cenu složitějších prvkových struktur, ale jednodušších struktur obvodových) a ve druhém napětovou veličinou (za cenu složitější obvodové struktury na úkor jednodušších prvků). Na začátku semestru jsem byl seznámen s novým nízkošumovým analogovým obvodem, který slouží jako brána mezi napětovým a proudovým módem a má navíc určité další prvky. Tento obvod je programovatelný, tedy přes paralelní digitální rozhraní jsme schopni regulovat jeho vlastnosti. Mým úkolem bylo analyzovat vnitřní strukturu nového obvodu, vytvořit rozhraní přes počítač, kterým se budou dát jeho vlastnosti nastavit, vytvořit zkušební desku plošného spoje pro měření parametrů obvodu a následně změřit základní parametry jeho bloků (přenosovou a fázovou charakteristiku).

# 1 TEORETICKÝ ZÁKLAD

Programovatelný analogový obvod, pro nějž níže navrhuji ovládací software, obsahuje transkonduktanční a transimpedanční zesilovač (zde blok LNA), dvoustupňový napěťový zesilovač s nastavitelným zesílením (zde jako blok VGA s podbloky VGA1, VGA2) a dvoustupňový napěťový zesilovač s fixním zesílením (zde jako blok FGA s podbloky A2, A3). Ke zpracování jsem použil obvody ATmega88, FT232BL (v druhé verzi zařízení nahrazen obvodem FT232RL) a sériovou komunikaci. V této části chci rámcově teoreticky popsat transkonduktanční a transimpedanční zesilovač z pohledu principu a nastínit diferenční variantu analogových obvodů.

## 1.1 Diferenční varianty analogových obvodů

Diferenční zpracování signálů nalézá v dnešní době telekomunikací a vůbec minimálními nároky na výskyt rušení čím dál tím více uplatnění. Tento princip se používá ve vysokofrekvenční technice, zpracování audio signálů, v konstrukci filtrů, apod. Podstata plyne z faktu, že pokud máme nediferenční systém, do signálu vedeného pouze jedním vodičem (včetně země), se cestou může indukovat rušení plynoucí ze špatné konstrukce napájecího zdroje nebo indukce rušení ze vzduchu. Ze zarušeného signálu lze šum filtrovat, avšak již nikdy tak přesně. Diferenční model v sobě zahrnuje přenos signálů po dvou vodičích. Pro takovýto systém platí vztahy (míněno pro napěťové signály):

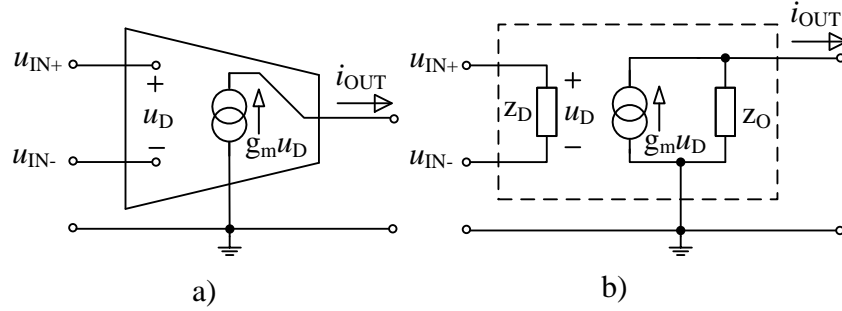
$$\begin{aligned} u_{IN} &= u_{IN+} - u_{IN-} \\ u_{OUT+} &= -u_{OUT-} = \frac{A_U}{2} u_{IN} \end{aligned} \quad (1.1)$$

Oba vstupní signály  $u_{IN+}$  a  $u_{IN-}$  jsou do systému přiváděny v inverzní podobě (např. pokud signál  $u_{IN+}$  bude sinusoida, signál  $u_{IN-}$  musí být sinusoida posunutá o  $90^\circ$ ). Pro vstupní signály platí požadavek polovičního rozkmitu oproti nediferenčnímu systému (zapříčiní redukci zkreslení a sníží nároky na případný napájecí zdroj). Oba se navzájem odečtou, dojde k eliminaci souhlasných a většinou i sudých harmonických složek signálu. Protože oba signály byly velikostně v jednotkovém poměru a po vzájemném odečtení se výsledek dostal do dvojnásobku okamžité hodnoty libovolného ze signálů vstupujících, je třeba, aby přenos  $A_U$  byl poloviční, tím dostaneme signály na původní velikost (na rozdíl od nediferenčního obvodu, kde je výstup pouze jeden a dělení energie signálu neřešíme). Výstupem je potom dvojice signálů  $u_{OUT+}$ ,  $u_{OUT-}$  z nichž oba jsou signály navzájem inverzní. Více o principu v [8].

## 1.2 Operační transkonduktanční zesilovač

Je považován za univerzální zapojení a součástku, protože dokáže v dnešní době vykonat většinu matematických operací. Jeho obvyklá konstrukce zahrnuje diferenční pár na

vstupu, který má jako svou zátěž připojené proudové zrcadlo, a většinou jednoduchý zesilovač třídy A. Svoji konstrukcí vykazuje konečné zesílení (dáno transkonduktancí  $g_m$ ) a tedy není nutné zavádět zpětnou vazbu, svým principem představuje převodník napětí-proud, tedy z diferenčních vstupních napětí  $u_{IN+}$ ,  $u_{IN-}$  na své transkonduktanci  $g_m$  vytvoří výstupní proud  $i_{OUT}$ . Nejčastěji je používán při stavbě filtrů v proudovém módu, zapojení se vyznačují jednoduchostí a minimálním počtem přídavných součástek. Na obr. 1.1 je uveden model zesilovače OTA.



Obr. 1.1: Operační transkonduktanční zesilovač – a) ideální a b) reálný model (upraveno podle [5])

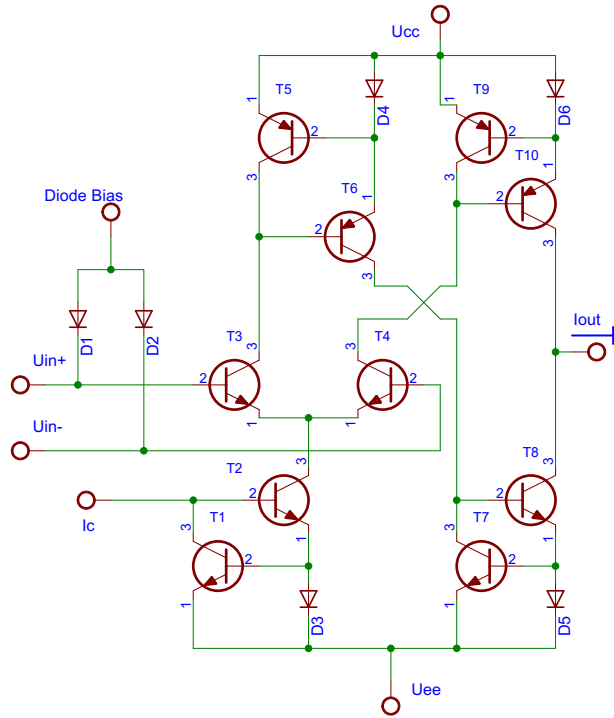
Aby se nemohl efekt zatížení vstupu zesilovače projevit na výstup, musí být impedance mezi vstupními diferenčními svorkami rovna nekonečnu (vstupní signály rozumíme napěťové), výstupní impedance potom, protože se jedná o ideální zdroj proudu, musí být též rovna nekonečnu (výstupní signál je rozuměn jako proudový). Pro ideální OTA platí rovnice (viz [5]):

$$i_{OUT} = g_m(u_{IN+} - u_{IN-}) \quad (1.2)$$

Prvek BOTA (Balanced OTA) je použit v mnou nastavovaném obvodu. Princip je naprosto stejný jako pro nediferenční zesilovač OTA. Jedinou výjimkou je diferenční proudový výstup (navzájem inverzní proudové signály s polovičním rozkmitem odpovídajícím poloviční transkonduktanci  $g_m$ ).

Pro demonstraci bych chtěl vysvětlit princip činnosti zapojení OTA s bipolárními tranzistory na obr. 1.2. Diody  $D_1$ ,  $D_2$  nastavují pracovní bod klidovým proudem do bází tranzistorů  $T_3$ ,  $T_4$  diferenčního páru. Takto zapojené diody však mohou dodávat při pro různá  $u_{IN+}$  a  $u_{IN-}$  nekonstantní proud, abychom zachovali linearitu buzení, je třeba zpravidla přidat určitou sério-paralelní kombinaci rezistorů, díky nimž se daří dodávaný proud pro určitý rozsah  $u_{IN+}$  a  $u_{IN-}$  linearizovat.

První proudové zrcadlo tvoří součástky  $T_1$ ,  $T_2$ ,  $D_3$  – jeho podstatou je externě přivezený proud  $I_C$  zrcadlit do proudu  $i_{C2} = i_{C3} + i_{C4}$ . Proud  $I_C$  otevře tranzistor  $T_2$ , proud  $i_{C2}$  roste do chvíle, dokud se na diodě  $D_3$  neobjeví  $u_{D3} = u_{BE1} = 0,65$  V, v tu chvíli se otevře  $T_1$  a začne proud  $I_C$  zmenšovat o svůj  $i_{C1}$  – tedy  $i_{B2} = I_C - i_{C1}$ . Proud  $i_{C2}$  se ustálí na takové hodnotě, při které je  $U_{D3} < 0,65$  V. V tomto případě má tranzistor  $T_1$  s diodou



Obr. 1.2: Demostrační bipolární OTA (upraveno podle [5])

$D_3$  charakter pojistky proti případnému většímu proudu  $i_{C2}$ , kombinace též zajistí, aby  $i_{C2} = \text{konst.}$  a vytvoří tak zdroj konstantního proudu.

Druhé zrcadlo tvořené součástkami  $T_5$ ,  $D_4$ ,  $T_6$  slouží ke kopii proudu  $i_{C5} = i_{C3}$  do proudu  $i_{C6}$  (dioda  $D_4$  tentokrát slouží pro otevření tranzistoru  $T_5$ , protože  $U_{D4} = u_{BE5}$ ). Třetí zrcadlo tvořené součástkami  $T_7$ ,  $D_5$ ,  $T_8$  převádí proud  $i_{C6}$  na proud  $i_{C8}$ . Čtvrté zrcadlo tvořené součástkami  $T_9$ ,  $D_6$ ,  $T_{10}$  provádí kopii proudu  $i_{C4}$  na proud  $i_{C10}$ . V poslední větvi, ve které odebíráme výstupní proud  $i_{OUT}$ , platí rovnice  $i_{OUT} = i_{C10} - i_{C8}$ .

Diferenční pár je tvořen tranzistory  $T_3$ ,  $T_4$ . V klidovém stavu – např.  $u_{IN+} = u_{IN-} = 0\text{ V}$ , jsou oba tranzistory nastaveny klidovým proudem do bází  $i_{B3} = i_{B4}$ , dále platí  $i_{C3} = i_{C4}$ , tedy  $i_{OUT} = i_{C10} - i_{C8} = i_{C4} - i_{C3} = 0\text{ A}$ . Pokud přivedeme na svorky nenulový signál (a v okamžité hodnotě bude např.  $u_{IN+} > u_{IN-}$ ), bude  $T_4$  protékat o něco větší proud  $i_{C4}$  než tranzistorem  $T_3$  a tento rozdíl bude zesílen jako  $i_{OUT}$ . Je třeba, aby oba tranzistory v páru byly stejných parametrů. Zapojení je odolné vůči teplotním výkyvům obou tranzistorů, protože ty se jeví jako souhlasný přírůstek na  $u_{BE}$  tranzistorů (ujede-li jedno, ujede i druhé).

Z důvodu, že oba tranzistory na svém vstupu pracují s napětím  $u_{BE}$ , nikoliv s proudem  $i_B$ , a funkce  $i_B = f(u_{BE})$  má exponenciální charakter, výstupní proud celého zapojení splňuje vztah (viz [5]):

$$i_{OUT} = i_{C4} - i_{C3} = I_C \tanh \frac{u_{IN+} - u_{IN-}}{2U_T}, \quad (1.3)$$

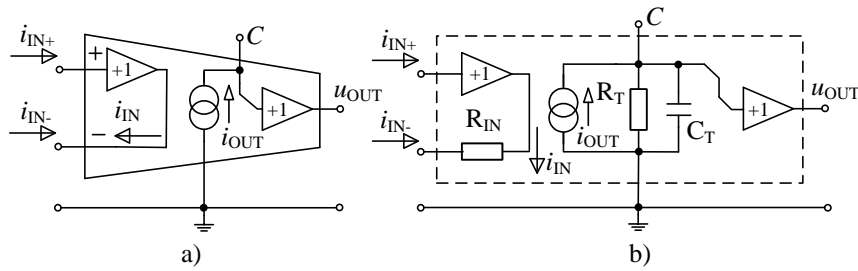
kde  $U_T \approx 25,5$  mV teplotní napětí. Pokud vezmeme v úvahu, že  $|u_{IN+} - u_{IN-}| \ll 2U_T$ , rovnice se zjednoduší na:

$$i_{OUT} = \frac{I_C}{2U_T} (u_{IN+} - u_{IN-}). \quad (1.4)$$

Z rovnice 1.4 pak plyne v porovnání s rovnicí 1.2, že  $g_m = \frac{I_C}{2U_T}$  (pokud bychom vzali v úvahu linearizaci vstupních nastavovacích klidových proudů do bází  $T_3$ ,  $T_4$ , vztah i řád výsledné  $g_m$  bude stejný). O principech diferenčního páru, proudových zrcadel a ostatních analogových záležitostech podrobně v [17].

### 1.3 Operační transimpedanční zesilovač

Je opět jedno z hodně používaných zapojení, slouží především pro zpracování vysokofrekvenčních signálů v celé šíři aplikací (např. síťové technologie s optickými přenosy, audiotechnika a obecně radiotechnika). Zapojení v principu obsahuje vstupní proudový snímač a proudem řízený zdroj proudu, který na transimpedanci vytváří výstupní napětí. Ideální TIA má mezi svými vstupními proudovými diferenčními svorkami nulovou impedanci, jeho výstupní impedance, protože se jedná o ideální zdroj napětí, má taktéž nulovou hodnotu. Jako u klasického operačního zesilovače je vhodné i pro TIA zavádět zápornou proudovou zpětnou vazbu (mají hodně společných znaků na rozdíl od OTA). Ideální a k němu reálná TIA na obr. 1.3.



Obr. 1.3: Transimpedanční zesilovač – a) ideální a b) reálný model (upraveno podle [3, 4])

Obsahuje diferenční vstupní svorky, kterými přivádíme  $i_{IN+}$  přes tzv. vysokoimpedanční a  $i_{IN-}$  přes tzv. nízkoimpedanční vstupní svorku do vstupní části zesilovače. Vstupní část tvoří napěťový sledovač připojený na vysokoimpedanční svorku a proudový sledovač



připojený na nízkoimpedanční svorku. Na proudovém sledovači se provede rozdíl vstupních diferenčních proudů (viz rovnice 1.6), se kterým se dále pracuje (vstup je díky vřazenému napětovému sledovači nesymetrický). Proudovým sledovačem (symbolizován odporem  $R_{IN}$ ) řídíme zdroj proudu řízený proudem  $i_{OUT}$  (mohl by tam být proudový obračec, většinou se ale aplikuje proudový konvektor), který následně protéká transimpedančním členem tvořeným prvky  $R_T$  a  $C_T$  (tyto prvky tvoří převodní transimpedanci  $r_m$ ) a vytváří na nich napětí  $u_{OUT}$ , které projde napětovým sledovačem a přenesení se na výstupní svorku. Časová konstanta transimpedančního členu má též zásadní význam na stabilitu zesilovače podle Nyquistova kritéria. Střídavé parametry zesilovače ovlivňují hlavně  $C_T$  a  $R_C$  (odpor připojený mezi svorkou C a zemí), odpor  $R_C$  se většinou volí v řádech  $k\Omega$ .

Pro další postup uvažuji příklad hodnot  $R_T, C_T$  z reálného TIA –  $R_T = 3\text{ M}\Omega$ ,  $C_T = 4,5\text{ pF}$ ,  $R_{IN} = 50\text{ }\Omega$ .

Stejně jako klasický operační zesilovač, TIA může a nemusí mít zavedenou zápornou zpětnou vazbu (její konkrétní podoba se liší podle typu aplikace). Pokud ji nezavedeme, transimpedanční člen na výstupní části před napětovým sledovačem rapidně sníží mezní frekvenci zesilovače  $f_m$  na hodnotu cca  $12\text{ kHz}$  podle rovnice 1.9, dále nebude nijak omezena převodní transimpedance  $r_m$  zesilovače, která je ideálně  $r_m \rightarrow \infty$  (pokud by součást transimpedančního členu  $R_T \rightarrow \infty$  pro ideální zdroj proudu), reálně  $r_m \approx 60\text{ }000$ .

Vztahy pro zesílení s uzavřenou smyčkou zpětné vazby jsou dány konkrétní aplikací obvodu (např. invertující zesilovač, převodník proud-napětí). Aplikace zapojení v nastavovaném PAO slouží jako převodník proud-napětí (zapojení se skrytou svorkou C obsahuje odpor ve zpětné vazbě  $R_{TIA}$ , pro převod platí rovnice 1.11).

Pro ideální TIA s otevřenou smyčkou zpětné vazby platí v zásadě tyto vztahy (podle [1]):

$$i_{IN} = i_{IN+} - i_{IN-} \quad (1.5)$$

$$|i_{OUT}| = |i_{IN}| \quad (1.6)$$

$$u_{OUT} = i_{IN} R_T \quad (1.7)$$

$$r_m = R_T \quad (1.8)$$

$$f_m \cong \frac{1}{2\pi R_T C_T} \quad (1.9)$$

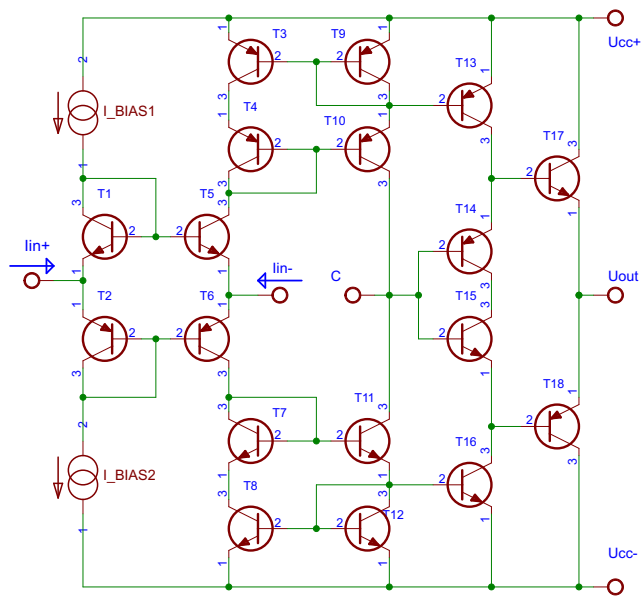
Aplikace v PAO jako převodník proud-napětí splňuje vztahy:

$$u_{OUT} = i_{IN} R_{TIA} \quad (1.10)$$

$$r_m = R_{TIA} \quad (1.11)$$

Prvek BTIA (Balanced TIA) je použit v mnou nastavovaném obvodu. Od klasického TIA se liší diferenčním výstupem a opět polovičním přenosem pro oba výstupy, tedy neinvertující výstup je přenesen přímo z napětového sledovače, a druhý invertující výstup je přenesen přes napětový invertor (v praxi by se mohly použít dvě navzájem oddělené

větvě, z jenž jedné je napětí na koncovém prvku pouze sledováno, ve druhé je invertováno (to by ale vyžadovalo dva vstupy pro připojení  $R_C$ ). Nyní bych chtěl demonstrovat princip jednoho zapojení se zesilovačem TIA.



Obr. 1.4: Demostrace zesilovače TIA (principiální zapojení, převzato z [1])

Na obr. 1.4 je uvedeno principiální zjednodušené schéma obvodu AD488: Zdroje konstantního proudu  $I_{BIAS1}$  a  $I_{BIAS2}$  poskytují určitý proudový offset, který je nutný nejen pro zpracování diferenčních proudů  $i_{IN+}$  a  $i_{IN-}$ , ale i pro nastavení pracovních bodů ostatních tranzistorů ve schématu. Proudové zrcadlo  $T_1, T_5$  přenáší  $I_{BIAS2}$  i do dalších větví zapojení, přičemž zrcadlo  $T_2, T_6$ , mezi něž přivádíme vstupní diferenční signály, vytváří jejich vzájemný rozdíl – tedy  $i_{C1} = I_{BIAS2}$  je po vstupu proudu  $i_{IN+}$  upraven na hodnotu  $i_{C2} = i_{C1} - i_{IN+}$ , která je zrcadlena na protější větev jako proud  $i_{C6}$ . Z proudu  $i_{C6}$  vystupuje proud  $i_{IN-}$ , vznikne jejich rozdíl, přičemž  $i_{C6} = (I_{BIAS2}) + i_{IN+} - i_{IN-}$ . Složená zrcadla  $T_3, T_4, T_9, T_{10}$  a  $T_7, T_8, T_{11}, T_{12}$  jsou navzájem komplementární a slouží pro přenos rozdílů vstupních proudů (proud  $i_{C6}$ ) a offsetového proudu  $I_{BIAS2}$  do větve se svorkou C zesilovače ( $i_{C10} = i_{C9} = I_{BIAS2}$ ,  $i_{C11} = i_{C12} = I_{BIAS2} + i_{IN+} - i_{IN-}$ ). Svorkou C bude za předpokladu připojeného zpětnovazebního odporu  $R_C$  vytékat proud  $i_{TI} = -I_{BIAS2} + I_{BIAS2} + i_{IN+} - i_{IN-}$  podle 1. Kirchhoffova zákona, vytvoří tedy na rezistoru

$R_{ZV}$  napětí  $u_{OUT} = R_{ZV}(i_{IN+} - i_{IN-})$ , které je přes emitorový sledovač  $T_{17}$ ,  $T_{18}$  přeneseno na výstup obvodu. Tranzistory  $T_{14}$ ,  $T_{15}$  linearizují převodní charakteristiku emitorového sledovače, která je v oblasti počátku mírně nelineární kvůli přechodovému zkreslení (tranzistor se neotevře od  $U_{BE} < 0,65 \text{ V}$ ). Tranzistory  $T_{13}$ ,  $T_{16}$  opět přivádí  $I_{BIAS2}$  a rozdíl diferenčních vstupních proudů pro zajištění funkce emitorového sledovače. Výstupní napětí  $u_{OUT}$  je emitorovým sledovačem posíleno pro odběr do dalších částí zapojení. Celý zesilovač je podél své horizontální osy symetrický pro sladění tepelných pohybů pracovních bodů tranzistorů. Impedance nízko-impedančního vstupu je generována lokálně a není závislá na zpětnovazebním odporu  $R_C$  připojeném ke svorce C. Více o TIA v [1, 4, 3].

## 1.4 Plně diferenciální proudový zesilovač

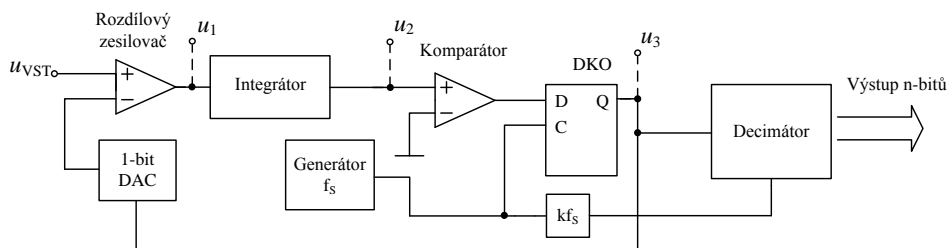
Pokud spojíme BOTA a BTIA do kaskády, vznikne zesilovač interně pracující v proudovém módu, s diferenčními napěťovými vstupy a diferenčními napěťovými výstupy, nebo-li plně diferenciální proudový zesilovač. Přenos takového zesilovače je pak dán kaskádním zapojením dílčích částí, platí vztah (BTIA je zapojen jako převodník proud-napětí):

$$A_{FDCA} = g_m R_{TIA} \quad (1.12)$$

V nastavovaném obvodu je přítomen FDCA, u něhož obsahuje BTIA proměnnou transimpedanci odporem ve zpětné vazbě  $R_{TIA}$ , tedy se dá lineárně měnit jeho zisk.

## 1.5 $\Sigma - \Delta$ převodník

Převodník  $\Sigma - \Delta$  je hojně využívaný nejen v hudební technice, ale i v měření, pokud nám nejde o značnou rychlost převodu. Současné implementace umožňují až 24bitové rozlišení s velmi vysokou linearitou převodu na úkor nižší rychlosti převodu a práce v pásmu do 10 kHz. Na obr. 1.5 je uvedeno blokové schéma  $\Sigma - \Delta$  převodníku (modulátoru) prvního řádu.



Obr. 1.5: Blokové schéma  $\Sigma - \Delta$  převodníku 1. řádu (upraveno podle [9])

Vstupní signál  $u_{VST}$  přivádíme do rozdílového zesilovače, kde je provedena úprava přičtením hodnoty  $\pm U_{REF}$  podle digitálně-analogového převodníku řízeného zpětnou vazbou, jdoucí z klopného obvodu D:

$$u_1 = u_{\text{VST}} + U_{\text{REF}}, \text{ pokud } u_3 = \log. 0 \quad (1.13)$$

$$u_1 = u_{\text{VST}} - U_{\text{REF}}, \text{ pokud } u_3 = \log. 1 \quad (1.14)$$

Napětí  $u_1$  je v integrátoru integrováno na

$$u_2 = \int u_1(t) dt. \quad (1.15)$$

Na komparátoru zjišťujeme, zda-li je hodnota  $u_2 < 0$  nebo  $u_2 > 0$  a podle toho nastavíme napětí  $u_3$  do  $\log. 0$  nebo  $\log. 1$ . Klopný obvod D si zapamatuje hodnotu, která přišla na jeho vstup v době hodnoty  $\log. 1$  hodin. Pokud je hodnota hodin v  $\log. 0$ , zapamatovaná hodnota se nemění. Výstupem z klopného obvodu je též řízen jednobitový digitálně-analogový převodník. Číselná hodnota jdoucí z klopného obvodu je snímána s frekvencí  $f_S$ , ale decimátorem je snímána s frekvencí  $k f_S$  (tedy maximální frekvence obsažená ve vstupním signálu  $u_{\text{VST}}$  musí být:

$$f_{\text{MAX}} = k \frac{f_S}{2} \quad (1.16)$$

podle vzorkovacího teorému). Decimátor se chová v řetězci jako digitální dolní propust (integrátor), pracuje s vyšší vzorkovací frekvencí  $k f_S$  a provádí součet jednotlivých bitů v určitém časovém sledu (chová se jako čítač). Díky tomu, že decimátor pracuje na  $k f_S > f_S$ , je posunut kvantovací šum (jinak rovnoměrně rozdělený ve spektru) na odvrácenou stranu od  $\frac{f_S}{2}$ , odkud je již celkem rovnoměrně rozložen ve spektru. Velmi malé počáteční rozlišení převodníku lze zvýšit převzorkováním, posunutím šumu, apod. Existuje i

$\Sigma - \Delta$  převodník s adaptivní modulací, kde je použit více úrovněový komparátor. Princip je složitější, ale převodník pak již lépe reaguje na strmější hrany signálu. Vzhledem k tomu, že modulátor kóduje přírůstky a úbytky z  $u_{\text{VST}}$ , není schopen zakódovat konstantní signál, kde je nerozhodný. Více informací o převodníku v [9].

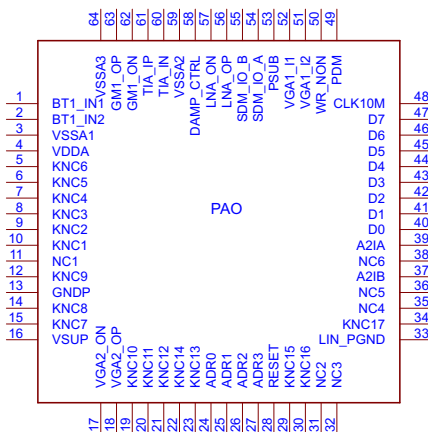
## 2 ANALÝZA PROGRAMOVATELNÉHO ANALOGOVÉHO OBVODU

### 2.1 Programovatelný analogový obvod

Obvod je vyroben v pouzdře LQFP (s vývody po stranách čtverce, úzké provedení). Jeho dílčími výkonnými částmi jsou zesilovače BOTA, BTIA, FDCA, VGA (napěťový zesilovač s proměnným zesílením) a  $\Sigma - \Delta$  modulátor. Obsahuje též pole přepínačů pro připojení jiných vnějších signálových cest do původního vestavěného řetězce výkonných částí, popřípadě volby konfigurace jejich propojení. Celému řetězci, který se dá za jistých podmínek propojit, předchází tzv.  $R_{IN}$  blok (redukce vstupního odporu). Ten má za úkol eliminovat převážný šum, který by mohl vstoupit do řetězce. Určité bloky výkonné i přepínací lze konfigurovat podle paměti se 14 B hodnotami, v nichž jsou zakódovány informace pro jejich nastavení (pro naše účely je jich ve skutečnosti užitečných pouze 6 B).

#### 2.1.1 Piny obvodu

Na obr. 2.1 a následně tab. 2.1 jsou uvedeny dostupné piny obvodu a jejich funkce. Ne všech 64 dostupných pinů obvodu je využito, některé je třeba nepřipojovat nebo rovnou uzemnit, plyne to z faktu, že obvod měl ve své dřívější verzi trochu jinou podstatu funkce (dříve se kontrolní registr řídil sériově a obvod se dal se svými druhy propojit po sériovém rozhraní).



Obr. 2.1: Programovatelný analogový obvod – Pinout

#### 2.1.2 Popis bloků obvodu

##### LNA – blok jedna

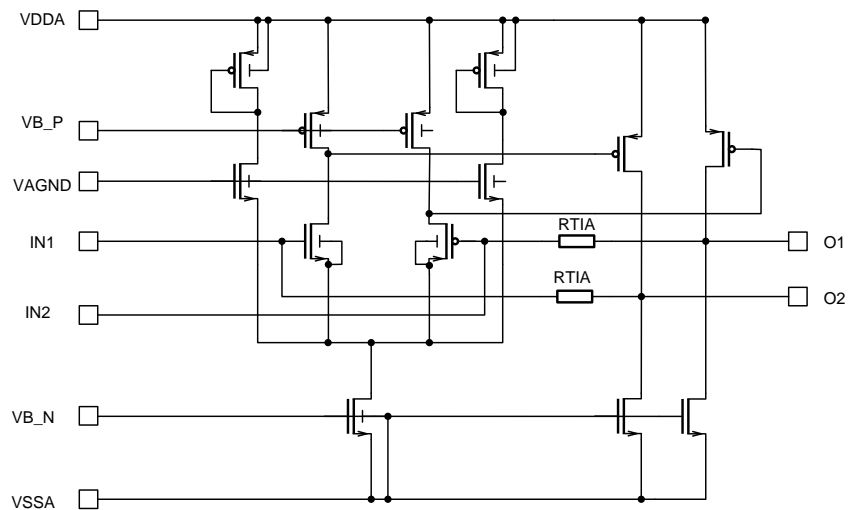
Blok LNA (Low Noise Amplifier) obsahuje kaskádu zesilovačů BOTA a BTIA, zesílení celého bloku je 39,1 dB. Bloku předchází  $R_{IN}$  člen, jehož impedanci jsme pomocí signálu  $U_{DAMP\_CTRL} \in \langle 0,0 \text{ V}; 3,3 \text{ V} \rangle$  schopni regulovat lineárně v rozsahu  $R_{IN} \in \langle 50 \text{ k}\Omega; 800 \Omega \rangle$

Blok LNA a VGA (podblok VGA1) mají v sobě implementovanou tzv. servo smyčku, kterou se rozumí zavedení záporné zpětné vazby mezi výstupem z BTIA a vstupem do BTIA. Protože jsou jednotlivé zesilovací bloky obvodu na sobě stejnosměrně vázány, jejím úkolem je kompenzace offsetu a dá se nastavit nebo nulovat bitovou hodnotou v kontrolních registrech. Pokud bychom ji nechali vypnutou, při uváděném zesílení bloku způsobí malý vstupní offset 10 mV na výstupu hodnotu kolem 1 V, což znemožní efektivní rozsah dynamiky signálu, jejím zapnutím je tento problém vyřešen (je doporučeno nechat ji zapnutou). Na obr. 2.5 je vyobrazena architektura LNA bloku jedna. Blok LNA rozhoduje o výkonu šumu nejvíce z celého řetězce. V tab. 2.2 jsou uvedeny klíčové vlastnosti LNA bloku. Zisk bloku jako celku je dán vztahem:

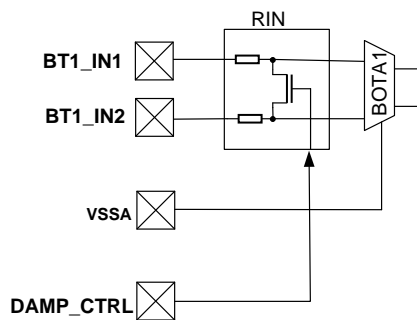
$$A_{\text{LNA}} = \frac{R_{\text{TIA}}}{R_{\text{gm}}} \quad (2.1)$$

Obr. 2.2: Architektura BOTA uvnitř programovatelného analogového obvodu (převzato z [10])

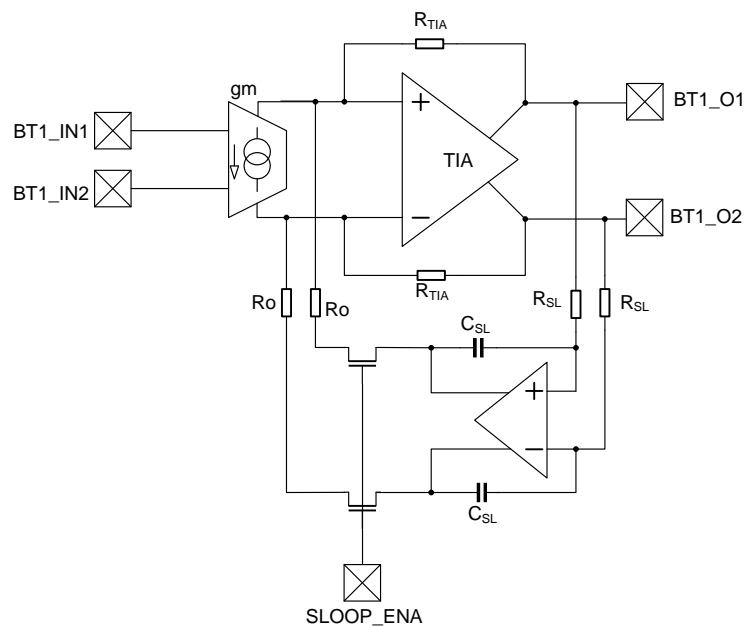
**Topologie zesilovače BTIA** Na obr. 2.3 je uvedena interní topologie zesilovače BTIA užíteho v dílčích blocích obvodu.



Obr. 2.3: Architektura BTIA uvnitř programovatelného analogového obvodu (převzato z [10])



Obr. 2.4: Redukce vstupní impedance před blokem jedna (upraveno podle [10])



Obr. 2.5: Architektura LNA bloku jedna (upraveno podle [10])

## VGA – blok dvě

VGA (Variable Gain Amplifier) je postaven opět na kombinaci klíčových prvků BOTA a BTIA s proměnným zesílením a skládá se ze 2 takových částí (pro identitu jsem je zařadil pod jeden blok se dvěma podbloky). Oba podbloky bloku VGA umožňují vysoké zesílení v rozdílu až 30 dB ve vysokém dynamickém rozsahu (jsou určeny pro řízení hlavního zesílení řetězce) díky nastavitelným transimpedancím a transkonduktancím dílčích částí. Oba VGA jsou propojeny interní sběrnici s kontrolními registry, která přes zapsanou hodnotu `VGA1_GainCode` a `VGA2_GainCode` umožňuje řízení zesílení obou zesilovačů nezávisle na sobě v rozsahu  $-10$  dB až 20 dB (lineárně v dB). Hodnoty kódu odpovídající příslušnému zesílení jsou uvedeny v tab. 2.4, dále jeho vlastnosti v tab. 2.3.

## FGA – blok tři

Podblok A2 je opět složený z dílčího BOTA a BTIA, tentokrát s fixním zesílením o hodnotě 20 dB (maximum zisku VGA). Výstup celého řetězce může tak dosáhnout maximálního rozkmitu výstupního napětí, pokud se jeden z bloků nedostane do své saturace. Klíčové vlastnosti bloku jsou uvedeny v tab. 2.5. Podblok A3 (konstrukčně FDCA) je proudový zesilovač s plně diferenciálním vstupem a výstupem (napětovým jako v případě ostatních bloků). Je opět složený z dílčích BOTA a BTIA, kde BTIA má binárně regulovatelnou transimpedanci o 5 dB nastavitelnou bitovou vlastností `SDBUFGAINp5dB` v poli kontrolních registrů. Vlastnosti implementovaného FDCA jsou uvedeny v tab. 2.6.

## SDM – blok čtyři

V řetězci je implementován  $\Sigma - \Delta$  modulátor druhého řádu pro přímý AD převod. Pracuje s hodinami 10 MHz přiváděnými externě na pinu CLK10M. Výstupem je pulzně hustotní modulace dostupná na pinu PDM. Referenční napětí 1,2 V je tvořeno interně. Vlastnosti zabudovaného SDM jsou uvedeny v tab. 2.7. Činnost je nejvíce názorná na omezeném výstupním průběhu při počátečním nediferenčním ohledávání PAO (viz obr. 2.7, kde **oranžový** průběh je výstupní napětí  $U_{OUT+}$  na pinu SDM\_IO\_A a **modrý** k tomu odpovídající PDM).

## Vnitřní signálový multiplex

Všechny bloky (tedy LNA, VGA, FGA a SDM) lze propojit pomocí implementovaných signálových přepínačů (s povahou multiplexeru). Dále lze ovládat další skupinu spínačů, kterou vyvádíme či přivádíme do řetězce pěti bloků externí signál, či pouze signál ještě jinak mezi bloky upravíme. Je zde implementovaných pět meziblokových a tři přiváděcí sady přepínačů pro individuální zvolení signálové cesty, dále dvě sady obracecích přepínačů. Meziblokové přepínače skýtají skupinu mezi blokem LNA (*LNA<sub>s1</sub>-SW*), mezi dvěma podbloky ve VGA (*VGA<sub>1s2</sub>-SW*), dále mezi blokem VGA a podblokem A2 bloku FGA (*VGA<sub>2s3</sub>-SW*), též mezi podbloky A2, A3 bloku FGA (*A2<sub>s4</sub>-SW*) a nakonec mezi podblokem A3 bloku FGA a SDM (*SDMB-SW*).



Každý mezipřepínač obsahuje pět spínačů říditelných přes vlastní 4bitovou hodnotu v kontrolních registrech. Vzhledem k tomu, že přes 4bitovou hodnotu řídíme kombinace pěti přepínačů, je jasné, že nejsou povoleny všechny kombinace (některé se zdají být i nesmyslné). Je dána tabulka kombinací, kde přesné kombinaci přísluší vlastní 4bitová hodnota v kontrolních registrech (tabulka kombinací je pro všech pět registrů identická). Hodnoty kombinací jsou uvedeny v tab. 2.8. Signálový přístup přes sadu přepínačů do jednotlivých bloků je obousměrný, tedy jsme schopni signál odebírat nebo přivádět a to v napěťové či proudové podobě.

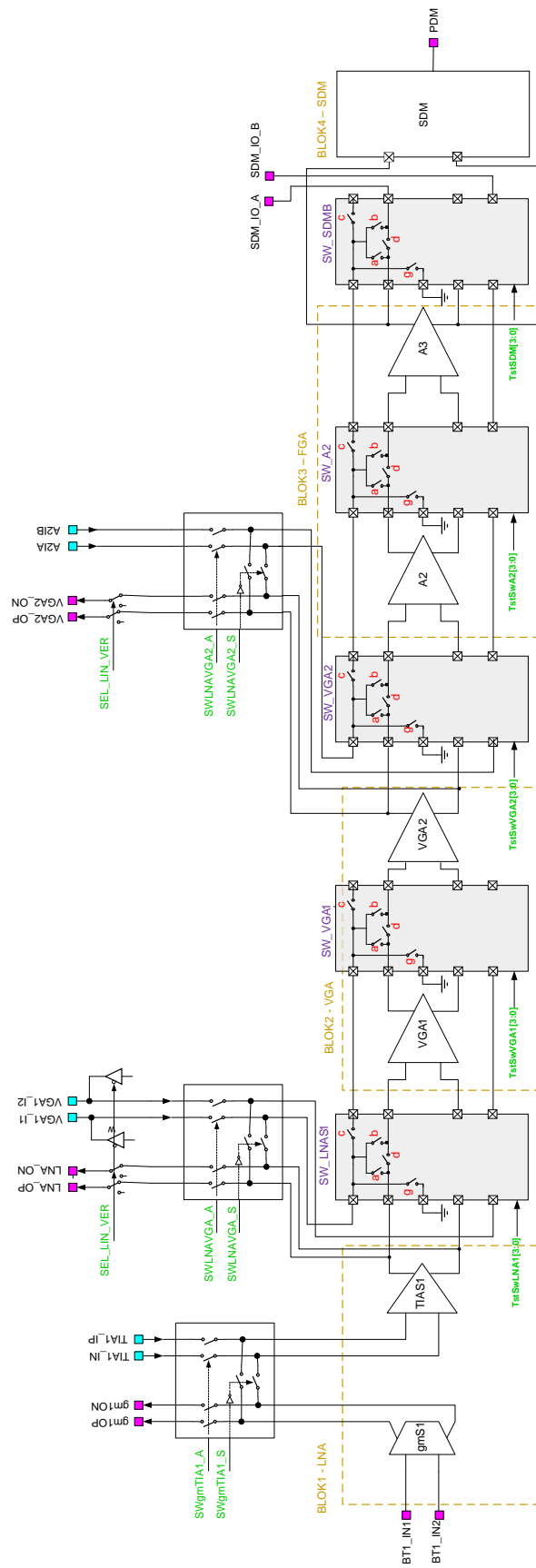
Přiváděcí sady přepínačů nám určují, zda-li rozpojíme signálovou cestu a vložíme mezi ni nějaký další prvek, či signál z předchozího bloku plně nahradíme signálem vlastním. Jsou umístěny mezi podbloky bloku LNA (*SWgmTIA1\_A*, *SWgmTIA1\_S*), mezi bloky LNA a VGA (*SWLNAVGA\_A*, *SWLNAVGA\_S*) a též mezi VGA a podblokem A2 bloku FGA (*SWLNAVGA2\_A*, *SWLNAVGA2\_S*). Jejich kombinace jsou dány přímo opět tentokrát samostatnými bitovými hodnotami v kontrolních registrech.

Obě skupiny obracečů přepínačů se dají ovládat jedním bitem *SEL\_LIN\_VER* umístěným v poli kontrolních registrů (slouží k odebrání signálu, jeho inverzi a následnému poslání zpět do řetězce). Jsou implementovány mezi bloky LNA, VGA a VGA, A2 (podblokem bloku FGA).

Komplexní schéma zobrazující rozložení celého analogového signálového multiplexeru v řetězci je uvedeno na obr. 2.6.

## Kontrolní registry

Pole kontrolních registrů je dostupné přes 8bitovou datovou sběrnici se 4bitovou adresovou sběrnici (adresace 0–13), obsahuje celkem čtrnáct registrů, které se dají pouze zapsat, nikoliv číst. Je třeba přiřadit konkrétní adresu, dále konkrétní data a signálem pro zápis  $\overline{WR} = \text{log.0}$  data zapsat, přičemž je doporučeno ponechat zapisovací impuls minimálně 60 ns v log. 0. Všechny vlastnosti celého PAO jsou přiřazeny jednotlivým registrům v poli, jejich rozvržení je uvedeno v tab. 2.9, kde jsou uvedeny pouze 6 B, které mají pro naše použití význam (ostatní vlastnosti souvisejí s dřívější funkcí obvodu a jsou pro naše účely neúčinné). Při zapisování neúčinných bytů hrozí nastavení kritických vlastností (např. vypínače napájení bloků), které až do budoucího resetu obvodu omezí nebo degraduje jeho funkce (z tohoto důvodu jsem se v nastavování omezil pouze na užitečné byty vlastností).



Obr. 2.6: Architektura vnitřního signálového multiplexu (upraveno podle [10])

Tab. 2.1: Popis pinů a jejich funkcí pro obvod

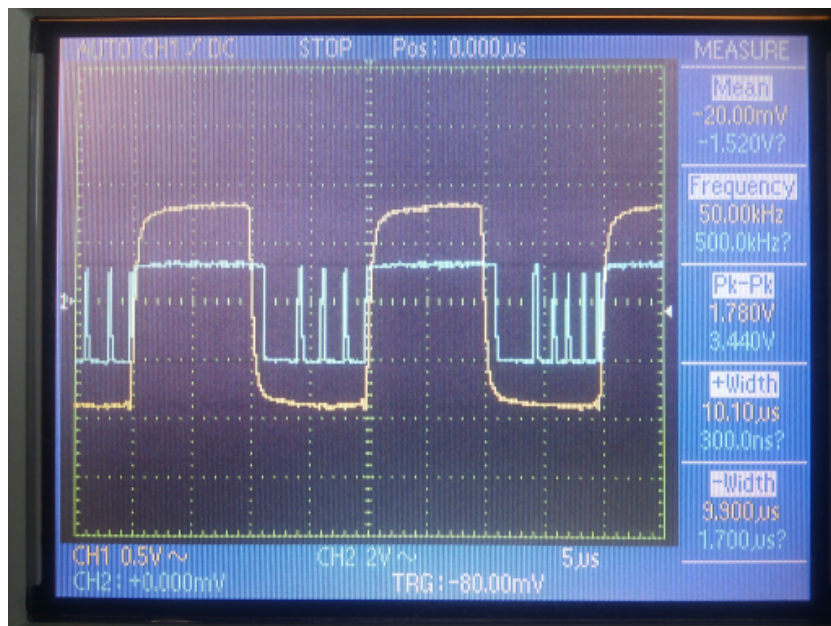
Číslo pinu	Název pinu	Popis funkce
1	BT1_IN1	Vstup BOTA negativní – blok 1
2	BT1_IN2	Vstup BOTA pozitivní – blok 1
3	VSSA1	Země BOTA – blok 1
4	VDDA	Výstup z vnitřního regulátoru 3,3 V
5	KNC6	Nepřipojeno a uzemněno
6	KNC5	Nepřipojeno a uzemněno
7	KNC4	Nepřipojeno a uzemněno
8	KNC3	Nepřipojeno a uzemněno
9	KNC2	Nepřipojeno a uzemněno
10	KNC1	Nepřipojeno a uzemněno
11	NC1	Nepřipojeno
12	KNC9	Nepřipojeno
13	GNDP	Digitální zem
14	KNC8	Nepřipojeno a uzemněno
15	KNC7	Nepřipojeno a uzemněno
16	VSUP	Napájecí pin +12V
17	VGA2_ON	Výstup z VGA2 negativní – blok 2
18	VGA2_OP	Výstup z VGA2 pozitivní – blok 2
19	KNC10	Nepřipojeno a uzemněno
20	KNC11	Nepřipojeno a uzemněno
21	KNC12	Nepřipojeno a uzemněno
22	KNC14	Nepřipojeno a uzemněno
23	KNC13	Nepřipojeno a uzemněno
24	ADR0	Adresový vstup 1 kontrolního registru
25	ADR1	Adresový vstup 2 kontrolního registru
26	ADR2	Adresový vstup 3 kontrolního registru
27	ADR3	Adresový vstup 4 kontrolního registru
28	RESET	Vstup resetu
29	KNC15	Nepřipojeno
30	KNC16	Nepřipojeno a uzemněno
31	NC2	Nepřipojeno
32	NC3	Nepřipojeno
33	LIN_PGND	Napájecí zem
34	KNC17	Nepřipojeno
35	NC4	Nepřipojeno
36	NC5	Nepřipojeno
37	A2IB	Vstup zesilovače A2 negativní – podblok bloku 3
38	NC6	Nepřipojeno
39	A2IA	Vstup zesilovače A2 pozitivní – podblok bloku 3
40	D0	Datový vstup 1 kontrolního registru
41	D1	Datový vstup 2 kontrolního registru
42	D2	Datový vstup 3 kontrolního registru
43	D3	Datový vstup 4 kontrolního registru
44	D4	Datový vstup 5 kontrolního registru
45	D5	Datový vstup 6 kontrolního registru
46	D6	Datový vstup 7 kontrolního registru
47	D7	Datový vstup 8 kontrolního registru
48	CLK10M	Vstup 10MHz hodin pro $\Sigma - \Delta$ modulátor – blok 4
49	PDM	Výstup PDM z $\Sigma - \Delta$ modulátoru (blok 4)
50	WR_NON (WR)	Vstupní signál pro zápis kontrolního registru (invertovaný)
51	VGA1_I2	Vstup z VGA2 pozitivní – blok 2
52	VGA1_I1	Vstup z VGA2 negativní – blok 2
53	PSUB	Země substrátu (digitální zem)
54	SDM_IO_A	Výstup z FDCA pozitivní (FGA) – blok 3
55	SDM_IO_B	Výstup z FDCA negativní (FGA) – blok 3
56	LNA_OP	Výstup pozitivní z bloku 1 (BOTA+BTIA)
57	LNA_ON	Výstup negativní z bloku 1 (BOTA+BTIA)
58	DAMP_CTRL	Vstup pro kontrolu rezistance $R_{IN}$ bloku
59	VSSA2	Analogová zem
60	TIA_IN	Vstup BTIA negativní – blok 1
61	TIA_IP	Vstup BTIA pozitivní – blok 1
62	GM1_ON	Výstup BOTA negativní – blok 1
63	GM1_OP	Výstup BOTA pozitivní – blok 1
64	VSSA3	Analogová zem

Tab. 2.2: Vlastnosti bloku LNA

Vlastnost	Parametr	Min	Typ	Max	Jednotka
BOTA_ZISK	BOTA hodnota transkonduktance	-	3	-	mS
BTIA_ZISK	BTIA hodnota transimpedance	-	30	-	k $\Omega$
LNA_ZISK	Zesílení bloku LNA v pásmu	36,1	39,1	42,1	dB
LNA_OFFSET	Hodnoty offsetu na výstupu LNA	-600	-	600	mV
LNA_BW_fl	$f_{\text{MIN}}$ hranice šířky pásma bloku LNA ( $-3$ dB)	20	-	-	kHz
LNA_BW_fh	$f_{\text{MAX}}$ hranice šířky pásma bloku LNA ( $-3$ dB)	150	-	-	kHz
LNA_NVSD	Maximální hodnota šumu na vstupu LNA pro $R_S = 1\text{k}\Omega$	-	5	-	$\text{nV}/\sqrt{\text{Hz}}$
LNA_THD	Celkové zkreslení harmonického signálu bloku LNA pro $U = 0,5\text{ V}_{\text{PP}}$ a $f = 80\text{ kHz}$	-	-	-30	dB
LNA_CMMO	Výstupní napětí bloku LNA pro souhlasnou hodnotu napětí na vstupech	1,2	-	2,2	V

Tab. 2.3: Vlastnosti bloku VGA

Symbol	Parameter	Min	Typ	Max	Jednotky
VGA_ZISK_MIN	Zesílení v pásmu pro kód 00	-7	-10	-11	dB
VGA_ZISK_MAX	Zesílení v pásmu pro kód 63	19	20	21	dB
VGA_ZISK_S	Krok zesílení	0,1	0,476	0,9	dB
VGA_OFFSET	Offsetové napětí vstupu	-	-	42	mV
VGA_BW_fl	$f_{\text{MIN}}$ šířky pásma ( $-3$ dB) pro kód 63	20	-	-	kHz
VGA_BW_fh	$f_{\text{MAX}}$ šířky pásma ( $-3$ dB) pro kód 63	150	-	-	kHz
VGA_N_PSD	Maximální vstupní hodnota šumu	-	500	-	$\text{nV}/\sqrt{\text{Hz}}$
VGA_LinRng	Harmonické zkreslení pro $U = 2,0\text{ V}$ a $f = 80\text{ kHz}$	-30	-	-	dB
VGA_CMMO	Výstupní napětí pro souhlasné signály na vstupech	1,5	-	1,8	V

Obr. 2.7: Ukázka funkce implementovaného  $\Sigma - \Delta$  modulátoru

Tab. 2.4: Kód zisku VGA podle hodnoty v kontrolním registru

Kód zisku VGA číslo [5:0]	Odpovídající zisk zisk [dB]	Kód zisku VGA číslo [5:0]	Odpovídající zisk zisk [dB]
0	-10,000	32	5,232
1	-9,524	33	5,708
2	-9,048	34	6,184
3	-8,572	35	6,660
4	-8,096	36	7,136
5	-7,62	37	7,612
6	-7,144	38	8,088
7	-6,668	39	8,564
8	-6,192	40	9,040
9	-5,716	41	9,516
10	-5,240	42	9,992
11	-4,764	43	10,468
12	-4,288	44	10,944
13	-3,812	45	11,420
14	-3,336	46	11,896
15	-2,860	47	12,372
16	-2,384	48	12,848
17	-1,908	49	13,324
18	-1,432	50	13,800
19	-0,956	51	14,276
20	-0,480	52	14,752
21	-0,004	53	15,228
22	0,472	54	15,704
23	0,948	55	16,180
24	1,424	56	16,656
25	1,900	57	17,132
26	2,376	58	17,608
27	2,852	59	18,084
28	3,328	60	18,560
29	3,804	61	19,036
30	4,280	62	19,512
31	4,756	63	19,988

Tab. 2.5: Vlastnosti podbloku A2 bloku FGA

Vlastnost	Parametr	Min	Typ	Max	Jednotky
A2_ZISK	Zesílení v pásmu	17,8	20,8	23,8	dB
A2_OFFSET	Výstupní offsetové napětí	-600	-	600	mV
A2_BW_fl	$f_{\text{MIN}}$ šířky pásma (-3 dB)	15	-	-	kHz
A2_BW_fh	$f_{\text{MAX}}$ šířky pásma (-3 dB)	160	-	-	kHz
A2_THD	Celkové harmonické zkreslení pro $U = 1 \text{ V}_{\text{PP}}$ a $f = 80 \text{ kHz}$	-	-	-30	dB
A2_CMMO	Rozsah výstupního napětí	1,2	-	2,2	V

Tab. 2.6: Vlastnosti podbloku A3 bloku FGA

Vlastnost	Parametr	Min	Typ	Max	Jednotky
A3_ZISK	Zisk v pásmu pro SDBUFFGAINp5dB = OFF	4	6	8	dB
A3_ZISK	Zisk v pásmu pro SDBUFFGAINp5dB = ON	8,8	10,8	12,8	dB
A3_OFFSET	Offset výstupu (pro SDBUFFGAINp5dB = OFF)	-40	-	40	mV
A3_OFFSET	Offset výstupu (pro SDBUFFGAINp5dB = ON)	-78	-	78	mV
A3_BW_fl	$f_{\text{MIN}}$ šířky pásma(-3 dB)	15	-	-	kHz
A3_BW_fh	$f_{\text{MAX}}$ šířky pásma(-3 dB)	160	-	-	kHz
A3_THD	Celkové harmonické zkreslení pro $U = 2,42 V_{\text{PP}}$ a $f = 80 \text{ kHz}$	-	-	-30	dB
A3_CMMO	Výstupní napětí pro napěťový mód	1,2	-	2,2	V

Tab. 2.7: Vlastnosti  $\Sigma - \Delta$  modulátoru

Vlastnost $\Sigma - \Delta$ převodníku	Hodnota	
Vzorkovací frekvence	10	MHz
Odstup signál - kvantizační šum	72	dB
Diferenciální nelinearita	$\pm 0,5$	LSB
Integrální nelinearita	$\pm 2$	LSB
Referenční napětí	1,2	V
Plný rozsah SDM	2,4	$V_{\text{PP}}$
Šířka pásma signálové přenosové funkce	75	kHz

Tab. 2.8: Kombinace meziblokových sad přepínačů pro řídicí hodnoty

Číslo v kontrolním registru [0:3]				Kombinace na nastavovaných přepínačích				
SWctrl3	SWctrl2	SWctrl1	SWctrl0	sw a	sw b	sw c	sw d	sw g
0	0	0	0	OFF	OFF	OFF	ON	ON
0	0	0	1	ON	OFF	OFF	ON	OFF
0	0	1	0	OFF	ON	OFF	ON	OFF
0	0	1	1	ON	ON	OFF	ON	OFF
0	1	0	0	OFF	OFF	ON	ON	OFF
0	1	0	1	ON	OFF	ON	ON	OFF
0	1	1	0	OFF	ON	ON	ON	OFF
0	1	1	1	ON	ON	ON	ON	OFF
1	0	0	0	OFF	OFF	OFF	OFF	ON
1	0	0	1	ON	OFF	OFF	OFF	OFF
1	0	1	0	OFF	ON	OFF	OFF	OFF
1	0	1	1	ON	ON	OFF	OFF	OFF
1	1	0	0	OFF	OFF	ON	OFF	OFF
1	1	0	1	ON	OFF	ON	OFF	OFF
1	1	1	0	OFF	ON	ON	OFF	OFF
1	1	1	1	ON	ON	ON	OFF	OFF

Tab. 2.9: Význam užitečných hodnot v poli kontrolních registrů

Byte	Rozsah	Vlastnost	Výchozí hodnota	Popis
DATA_05	7	CLP_OFF	0	
	6	SERVO_ENAS1	1	Servo-smyčky pro blok LNA
	5:0	VGA_1GainCode[5:0]	000000	Nastavení zesílení podbloku VGA1
DATA_06	7	SDBUFp5dB	0	Zvýšení zisku A3 o +5 dB
	6	SERVO_ENAS2	1	Servo-smyčky pro podblok VGA1
	5:0	VGA_2GainCode[5:0]	000000	Nastavení zesílení podbloku VGA2
DATA_08	7:4	TstswA2[3:0]	0000	Nastavení přepínačů SW_A2
	3:0	TstswLNA1[3:0]	0000	Nastavení přepínačů SW_LNAS1
DATA_09	7:4	TstswVGA2[3:0]	0000	Nastavení přepínačů SW_VGA2
	3:0	TstswVGA1[3:0]	0000	Nastavení přepínačů SW_VGA1
DATA_10	7	n.u.		
	6	POR_Test_Ena	0	-
	5	POR_TST_IN_TMA	0	-
	4	TstBuff_Pass	0	-
	3:0	TstSDM[3:0]	0000	Nastavení přepínačů SW_SDMB
DATA_14	7	PD_LIN	0	Kritická vlastnost – degraduje SEL_LIN_ver
	6	SEL_LIN_ver	1	Nastavení obracečů signálu
	5	SWVGAA2_A	0	Nastavení přepínačů pro A2IA A2IB
	4	SWVGAA2_S	0	Nastavení přepínačů pro A2IA A2IB
	3	SWLNA_VGA_A	0	Nastavení přepínačů pro LNA-BTIA
	2	SWLNA_VGA_S	1	Nastavení přepínačů pro LNA-BTIA
	1	SWgmTIA1_A	1	Nastavení přepínačů pro LNA
	0	SWgmTIA1_S	0	Nastavení přepínačů pro LNA

## 3 NÁVRH SOFTWARE

### 3.1 Ovládací program pro PC

#### 3.1.1 Úvahy nad řešením ovládací aplikace

##### Implementace do prostředí LabView

Cílem by bylo vytvoření programového modulu v prostředí LabView a přípravku, přes který by bylo možno nastavovat a měřit vlastnosti PAO. Řešení by bylo velice univerzální pro vědecké použití, avšak mým cílem bylo zvládnutí programování v jazyce C# i základů komunikace s PAO na desce. Od této chvíle se řešení ubíralo cestou tvorby vlastní nastavovací aplikace. Další myšlenkou bylo, aby PAO byl schopen připojený přes mikrokontrolér schopen zapamatovat si svoje vlastnosti.

##### Vlastní ovládací aplikace

Další úvaha počítala s propojením vlastní desky s PAO paralelním portem, který by ovládal mikrokontrolér umístěný na desce, s ovládací aplikací v jazyce C#. Nedostatkem je dnešní nedostupnost paralelních portů na základních deskách PC a nutnost řešení dokoupeným převodníkem USB-paralelní port. Snažil jsem se vytvořit přípravek s implementovanou verzí virtuálního zařízení USB (s pomocí knihovny V-USB, více informací na <http://www.obdev.at/>), který by se na USB hostu PC hlásil jako nové zařízení, přes které by bylo možno řídit PAO. Bylo sestaveno pokusné zapojení pro test zapojení, avšak řešení bylo shledáno nevhodným, protože přísnosti standardu USB jednak moje pokusné zapojení nevyhovovalo a za druhé bylo náchylné na minimální změnu jakéhokoliv parametru použitých součástek. Další úvahou bylo použití sériového portu, u něhož jsem zůstal s použitím převodníku FT232BL (později FT232RL), který na USB sběrnici zpřístupní virtuální COM port. Na použitém mikrokontroléru pak běží USART, který logicky komunikuje s vlastní aplikací. Mikrokontrolér pak podle pokynů nastavuje PAO. Vizualizace nastavení obvodu je implementována na ovládacím animovaném schématu.

#### 3.1.2 Okénko do objektově orientovaného programování

Při návrhu ovládací aplikace jsem se rozhodl použít jazyk C# a techniku objektově orientovaného programování. Mezi hlavní rozdíly procedurálního a objektového programování patří zavedení objektů, zapouzdření, dědičnosti a polymorfismus. Smyslem objektů je na myšlenku či určitý princip, které přetváříme do kódu vztáhnout model z reálného světa, kdy může mít určitá věc k sobě přidružené vlastnosti, které vyjadřují, v jakém stavu se právě nachází a dále má přidružené metody, kterými jsme schopni její stav nějak změnit (k těmto vlastnostem přistupujeme nejlépe pouze prostřednictvím metod). Třída je potom taková šablona k vytvoření objektu. Při tvorbě nové instance třídy (jejího nového objektu) voláme zpravidla konstruktor třídy (její „první“ metodu), která může vlastnosti nového



objektu nastavit do nějakého výchozího stavu (odstranění nepoužívaného objektu je automatické pomocí „třídiče odpadků“). Událost třídy je potom datový prostředek, kterým třída informuje svoje okolí o změně svého stavu (při vyvolání tohoto prostředku může sebou přenášet nějaké parametry), okolí potom může na událost nějak reagovat. Vlastností je rozuměna funkce, která veřejně zpřístupní vnitřní proměnné třídy (možnost jejich zápisu či čtení). Principem zapouzdření rozumíme skrytí implementačních detailů (tedy že programátora nemusí zajímat, jak daná metoda pracuje uvnitř, pouze víme, co vykoná a co do ní případně vložit za případné parametry, popřípadě co od ní očekávat za návratovou hodnotu). Při implementaci několika prvků, které mohou mít společné vlastnosti či metody (nebo se s nimi pracuje stejným způsobem) můžeme využít princip dědičnosti, kdy vytvoříme určitý základní polotovar (třidu), kterou mohou budoucí třídy s podobnými vlastnostmi zdědit a využít tak její vlastnosti a metody po zdědění přístupné. Polymorfismus potom zajišťuje možnost zacházet s podobnými objekty stejně. Více velice názorně v [11].

### 3.1.3 Architektura ovládací aplikace

Ovládací aplikace je koncipována ve stylu objektově-orientovaného programování. Skládá se ze dvou hlavních tříd – **UsbCom** a **PAOProperties** a dvou pomocných tříd – **Profiler** a **InfoConsole**. Z hlavních tříd – třída **UsbCom** zajišťuje svými metodami přístup k sériové komunikaci přes programové rozhraní s FT232BL sériovým rozhraním, zatímco třída **PAOProperties** zastřešuje vlastnosti nastavovaného obvodu. Pro pomocné třídy – třída **Profiler** obsahuje mechanismus pro uložení či načtení určité konfigurace nastavení a třída **InfoConsole** zajišťuje kontrolu nad textovou konzolou pro výpis provedených akcí. Třída **UsbCom** obsahuje objekty tříd **PAOProperties** a **Profiler** a třída **Form1** obsahuje instance tříd **UsbCom** a **InfoConsole** (čímž je dosažena jistá hierarchičnost co se týče obsluhy událostí v programu). Popis obsažené třídy **MsgBoxMod** zde záměrně není uveden, protože se jedná pouze o moji stylistickou úpravu původního **MessageBoxu**.<sup>1</sup>

### 3.1.4 Třída **PAOProperties**

#### Události

**PropertiesWillBeChanged** Pokud uživatel načte akcí ve třídě **Form1** slovník bytů uložený v pomocném mikrokontroléru **ATmega88**, zavolá se tato událost, která následně ve třídě **UsbCom** využije metodu **Dictionary2Properties**, která přeloží 14 B slovníku na jednobitové a vícebitové vlastnosti **PAO**. Událost slouží jako most, přes který zavoláme událost ze třídy **Form1**, která se ale provede v instanci třídy **UsbCom**. Událost nepřenáší žádný parametr.

---

<sup>1</sup>Veškeré události v programu obsahují za svým jménem postfix „Status“, který zde není uveden (tedy **PropertiesWillBeChanged** → **PropertiesWillBeChangedStatus**)

**PaoPropertiesChanged** Pokud uživatel ve ovládacím schématu změní jakoukoliv vizualizovanou vlastnost PAO, provede se tato událost, která s sebou zavolá metodu `BuildDictionary`, která vytvoří z vlastností PAO slovník 14 B, který dále posíláme do mikrokontroléru. Událost obsahuje parametr vytvořeného slovníku.

## Konstruktor

Konstruktor `public PAOProperties()` nastaví všechny vlastnosti do jejich původních hodnot, tak, jak je to uvedeno ve specifikaci obvodu. Dále sestaví slovník kombinací přepínačů analogového signálového multiplexeru k datovým slovům pro hodnoty v kontrolních registrech.

## Vlastnosti

Obsahuje prototypy všech vlastností PAO, které jsme schopni nastavovat. Lze přečíst či zapsat jejich hodnotu. Zápis jejich hodnot je svázán s událostí `PaoPropertiesChanged`. Příklad takové vlastnosti:

```
public bool SERVO_ENAS2 //servo-smýčka2
{
    get
    {
        return this.SERVO_ENAS2_; //vrať hodnotu privátní boolovské proměnné SERVO_ENAS2_
    }
    set
    {
        this.SERVO_ENAS2_ = value; //zapiš její hodnotu
        //zavolej událost PAOPropertiesChanded
        PAOPropertiesChandedStatusMake(new PAOPropertiesChandedEventArgs(BuildDictionary←
            ());
    }
}
```

## Hlavní metody

**Dictionary2Properties** Metoda vezme slovník 14 B a dekoduje z nich veškeré vlastnosti PAO (např. *TxCPL*, *SERVO\_ENAS1*, *SERVO\_ENAS2*, apod.). Pokud nahrajeme uložená data v mikrokontroléru do programu, zavolá se tato metoda, která přeloží slovník bytů pro nastavení ovládacího schématu.

**MuxSwCombinationToSettingByte** Metoda se volá vždy při změně nastavení určitého přepínače, který je součástí signálového multiplexeru, a zkontroluje, zda-li nastavená kombinace vyhovuje platným kombinacím. Pokud vyhovuje, zapíše vlastnost do příslušného pole bitů a vrátí hodnotu `true`, pokud nevyhovuje, vrátí pouze `false`.

**MuxCombToSw** Metoda pro bytovou hodnotu v parametru vrací kombinaci přepínačů pro analogový multiplexer ve formátu pěti boolovských proměnných zapouzdřených do třídy MuxSettingsByteToSwitch.

**BuildDictionary** Metoda sestaví z vlastností PAO 14 B slovník pro programovací mikrokontrolér. Metoda se volá vždy, pokud ve formuláři něco změníme.

### 3.1.5 Třída UsbCom

#### Události

Všechny události krom dvou níže uvedených odesílají do instance třídy Form1 svůj parametr, inicializaci struktury TransferRes, a jsou čistě informativního charakteru (pro programovou konzolu). Asociace jednotlivých prvků této struktury jednotlivými událostmi je uvedena v tab. 3.1 (v pořadí typ příkazu, čas odeslání příkazu, dva jeho parametry, výsledek).

**Reakce na odeslání programového rámce příkazu** Pod reakcemi na odeslání programového rámce příkazu rozumíme události SendingByteToRAM, SendingByteToEEPROM, RecievingByteFromEEPROM, RecievingByteFromRAM, ProgrammingPAO, CircuitEchoing. Událost je vyvolána vždy, když je vyslán programový rámec metodou UsbSendData do programovacího mikrokontroléru.

**Reakce na přijetí rámce odpovědi** Pod reakcí na rámec odpovědi rozumíme volání události SendedByteToRAM, SendedByteToEEPROM, RecievedByteFromEEPROM, RecievedByteFromRAM, ProgrammedPAO, CircuitEchoed. Událost je vyvolána vždy, když metoda UsbSendData vrátí výsledek prováděné akce.

Tab. 3.1: Využití parametrů informační struktury TransRes

Událost	CommandTypeX	EventTimeX	adX	valX	resX
SendingByteToRAM	X	X	X	-	-
SendedByteToRAM	X	X	X	X	X
SendingByteToEEPROM	X	X	X	-	-
SendedByteToEEPROM	X	X	X	X	X
RecievingByteFromEEPROM	X	X	X	-	-
RecievedByteFromEEPROM	X	X	X	X	X
RecievingByteFromRAM	X	X	X	-	-
RecievedByteFromRAM	X	X	X	X	X
ProgrammingPAO	X	X	-	-	-
ProgrammedPAO	X	X	-	-	X
CircuitEchoing	X	X	-	-	-
CircuitEchoed	X	X	-	-	X

Dále třída UsbCom obsahuje neparametrické události, které mají určitý výkonný charakter.

**ReloadThumbinals** Událost poskytuje pověření pro opětovné vykreslení seznamu jedno a více bitových vlastností v instanci třídy **Form1**, je volána v souběhu s událostí **PAOPropertiesChanged** ze třídy **PAOProperties**.

**ReloadScheme** Tato událost poskytne podnět pro překreslení nastavovacího schématu podle aktuálních vlastností PAO, opět je svázána s událostí **PropertiesWillBeReloaded** ve třídě **PAOProperties**.

**EmptyDeviceHandler** Pokud uživatel po startu programu nevybere pro komunikaci žádné zařízení, nebo je-li při aktivní komunikaci s programovacím zařízením vnějším zásahem přerušeno spojení (např. odpojen kabel), je zavolána tato událost, která vypíše chybovou zprávu do programové konzole a zablokuje další akce (vykonávání operací nad programovacím zařízením).

**GetConnectedDevicesHandler** Když uživatel obnoví seznam připojených zařízení, je zavolána tato událost, která vrací jako parametr instanci třídy **FindedDevices**. Tato instance obsahuje strukturu **ListOfFTDIs** s popisem připojených zařízení (jméno, sériové číslo, ID, adresa a index) pomocí nichž lze se zařízením komunikovat. Dále obsahuje logické proměnné pro identifikaci důvodu případného neúspěchu operace s ovladačem.

**LoadProgressBarInfo** V případě vykonávání určité operace nad programovacím zařízením (např. odesílání dat do jeho EEPROMky) je volána tato událost, která nese ve svém parametru údaj o čísle zpracovávaného bytu a úspěchu operace nad ním. Na základě těchto údajů je pak vykreslován údaj progressBaru v okně programu.

**ProtocolCrash** V těle metody **UsbSendData** pokud se čeká na počet 5 bytů rámce odpovědi v zásobníku více jak definovaný počet sekund (nyní 10), zavolá se tato událost, přes kterou následně při její obsluze dojde k výzvě uživatele k restartu zařízení a ukončení programu (program tedy nezůstane v případě chyby vyset).

Odesílací programový rámec:

0xFF	Požadavek	Adresa	Hodnota	0xFF
------	-----------	--------	---------	------

Návratový programový rámec:

msgOK msgErr	Data1	Data2	Data3	Data4
-----------------	-------	-------	-------	-------

Obr. 3.1: Sériový rámec odesílaný do programovacího mikrokontroléru

## Hlavní metody

Většina metod obražená v třídě `UsbCom` je určena k odeslání nebo příjmu programového rámce přes stěžejní metodu `UsbSendData`. Koncepce byla zvolena taková, že sdružováním elementárních metod, které umí od začátku vyslat do cílového zařízení skupinu bytů nebo ji přečíst, získáme metody složitější, avšak na případné odladování chyb přijatelnější. Dílčí metody pro přenos dat `SendSingleDataToRAM`, `SendSingleDataToEEPROM`, `GetSingleDataFromEEPROM`, `GetSingleDataFromRAM`, mají v základu stejný princip funkce, pouze sestavují jiný programový odesílaný rámec. Na obr. 3.1 jsou uvedeny náznaky sériových rámců posílaných do programovacího mikrokontroléru.

Za dílčí metody určené pro přenos dat považuji `SendSingleDataToRAM` a `GetSingleDataFromRAM` (princip je stále stejný):

```
private bool SendSingleDataToRAM(byte address)
{
    bool result = false;
    //zásobník pro data posílaná zpět mikrokontrolérem
    List<Byte> buffer = new List<Byte>();
    //pokud je v slovníku vlastností taková adresa obsažena
    if (BytesMemory.ContainsKey(address))
    {
        //událost — posílám byte adresy address
        SendingByteToRAMHandlerStatusMake(new SendingByteToRAMEventArgs(new TransferRes((←
            int)CommandTypes.WriteByteToRAM, DateTime.Now, address, BytesMemory[address])));
        //odešli programový rámec
        if (UsbSendData((byte)CommandTypes.WriteByteToRAM, address, BytesMemory[address], ←
            out buffer, baudrate, DataBits, StopBits, Parity, FlowCtrlType, FCtrlXon, ←
            FCtrlXoff, ReadTimeout, WriteTimeout) == true)
        {
            //pokud data dorazila v pořádku, můžeme si je nechat odeslat zpět a porovnat s ←
            daty odesílanými.
            result = ((buffer[3] == address) && (buffer[2] == BytesMemory[address])); // buffer←
            [2] = INDEX, buffer[1]=DATA
        }
        else
        {
            AddError((int)Errors.UsbSendDataFunctionFailed, "SendSingleDataToRAM", String.Empty←
            );
            result = false;
        }
        //událost — poslal jsem byte adresy address s úspěchem result
        SendByteToRAMHandlerStatusMake(new SendByteToRAMEventArgs(new TransferRes((←
            int)CommandTypes.WriteByteToRAM, DateTime.Now, address, BytesMemory[address], ←
            result)));
    }
    else {
        AddError((int)Errors.NotExistKeyInDictionary, "SendSingleDataToRAM", String.Empty);
        result = false;
    }
    return result;
}

//druhá metoda v pořadí
```

```

private bool GetSingleDataFromRAM(byte address)
{
    bool result = false;
    //zásobník pro data posílaná zpět mikrokontrolérem
    List<Byte> buffer = new List<Byte>();
    //událost — získávám byte adresy address
    RecivingByteFromRAMHandlerStatusMake(new RecivingByteFromRAMEventArgs(new ←
        TransferRes((int)CommandTypes.ReadByteFromRAM,DateTime.Now,address)));
    //odeslání programového rámce
    if (UsbSendData((byte)CommandTypes.ReadByteFromRAM, address,0, out buffer, ←
        baudrate, DataBits, StopBits, Parity, FlowCtrlType, FCtrlXon, FCtrlXoff, ←
        ReadTimeOut, WriteTimeOut) == true)
    {
        //pokud data dorazily v pořádku, buffer[3] bude obsahovat negovaný obsah buffer←
        [2]
        if (buffer[2] == NegateByte(buffer[3]))
        {
            BytesMemory[address] = (byte)buffer[2];
            result = true;
        }
        else
        {
            AddError((int)Errors.ControlNegationTestFailed, "GetSingleDataFromRAM",String.←
                Empty);
        }
        //událost — získal jsem byte adresy address s hodnotou buffer[2]
        ReceivedByteFromRAMHandlerStatusMake(new ReceivedByteFromRAMEventArgs(new ←
            TransferRes((int)CommandTypes.ReadByteFromRAM,DateTime.Now,address,BytesMemory←
            [address],result)));
    }
    else
    {
        AddError((int)Errors.UsbSendDataFunctionFailed, "GetSingleDataFromRAM",String.←
            Empty);
        result = false;
    }
    return result;
}

```

Třída `UsbCom` dále obsahuje metodu pro příkaz programování PAO mikrokontrolérem `ProgrammPAOFromRAM` a metodu pro test správné komunikace s mikrokontrolérem `CircuitEcho` (popis kódu zde již neuvádím, protože je opět principiálně stejný s metodami uvedenými výše).

Další důležitou metodou pro komunikaci s obvodem FT232BL (později FT232RL) je metoda `UsbSendData`. Preferoval jsem komunikaci přes přímé ovladače, proto má metoda takovouto strukturu:

```

public bool UsbSendData(byte request, byte address, byte value, out List<byte> ←
    vystup, uint baudrate, byte DataBits, byte StopBits, byte Parity, ushort ←
    FlowCtrlType, byte FCtrlXon,byte FCtrlXoff, uint ReadTimeOut, uint ←
    WriteTimeOut)
{
    //5-bytový protokol

```

```

bool repeat = false;
int countOfRepeat = 0;

bool returnval = true;
vystup = new List<byte>();
listofErrX.Clear();
FTDI.FT_STATUS ftStatus = FTDI.FT_STATUS.FT_OK;

UInt32 ftdiDeviceCount = 0;

//je nějaké zařízení připojeno?
ftStatus = myFtdiDevice.GetNumberOfDevices(ref ftdiDeviceCount);
if (ftStatus == FTDI.FT_STATUS.FT_OK)
{
    if ((ftdiDeviceCount == 0) && (ActiveDeviceLoc != 999999))
    {
        //pokud uživatel nevybral zařízení nebo prostě není nic připojené
        EmptyDeviceHandlerStatusMake();
    }

    if (ftdiDeviceCount != 0)
    {
        FTDI.FT_DEVICE_INFO_NODE[] ftdiDeviceList = new FTDI.FT_DEVICE_INFO_NODE[←
            ftdiDeviceCount];

        ftStatus = myFtdiDevice.GetDeviceList(ftdiDeviceList);
        if (ftStatus == FTDI.FT_STATUS.FT_OK)
        {
            if (DeviceIsSet == true)
            {
                //otevři zařízení podle adresy
                ftStatus = myFtdiDevice.OpenByLocation(ActiveDeviceLoc);

                if (ftStatus == FTDI.FT_STATUS.FT_OK)
                {
                    //nastav symbolovou rychlost přenosu
                    ftStatus = myFtdiDevice.SetBaudRate(baudrate);
                    if (ftStatus == FTDI.FT_STATUS.FT_OK)
                    {
                        //nastav charakteristiku dat – počet datových, stopbitů a paritu
                        ftStatus = myFtdiDevice.SetDataCharacteristics(DataBits, StopBits, Parity);
                        if (ftStatus == FTDI.FT_STATUS.FT_OK)
                        {
                            //nastav řízení toku a handshaking
                            ftStatus = myFtdiDevice.SetFlowControl(FlowCtrlType, FCtrlXon, FCtrlXoff);
                            if (ftStatus == FTDI.FT_STATUS.FT_OK)
                            {
                                //jak dlouho budeme čekat až se buffery naplní daty
                                ftStatus = myFtdiDevice.SetTimeouts(ReadTimeOut, WriteTimeOut);
                                if (ftStatus == FTDI.FT_STATUS.FT_OK)
                                {
                                    //pokud vznikne chyba, 3x opakuj, pak ukonči s chybou
                                    REPEAT:
                                    UInt32 numBytesWritten = 0;
                                    //zapiš programový rámec do převodníku
                                    ftStatus = myFtdiDevice.Write(new byte[] { 0xFF, request, adress, value, 0xFF }, ←
                                        (new byte[] { 0xFF, request, adress, value, 0xFF })).Length, ref ←

```

```

        numBytesWritten);
    if (ftStatus == FTDI.FT_STATUS.FT_OK)
    {
        UInt32 numBytesAvailable = 0;
        //zaznamenej si čas, od kdy začínáš testovat Rx zásobník FT232BL
        double checkval = ConvertToUnixTimestamp(DateTime.Now);
        bool crash = false;
        //čekej v aktivní smyčce, dokud nebude připraveno k příjmu 5 bytů programového ↵
        rámce
    do
    {
        ftStatus = myFtdiDevice.GetRxBytesAvailable(ref numBytesAvailable);
        if (ftStatus == FTDI.FT_STATUS.FT_OK)
        {

        }
        else {

        }
    }
    Thread.Sleep(10);

    if ((checkval + 5) < ConvertToUnixTimestamp(DateTime.Now))
    {
        //standardně je tam něco za méně jak 500 ms .... ukonči čekání
        crash = true;
        break;
    }

} while (numBytesAvailable < (new byte[] { 0xFF, request, adress, value, 0xFF }).↵
    Length);

if (crash == true)
{
    //prostřednictvím události shod aplikaci a ulož aktuální nastavení schématu
    ProtocolCrashStatusMake();
    returnval = false;
    crash = false;
}

byte[] readData = new byte[] { 0x00, 0x00, 0x00, 0x00, 0x00 };
byte lineStatus = 0x00;
FTDI.FT_STATUS errorCheck_ftStatus = FTDI.FT_STATUS.FT_OK;

UInt32 numBytesRead = 0;
//data jsou připravena, přečti je
errorCheck_ftStatus = myFtdiDevice.GetLineStatus(ref lineStatus);
ftStatus = myFtdiDevice.Read(readData, numBytesAvailable, ref numBytesRead);
if ((ftStatus == FTDI.FT_STATUS.FT_OK) && (errorCheck_ftStatus == FTDI.FT_STATUS.↵
    FT_OK))
{
    if (GetSingleBitFromByte(lineStatus, 1) || GetSingleBitFromByte(lineStatus, 2) || ↵
        GetSingleBitFromByte(lineStatus, 3))
    {
        //atmel poslal data s chybou parit nebo rámce nebo přetečení
        AddError((byte)Errors.UsbDeviceReadErrFromAtmel, "UsbSendData", ftStatus.ToString↵
            ());
        returnval = false;
    }
}

```



```

}
else{
//atmel poslal data správně
if (readData[1] == (byte)MsgReturns.msgOK)
{
//data dorazila do mikrokontroléru v pořádku (bez chyby), výskyt chyby byl ←
  minimální ..... vše úspěšné .... připrav výstup
for (int i = 0; i < readData.Length; i++)
{
vystup.Add(readData[i]);
}
}
else
{
if (readData[1] == (byte)MsgReturns.msgErr)
{
//vznikl problém, při příjmu do mikrokontroléru (parita, rámeček, jiná chyba), ←
  opakuj ...
countOfRepeat++;
if (repeat == false)
{
repeat = true;
}
else
{
if (countOfRepeat > 3)
{
//chyba se opakuje víc jak 3x .... ukonči s chybou
repeat = false;
}
}
}
if (repeat == true)
{
goto REPEAT;
}
//přidej chybu do Podrobného výpisu chyb (pro DEBUG režim)
AddError((byte)Errors.UsbDeviceReadFrameSendErr, "UsbSendData", ftStatus.ToString()←
  ());
returnval = false;
}
}
ftStatus = myFtdiDevice.Close();
if (ftStatus != FTDI.FT_STATUS.FT_OK)
{
AddError((int)Errors.UsbDeviceClosingFailed, "UsbSendData", ftStatus.ToString());
returnval = false;
}
}
else
{
AddError((int)Errors.UsbDeviceReadFailed, "UsbSendData", ftStatus.ToString());
returnval = false;
}
}
else
{
AddError((int)Errors.UsbDeviceWriteFailed, "UsbSendData", ftStatus.ToString());

```

```

returnval = false;
}
}
else
{
returnval = false;
AddError((int)Errors.UsbDeviceTimeoutValFailed, "UsbSendData", ftStatus.ToString())←
);
}
}
else
{
returnval = false;
AddError((int)Errors.UsbDeviceFlowCtrlFailed, "UsbSendData", ftStatus.ToString());
}
}
else
{
returnval = false;
AddError((int)Errors.UsbDeviceCharacterFailed, "UsbSendData", ftStatus.ToString())←
;
}
}
else
{
returnval = false;
AddError((int)Errors.UsbDeviceBaudrateFailed, "UsbSendData", ftStatus.ToString());
}
}
else
{
returnval = false;
AddError((int)Errors.CouldntOpenUsbDevice, "UsbSendData", ftStatus.ToString());
}
}
else {
//není vybrané žádné zařízení
EmptyDeviceHandlerStatusMake();
}
}
else {
returnval= false;
AddError((int)Errors.CouldntDetectUsbDevice, "UsbSendData", ftStatus.ToString());
}
}
else {
returnval= false;
AddError((int)Errors.CouldntDetectUsbDevice, "UsbSendData", ftStatus.ToString());
}
}
else
{
returnval= false;
AddError((int)Errors.CouldntDetectUsbDevice, "UsbSendData", ftStatus.ToString());
}
}
return returnval;
}

```

Poslední důležitou metodou pro vyhledání připojených FTDI zařízení je metoda `FindFTDIDevice`. Pomocí svého výstupu umožňuje uživateli vybrat se kterým zařízením se bude pracovat a tím rozšiřuje program na podporu více zařízení:

```
public void FindFTDIDevice()
{
    UInt32 ftdiDeviceCount = 0;
    List<ListOfFTDIs> list = new List<ListOfFTDIs>();
    FTDI.FT_STATUS ftStatus = FTDI.FT_STATUS.FT_OK;

    //pokud neprobíhá aktivní komunikace se zařízením
    if (myFtdiDevice.IsOpen == false)
    {
        //získání počtu připojených zařízení
        ftStatus = myFtdiDevice.GetNumberOfDevices(ref ftdiDeviceCount);
        if (ftStatus == FTDI.FT_STATUS.FT_OK)
        {
            if (ftdiDeviceCount != 0)
            {
                //pokud jsou nějaká připojena, vytvoř jejich seznam
                FTDI.FT_DEVICE_INFO_NODE[] ftdiDeviceList = new FTDI.FT_DEVICE_INFO_NODE[↵
                    ftdiDeviceCount];
                ftStatus = myFtdiDevice.GetDeviceList(ftdiDeviceList);
                if (ftStatus == FTDI.FT_STATUS.FT_OK)
                {
                    for (int i = 0; i < ftdiDeviceCount; i++)
                    {
                        if (ftdiDeviceList[i].ID != 0)
                        {
                            //atributy jednotlivých zařízení (popis, ID, index, adresa, seriové číslo) ↵
                                naházej do seznamu
                            ListOfFTDIs itemOfL = new ListOfFTDIs((uint)i, ftdiDeviceList[i].LocId, ↵
                                ftdiDeviceList[i].ID, ftdiDeviceList[i].Description, ftdiDeviceList[i].↵
                                    SerialNumber);
                            list.Add(itemOfL);
                        }
                    }
                }
                else {

                    GetConnectedDevicesStatusMake(new FindedDevices(list, true, false, false, string.↵
                        Empty, string.Empty));

                }
            }
            GetConnectedDevicesStatusMake(new FindedDevices(list, true, false, false, string.↵
                Empty, string.Empty));
        }
        else
        {
            //chyba ovladače zařízení
            GetConnectedDevicesStatusMake(new FindedDevices(list, false, true, false, ftStatus.↵
                ToString(), "GetDeviceList()"));
        }
    }
    else
    {
        //žádné zařízení není připojeno, seznam je prázdný
        GetConnectedDevicesStatusMake(new FindedDevices(list, true, false, false, string.↵
```

```

        Empty, string.Empty));
    }
}
else
{
    //chyba ovladače zařízení
    GetConnectedDevicesStatusMake(new FindedDevices(list, false, false, true, ftStatus←
        .ToString(), "GetNumberOfDevices()"));
}
}
}
}

```

## Díličí metody

Další skupinou metod v třídě `UsbCom` jsou metody, které aplikují význam jednotlivých elementárních metod na celý slovník dat pro kontrolní registry PAO. Jsou to metody `SendAllValuesToRAM`, `SendAllValuesToEEPROM`, `GetAllValuesFromEEPROM`, `GetAllValuesFromRAM`. Metodika jejich návrhu je taková, že pokud je v přenosu jednoho rámce zaznamenána chyba, opakuje se přenos jednoho rámce ještě třikrát, až poté je celý přenos ukončen s chybou.

## Konstruktor

Konstruktor třídy `public UsbCom(uint baudrateX, uint ReadTimeOutX, uint WriteTimeOutX, byte DataBitsX, byte StopBitsX, byte ParityX, byte FCtrlXonX, byte FCtrlXoffX, ushort FlowCtrlTypeX)` v sobě obsahuje hodnoty pro nastavení parametrů sériového přenosu od převodníku do programovacího mikrokontroléru. Tyto hodnoty jsou reprezentovány pojmenovanými konstantami.

### 3.1.6 Třída Profiler

Třída umožňuje uživateli pracovat s definovanými nastaveními schématu a vykonávat nad nimi správné operace (načíst, uložit, přejmenovat a smazat). Jednotlivá nastavení popsána v jazyce XML a ukládána do souborů v podadresáři programu.

## Události

**FilesInDirectory** Pokud uživatel provede výkonnou operaci nad určitým profilem, při které dojde ke změně výsledného souboru s profilovým nastavením, je zavolána tato událost, jejímž parametrem je seznam jednotlivých nastavení (fakticky souborů v podadresáři programu) pro následné zobrazení v okně programu.

## Hlavní metody

Hlavní skupina metod umožňuje přeložit aktuální 14B slovník vlastností PAO (aktualizovaný při libovolné změně nastavovacího schématu) na dokument XML a to oběma směry.

Jedná se o metody `ConvertMemToXml`, `ConvertXmlToMem`, z nichž uvedu pro ilustraci kód metody `ConvertMemToXml`, protože druhá metoda je analogicky stejná:

```
public string ConvertMemToXml(Dictionary<byte, byte> bb)
{
    StringWriter sw = new StringWriter();
    XmlWriter xml = XmlWriter.Create(sw);

    xml.WriteStartDocument();
    xml.WriteStartElement("BytesMemory");

    foreach (KeyValuePair<byte, byte> e in bb)
    {
        xml.WriteElementString("Byte_" + e.Key.ToString(), e.Value.ToString());
    }

    xml.WriteEndElement();
    xml.WriteEndDocument();
    xml.Close();

    return sw.ToString();
}
```

## Dílní metody

Dílní metody třídy slouží pro čtení a zápis surových XML dat – metody `WriteXmlFile`, `ReadXmlFile` dále pak pro elementární operace se souborovým systémem – metody `RenameFile`, `DeleteFile`, `TestExistFile` a `SearchFiles` (která slouží jako nosná metoda pro událost `FilesInDirectory`).

## Konstruktor

Konstruktor třídy nastaví svým parametrem jméno podadresáře, ve kterém se budou vyskytovat soubory s nastavením a kde se budou takové soubory hledat.

### 3.1.7 Třída `InfoConsole`

Třída poskytuje instanci systémové konzoly pro zápis textových zpráv opatřených časovým razítkem v různých barvách (`informace`, `úspěch`, `chyba`, `čas`, `profiler`).

## Hlavní metody

**`AdvancedCharactersInConsole`** Metoda umožňuje do konzole přidat text o určité barvě a s určitým časovým razítkem.

## Dílčí metody

Mezi dílčí metody řadím metody `OpenConsoleWindow` a `CloseConsoleWindow`, které umožňují zobrazit či skrýt okno konzole.

### 3.1.8 Třída `Form1`

Třída `Form1` již plně implementuje výše zmíněné třídy `UsbCom` a `InfoConsole`. Jejím hlavním úkolem je obsloužit vizualizaci nastavení obvodu uživateli přes ovládací schéma. Systém renderování ovládacího schématu je založen na prohlížení vždy určité části obrázku, která má přesně definované umístění ovládacích prvků. Podle toho, zda-li jsou prvky v právě zobrazovaném obdélníku, jsou buď zobrazeny nebo skryty, přičemž ovládání je realizováno pomocí 5 středových tlačítek. Krom samotného renderování schématu jsou dalšími důležitými částmi náhledy nastavených vlastností, seznam připojených zařízení s možností jeho výběru, menu pro ovládání programovacího zařízení a menu pro obsluhu profilů včetně seznamu dostupných nastavení. Pohyb v renderovaném schématu lze učinit jednak přes sérii ovládacích tlačítek, ale i prostřednictvím klávesnice šipek. Metody, které jsou v třídě obsaženy mají dvojí charakter, buď zpřístupňují určitou funkci uživateli na formuláři aplikace, nebo se starají o kreslicí operace (např. kreslení obrázku do obrázku). Z kreslicích metod chci ukázat kód metody `SchemePaint`, protože je z této oblasti metod klíčová:

```
public void SchemePaint() {  
  
    Graphics g = RenderWin.CreateGraphics();  
  
    g.CompositingMode = System.Drawing.Drawing2D.CompositingMode.SourceOver;  
    g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;  
    g.CompositingQuality = System.Drawing.Drawing2D.CompositingQuality.AssumeLinear;  
    g.SmoothingMode = SmoothingMode.HighQuality;  
    g.PixelOffsetMode = System.Drawing.Drawing2D.PixelOffsetMode.HighQuality;  
  
    Rectangle destRect = new Rectangle(0, 0, RenderWin.Width, RenderWin.Height);  
  
    Rectangle srcRect = new Rectangle(rohx, rohy, RenderWin.Width, RenderWin.Height);  
    GraphicsUnit units = GraphicsUnit.Pixel;  
  
    g.Clip = new Region(destRect);  
  
    if (Err == false)  
    {  
        RenderWin.Update();  
        g.DrawImage(Scheme, destRect, srcRect, units);  
    }  
}
```

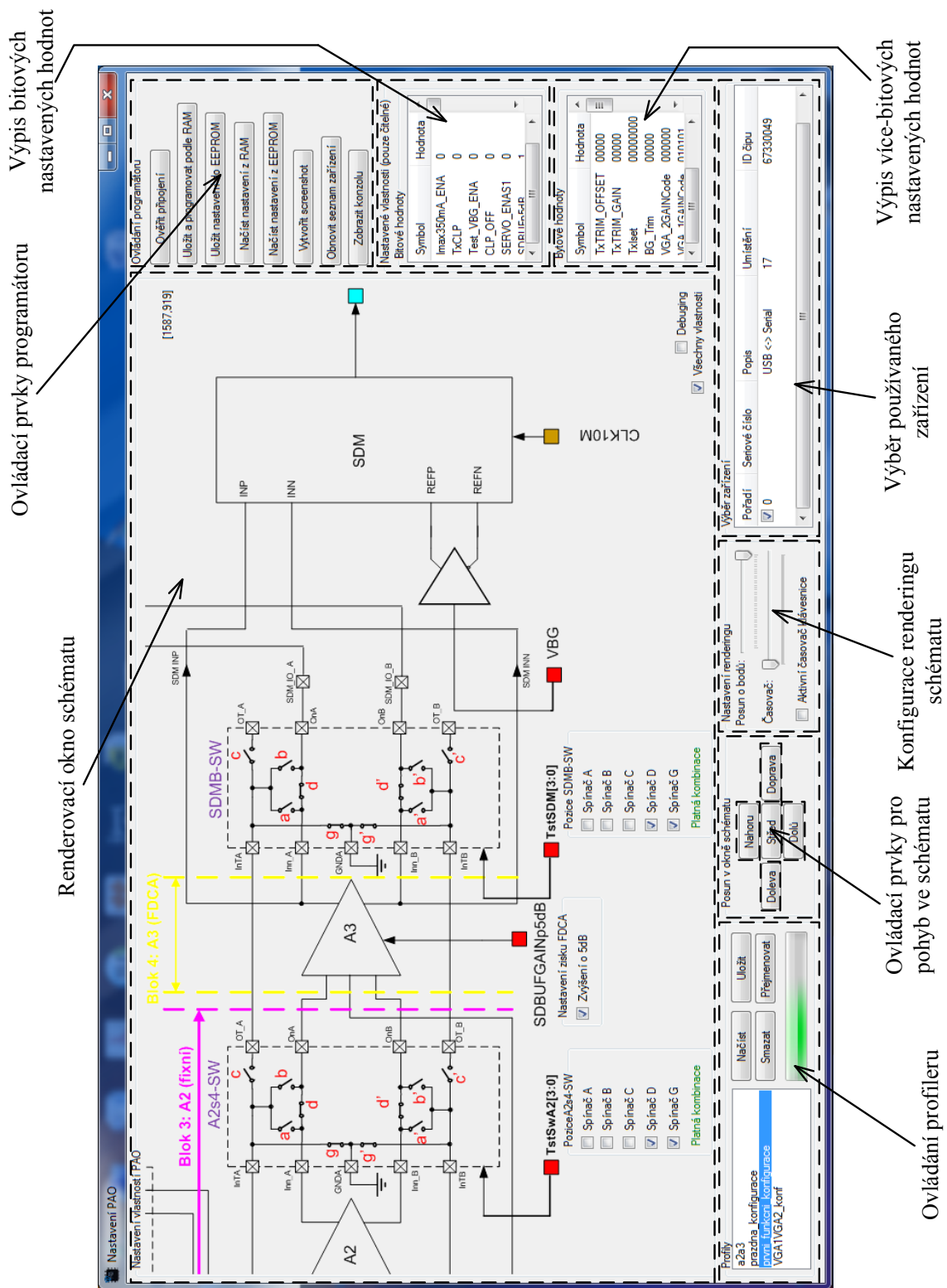
Konečnou důležitou věcí, o které se chci zmínit, je stanovení parametrů USARTu uvnitř převodníku FT232BL (později FT232RL), které musí být shodné s parametry pro USART mikrokontroléru – byly stanoveny tyto požadavky:

- Přenosová rychlost: 1000,
- Počet datových bitů: 8,
- Počet stop bitů: 2,
- Počet paritních bitů: 1 (sudá),
- Handshaking: ne.

Parametry jsou fakticky nastaveny při inicializaci objektu třídy `UsbCom` v jejím konstruktoru. Hodnota znakové rychlosti byla zvolena experimentálně na základě rychlé odezvy programovacího zařízení a minimalizace možnosti ztráty synchronizace.

### **3.1.9 Popis rozvržení okna aplikace**

Rozvržení okna ovládací aplikace je ukázáno na obr. 3.2.



Obr. 3.2: Vzhled ovládací aplikace



## 3.2 Firmware pro mikrokontrolér

Software pro mikrokontrolér Atmel ATmega88 splňuje koncepci zpracování sériového rámce, jehož podoba byla stanovena při tvorbě aplikace pro PC. Jeho důležitou vlastností je správně nastavený USART obvodu, ostatní činnosti se již týkají pouze přiřazení I/O signálů na určité porty mikrokontroléru (nejsou použity žádné další periferie kromě USARTu). Princip činnosti je takový, že mikrokontrolér čeká na přijetí 5 B po sériovém rozhraní (je použito počítadlo), dále zpracuje druhý byte s kódem příkazu (popřípadě třetí s adresou a čtvrtý s hodnotou), a pošle zpět opět 5bytový rámec, kde v prvním bytu je znamení `msgOk`, `msgErr` a zbytek jsou uživatelsky definovaná data (vrácené parametry). Po zapnutí mikrokontroléru a uplynutí ochranné doby dojde k načtení obsahu EEPROM do RAM (14B užitečných i neužitečných vlastností) a provede se zápis užitečných bytů nastavení do programovatelného analogového obvodu. Poté následuje smyčka v níž mikrokontrolér čeká na příkazy ovládací aplikace a podle nich vykonává elementární operace nad sebou.

### 3.2.1 Důležité funkce

**UsartInit** Funkce `void USART_Init(unsigned int baud)` určuje nastavení USARTu mikrokontroléru, parametry USARTu nastavované zde musí odpovídat parametrům USARTu nastavených instancí třídy FTDI:

```
void USART_Init(unsigned int baud)
{
    unsigned int rychlost = (F_CPU/16*baud)-1;

    //nastavení BAUD_RATE
    UBRROH = (unsigned char)(rychlost>>8);
    UBRROL = (unsigned char)rychlost;

    //povolení odesílání a přijímání
    UCSROB = (1<<TXEN)|(1<<RXEN); //set RXEN, TXEN, ostatní je v nule

    //nastavení parametru pro UCSRC
    //UMSEL = 0; asynchronní režim
    /* USBs - set frame format: 8data, 2stop bit */// ..... a vše ostatní je v nule
    UCSROC = (1<<USBS0)|(3<<UCSZ00)|(1<<UPM01);
}
```

**FProgrammPAO** Zavoláním této funkce naplníme obvod požadovanými vlastnostmi, provede se tak odeslání pouze užitečných vlastností. Funkce se volá vždy po startu obvodu a po uživatelské iniciativě v okně aplikace (položka *Uložit a programovat podle RAM*):

```
bool FProgrammPAO() {

    PinToReset(true); //resetovací pin na log. 1
    _delay_ms(TIME_TO_OP_US);
}
```

```

    PinToReset(false); //resetovací pin na log. 0
    _delay_ms(TIME_TO_OP_US);
    _delay_ms(TIME_TO_OP_US);

    for(uint8_t i=1; i<=14; i++){

        //zajímej se pouze o užitečná nastavení
        if((i==5)|| (i==6)|| (i==8)|| (i==9)|| (i==10)|| (i==14)){

            SetAdressRegistersToProgram(i-1); //adresace 0-13
            _delay_ms(TIME_TO_OP_US); //čas pro ustálení dat na sběrnici
            SetDataValuesToProgramm(DataByteStorage[i-1]);
            _delay_ms(TIME_TO_OP_US);
            PinToWriteSettigs(true); //zapisovací pin na log. 0
            _delay_ms(TIME_TO_OP_US);
            PinToWriteSettigs(false); //zapisovací pin na log. 1
            _delay_ms(TIME_TO_OP_US);

        }
    }

    //po programování vyčisti sběrnice od dat
    SetDataValuesToProgramm(0x00);
    SetAdressRegistersToProgram(0x00);

    return true;
}

```

**SerialCommunication** Funkce SerialCommunication je vykonávána ve smyčce po části inicializační ve struktuře hlavní funkce Main. Uvádím zde pouze fragment funkce, pro představu kompletního kódu je třeba situovat ve struktuře funkce o další dvě podmínky navíc (práce s EEPROMkou):

```

void SerialCommunication(void){
    RecieveBuffer[0] = 0;
    RecieveBuffer[1] = 0;
    RecieveBuffer[2] = 0;
    RecieveBuffer[3] = 0;
    RecieveBuffer[4] = 0;

    RecieveErr = false;

    TransmitBuffer[0] = 0;
    TransmitBuffer[1] = 0;
    TransmitBuffer[2] = 0;
    TransmitBuffer[3] = 0;
    TransmitBuffer[4] = 0;

    //přijmi za sebou 5 bytů
    for(MagicCounter=0; MagicCounter<5; MagicCounter++){
        {
            RecieveBuffer[MagicCounter] = USART_Receive();
        }
    }
}

```

```

//pokud detekuješ chybu rámce nebo parity, nastav že při příjmu vznikla chyba
if (CHECKBIT(UCSROA, FEO) || CHECKBIT(UCSROA, UPEO)){
    RecieveErr=true;
}
}
//začátek a konec rámce odpovědi
TransmitBuffer[0] = 0xFF;
TransmitBuffer[4] = 0xFF;

if(RecieveErr == true){
    //nastav chybu přijetí programového rámce na základě chyby příjmu USARTu
    TransmitBuffer[1] = (uint8_t)msgErr;
    RecieveErr=false;
}
else{
    TransmitBuffer[1] = (uint8_t)msgOK;
}
//každé elementární operaci odpovídá jedna podmínka s přiřazenou výkonnou funkcí

if(RecieveBuffer[1] == (uint8_t)ProgrammPA0FromRAM){
    FProgrammPA0();
    TransmitBuffer[2] = 0xEE;
}

if(RecieveBuffer[1] == (uint8_t)ReadByteFromRAM){
    TransmitBuffer[2] = FReadByteFromRAM(RecieveBuffer[2]);
    TransmitBuffer[3] = ~TransmitBuffer[2];
}

if(RecieveBuffer[1] == (uint8_t)WriteByteToRAM){
    TransmitBuffer[2] = FWriteByteToRAM(RecieveBuffer[2], RecieveBuffer[3]);
    TransmitBuffer[3] = RecieveBuffer[2];
}

//odešli rámec odpovědi
for (MagicCounter=0; MagicCounter<5; MagicCounter++){
    USART_Transmit(TransmitBuffer[MagicCounter]);
}
}

```

### 3.2.2 Zhodnocení návrhu softwaru

Bylo vytvořeno jednoduché a intuitivní rozhraní pro rychlou konfiguraci vlastností PAO zajišťující komfort pro uživatele. Nastavovací aplikace podporuje uložení a pozdější načtení celých konfigurací, dále byla zabudována podpora nastavování více zařízení. První verze programovacího zařízení navíc po připojení napájení vloží do PAO konfiguraci uloženou ve své EEPROMce, druhá verze programovacího zařízení musí být v okamžiku připojení USB kabelu (zároveň i napájecího) připojena ke druhé verzi zkušební desky pro úspěšné nastavení PAO. Pro svůj běh aplikace potřebuje prostředí NET Framework 2.0 a vyšší a případné připojení k internetu (pro stáhnutí ovladačů k příslušnému připojenému převodníku sériové komunikace). Při řešení některých programátorských problémů byly použity tipy a triky z [15].

## 4 KONSTRUKCE DESEK PLOŠNÝCH SPOJŮ

### 4.1 První verze zařízení

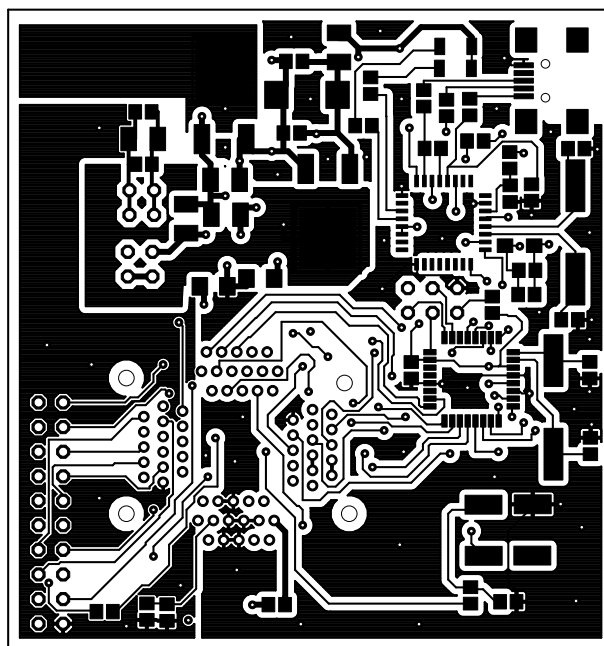
#### 4.1.1 Návrh zkušební desky plošného spoje

Nastavovaný obvod, který se řadí do nízkošumové oblasti, byla zvolena konstrukce čtyřvrstvého plošného spoje. Podstata čtyřvrstvé desky spočívá v oddělení napájecích vrstev a jejich zemí od signálových vrstev. Deska byla navrhována v systému EAGLE, který svými návrhářskými prostředky dokáže zachovat návrhářská pravidla jako jsou úhly mezi cestami spoje, dostatečná vzdálenost mezi vodiči na desce, dostatečná vzdálenost mezi komponentami na desce, prokopy apod. Při mém návrhu jsem postupoval tak, že jsem vytvořil na vnějších vrstvách signálové cesty. Na první vnitřní vrstvě napájecí spoje a na druhé vnitřní napájecí zem. Byla zvolena 5tá konstrukční třída z důvodu vedení vodiče v 2 mm mezeře mezi řadou pinů OTQ64 patice. Při celém postupu jsem se snažil dodržet určitá pravidla správného návrhu, a to:

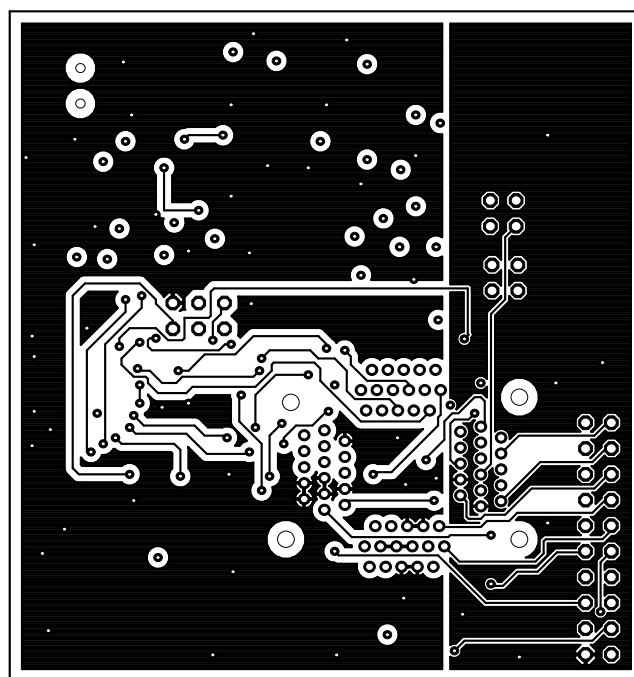
- zdroje rušení (hlavně krystaly) udržovat co nejblíže k prvku, kterému signál dodávají a pokud možno při co nejkratší signálové cestě a co nejdále od napájecích svorek,
- blokovací kondenzátory umístit do nejblíže prvku, jehož napájení filtrují,
- všeobecnou snahu o co nejkratší signálovou cestu, napájení a země schovat do vrstev vnějších,
- nevyužité plochy spoje navzájem propojit na zem a zastínit tak napájecí vrstvu.

Na desce se nachází oblast s integrovaným obvodem FT232BL a jeho okolím, dále oblast s integrovaným obvodem ATMEGA88 a jeho okolím, integrovaný oscilační obvod DXO57-12-50-T-3.3-TS zajišťující hodinový signál pro  $\Sigma-\Delta$  převodník integrovaný v PAO a patice Enplas OTQ64 spolu s konektorem.

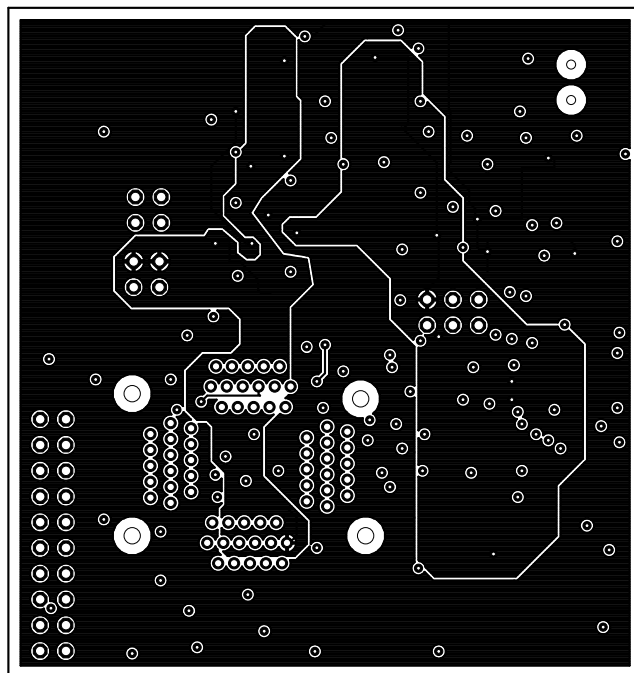
Rozkreslení jednotlivých vrstev návrhu uvádím na obr. 4.1, 4.2, 4.3, 4.4, rozpis použitých součástek je uveden v tab. 4.1 a výkres schématu v příloze C.2.



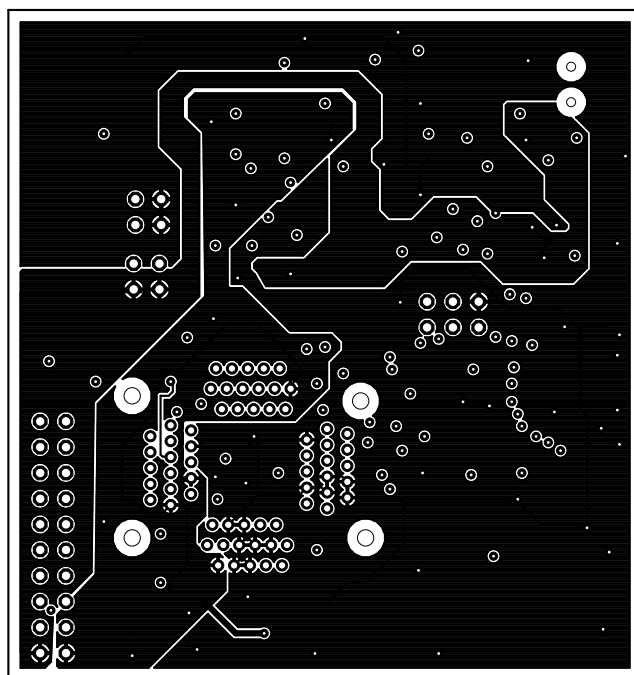
Obr. 4.1: Horní vrstva zkušební DPS (první verze)



Obr. 4.2: Spodní vrstva zkušební DPS (první verze)



Obr. 4.3: Napájecí mezi-vrstva zkušební DPS (první verze)



Obr. 4.4: Mezi-vrstva zemí zkušební DPS (první verze)

Tab. 4.1: Rozpis součástek na zkušební DPS (první verze)

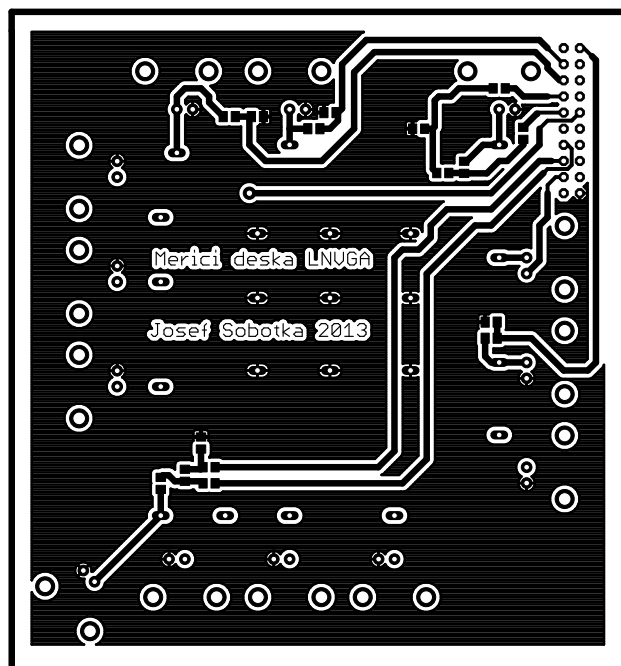
Množství	Hodnota	Pouzdro	Součástka
2	-	JP2Q	DAMP_CTRL, MAIN_SUPPLY_12V
1	-	JP3Q	JP1
2	-	JP5Q	I/O_PINS_1, I/O_PINS_1
2	0R0	R-EU_R1206	R2, R3
1	1k5	R-EU_R0805	R9
1	1 uF/25 V	C-EUC0805	C21
1	2,2 V/20 mA	LED_HSMG-C265_GREEN	RX_LED
1	2,2 V/20 mA	LED_HSMG-C265_GREEN	TX_LED
1	3,3 V/1,5 A	LM317MABDT	IC2
1	3,6864 MHz	CRYSTALHC49UP	Q1
1	4k7	R-EU_R0805	R8
1	5 V/1,5 A	LM317MADTRK	IC1
1	6,000 MHz	CRYSTALHC49UP	Q2
1	10k	R-EU_R0805	R11
1	10 uF/25 V	CPOL-EUSMCA	C8
5	10 uF/25 V	CPOL-EUSMCB	C1, C2, C5, C6, C23
1	10 uF/25 V	CPOL-EUSMCD	C14
1	15 pF/25 V	C-EUC0805	C13
1	16k	R-EU_R0805	R1
2	27R	R-EU_R0805	R6, R7
1	33 nF/25 V	C-EUC0805	C12
2	36 pF/25 V	C-EUC0805	C18, C19
2	39 pF/25 V	C-EUC0805	C3, C4
2	56R	R-EU_R0805	R12, R13
2	100R	R-EU_R0805	R5, R10
8	100 nF/25 V	C-EUC0805	C7, C9, C10, C11, C15, C17, C20, C22
2	100 pF/25 V	C-EUC0805	C16, C24
1	470R	R-EU_R0805	R4
1	FT232BL	FT232BL	IC4
1	MEGA88	MEGA8A-AI	IC3
1	On Semi (IK)	OTQ64_05_01_DEVICE	LVNVA
1	10,000 MHz	SM77H	QG1
1	USB AB mini 5-pin	MINI-USB-SHIELD-UX60-MB-5ST	X1

#### 4.1.2 Návrh měřicí desky plošného spoje

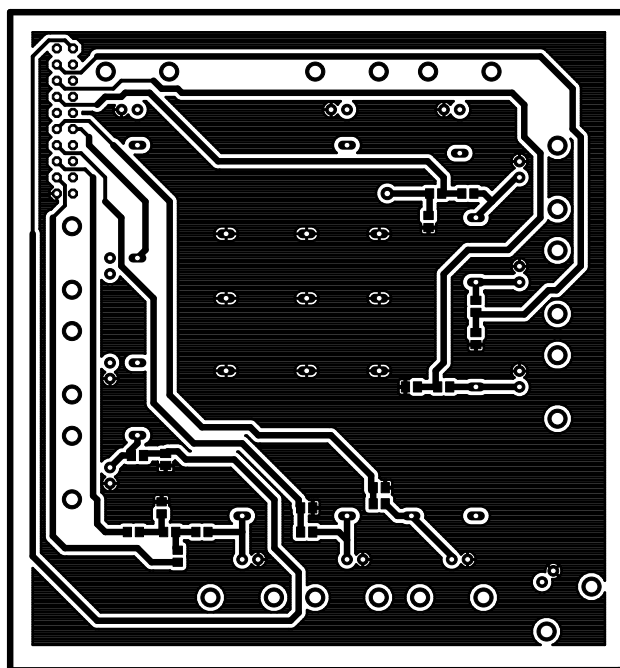
Měřicí deska poskytuje konektorové vybavení pro připojení krokosvorek a BNC konektorů koaxiálních kabelů, zároveň obsahuje pájecí plošky pro případné připojení rezistorů upravených impedancí vstupně-výstupních signálů. V dřívější verzi byly diferenční signály rozděleny vždy na dvojici samostatných signálů vedených proti zemi, v pozdějším provedení (zde uvedeném) je diferenční signál ze signálu z generátoru vytvářen na symetrizačním členu <sup>1</sup>. Dále byly pro vstupní signály dodatečně přidány oddělovací kondenzátory pro zachování stejnosměrného předpětí generovaného na vstupech zesilovačů uvnitř obvodu, na které se superponuje vkládaný signál. Jedná se o podpůrnou desku, proto byla zvolena dvouvrstvá konstrukce. Vzhledem k počtu konektorů byla snaha tahat spoje s co nejmenší parazitní indukčností a kapacitou, avšak jejich možný výskyt by neměl degradovat výsledné testování, protože v oblasti pásma práce obvodu se negativní jevy nemají šanci projevit. Tato deska slouží jako podpůrná pro první verzi měřicí DPS.

Rozkreslení jednotlivých vrstev návrhu uvádím na obr. 4.5, 4.6, rozpis použitých součástek je uveden v tab. 4.2 a výkres schématu v příloze C.2.

<sup>1</sup>Byl ponechán upravený výkres, protože i takovou podobu má upravená deska.



Obr. 4.5: Horní vrstva měřicí DPS (první verze)



Obr. 4.6: Spodní vrstva měřicí DPS (první verze)

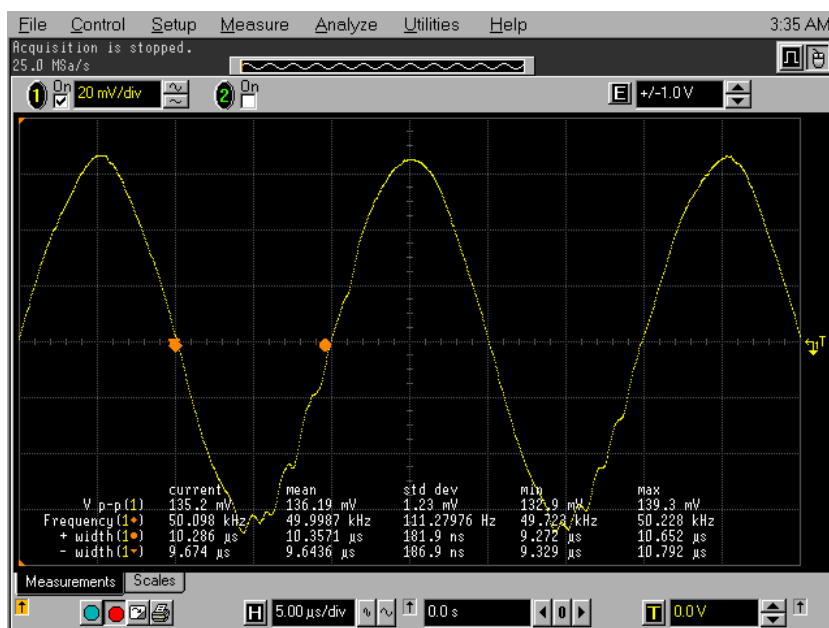


Tab. 4.2: Rozpis součástek na měřicí DPS

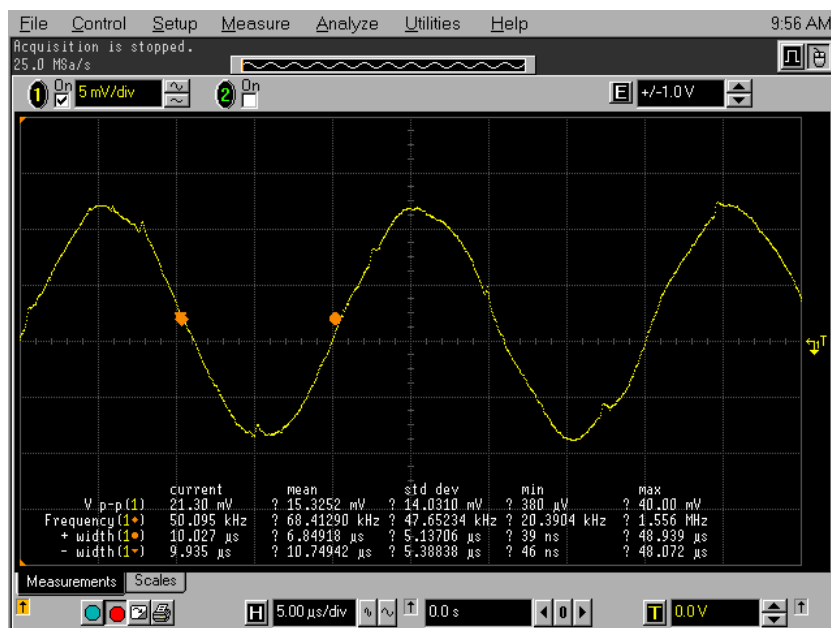
Množství	Hodnota	Pouzdro	Součástka
22	-	JMP1	JMP1, JMP2, JMP3, JMP4, JMP5, JMP6, JMP7, JMP8, JMP9, JMP10, JMP11, JMP12, JMP13, JMP14, JMP15, JMP16, JMP17, JMP18, JMP19, JMP20, JMP21, JMP22
1	-	JP10Q	JP1
16	-	M0805	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16
8	10 nF/25 V	C0805	C1, C3, C5, C8, C11, C12, C17, C18
4	10 pF/25 V	C0805	C2, C6, C15, C16
6	1 uF/25 V	C0805	C4, C7, C9, C10, C13, C14
13	BNC	BNC	A2I, BT1_IN, GM1_ON, GM1_OP, LNA_ON, LNA_OP, PDM, SDM_IO_A, SDM_IO_B, TIA_I, VGA1_I, VGA2_ON, VGA2_OP

### 4.1.3 Nevýhody první verze zařízení

První verze zkušební DPS se vyznačuje výskytem slabé kapacitní vazby (asi 0,1 pF) mezi vstupy a výstupy určitých bloků obvodu, která se ale projevuje při tak vysokém zesílení kanálu kladnou zpětnou vazbou (nejvýrazněji se tato vazba se projevuje, pokud není na vstupech BT1\_IN přiveden žádný signál, slabý projev můžeme zaznamenat v pozorování výstupního signálu, kde se občasně projevují zákmity – viz obr. 4.7,4.8), taktéž při rozmítání signálu na vstupu bloků VGA si šlo při pozorování výstupu všimnout oscilací výstupní amplitudy signálu. Proto bylo přikročeno k vývinu druhé verze, kde se snažím tento jev eliminovat.



Obr. 4.7: Zákmity na výstupním signálu celého řetězce (první verze zařízení)



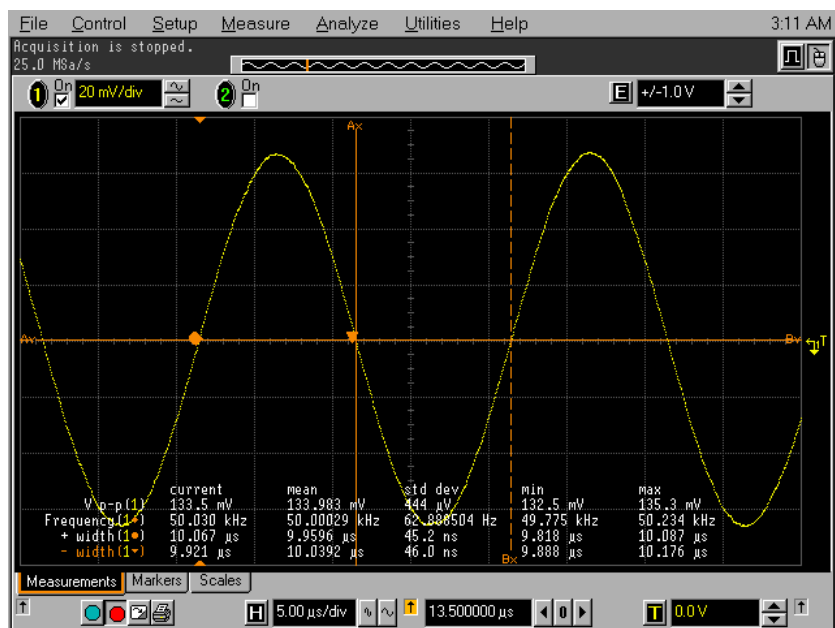
Obr. 4.8: Zákmity na výstupním signálu bloku LNA (první verze zařízení)

## 4.2 Druhá verze zařízení

Ve druhé verzi je nastavovací mikrokontrolér spolu s sériovým převodníkem na samostatné desce a patice pro programovatelný analogový obvod spolu s přípojnými konektory též na zvláštní desce (cílem je maximální eliminace možného digitálního rušení a možnost nastavit více obvodů na více deskách za sebou jedním programovacím zařízením). Rozkreslení jednotlivých vrstev návrhu programovací desky uvádím na obr. 4.10, 4.11, rozpis použitých součástek je uveden v tab. 4.3 a výkres schématu v příloze C.2, dále rozkreslení jednotlivých vrstev návrhu druhé verze zkušební desky uvádím na obr. 4.12, 4.13, 4.14, 4.15, rozpis použitých součástek je uveden v tab. 4.4 a výkres schématu v příloze C.2

## 4.3 Zhodnocení druhé verze zařízení

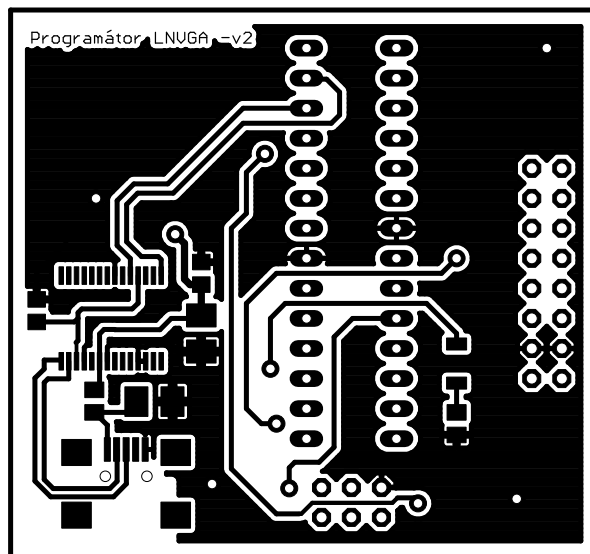
Ve druhé verzi zařízení se podařilo úspěšně odstranit zákmity ve výstupním signálu celého řetězce LNVGA (viz obr. 4.9), auto-oscilace při vstupu „ve vzduchu“ byly zpozorovány až při zesílení celého kanálu  $A_{\text{konf}} \cong 60$  dB. Zákmity ve výstupním signálu bloku LNA se nezdařilo odstranit, pouze omezit. Dále byl odstraněn jev, při kterém blok FGA a jeho dílčí části při samostatném použití vykazovaly chybu zesílení  $\Delta_A \cong 10$  dB (viz níže v kapitole 5.2.1).



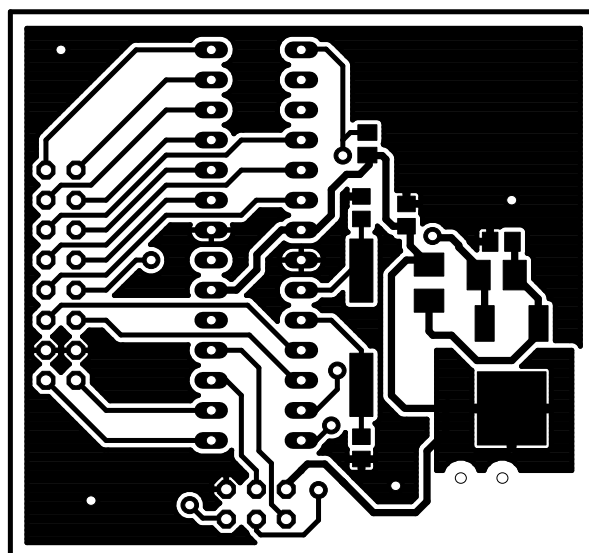
Obr. 4.9: Výstupní signál celého řetězce (druhá verze zařízení)

Tab. 4.3: Rozpis součástek na novou desku programovacího zařízení

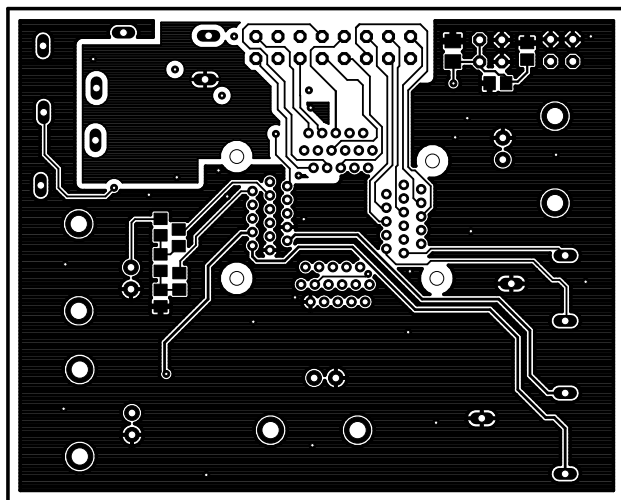
Množství	Hodnota	Pouzdro	Součástka
1	-	JP3Q	JP2
1	-	JP8Q	JP1
1	-	SOD80-R	PROCESS_DIODE
1	0R0	M0805	R2
3	100 nF/10 V	C0805	C1, C7, C8
1	10 nF/10 V	B/3528-21R	C3
2	10 uF/10 V	B/3528-21R	C5, C6
1	16k	M0805	R1
1	3,6864 MHz	HC49UP	Q1
2	39 pF/10 V	C0805	C2, C4
1	4,7 uF/10 V	B/3528-21R	C9
1	56R	M0805	R3
1	ATMEGA48/88/168-PU	DIL28-3	IC3
1	FT232RL	SSOP28	IC2
1	LM1086IS-3.3	DPACK	IC1
1	MC32598 - MINI USB TYPE AB	UX60-MB-5ST	X1
1	MI0805K400R-10	M0805	FERITOVÝ_KORÁLEK



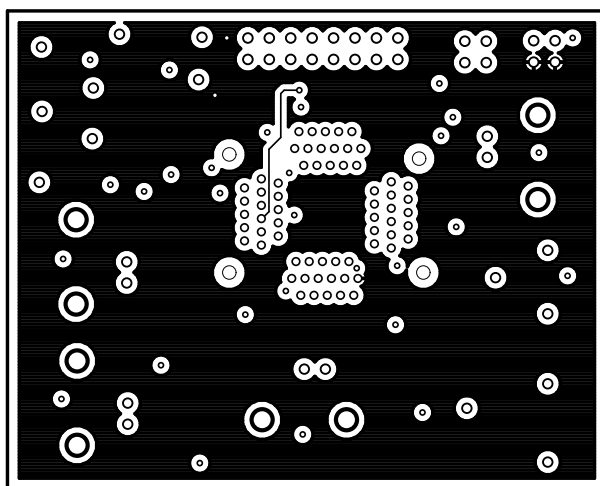
Obr. 4.10: Horní vrstva programovací DPS (druhá verze)



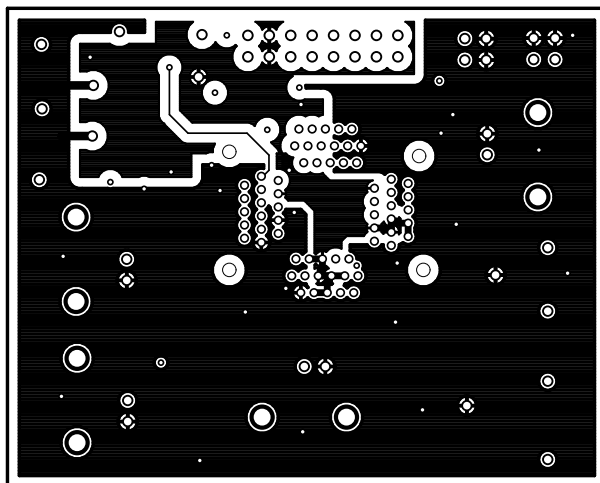
Obr. 4.11: Spodní vrstva programovací DPS (druhá verze)



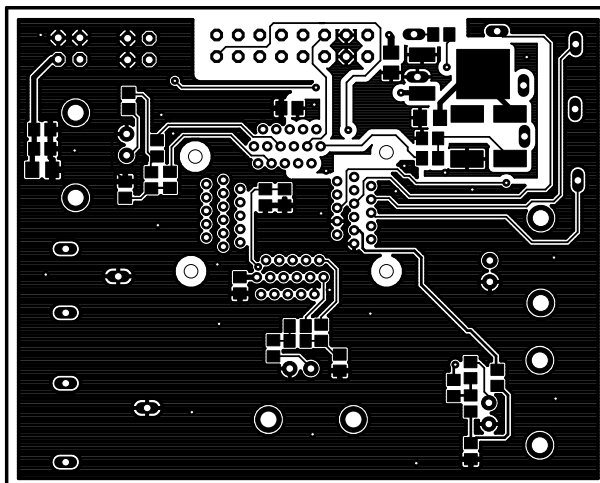
Obr. 4.12: Horní vrstva zkušební DPS (druhá verze)



Obr. 4.13: Napájecí mezi-vrstva zkušební DPS (druhá verze)



Obr. 4.14: Mezi-vrstva zemí zkušební DPS (druhá verze)



Obr. 4.15: Spodní vrstva zkušební DPS (druhá verze)

Tab. 4.4: Rozpis součástek na zkušební desku (druhá verze)

Množství	Hodnota	Pouzdro	Součástka
14	-	JMP1	GM1_ON, GM1_OP, LNA_ON, LNA_OP, PMD_OUT, SDM_IO_A, SDM_IO_B, VGA2_ON, VGA2_OP, XGND, XGND1, XGND2, XGND3, XGND4
2	-	JP2Q	DAMP_CTRL, SUPPLY_VOLTAGE
1	-	JP8Q	JP1
1	0R0	M0805	R2
2	100R	M0805	R1, R3
4	100 nF/16 V	C0805	C1, C5, C7, C8
2	100 pF/16 V	C0805	C4, C6
8	10 nF/16 V	C0805	C13, C15, C16, C18, C19, C21, C22, C24
4	10 pF/16 V	C0805	C14, C17, C20, C23
4	10 uF/16 V	R/2012-12R	C9, C10, C11, C12
1	15 pF/16 V	C0805	C3
9	1 uF/16 V	C0805	C2, C25, C26, C27, C28, C29, C30, C31, C32
4	BNC	BNC	A2I, BT1_IN, TIA1_I, VGA1_I
1	LM317MABDT	DPACK	IC1
1	On Semiconductor (IK)	OTQ_64_05_01	LNVGA
1	DXO57-12-50-T-3.3-TS	SM77H	QG1

## 5 MĚŘENÍ NA PROGRAMOVATELNÉM ANALOGOVÉM OBVODU

### 5.1 Úvodní měření obvodu

Před samotným měřením programovatelného analogového obvodu prostřednictvím navrženého vybavení na obvodovém analyzátoru bylo třeba provést oživení obvodu a úvodní měření, tj. stanovit vhodné úrovně vstupního signálu, při kterých interní zesilovače obvodu nedosáhnou brzy saturace a funkce jejich výstupního průběhu nebude omezena (zůstane čistě harmonická). Celé oživení (s následným úvodním měřením) jsem rozvrhl celkem na 6 částí. Jedná se nejprve o snahu provést vstupní signál všemi interními zesilovači obvodu a poté pouze dílčími bloky, jak nám to umožňuje vnitřní signálový multiplex. U první verze zařízení jsem ve všech případech postupoval tak, že diferenciální signálový pár jsem nahradil pouze signálem jedním (druhý signál představoval zem), čímž se celý postup výrazně zjednodušil, avšak po doporučení konstruktéra obvodu bylo buzení předěláno na čistě diferenční (použito i v druhé verzi zařízení). Sledování výstupních průběhů bylo v první verzi vedeno nediferenčně (sledováním záporného či kladného kanálu), později byla použita ke sledování diferenční sonda. Jako referenční frekvenci vstupního signálu jsem zvolil  $f = 50 \text{ kHz}$ , protože se nachází přibližně uprostřed garantované pracovní šířky pásma většiny komponent. Pro představu popisovaných stavů byly jednotlivé obrazovky osciloskopu ofoceny a vloženy do příloh, v dokumentu se jich nachází pouze několik. K nim se poté dá dostat přes dynamický odkaz v dokumentu (avšak pracuje pouze ve verzi uložené na CD).

Při úvodním pozorování a měření všech stanovených částí byla pro určitá kritická vstupní napětí  $U_{IN}$  zpozorována tvarová zkreslení funkce sinus, která byla zapříčiněna buď přebuzením zesilovače (špičaté tvary), omezením (tvar do obdélníka), nastartováním kladné zpětné vazby (výskyt zákmitů) nebo malou rychlostí přeběhu (tvarové deformace). V následujících kapitolách je pro konfigurace o více než jedné hodnotě zesílení či více než jednoho nastavitelného zesilovače uvedena tabulka, ve které jsou zachyceny kritické hodnoty vstupních napětí, pro které tyto nežádoucí jevy nastávaly. V následujících napěťových údajích týkajících se úvodního měření jsou uvažovány hodnoty napětí  $U_{IN}$  a  $U_{OUT}$  jako hodnoty špička-špička (proto nejsou údaje nazývány amplitudami, i když se jedná o ustálený harmonický stav). Níže v popisu jsou vstupy a výstupy bloků či jejich podbloků popsány diferenciálními páry, které vznikly sloučením dílčích pinů obvodu (např. diferenční vstupy BT1\_IN1 a BT1\_IN2 dají dohromady diferenciální pár BT1\_IN), tento popis je v rámci oživování a úvodního měření dodržen.

#### 5.1.1 Blok LNA

Při měření byl jako vstup použit BT1\_IN a výstup LNA\_O. Experimentálně-vizuálně zjištěná mez hodnot rozkmitu vstupního signálu se pohybuje cca do  $U_{IN} = 3 \text{ mV}_{PP}$ , jinak nastane napřed omezení záporných půlvln a později i kladných půlvln. Zapnutá servo-smyčka bloku LNA se projeví při dosažení saturace zesilovače a to tak, že mírně zderivuje ořezaný

sinus, pokud ji necháme vypnutou, zůstane ořezán do obdélníku (při normálním chodu zapnutá servo-smyčka pouze mírně zvýší hodnotu výstupního signálu zpětnou vazbou). Pro dosažení optimální úrovně rozkmitu vstupního signálu byl použit odporový dělič 1:10. Bylo tak dosaženo hodnot vstupních napětí  $U_{IN} = 2 \text{ mV}_{PP}$  a  $U_{IN'} = 10 \text{ mV}_{PP}$ . Pro zapnutou i vypnutou servo-smyčku vychází na výstupním průběhu přibližně  $U_{OUT} = 213 \text{ mV}_{PP}$  a  $U_{OUT'} = 963 \text{ mV}_{PP}$ . Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.5 (veškerá pozorování byla provedena na první verzi zařízení). Při známém zesílení bloku LNA můžeme udělat zkoušku:

$$A_{LNA} = 20 \log \frac{U_{OUT}}{U_{IN}} = 20 \log \frac{0,213}{0,002} = 40,54 \text{ dB} \quad (5.1)$$

$$A_{LNA} = 20 \log \frac{U_{OUT'}}{U_{IN'}} = 20 \log \frac{0,963}{0,010} = 39,67 \text{ dB} \quad (5.2)$$

### 5.1.2 Blok VGA

#### Podblok VGA2 bloku VGA

Podblok VGA1 bloku VGA byl přemostěn přes signálové multiplexery *LNAs1-SW* a *VGA1s2-SW*. Jako vstup byl použit VGA1\_I a výstup VGA2\_O. Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.1 (veškerá pozorování byla provedena na první verzi zařízení). Opět experimentálně-vizuálně byly zjištěny pro 4 hodnoty zesílení z celkových 64 krajní hodnoty vstupních napětí, od kterých dochází k omezení či zkreslení výstupního napětového signálu zesilovače:

Tab. 5.1: Pozorování podbloku VGA2 bloku VGA

Zkoumané hodnoty $A_{VGA2}$	Zjištěné kritické $U_{IN}$
-10 dB	> 1,503 V <sub>PP</sub>
0 dB	> 1,684 V <sub>PP</sub>
10 dB	> 1,503 V <sub>PP</sub>
20 dB	> 470 mV <sub>PP</sub>

Pokud je použit zesilovač VGA2 samostatně, je vhodné nechat nastavenou vlastnost *SWgmTIA1\_S*, poněvadž pak nemá změna nastavení servo-smyčky bloku LNA žádný vliv na funkci zesilovače (v případě nulování vlastnosti *SWgmTIA1\_S* dochází při jejím zapnutí či vypnutí ke zkreslení). Při pozorování zcela nezkraslených průběhů byly získány pro hodnoty vstupních napětí  $U_{IN} = 1,602 \text{ V}_{PP}$  a  $U_{IN'} = 1,873 \text{ V}_{PP}$  hodnoty výstupních napětí  $U_{OUT} = 4,100 \text{ V}_{PP}$  a  $U_{OUT'} = 640 \text{ mV}_{PP}$  pro  $A_{VGA2} = 10 \text{ dB}$  ( $VGA2GainCode = 42$ ) a  $A_{VGA2} = -10 \text{ dB}$  ( $VGA2GainCode = 0$ ). Při známém zesílení podbloku VGA2 bloku VGA můžeme udělat zkoušku:

$$A_{VGA2_{10 \text{ dB}}} = 20 \log \frac{U_{OUT}}{U_{IN}} = 20 \log \frac{4,100}{1,602} = 8,16 \text{ dB} \quad (5.3)$$

$$A_{VGA2_{-10 \text{ dB}}} = 20 \log \frac{U_{OUT'}}{U_{IN'}} = 20 \log \frac{0,640}{1,873} = -9,32 \text{ dB} \quad (5.4)$$



## Podbloky VGA1 a VGA2 společně

Pro otestování bloku VGA byl opět využit vstup VGA1\_I a výstup VGA2\_O, signál byl tentokrát poslán přes signálové multiplexery *LNAs1-SW* a *VGA1s2-SW* v jejich výchozí konfiguraci přes oba zesilovače. Při intuitivním nastavení obvodu mě zarazilo, že je třeba (skoro i nesmyslně) nulovat vlastnost *SWLNAVGA\_S*, aby se dvojice zesilovačů chytla (vysvětlením může být přivedení reference z výstupu bloku LNA, na které se následně nasuperponuje vstupní napětí ze vstupu VGA1\_I, platí pro obě verze zařízení). Pro měření byla v tomto případě naprosto kritická konfigurace:

- *SWgmTIA1\_S* = ON,
- *SWLNAVGA\_S* = OFF,
- *SERVO\_ENAS1* = ON,
- *SERVO\_ENAS2* = ON.

Změnou libovolného nepřemostění byla naprosto potlačena funkce obou bloků, nulováním vlastnosti *SERVO\_ENAS1* došlo k omezení výstupního průběhu daleko dříve a vlastnost *SERVO\_ENAS2* neměla na celou práci vliv. Z množiny diskrétních zesílení jsem použil 4 hodnoty  $\{-10, 0, 10, 20\}$  a z nich vytvořil 16 kombinací, které jsem nastavil a pozoroval. Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.6 (veškerá pozorování byla provedena na první verzi zařízení). Nejčastějšími jevy byly odlišné tvary záporných a kladných půlvln, které následovalo omezení (většinou prvně shora, potom zdola), méně časté bylo odlišné trvání kladného a záporného půlkmitu. V tab. 5.2 chci popsat, kdy k jevům omezení či nesouměrnosti obou půlvln docházelo.

Tab. 5.2: Pozorování bloku VGA

$A_{VGA1}$	$A_{VGA2}$	Kritické $U_{IN}$
-10 dB	-10 dB	> 330 mV <sub>PP</sub>
-10 dB	0 dB	> 330 mV <sub>PP</sub>
-10 dB	10 dB	> 330 mV <sub>PP</sub>
-10 dB	20 dB	> 330 mV <sub>PP</sub>
0 dB	-10 dB	> 330 mV <sub>PP</sub>
0 dB	0 dB	> 330 mV <sub>PP</sub>
0 dB	10 dB	> 330 mV <sub>PP</sub>
0 dB	20 dB	> 250 mV <sub>PP</sub>
10 dB	-10 dB	> 100 mV <sub>PP</sub>
10 dB	0 dB	> 100 mV <sub>PP</sub>
10 dB	10 dB	> 100 mV <sub>PP</sub>
10 dB	20 dB	> 50 mV <sub>PP</sub>
20 dB	-10 dB	> 50 mV <sub>PP</sub>
20 dB	0 dB	> 50 mV <sub>PP</sub>
20 dB	10 dB	> 50 mV <sub>PP</sub>
20 dB	20 dB	< 30 mV <sub>PP</sub>

Při pozorování nezkreslených průběhů byla získána pro hodnotu vstupního napětí  $U_{IN} = 142 \text{ mV}_{PP}$  a kombinaci zesílení  $A_{VGA1} = 0 \text{ dB}$ ,  $A_{VGA2} = 20 \text{ dB} \rightarrow \text{VGA1GainCode} = 21$ ,  $\text{VGA2GainCode} = 63$  hodnota výstupního napětí  $U_{OUT} = 2,242 \text{ mV}_{PP}$  a pro hodnotu vstupního napětí  $U_{IN'} = 38 \text{ mV}_{PP}$  a kombinaci zesílení  $A_{VGA1} = 0 \text{ dB}$ ,  $A_{VGA2} = 20 \text{ dB} \rightarrow \text{VGA1GainCode} = 21$ ,  $\text{VGA2GainCode} = 63$  hodnota výstupního napětí  $U_{OUT'} = 642 \text{ mV}_{PP}$ . Při známém zesílení podbloků bloku VGA můžeme udělat zkoušku:

$$A_{VGA20 \text{ dB}} = 20 \log \frac{U_{OUT}}{U_{IN}} = 20 \log \frac{2,242}{0,142} = 23,96 \text{ dB} \quad (5.5)$$

$$A_{VGA_{20\text{ dB}}} = 20 \log \frac{U_{OUT'}}{U_{IN'}} = 20 \log \frac{0,642}{0,038} = 24,55 \text{ dB} \quad (5.6)$$

### 5.1.3 Blok FGA

#### Bloky A2 a A3 jako celek

Pro otestování podbloků A2 a A3 za sebou jsem využil jako vstup A2I a výstup SDM\_IO. Pro korektní funkci bloku FGA je nutné vypnout vlastnost nepřemostění pro první verzi zařízení natrvalo, pro druhou je nutné nepřemostění *SWLNAVGA2\_S* vypnout a zapnout (empiricky zjištěno – jedná o přivedení stejnosměrné reference z výstupu podbloku VGA2 bloku VGA, na který se následně nasuperponuje náš vstupní signál ze vstupu A2I). Oba zesilovače za sebou se ukazují jako velmi lineární (ke tvarovému zkreslení u nich vůbec nedošlo). Experimentálně-vizuálně zjištěná mez hodnot rozkmitu vstupního signálu pro *SDBUFFGAINp5dB = OFF* se pohybuje cca do  $U_{IN} = 230 \text{ mV}_{PP}$ , poté nastane symetricky omezení obou půlvln. Pokud zapneme vlastnost *SDBUFFGAINp5dB = ON*, získáme na zesilovači A3 možnost zvýšit jeho zesílení z původních  $A_{A3} \cong 6 \text{ dB}$  na  $A_{A3_{BOOST}} \cong 11 \text{ dB}$ , přitom je linearita zesilovače zachována. Experimentálně-vizuálně zjištěná mez hodnot rozkmitu vstupního signálu pro *SDBUFFGAINp5dB = ON* se pohybuje cca do  $U_{IN} = 126 \text{ mV}_{PP}$ , po této hranici nastane nesymetrické omezení (napřed horní půlvlny a později dolní). Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.2 (veškerá pozorování byla provedena na druhé verzi zařízení).

Při pozorování nezkreslených průběhů byla získána pro vlastnost *SDBUFFGAINp5dB = OFF* a pro hodnotu vstupního napětí  $U_{IN} = 18 \text{ mV}_{PP}$  hodnota výstupního napětí  $U_{OUT} = 336 \text{ mV}_{PP}$  a dále pro vlastnost *SDBUFFGAINp5dB = ON* při hodnotě vstupního napětí  $U_{IN'} = 18 \text{ mV}_{PP}$  hodnota výstupního napětí  $U_{OUT'} = 556 \text{ mV}_{PP}$ . Při známém zesílení podbloků bloku FGA můžeme udělat zkoušku:

$$A_{FGA_{26\text{ dB}}} = 20 \log \frac{U_{OUT}}{U_{IN}} = 20 \log \frac{0,336}{0,018} = 25,42 \text{ dB} \quad (5.7)$$

$$A_{FGA_{31\text{ dB}}} = 20 \log \frac{U_{OUT'}}{U_{IN'}} = 20 \log \frac{0,556}{0,018} = 29,79 \text{ dB} \quad (5.8)$$

#### Podblok A2

Pro otestování samostatného podbloku A2 bylo použito stejného vstupu a výstupu jako při testování podbloků A2 a A3 jako celku, pouze byl podblok A3 prostřednictvím signálových multiplexerů *A2s4-SW* a *SDMB-SW* přemostěn. Pro korektní funkci podbloku A2 je opět nutné pro první verzi zařízení vypnout (pro druhou je nutné vypnout a zapnout) vlastnost nepřemostění *SWLNAVGA2\_S* (empiricky zjištěno – jedná o přivedení stejnosměrné reference z výstupu podbloku VGA2 bloku VGA, na který se následně nasuperponuje náš vstupní signál ze vstupu A2I). Blok se vyznačuje svojí linearitou pro velký rozkmit vstupních signálů (ke tvarovému zkreslení u něho vůbec nedošlo), experimentálně-vizuálně zjištěná mez hodnoty rozkmitu vstupního signálu se pohybuje cca do  $U_{IN} = 370 \text{ mV}_{PP}$

poté nastává omezení shora. Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.3 (veškerá pozorování byla provedena na druhé verzi zařízení). Při pozorování nezkreslených průběhů byla získána pro hodnotu vstupního napětí  $U_{\text{IN}} = 99 \text{ mV}_{\text{PP}}$  hodnota výstupního napětí  $U_{\text{OUT}} = 900 \text{ mV}_{\text{PP}}$ . Při známém zesílení podbloku A2 bloku FGA můžeme udělat zkoušku:

$$A_{\text{A2}} = 20 \log \frac{U_{\text{OUT}}}{U_{\text{IN}}} = 20 \log \frac{0,900}{0,099} = 19,17 \text{ dB} \quad (5.9)$$

### Podblok A3

Pro otestování podbloku A3 bylo použito opět stejného vstupu a výstupu jako pro testování bloku FGA, pouze pomocí signálových multiplexerů *VGA2s3-SW* a *A2s4-SW* byl vstup a výstup nasměrován přímo do podbloku A3. Pro korektní funkci podbloku A3 je opět nutné pro druhou verzi zařízení vypnout a následně zapnout vlastnost nepřemostění (pro první verzi stačí natrvalo vypnout) *SWLNAVGA2\_S* (empiricky zjištěno – jedná o přivedení stejnosměrné reference z výstupu podbloku VGA2 bloku VGA, na který se následně nasuperponuje náš vstupní signál ze vstupu A2I). Podblok A3 je stejně jako podblok A2 lineární v celém rozsahu. Pro vlastnost *SDBUFFGAINp5dB* = OFF od experimentálně-vizuálně zjištěné hodnoty vstupního signálu  $U_{\text{IN}} = 900 \text{ mV}_{\text{PP}}$  se dostane kladná půlvlna do tvaru trojúhelníku, ale omezení nedosáhne, pokud je vlastnost *SDBUFFGAINp5dB* = ON nastane omezení shora při experimentálně-vizuálně zjištěné velikosti hodnoty vstupního signálu  $U_{\text{IN}} = 600 \text{ mV}_{\text{PP}}$ . Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.4 (veškerá pozorování byla provedena na druhé verzi zařízení). Při pozorování nezkreslených průběhů byly získány pro hodnoty vstupních napětí při vlastnosti *SDBUFFGAINp5dB* = OFF  $U_{\text{IN}} = 199 \text{ mV}_{\text{PP}}$ , a při *SDBUFFGAINp5dB* = ON a  $U_{\text{IN}'} = 199 \text{ mV}_{\text{PP}}$  hodnoty výstupních napětí  $U_{\text{OUT}} = 390 \text{ mV}_{\text{PP}}$ ,  $U_{\text{OUT}'} = 674 \text{ mV}_{\text{PP}}$ . Při známém zesílení podbloku A3 bloku FGA můžeme udělat zkoušku:

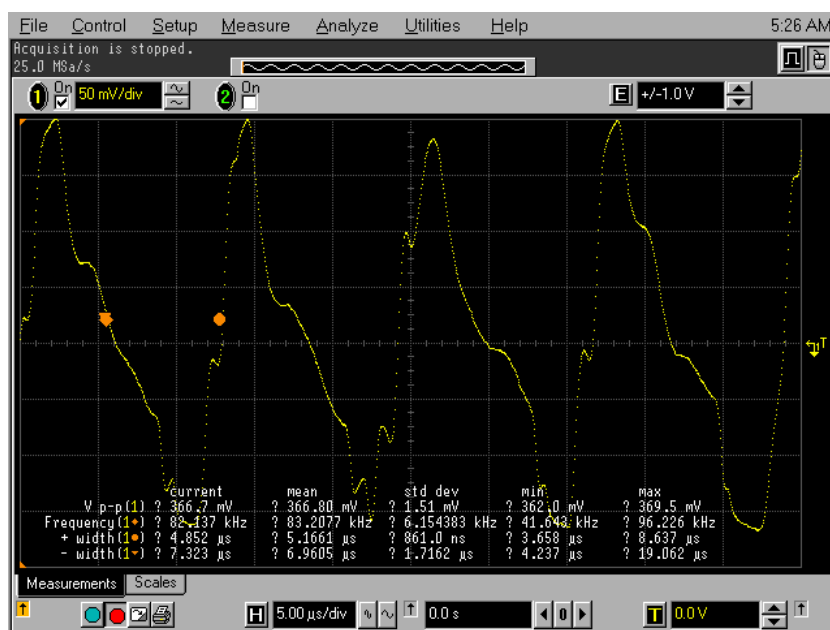
$$A_{\text{A3}} = 20 \log \frac{U_{\text{OUT}}}{U_{\text{IN}}} = 20 \log \frac{0,390}{0,199} = 4,54 \text{ dB} \quad (5.10)$$

$$A_{\text{A3}+5 \text{ dB}} = 20 \log \frac{U_{\text{OUT}'}}{U_{\text{IN}'}} = 20 \log \frac{0,674}{0,199} = 10,59 \text{ dB} \quad (5.11)$$

#### 5.1.4 Celý LNVGA

Při měření byl jako vstup řetězce použit BT1\_IN a výstup řetězce SDM\_IO. Dílčí bloky obvodu jsou do tohoto řetězce zapojeny při výchozí konfiguraci, bylo třeba tedy pouze měnit nastavení servo-smýček (*SERVO\_ENAS1*, *SERVO\_ENAS2*), nastavení zesílení kaskády podbloků bloku VGA ( $A_{\text{VGA1}}$ ,  $A_{\text{VGA2}}$ ) a vlastnost *SDBUFFGAINp5dB* podbloku A3 bloku FGA. Z diskrétní množiny zesílení bloku VGA jsem vybral 4 hodnoty zesílení  $\{-10, 0, 10, 20\}$ , pro které jsem testoval všechny kombinace nastavení servo-smýček a vlastnosti *SDBUFFGAINp5dB* podbloku A3 bloku FGA. Tato nastavení byla testována vstupním napětím o hodnotách  $U_{\text{IN}} \in \{4,4 \text{ mV}_{\text{PP}}; 13,5 \text{ mV}_{\text{PP}}; 24 \text{ mV}_{\text{PP}}\}$  (k hodnotám jsem dospěl experimentálně, protože

při nich se zobrazovaná funkce výstupního napětí nejvíce lišila). Pozorování celého řetězce se rozdělilo na dvě části, při nichž hrály dominantní role zesílení podbloky bloku VGA. Odkazy na fotografie pozorovaných průběhů jsou dostupné v tab. A.7,A.8,A.9 (veškerá pozorování byla provedena na první verzi zařízení). Pokud bylo v konfiguraci přítomno maximum libovolného podbloku VGA (5 konfigurací VGA z celkových 9), bylo možno pro libovolnou hodnotu vstupního napětí pozorovat saturovaný obdélník, pokud byl vstupní signál odpojen a vstup ponechán „ve vzduchu“, objevil se pro první verzi zařízení na výstupu průběh podobný obdélníku s nestabilní frekvencí  $f_{\text{AUTOOSC}} \approx 100 \text{ kHz}$  (nastavením servo-smyček se frekvence auto-oscilace výrazně neměnila), viz obr. 5.1, pro druhou verzi zařízení byl výstup pro vstup „ve vzduchu“ čistý. Popis pozorovaných průběhů pro zbylé čtyři konfigurace (protože se jedná o celý řetězec, tak i s popisem tvaru signálu) je uveden v tabulce 5.3. Pokud se v nastavovaných 4 konfiguracích objevily obdélníky s zakmitanými místy nekonečných derivací, vypnutím servo-smyčky  $\text{SERVO\_ENAS1} = \text{OFF}$  byly tyto překmity obvykle vykompenzovány a navíc obdélníky získaly větší sklon nástupných a sestupných hran (servo-smyčka podbloku VGA1 bloku VGA neměla na tvar průběhu žádný zvláštní vliv).



Obr. 5.1: Auto-oscilace celkového řetězce pro vstup „ve vzduchu“ u první verze zařízení

Při pozorování nezkreslených průběhů byly získány hodnoty výstupních napětí  $U_{\text{OUT}}$ ,  $U_{\text{OUT}'}$ ,  $U_{\text{OUT}''}$ ,  $U_{\text{OUT}'''}$  pro hodnotu vstupního napětí  $U_{\text{IN}} = U_{\text{IN}'} = U_{\text{IN}''} = U_{\text{IN}'''} = 4,4 \text{ mV}_{\text{PP}}$  a příslušnou konfiguraci zesílení řetězce  $A_{\text{KONF}}$ :

- $U_{\text{IN}} = 4,4 \text{ mV}_{\text{PP}}$  pro  $A_{\text{VGA1}} = -10 \text{ dB}$ ,  $A_{\text{VGA2}} = -10 \text{ dB}$ ,  $\text{SERVO\_ENAS1} = \text{ON}$ ,  $\text{SERVO\_ENAS2} = \text{ON}$ ,  $\text{SDBUFFGAINp5dB} = \text{ON} \rightarrow U_{\text{OUT}} = 1,352 \text{ V}_{\text{PP}} \rightarrow A_{\text{KONF1}} = 50 \text{ dB}$ ,

- $U_{IN'} = 4,4 \text{ mV}_{PP}$  pro  $A_{VGA1} = -10 \text{ dB}$ ,  $A_{VGA2} = -10 \text{ dB}$ ,  $SERVO\_ENAS1 = \text{ON}$ ,  $SERVO\_ENAS2 = \text{ON}$ ,  $SDBUFFGAINp5dB = \text{OFF} \rightarrow U_{OUT'} = 789 \text{ mV}_{PP} \rightarrow A_{KONF2} = 45 \text{ dB}$ ,
- $U_{IN''} = 4,4 \text{ mV}_{PP}$  pro  $A_{VGA1} = -10 \text{ dB}$ ,  $A_{VGA2} = 0 \text{ dB}$ ,  $SERVO\_ENAS1 = \text{ON}$ ,  $SERVO\_ENAS2 = \text{OFF}$ ,  $SDBUFFGAINp5dB = \text{OFF} \rightarrow U_{OUT''} = 2,499 \text{ V}_{PP} \rightarrow A_{KONF3} = 55 \text{ dB}$ ,
- $U_{IN'''} = 4,4 \text{ mV}_{PP}$  pro  $A_{VGA1} = -10 \text{ dB}$ ,  $A_{VGA2} = 0 \text{ dB}$ ,  $SERVO\_ENAS1 = \text{ON}$ ,  $SERVO\_ENAS2 = \text{OFF}$ ,  $SDBUFFGAINp5dB = \text{ON} \rightarrow U_{OUT'''} = 3,498 \text{ V}_{PP} \rightarrow A_{KONF4} = 60 \text{ dB}$ .

$$A_{LNVGA1} = 20 \log \frac{U_{OUT}}{U_{IN}} = 20 \log \frac{1,352}{0,0044} = 49,75 \text{ dB} \quad (5.12)$$

$$A_{LNVGA2} = 20 \log \frac{U_{OUT'}}{U_{IN'}} = 20 \log \frac{0,789}{0,0044} = 45,07 \text{ dB} \quad (5.13)$$

$$A_{LNVGA3} = 20 \log \frac{U_{OUT''}}{U_{IN''}} = 20 \log \frac{2,499}{0,0044} = 55,08 \text{ dB} \quad (5.14)$$

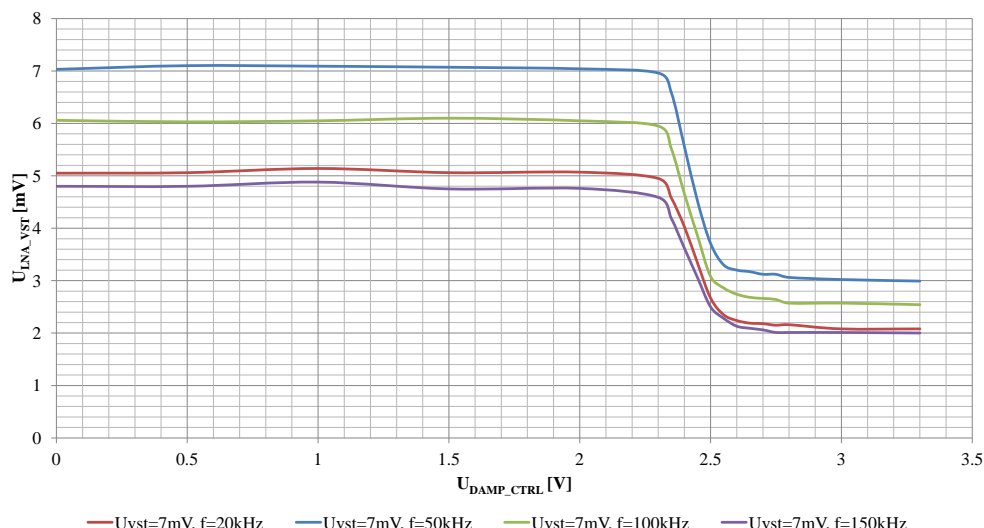
$$A_{LNVGA4} = 20 \log \frac{U_{OUT'''}}{U_{IN'''}} = 20 \log \frac{3,498}{0,0044} = 58,00 \text{ dB} \quad (5.15)$$

Tab. 5.3: Pozorování řetězce LNVGA

$A_{VGA1}$	$A_{VGA2}$	Kritické $U_{IN}$	Důsledky překročení
		$> 4,4 \text{ mV}$	občasné zakmitlá sinusovka
$-10 \text{ dB}$	$-10 \text{ dB}$	$> 13,5 \text{ mV}_{PP}$	oblý trojúhelník na dolní špičce zakmitlý
		$> 24 \text{ mV}_{PP}$	lehce zderivovaný obdélník
		$> 4,4 \text{ mV}_{PP}$	na dolní půlvlně zakmitaná sinusovka
$-10 \text{ dB}$	$0 \text{ dB}$	$> 13,5 \text{ mV}_{PP}$	čistý obdélník téměř bez zákmitů
		$> 24 \text{ mV}_{PP}$	obdélník (při $SDBUFFGAINp5dB = \text{OFF}$ dole celkově zakmitaný, jinak ne)
		$> 4,4 \text{ mV}_{PP}$	na dolní půlvlně zakmitaná sinusovka
$0 \text{ dB}$	$-10 \text{ dB}$	$> 13,5 \text{ mV}_{PP}$	obdélník téměř bez zákmitů (pro $SDBUFFGAINp5dB = \text{OFF}$ klesají hrany pomaleji)
		$> 24 \text{ mV}_{PP}$	obdélník (pro $SDBUFFGAINp5dB = \text{ON}$ dole zakmitaný)
		$> 4,4 \text{ mV}_{PP}$	obdélník (pro $SDBUFFGAINp5dB = \text{OFF}$ dole dost zakmitaný, jinak čistý)
$0 \text{ dB}$	$0 \text{ dB}$	$> 13,5 \text{ mV}_{PP}$	obdélník (pro $SDBUFFGAINp5dB = \text{OFF}$ zakmitané svislé hrany)
		$> 24 \text{ mV}_{PP}$	obdélník (pro $SDBUFFGAINp5dB = \text{OFF}$ zakmitané svislé hrany)

## 5.2 Předřadný blok $R_{IN}$

Regulací vstupní impedance pomocí vnějšího napětí  $U_{DAMP\_CTRL}$  jsme schopni velmi velký odpor zesilovače BOTA bloku LNA (obvykle v řádu jednotek  $M\Omega$ ) snížit pomocí paralelně přidaného  $R_{IN}$  (ten lze plynule regulovat -  $R_{IN} \in (800 \Omega, 50 \text{ k}\Omega)$ ). Vzniká tak spolu s výstupní impedancí zdroje vstupního signálu dělič napětí, který je schopen vstupní napětí podělit až 2,35krát (viz graf na obr. 5.2).



Obr. 5.2: Grafická závislost  $U_{VST\_LNA}(U_{DAMP\_CTRL})$  pro  $U_{BT1\_IN} = \text{konst.}$

### 5.2.1 Zhodnocení úvodních měření

Výpočty zde uvedené je třeba chápat jako orientační, jako důkaz, že obvod je skutečně nastavován, napětí na vstupu zesilovače není úplně přesně změřeno. Možné prahové hodnoty vstupních napětí, při kterých dochází k postupnému zkreslování výstupního signálu, byly změřeny při nediferenčním ohledávání obvodu (při diferenčním zkoumání se prahové hodnoty změnily a navíc určitá zkreslení úplně zmizela), nepřesnost prahových hodnot je nejmarkantnější pro blok FGA, který se po diferenčním ohledání zdá být lineárnější více, než při nediferenčním ohledávání. Dále byly stanoveny kritické konfigurace vlastností pro správnou funkci bloků VGA a FGA (dočasná i trvalá nastavení vlastností  $SWLNA\_VGA2\_S$  a  $SWLNAVGA\_S$ ). Bylo dokázáno, že zapojení je elektronicky funkční, tedy mikrokontrolér dodává potřebná data a obvod podle nich nastavuje svoje vlastnosti, byly stanoveny přibližné hodnoty vstupních napětí, pro které není výstup jednotlivých částí, byť někdy částečně zkreslen, omezen. Odchyłky od hodnot zesílení požadovaných v jednotlivých výpočtech zde uvedených jsou většinou konstantní v rámci jednotlivých bloků i při změnách jejich nastavení, taky jsou závislé na hodnotách frekvence vstupního signálu (vady navržených desek plošných spojů krom výskytu slabé kapacitní vazby z výstupu na vstup zesilovačů mohou vyloučit)<sup>1</sup>.

## 5.3 Analýza obvodovým analyzátozem

Z přibližných hodnot vstupních napětí zjištěných při ožiování, pro které nenastalo omezení nebo zkreslení na výstupu, bylo realizováno měření na obvodovém analyzátozem. Rozmítání vstupního signálu bylo nejprve provedeno pro frekvenční interval  $\langle 100 \text{ Hz}; 10 \text{ MHz} \rangle$  pro představu, jak se obvod chová v celé spektrální šíři. Dále bylo prováděno pro oblast kolem

<sup>1</sup>Pro první verzi zařízení vznikala při pozorování bloku FGA a jeho dílčích částí chyba  $\Delta_A \cong 10 \text{ dB}$ , která se v rámci řetězce LNVGA neprojevovala (pro druhou verzi zařízení byla chyba odstraněna).

garantované šířky pásma ve frekvenčním intervalu  $\langle 10 \text{ kHz}; 500 \text{ kHz} \rangle$ , kde je již provedeno více podrobných měření. Měření byla vždy prováděna ze vstupu na výstup příslušných bloků (stejnou signálovou cestou jako při ožívování), pro více možností nastavení zesílení určitých bloků bylo prováděno vícenásobné měření, přičemž jednomu znázorněnému průběhu frekvenční závislosti zesílení  $A(f)$  včetně charakteristiky fázové  $\varphi(f)$  odpovídá jedna konfigurace v legendě popsaná. Vstupní signál je uváděn v úrovněvé veličině  $L_{VST}$  vztažené k referenční hodnotě  $U_{IN_{0dBm}} = 224 \text{ mV}$ , protože vstupní impedance bloků či jejich podbloků není rovna výstupní impedanci obvodového analyzátoru  $Z_{OUT} = 50 \Omega$ . Hodnoty úrovní vstupního signálu  $L_{VST}$  jsou voleny tak, aby se výstup měřeného celku nedostal do omezení. Popsaná zesílení  $A_{KONF}$  pro měřené celky obsahující více než jednu možnost konfigurace výsledného zesílení celku jsou uvedena v součtovém tvaru, tento tvar odpovídá měřeným celkům takto:

- $A_{KONF} = A_{VGA1} + A_{VGA2}$  pro blok VGA,
- $A_{KONF} = A_{LNA} + A_{VGA1} + A_{VGA2} + A_{A2} + A_{A3}$  pro celek LNVGA,
- $A_{KONF} = A_{A2} + A_{A3}$  pro blok FGA.

Protože tyto grafické závislosti jsou výstupem celé práce, byly pro svoji důležitost umístěny do přílohy B.

Na základě zachycených průběhů přenosové a fázové frekvenční charakteristiky (kladného diferenčního páru) byly určeny hodnoty maximálních zesílení  $A_{MAX}$ , k nim příslušejících frekvencí  $f_{MAX}$ , na kterých se  $A_{MAX}$  nachází, dále zjištěné  $f_{MIN}$  a  $f_{MAX}$  šířky pásma při poklesu  $A_{MAX}$  o 3 dB. Na odpovídajících fázových charakteristikách (vždy klesajících) bylo určeno kolikrát blok otáčí fázi a pokud otáčí, tak na jakých frekvencích (viz tab. 5.4,5.5).

### 5.3.1 Zhodnocení měření na obvodovém analyzátoru

Měření bylo stejně jako ožívování prováděno na první verzi zařízení v přivádění vstupního signálu na vstupní piny určitého celku a následného odebrání kladného kanálu výstupního signálu (měřicí sonda k obvodovému analyzátoru byla pouze nediferenční). Konfigurace nastavení schématu byly shodné pro úvodní měření (včetně nesmyslně vypnutých nepřemostění pro měření bloků VGA a FGA (včetně jeho podbloků)). Pro první verzi zkušební desky vznikl při měření podbloků bloku VGA (VGA2 samostatně či s VGA1 společně) problém takový, že jakmile se vstupní signál „přiblížil“ frekvenci, na které se přes parazitní kapacitní vazby z výstupu na vstup jednoho, či skupiny zesilovačů mohlo nastartovat samo-kmitání přes slabou kladnou zpětnou vazbu, pak s postupným rozmítáním od této frekvence bylo možno sledovat rozkmitaný poměr výstupního ku vstupnímu signálu na obrazovce obvodového analyzátoru – tento problém se dal eliminovat tím způsobem, že jsme uložily požadovanou konfiguraci do paměti EEPROM nastavovacího mikrokontroléru, celé zařízení se zapnutím naprogramovali při nejbližším dalším průchodu přes začátek frekvenčního intervalu a následně tento první změřený interval po startu zařízení zaznamenali (tento empirický postup vyhovoval pro všechny konfigurace zesilovače VGA2, a většinu konfigurací skupiny VGA1 a VGA2 kromě nastavení  $A_{KONF} = (19+19) \text{ dB}$ ). Tento jev

jako ostatní jiné nežádoucí byly odstraněny v druhé verzi zařízení. K hodnotám  $A_U$  odečtených v jednotlivých grafech je třeba pomyslně přičíst hodnotu 6 dB pro zjednodušenou představu diferenčního měření. Jak lze z naměřených přenosových a fázových frekvenčních charakteristik zjistit, bloky LNA a VGA mají své předepisované zesílení pouze v určitém úzkém frekvenčním pásmu a blok FGA zesiluje předepisovaným zesílením v širším frekvenčním pásmu.

Tab. 5.4: Pohled na jednotlivé bloky (podbloky) a celky na širší části spektra

Blok (podblok)	$A_{KONF}$ [dB]	$A_{MAX}$ [dB]	$f_{A_{MAX}}$ [kHz]	$f_{MIN}$ [kHz]	$f_{MAX}$ [kHz]	Otočení fáze [kolikrát]	Frekvence zlomu [kHz]
LNA	39	32	60	15	250	1	3 MHz
VGA	(10+0)	4	100	11	879	2	40 kHz; 3,5 MHz
A2	20	10	100	10	590	1	490 kHz
A3	6	0	0,5	-	551	1	650 kHz
FGA	(20+6)	17	100	12	437	1	750 kHz
LNPGA	(39-10-10+20+6)	40	65	35	150	3	61 kHz; 700 kHz; 2,1 MHz

Tab. 5.5: Pohled na jednotlivé bloky (podbloky) a celky na pracovní části spektra

Blok (podblok)	$A_{KONF}$ [dB]	$A_{MAX}$ [dB]	$f_{A_{MAX}}$ [kHz]	$f_{MIN}$ [kHz]	$f_{MAX}$ [kHz]	Otočení fáze [kolikrát]	Frekvence zlomu [kHz]
VGA2	-10	-13	350	20	-	1	190 kHz
VGA2	0	-9	105	11	-	1	190 kHz
VGA2	10	3	100	19	-	1	190 kHz
VGA2	20	12	50	15	210	1	190 kHz
VGA	(10-10)	-12	50	13	-	-	-
VGA	(-10+10)	-12	50	13	100	-	-
VGA	(0+10)	-1,7	66	13	108	-	-
VGA	(10+0)	-1,8	66	13	350	-	-
VGA	(0+0)	-11	257	16	350	-	-
VGA	(-10+0)	-3	70	15	110	1	300 kHz
VGA	(0-10)	-4	66	11	270	-	-
VGA	(-10-10)	-11	97	13	370	-	-
VGA	(10+10)	17	18	-	350	-	-
VGA	(-10+20)	23	71	18	420	-	-
VGA	(20-10)	12	47	16	210	-	-
VGA	(20+10)	30	50	13	140	1	450 kHz
VGA	(10+20)	30	50	13	170	1	450 kHz
VGA	(0+20)	20	50	15	170	-	-
VGA	(20+0)	20	45	15	160	-	-
VGA	(20+20)	35	38	17	136	1	350 kHz
A3	6	0	10	-	-	-	-
A3	11	5	10	-	-	-	-
FGA	(20+6)	17	100	12	437	-	-
FGA	(20+11)	24	80	17	344	-	-
LNPGA	(39-10-10+20+6)	39	70	30	170	1	62 kHz
LNPGA	(39-0-10+20+11)	53	70	30	170	1	62 kHz
LNPGA	(39-0-10+20+6)	49	70	30	170	1	62 kHz
LNPGA	(39-10-0+20+11)	53	70	30	170	1	62 kHz
LNPGA	(39-10-0+20+6)	49	70	30	170	1	62 kHz
LNPGA	(39-10-10+20+11)	43	70	30	170	1	62 kHz



## 6 ZÁVĚR

Vycházel jsem z prostudování technické zprávy k novému PAO z firmy On Semiconductor (struktura, principiální funkce základních stavebních jednotek obvodu – zesilovačů BOTa a BTIA). Na základě dohody s budoucími testery obvodu jsem dospěl k návrhu rozhraní ovládací aplikace pro snadné nastavení vlastností obvodu. Dále bylo zvoleno řešení pro komunikaci s obvodem, vytvořen firmware pro programovací mikrokontrolér. Původní aplikace byla vylepšena o možnost ukládání nastavených konfigurací a jejich zpětné načtení, byl zredukován počet bytů přenášených do programovacího mikrokontroléru. Byla navržena první verze zkušební desky, která obsahovala programovací část (digitální) spolu s částí analogovou-signálovou. K první verzi zkušební desky byla dále navržena podpůrná měřicí deska s konektorovým vybavením. První verze se vyznačuje výraznými zákmity ve výstupním signálu celého řetězce LNVGA, méně výraznými zákmity ve výstupním signálu prvního bloku LNA a chybou zesílení bloku FGA a jeho dílčích částí  $\Delta_A \cong 10$  dB. Na základě odstranění nedostatků první verze zkušební desky byla navržena druhá verze splňující koncepci oddělené programovací části a přítomnosti konektorového vybavení. Současně byla navržena, vyrobena a osazena deska programovacího zařízení. U druhé verze zkušební desky se nevyskytují zákmity ve výstupním signálu celého řetězce LNVGA, avšak přetrvávají ve výstupním signálu bloku LNA, blok FGA spolu s dílčími částmi pracuje správně. Na základě navrženého ovládaní byly empiricky stanoveny kritické vlastnosti programovatelného analogového obvodu a jejich hodnoty, při kterých se jednotlivé bloky či vnitřní celky uvnitř PAO (na základě pozorování z výstupu na vstup) začnou chovat nestandardně. Též byly stanoveny přibližné kritické hodnoty vstupních napětí, pro které je výstupní signál bloků či jejich podbloků neharmonického charakteru. Pro hodnoty vstupních napětí mimo tyto kritické hodnoty bylo provedeno úvodní měření, při kterém byla otestována zesilovací funkce jednotlivých bloků včetně jejich podbloků. Posledním krokem bylo naměření přenosových a fázových frekvenčních charakteristik a následné zpracování těchto charakteristik (zejména určení maximálního zesílení  $A_{MAX}$  při frekvenci  $f_{A_{MAX}}$ , určení okrajových frekvencí  $f_{MIN}$ ,  $f_{MAX}$  šířky pásma pro pokles zesílení o 3 dB). Dílčí chyby zesílení jednotlivých bloků či jejich podbloků a celků mohou být dány též tím, že při úvodních měřeních a následném měření na obvodovém analyzátoru bylo operováno se třemi kusy prototypů obvodu (chyba zesílení proto může být kus od kusu klidně i  $\delta_A = 20\%$ ). Obvody s programovatelnými funkcemi se jeví jako perspektiva do budoucích let, kdy nebude nutné např. měnit parametry zesilovače připojením vnějších prvků, ale pouze datovou hodnotou uvnitř nějaké řídicí části.

## LITERATURA

- [1] AD844 60MHz 2000V/us Monolithic Op Amp. Datasheet, Analog Devices [online]. 2001, poslední aktualizace 19.02.2009 [cit. 20.11.2012]. Dostupné z URL: <<http://www.analog.com/en/other-products/militaryaerospace/ad844/products/product.html>>.
- [2] ATmega88 Microcontroller. Datasheet, Atmel Corp. [online]. 2011, poslední aktualizace 04.11.2011 [cit. 20.11.2012]. Dostupné z URL: <<http://www.atmel.com/Images/doc2545.pdf>>.
- [3] AXMAN, V. Využití transimpedančních zesilovačů v aktivních filtrech. In: *Proceedings of the 3-rd Conference of Czech Student AES Section on Audio Technologies and Processing* [online]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2002. p. 53-59. ISBN: 80-214-2128-2. Dostupné z URL: <<http://radio.feld.cvut.cz/AES/atp2002/proc/paper06.pdf>>.
- [4] BIOLEK, D. Řešení obvodů s transimpedančními operačními zesilovači pomocí grafů signálových toků. [online]. 1994, s. 6 [cit. 20.11.2012]. Dostupné z URL: <[http://user.unob.cz/biolek/veda/articles/EDS94\\_1.pdf](http://user.unob.cz/biolek/veda/articles/EDS94_1.pdf)>.
- [5] FRANCO, S. *Design with operational amplifiers and analog integrated circuits*. 3rd ed. Boston: McGraw-Hill, 2002, xiv, 658 s. ISBN 0-07-232084-2.
- [6] FT232BL USB UART IC. Datasheet, FTDI Ltd. [online]. 2002, poslední aktualizace 07.01.2011 [cit. 20.11.2012]. Dostupné z URL: <<http://www.analog.com/en/other-products/militaryaerospace/ad844/products/product.html>>.
- [7] GRAY, P. R et al *Analysis and design of analog integrated circuits*. 4th ed. New York: John Wiley & Sons, 2001, xviii, 875 s. ISBN 0471321680.
- [8] JEŘÁBEK, J. *Kmitočtové filtry s proudovými aktivními prvky: Frequency filters with active current elements*. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2007. 1 elektronický optický disk [CD-ROM / DVD]. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Kamil Vrba.
- [9] KOLEKTIV UTEE. *Měření v elektrotechnice (BMVA): Analogově-číslicové a číslicově-analogové převodníky* [online]. Ústav teoretické a experimentální elektrotechniky, 2002 [cit. 20.11.2012]. Studijní materiály. Dostupné z URL: <[http://www.utee.feec.vutbr.cz/CZ/Dokumenty\\_ke\\_stazeni/BMVA/prevodniky\\_AC\\_CA.pdf](http://www.utee.feec.vutbr.cz/CZ/Dokumenty_ke_stazeni/BMVA/prevodniky_AC_CA.pdf)>.
- [10] KOUDAR, I. *Low Noise Variable Gain Analog Frontend*. Technická zpráva. ON Semiconductor, 2012.
- [11] LATTENBERG, I. *Objektově orientované programování*. Brno: VUT v Brně, 2012. s. 141. ISBN: 978-80-214-4447-8.

- [12] OLMR, V. HW server představuje - Sériová linka RS-232. In: *HW server s.r.o.* [online]. 2005, 12.12.2005 [cit. 20.11.2012]. Dostupné z URL: <<http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>>.
- [13] OLMR, V. USB - Universal Serial Bus - Popis rozhraní. In: *HW server s.r.o.* [online]. 2002, 06.05.2002 [cit. 20.11.2012]. Dostupné z URL: <<http://www.hw.cz/docs/usb/usb.html>>.
- [14] POOLE, I. RS232 serial interface tutorial. In: *Radio-Electronics.com* [online]. 2012 [cit. 20.11.2012]. Dostupné z URL: <[http://www.radio-electronics.com/info/telecommunications\\_networks/rs232/rs232-serial-interface-basics-tutorial.php](http://www.radio-electronics.com/info/telecommunications_networks/rs232/rs232-serial-interface-basics-tutorial.php)>.
- [15] STACK EXCHANGE INC. *Stack Overflow* [online]. 2013. vyd. 2013 [cit. 2013-05-23]. Dostupné z URL: <<http://stackoverflow.com/>>.
- [16] *Universal Serial Bus Specification: Revision 2.0*. USB Implementers Forum, Inc., 2000. Dostupné z URL: <<http://www.usb.org/developers/docs/>>.
- [17] VRBA, K.; HERENCŠÁR, N.; KOTON, J. *Analogová technika*. Brno: VUT v Brně, 2011. s. 104. ISBN: 978-80-214-4589- 5.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BOTA	Operační transkonduktanční zesilovač s diferenčním výstupem – Balanced operational transconductance amplifier
BTIA	Transimpedanční zesilovač (operační) s diferenčním výstupem – Balanced operational transimpedance amplifier
LNPGA	Nízkošumový zesilovač s nastavitelným zesílením (řetězec) – Low noise variable gain amplifier (chain)
FDCA	Plně diferenciální proudový zesilovač – Fully differential current amplifier
LNA	Nízkošumový zesilovač – Low noise amplifier
OTA	Operační transkonduktanční zesilovač – Operational transconductance amplifier
PAO	Programovatelný analogový obvod – Programmable analog circuit
$R_{IN}$	Vstupní impedance řetězce – Input impedance chain
TIA	Transimpedanční zesilovač – Transimpedance amplifier
VGA	Zesilovač s proměnným zesílením – Variable gain amplifier
FGA	Zesilovač s fixním zesílením – Fix gain amplifier
$u_{IN}$	Nediferenční vstupní napětí – Single-ended input voltage
$u_{IN-}$	Diferenční vstupní napětí (negativní složka) – Differential input voltage (negative channel)
$u_{IN+}$	Diferenční vstupní napětí (pozitivní složka) – Differential input voltage (positive channel)
$u_{OUT}$	Nediferenční výstupní napětí – Single-ended output voltage
$u_{OUT+}$	Diferenční výstupní napětí (pozitivní složka) – Differential output voltage (positive channel)
$u_{OUT-}$	Diferenční výstupní napětí (negativní složka) – Differential output voltage (negative channel)
$g_m$	Převodní transkonduktance – Conversion transconductance
$r_m$	Převodní transimpedance – Conversion transimpedance
$i_{OUT}$	Výstupní nediferenční proud – Single-ended output current
$f_m$	Mezní frekvence – Cutoff frequency

$i_{\text{IN}}$	Nediferenční vstupní proud – Single-ended input current
$i_{\text{IN}-}$	Diferenční vstupní proud (negativní složka) – Differential input current (negative channel)
$i_{\text{IN}+}$	Diferenční vstupní proud (pozitivní složka) – Differential input current (positive channel)
$i_{\text{BIAS}}$	Před-nastavující proud – Biasing current
$R_{\text{TIA}}$	Zpětnovazební odpor převodníku proud-napětí – The feedback resistor in current-voltage converter
$U_{\text{DAMP\_CTRL}}$	Napětí nastavující vstupní impedanci – Voltage adjusting input impedance
$L_{\text{VST}}$	Vstupní úroveň signálu ( $L_{\text{VST}} = 0 \text{ dBm}$ pro $U_{\text{IN}} = 224 \text{ mV}$ na zátěži $Z = 50 \Omega$ ) – Input signal level ( $L_{\text{VST}} = 0 \text{ dBm}$ for $U_{\text{IN}} = 224 \text{ mV}$ on load $Z = 50 \Omega$ )
USART	Univerzální synchronní/asynchronní přijímač/vysílač – Universal synchronous/asynchronous receiver/transmitter
USB	Univerzální sériová sběrnice – Universal serial bus
PDM	Pulzně hustotní modulace – Pulse density modulation

## SEZNAM PŘÍLOH

<b>A</b>	<b>Odkazy do fotogalerie pozorování</b>	<b>78</b>
<b>B</b>	<b>Výsledky nediferenčního měření na obvodovém analyzátoru</b>	<b>83</b>
<b>C</b>	<b>Výkresy schémat desek plošných spojů</b>	<b>95</b>
	C.1 Popis obsahu přiloženého CD-ROM . . . . .	95
	C.2 Výkresy schémat . . . . .	95

## A ODKAZY DO FOTOGALERIE POZOROVÁNÍ

Zde jsou uvedeny odkazy na fotografie z pozorování v podadresáři **mereni** adresáře s elektronickou verzí práce. Uváděná napětí na obrazovce osciloskopu jsou po atenuaci konstantou 10 (správnou hodnotu napětí dostaneme po vynásobení deseti).

Tab. A.1: Odkazy do fotogalerie – pozorování podbloku VGA2 bloku VGA

Pozorování podbloku VGA2 bloku VGA – fotogalerie	
$SW_{gmTIA1\_S} = ON, A_{VGA2} = -10 \text{ dB}, U_{IN} = 1,873 \text{ V}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SW_{gmTIA1\_S} = ON, A_{VGA2} = 0 \text{ dB}, U_{IN} = 1,879 \text{ V}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SW_{gmTIA1\_S} = ON, A_{VGA2} = 10 \text{ dB}, U_{IN} = 1,602 \text{ V}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SW_{gmTIA1\_S} = ON, A_{VGA2} = 20 \text{ dB}, U_{IN} = 578 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SW_{gmTIA1\_S} = OFF, A_{VGA2} = 10 \text{ dB}, SERVO\_ENAS1 = OFF, U_{IN} = 455 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SW_{gmTIA1\_S} = OFF, A_{VGA2} = 10 \text{ dB}, SERVO\_ENAS1 = ON, U_{IN} = 455 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>

Tab. A.2: Odkazy do fotogalerie – pozorování FGA

Pozorování bloku FGA – fotogalerie	
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 18 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 98 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 199 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 18 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 98 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 199 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>

Tab. A.3: Odkazy do fotogalerie – pozorování podbloku A2 bloku FGA

Pozorování podbloku A2 bloku FGA – fotogalerie	
$U_{IN} = 18 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$U_{IN} = 99 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$U_{IN} = 199 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$U_{IN} = 298 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$U_{IN} = 396 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$U_{IN} = 454 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>

Tab. A.4: Odkazy do fotogalerie – pozorování podbloku A3 bloku FGA

Pozorování podbloku A3 bloku FGA – fotogalerie	
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 19 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 98 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 199 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 495 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 998 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = OFF, U_{IN} = 1,497 \text{ V}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 199 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 303 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 390 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SDBUFFGA_{INp5dB} = ON, U_{IN} = 495 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>

Tab. A.5: Odkazy do fotogalerie – pozorování LNA

Pozorování bloku LNA – fotogalerie	
$SERVO\_ENAS1 = OFF, U_{IN} = 2 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = ON, U_{IN} = 2 \text{ mV}_{PP}$ (normální chod)	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = ON, U_{IN} = 10 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = OFF, U_{IN} = 10 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = ON, U_{IN} = 20 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = OFF, U_{IN} = 20 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = ON, U_{IN} = 201 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>
$SERVO\_ENAS1 = OFF, U_{IN} = 201 \text{ mV}_{PP}$	<a href="#">Pouze ve verzi na CD</a>

Tab. A.6: Odkazy do fotogalerie – pozorování VGA

[illegible]



Tab. A.7: Odkazy do fotogalerie – pozorování LNVGA (část 1)

[illegible]

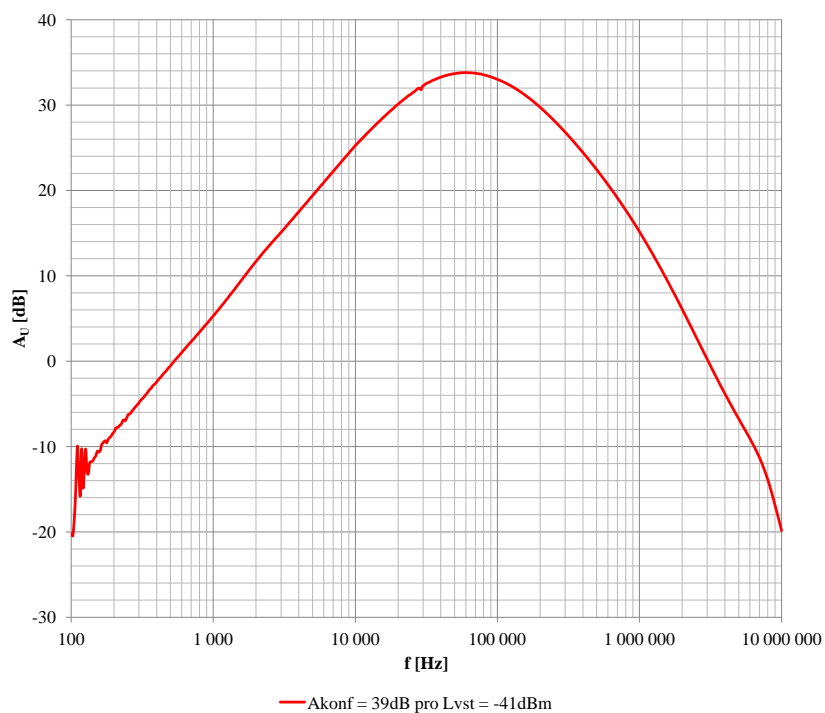
Tab. A.8: Odkazy do fotogalerie – pozorování LNVGA (část 2)

[illegible]

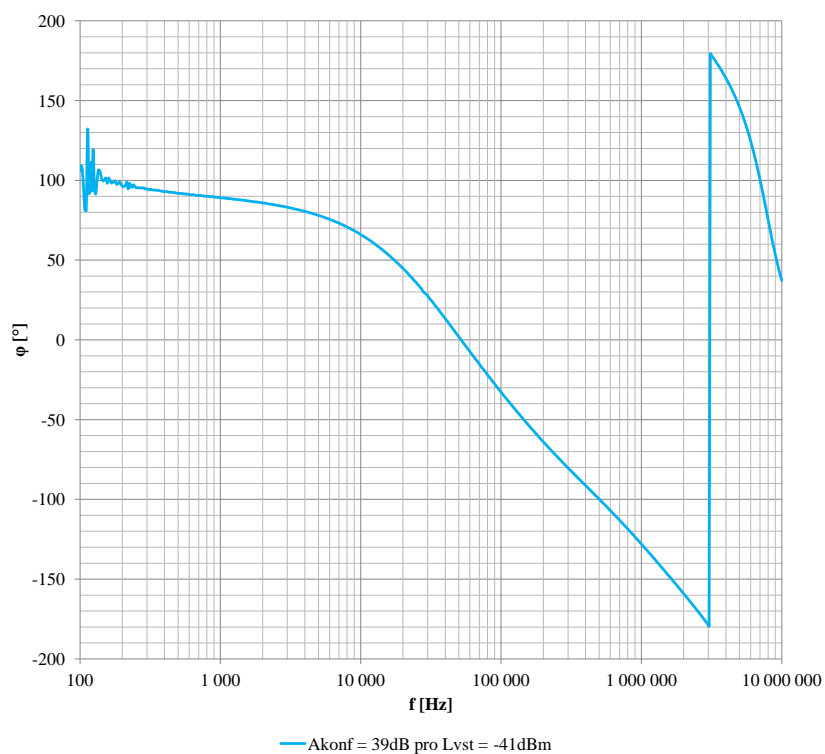
Tab. A.9: Odkazy do fotogalerie – pozorování LNVGA (část 3)

[illegible]

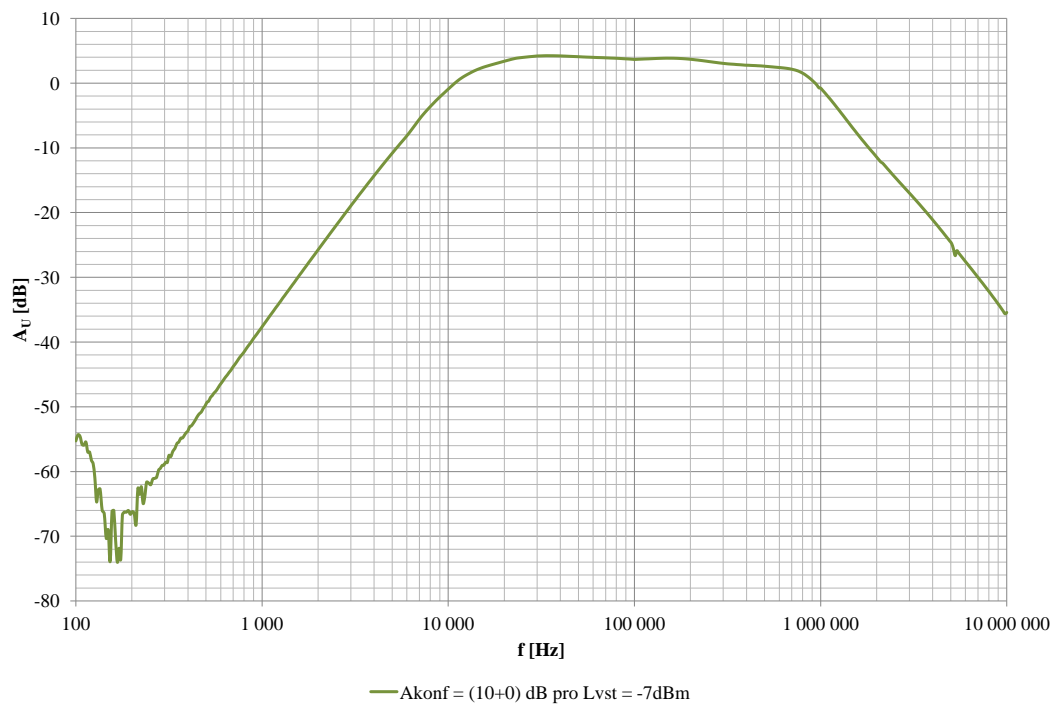
## B VÝSLEDKY NEDIFERENČNÍHO MĚŘENÍ NA OB- VODOVÉM ANALYZÁTORU



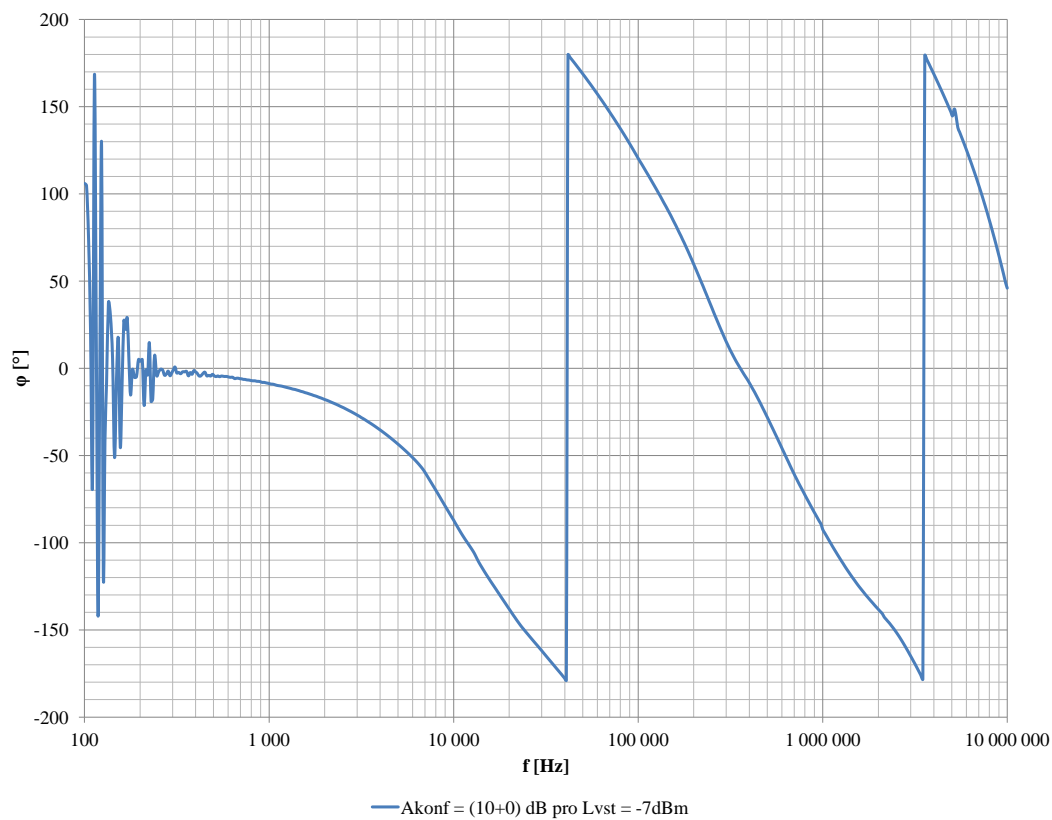
Obr. B.1: Frekvenční závislost zesílení  $A_U(f)$  pro blok LNA (větší oblast spektra)



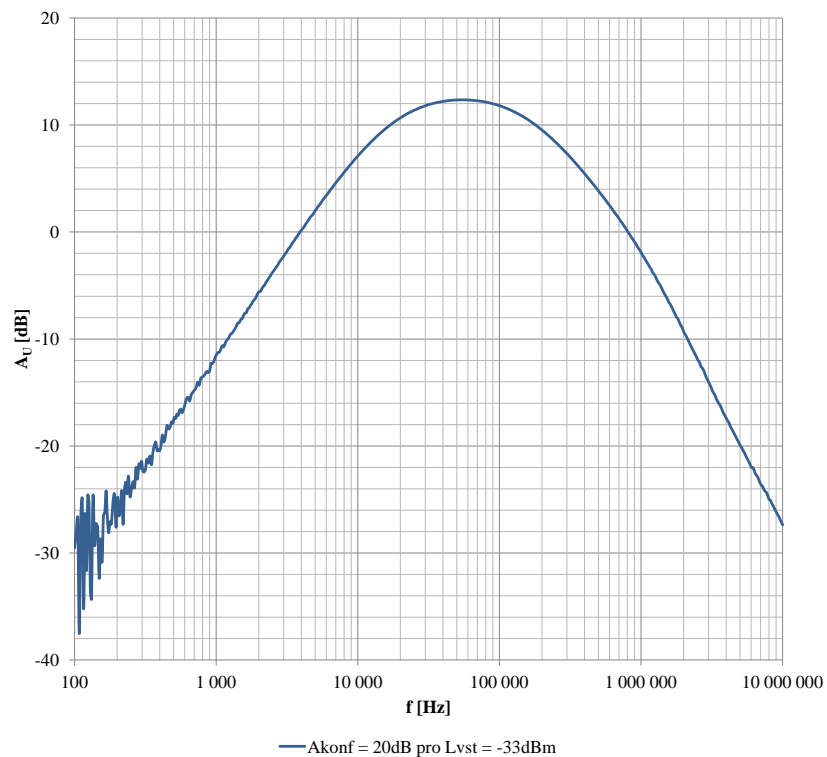
Obr. B.2: Fázová charakteristika  $\varphi(f)$  pro blok LNA (větší oblast spektra)



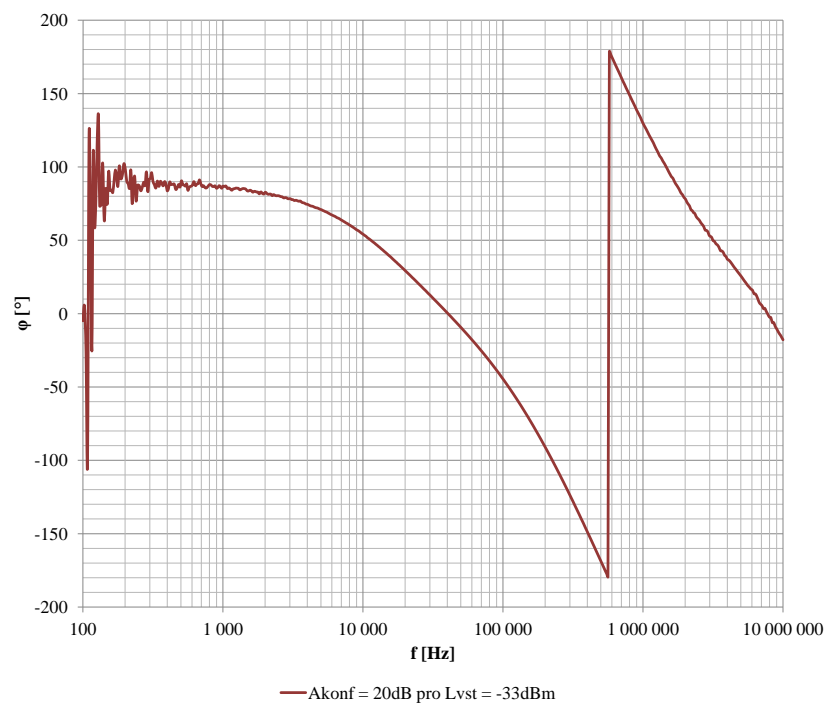
Obr. B.3: Frekvenční závislost zesílení  $A_U(f)$  pro blok VGA (větší oblast spektra)



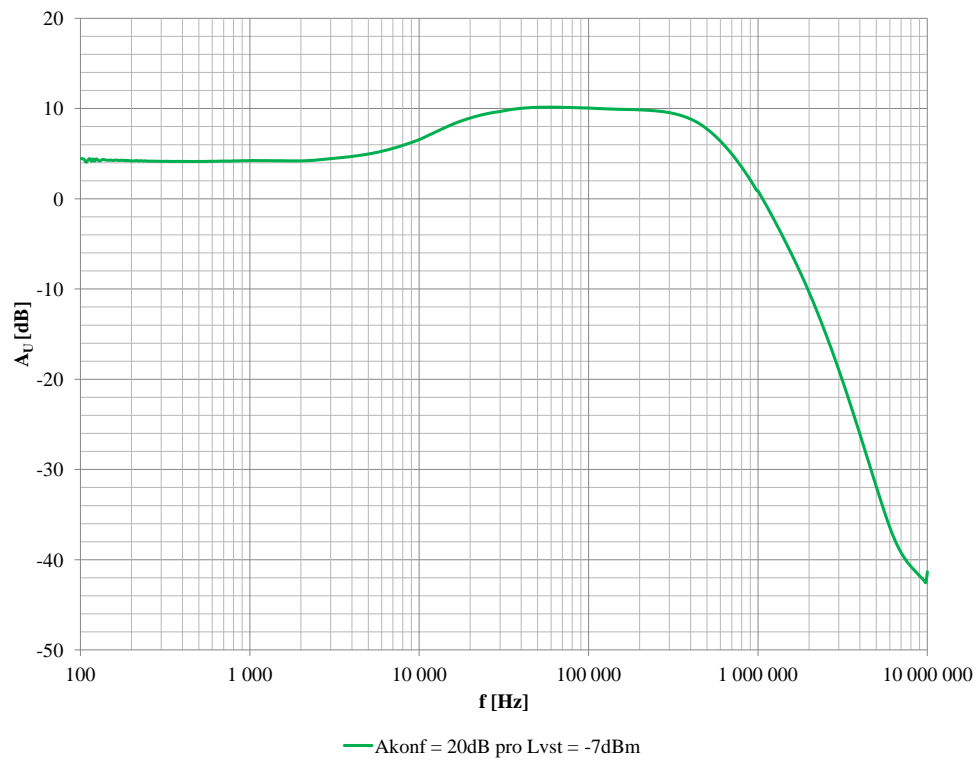
Obr. B.4: Fázová charakteristika  $\varphi(f)$  pro blok VGA (větší oblast spektra)



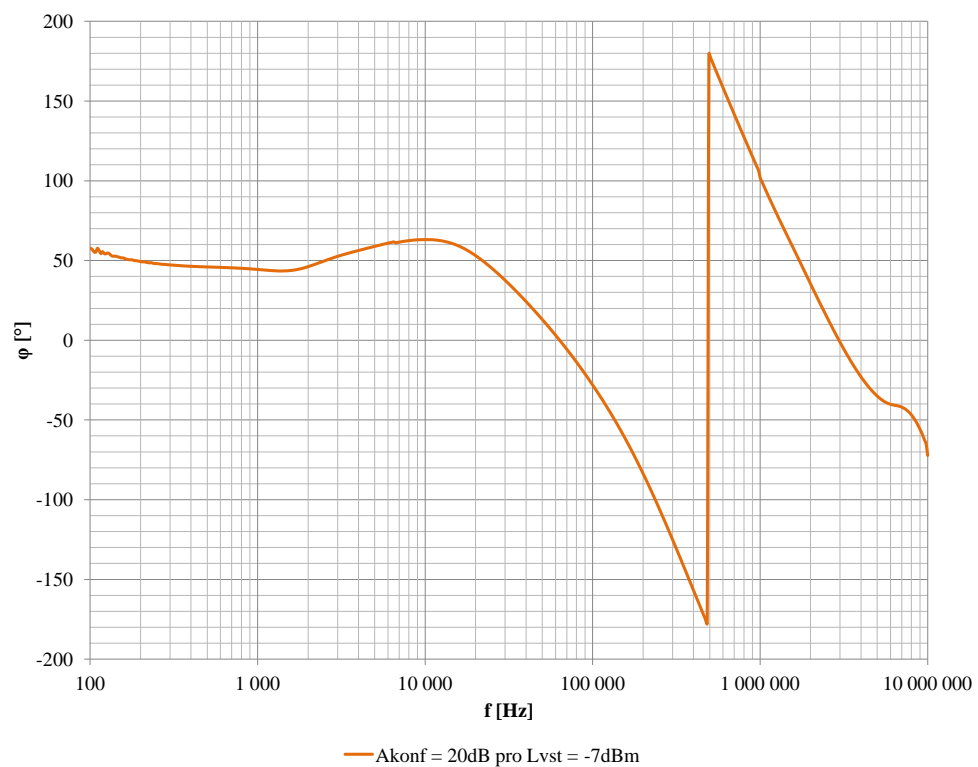
Obr. B.5: Frekvenční závislost zesílení  $A_U(f)$  pro podblok VGA2 bloku VGA (větší oblast spektra)



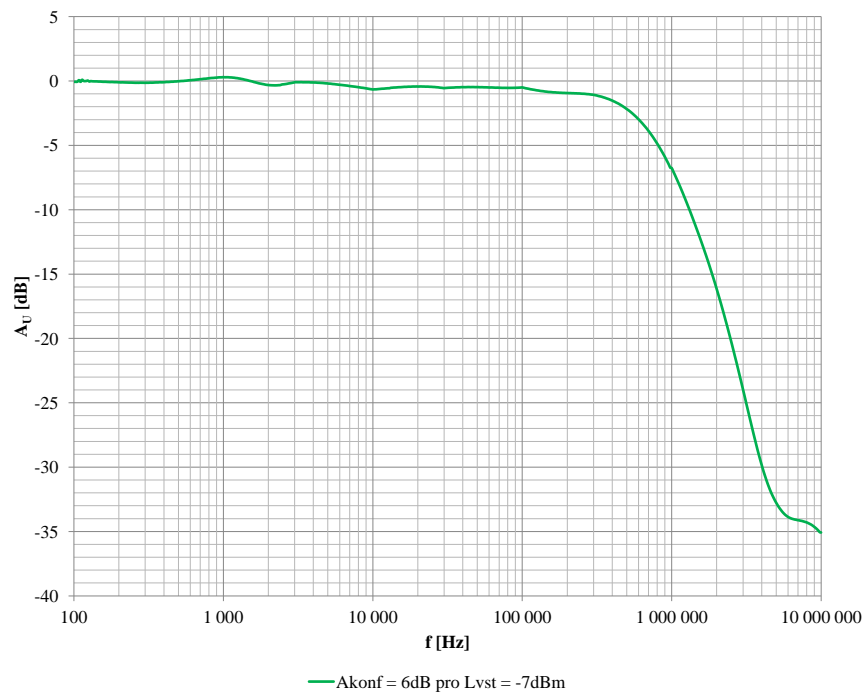
Obr. B.6: Fázová charakteristika  $\varphi(f)$  pro podblok VGA2 bloku VGA (větší oblast spektra)



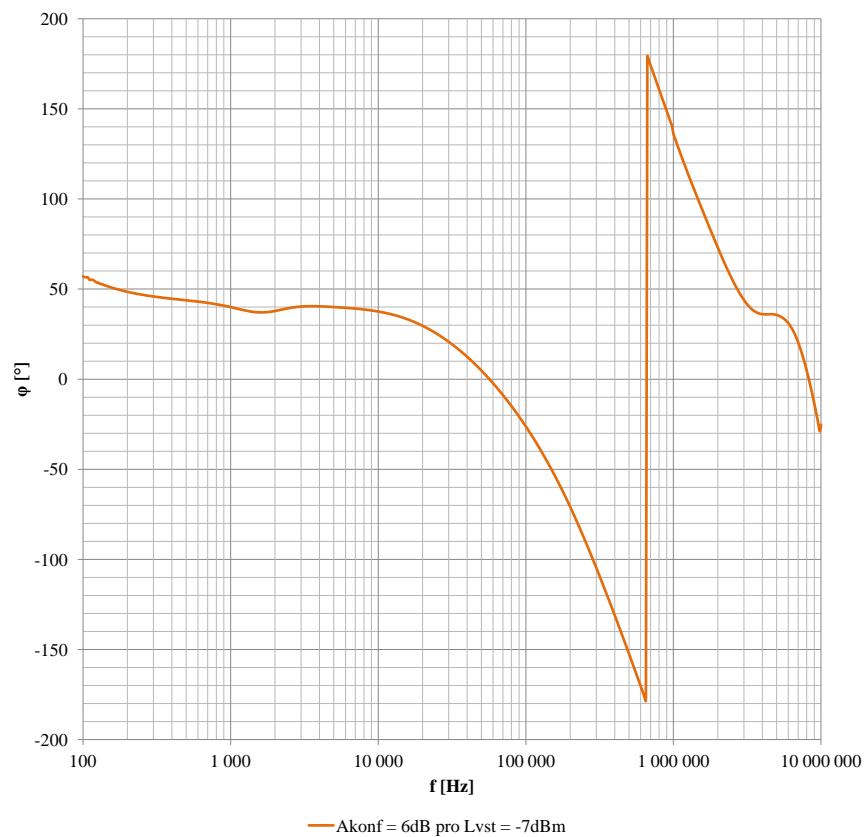
Obr. B.7: Frekvenční závislost zesílení  $A_U(f)$  pro podblok A2 bloku FGA (větší oblast spektra)



Obr. B.8: Fázová charakteristika  $\varphi(f)$  pro podblok A2 bloku FGA (větší oblast spektra)

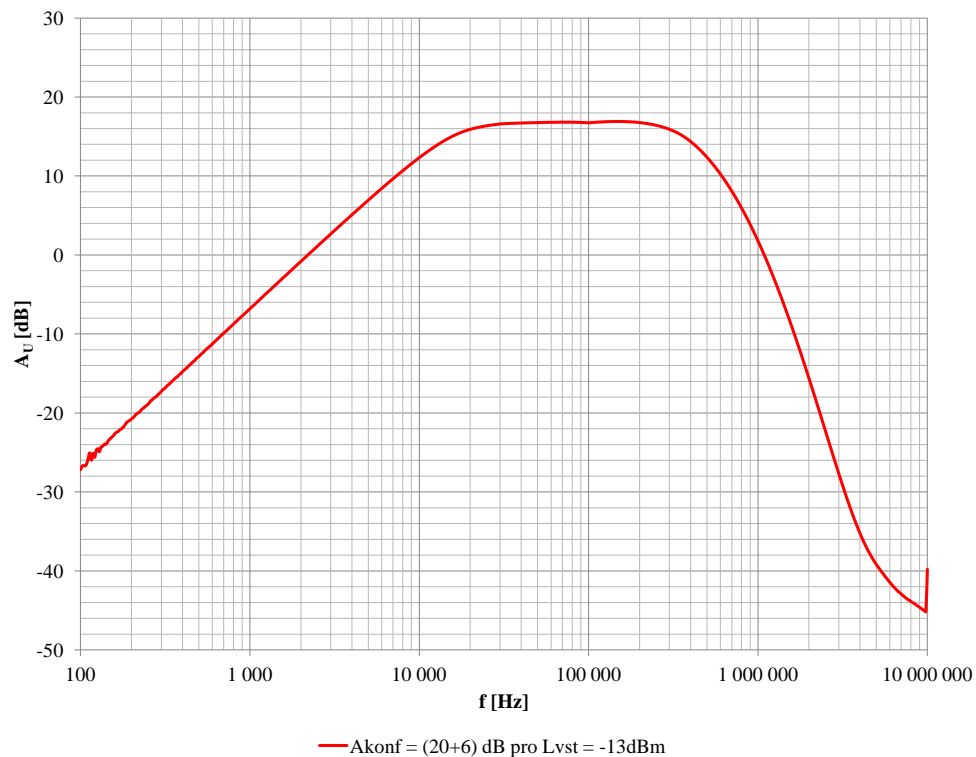


Obr. B.9: Frekvenční závislost zesílení  $A_U(f)$  pro podblok A3 bloku FGA (větší oblast spektra)

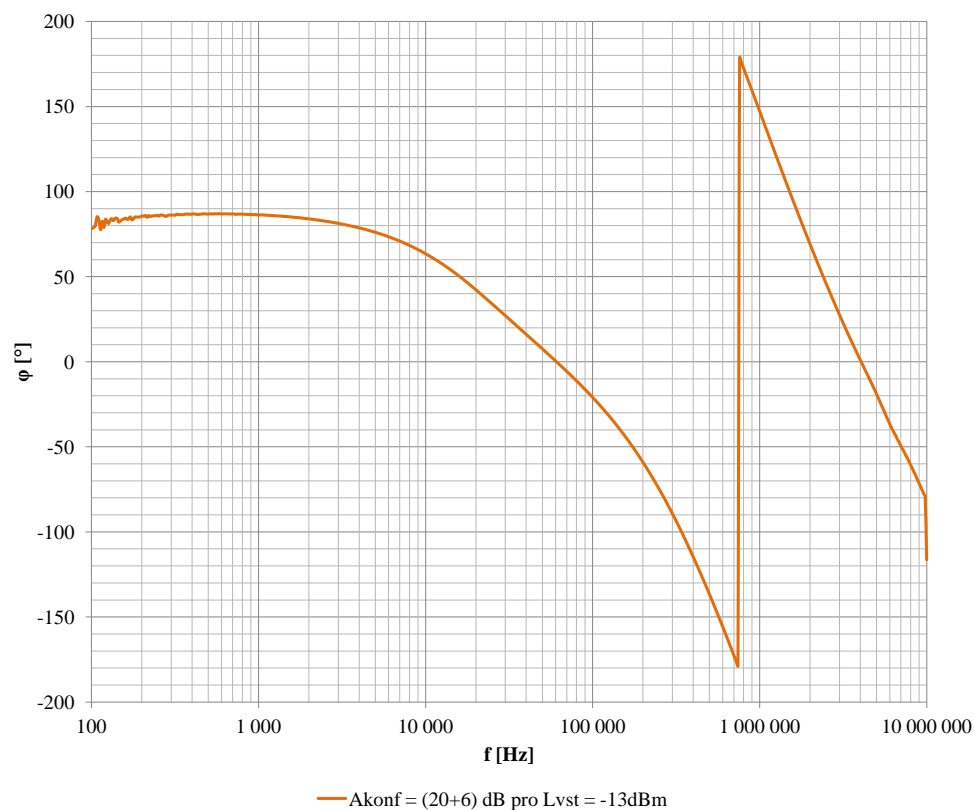


Obr. B.10: Fázová charakteristika  $\varphi(f)$  pro podblok A3 bloku FGA (větší oblast spektra)

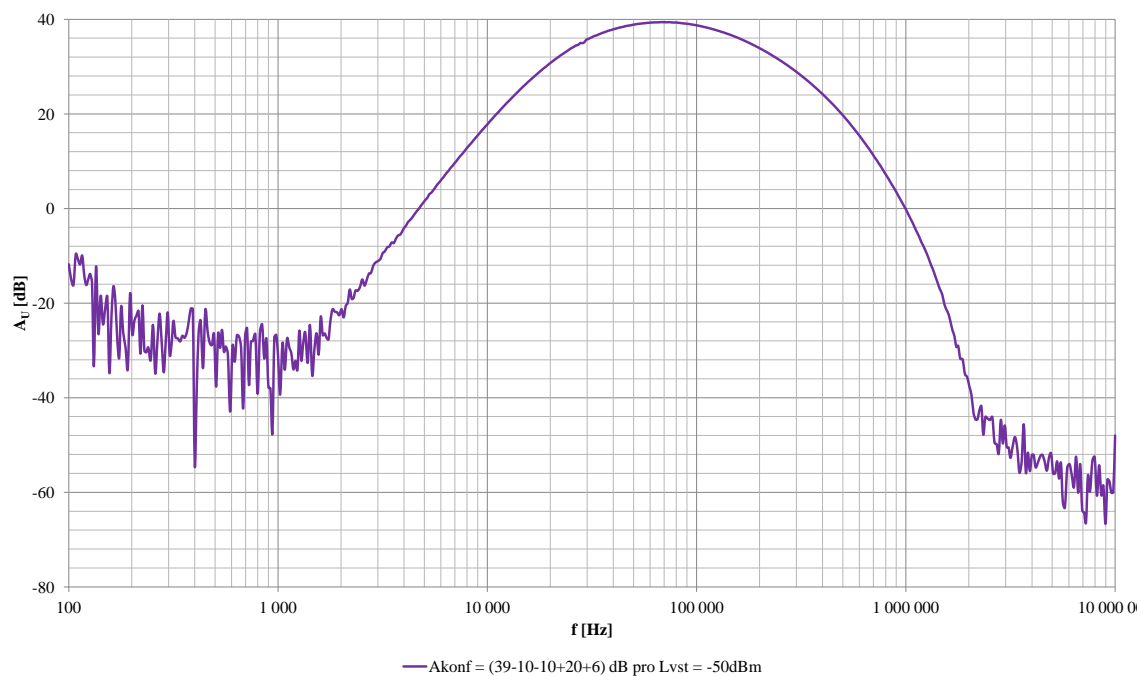




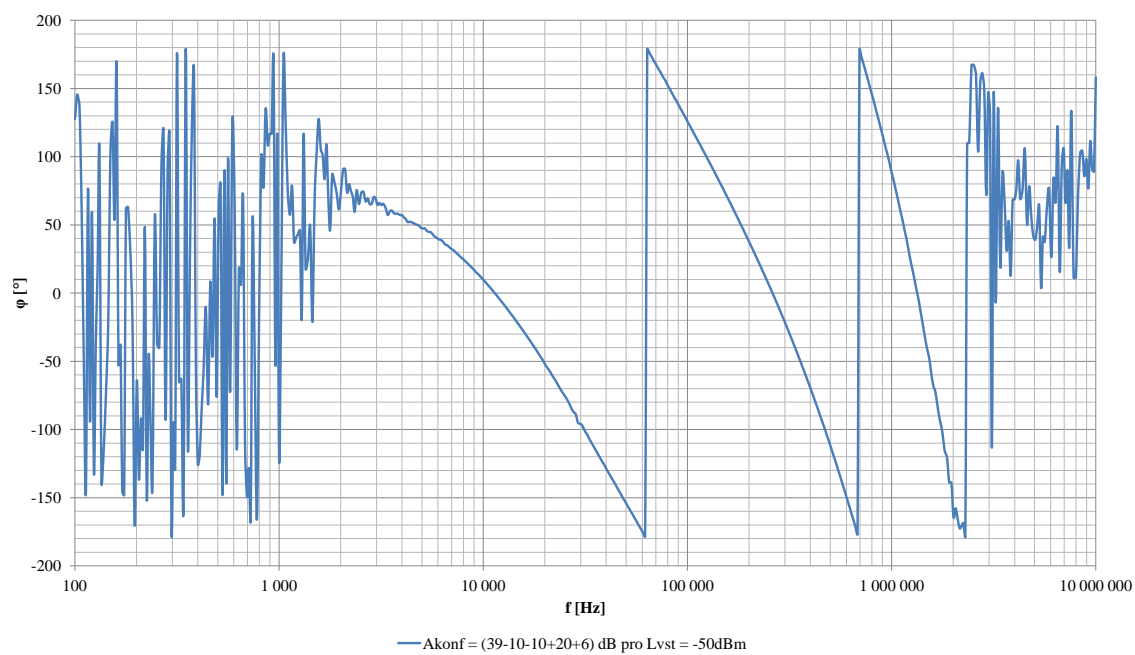
Obr. B.11: Frekvenční závislost zesílení  $A_U(f)$  pro blok FGA (větší oblast spektra)



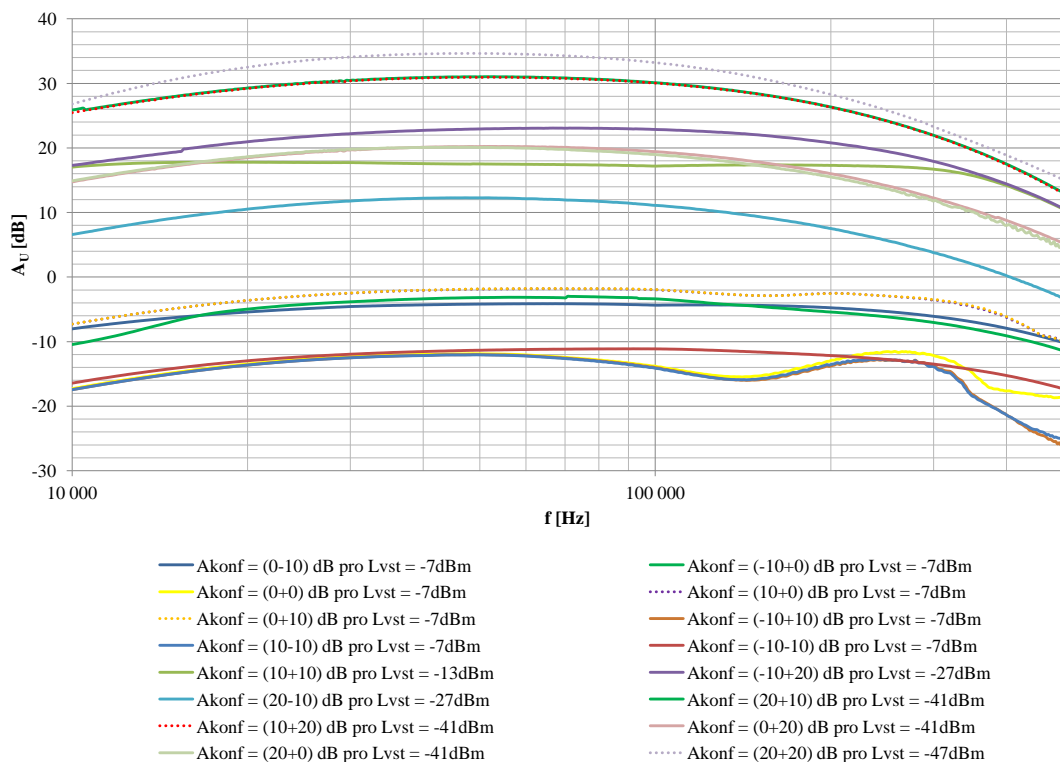
Obr. B.12: Fázová charakteristika  $\varphi(f)$  pro blok FGA (větší oblast spektra)



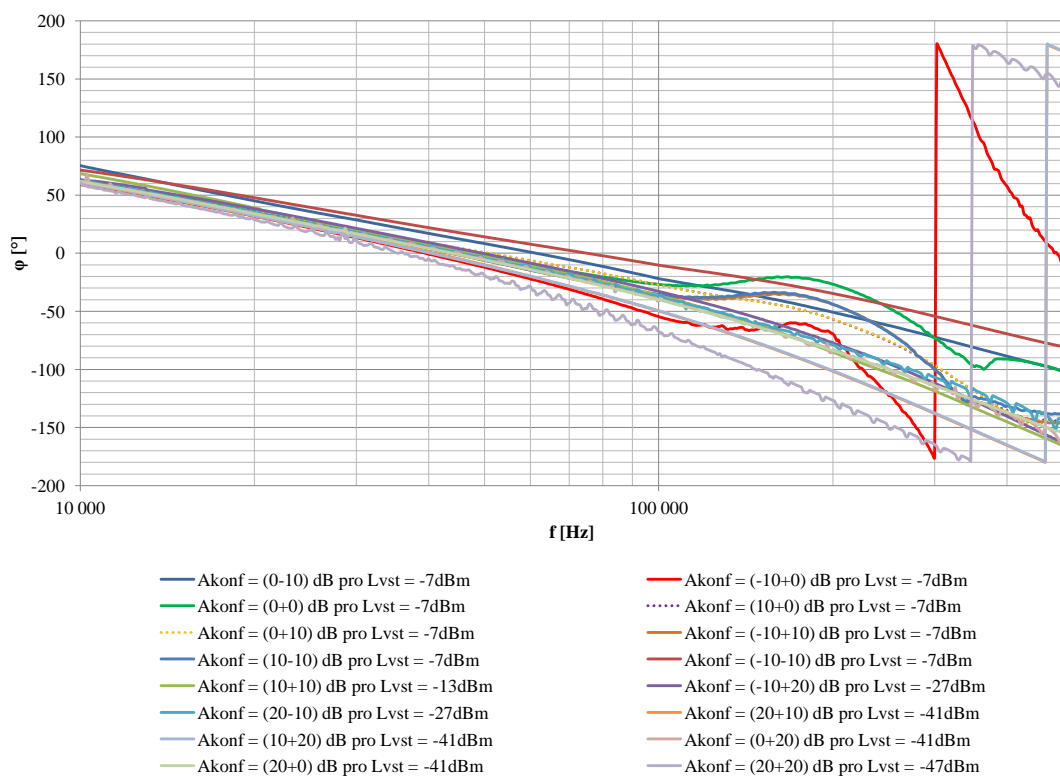
Obr. B.13: Frekvenční závislost zesílení  $A_U(f)$  pro řetězec LNVGA (větší oblast spektra)



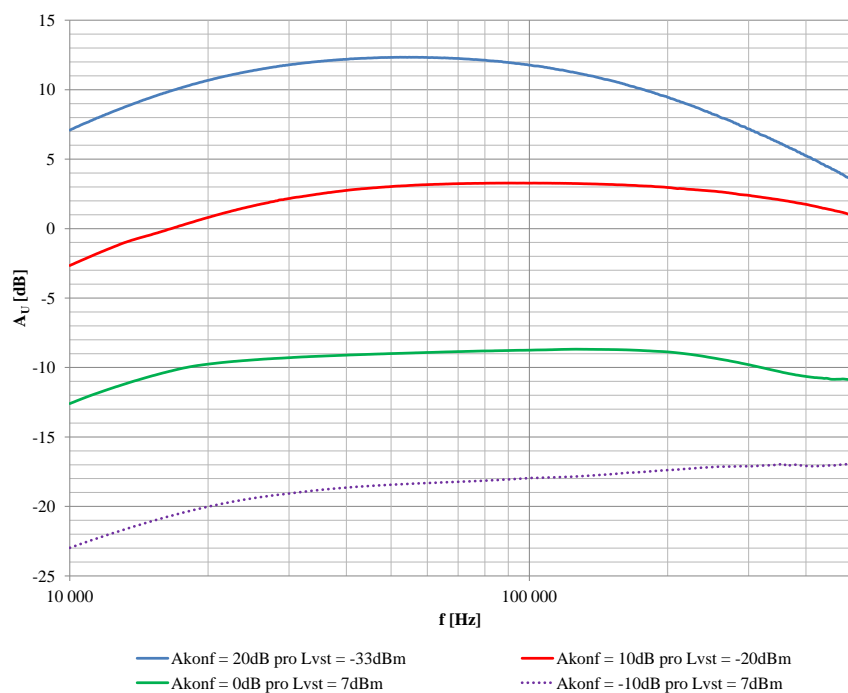
Obr. B.14: Fázová charakteristika  $\varphi(f)$  pro řetězec LNVGA (větší oblast spektra)



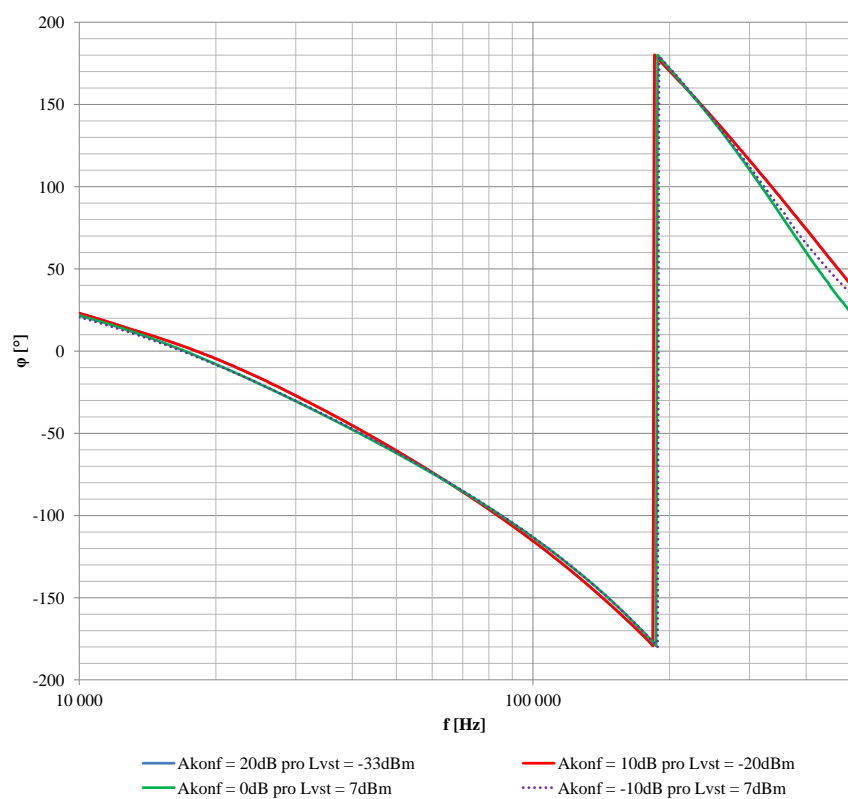
Obr. B.15: Frekvenční závislost zesílení  $A_U(f)$  pro blok VGA (detail v pracovní oblasti)



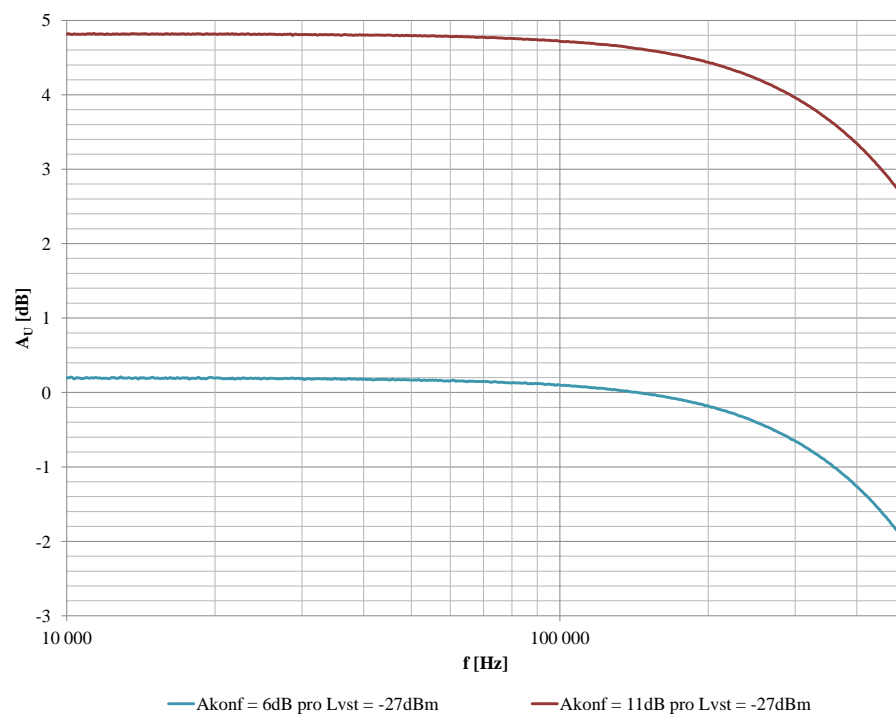
Obr. B.16: Fázová charakteristika  $\varphi(f)$  pro blok VGA (detail v pracovní oblasti)



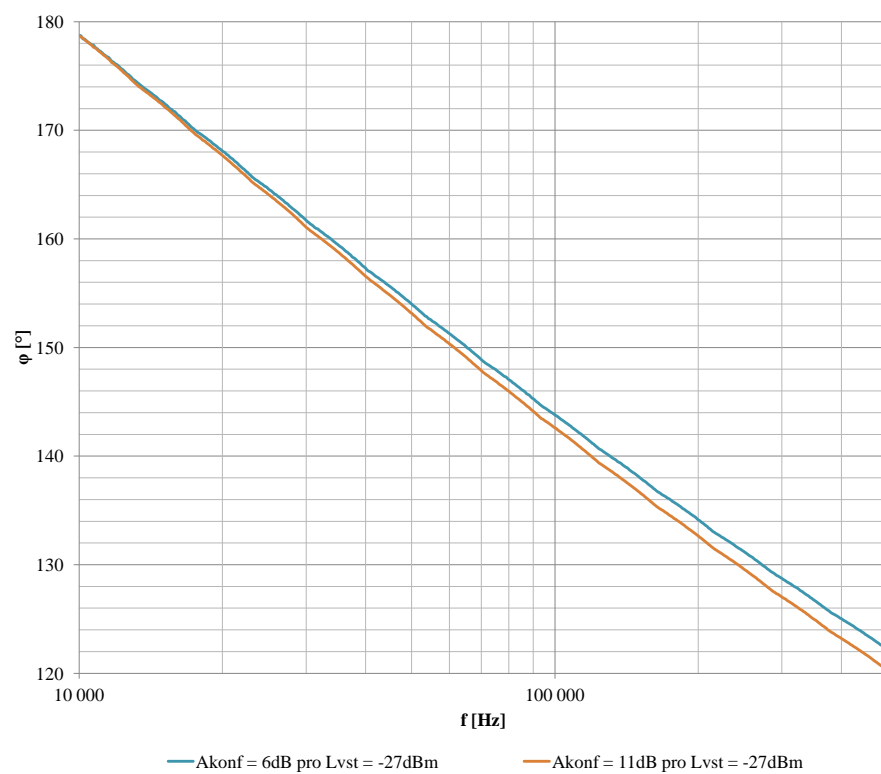
Obr. B.17: Frekvenční závislost zesílení  $A_U(f)$  pro podblok VGA2 bloku VGA (detail v pracovní oblasti)



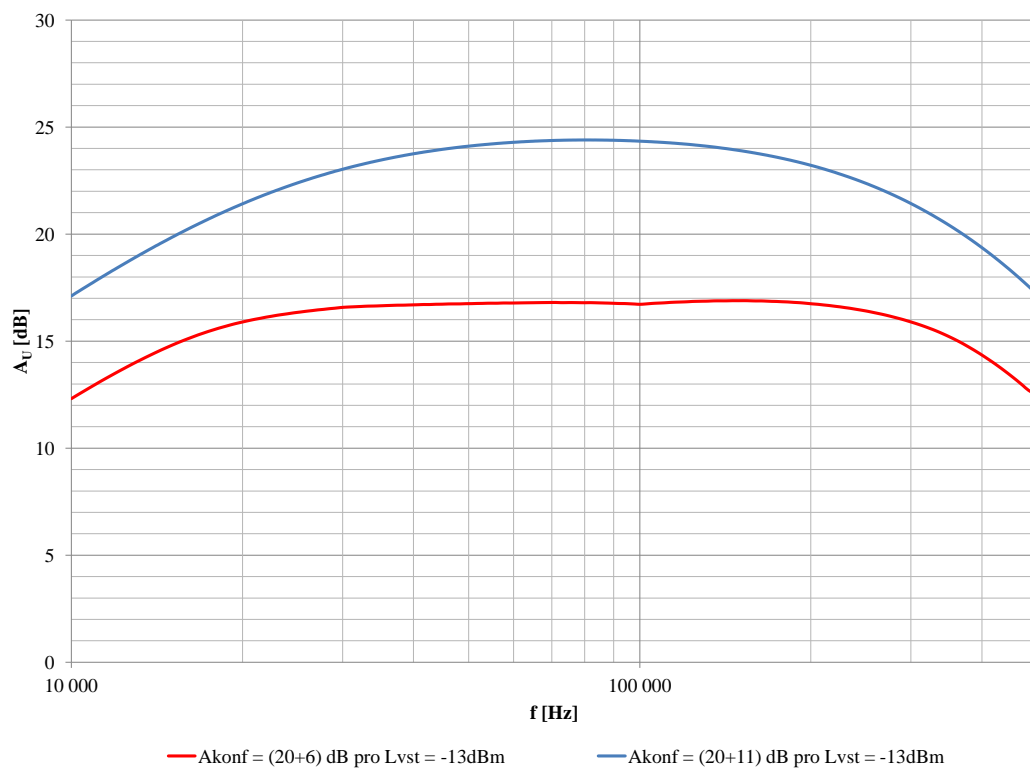
Obr. B.18: Fázová charakteristika  $\varphi(f)$  pro podblok VGA2 bloku VGA (detail v pracovní oblasti)



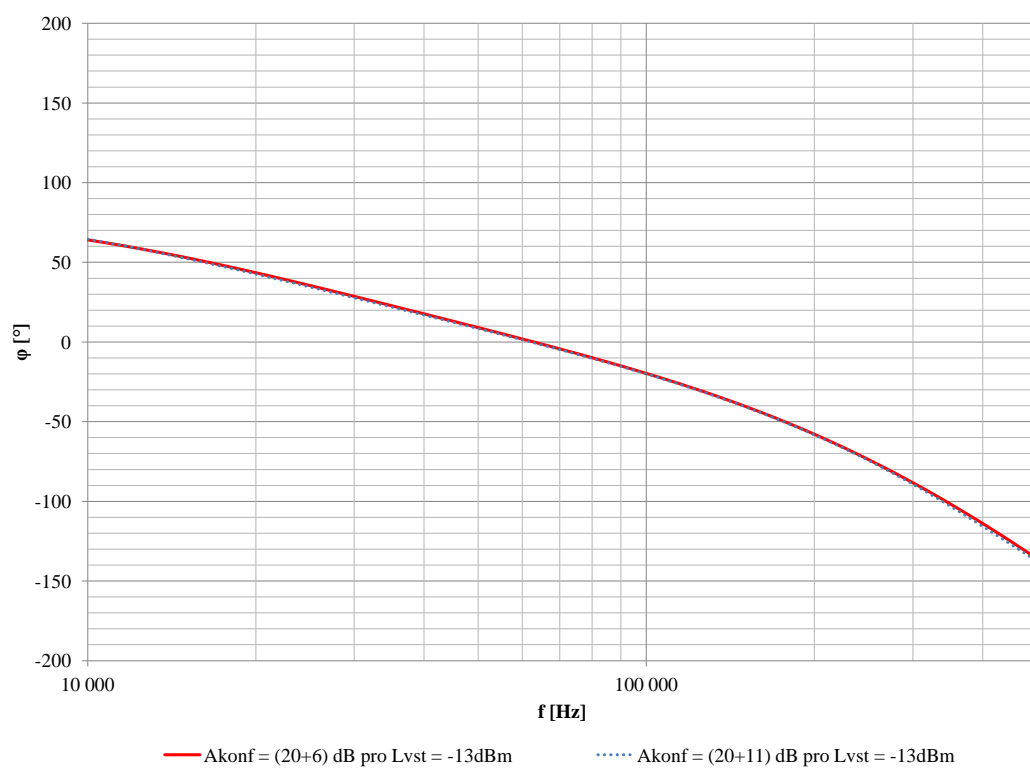
Obr. B.19: Frekvenční závislost zesílení  $A_U(f)$  pro podblok A3 bloku FGA (detail v pracovní oblasti)



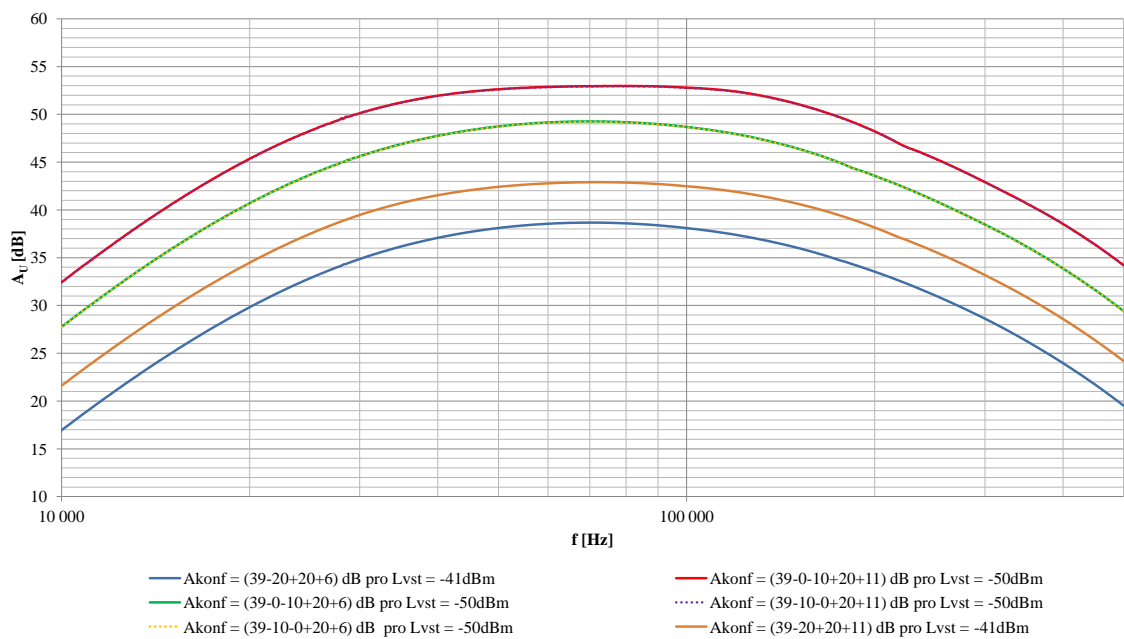
Obr. B.20: Fázová charakteristika  $\varphi(f)$  pro podblok A3 bloku FGA (detail v pracovní oblasti)



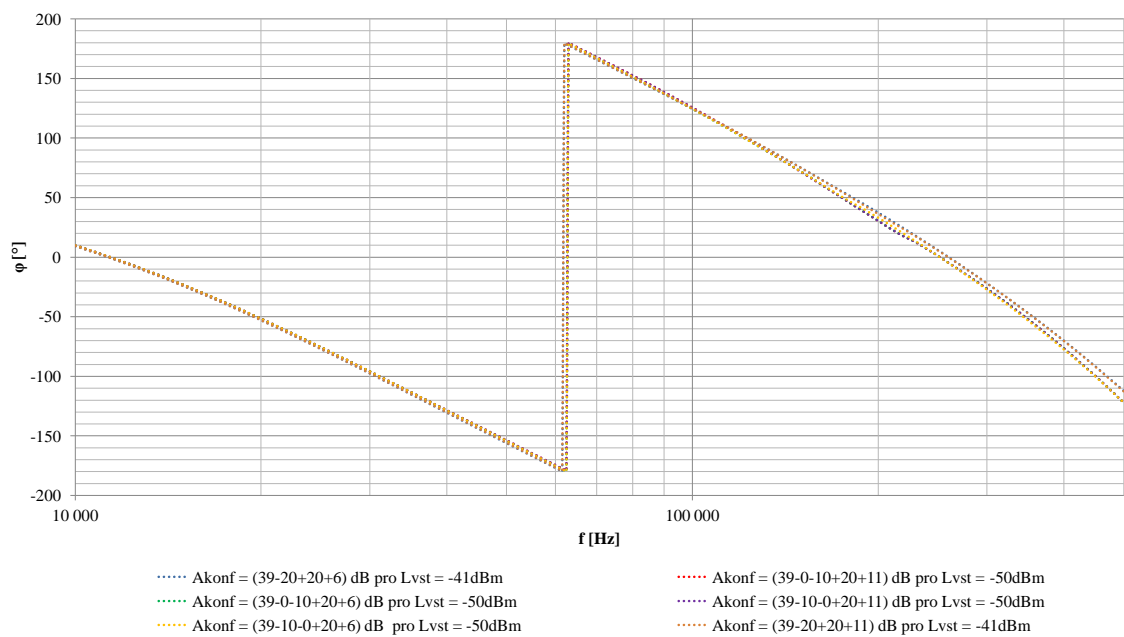
Obr. B.21: Frekvenční závislost zesílení  $A_U(f)$  pro blok FGA (detail v pracovní oblasti)



Obr. B.22: Fázová charakteristika  $\varphi(f)$  pro blok FGA (detail v pracovní oblasti)



Obr. B.23: Frekvenční závislost zesílení  $A_U(f)$  pro řetězec LNVGA (detail v pracovní oblasti)



Obr. B.24: Fázová charakteristika  $\varphi(f)$  pro řetězec LNVGA (detail v pracovní oblasti)

## **C VÝKRESY SCHÉMAT DESEK PLOŠNÝCH SPOJŮ**

### **C.1 Popis obsahu přiloženého CD-ROM**

Na přiloženém médiu se nachází kromě elektronické verze bakalářské práce též projekty obou ovládacích aplikací včetně stručného návodu k použití (ovládací aplikace pro PC a firmware pro mikrokontrolér) a projekty návrhů desek plošných spojů (v systému EAGLE).

### **C.2 Výkresy schémat**

Na samostatných listech vkládám schémata desek plošných spojů navrhovaných v systému EAGLE.