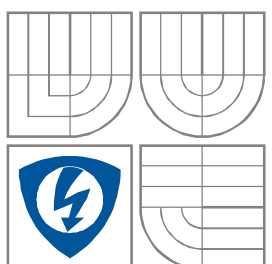


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

KMITOČTOVÉ SYNTÉZÁTORY

FREQUENCY SYNTHESIZERS

DIPLOMOVÁ PRÁCE

DIPLOMA THESIS

AUTOR PRÁCE

AUTHOR

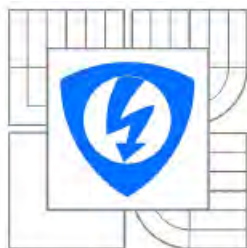
Bc. JOSEF LAPČÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ DŘÍNOVSKÝ, Ph.D.

BRNO, 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Josef Lapčík
Ročník: 2

ID: 73018
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Kmitočtové syntezátory

POKYNY PRO VYPRACOVÁNÍ:

Podrobně prostudujte možnosti generování harmonických signálů, pomocí přímé a nepřímé kmitočtové syntézy. Na základě vašeho studia navrhnete dva typy syntezátorů s odlišným principem jako podklady pro laboratorní přípravek. Vámi navržené přípravy by měli také obsahovat nezbytný komunikační hardware, který umožní řízení těchto syntezátorů pomocí PC a bude k němu připojován nejlépe pomocí USB sběrnice.

Realizujte vámi navržené syntezátory jako laboratorní přípravy a provedte na nich měření základních parametrů.

Na základě vámi realizovaných přípravek vypracujte vhodné laboratorní návody pro předmět Radioelektronická měření. Tyto návody doplňte modelovým příkladem pro měření v programovém prostředí VEE.

DOPORUČENÁ LITERATURA:

[1] HANUS, S., SVAČINA, J. Vysokofrekvenční a mikrovlnná technika. Elektronický učební text, Brno: FEKT VUT v Brně, 208 stran, www.feec.vutbr.cz.

[2] Agilent technologies. Phase-Locked Loops (PLL). [online], [cit. dne 18.12.2009], dostupné na www.home.agilent.com/agilent/application.jspx?nid=-34525.0.00&pageMode=OVW&lc=eng&cc=US.

Termín zadání: 7.2.2011

Termín odevzdání: 20.5.2011

Vedoucí práce: Ing. Jiří Dřínovský, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

Anotace

Diplomová práce se zabývá rozborem a dělením frekvenčních syntezátorů a návrhem syntezátorů DDS a PLL. Jsou rozebrány základní typy syntezátorů a odlišnosti mezi způsoby jejich činnosti. V části návrhu jsou v práci podrobně popsány základní obvody obou syntezátorů a další obvody, které jsou pro návrh důležité. Po jednotlivých částech je podrobně rozebrán návrh DDS a PLL syntezátoru. Oba syntezátory jsou přímo realizovány a jsou k nim vytvořeny samostatné řídicí aplikace. PLL syntezátor je přizpůsoben řízení i přes prostředí Agilent VEE. V Agilent VEE je navržena konkrétní vzorová aplikace, která slouží jako podklad pro přiloženou laboratorní úlohu.

Klíčová slova:

Frekvenční syntéza, přímá digitální syntéza, fázový závěs, předdělič, napětím řízený oscilátor, stabilizátor napětí, USB řadič, řídicí slovo.

Annotation

This diploma thesis concerns with analysis and dividing of frequency synthesizers and design of DDS, PLL synthesizers. Base types of frequency synthesizers are described including differences between methods of their operation. Base circuits of both – DDS and PLL synthesizers and other important circuits are described in details at design part of this thesis. Design of DDS and PLL synthesizer is described in particular sections. Both synthesizers are directly realized and stand-alone control applications are created. PLL synthesizer is also ready to control thru Agilent VEE program environment. Particular example application is designed in Agilent VEE. This application is used as basis of attached lab project.

Key words:

Frequency synthesis, direct digital synthesis (DDS), phase locked loop (PLL), prescaler, voltage control oscillator (VCO), voltage stabilizer, USB controller, control word.

Bibliografická citace

Bc. LAPČÍK, J. *Kmitočtové syntezátory*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 58 s, 6 příloh. Vedoucí diplomové práce Ing. Jiří Dřínovský, Ph.D.

Prohlášení

Jako autor diplomové práce na téma „Kmitočtové syntezátory“ dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 20. května 2011

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jiřímu Dřínovskému, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 20 května 2011

.....
podpis autora

Obsah

1	Úvod.....	- 1 -
2	Rozdělení syntezeátorů.....	- 1 -
2.1	Syntezeátory s přímou koherentní syntézou	- 1 -
2.2	Syntezeátory s nepřímou koherentní syntézou	- 3 -
2.2.1	Syntezeátory bez předděliče	- 3 -
2.2.2	Syntezeátory s pevným předděliče	- 5 -
2.2.3	Syntezeátory s řízeným předděličem	- 6 -
2.3	Syntezeátory s přímou nekoherentní syntézou	- 7 -
3	Návrh DDS syntezeátoru - s přímou koherentní syntézou	- 8 -
3.1	AD9958	- 8 -
3.2	USB řadič EZ-USB FX2 – CY7C68013	- 12 -
3.3	Návrh zdroje referenčního hodinového signálu.....	- 13 -
3.4	Návrh výstupních obvodů.....	- 13 -
3.5	Návrh komunikačního rozhraní	- 15 -
3.6	Návrh napájení	- 18 -
3.7	Deska plošného spoje.....	- 19 -
3.8	Komunikace s PC.....	- 22 -
3.9	Návrh řídicí aplikace	- 24 -
3.9.1	Popis fungování aplikace	- 24 -
3.10	Měření výstupní frekvence.....	- 25 -
4	Návrh PLL syntezeátoru – s nepřímou koherentní syntézou.....	- 29 -
4.1	ADF4252	- 29 -
4.2	USB řadič FT232R.....	- 31 -
4.3	VCO ROS-2500+	- 33 -
4.4	Návrh zdroje referenčního hodinového signálu.....	- 33 -
4.5	Návrh výstupních obvodů a smyčky	- 34 -
4.6	Návrh komunikačního rozhraní	- 37 -
4.7	Návrh napájení	- 39 -
4.8	Deska plošného spoje.....	- 40 -
4.9	Komunikace s PC.....	- 43 -
4.10	Návrh řídicí aplikace	- 43 -
4.10.1	Popis fungování aplikace	- 43 -
4.11	Řízení pomocí Agilent VEE.....	- 44 -

4.11.1	Popis fungování knihovny pro Agilent VEE.....	- 44 -
4.11.2	Návrh VEE programu pro řízení syntezátoru.....	- 45 -
4.12	Měření výstupní frekvence.....	- 47 -
5	Závěr	- 49 -
6	LITERATURA.....	- 50 -
7	SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK	- 51 -
8	Seznam příloh	- 52 -

1 Úvod

Tato Diplomová práce se zabývá Kmitočtovými syntezátory, což jsou zdroje signálů (obvykle sinusového nebo pravouhlého průběhu) přesných kmitočtů, které mohou nabývat diskretních hodnot v určitém rozsahu. Na rozdíl od běžných oscilátorů nelze výstupní kmitočet měnit plynule, ale po obvykle ekvidistantních krocích. Rozdíl dvou po sobě následujících výstupních kmitočtů se nazývá *kmitočtový krok syntezátoru*.

Syntezátory jsou realizovány velmi často jako hybridní, využívající obvody analogové i digitální. Dnes se syntezátory vyrábějí i jako integrované obvody v klasickém i SMT provedení. V této Diplomové práci bude využito právě syntezátorů v integrovaných obvodech. Řízení se provádí mikroprocesorem pomocí paralelní nebo sériové sběrnice.

2 Rozdělení syntezátorů

Podle způsobu vytváření výstupního signálu rozdělujeme kmitočtové syntezátory na *syntezátory s přímou syntézou* a *syntezátory s nepřímou syntézou*. U syntezátorů s přímou syntézou se výstupní kmitočet vytváří z kmitočtu jednoho nebo několika základních generátorů pomocí aritmetických operací – sčítání, odečítání, násobení a dělení realizovaných soustavou směšovačů, násobičů a děličů kmitočtu [1]. Syntezátory s nepřímou syntézou využívají zpětné vazby realizované fázovým závěsem – PLL (Phase-Locked Loop). Kromě směšovačů, násobičů a děličů kmitočtu, obsahují i napětím řízené oscilátory, programovatelné děliče kmitočtu, kmitočtové fázové detektory atd.

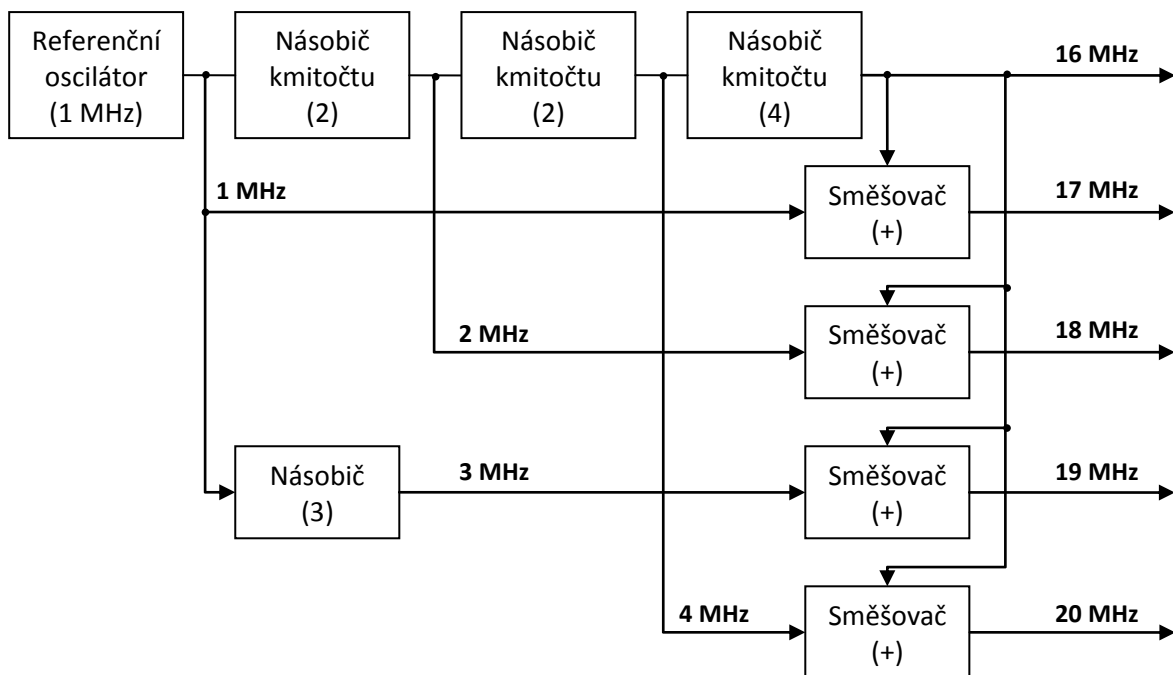
Syntezátory dále dělíme na *koherentní* a *nekoherentní* a to podle počtu základních generátorů. V koherentních syntezátorech se používá pouze jeden základní generátor a výstupní signály jsou tedy vzájemné koherentní (synchronní). Stabilita výstupního kmitočtu je zde dána stabilitou základního generátoru. U nekoherentních syntezátorů se využívá několik základních generátorů, takže jsou výstupní signály nekoherentní a stanovení stability výstupního kmitočtu je náročnější.

2.1 Syntezátory s přímou koherentní syntézou

Jak již bylo řečeno výše, základním blokem je referenční oscilátor s vysokou stabilitou, z jehož signálu se odvozují všechny výstupní signály syntezátoru. Stabilita a přesnost výstupního kmitočtu syntezátoru jsou dány pouze stabilitou a přesností kmitočtu referenčního oscilátoru.

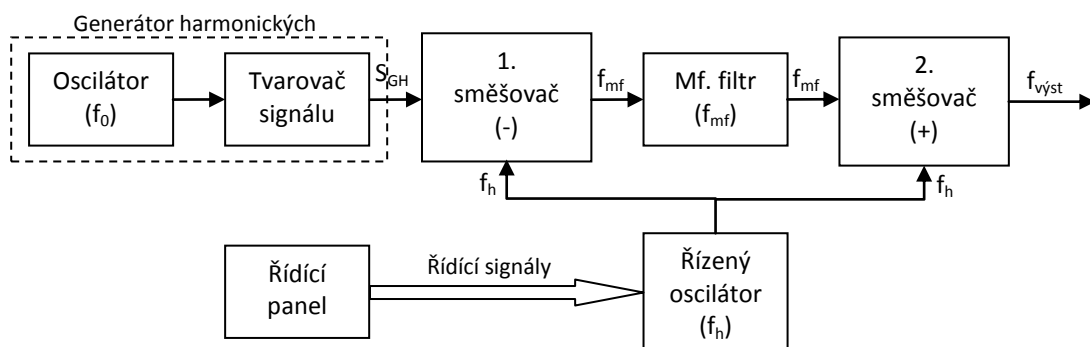
U těchto syntezátorů rozlišujeme dva způsoby, na základě kterých pracují:

1. **Metoda přímé přeměny**, kterou ilustruje obr. 1 a která je založena na přímém násobení a směšování signálu referenčního oscilátoru (1 MHz). Kmitočet referenčního oscilátoru zároveň určuje kmitočtový krok syntezátoru (1 MHz). Vzhledem k tomu, že násobiče a směšovače kmitočtu jsou nelineární obvody, je zapotřebí na jejich výstupu potlačit selektivními obvody nežádoucí signály. Tato metoda řešení syntezátoru se používá jen v případě, že požadujeme relativně malý počet výstupních kmitočtů, přičemž je zde výhoda využití všech kmitočtů současně.



Obr. 1 Syntezátor s přímou koherentní syntézou – metoda přímé přeměny

2. **Metoda harmonických**, kterou ilustruje obr. 2. Zde je základním blokem *generátor harmonických*, který obsahuje zdroj sinusového signálu s přesným a stabilním kmitočtem f_0 , za nímž je zařazen tvarovací obvod, který mění průběh signálu tak, aby obsahoval pokud možno co nejvíce harmonických složek. Tvarovačem může být například oboustranný omezovač.



Obr. 2 Syntezátor s přímou koherentní syntézou – metoda harmonických

Signál S_{GH} na výstupu generátoru harmonických obsahuje teoreticky nekonečně mnoho harmonických složek, což lze vyjádřit pomocí:

$$S_{GH} \approx \sum_{m=x_1}^{x_2} m f_0 + R, \quad (1)$$

kde x_1 je řád nejnižší harmonické požadované na výstupu syntezátoru, x_2 je řád nejvyšší harmonické požadované na výstupu syntezátoru a symbol R označuje souhrnně všechny zbývající harmonické, které nejsou využity. Výstupní kmitočty syntezátoru jsou tedy v rozsahu $x_1 f_0$ až $x_2 f_0$ s krokem f_0 . Ze signálu S_{GH} je požadovaná harmonická vybrána aktivní filtrací s dvojitým směřováním. V prvním směšovači se vytvoří mezifrekvenční signál f_{mf} , jako rozdílový produkt se signálem

řízeného oscilátoru f_h , který zůstává konstantní pro libovolný výstupní kmitočet. Signál dále prochází mezifrekvenčním filtrem s vysokou selektivitou, ve kterém jsou všechny nežádoucí signály potlačeny. Ve druhém směšovači se již vytváří výstupní signál f_{vyst} a to směšováním opět se signálem řízeného oscilátoru f_h - tentokrát ale součtový produkt.

Díky dvojímu směšování nezávisí kmitočet výstupního signálu na kmitočtu řízeného oscilátoru f_h . Na výstupní signál se tedy nepřenáší ani žádné negativní vlivy řízeného oscilátoru jako je fázový šum, nestabilita apod.

Tuto vlastnost lze jednoduše ukázat na příkladu požadovaného výstupního kmitočtu $f_{vyst} = x_k f_0$, kde je třeba nastavit řízený oscilátor na kmitočet

$$f_h = x_k f_0 - f_{mf} \quad (2)$$

Pro kmitočet mezifrekvenčního signálu vytvořeného v 1. směšovači platí

$$f_{mf} = x_k f_0 - f_h, \quad (3)$$

neboť ze spektra signálu S_{GH} splňuje rovnici (3) pouze harmonická složka s kmitočtem $x_k f_0$. Na výstupu 2. směšovače se vytváří signál

$$f_{vyst} = f_{mf} + f_h. \quad (4)$$

Z výše uvedených rovnic tedy plyne:

$$f_{vyst} = f_{mf} + f_h = x_k f_0 - f_h + f_h = x_k f_0. \quad (5)$$

2.2 Syntezátory s nepřímou koherentní syntézou

Základním blokem je opět referenční oscilátor s vysokou stabilitou a přesností výstupního kmitočtu a fázový závěs, zapojený jako násobič kmitočtu. V zapojení syntezátoru se mohou dále nacházet násobiče, děliče kmitočtu, případně i směšovače pracující se součtovým nebo rozdílovým směšovacím produktem.

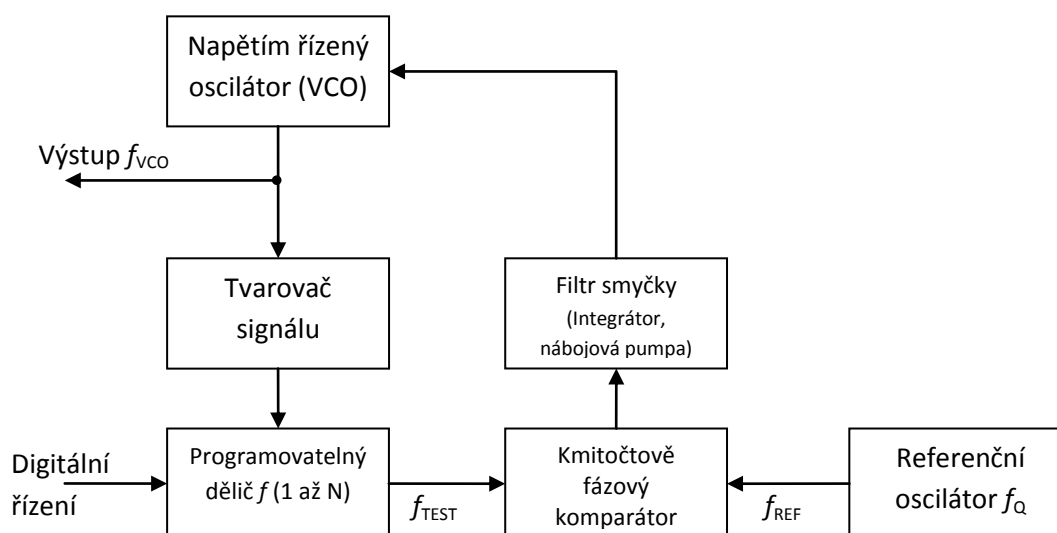
2.2.1 Syntezátory bez předděliče

Zobrazen na obr. 3. *Napětím řízený oscilátor (VCO)* je řízen signálem z filtru smyčky a jeho výstupní signál s kmitočtem f_{VCO} je současně i výstupním signálem syntezátoru. *Tvarovač signálu* upraví signál VCO na signál pravouhlý kvůli zpracování v digitálních obvodech *programovatelného děliče kmitočtu* s dělicím poměrem nastavitelným v rozsahu 1 až N. Řízení programovatelného děliče a tedy i celého syntezátoru se provádí přes paralelní či sériovou linku. Kmitočtový krok syntezátoru Δf_{VCO} je dán změnou kmitočtu výstupního signálu při změně dělicího poměru o jedničku.

Signál z výstupu programovatelného děliče (f_{TEST}) přichází na jeden vstup *kmitočtové fázového komparátoru*. Na jeho druhý vstup je přiváděn signál s kmitočtem f_{REF} , což je v tomto případě přímo výstupní signál *referenčního oscilátoru* s kmitočtem f_0 , který bývá konstruován jako krystalový oscilátor. Výstupní napětí kmitočtového fázového komparátoru může nabývat kladných nebo záporných hodnot. V případě, že oba vstupní signály jsou kmitočtově i fázově shodné, výstupní napětí komparátoru je nulové a syntezátor je v ustáleném stavu [1].

Posledním blokem je *filtr smyčky*, což je integrátor, který podle polarity vstupního napětí zvyšuje nebo snižuje své výstupní napětí s rychlostí závislou na časové konstantě integrátoru. Pokud je na vstupu nulové napětí, výstupní napětí se nemění a zůstává konstantní. Filtr smyčky bývá často doplněn zesilovačem, který upravuje rozsah výstupních

napětí do úrovní potřebných pro řízení VCO. Jako integrátor se používá buď integrátor s operačním zesilovačem buzeným z napěťového zdroje, nebo integrátor tvořený kondenzátorem a proudovým zdrojem, který kondenzátor nabíjí, resp. vybíjí podle polarity vstupního signálu (nábojová pumpa).



Obr. 3 Syntezátor s nepřímou koherentní syntézou – bez předděliče

Činnost můžeme ilustrovat na jednoduchém příkladě – prostřednictvím digitálního řízení se nastaví programovatelný dělič kmitočtu na hodnotu N . Ustálený stav syntezátoru je dán rovností kmitočtů na vstupech kmitočtově fázového komparátoru

$$f_{TEST} = f_{REF}. \quad (6)$$

V tomto ustáleném stavu je na výstupu integrátoru konstantní napětí, při kterém VCO kmitá na požadovaném kmitočtu f_{VCO} . Dostáváme:

$$\frac{f_{VCO}}{N} = f_Q \rightarrow f_{VCO} = Nf_Q. \quad (7)$$

Při změně dělicího poměru N se poruší rovnost kmitočtů na vstupech kmitočtově fázového komparátoru. To způsobí vzrůst nebo pokles napětí integrátoru a tím se VCO začne přeladovat k novému kmitočtu f_{VCO} . Přeladování trvá do doby, než je splněna opět podmínka (6) – nastává ustálený stav.

Kmitočtový krok syntezátoru je v tomto případě

$$\Delta f_{VCO} = Nf_Q - (N - 1)f_Q = f_Q. \quad (8)$$

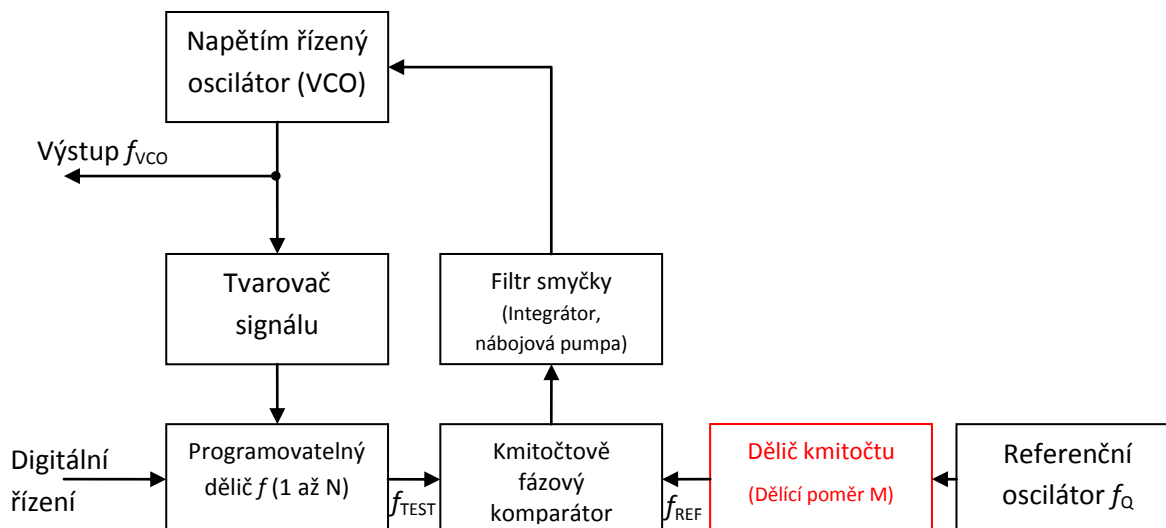
Krok syntezátoru je tedy roven přímo kmitočtu referenčního oscilátoru f_Q , který běžně bývá volen v rozsahu 1 MHz až 50 MHz. Z toho plyne, že takto navržený oscilátor by měl velmi „hrubý“ krok. Tuto nevýhodu lze řešit zapojením děliče kmitočtu, většinou s konstantním dělicím poměrem M , mezi referenční oscilátor a komparátor – obr. 4.

Při stanovení výstupního kmitočtu se vychází opět z podmínky (6) a po dosažení lze psát:

$$\frac{f_{VCO}}{N} = \frac{f_Q}{M} \rightarrow f_{VCO} = \frac{N}{M} f_Q. \quad (9)$$

Kmitočtový krok syntezátoru:

$$\Delta f_{VCO} = \frac{N}{M} f_Q - \frac{N-1}{M} f_Q = \frac{f_Q}{M}. \quad (10)$$

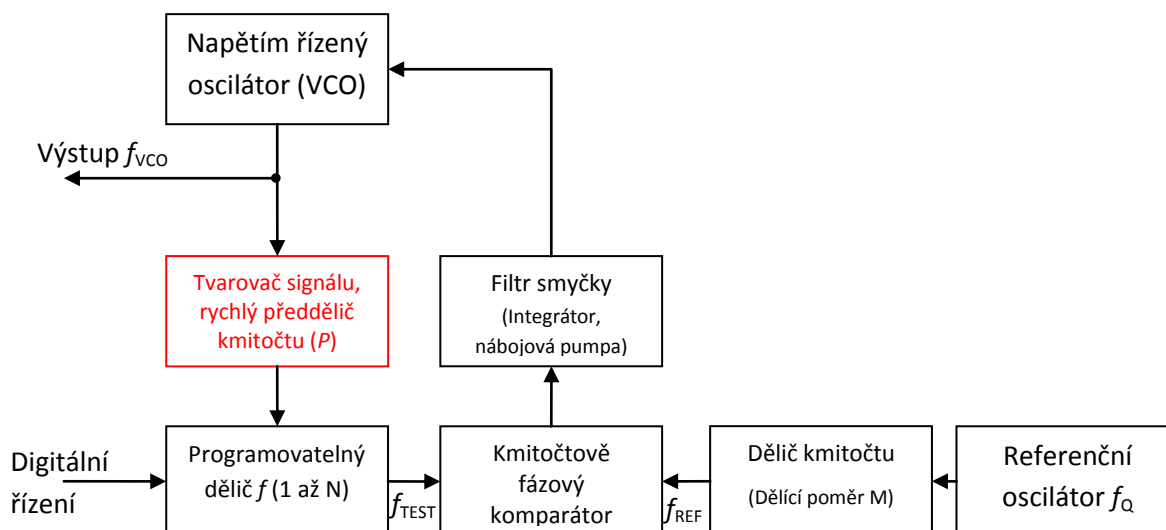


Obr. 4 Syntezátor s nepřímou koherentní syntézou s děličem f_Q – bez předděliče

Velkou nevýhodou zde uvedených syntezátorů je omezený kmitočtový rozsah, přibližně do desítek MHz. Programovatelný dělič musí zpracovat i nejvyšší kmitočet VCO a tím se promítá negativně jeho kmitočtové omezení.

2.2.2 Syntezátory s pevným předděličem

Zobrazen na obr. 5. Cílem je dosažení zvětšení kmitočtového rozsahu a to zařazením rychlého předděliče kmitočtu s dělicím poměrem P (pevný nebo nastavitelný v několika málo krocích) před programovatelný dělič kmitočtu. Díky tomu je kmitočet přivedený na vstup programovatelného děliče P -krát menší než nejvyšší kmitočet VCO. Součástí předděliče je i tvarovač signálu.



Obr. 5 Syntezátor s nepřímou koherentní syntézou – s pevným předděličem

Při stanovení výstupního kmitočtu se vychází opět z podmínky (6) a po dosažení lze psát:

$$\frac{f_{VCO}}{PN} = \frac{f_Q}{M} \rightarrow f_{VCO} = \frac{PN}{M} f_Q. \quad (11)$$

Kmitočtový krok syntezátoru:

$$\Delta f_{VCO} = \frac{PN}{M} f_Q - \frac{P(N-1)}{M} f_Q = \frac{P}{M} f_Q. \quad (12)$$

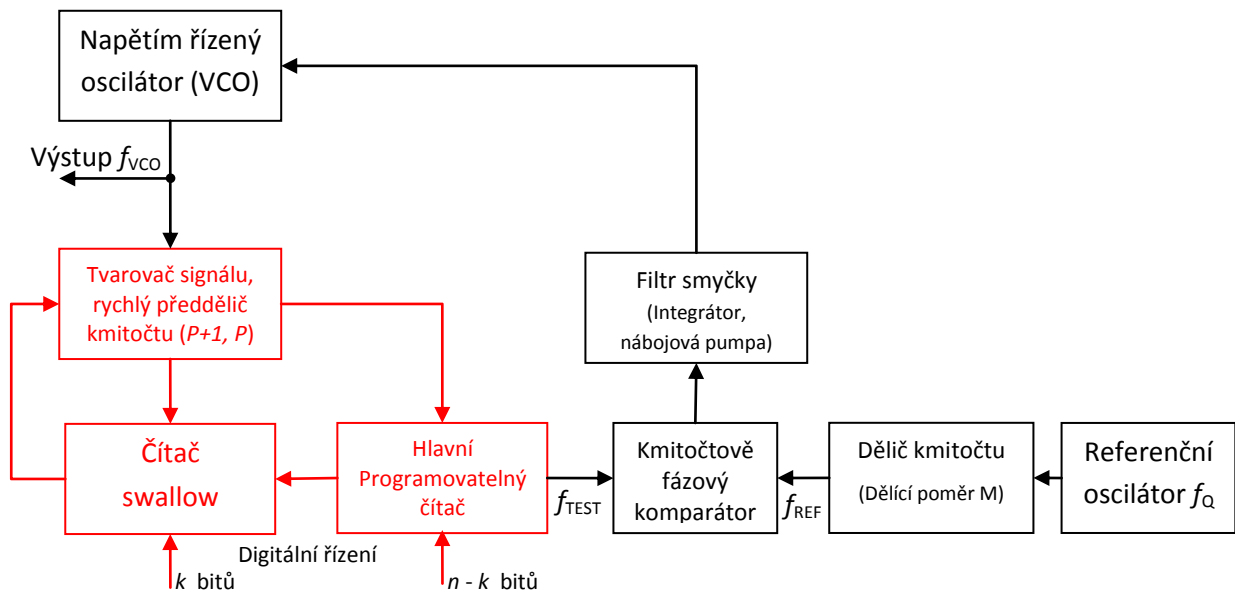
Kmitočtový krok se nám P -krát zvýšil a je nyní „hrubší“

$$\Delta f_{VCO} = P f_{REF}. \quad (13)$$

Abychom tuto nevýhodu eliminovali, museli bychom P -krát snížit pracovní kmitočet komparátoru f_{REF} , což znamená buď P -krát zvýšit dělicí poměr M nebo P -krát snížit kmitočet referenčního oscilátoru f_Q . Zmenšení pracovního kmitočtu komparátoru by však mělo za následek menší rychlost přeladování syntezátoru a zvýšení časové konstanty filtru smyčky (integrátoru) [1].

2.2.3 Syntezátory s řízeným předděličem

Zobrazen na obr. 6. Řeší obě výše zmíněné nevýhody – omezený kmitočtový rozsah i snížení pracovního kmitočtu komparátoru.



Obr. 6 Syntezátor s nepřímou koherentní syntézou – s řízeným předděličem

Programovatelný dělič kmitočtu je rozdělen na dvě části. První částí je *čítač swallow*, který je nastavován k bity řídicího slova s váhami $2^0, 2^1 \dots 2^{k-1}$. Velikost nastavení je vyjádřena číslem A. Druhou částí je *hlavní programovatelný čítač*, nastavovaný $n-k$ bity řídicího slova s váhami $2^k, 2^{k+1} \dots 2^{n-1}$. Velikost nastavení je vyjádřena číslem B. Řízený předdělič sníží kmitočet $(P+1)$ -krát a P -krát. Impulzy s kmitočtem $(P+1)$ -krát menším jsou přiváděny na vstup čítače swallow a impulzy s kmitočtem P -krát menším jsou přiváděny na vstup hlavního čítače. Impulzy v obou případech dekrementují stavy uvedených čítačů.

Jakmile jsou oba čítače nastaveny řídicím slovem, je řízený předdělič kmitočtu přepnut na dělicí poměr $P+1$. Po načítání A impulsů je čítač swallow vynulován a hlavní programovatelný čítač je nyní ve stavu B-A. řízený předdělič je poté přepnut na dělicí poměr P , který zůstává až do vynulování hlavního programovatelného čítače. Po vynulování je vyslán impuls ke Kmitočtově fázovému komparátoru, oba čítače se znovu nastaví podle řídicího slova, řízený předdělič se přepne na dělicí poměr $P+1$ a cyklus se opakuje. Pro vygenerování jednoho impulsu na výstupu dělicí soustavy je zapotřebí N vstupních impulsů:

$$N = (P + 1)A + P(B - A) = A + PB. \quad (14)$$

N udává dělicí poměr soustavy s řízeným předděličem. Výstupní kmitočet je potom:

$$f_{VCO} = \frac{N}{M} f_Q = \frac{A+PB}{M} f_Q. \quad (15)$$

Kmitočtový krok syntezátoru:

$$\Delta f_{VCO} = \frac{A+PB}{M} f_Q - \frac{A+PB-1}{M} f_Q = \frac{f_Q}{M}. \quad (16)$$

Nevýhoda syntezátoru s pevným předděličem je tedy odstraněna.

Dělicí poměr řízeného předděliče se většinou volí $P = 2^k$, kde k je počet bitů dolní části řídicího slova. V praxi bývají nejčastější dělicí poměry $P/(P+1) = 8/9, 32/33, 64/65, 128/129$ nebo $5/6, 10/11, 40/41$.

Maximální stav čítače swallow lze vyjádřit:

$$A_{MAX} = P - 1 = 2^k - 1 \rightarrow P = A_{MAX} + 1. \quad (17)$$

Na základě podmínky $B \geq A$, která musí platit, lze vyvodit rozsah hodnot čítače swallow:

$$A_{MAX} = B_{MIN} \quad \text{a} \quad A_{MIN} = 0. \quad (18)$$

Z uvedených vztahů lze odvodit vztah pro minimální dělicí poměr celé soustavy:

$$N_{MIN} = A_{MIN} + PB_{MIN} = 0 + (A_{MAX} + 1)A_{MAX} = P(P - 1) = 2^k(2^k - 1). \quad (19)$$

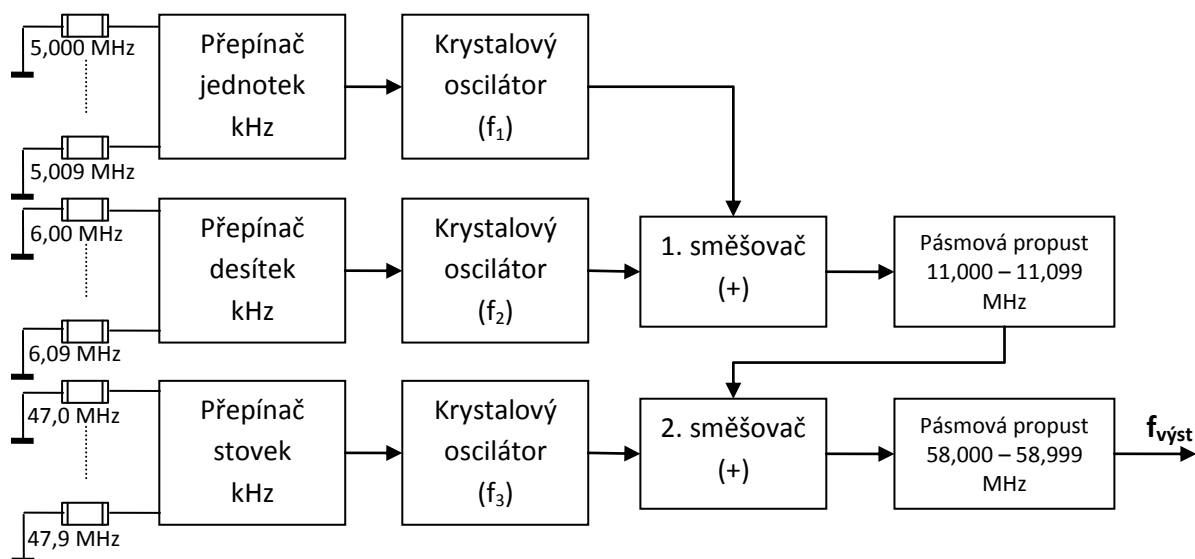
Maximální dělicí poměr je při n bitovém řídicím slově dán:

$$N_{MAX} = 2^n - 1. \quad (20)$$

U syntezátorů s řízeným předděličem musí dělicí poměry N nabývat pouze hodnot v uvedených mezích, což představuje omezení tohoto způsobu kmitočtové syntézy.

2.3 Syntezátory s přímou nekoherentní syntézou

Základ tvoří několik krystalem řízených oscilátorů, z jejichž signálů se vytváří potřebné výstupní signály. Díky tomu je stabilita a přesnost výstupního signálu závislá na příslušných parametrech všech oscilátorů. Zjednodušeně platí, že výstupní kvalitativní parametr syntezátoru je n -krát horší než odpovídající parametr jednoho z n použitých oscilátorů.



Obr. 7 Syntezátor s přímou nekoherentní syntézou (základem je 30 krystalových oscilátorů)

3 Návrh DDS syntezátoru - s přímou koherentní syntézou

Pro DDS (Direct Digital Synthesis) byl zvolen obvod firmy Analog Devices, protože tato firma nabízí zřejmě nejširší výběr obvodů pro syntezátory. Konkrétně byl zvolen obvod AD9958, který představuje špičku dvoukanalových vysokofrekvenčních syntéz ... $f_{osc} = 500 \text{ MHz}$.

3.1 AD9958

AD9958 se skládá ze dvou DDS zdrojů, které poskytují nezávislé řízení frekvence, fáze a amplitudy obou kanálů. Tato flexibilita může být použita ke korekci nevyváženosti mezi signály v důsledku analogového zpracování jako např. filtrace, zesílení nebo nepřizpůsobení na PCB. Protože oba kanály sdílí zdroj základního kmitočtu, jsou synchronizovány. AD9958 může provádět až 16ti-úrovňovou modulaci frekvence, fáze nebo amplitudy (FSK, PSK, ASK). Modulace je prováděna přivedením dat na příslušné piny. Dále AD9958 podporuje lineární rozmítání frekvence, fáze a amplitudy pro aplikace jako je například radar.

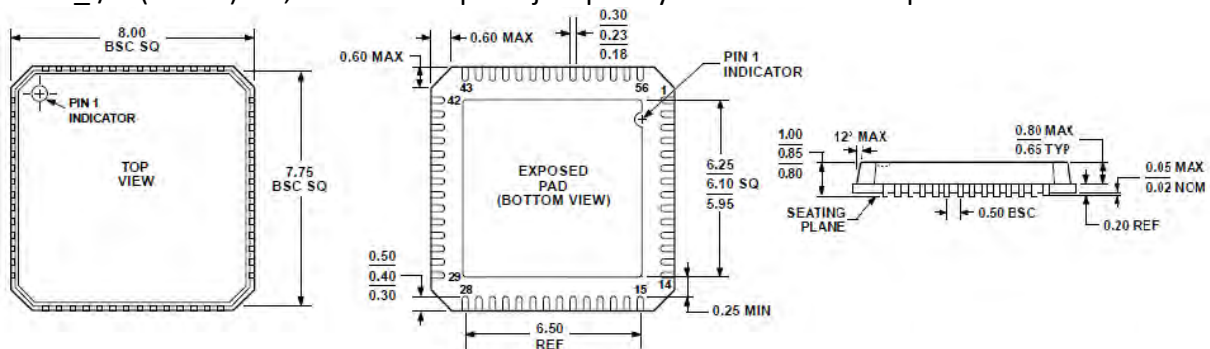
Sériový I/O port nabízí možnosti rozmanité konfigurace, která poskytuje vysokou flexibilitu. Sériový I/O port nabízí SPI-kompatibilní (*Serial Peripheral Interface*) mód, který je virtuálně identický s SPI obsluhou v dřívějších Analog Devices produktech pro DDS syntézu. Čtyři datové piny (SDIO_0/SDIO_1/SDIO_2/SDIO_3) nabízí možnost nastavení čtyř programovacích módů sériových I/O operací.

AD9958 používá pokročilou DDS technologii, která nabízí nízké energetické ztráty při vysokém výkonu. Obsahuje dva integrované 10ti-bitové DAC převodníky s velkou šířkou pásma. Každý kanál má vyhrazeno 32bitové slovo pro nastavení frekvence, 14 bitů pro nastavení offsetu fáze a 10 bitů pro nastavení výstupní napěťové úrovně.

Výstupy DAC musí být korektně zakončeny a to buď rezistorem, nebo oddělovacím transformátorem. Každý DAC má svůj vlastní programovatelný zdroj, což umožňuje rozdílné napájení pro oba kanály.

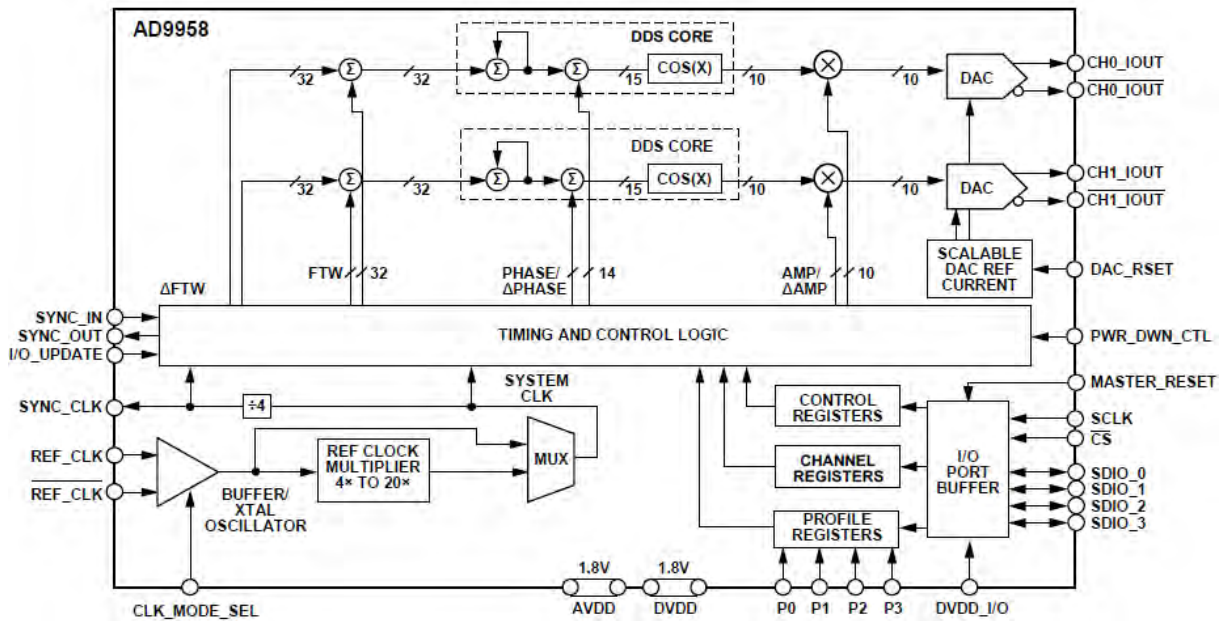
DDS funguje jako frekvenční dělič s vysokým rozlišením, kde REFCLK je vstupem a DAC poskytuje výstup. REFCLK vstup je stejný pro oba kanály a může být přiveden přímo nebo v kombinaci s integrovaným násobičem (PLL) pro získání maximální frekvence až 500 MHz. Násobič je programovatelný od 4 do 20 v krocích po 1. REFCLK vstup obsahuje též oscilátor, díky čemuž je možné připojit externí krystal jako zdroj kmitočtu. Krystal musí mít frekvenci mezi 20 MHz a 30 MHz a může být použit opět v kombinaci se vstupním násobičem.

Jako pouzdro AD9958 je použito 56pinové LFCSP – obr. 8. DDS jádro je napájeno 1,8 V zdrojem. Digitální I/O rozhraní (SPI) pracuje při 3,3 V a požaduje připojení pinu DVDD_I/O (Pin 49) k 3,3 V. AD9958 pracuje v průmyslovém rozsahu teplot $-40 \text{ }^\circ\text{C}$ až $+75 \text{ }^\circ\text{C}$.



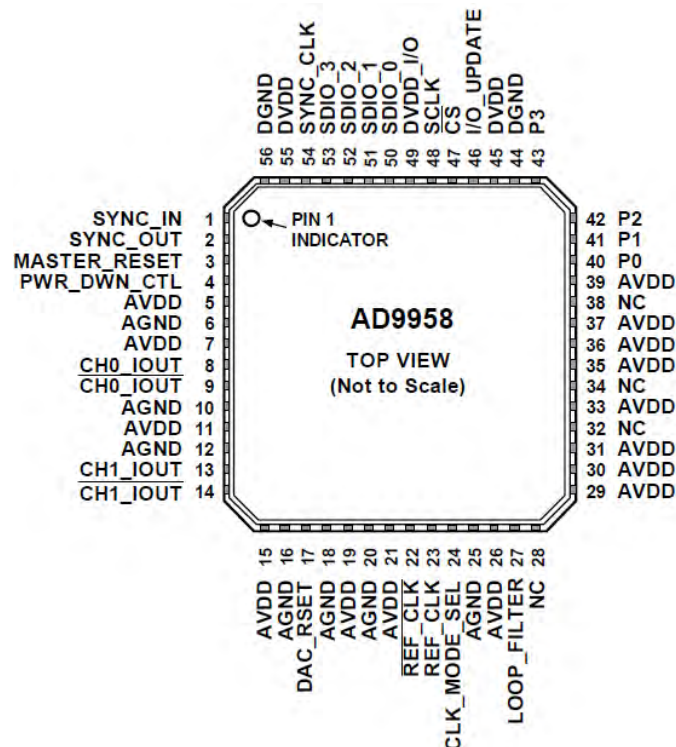
Obr. 8 AD9958 - Rozměry pouzdra LFCPS v mm [2]

Na obr. 9 je možno vidět kompletní vnitřní blokovou strukturu obvodu AD9958 převzatou z [2].



Obr. 9 Vnitřní bloková struktura AD9958 [2]

Další obrázek – obr. 10 ukazuje konfiguraci pinů doplněnu v následující tabulce – tab. 1 o jejich popis včetně popisu jednotlivých funkcí.



Obr. 10 Popis pinů pouzdra obvodu AD9958 [2]

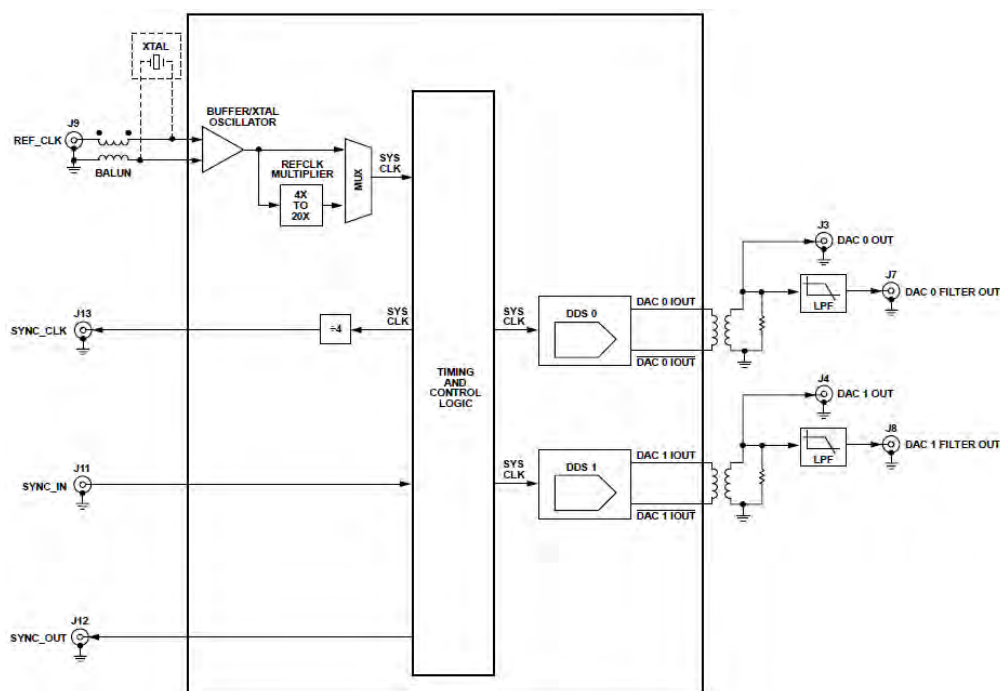
Poznámky: Odkrytý PAD na spodní straně pouzdra musí být uzemněn.
 PIN 49 je DVDD_I/O a je připojen k napětí 3,3 V.
 NC = No Connect

Tab. 1 Popis funkce pinů

Pin č.	Označení	I/O	Popis
1	SYNC_IN	I	Pro synchronizaci více AD9958. Připojen k SYNC_OUT řídicího AD9958.
2	SYNC_OUT	O	Pro synchronizaci více AD9958. Připojen k SYNC_IN pomocného AD9958.
3	MASTER_RESET	I	Aktivní reset. Resetuje interní registry do jejich defaultního stavu, jak je popsáno v mapách registrů.
4	PER_DWN_CTL	I	Externí Power-Down řízení.
5, 7, 11, 15, 19, 21, 26, 29, 30, 31, 33, 35, 36, 37, 39	AVDD	I	Analogové napájení (1,8 V).
6, 10, 12, 16, 18, 20, 25	AGND	I	Analogová zem.
45, 55	DVDD	I	Digitální napájení (1,8 V).
44, 56	DGND	I	Digitální zem.
8	CH0_IOUT	O	Pravý DAC výstup. Zakončen v AVDD.
9	$\overline{\text{CH0_IOUT}}$	O	Komplementární DAC výstup. Zakončen v AVDD.
13	CH1_IOUT	O	Pravý DAC výstup. Zakončen v AVDD.
14	$\overline{\text{CH1_IOUT}}$	O	Komplementární DAC výstup. Zakončen v AVDD.
17	DAC_RSET	I	Stanovuje referenční proud pro oba DAC. 1,91 k Ω rezistor je připojen z pinu 17 na AGND.
22	REF_CLK	I	Komplementární referenční frekvenční vstup. Pokud REF_CLK pracuje v jednoduchém módu, tento pin by měl být připojen k AVDD nebo AGND přes 0,1 μF kondenzátor.
23	REF_CLK	I	Referenční frekvenční vstup. Pokud REF_CLK pracuje v jednoduchém módu, je použit jako vstup.
24	CLK_MODE_SEL	I	Řídicí pin oscilátoru. Pozor: Nepřekračujte napětí 1,8 V. Pokud je přivedeno 1,8 V, je akceptován jako REF_CLK zdroj krystal – oscilátor je aktivní. Pokud je nastaveno napětí nízké, je oscilátor vyřazen.
27	LOOP_FILTER	I	Připojen k externí nulové kompenzační síti PLL filtru. Běžně se síť skládá z 0 Ω rezistoru s 680 pF kondenzátorem připojeným k AVDD.
28, 32, 34, 38	NC	N/A	No Connection.
40, 41, 42, 43	P0, P1, P2, P3	I	Piny pro modulaci (FSK, PSK, ASK), pro start/stop rozmítacích akumulátorů nebo pro zvyšování/snižování výstupní úrovně napětí. Data jsou synchronizována s SYNC_CLK (Pin 54). Vstupní data musí souhlasit s nastavením a aktivní dobou SYNC_CLK. Funkčnost je kontrolována pomocí PPC (Profile Pin Configuration) bitů.

Pin č.	Označení	I/O	Popis
46	I/O_UPDATE	I	Náběžná hrana převádí data z bufferu sériového I/O portu do aktivních registrů. Je synchronní s SYNC_CLK (Pin 54). Musí souhlasit s nastavením a aktivní dobou SYNC_CLK pro garantování fixního zpoždění dat do DAC výstupu, jinak vzniká nejistota ± 1 SYNC_CLK periody. Minimální šířka impulzu je jedna SYNC_CLK perioda.
47	\overline{CS}	I	Aktivní výběr čipu. Umožňuje čteným zařízením sdílet běžnou I/O sběrnici (SPI).
48	SCLK	I	Kmitočet sériového spojení pro I/O operace. Data jsou zapsána při náběžné hraně SCLK a čtena při sestupné hraně.
49	DVDD I/O	I	3,3 V napájení pro SPI Port a Digitální I/O.
50	SDIO_0	I/O	Datový pin SDIO_0 je vyhrazen pro sériový port.
51, 52, 53	SDIO_1, SDIO_2, SDIO_3	I/O	Datové piny SDIO_1 až SDIO_3 mohou být použity pro sériový port nebo pro nastavení výstupní napěťové úrovně DAC.
54	SYNC_CLK	O	SYNC_CLK běží na $\frac{1}{4}$ systémové frekvence. Může být zakázán. I/O_UPDATE a Piny 40 až 43 jsou synchronní s SYNC_CLK. Pro garantování fixního zpoždění dat do DAC výstupu, I/O_UPDATE nebo data (Piny 40 až 43) musí souhlasit s nastavením a aktivní dobou vzhledem k náběžné hraně SYNC_CLK, jinak vzniká nejistota ± 1 SYNC_CLK.

Na následujícím obr. 11 je již vidět blokový návrh DDS syntezátoru doplněný o další potřebné prvky na jeho vstupu a výstupech. Těmito prvky jsou především oddělovací transformátory a výstupní filtry.

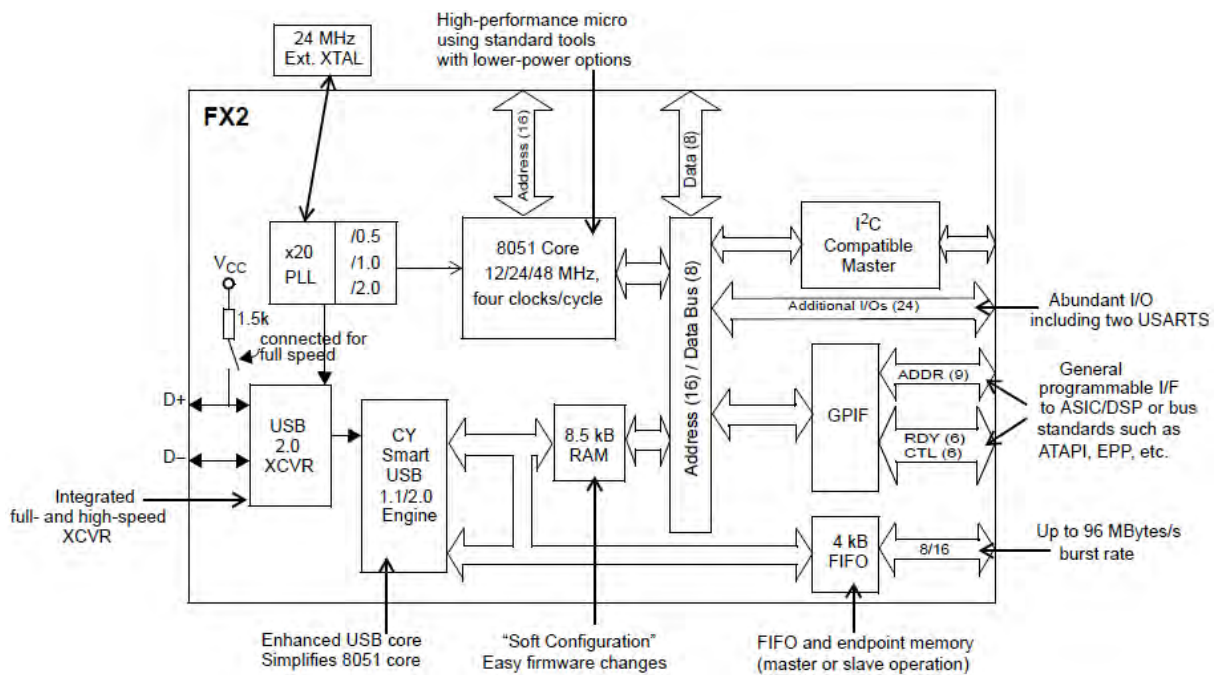


Obr. 11 Blokové schéma DDS syntezátoru

3.2 USB řadič EZ-USB FX2 – CY7C68013

EZ-USB FX2 je USB 2.0 integrovaný mikrokontroler. Integruje v sobě USB 2.0 řadič, SIE, 8051 mikrokontroler a programovatelné periferie v jednom čipu. Architektura FX2 umožňuje přenos dat rychlostí až 56 MBps, tedy maximum rychlosti USB 2.0 a to při zachování malých vnějších rozměrů umožňujících integrování do pouzdra 56 SSOP. Většina USB 1.1 a 2.0 protokolů je přímo integrována v hardwaru, což dává větší prostor pro tvorbu specifických aplikačních funkcí namísto řešení problémů s kompatibilitou USB. GPIF (General Programmable Interface) a koncový Master/Slave FIFO (8- nebo 16-bit) poskytuje snadný přechod na běžná rozhraní jako ATA, UTOPIA, EPP, PCMCIA a mnoho DSP/procesorů.

Na obr. 12 je vidět vnitřní bloková struktura USB-FX2 převzata z [3]. Kromě zmíněného pouzdra 56 SSOP, se vyrábí ještě v dalších třech provedení a to – 56 QFN, 100 TQFP, 128 TQFP.



Obr. 12 Vnitřní bloková struktura USB-FX2 (CY7C68013) [3]

Další obrázek – obr. 13 ukazuje konfiguraci pinů výše zmíněného pouzdra 56 SSOP.

1	PD5/FD13	PD4/FD12	56
2	PD6/FD14	PD3/FD11	55
3	PD7/FD15	PD2/FD10	54
4	GND	PD1/FD9	53
5	CLKOUT	PD0/FD8	52
6	VCC	*WAKEUP	51
7	GND	VCC	50
8	*RDY0*/SLRD	RESET#	49
9	RDY1*/SLWR	GND	48
10	AVCC	PA7*/FLAGD/SLCS#	47
11	XTALOUT	PA6/PKTEND	46
12	XTALIN	PA5/FIFOADR1	45
13	AGND	PA4/FIFOADR0	44
14	VCC	PA3*/WU2	43
15	DPLUS	PA2*/SLOE	42
16	DMINUS	PA1/INT1#	41
17	GND	PA0/INT0#	40
18	VCC	VCC	39
19	GND	CTL2*/FLAGC	38
20	IFCLK	CTL1*/FLAGB	37
21	RESERVED	CTL0*/FLAGA	36
22	SCL	GND	35
23	SDA	VCC	34
24	VCC	GND	33
25	PB0/FD0	PB7/FD7	32
26	PB1/FD1	PB6/FD6	31
27	PB2/FD2	PB5/FD5	30
28	PB3/FD3	PB4/FD4	29

Obr. 13 Popis pinů pouzdra 56 SSOP obvodu USB-FX2 (CY7C68013) [3]

3.3 Návrh zdroje referenčního hodinového signálu

Volba zdroje referenčního hodinového signálu je velmi důležitá pro výsledné kvalitativní parametry výstupního signálu syntezátoru. Jsou k dispozici dvě možnosti řešení a to připojení krystalového oscilátoru a využití vnitřní PLL smyčky pro násobení jeho kmitočtu v rozsahu 4x – 20x nebo připojení externího zdroje hodinového signálu s vysokým kmitočtem. Řešení s připojením externího zdroje hodinového signálu – generátoru (až do frekvence 500 MHz) je sice co do kvality lepší, ale komplikované a ztrácí se zvláště v tomto případě smysl práce – vytvořit samostatný generátor kmitočtu.

Proto byl zvolen jako zdroj hodinového signálu krystalový oscilátor s frekvencí 25 MHz (povolený rozsah je 20 – 30 MHz) v pouzdře HC49U-H. Obvodové řešení je velmi snadné – krystalový oscilátor je připojen na piny REF_CLK a REF_CLKB a k oběma pinům jsou zároveň připojeny zatěžovací kapacity 39 pF. Frekvence krystalového oscilátoru bude násobena s pomocí vnitřní PLL smyčky v rozsahu 4x – 20x a to na základě nastavení v registru *Function Register 1*, bity [22:18]. Pokud by došlo k nastavení mimo rozsah povolených hodnot, bude automaticky násobeno 1x. Nevýhodou použití krystalového oscilátoru je, že všechny parazitní vlastnosti oscilátoru budou násobeny spolu s frekvencí a tedy až 20x zvětšeny. Je třeba si dát pozor i na teplotní závislost kmitočtu a při konstrukci zohlednit umístění krystalu mimo součástky, u nichž lze očekávat zvýšený ohřev.

Volba zdrojů referenčního hodinového kmitočtu se provádí přes pin CLK_MODE_SEL a to podle tab. 2.

Tab. 2 Konfigurace referenčního hodinového signálu [2]

CLK_MODE_SEL	PLL násobič (M)	Krystal Osc.	Systémová f	Min/Max f [MHz]
High = 1,8 V	$4 \leq M \leq 20$	Aktivní	$f_{SYS} = f_{OSC} \times M$	$100 < f_{SYS} < 500$
High = 1,8 V	$M < 4$ nebo $M > 20$	Aktivní	$f_{SYS} = f_{OSC}$	$20 < f_{SYS} < 30$
Low	$4 \leq M \leq 20$	Neaktivní	$f_{SYS} = f_{REF} \times M$	$100 < f_{SYS} < 500$
Low	$M < 4$ nebo $M > 20$	Neaktivní	$f_{SYS} = f_{REF}$	$0 < f_{SYS} < 500$

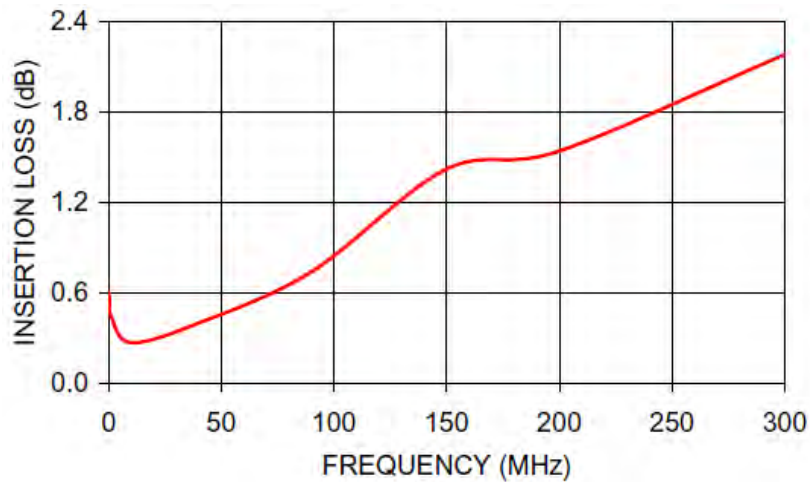
3.4 Návrh výstupních obvodů

Jsou k dispozici 2 výstupy DAC převodníků. Proudové výstupy převodníků mohou být modelovány jako zdroje proudu s vysokou výstupní impedancí (~100 kΩ). Oba převodníky mají komplementární výstupy, které tak poskytují plný rozsah výstupního proudu ($I_{OUT} + I_{\overline{OUT}}$). Proudový rozsah je nastavován externím rezistorem (R_{SET}) a skokově v jednom z registrů. Rezistor R_{SET} je zapojen mezi pin DAC_RSET a zem. Velikost proudu je nepřímo úměrná velikosti R_{SET} podle vztahu:

$$R_{SET} = \frac{18,91}{I_{OUT(max)}}. \quad (21)$$

Maximální výstupní proud je omezen na 15 mA, ale je vhodné nepřekročit 10 mA pro získání optimální hodnoty SFDR (*Spurious-Free Dynamic Range*). Výstupní napětí je mezi $AVDD + 0,5$ V a $AVDD - 0,5$ V ($AVDD = 1,8$ V). Pokud by se napětí dostalo mimo tyto meze, může dojít k harmonickému zkreslení výstupního signálu. Je proto nutné dbát zvýšené pozornosti impedančnímu zatížení.

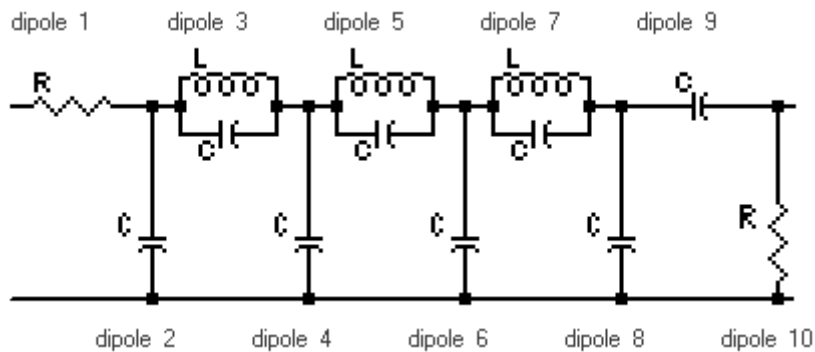
Výstupní obvod je řešen na základě doporučení [2] transformátorem 1:1. Konkrétně byly použity transformátory ADTT1-1 od Mini-Circuits s frekvenčním rozsahem 0,3–300 MHz a vložitelným útlumem 3 dB (průběh vložitelného útlumu je na obr. 14). Použití transformátoru zaručuje zároveň galvanické oddělení výstupu a tak ochranu syntezátoru.



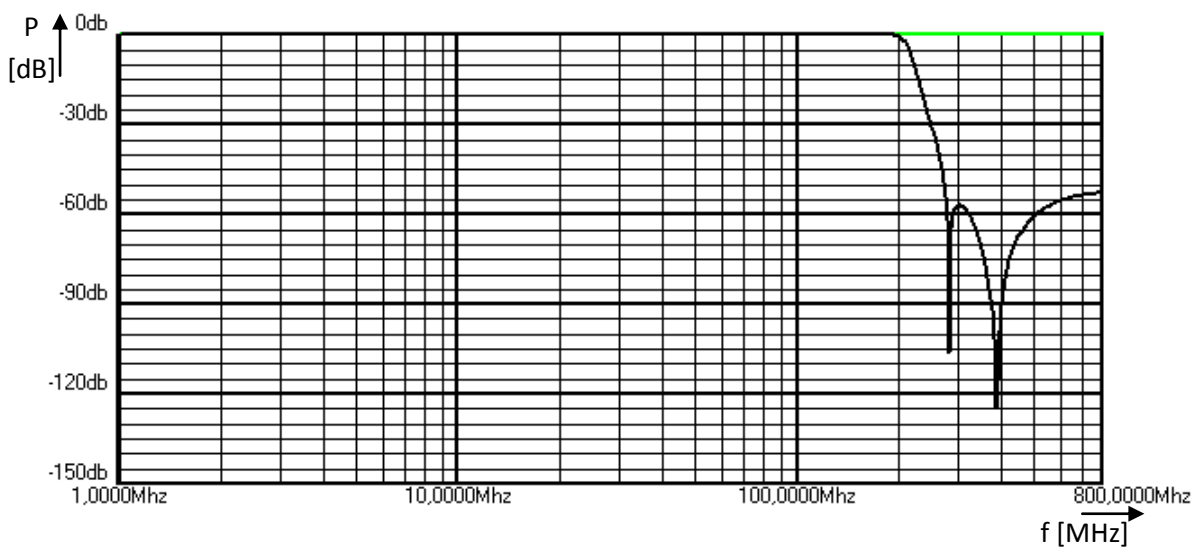
Obr. 14 Závislost vložného útlumu ADTT1-1 na frekvenci [5]

Za transformátory jsou vyvedeny zatím nefiltrované výstupy DAC0_OUT a DAC1_OUT. Následují dva shodné rekonstrukční filtry. Jedná se o eliptické dolní propusti sedmého řádu s lomovým kmitočtem 200 MHz a přizpůsobenou vstupní i výstupní impedancí na 50 Ω.

Filtr byl navržen v programu AADE Filter Design V4.5. Názorné schéma filtru je na obr. 15. Výsledek simulace závislosti přenosu na frekvenci je na obr. 16.



Obr. 15 Schéma výstupního filtru v programu AADE Filter Design V4.5

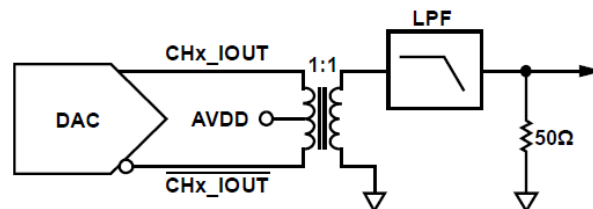


Obr. 16 Simulace závislosti přenosu na frekvenci v programu AADE Filter Design V4.5

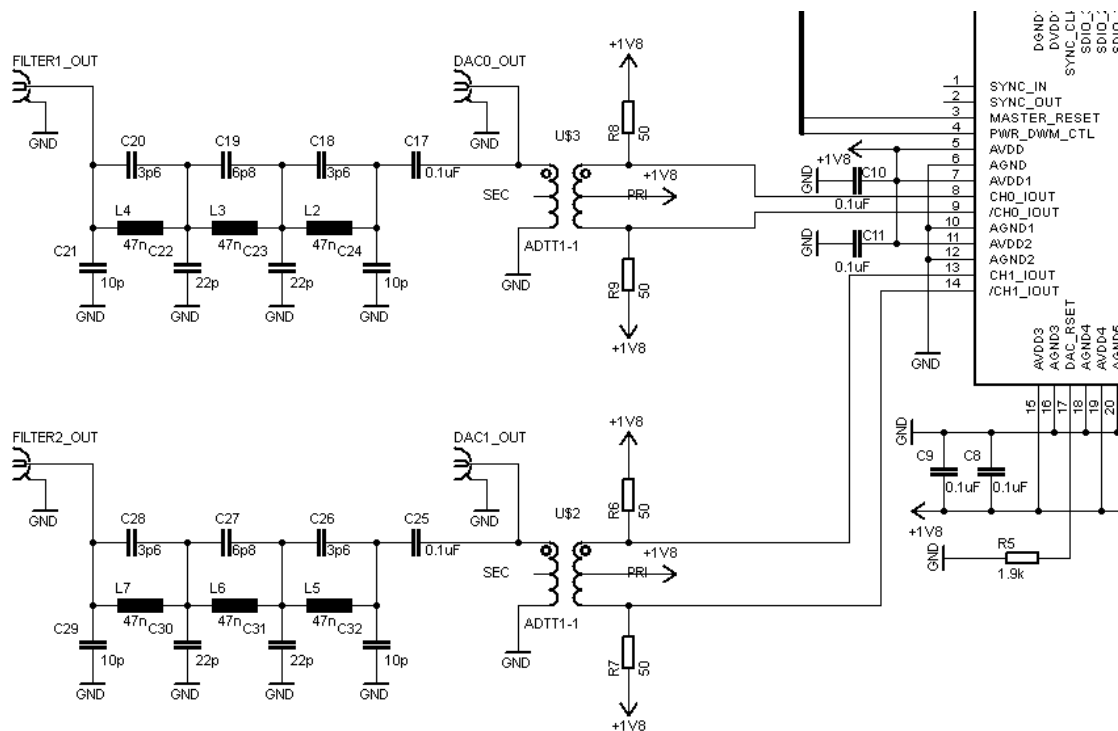
Za oběma filtry jsou vyvedeny výstupy FILTER1_OUT, FILTER2_OUT. Všechny výstupy jsou řešeny pomocí SMA Female konektorů se čtvercovou základnou, otočené o 90°.

Poslední významnou částí je filtr smyčky PLL násobičky integrované v obvodu. Zde jsou k pinu LOOP_FILTER na základě doporučení [2] připojeny sériově rezistor R1 = 690 Ω a kondenzátor C1 = 680 pF.

Na obr. 18 je možno vidět schéma výsledného zapojení výstupních obvodů navržené v programu EAGLE 5.7.0. Pro srovnání je na předchozím obr. 17 znázorněno blokové schéma doporučeného zapojení výstupů DAC podle Analog Devices.



Obr. 17 Doporučené zapojení výstupních obvodů podle [2]



Obr. 18 Navržené zapojení výstupních obvodů v programu EAGLE 5.7.0

3.5 Návrh komunikačního rozhraní

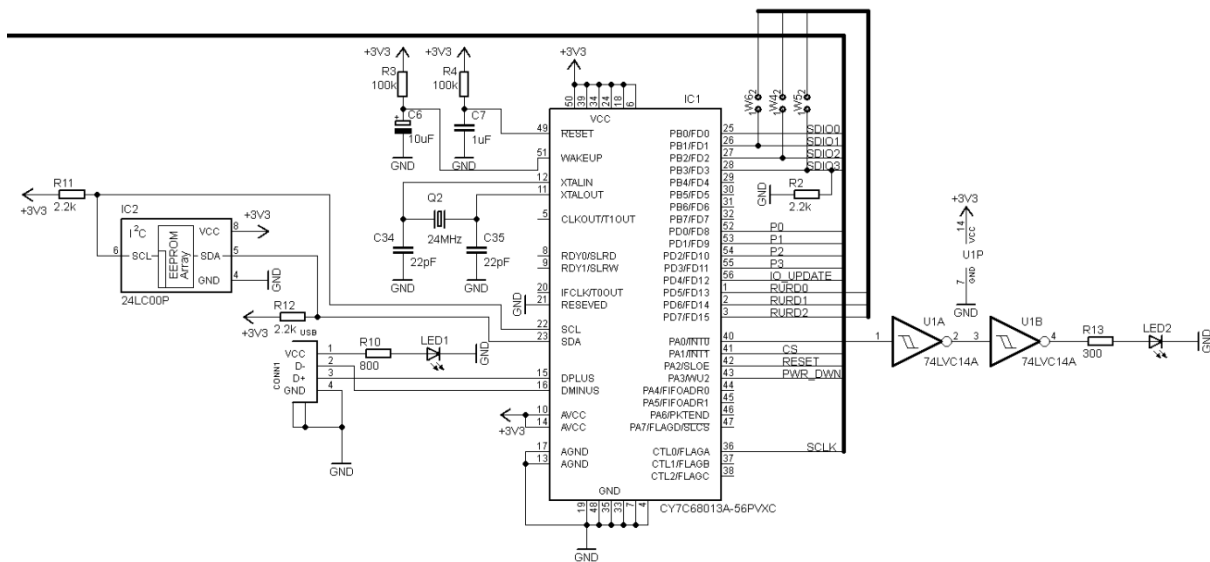
Pro návrh je využit již popsáný USB2.0 mikrokontroler CYPRESS CY7C68013. Pro univerzálnost a případné řešení možných problémů je vyvedeno i sériové rozhraní, které umožňuje komunikaci pomocí RS232.

Návrh zapojení mikrokontroleru CY7C68013 vychází ze zapojení *AD9958 Evaluation Board* [6]. K samotnému mikrokontroleru je zapotřebí připojit kromě USB portu 128-bitovou sériovou EEPROM řešenou v našem případě integrovaným obvodem 24LC00 od Microchip. EEPROM je připojena k I2C sběrnici a slouží případně k uložení základních identifikačních hodnot, které se načítají při spuštění. Dále je k pinům XTALIN a XTALOUT připojen 24 MHz

krystal spolu se dvěma 22 pF zatěžovacími kapacitami. Jeho frekvence je vnitřní PLL násobičkou převedena podle potřeb až na 480 MHz.

V zapojení jsou použity 2 LED diody sloužící k identifikaci stavu USB komunikace. LED1 je připojena přímo k napájení z USB a indikuje tedy připojení k aktivnímu zařízení. Napájení z USB není dále využíváno, pro celý obvod je řešeno samostatně z důvodu eliminace možných rušení (viz. kapitola 3.6 *Návrh napájení*). LED2 je připojena k pinu PA0/INT0# a signalizuje korektní připojení obvodu k PC. LED2 je připojena přes dvě invertující Schmittova hradla obvodu 74LVC14A, která z pomalu se měnícího signálu vytvoří ostře definovaný obdélníkový signál vhodný pro zobrazení signalizace.

Za povšimnutí stojí ještě jumpery W4, W5, W6, které po připojení umožňují řízení Ramp Up/Ramp Down funkce pro řízení výstupní amplitudy pomocí SDIO_1 až SDIO_3. Obr. 19 zobrazuje kompletní zapojení mikrokontroleru CY7C68013 se všemi výše zmíněnými obvody. Komunikace s AD9958 je řešena pro přehlednost sběrnic zahrnující signály z AD9958 ... SDIO_0 ÷ SDIO_3, P0 ÷ P3, IO_UPDATE, \overline{CS} , RESET, PWR_DWN, SCLK. Význam jednotlivých signálů je popsán v tab. 1.



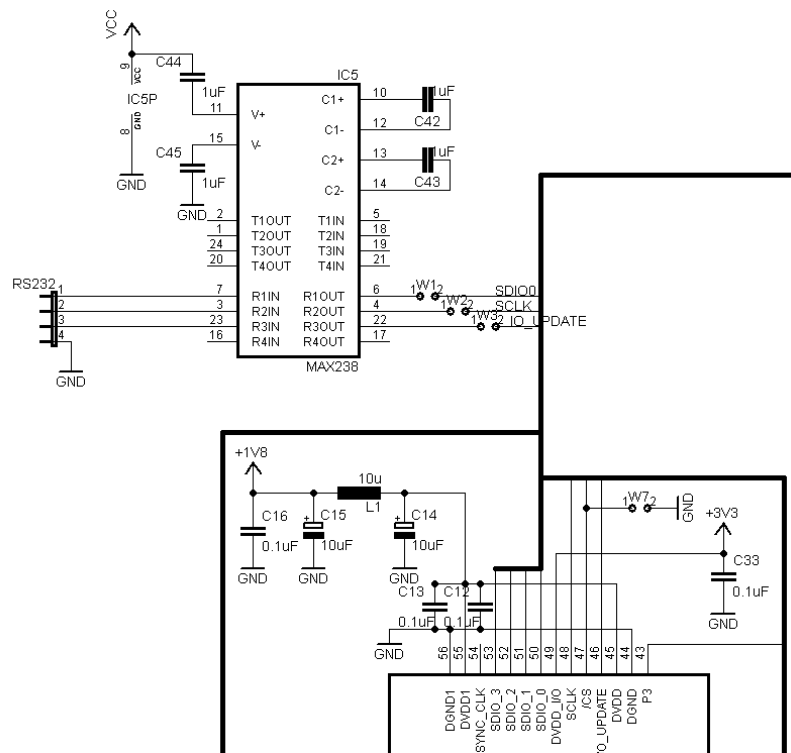
Obr. 19 Zapojení mikrokontroleru CY7C68013 navržené v EAGLE 5.7.0

Pro možnost sériové komunikace přes port RS232 jsou vyvedeny signály SDIO_0, SCLK a IO_UPDATE přes převodník napětových úrovní MAX238. V případě potřeby stačí zapojit jumpery W1, W2, W3 a W7. Jumper W7 je připojen k signálu \overline{CS} sloužícímu k výběru obvodu v případě sdílení datové sběrnice. Připojením k log. 0 je obvod trvale vybrán. Obr. 20 zobrazuje kompletní zapojení obvodu MAX238.

Je využíván základní *jednobitový sériový 2vodičový* mód komunikace, při němž je využíván pouze pin SDIO_0. V tab. 3 jsou pro přehlednost vypsány všechny dostupné komunikační režimy včetně způsobu jejich výběru v *channel select register* CSR[2:1]. Maximální frekvence SCLK je 200 MHz a při využití pinů SDIO_0 ÷ SDIO_3 lze dosáhnout maximální propustnosti 800 Mbps.

Jednobitový sériový 2vodičový mód umožňuje čtení/zápis všech registrů AD9958. Jsou podporovány oba přenosové formáty – od MSB či LSB. Jednobitový sériový mód může mít

interface nakonfigurován buď na jednom I/O pinu, což umožňuje 2vodičové rozhraní nebo na dvou pinech pro vstup (SDIO_0)/výstup (SDIO_2), což umožňuje 3vodičové rozhraní. Pin SDIO_3 funguje jako SYNC_I/O.



Obr. 20 Zapojení obvodu MAX238 navržené v EAGLE 5.7.0

Tab. 3 Podporované komunikační režimy a způsob jejich výběru [2]

Mód komunikace	Pin						Výběr CSR[2:1]
	SCLK	\overline{CS}	SDIO_0	SDIO_1	SDIO_2	SDIO_3	
Jednabitový sériový 2vodičový	Sériový takt	Výběr čipu	Data I/O	Nepoužit	Nepoužit	SYNC I/O	00
Jednabitový sériový 3vodičový			Data Input	Nepoužit	Data Output	SYNC I/O	01
Dvoubitový sériový			Data I/O	Data I/O	Nepoužit	SYNC I/O	10
Čtyřbitový sériový			Data I/O	Data I/O	Data I/O	Data I/O	11

Pro popis komunikace je dále nezbytná podoba instrukčního slova. To je popsáno následující tabulkou 4.

Tab. 4 Podoba instrukčního slova [2]

D7	D6	D5	D4	D3	D2	D1	D0
R/ \overline{W}	x	x	A4	A3	A2	A1	A0

Bit D7 instrukčního slova (R/\bar{W}) rozhoduje, zda-li po příchodu slova dochází ke čtení nebo zápisu. Log. 1 znamená operaci čtení, log. 0 naopak zápis. Bity D4 až D0 adresují příslušný registr.

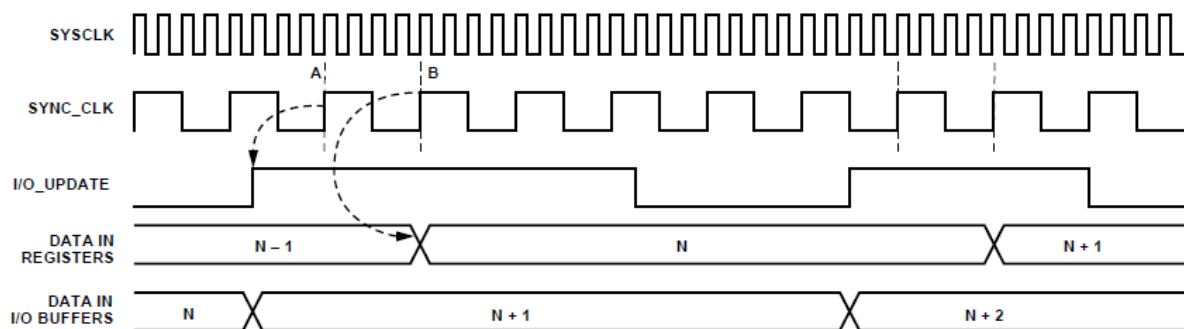
Jako příklad uveďme instrukční slovo potřebné pro zápis do *Function Register 1* (FR1, Register 0x01) ve formátu od MSB. Podoba instrukčního slova je 0b00000001. Interní kontroler rozpozná operaci zápisu (podle D7 = 0) tří bytů začínajících od MSB (FR1[23]). Byty jsou zapsány při každé následující náběžné hraně SCLK, dokud není přenesen bit 0. Jakmile je zapsán poslední datový bit, komunikační cyklus je kompletní a další byte je považován jako instrukční. Z příkladu vyplývá, že programování probíhá po kvantech podle velikosti registru, k němuž je přistupováno.

Pro zápis do FR1 ve formátu od LSB, posíláme instrukční slovo 0b00000001 od LSB (D0). Po této instrukci rozpozná interní kontroler operaci zápisu tří bytů, začínajících od LSB (FR1[0]). Byty jsou zapsány při každé následující náběžné hraně SCLK, dokud není přenesen bit 23. Poté je opět další byt považován jako instrukční.

Samotná volba formátu dat – od MSB či od LSB je nastavována v *Channel Select Register* CSR[0]. Standardně je nastaven formát od MSB. LSB formát se nastavuje log. 1 v CSR[0].

Jak je zapsáno v tab. 1, signál IO_UPDATE převádí data ze sériového I/O bufferu do aktivních registrů. Zde je dobré poznamenat, že tento převod může být proveden buď po naprogramování každého registru zvlášť, nebo po naprogramování všech požadovaných registrů. Druhé jmenované řešení je výhodnější z důvodu menší komunikační režie. Výjimkou jsou *channel enable bits* v *Channel Select Register* (CSR) – tyto bity nepotřebují IO_UPDATE pro povolení.

Obr. 21 znázorňuje časový průběh signálů při přenosu dat z I/O bufferu do aktivních registrů AD9958.



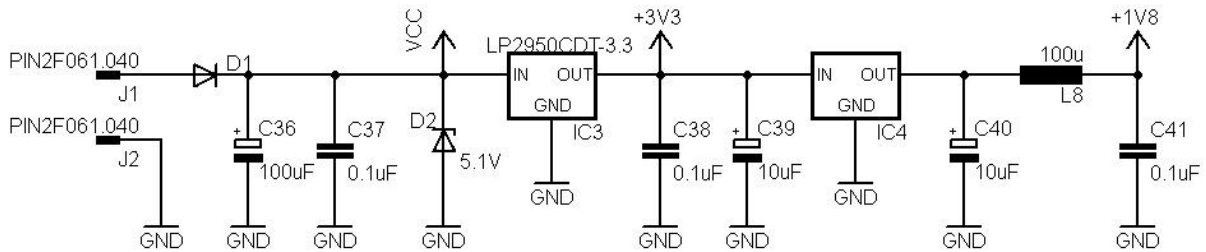
Obr. 21 Signál IO_UPDATE převádí data z I/O bufferu do aktivních registrů AD9958 [2]

3.6 Návrh napájení

K napájení jednotky DDS AD9958 jsou zapotřebí dvě napětí a to 1,8 V pro DDS jádro a 3,3 V pro digitální I/O rozhraní. Napájecí napětí pro USB2.0 mikrokontroler CYPRESS CY7C68013 je 3,3 V stejně jako pro připojenou EEPROM paměť a obvod Schmittových hradel 74LVC14A. Obvod sériové komunikace MAX238 je napájen napětím až 5 V.

Jak bylo řečeno v předchozí kapitole, napájení přímo z USB není využito, ale je řešeno samostatně z důvodu omezení možných rušení. Obr. 22 zobrazuje návrh napájecích obvodů. Uvažuje se připojení napětí 5 V, kterým je přímo napájen obvod MAX238. Napětí je dále stabilizátorem LP2950 převedeno na hodnotu 3,3 V, která je využita výše zmíněnými obvody. Nakonec je napětí převedeno stabilizátorem LM1117 na hodnotu 1,8V pro napájení jádra

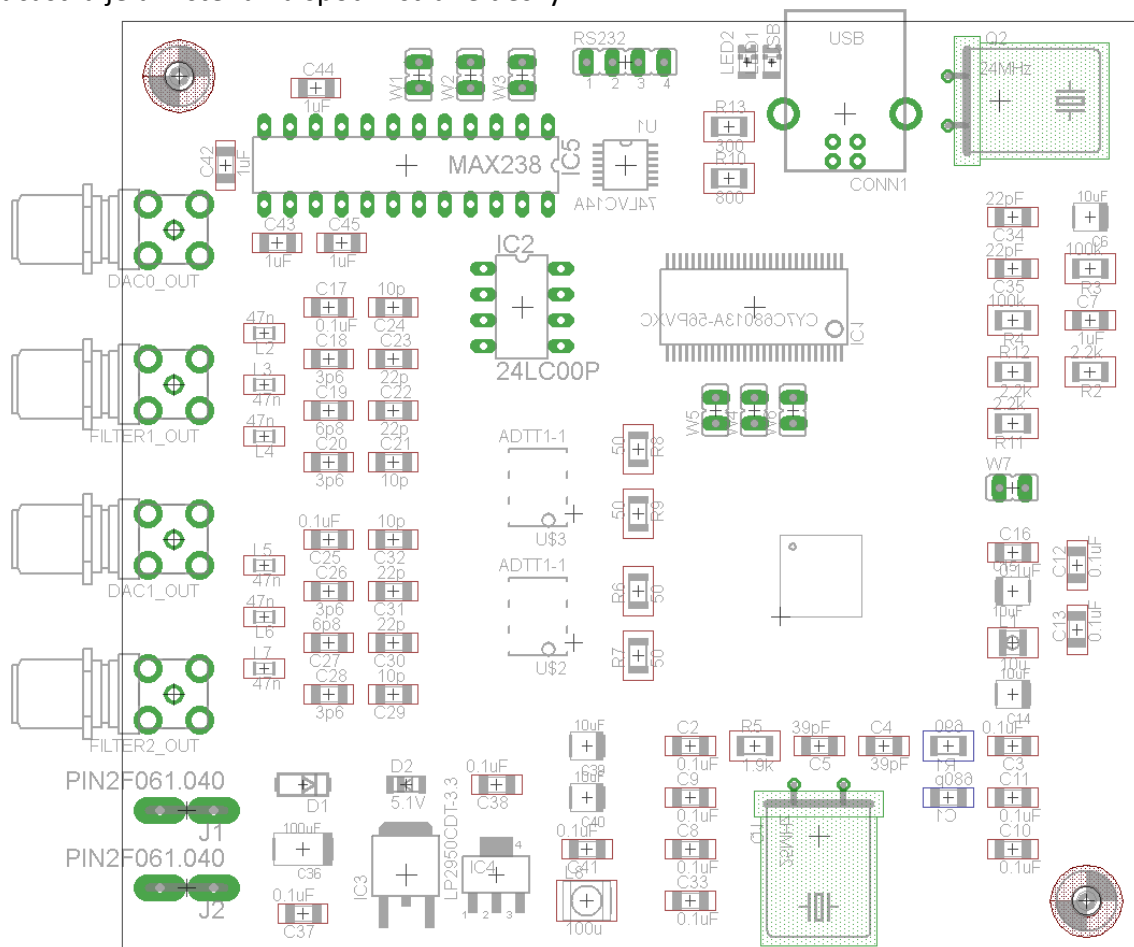
DDS jednotky. Na vstupu napájecího obvodu jsou připojeny dvě diody, které plní funkce ochrany a to ochranu proti případnému přepólování – dioda D1 (typ SS16) zapojena v sérii v kladné větvi, a ochranu proti přivedení vyššího napětí – dioda D2 (Zenerova dioda 5,1 V) zapojena mezi kladnou větví a zemí. Vyhlazovací kondenzátory jsou umístěny jak u napájecích stabilizátorů, tak i přímo u jednotlivých pinů použitých obvodů.



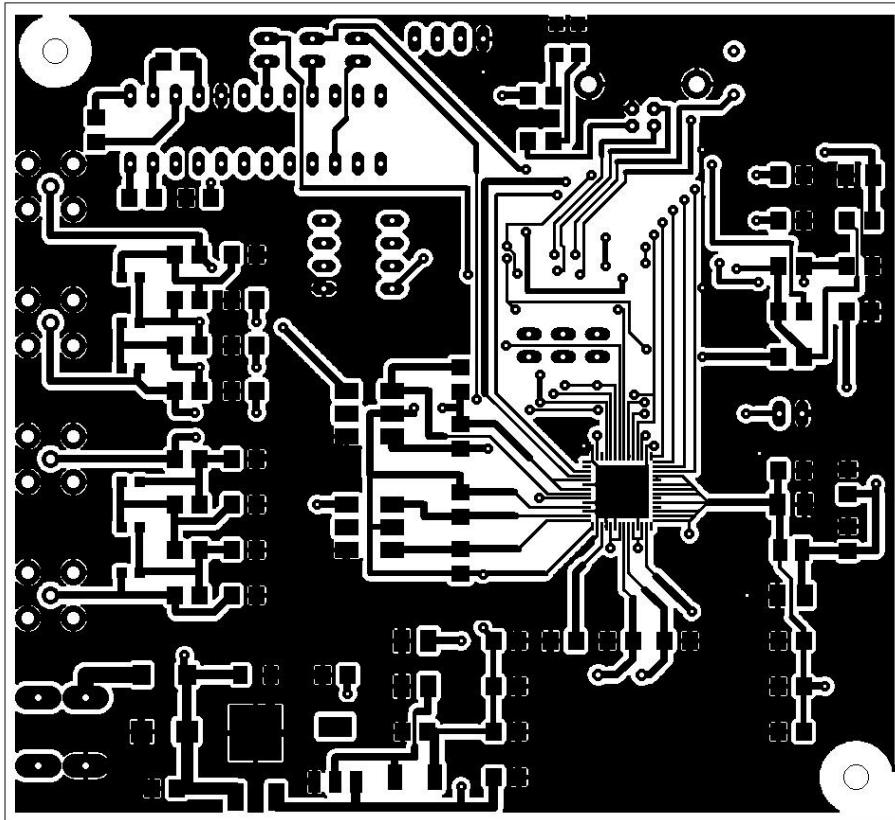
Obr. 22 Zapojení napájení navržené v EAGLE 5.7.0

3.7 Deska plošného spoje

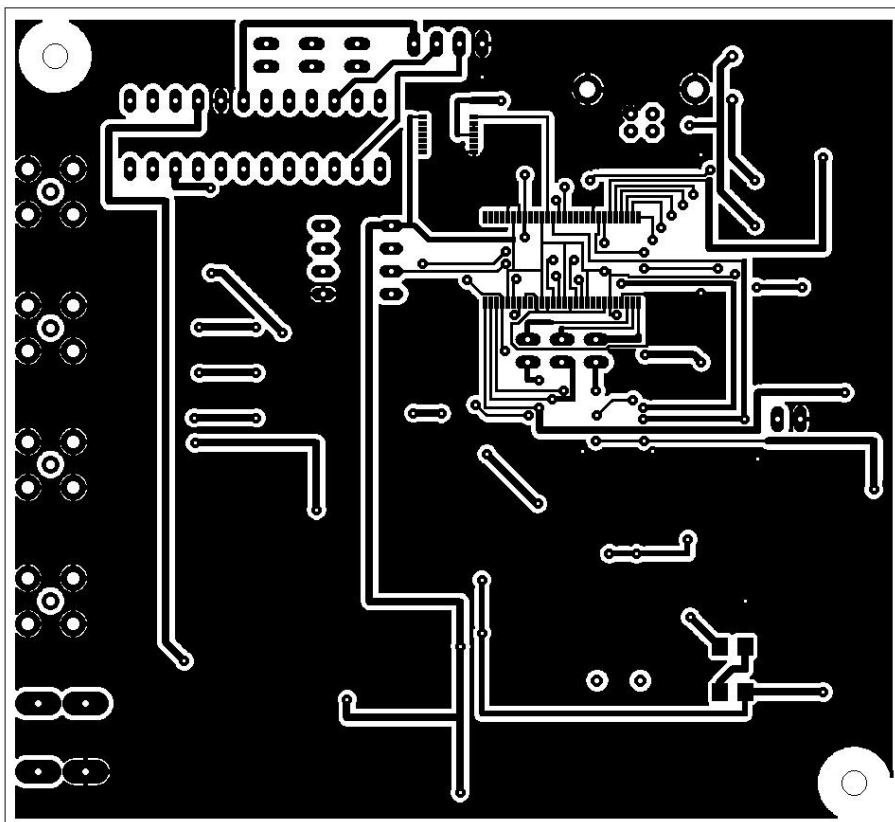
Obr. 24 a 25 zobrazují vrstvu TOP a BOTTOM desky plošného spoje syntezátoru. Pro výrobu desky je použit materiál FR4. Kondenzátory a rezistory jsou použity SMD velikosti 1206. Tlumivky ve výstupních filtrech jsou použity ve velikosti SMD 0805, tlumivka L1 má velikost SMD 1206 a tlumivka L8 velikost SMD 1812. USB konektor je typu B. Následující obr. 23 zobrazuje rozmístění součástek na desce. Zrcadlově napsaný název značí, že součástka je umístěna na spodní straně desky.



Obr. 23 Rozmístění součástek DDS syntezátoru na DPS (EAGLE 5.7.0)

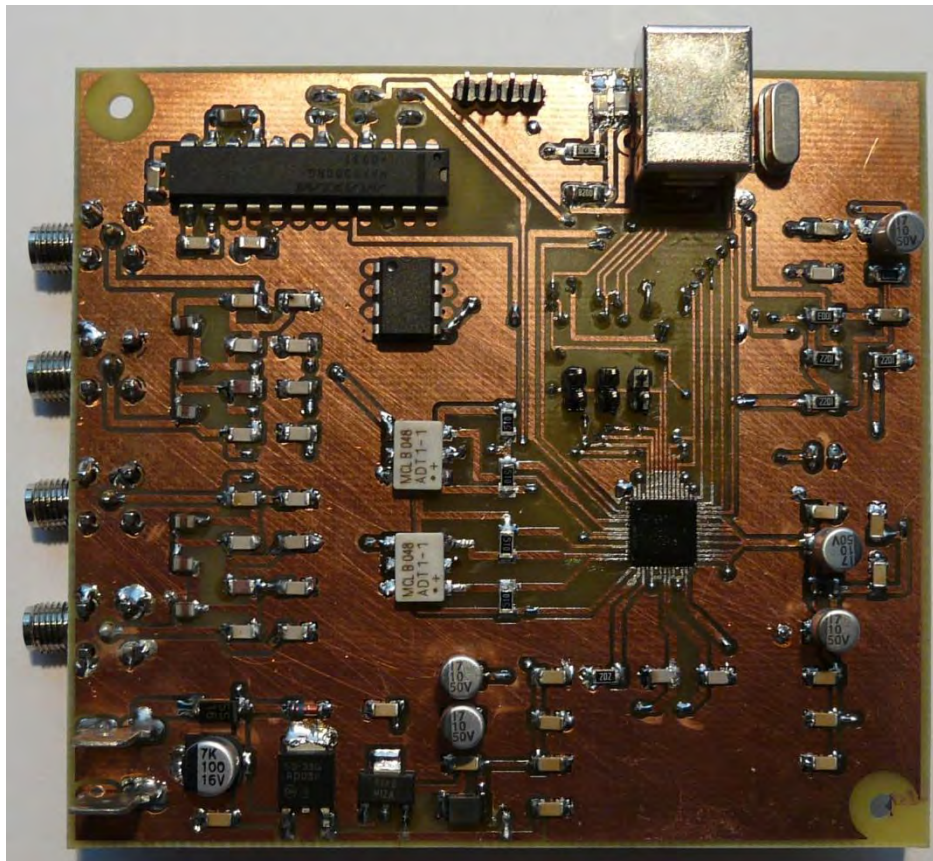


Obr. 24 Návrh DPS – vrstva TOP (EAGLE 5.7.0)

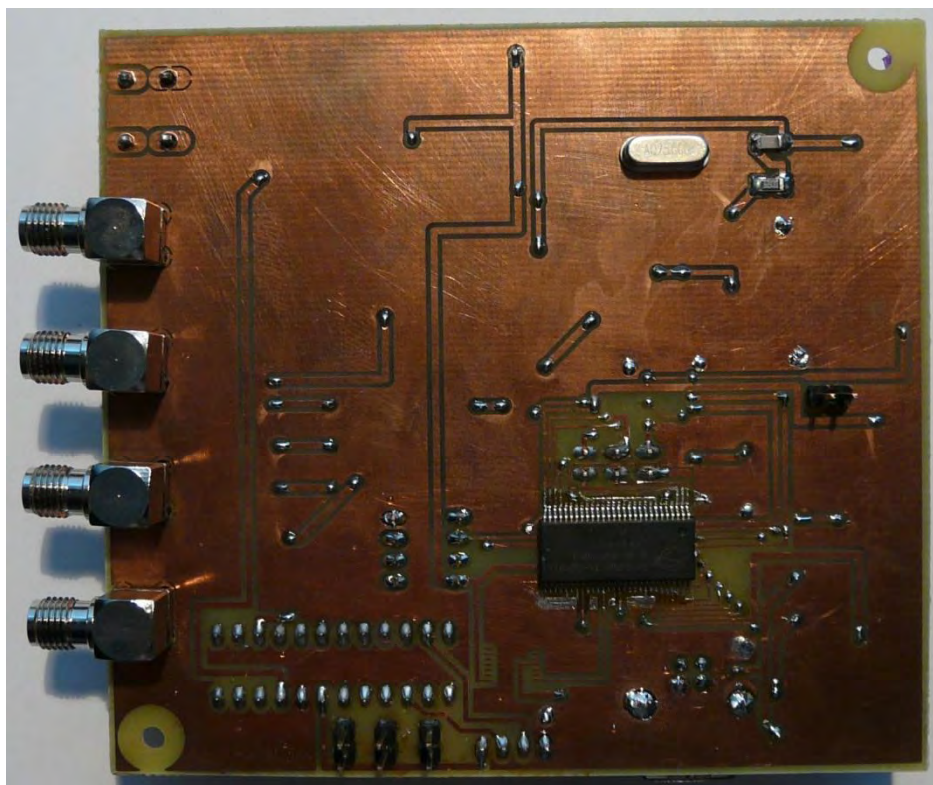


Obr. 25 Návrh DPS – vrstva BOTTOM (EAGLE 5.7.0)

Na obr. 26 a 27 jsou fotografie realizovaného obvodu.



Obr. 26 Fotografie DDS syntezátoru (TOP)

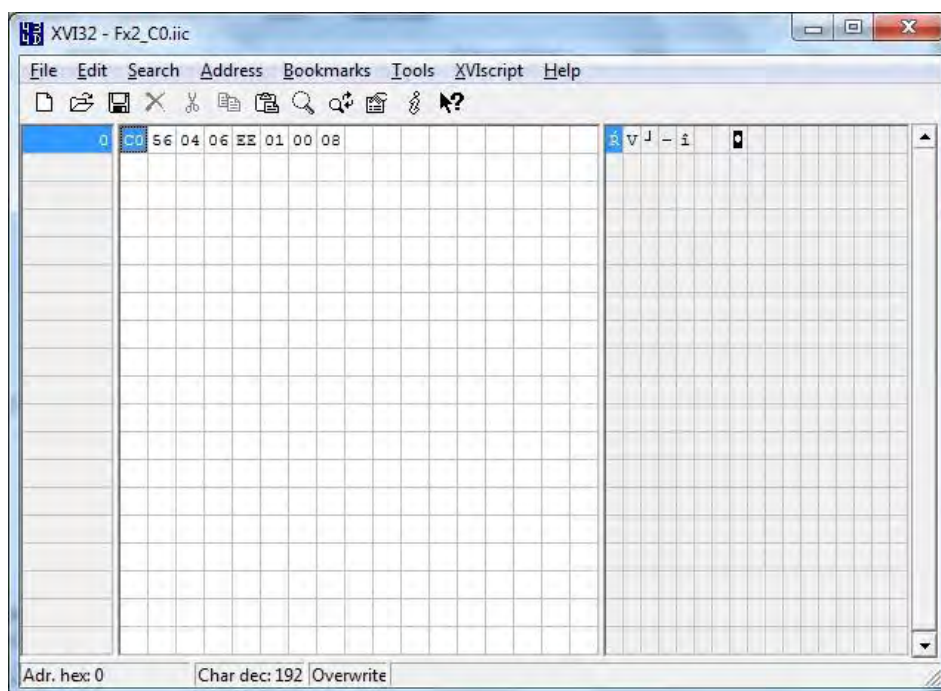


Obr. 27 Fotografie DDS syntezátoru (BOTTOM)

Po přivedení požadovaného napájecího napětí (3,5 V ÷ 5 V) je proudový odběr 0,08 – 0,09 A, což odpovídá předpokladu. Dále se napájecí proud mírně mění podle nastavení především referenčního kmitočtu – tedy velikosti PLL násobičky (4 ÷ 20). Na výstupech není ihned po připojení žádná výstupní frekvence.

3.8 Komunikace s PC

Po připojení k PC se USB2.0 mikrokontroler CYPRESS CY7C68013 hlásí jako nativní USB zařízení. Od CYPRESS dodávaný software pro práci s mikrokontrolerem obsahuje aplikaci umožňující přehrání EEPROM paměti a přepsání identifikace VID/PID. Je k dispozici i vzorový image EEPROM paměti. Tento image je potřeba upravit v libovolném HEX editoru a změnit hodnoty odpovídající VID_0456/PID_EE06, jak je zobrazeno na obr. 28. Poté stačí image nahrát do paměti. Nyní je již mikrokontroler rozpoznán ovladačem Analog Devices jako AD9958 Evaluation Board.



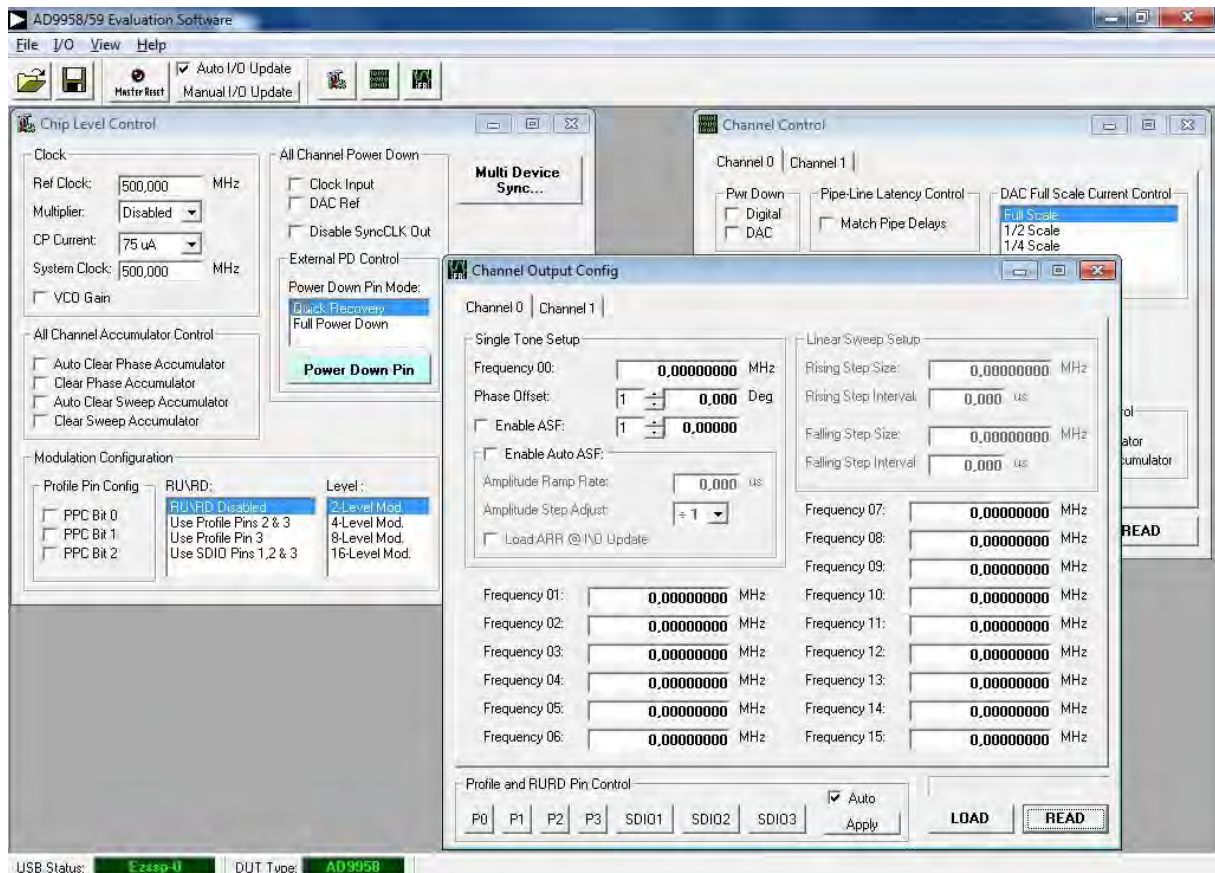
Obr. 28 Editace image EEPROM paměti DDS syntezátoru

Díky shodné identifikaci dojde k nahrání firmware od Analog Devices a to umožní použití dodávaných knihoven pro komunikaci pomocí USB rozhraní, na nichž bude stát vlastní softwarová aplikace.

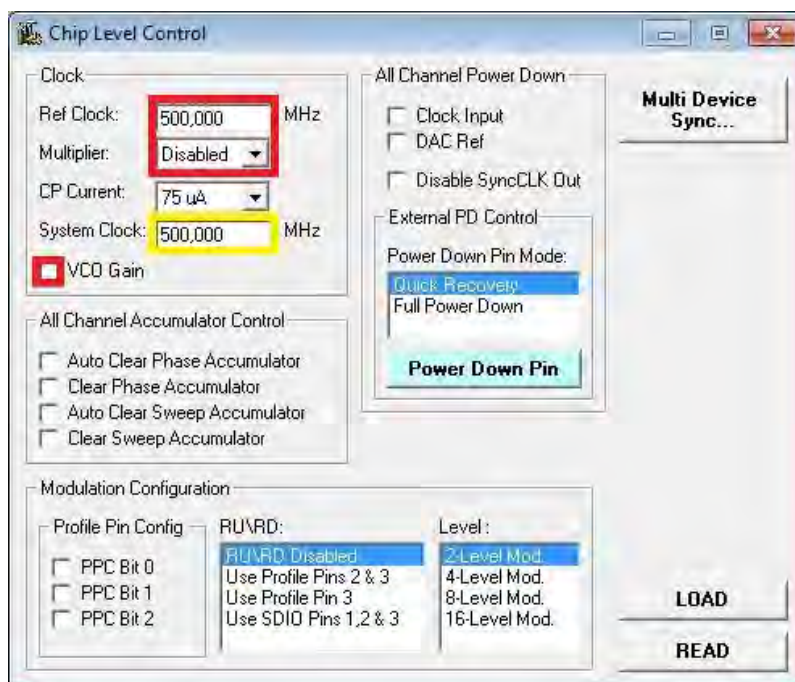
Nespornou výhodou je zároveň možnost použití i přímo dodávané aplikace pro Evaluation Board, která umožňuje velmi komplexní nastavení všech parametrů syntezátoru včetně řízení amplitudy, modulací, atd. Rozhraní aplikace je na obr. 29. K volnému stažení jsou i kompletní zdrojové kódy napsané ve Visual Basic 6.0.

Standardně je v aplikaci nastavena referenční frekvence na 500 MHz – obr. 30. To je pro použité zapojení potřeba změnit a to zadáním referenční frekvence 25 MHz (frekvence použitého krystalu) a zapnutím PLL násobičky na hodnotu 20, přičemž automaticky bude zvolen VCO Gain. Výsledná systémová frekvence bude maximálních 500 MHz. Změna se zapíše stiskem tlačítka LOAD v příslušném okně.

Nyní je již možné provádět různá nastavení syntezátoru a to stejným způsobem - po zadání požadovaných změn se tlačítkem LOAD nahrají nastavení do registrů syntezátoru a dojde tak k jejich aplikaci.



Obr. 29 Aplikace pro řízení syntezátoru od Analog Devices



Obr. 30 Aplikace Analog Devices – detail okna s nastavením Ref. kmitočtu

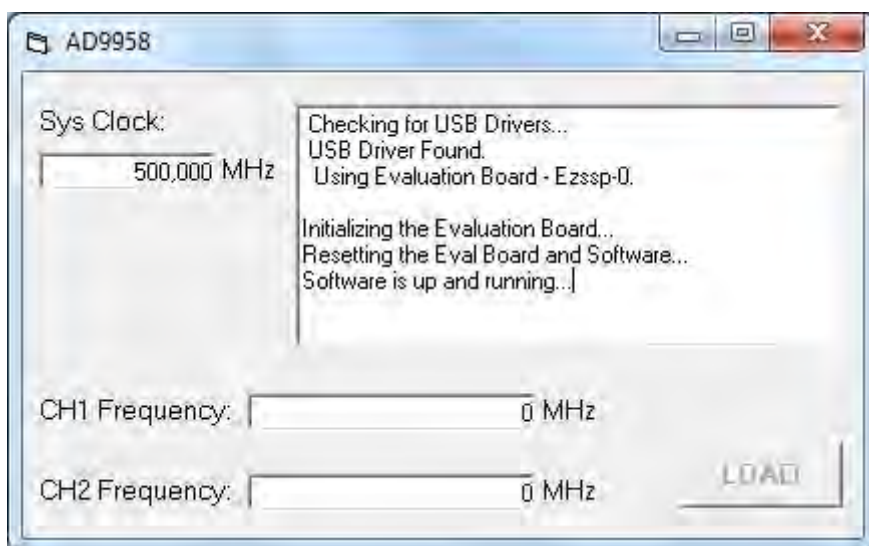
3.9 Návrh řídicí aplikace

Aplikace je postavena na knihovnách od Analog Devices sloužících k ovládání komunikace přes USB rozhraní. Vzhledem k tomu, že knihovny jsou napsány v jazyce Visual Basic 6.0 (VB6.0) a konverze do novějšího programovacího jazyka je obtížná, je i vlastní aplikace napsána ve VB6.0.

Logika programu je taková, že jsou po inicializaci alokovány proměnné velikostí odpovídající registrům AD9958 – registrové mapy. Tyto mapy jsou plněny při zadávání nastavení z prostředí řídicí aplikace, ale nejsou nahrávány ihned do syntezátoru. Syntezátor je schopen přijmout jeden, ale i všechny řídicí registry a až poté provést jejich aplikaci. Toho je právě využito – uživatel může zadat několik změn, které zasahují do libovolného množství registrů syntezátoru a až poté provést jejich hromadné nahrání a aplikaci.

3.9.1 Popis fungování aplikace

Kompletní výpis je v příloze 3. Po spuštění je inicializována procedura Main(), která nejprve přiřadí VID/PID odpovídající syntezátoru. Poté je provedena kontrola, zda-li je nainstalován ovladač USB. Pokud je instalace v pořádku, je ověřeno připojení AD9958 k PC. V okně běžícího programu, zobrazeném na obr. 31, jsou tyto kroky signalizovány po spuštění ve stavovém TextBoxu příslušnými zprávami o úspěch či neúspěchu provedené kontroly.



Obr. 31 Navržená aplikace pro ovládání DDS syntezátoru

Dále jsou alokovány mimo jiné již zmíněné registrové mapy a ověřen stav signálu RURD_2, který pro fungování standardní komunikace musí být nastaven na log. „low“. Syntezátor je na závěr provedených kontrol resetován do defaultního stavu.

Pro potřeby aplikace je zbytečné umožňovat nastavování referenční frekvence jako u aplikace Analog Devices, kde je to nutno provést po každém spuštění. Procedura Main() proto provede ještě nastavení referenční frekvence na (25 x 20)MHz a aktivaci VCO Gain.

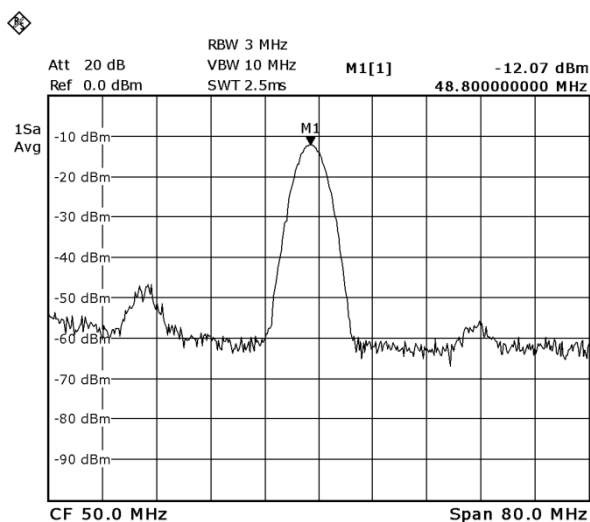
Požadovaná frekvence pro oba výstupní kanály se zadává do zbylých TextBoxů. Zadávání je ošetřeno proti zapsání jiných, než číselných hodnot. Požadovaná frekvence dále nemůže přesahovat frekvenci systémovou, tedy 500 MHz. Tlačítko LOAD je neaktivní do zadání frekvence alespoň pro jeden výstupní kanál. Po zadání frekvence a stisku tlačítka Enter nebo kliknutí myši mimo příslušný TextBox, je zadaná hodnota převedena na nejbližší možnou odpovídající rozlišení registru definujícímu výstupní frekvenci syntezátoru. Provedené změny jsou uloženy v registrových mapách v PC. Při stisku tlačítka LOAD je celá

příslušná část registrových map zapsána do odpovídajících registrů syntezátoru a signálem IO Update jsou ihned změny aplikovány. V době zápisu do registrů není z důvodu ochrany proti chybě tlačítko LOAD povoleno. Zápis trvá vzhledem k vysoké rychlosti komunikace jen velmi krátkou dobu. Na výstupu daného kanálu je možno ihned vidět nastavenou frekvenci.

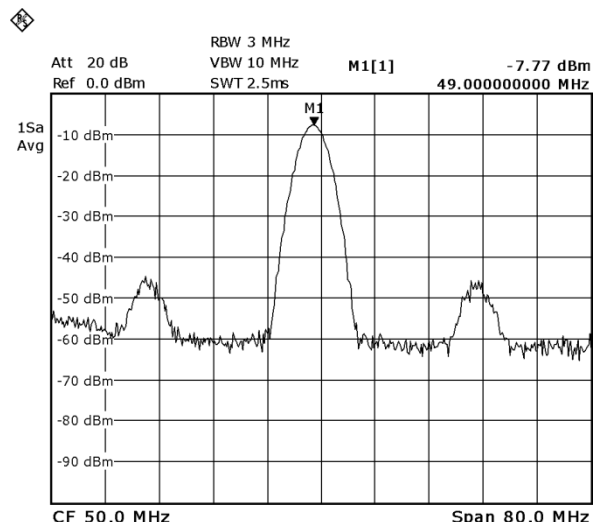
3.10 Měření výstupní frekvence

Na následujících obrázcích jsou spektrálním analyzátozem Rohde&Schwarz ZVL změřené výstupní frekvence a to vždy pro daný kanál (CH1 [obr. 32 ÷ 41], CH2 [obr. 42 ÷ 51]) před a za výstupní dolní propustí (DP). Proměřeny jsou frekvence 50 MHz, 100 MHz, 125 MHz, 200 MHz a 250 MHz.

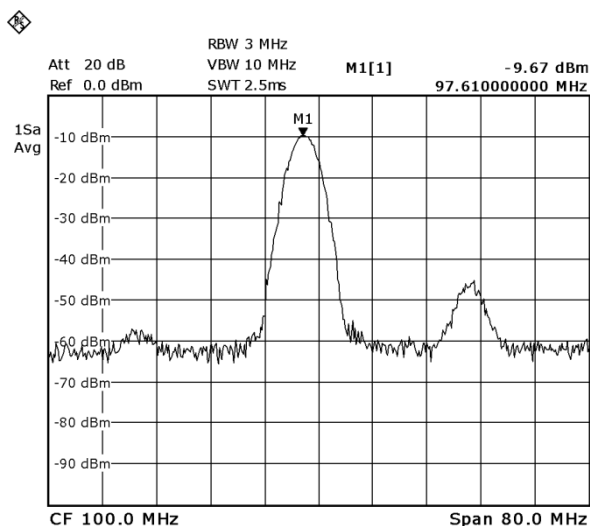
V každém spektru je výstupní signál změřen pomocí markeru pro přesné určení jeho výkonové úrovně a frekvenční odchylky.



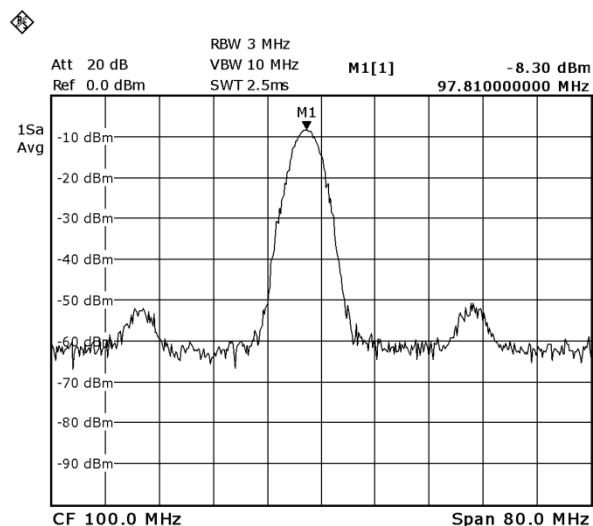
Obr. 32 CH1 - frekvence 50 MHz, před DP



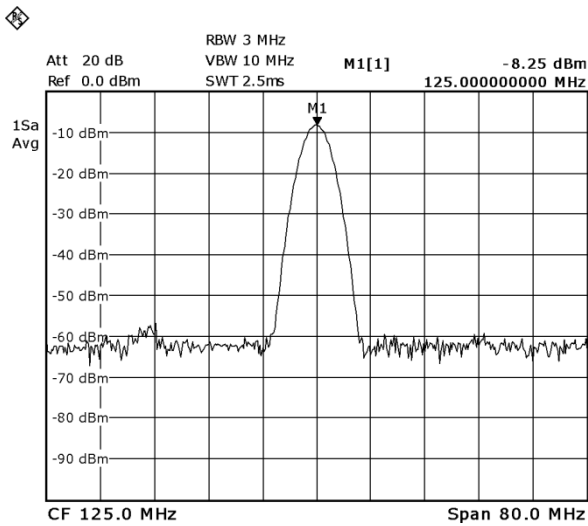
Obr. 33 CH1 - frekvence 50 MHz, za DP



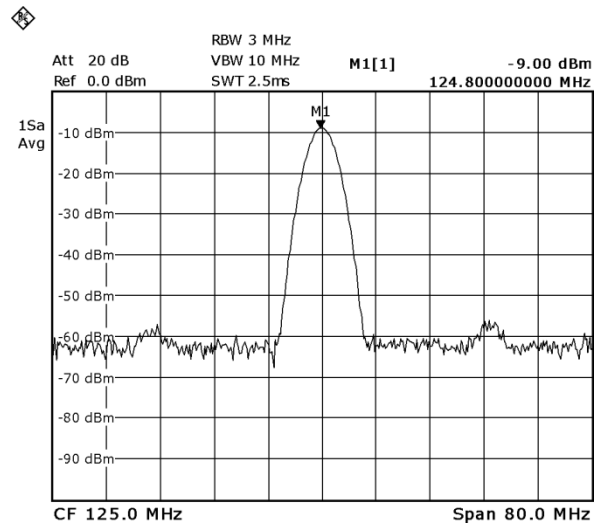
Obr. 34 CH1 - frekvence 100 MHz, před DP



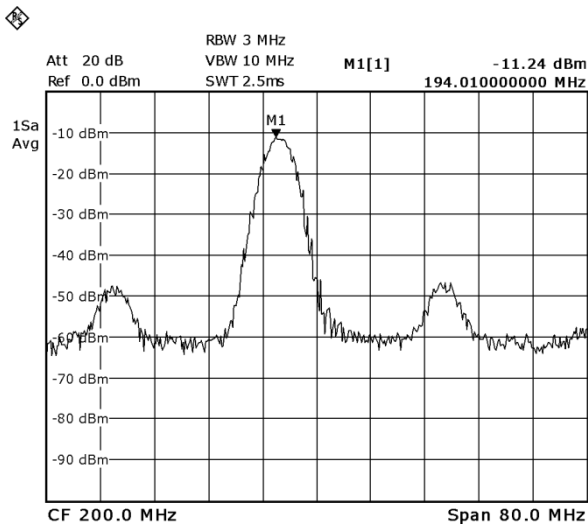
Obr. 35 CH1 - frekvence 100 MHz, za DP



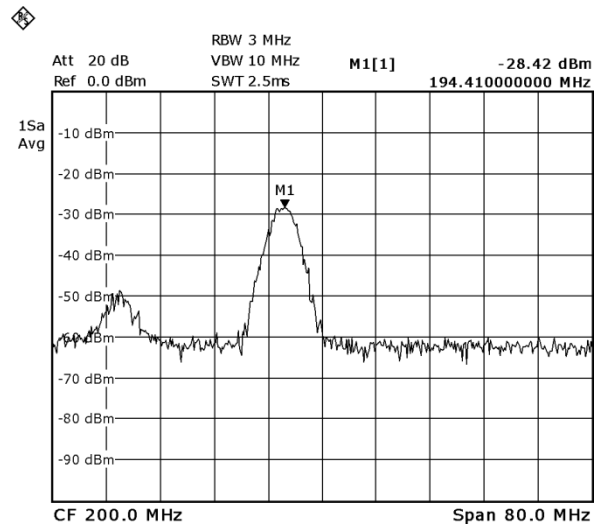
Obr. 36 CH1 - frekvence 125 MHz, před DP



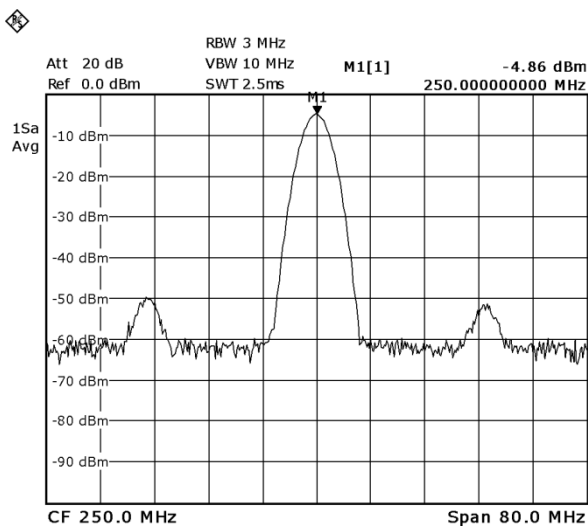
Obr. 37 CH1 - frekvence 125 MHz, za DP



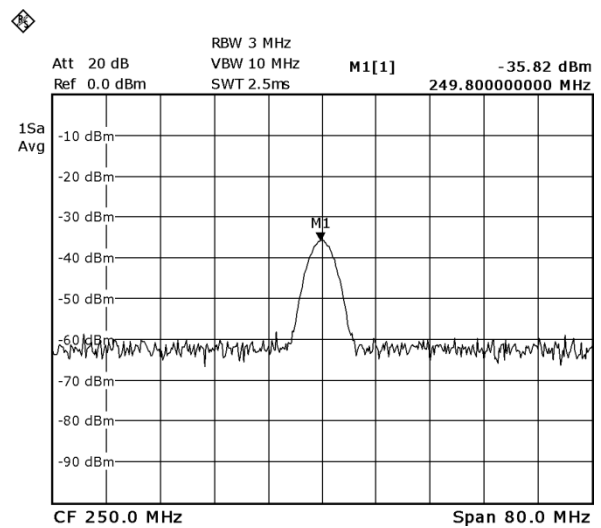
Obr. 38 CH1 - frekvence 200 MHz, před DP



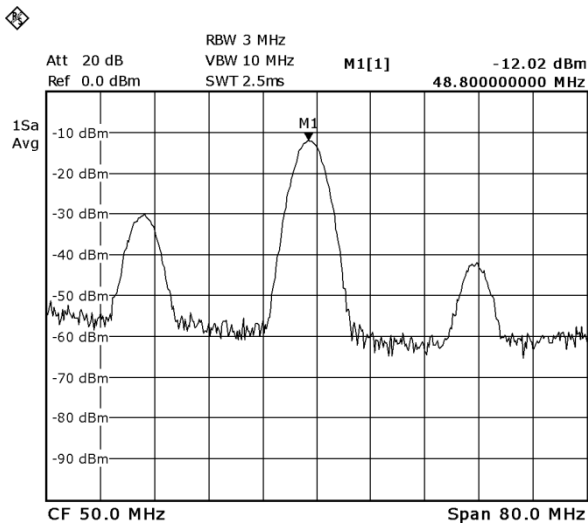
Obr. 39 CH1 - frekvence 200 MHz, za DP



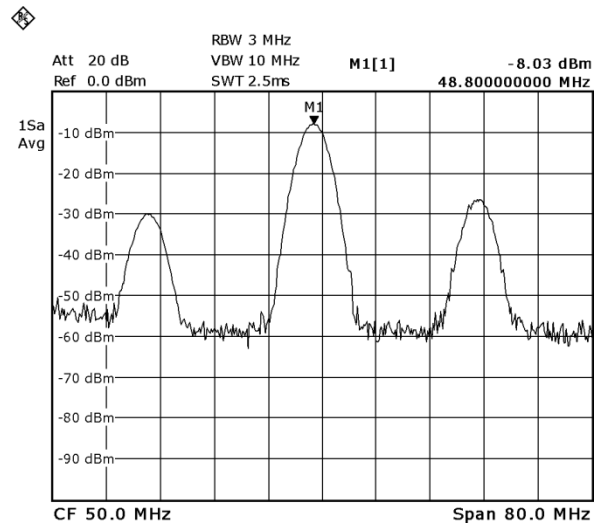
Obr. 40 CH1 - frekvence 250 MHz, před DP



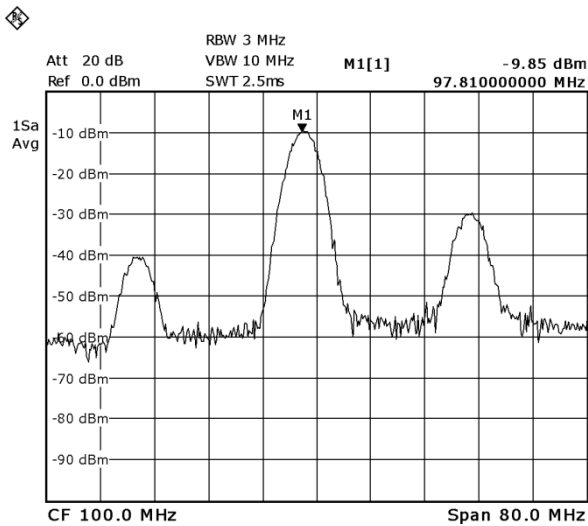
Obr. 41 CH1 - frekvence 250 MHz, za DP



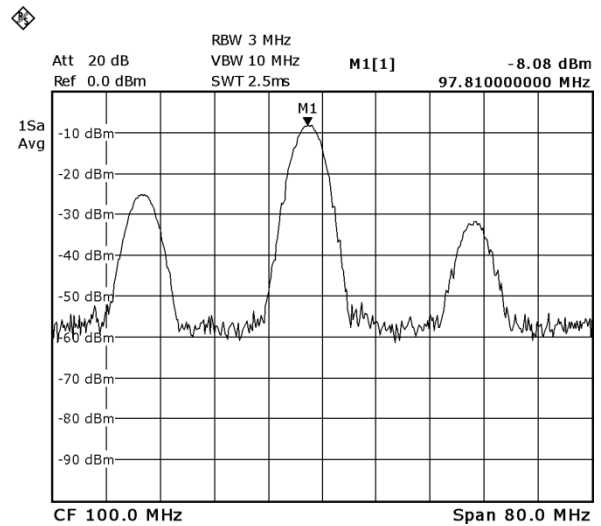
Obr. 42 CH2 - frekvence 50 MHz, před DP



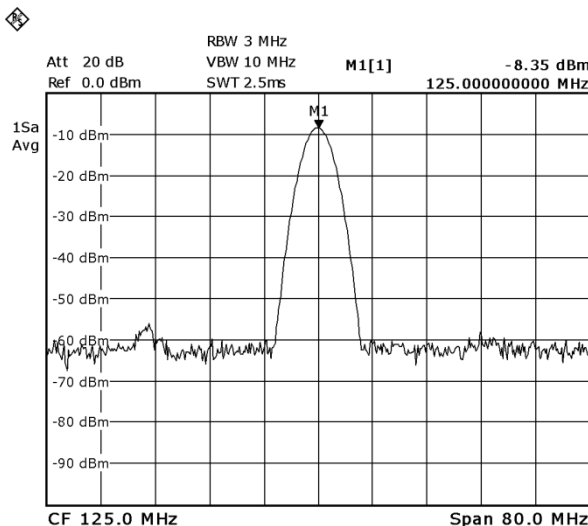
Obr. 43 CH2 - frekvence 50 MHz, za DP



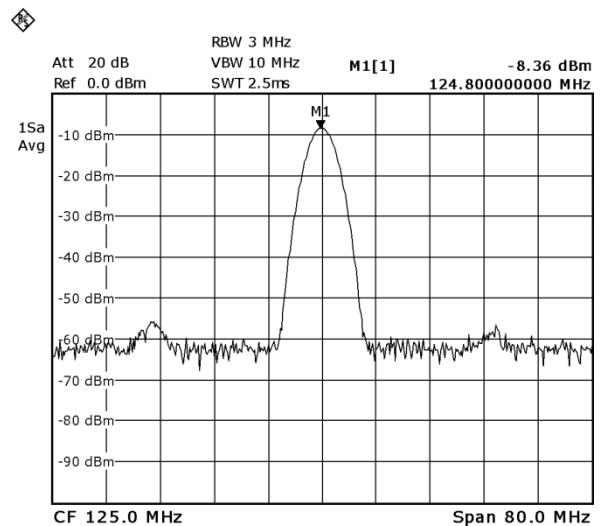
Obr. 44 CH2 - frekvence 100 MHz, před DP



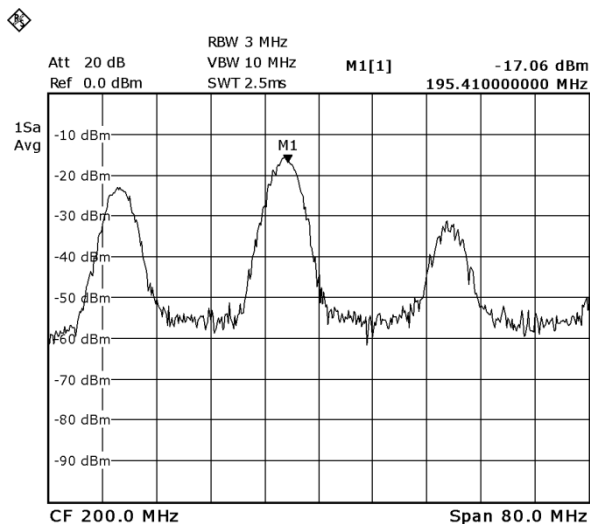
Obr. 45 CH2 - frekvence 100 MHz, za DP



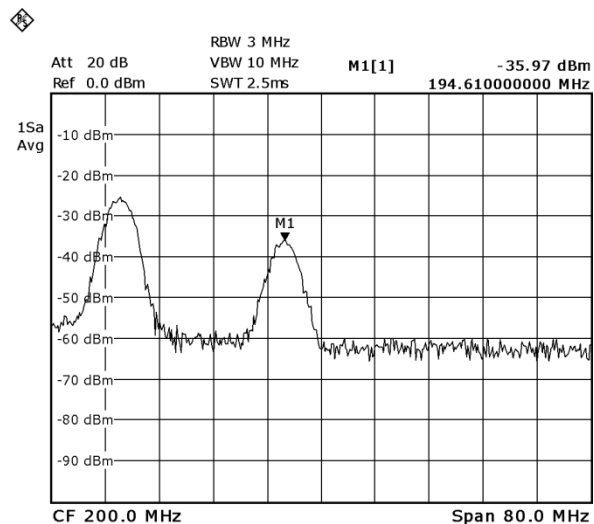
Obr. 46 CH2 - frekvence 125 MHz, před DP



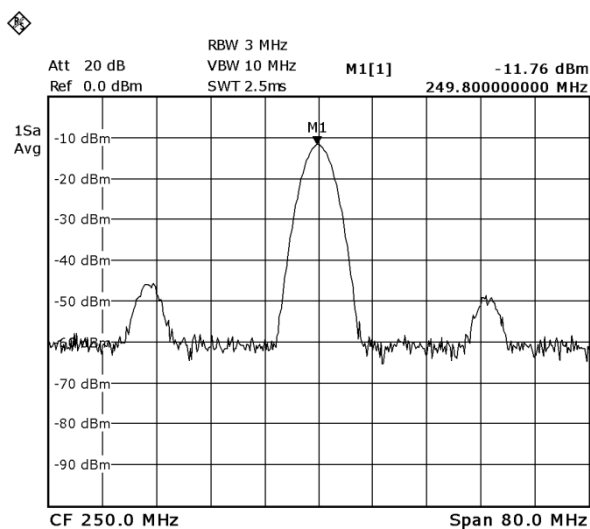
Obr. 47 CH2 - frekvence 125 MHz, za DP



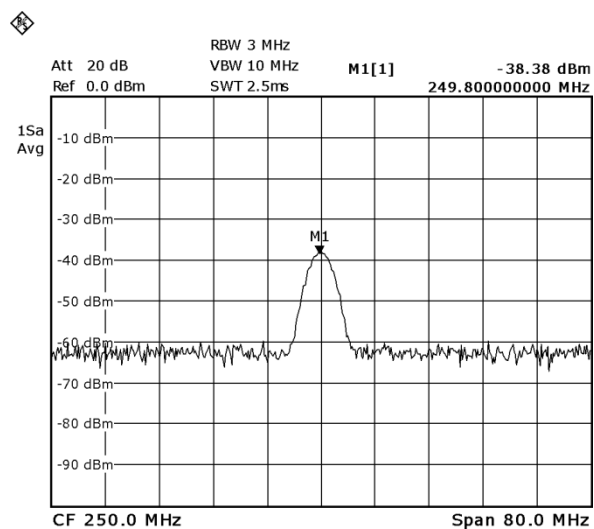
Obr. 48 CH2 - frekvence 200 MHz, před DP



Obr. 49 CH2 - frekvence 200 MHz, za DP



Obr. 50 CH2 - frekvence 250 MHz, před DP



Obr. 51 CH2 - frekvence 250 MHz, za DP

V naměřených spektrech jsou vidět v některých případech frekvenční odchylky, jejichž hodnoty se pohybují do 6 MHz. Bohužel se v měřených spektrech objevuje i parazitní harmonická složka s frekvencí 62,5 MHz. To vyžaduje další zkoumání především návrhu DPS, kde pravděpodobně k jejímu vzniku dochází. Výstupní výkon se při většině frekvencí pohybuje nad hodnotou -10 dBm. Vliv výstupní dolní propusti je patrný u frekvencí vyšších než 200 MHz, kde dochází k tlumení signálu.

Syntezátor byl v průběhu různých měření ponechán v provozu v řádu hodin a nebyl zjištěn žádný problém způsobený delší činností. Nebyl pozorován nadměrný ohřev částí syntezátoru ani jiné vlivy, které by mohly mít vliv na změny vlastností v průběhu delšího provozu.

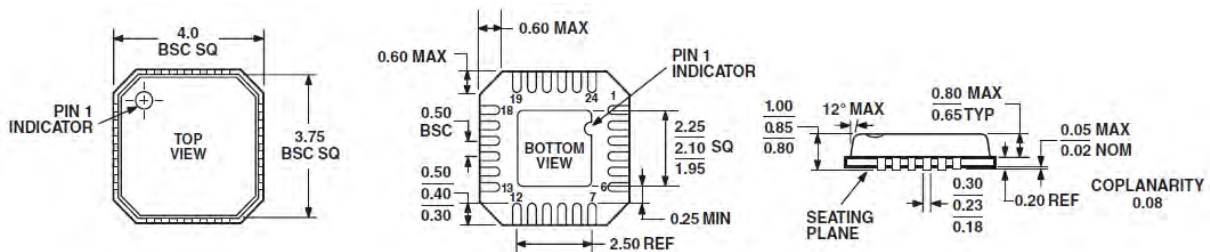
4 Návrh PLL syntezátoru – s nepřímou koherentní syntézou

Pro PLL (Phase Locked Loop) syntezátor byl zvolen opět obvod firmy Analog Devices. Konkrétně byl zvolen obvod ADF4252, který je určen zejména pro základnové stanice mobilních sítí (GSM, CDMA, ...) a má v sobě implementovány dva syntezátory kmitočtu pro rozsah 250 MHz – 3 GHz a 50 MHz – 1,2 GHz.

4.1 ADF4252

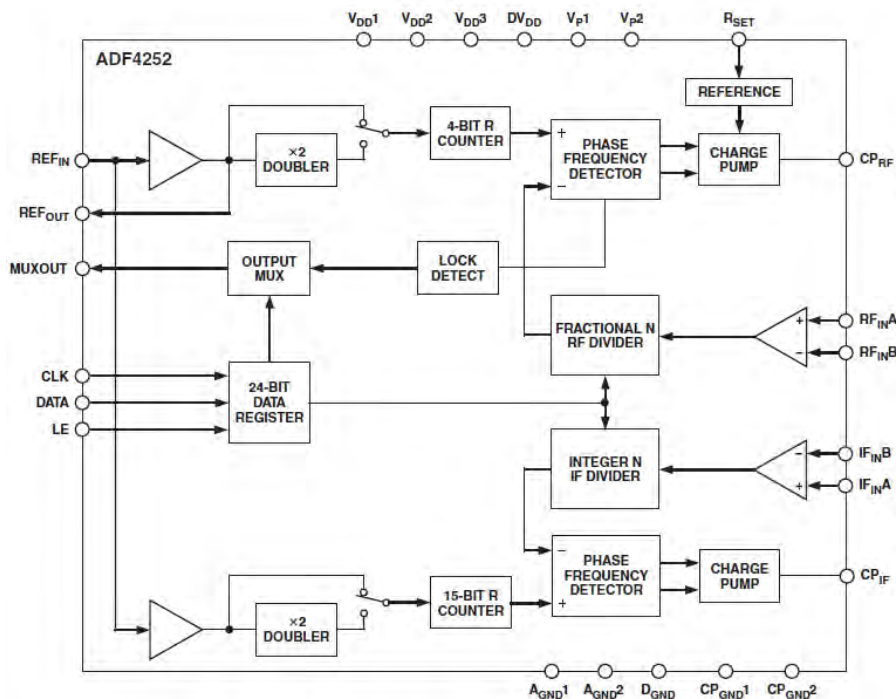
ADF4252 je duální frakční-N/integer-N frekvenční syntezátor, který může být použit k implementaci lokálních oscilátorů (LO) pro bezdrátové přijímače a vysílače. RF i IF syntezátor obsahuje nízkošumový digitální PFD (Phase Frequency Detector), nábojovou pumpu a programovatelný referenční dělič. RF syntezátor má Σ - Δ frakční interpolátor, který umožňuje programovatelné frakční-N dělení. IF syntezátor má programovatelné integer-N čítače. PLL je implementováno, pokud je syntezátor použit s externím filtrem smyčky a VCO (Voltage Controlled Oscillator).

Řízení všech interních registrů probíhá skrze jednoduché rozhraní (3 vodiče). Požadované napájení je 2,7 V – 3,3 V a může být odpojeno v době, kdy syntezátor není používán. Jako pouzdro je použito 24pinové LFCSP, jehož rozměry jsou vidět na obr. 52.



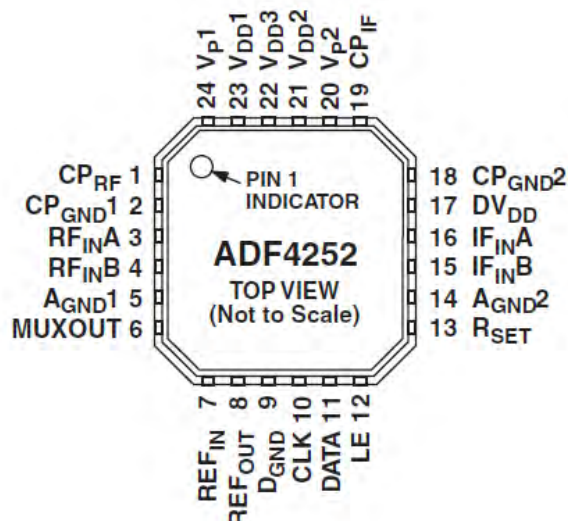
Obr. 52 ADF4252 - Rozměry pouzdra LFCSP-24 v mm [4]

Na obr. 53 je možno vidět kompletní vnitřní blokovou strukturu obvodu ADF4252 převzatou z [4].



Obr. 53 Vnitřní bloková struktura ADF4252 [4]

Další obrázek – obr. 54 ukazuje konfiguraci pinů doplněnu v následující tabulce – tab. 5 o jejich popis včetně popisu jednotlivých funkcí.



Obr. 54 Popis pinů pouzdra obvodu ADF4252 [4]

Tab. 5 Popis funkce pinů

Pin č.	Označení	I/O	Popis
1	CP _{RF}	O	Výstup RF nábojové pumpy. Běžně připojen k filtru smyčky, který je připojen na vstup VCO.
2	CP _{GND1}	I	Zem RF nábojové pumpy.
3	RF _{INA}	I	Vstup RF předděliče. Tento signál je běžně brán z VCO.
4	RF _{INB}	I	Komplementární vstup RF předděliče.
5	A _{GND1}	I	Analogová zem RF syntezátoru.
6	MUXOUT	O	Výstup multiplexoru umožňující buď výběr RF, IF; škálování RF, IF nebo externí přístup k referenční frekvenci.
7	REF _{IN}	I	Referenční vstup. CMOS vstup s nominálním prahovým napětím $V_{DD}/2$ a vstupní impedancí 100 k Ω . Signál může být přiveden z TTL nebo CMOS krystalového oscilátoru.
8	REF _{OUT}	O	Referenční výstup.
9	D _{GND}	I	Digitální zem pro Frakční Interpolátor.
10	CLK	I	Hodinový vstup. Hodinový signál slouží k převodu sériových dat do registrů. Data jsou převedena do posuvného registru s náběžnou hranou CLK. Jedná se o CMOS vstup s vysokou impedancí.
11	DATA	I	Sériový datový vstup. Sériová data jsou načtena v pořadí od MSB a se třemi LSB kontrolními bity. Jedná se o CMOS vstup s vysokou impedancí.
12	LE	I	Načtení povoleno. CMOS vstup. Při vysoké úrovni jsou data v posuvných registrech načtena. Výběr umístění načtených dat se provádí na základě kontrolních bitů.

Pin č.	Označení	I/O	Popis
13	R _{SET}	I	Připojení rezistoru mezi tento pin a zem nastavuje minimální výstupní proud nábojové pumpy. Vztah mezi I _{CP} a R _{SET} je $I_{CPmin} = \frac{1,6875}{R_{SET}}$.
14	A _{GND2}	I	Zem pro IF syntezátor.
15	IF _{INB}	I	Komplementární vstup IF předděliče.
16	IF _{INA}	I	Vstup IF předděliče. Signál je běžně brán z VCO.
17	DV _{DD}	I	Kladné napájení po frakční interpolační část. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. DV _{DD} musí mít stejné napětí jako V _{DD1} , V _{DD2} a V _{DD3} .
18	CP _{GND2}	I	Zem IF nábojové pumpy.
19	CP _{IF}	O	Výstup IF nábojové pumpy. Běžně připojen k filtru smyčky, který signál přivádí k externímu VCO.
20	V _{p2}	I	Napájení IF nábojové pumpy. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. Napětí by mělo být větší nebo rovno V _{DD2} .
21	V _{DD2}	I	Kladné napájení IF části. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. V _{DD2} nabývá hodnoty 3 V ± 10 %. V _{DD2} musí mít stejné napětí jako V _{DD1} , V _{DD3} a DV _{DD} .
22	V _{DD3}	I	Kladné napájení pro RF digitální část. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. V _{DD3} nabývá hodnoty 3 V ± 10 %. V _{DD3} musí mít stejné napětí jako V _{DD1} , V _{DD2} a DV _{DD} .
23	V _{DD1}	I	Kladné napájení pro RF analogovou část. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. V _{DD1} nabývá hodnoty 3 V ± 10 %. V _{DD1} musí mít stejné napětí jako V _{DD2} , V _{DD3} a DV _{DD} .
24	V _{p1}	I	Napájení RF nábojové pumpy. Blokovací kondenzátory by měly být umístěny co nejbližší k pinu. Napětí by mělo být vyšší nebo rovno V _{DD1} .

4.2 USB řadič FT232R

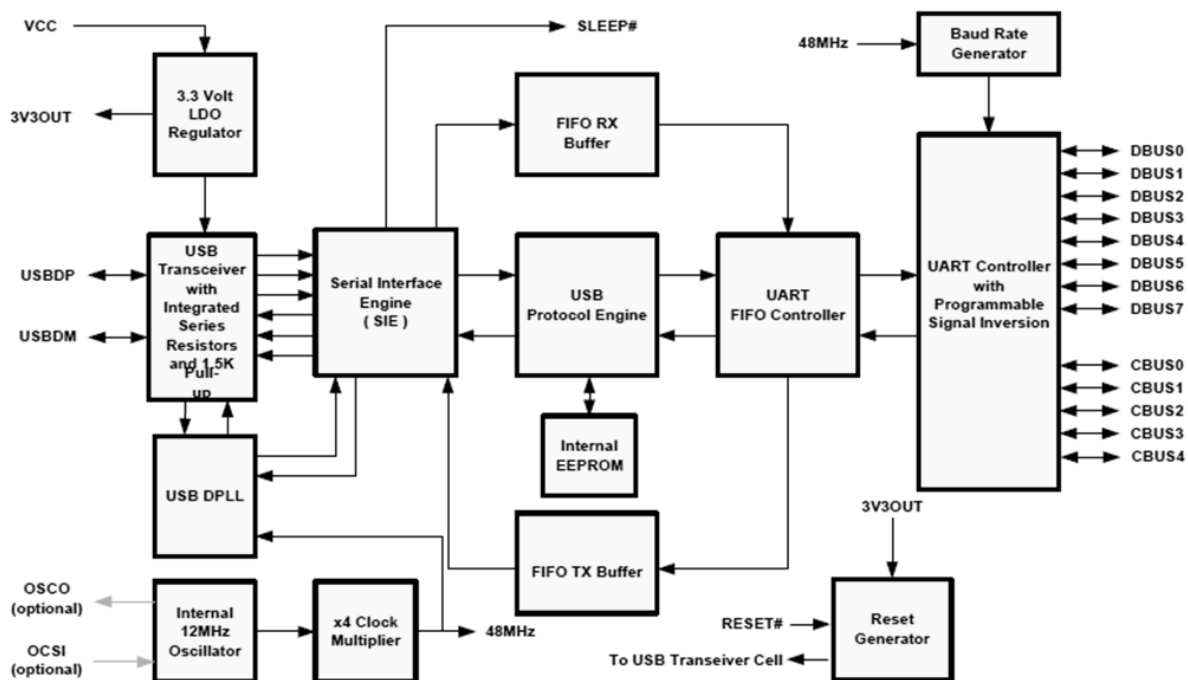
FT232R slouží jako převodník USB na UART rozhraní, které bude využíváno pro komunikaci s ADF4252. K ADF4252 nejsou na rozdíl od AD9958 poskytovány knihovny ani firmware pro komunikaci pomocí USB řadiče EZ-USB FX2 od Cypress. To značně komplikuje jeho použití, jelikož nemůže být, z důvodu odlišného způsobu komunikace, použit shodný firmware a jeho naprogramování je komplikované a bez Evaluation Board od Cypress možné pouze se značnou dávkou zkušeností.

Z těchto důvodů byl zvolen FT232R, který sice nedosahuje tak vysoké přenosové rychlosti, ale vnáší mnohá zjednodušení do pozdějšího návrhu způsobu komunikace. Mezi tato zjednodušení patří:

- Kompletní integrovaný protokol USB bez nutnosti editace firmware.
- Integrovaná 1024 bitová EEPROM pro uložení deskriptorů a konfigurace.
- Integrovaný generátor kmitočtu bez nutnosti připojení externího krystalu.
- Rychlost datového přenosu od 300 baud do 3 Mbaud.

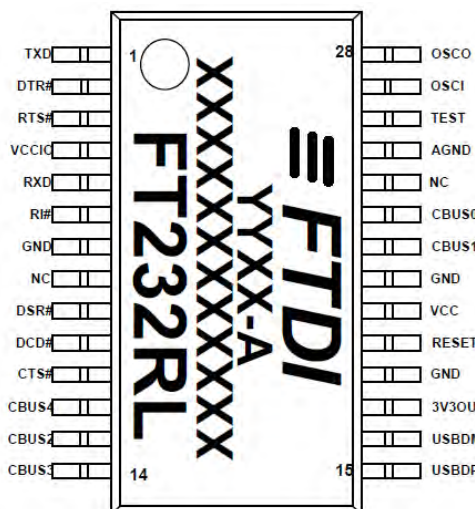
- Virtual Com Port (VCP) a přímé (D2XX) ovladače, které v mnoha případech eliminují nutnost vývoje vlastních USB ovladačů.
- Možnosti jednoduchého nastavení parametrů UART komunikace (počet datových bitů, stop bitů, atd.).
- Napájení z USB i možnost samostatného přivedení napájení.
- Integrovaný napěťový konvertor pro výstupní logické úrovně UART.
- Možnost invertování UART signálu.

Na obr. 55 je vidět vnitřní bloková struktura FT232R převzata z [16]. Kromě dále v návrhu použitého pouzdra 28 SSOP, se vyrábí ještě v provedení 32 QFN.

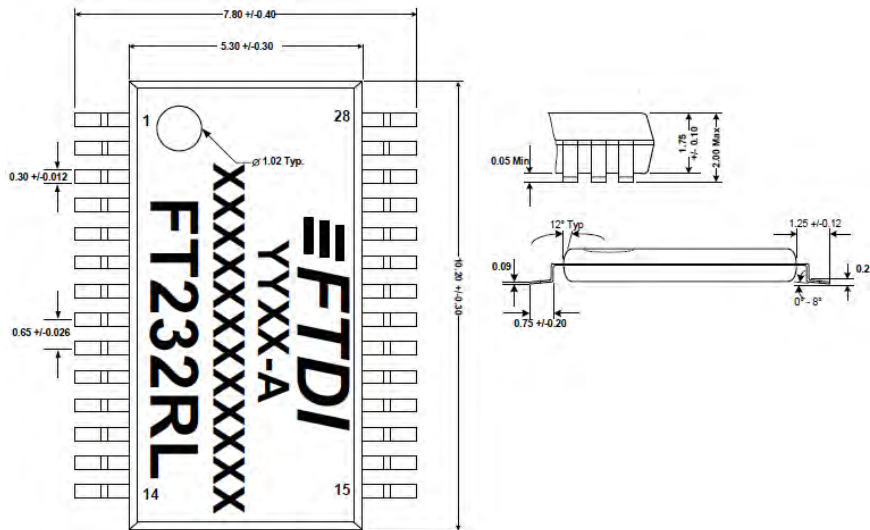


Obr. 55 Vnitřní bloková struktura FT232R [16]

Na následujícím obr. 56 je zobrazena konfigurace pinů použitého pouzdra 28 SSOP a na obr. 57 jsou zakresleny vnější rozměry pouzdra.



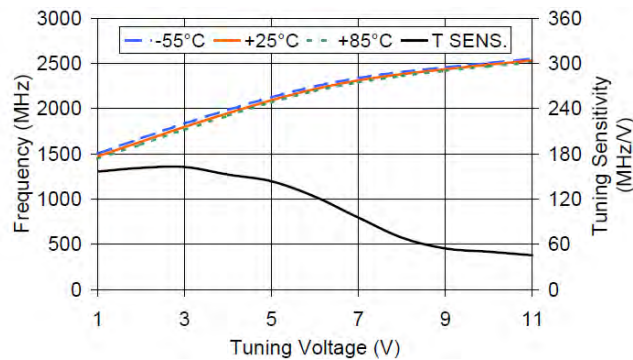
Obr. 56 Popis pinů pouzdra 28 SSOP obvodu FT232R [16]



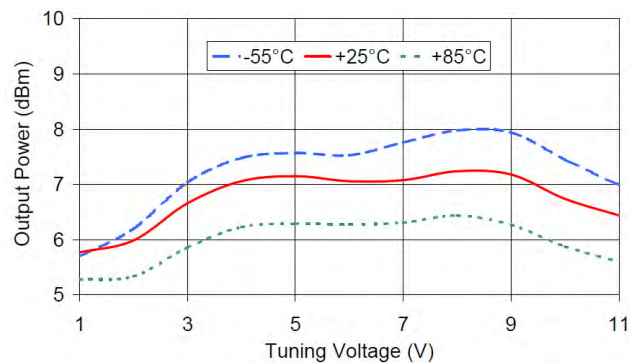
Obr. 57 FT232R - Rozměry pouzdra 28 SSOP v mm [16]

4.3 VCO ROS-2500+

VCO ROS-2500+ má kmitočtový rozsah 1,6 GHz – 2,5 GHz, což je pod maximální frekvenci 3 GHz syntezátoru ADF4252. Střední výstupní výkon je 6,5 dBm a ladicí napětí 0,5 V – 14 V. Na následujících obr. 58 a 59 jsou vidět grafy závislosti frekvence spolu se změnou frekvenční citlivosti na napětí a závislost výstupního výkonu na napětí.



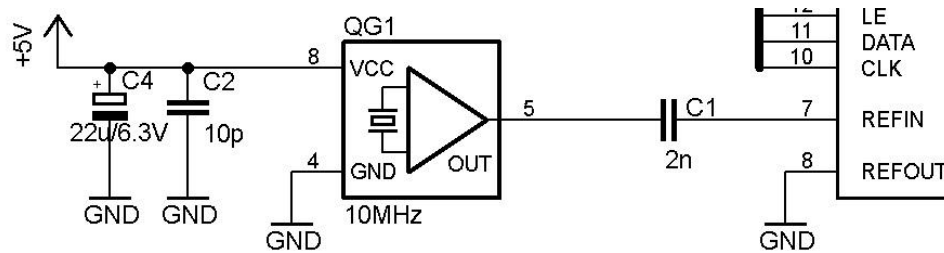
Obr. 58 Závislost frekvence a velikosti frekvenční citlivosti ROS-2500+ na napětí [8]



Obr. 59 Závislost výstupního výkonu ROS-2500+ na napětí [8]

4.4 Návrh zdroje referenčního hodinového signálu

Jako zdroj referenčního hodinového signálu bude sloužit krystalový oscilátor s frekvencí 10 MHz v pouzdra DIL08S. Zapojení zobrazené na obr. 60 je doplněno o vyhlazovací kondenzátory na základě doporučení [7].

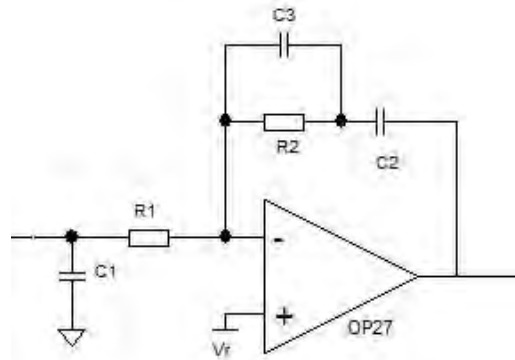


Obr. 60 Zapojení zdroje referenčního hodinového signálu navržené v EAGLE 5.7.0

4.5 Návrh výstupních obvodů a smyčky

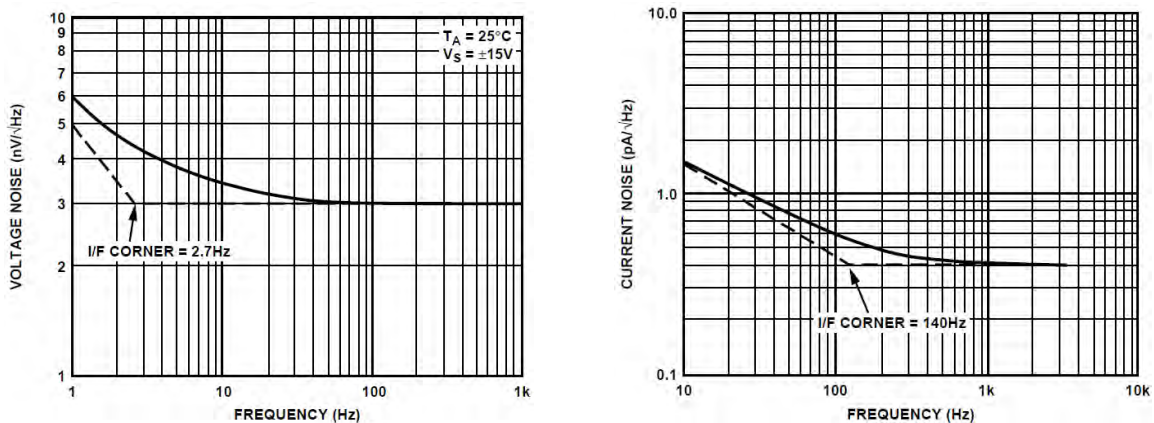
Je zapotřebí použít aktivní filtr smyčky, protože ladicí napětí VCO ROS-2500+ je v rozmezí 0,5 V – 14 V. Pro volbu typu filtru smyčky a jeho další návrh slouží např., v tomto návrhu použitý, program ADIsimPLL 3.30.

U obvodu syntezátoru ADF4252 jsou využity RF vstupy/výstupy (frekvenční rozsah 250 MHz – 3 GHz). Navržený filtr je 4. řádu a jeho schéma je na obr. 61.



Obr. 61 Schéma aktivního filtru 4. řádu

Operační zesilovač byl zvolen od Analog Devices a to typ OP27 z důvodu velmi dobrých šumových vlastností. Napěťová a proudová šumová hustota je na obr. 62.

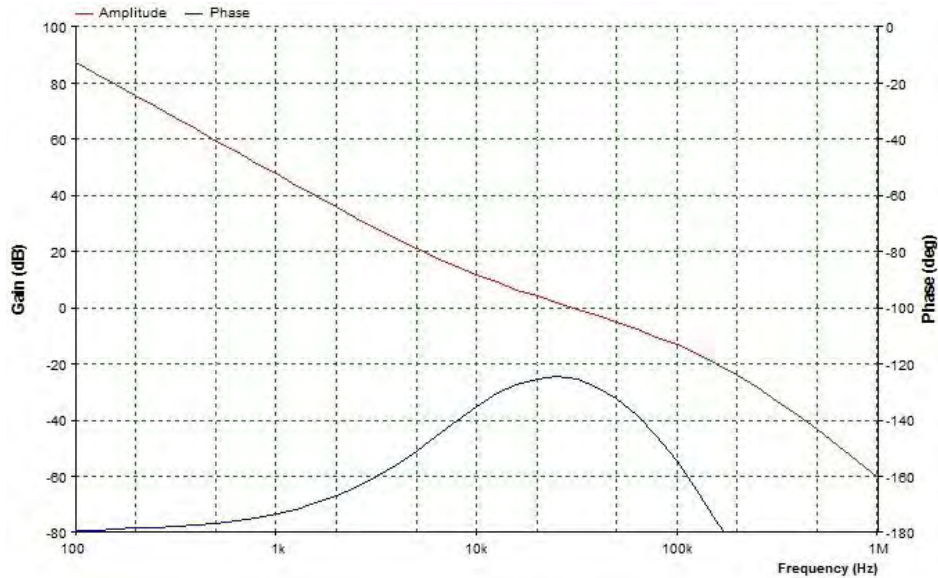


Obr. 62 Napěťová a proudová šumová hustota Analog Devices OP27 [9]

ADIsimPLL obsahuje knihovny parametrů jak pro operační zesilovač OP27, tak pro VCO ROS-2500+, není proto nutné jejich parametry zdávat ručně. Dále je třeba nastavit šířku pásma smyčky, která ovlivňuje rychlost ustálení frekvence a to tak, že čím je šířka větší, tím je ustálení rychlejší (zde 30 kHz ... musí být volena s ohledem na velikost fázového šumu, který současně ovlivňuje), a fázovou jistotu (55°). Fázová jistota je rozdíl mezi 180° a fází

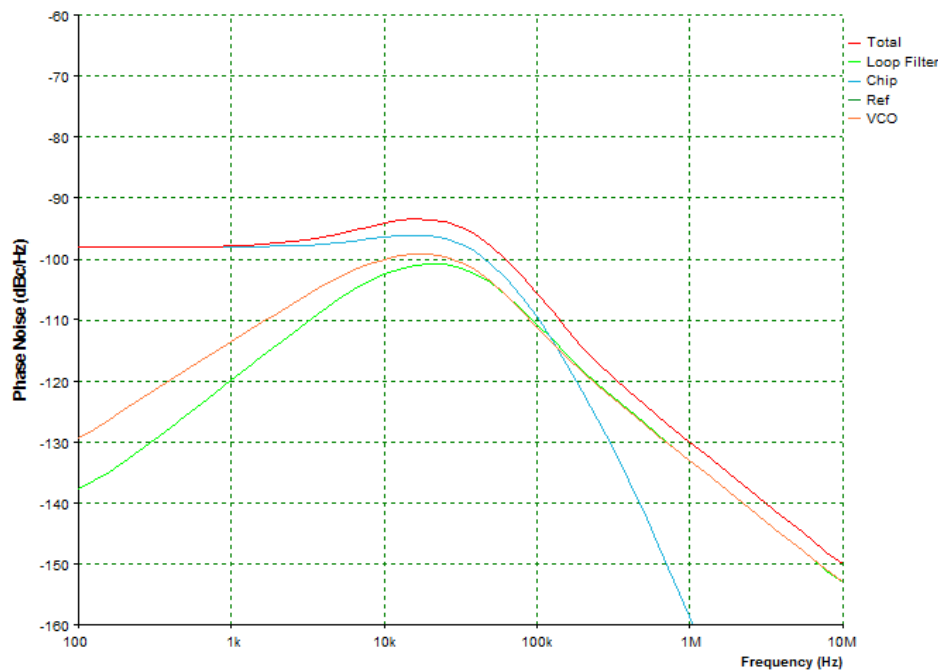
přenosové funkce otevřené smyčky na kmitočku rovném šířce pásma smyčky, tzn. kde má přenosová funkce otevřené smyčky jednotkový zisk. Fázová jistota je úměrná činiteli tlumení a pro zajištění stability by měla být volena nejméně 40° až 45° .

Po zadání všech parametrů program vypočítá hodnoty součástek ve filtru a vykreslí vybrané charakteristiky jak v závislosti na frekvenci, tak na čase. Získané hodnoty jsou vztaženy v tomto případě k frekvenci 2 GHz. Je možné upravovat dílčí parametry návrhu a sledovat změny vlastností filtru v celkovém pohledu. Následující obr. 63 ukazuje předpokládaný zisk otevřené smyčky v závislosti na frekvenci. Dle předpokladu zde jednotkový zisk (0 dB) odpovídá 30 kHz a fázová jistota $180^\circ - 125^\circ = 55^\circ$.



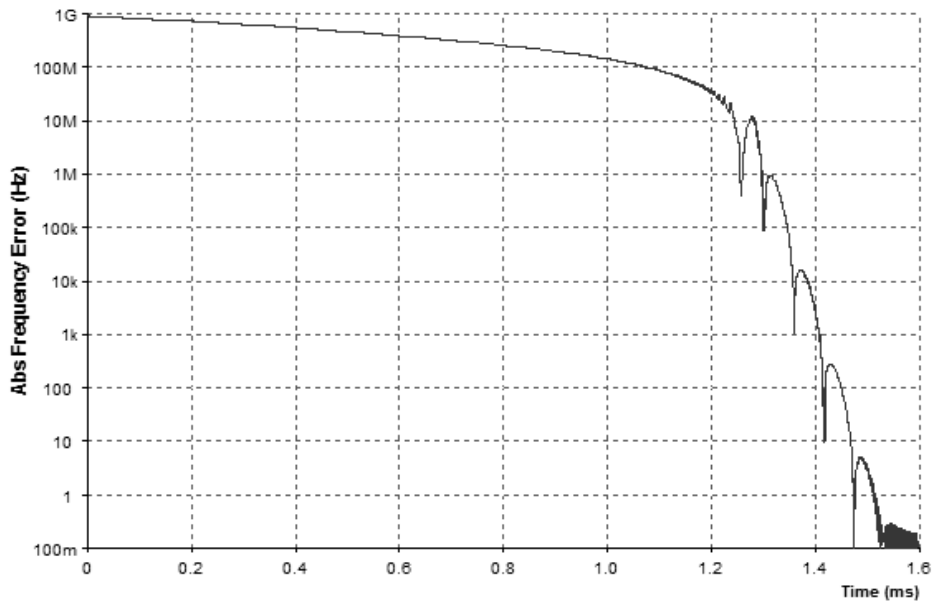
Obr. 63 Zisk otevřené smyčky (pro 2 GHz) simulovaný programem ADIsimPLL

Další graf na obr. 64 získaný v programu ADIsimPLL ukazuje předpokládaný fázový šum v závislosti na frekvenci a to celkový i vztažený k jednotlivým prvkům smyčky.



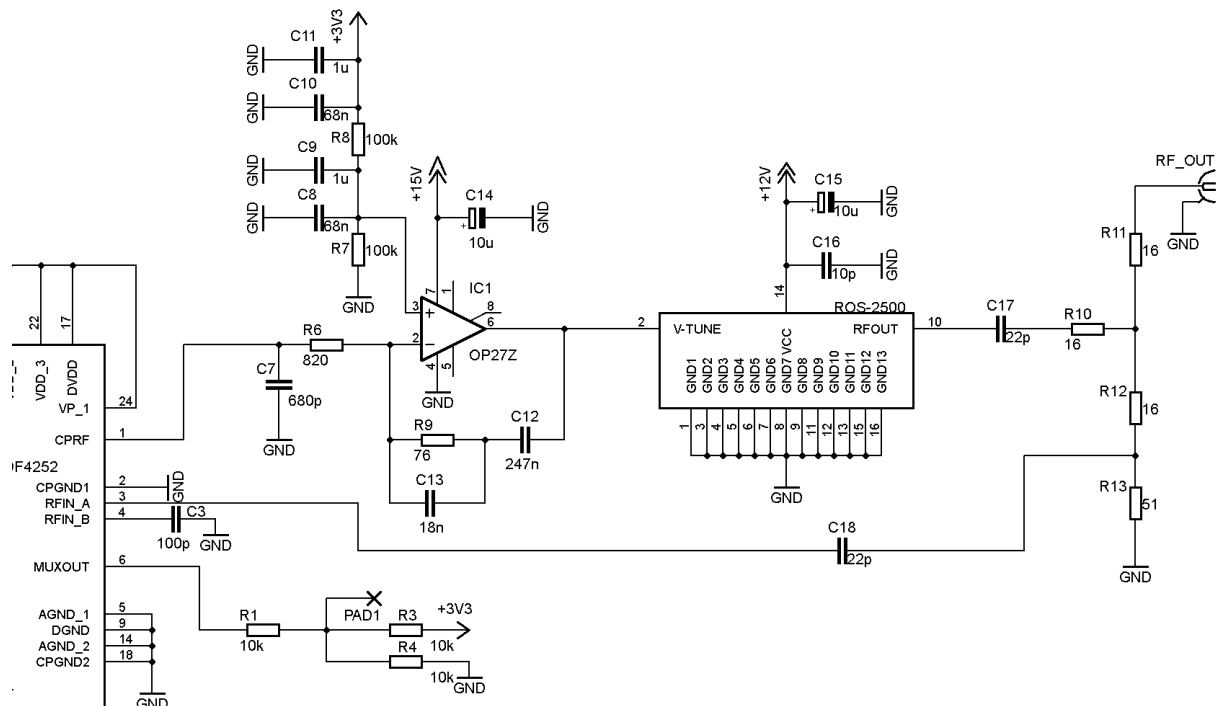
Obr. 64 Fázový šum (pro 2 GHz) simulovaný programem ADIsimPLL

Poslední zajímavý výsledek simulace je na obr. 65 a jedná se o dobu zachycení smyčky při maximální změně frekvence.



Obr. 65 Doba zachycení smyčky pro změnu z 1600 MHz na 2500 MHz (ADIsimPLL)

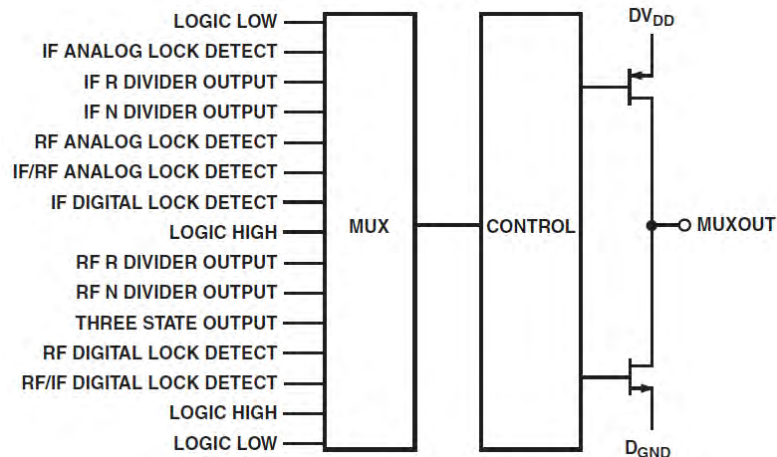
Na obr. 66 je schéma zapojení smyčky a výstupních obvodů navržené v EAGLE 5.7.0 na základě získaných parametrů. Na neinvertující vstup operačního zesilovače je pomocí děliče složeného z rezistorů R7, R8 přivedeno napětí poloviční oproti napětí na RF nábojové pumpě (VP_1). Výstupní napětí nábojové pumpy je rovno právě polovině jejího napájecího napětí. Součástí děliče napětí jsou i kondenzátory, které slouží k blokování šumu vznikajícího na rezistorech děliče.



Obr. 66 Zapojení smyčky a výstupních obvodů navržené v EAGLE 5.7.0

Na výstupu VCE je zapojen dělič výkonu (R10, R11, R12) s vložitelným útlumem 6 dB. Vzhledem k výstupnímu výkonu VCO 6,5 dBm se bude výstupní výkon celého syntezátoru pohybovat blízko hodnoty 0 dBm. Konektor RF_OUT je řešen jako SMA Female se čtvercovou základnou, otočený o 90°.

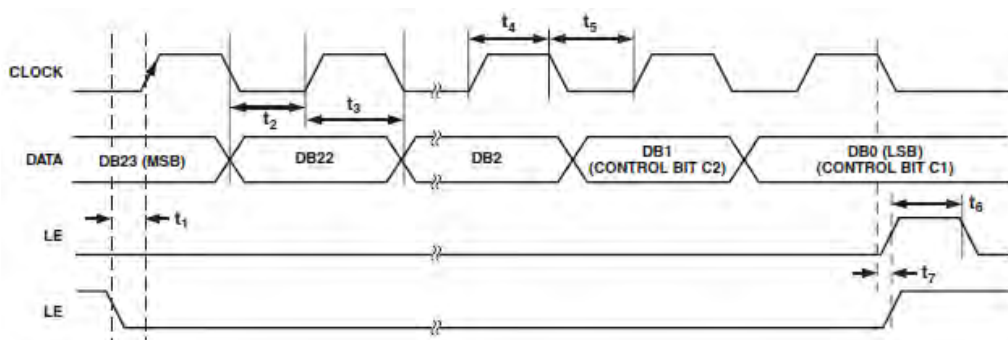
Za pozornost stojí dále výstup MUXOUT, který umožňuje uživateli přistupovat k různým vnitřním signálům čipu. Stav MUXOUT je řízen bity M4 ÷ M1 v *master register*. Na obr. 67 je blokový diagram MUXOUT obvodu převzatý z [4]. V navrženém zapojení je výstup vyveden na PAD1, což bude umožňovat snadné připojení osciloskopu či jiných měřicích přístrojů.



Obr. 67 Blokový diagram obvodu MUXOUT [4]

4.6 Návrh komunikačního rozhraní

ADF4252 disponuje SPI kompatibilním sériovým rozhraním. Datový přenos je uskutečňován pomocí signálů CLK, DATA a LE. V době, kdy je LE (Latch Enable) neaktivní (low), jsou s náběžnou hranou CLK přenášeny do příslušného vstupního registru data. Po naplnění každého registru musí přijít signál LE (high) pro aplikaci přenesených dat. Průběh přenosu je znázorněn na obr. 68 [4].

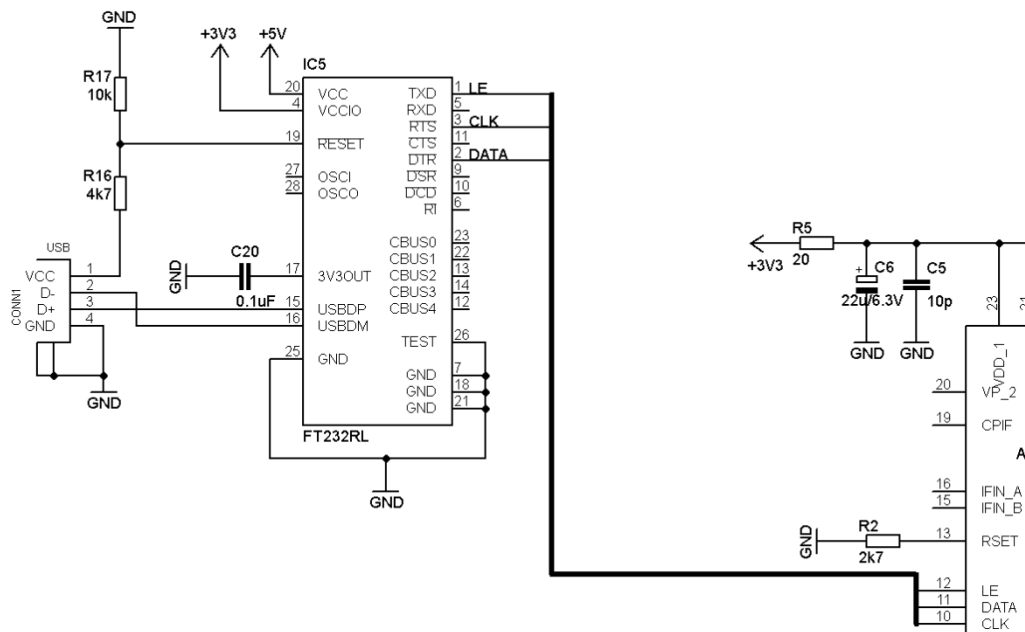


Obr. 68 Diagram signálů datové komunikace ADF4252 [4]

Maximální frekvence sériového přenosu je 20 MHz, což znamená, že maximální rychlost update je 833 kHz – tedy update možný každou 1,2 μ s. Systémy využívající tento syntezátor obvykle pracují s časy mnohem delšími – ve stovkách mikrosekund, takže hodnota 1,2 μ s, je více než dostatečná.

Na obr. 69 je návrh zapojení s čipem FT232R. Důvody použití FT232R jsou vysvětleny v kapitole 4.2. Zapojení není komplikované, napájení je řešeno samostatně přivedením

5 V na vstup VCC. Napájení z USB slouží pouze k přivedení čipu z energeticky úsporného režimu spánku, do nějž se přepne po odpojení USB. Požadované napěťové úrovně pro komunikaci s ADF4252 jsou 0 V - 3,3 V. Z toho důvodu je na vstup VCCIO přivedeno napětí 3,3 V, které určuje právě napěťové úrovně komunikačních pinů. Signály LE, DATA, CLK jsou přivedeny na piny TxD, DTR a RTS. U signálů DTR a RTS lze přímo nastavovat logickou úroveň, takže se jimi budou dobře generovat průběhy datového signálu a signálu hodin o požadované periodě. Signál TxD je datovým výstupem UART, tudíž jsou na něj posílána data formátována do standardizovaných rámců, což velmi komplikuje přesné řízení jeho výstupu. Z toho důvodu je použit pro generování jednoduchého signálu LE.



Obr. 69 Zapojení komunikačního rozhraní s FT232R navržené v EAGLE 5.7.0

ADF4252 obsahuje 7 registrů, které slouží k jeho řízení. *Master Register* je společný oběma integrovaným syntezátorům, zbylých šest je rozděleno po třech pro oba syntezátory. V tomto návrhu je využíván pouze jeden ze syntezátorů. Řízení je tedy prováděno pomocí celkem čtyř registrů. Mapa registrů převzatá z [4] je na obr. 70. Identifikace příslušného registru se provádí pomocí třech posledních bitů. Výstupní frekvence je určena podle následujícího vztahu:

$$f_{OUT} = \left(INT + \frac{FRAC}{MOD} \right) \cdot \left(REF_{IN} \cdot \frac{1+D}{R} \right), \quad (22)$$

kde INT = Integer Division Factor, $FRAC$ = Fractionality, MOD = Modulus, REF_{IN} = Reference Frequency input, D = (REF_{IN}) Doubler, R = (REF_{IN}) Division factor.

Povolené rozsahy jednotlivých hodnot spolu s příklady jejich nastavení lze nalézt v [4]. Na stránkách Analog Devices je navíc dostupná online aplikace pro výpočet hodnot registrů na základě zadané frekvence. Prescaler v *RF R Divider Register* je nastaven na 8/9 (1). V *RF Control Register* jsou dále nastavení NOISE AND SPUR, které je navrženou aplikací standardně ve stavu LOWEST SPUR (000), nastavení napájecího proudu smyčky, který je součástí samotného návrhu smyčky a v tomto případě je 4,38 mA (11), nastavení Phase Detector polarity – NEGATIVE (0) a možnosti inicializace Power Down, aktivace třístavového výstupu a resetování interního čítače RF syntezátoru.

V *Master Register* jsou nastavení výstupu MUXOUT, který je standardně ve stavu LOGIC LOW (00), XO DISABLE nastaven na (1) z důvodu připojení externího oscilátoru (v případě opačného nastavení jsou využívány piny REF_{IN} a REF_{OUT} k vytvoření oscilátoru) a opět možnosti inicializace Power Down, třístavového výstupu a resetování interního čítače, které jsou ale v tomto případě platné pro oba integrované syntezátory (RF, IF).

RF N DIVIDER REG

RESERVED		8-BIT RF INTEGER VALUE (INT)									12-BIT RF FRACTIONAL VALUE (FRAC)										CONTROL BITS		
DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
P1	N6	N7	N6	N5	N4	N3	N2	N1	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	C3 (0)	C2 (0)	C1 (0)

RF R DIVIDER REG

PRESCALER		RF REF IN DOUBLER		4-BIT RF R COUNTER				12-BIT INTERPOLATOR MODULUS VALUE (MOD)													CONTROL BITS		
DB20	DB19	DB18	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
P3	P2	R4	R3	R2	R1	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	C3 (0)	C2 (0)	C1 (1)			

RF CONTROL REG

NOISE AND SPUR SETTING 3				RESERVED				NOISE AND SPUR SETTING 2			RF CP CURRENT SETTING			RESERVED		RF PD POLARITY		NOISE AND SPUR SETTING 1			RF POWER-DOWN		RF CP THREE-STATE		RF COUNTER RESET		CONTROL BITS		
DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0														
N3	T3	T2	T1	N2	CP2	CP1	0	P8	N1	P6	P5	P4	C3 (0)	C2 (1)	C1 (0)														

MASTER REG

MUXOUT				XO DISABLE		POWER-DOWN		CP THREE-STATE		COUNTER RESET		CONTROL BITS		
DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0				
M4	M3	M2	M1	P12	P11	P10	P9	C3 (0)	C2 (1)	C1 (1)				

Obr. 70 Mapa registrů pro řízení RF syntezátoru ADF4252 [4]

4.7 Návrh napájení

Syntezátor ADF4252 je napájen 3,3 V. Krystalový oscilátor a obvod FT232R jsou napájeny napětím 5 V. VCO ROS-2500+ požaduje napájecí napětí 12 V a operační zesilovač OP27 napětí +15 V.

Napájení celého obvodu je opět řešeno samostatně, jak ukazuje obr. 71. Na napájecí svorky je přivedeno napětí 15 V, které je dále přivedeno na vstupy tří stabilizátorů:

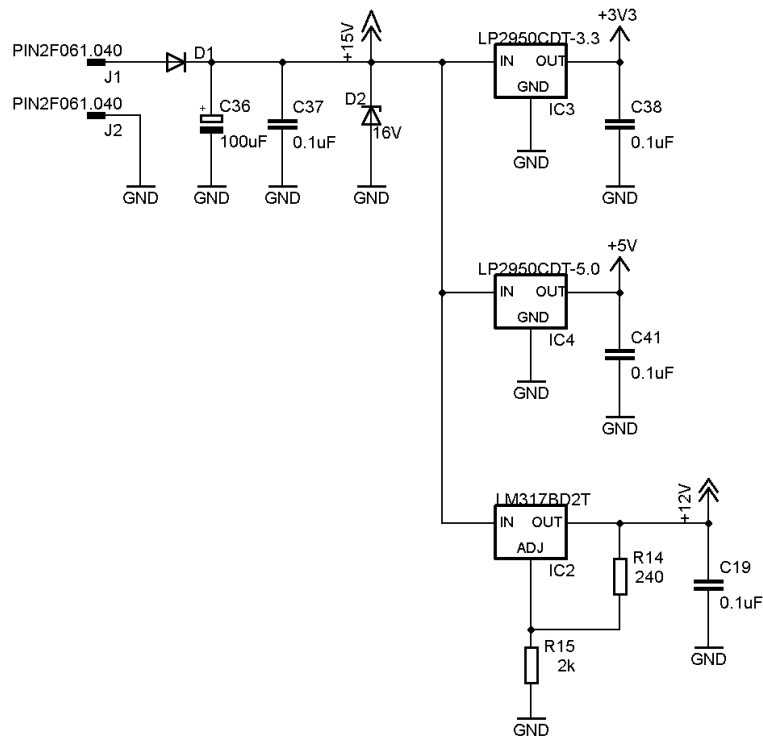
- LP2950CDT-3.3 pro získání napětí 3,3 V,
- LP2950CDT-5.0 pro získání napětí 5,0 V,
- LM317BD2T pro získání napětí 12 V

Stabilizátor LM317BD2T je nastavitelný pomocí rezistorů R14, R15 a to podle vztahu:

$$V_{out} = 1.25V \left(1 + \frac{R_{15}}{R_{14}} \right) + I_{Adj} R_{15}, \quad (23)$$

kde I_{Adj} (Adjust) dosahuje maximálně 100 μ A [10].

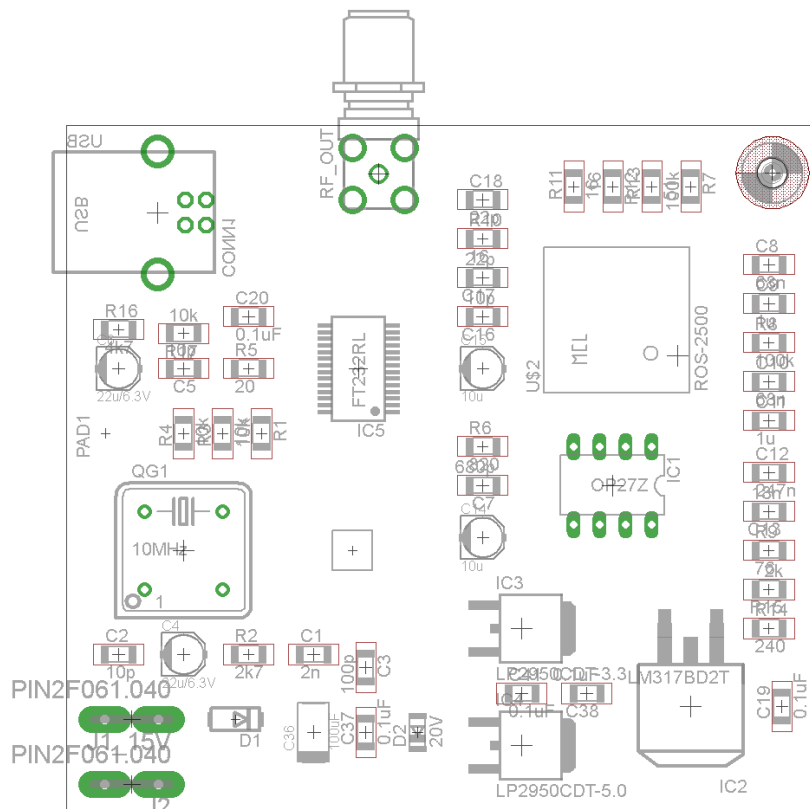
Na vstupu napájecího obvodu jsou připojeny dvě diody, které plní funkce ochrany a to ochranu proti případnému přepólování – dioda D1 (typ SS16) zapojena v sérii v kladné větvi, a ochranu proti přivedení vyššího napětí – dioda D2 (Zenerova dioda 16 V) zapojena mezi kladnou větví a zemí. Vyhlašovací kondenzátory jsou umístěny jak u napájecích stabilizátorů, tak i přímo u jednotlivých pinů použitých obvodů.



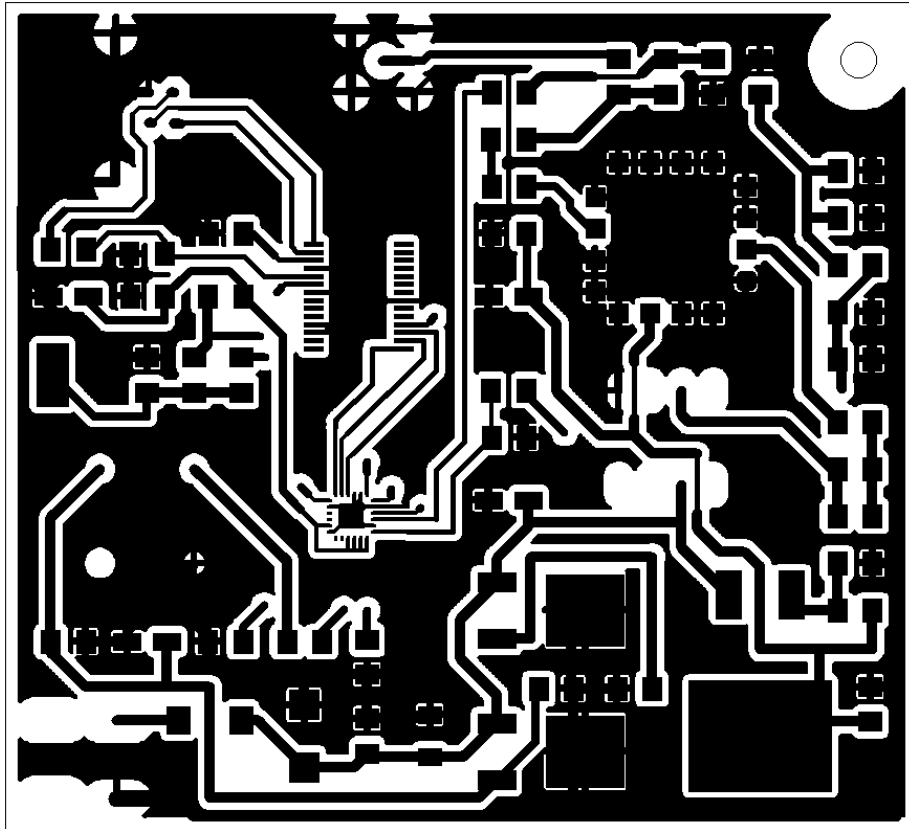
Obr. 71 Zapojení napájení navržené v EAGLE 5.7.0

4.8 Deska plošného spoje

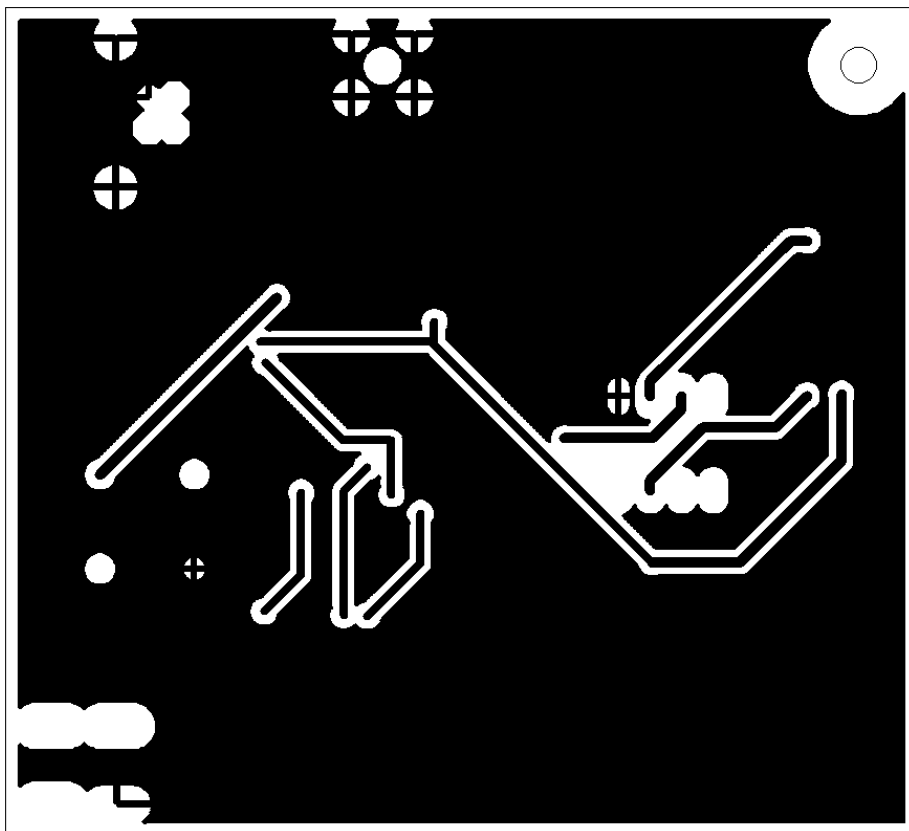
Obr. 73 a 74 zobrazují vrstvu TOP a BOTTOM desky plošného spoje syntezátoru. Pro výrobu desky je použit materiál FR4. Kondenzátory a rezistory jsou použity SMD velikosti 1206. USB konektor je typu B. Následující obr. 72 zobrazuje rozmístění součástek na desce.



Obr. 72 Rozmístění součástek PLL syntezátoru na DPS (EAGLE 5.7.0)

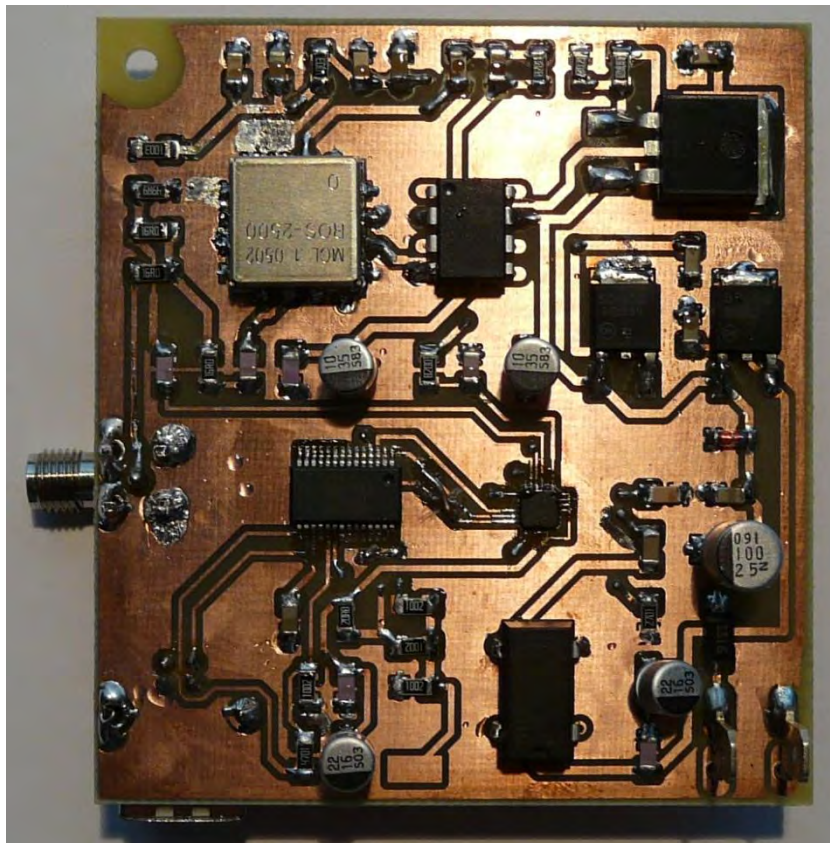


Obr. 73 Návrh DPS – vrstva TOP (EAGLE 5.7.0)

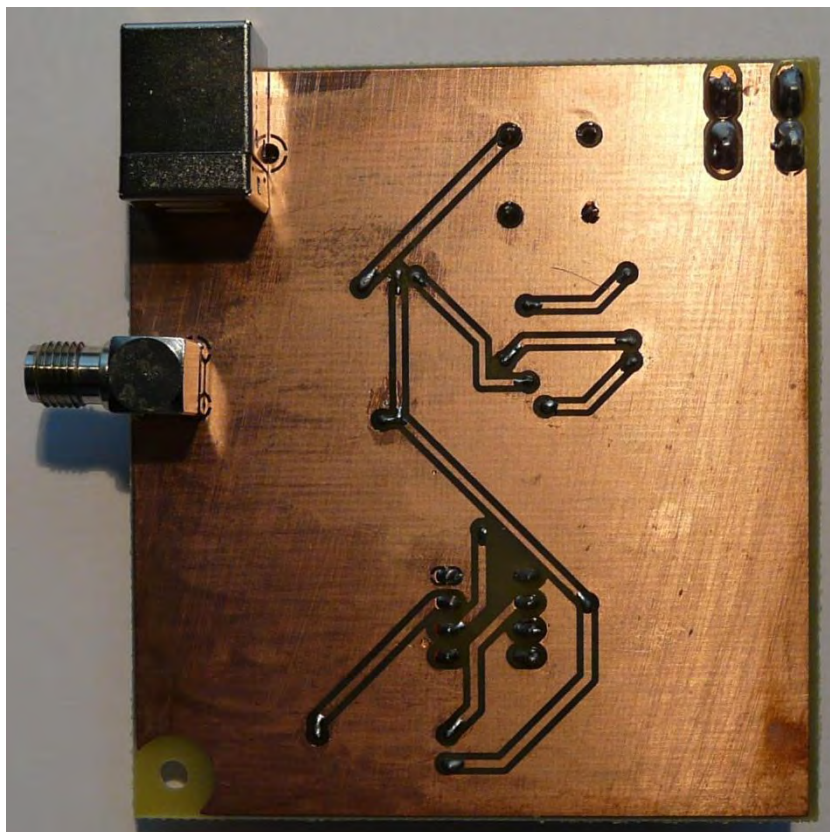


Obr. 74 Návrh DPS – vrstva BOTTOM (EAGLE 5.7.0)

Na obr. 75 a 76 jsou fotografie realizovaného obvodu.



Obr. 75 Fotografie PLL syntezátoru (TOP)



Obr. 76 Fotografie PLL syntezátoru (BOTTOM)

Po přivedení požadovaného napájecího napětí (15 V) je proudový odběr přibližně 0,05 A, což odpovídá předpokladu. To platí pro stav při odpojeném USB, kdy je čip FT232R v úsporném režimu. Po připojení USB dojde k nárůstu odebíraného proudu na hodnotu okolo 0,07 A. Při změnách výstupní frekvence v provozu se již napájecí proud zřetelně nemění. Na výstupu se po přivedení napájecího napětí objeví maximální frekvence použitého VCO (~2600 MHz). Tato frekvence vykazuje nestabilitu, jelikož není řízena samotným syntezátorem.

4.9 Komunikace s PC

Po připojení se řadič FT232R standardně přihlásí a nainstalují se k němu dodávané ovladače. Automaticky je vytvořen i virtuální COM port. Servisní aplikace *FT Prog.* umožňuje snadné čtení a editaci parametrů uložených v EEPROM na čipu. Vzhledem k tomu, že jsou standardně komunikační výstupy, včetně použitých RTS, DTR a TxD, invertovány, je vhodné inverzi vypnout pro usnadnění pozdějšího programování řídicí aplikace. Další nastavení není zapotřebí měnit.

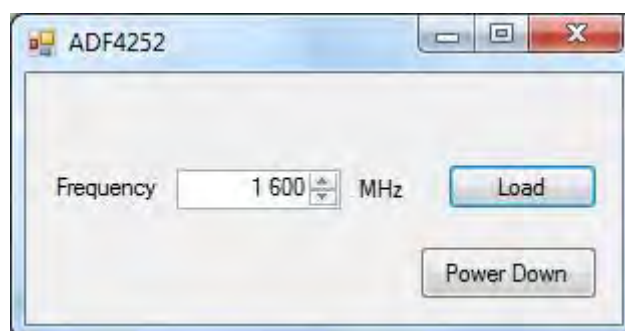
4.10 Návrh řídicí aplikace

Aplikace je postavena na programovacím jazyce Microsoft C#. Vzhledem ke zjednodušení díky FT232R, nejsou zapotřebí žádné dodatečné knihovny pro komunikaci. Logika programu je taková, že je napsána univerzální funkce pro zápis dat libovolného registru, která současně generuje při každém datovém bitu náběžnou hranu signálu CLK a po přenesení všech datových bitů generuje signál LE pro aplikaci dat v daném registru. Funkce je využívána dle potřeby při inicializaci aplikace a při uživatelem provedené změně nastavení.

4.10.1 Popis fungování aplikace

Kompletní výpis je v příloze 4. Aby nenastávaly problémy s identifikací COM portu na daném PC, je aplikace vázána na samotný FTDI čip dle jeho *Description* (FT232R USB UART).

Po spuštění aplikace dojde k ověření připojení syntezátoru. Pokud by nastal problém, objeví se chybové hlášení a dojde k zavření okna aplikace. V případě, že je syntezátor bez problému identifikován, provede se automaticky inicializace, která zapíše data do *Master Register*, *RF Control Register* a *RF R Divider Register*. Obsahy těchto registrů jsou blíže popsány v kapitole 4.6. Při změně frekvence uživatelem je přístupováno pouze do posledního registru *RF N Divider Register*. Okno aplikace je na obr. 77.



Obr. 77 Navržená aplikace pro ovládání PLL syntezátoru

Frekvenční rozsah syntezátoru je dán rozsahem použitého VCO, tedy $1600 \div 2500$ MHz. Krok je v navržené aplikaci možný po 1 MHz. Zadávání je ošetřeno proti zapsání jiné než číselné hodnoty. Pro nahrání nastavení do syntezátoru slouží tlačítko Load. Při jeho stisku dojde k převodu zadané frekvence na odpovídající hodnoty *RF N Divider Register*.

Desítky MHz se promítnou do osmi-bitové hodnoty INT a to přímým převodem na binární číslo (např. 1602 MHz \rightarrow $160_{10} = 1010000_2$). Jednotky MHz se promítnou do dvanácti-bitové hodnoty FRAC a to tak, že jsou roznásobeny číslem 50 (hodnota MOD zapsaná při inicializaci je 500) a převedeny na binární číslo (např. 1602 MHz \rightarrow $[2 \cdot 50]_{10} = 000001100100_2$). Uvedený převod je použit konkrétně zde v navržené aplikaci a vychází z nastavení syntezátoru, provedeném při inicializaci. Existuje mnoho různých nastavení, která lze přizpůsobit požadované aplikaci. Při využití online aplikace Analog Devices pro volbu nastavení registrů lze snadno tato nastavení určit.

Po převodu zadané frekvence je celý registr nahrán do syntezátoru a změna ihned aplikována. Na výstupu syntezátoru lze pozorovat nastavenou frekvenci. V době datového přenosu nejsou tlačítka Load a Power Down povolena.

Tlačítko Power Down slouží k softwarovému vypnutí syntezátoru. Přistupuje do *Master Register*, kde aktivuje příslušný bit a celý registr nahraje do syntezátoru. V době datového přenosu nejsou tlačítka povolena. Po vypnutí syntezátoru je tlačítko Load neaktivní a tlačítko Power Down změní svou funkci na Power Up pro opětovné zapnutí.

Komunikační port FTDI čipu je trvale otevřen od spuštění aplikace. K jeho uzavření dojde až při jejím ukončení.

4.11 Řízení pomocí Agilent VEE

Agilent VEE umožňuje import knihoven postavených na Microsoft .NET a volání jejich funkcí přímo v prostředí uživatelského programu. Ve vývojovém prostředí Microsoft Visual Studio pro jazyk C# lze snadno požadovanou knihovnu vytvořit s využitím kódu řídicí aplikace syntezátoru. Základní funkce knihovny sloužící pro generování signálů pro samotný zápis dat do syntezátoru je shodná.

4.11.1 Popis fungování knihovny pro Agilent VEE

Kompletní výpis je v příloze 5. Knihovna je stejně jako v případě řídicí aplikace vázána na *Description* (FT232R USB UART) čipu. Inicializace knihovny provede shodně zápis dat do *Master Register*, *RF Control Register* a *RF R Divider Register*. Dále knihovna obsahuje tři veřejné funkce, které jediné lze volat z prostředí Agilent VEE. Jsou to funkce *SetFrequency(f)*, *PowerDown()*, *PowerUp()*.

Princip funkce *SetFrequency(f)* je shodný se zadáváním frekvence v prostředí řídicí aplikace. Je očekávána celočíselná hodnota v rozsahu 1600 až 2500 MHz a to ve formátu integer či real. V případě, že bude rozsah překročen, vrátí funkce hlášení o chybě "error: out of range". Hlášení o chybě bude vráceno i v případě nepřipojení syntezátoru "error: USB device not found". V případě úspěšného provedení zápisu, vrátí naopak hlášení "OK". Na začátku funkce je nastavena čekací doba 1 s, která je zde z bezpečnostního důvodu, aby nedošlo ke kolizi se zápisem inicializačních dat, což by v prostředí VEE mohlo nastat, jelikož v uživatelské aplikaci může být řazena inicializace knihovny a následný zápis frekvence ihned za sebou. Na rozdíl od řídicí aplikace je po inicializaci komunikační port uzavřen, aby nevznikaly problémy s jeho otevřením při ladění uživatelského VEE programu. Z tohoto důvodu je na začátku volané funkce port otevřen a po jejím skončení opět uzavřen.

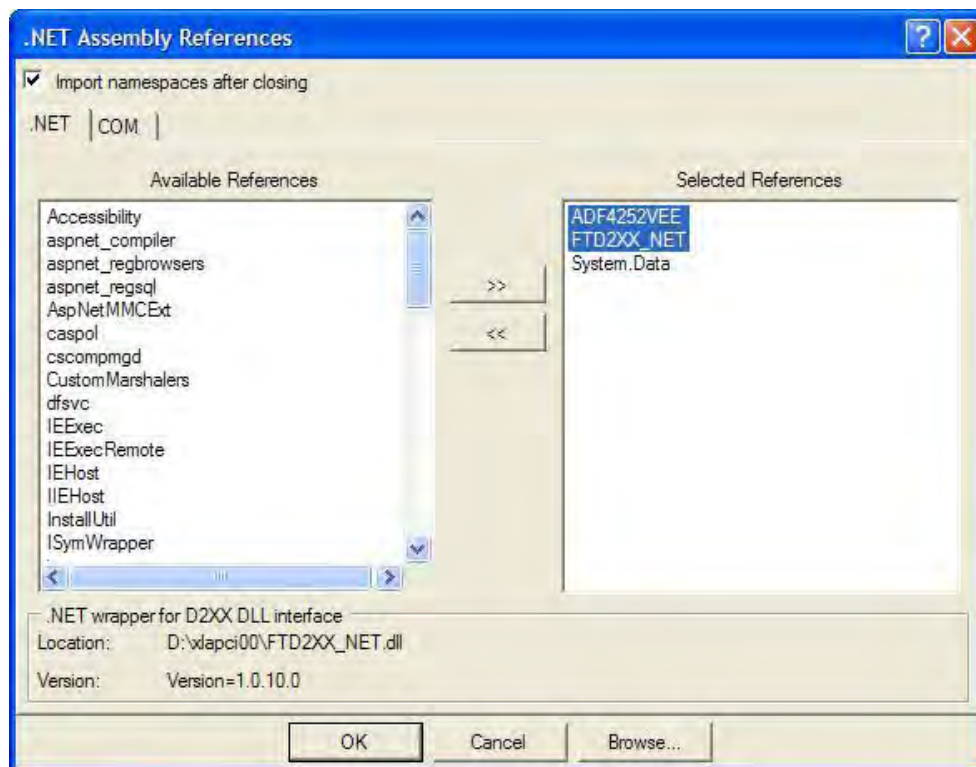
Funkce *PowerDown()* slouží k softwarovému vypnutí syntezátoru. Přistupuje do *Master Register*, kde aktivuje příslušný bit, celý registr nahraje do syntezátoru a v případě úspěšného provedení zápisu vrátí hlášení "OK". V případě nepřipojení syntezátoru je vráceno "error: USB device not found". Na začátku funkce je opět nastavena ochranná čekací doba 1 s. Komunikační port je otevřen pouze po dobu provádění funkce.

Funkce *PowerUp()* slouží k softwarovému zapnutí syntezátoru v případě, že byl dříve vypnut pomocí funkce *PowerDown()*. Přistupuje do *Master Register*, kde deaktivuje příslušný bit, celý registr nahraje do syntezátoru a v případě úspěšného provedení zápisu vrátí hlášení "OK". V případě nepřipojení syntezátoru je vráceno "error: USB device not found". Na začátku funkce je opět nastavena ochranná čekací doba 1 s. Komunikační port je otevřen pouze po dobu provádění funkce.

4.11.2 Návrh VEE programu pro řízení syntezátoru

Zde popsaná aplikace pro řízení navrženého PLL syntezátoru slouží jako vzorový příklad a bude základem laboratorní úlohy v příloze 6. Jejím cílem je automatické přeladování syntezátoru v zadaném rozsahu frekvencí a následné automatické měření skutečné výstupní frekvence a výkonové úrovně pomocí spektrálního analyzátoru R&S ZVL. Získané údaje budou aplikací dále zpracovány za účelem získání hodnoty průměrné frekvenční odchylky přes všechna měření a grafu závislosti výstupního výkonu na frekvenci.

Nejprve je potřeba provést import vytvořené knihovny pro řízení syntezátoru. Okno sloužící pro import je na obr. 78. Lze jej nalézt přes kontextové menu v „Device->.NET Assembly References...“.

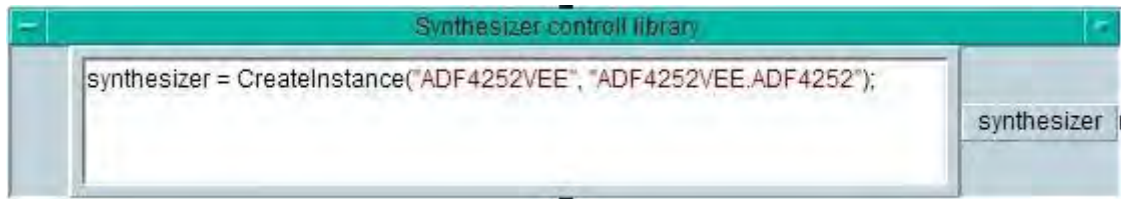


Obr. 78 Agilent VEE – okno sloužící k importu řídicí knihovny syntezátoru

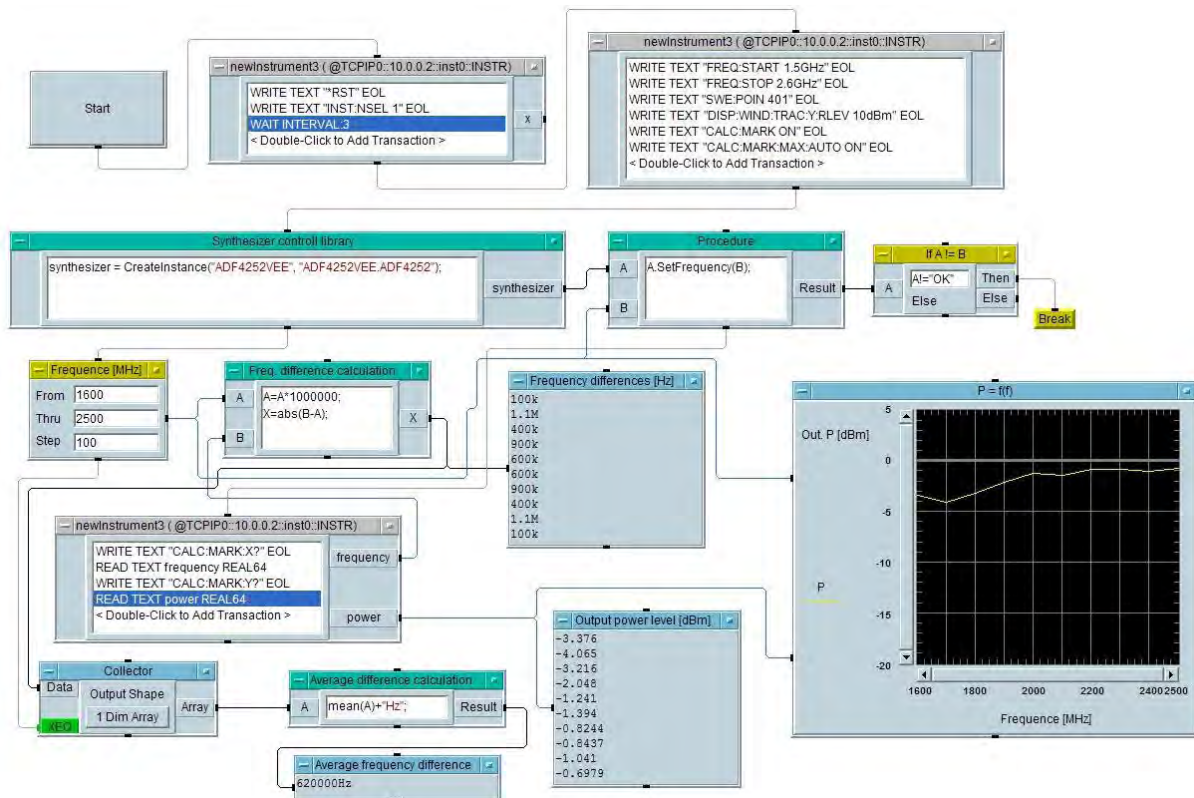
Inicializace knihovny v rámci uživatelského programu VEE se provádí uvnitř bloku *Formula* (*Synthesizer controll library*) - obr. 79. Na obr. 80 je pak kompletní podoba výsledného VEE programu, kde je jednoduše vidět způsob volání funkcí inicializované knihovny skrze připojený další blok *Formula* (*Procedure*). Frekvence je měněna ve *For*-cyklu (*Frequency*) v plném rozsahu 1600 ÷ 2500 MHz s krokem 100 MHz. Funkce knihovny vrací, v případě úspěšného provedení, hlášení "OK", které je vždy ověřováno a v případě, že by nebylo vráceno, cyklus se ukončí.

Další blok Formula (Freq. difference calculation) provádí odečet změřené a zadané frekvence. Z těchto výsledků je pomocí bloku Collector vytvořeno jednorozměrné pole, které je po naplnění statisticky vyhodnoceno určením průměrné hodnoty v Average difference calculation a výsledek je na závěr zobrazen v Average frequency difference.

Hodnoty změřeného výstupního výkonu syntezátoru jsou posílány přímo do grafu X vs Y ($P = f(f)$), kde jsou zobrazeny v závislosti na zadávané frekvenci. Zobrazení konkrétních hodnot zjištěných frekvenčních odchylek a změřeného výstupního výkonu zde slouží pouze pro názornost a v případě provádění velkého souboru měření by nebyla tato zobrazení příliš účelná.



Obr. 79 Agilent VEE – inicializace řídicí knihovny syntezátoru uvnitř bloku Formula



Obr. 80 Agilent VEE – navržená vzorová aplikace s řízením PLL syntezátoru

Pro všechna měření reálných výstupních hodnot je použit spektrální analyzátor R&S ZVL, který se dá řídit z prostředí Agilent VEE přes LAN připojení. Řízení se provádí v objektu DIRECT I/O skrze příkazy, které jsou k nalezení v operačním manuálu [17]. K dispozici je široké spektrum příkazů, umožňujících nastavování všech parametrů měření analyzátoru. Před zadáváním parametrů měření je vhodné provést restart do defaultního nastavení sloužícího jako pevný výchozí stav analyzátoru. Řazení dalších použitých příkazů je logické a vychází z jejich konkrétní funkce.

Příkazy použité ve vzorové aplikaci v zadávaném pořadí:

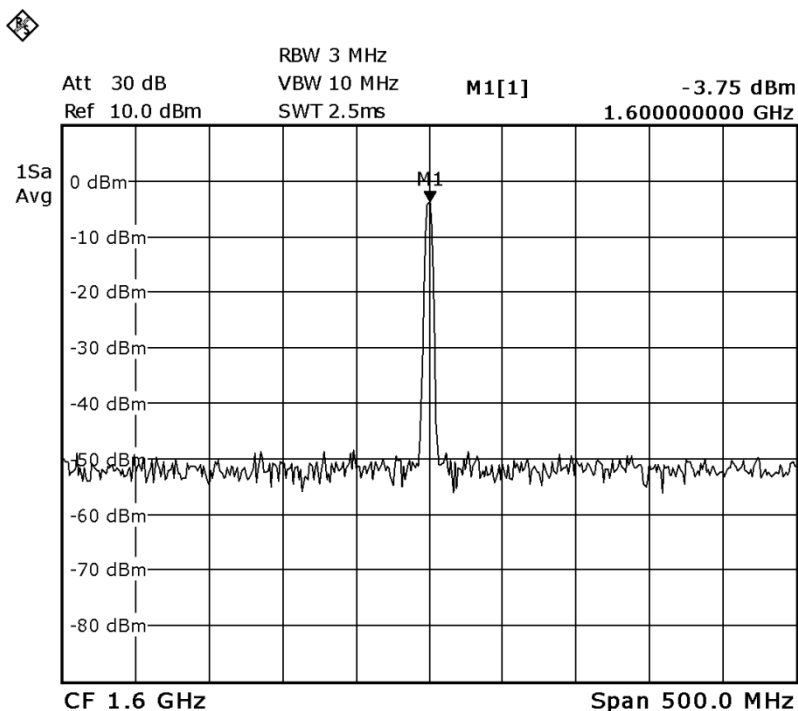
- *RST → provede restart analyzátoru do defaultního nastavení
- INST:NSEL 1 → přepne mód na spektrální analyzátor
- FREQ:START 1.5GHz → nastaví počáteční frekvenci měření na 1,5 GHz
- FREQ:STOP 2.6GHz → nastaví konečnou frekvenci měření na 2,6 GHz
- SWE:POIN 401 → nastaví počet bodů měření na 401
- DISP:WIND:TRAC:Y:RLEV 10dBm → nastaví referenční výkonovou úroveň na 10 dBm
- CALC:MARK ON → zapne marker 1
- CALC:MARK:MAX:AUTO ON → zapne funkci markeru pro kontinuální sledování maximálního výkonu v měřeném spektru
- CALC:MARK:X? → požadavek na vrácení frekvenční pozice markeru – tj. změření frekvence měřeného signálu ve spektru
- CALC:MARK:Y? → požadavek na vrácení výkonové pozice markeru – tj. změření výkonové úrovně měřeného signálu ve spektru

Po zadání příkazu pro přepnutí do módu spektrálního analyzátoru je vložen čekací interval 3 s a to z důvodu, že se jedná o komplikovanější operaci a před odesláním dalšího příkazu je potřeba zajistit ochranu proti možné kolizi.

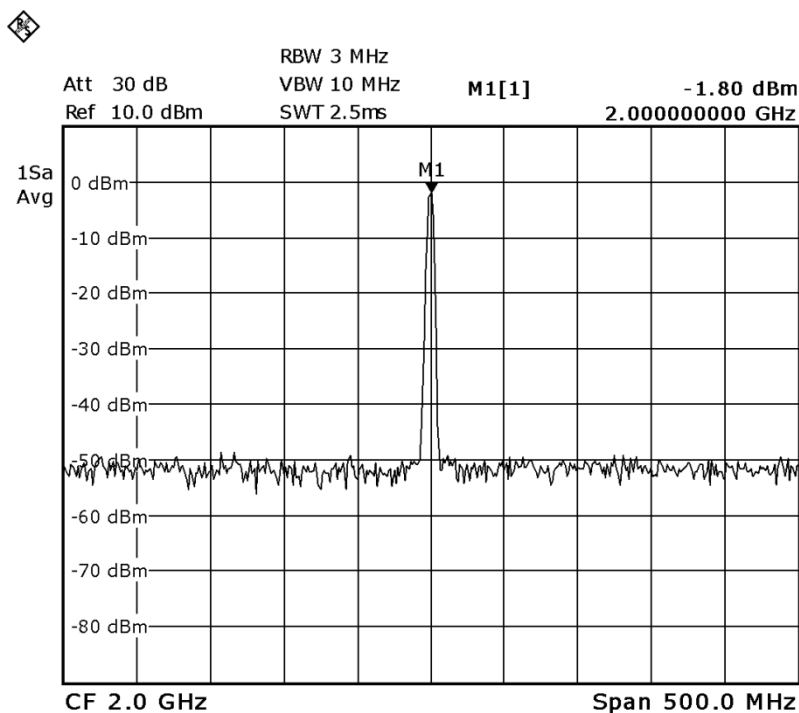
4.12 Měření výstupní frekvence

Na následujících obrázcích 81 ÷ 83 jsou spektrálním analyzátozem Rohde&Schwarz ZVL změřeny výstupní frekvence. Proměřeny jsou frekvence 1600 MHz, 2000 MHz a 2500 MHz.

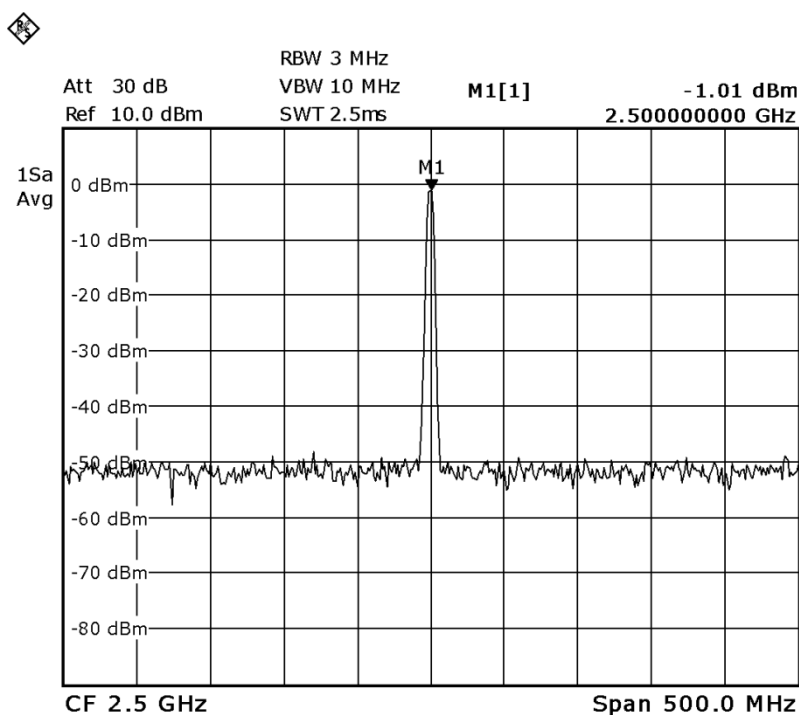
V každém spektru je výstupní signál změřen pomocí markeru pro přesné určení jeho výkonové úrovně a frekvenční odchylky.



Obr. 81 Frekvence 1600 MHz – PLL syntezátor



Obr. 82 Frekvence 2000 MHz – PLL syntezátor



Obr. 83 Frekvence 2500 MHz – PLL syntezátor

Výstupní frekvence je ve všech měřených případech velmi přesná bez viditelné odchylky od frekvence nastavené. Hladina šumu je pod úrovní -50 dBm. Výstupní výkon se pohybuje od -3,75 dBm (1600 MHz) do -1,01 dBm (2500 MHz).

Syntezátor byl v průběhu různých měření ponechán v provozu v řádu hodin a nebyl zjištěn žádný problém způsobený delší činností. Nebyl pozorován nadměrný ohřev částí syntezátoru ani jiné vlivy, které by mohly mít vliv na změny vlastností v průběhu delšího provozu.

5 Závěr

V práci byly rozebrány jednotlivé typy frekvenčních syntezátorů a rozděleny podle daných kritérií včetně popisu rozdílů mezi jejich činností. Dále se práce zaměřuje na návrh dvou syntezátorů a to s přímou digitální syntézou (DDS) a s fázovým závěsem (PLL). Jsou podrobně popsány základní stavební obvody obou syntezátorů. Jedná se konkrétně o obvod AD9958 (DDS) a ADF4252 (PLL). V části návrhu DDS syntezátoru je popsán obvod pro komunikaci s PC pomocí sběrnice USB a to USB 2.0 řadič CYPRESS EZ-USB FX2. V části návrhu PLL syntezátoru je popsán USB řadič FTDI FT232R.

Návrh DDS i PLL syntezátoru je kompletně popsán a to po jednotlivých částech výsledného obvodu. Jsou navrženy a popsány desky plošného spoje a jsou přiloženy fotografie jejich realizací. U obou syntezátorů je funkční komunikace s PC pomocí USB rozhraní a jsou navrženy aplikace pro jejich řízení z prostředí Windows.

V případě PLL syntezátoru je realizován i způsob jeho řízení v prostředí Agilent VEE. Řízení DDS syntezátoru stejným způsobem z prostředí Agilent VEE vyžaduje z důvodu jiného USB řadiče přepracování dodávaných knihoven do novějšího programovacího jazyka, což je velmi komplikované a proveditelné jen se značnou znalostí programování v obou jazycích.

Výstupy obou syntezátorů jsou proměřeny pomocí spektrálního analyzátoru R&S ZVL a v příslušných kapitolách popsány a vyhodnoceny.

Na základě možnosti řízení PLL syntezátoru z prostředí Agilent VEE je navržena VEE uživatelská aplikace, která je základem pro, v příloze 6, přiloženou laboratorní úlohu.

6 LITERATURA

- [1] HANUS, S.; SVAČINA, J. *Vysokofrekvenční a mikrovlnná technika*. Elektronické skriptum. Brno: FEKT VUT v Brně, 2002.
- [2] ANALOG DEVICES. *AD9958* [online]. Datasheet [cit. 2008]. Dostupný z WWW: www.analog.com.
- [3] CYPRESS SEMICONDUCTOR *CY7C68013* [online]. Datasheet [cit. prosinec 2002]. Dostupný z WWW: www.cypress.com.
- [4] ANALOG DEVICES. *ADF4252* [online]. Datasheet [cit. 2003]. Dostupný z WWW: www.analog.com.
- [5] MINI-CIRCUITS. *ADTT1-1* [online]. Datasheet. Dostupný z WWW: www.minicircuits.com.
- [6] ANALOG DEVICES. *AD9958_EB* [online]. Datasheet [cit. 2006]. Dostupný z WWW: www.analog.com.
- [7] ANALOG DEVICES. *ADF4252EB1* [online]. Datasheet [cit. 1999]. Dostupný z WWW: www.analog.com.
- [8] MINI-CIRCUITS. *ROS-2500+* [online]. Datasheet. Dostupný z WWW: www.minicircuits.com.
- [9] ANALOG DEVICES. *OP27* [online]. Datasheet [cit. 2006]. Dostupný z WWW: www.analog.com.
- [10] ON SEMICONDUCTOR. *LM317* [online]. Datasheet [cit. 2008]. Dostupný z WWW: onsemi.com.
- [11] MACKŮ, R. *Návrh generátoru využívající přímou číslicovou syntézu (DDS)*. Diplomová práce. Brno: FEKT VUT v Brně, 2007.
- [12] DUFEK, I. *Automatický anténní analyzátor*. Diplomová práce. Brno: FEKT VUT v Brně, 2010.
- [13] MICROCHIP. *24LC00* [online]. Datasheet [cit. 2005]. Dostupný z WWW: www.microchip.com.
- [14] PHILIPS. *74LVC14A* [online]. Datasheet [cit. 2005]. Dostupný z WWW: www.philips.com.
- [15] MAXIM. *MAX* [online]. Datasheet [cit. 2004]. Dostupný z WWW: www.maxim-ic.com.
- [16] FTDI Chip. *FT232R* [online]. Datasheet [cit. 2010]. Dostupný z WWW: www.ftdichip.com.
- [17] R&S ZVL *Vector Network Analyzer* [online]. Operating Manual [cit. 2009]. Dostupný z WWW: www2.rohde-schwarz.com.

7 SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

<i>PLL</i>	Phase Locked Loop
<i>DDS</i>	Direct Digital Synthesis
<i>SMD</i>	Surface Mounted Device
<i>SMA</i>	Typ vysokofrekvenčního konektoru
<i>LSB</i>	Last Significant Bit
<i>MSB</i>	Most Significant bit
f_{GH}	Výstupní signál generátoru harmonických
f_0	Frekvence oscilátoru
f_h	Frekvence řízeného oscilátoru
f_{mf}	Mezifrekvenční signál
$f_{výst}$	Výstupní signál syntezátoru
x_k	Řád harmonické
f_{VCO}	Výstupní signál VCO
f_{TEST}	Signál z výstupu programovatelného děliče
f_{REF}	Výstupní signál referenčního oscilátoru
f_Q	Kmitočet referenčního oscilátoru
<i>N</i>	Dělicí poměr programovatelného děliče
<i>M</i>	Dělicí poměr děliče kmitočtu
<i>P</i>	Dělicí poměr rychlého předděliče kmitočtu
A_{MAX}	Maximální stav čítače swallow
N_{MAX}	Maximální dělicí poměr při <i>n</i> -bitovém slově
f_{osc}	Výstupní frekvence syntezátoru
R_{SET}	Externí rezistor pro nastavení proudového rozsahu AD9958
f_{OSC}	Frekvence krystalu pro AD9958
f_{SYS}	Systémová frekvence AD9958
<i>M</i>	PLL násobič AD9958
I_{OUT}	Výstupní proud DAC převodníku AD9958
$I_{\overline{OUT}}$	Komplementární výstupní proud DAC převodníku AD9958
V_{out}	Výstupní napětí stabilizátoru LM317BD2T
I_{Adj}	<i>Adjust</i> proud stabilizátoru LM317BD2T (< 100 μ A)
<i>VID</i>	Vendor ID USB řadiče
<i>PID</i>	Product ID USB řadiče
<i>DP</i>	Filtr Dolní Propust
REF_{IN}	Kmitočet referenčního oscilátoru PLL syntezátoru
<i>D</i>	Hodnota násobiče PLL syntezátoru
<i>R</i>	Hodnota děliče PLL syntezátoru
<i>UART</i>	Universal Asynchronous Receiver/Transmitter
<i>VEE</i>	Visual Engineering Environment
<i>USB</i>	Universal Serial Bus

8 Seznam příloh

Příloha 1: Schéma DDS syntežátoru

Příloha 2: Schéma PLL syntežátoru

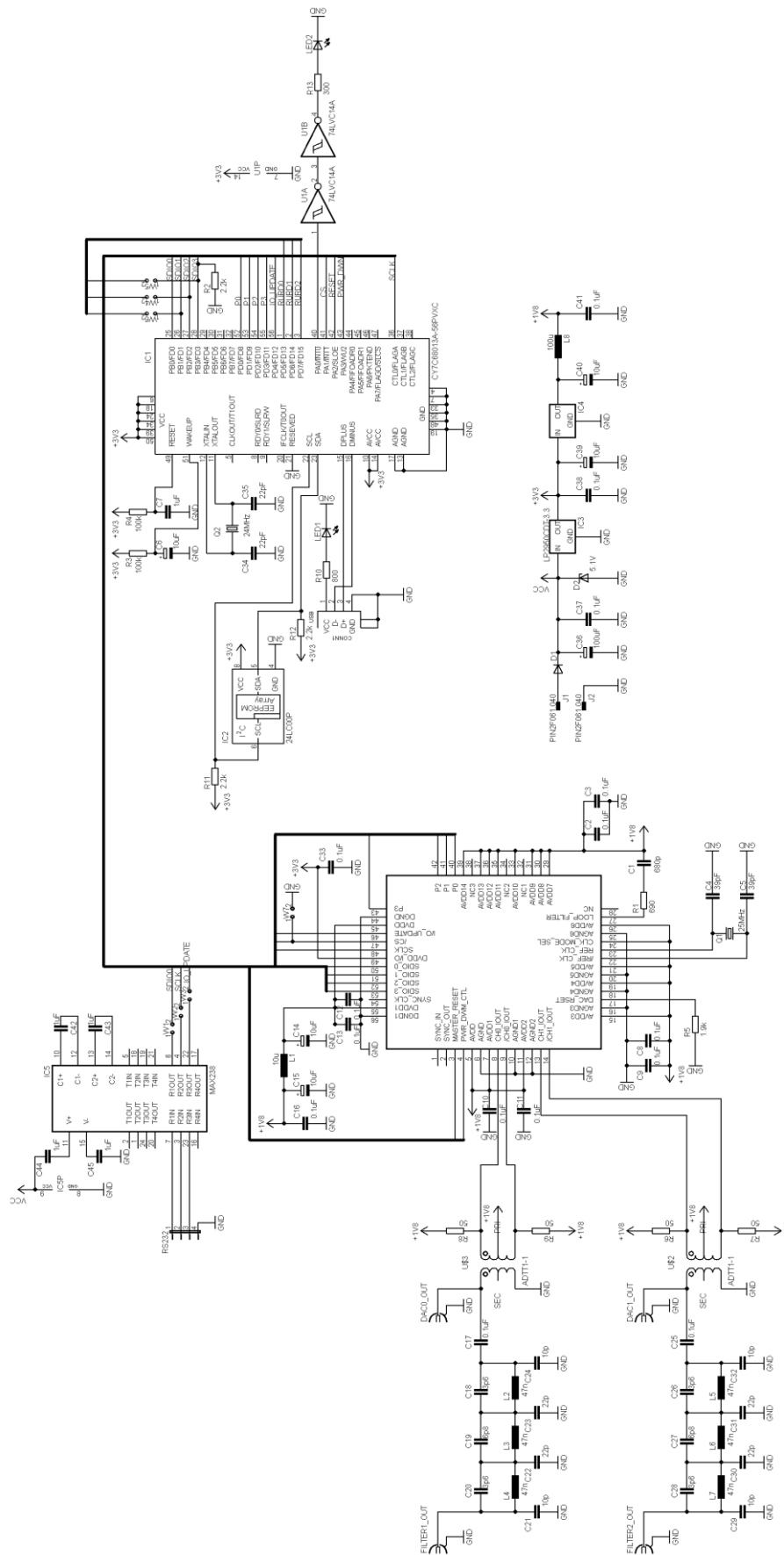
Příloha 3: Kód řídicí aplikace DDS syntežátoru

Příloha 4: Kód řídicí aplikace PLL syntežátoru

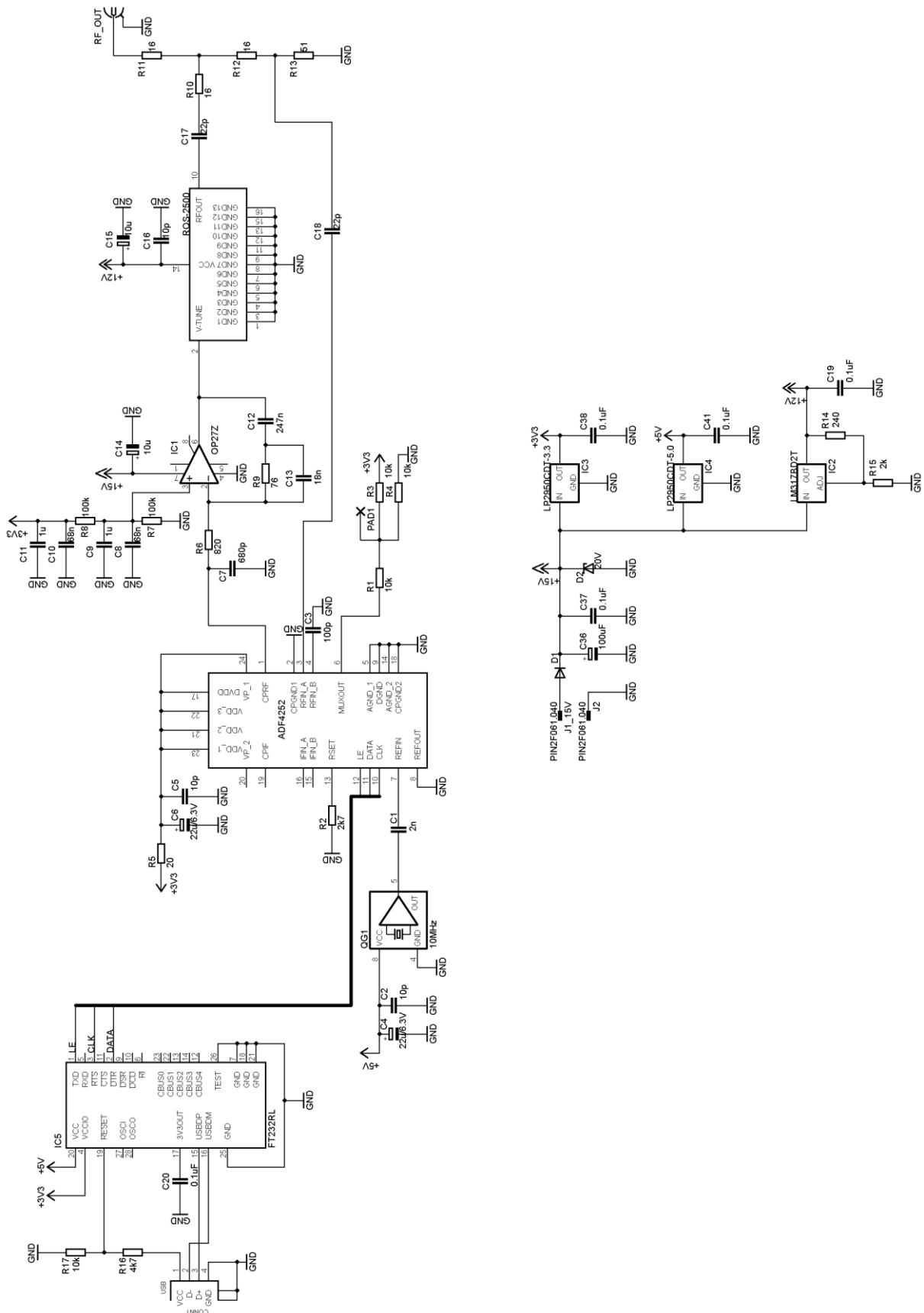
Příloha 5: Kód knihovny pro řízení PLL syntežátoru z prostředí Agilent VEE

Příloha 6: Laboratorní úloha: Měření na PLL syntežátoru

Příloha 1: Schéma DDS syntezátoru



Příloha 2: Schéma PLL syntezátoru



Příloha 3: Kód řídicí aplikace DDS syntezátoru

```
'*****
'* Main Module
'*****

'A globally accessible usb class for the EZ-USB FX2
Public clsEZUSBDev01 As clsEZUSB
'A globally accessible evaluation board class
Public EvBd As cAD9959EvalBd

'Define public constants for the tuning word sizes
Public Const OFTWBits = 32 'Output frequency tuning word

Sub Main()
    Dim sRegVal As String

    'Create a new instance of the usb class
    Set clsEZUSBDev01 = New clsEZUSB

    'Initailize the USB
    clsEZUSBDev01.VendorID = &H456    'Analog Devices Inc.
    clsEZUSBDev01.ProductID = &HEE07 'AD9958 Evaluation Board

    Form1.AddStatLine " Checking for USB Drivers..."
    If clsEZUSBDev01.USBDriverIsInstalled() Then
        Form1.AddStatLine " USB Driver Found."

        'Initialize the device list
        clsEZUSBDev01.GetDevices

        If clsEZUSBDev01.DeviceCount = 1 Then
            'Display the status on the splash screen
            Call clsEZUSBDev01.SelectDevice(0, True)
            Form1.AddStatLine " Using Evaluation Board - " &
                clsEZUSBDev01.DriverName & "."
        Else
            'Display an error message on the splash screen
            Form1.SetStatusTextColor vbRed
            Call clsEZUSBDev01.SelectDevice(0, True)
            Form1.AddStatLine "* Error: Evaluation Board Not Found!!!"
        End If
    Else
        'Display an error message on the splash screen
        Form1.SetStatusTextColor vbRed
        Form1.AddStatLine "* Error: USB Driver Not Found."
    End If

    'Set EvBd variables
    Set EvBd = New cAD9959EvalBd
    EvBd.AutoIOUpdate = True

    Form1.AddStatLine ""
    Form1.AddStatLine "Initializing the Evaluation Board..."
    'Initialize the evaluation board
    EvBd.InitIOBufs
    'Initialize the I/O lines make sure that the RURD_2 line
    'is low because if they have it jumpered over to the dut
```

```

'and it is high you may not be able to communicate with the
'DUT because it comes up as I/O Reset by default
If EvBd.RURD_2 = abvHigh Then
    EvBd.RURD_2 = abvLow
    EvBd.USBWritePortBuffVal fx2_PortD
End If

Form1.AddStatLine "Resetting the Eval Board and Software..."
'Reset the Evaluation board
EvBd.ResetDUT
Form1.AddStatLine "Software is up and running..."

Form1.txtSysClk.Text = Format(500, "0.000")
'Multiplier set to value "20" and VCO gain Enable
sRegVal = "110100000000000000000000"
EvBd.SetRegMapValue rm9959_NewRegMapVals, 1, sRegVal

'Load new register maps into device
EvBd.AutoIOUpdate = False
'Load the data into the DUT
EvBd.USBSerialLoad 0, EvBd.GetRegMapValue(rm9959_NewRegMapVals, 0), 1
'Load the data into the DUT
EvBd.USBSerialLoad 1, EvBd.GetRegMapValue(rm9959_NewRegMapVals, 1)
'Load the last control register and do an IO Update
EvBd.USBSerialLoad 2, EvBd.GetRegMapValue(rm9959_NewRegMapVals, 2)

EvBd.StrobeIOUpdate

End Sub

```

```

'*****
'* Form1
'*****
Private Sub cmdLoad_Click()
    Dim cntr As Integer
    Dim sChRegVal As String

    EvBd.AutoIOUpdate = False

    'Disable the Load button
    cmdLoad.Enabled = False

    For cntr = 2 To 3
        'Select the channel
        EvBd.SelectChannel cntr
        sChRegVal = EvBd.GetChRegValue(rm9959_NewRegMapVals, 4, cntr)
        'Load the value
        EvBd.USBSerialLoad 4, sChRegVal
    Next cntr

    EvBd.StrobeIOUpdate
    cmdLoad.Enabled = True
End Sub

Private Sub Form_Load()

    'Initialize of Main procedure
    mMain.Main

End Sub

'Adds a line to the status window
Public Sub AddStatLine(ByVal strLine As String)
    With txtStatus
        If .Text = "" Then
            .Text = strLine
        Else
            .SelStart = Len(.Text)
            .SelText = vbNewLine & strLine
        End If
        'Refresh the textbox
        .Refresh
    End With
End Sub

'Sets the foreground color of the text in the status text box
Public Sub SetStatusTextColor(ByVal fColor As Long)
    With txtStatus
        .ForeColor = fColor
    End With
End Sub

Private Sub txtChOutFreq0_KeyPress(Index As Integer, KeyAscii As Integer)
    Dim cncl As Boolean

    Select Case KeyAscii
        Case 13: 'Enter
            KeyAscii = 0
            'Validate frequency
            txtChOutFreq0_Validate Index, cncl
    End Select
End Sub

```

```

        Case 8, 44, 48 To 57 'Numeric Keys, Backspace & Decimal separator
            KeyAscii = KeyAscii
        Case Else: 'Not a passable key
            KeyAscii = 0
    End Select
End Sub

Private Sub txtChOutFreq0_Validate(Index As Integer, Cancel As Boolean)
    Dim OutFreqval As Double
    Dim TWDecVal As Variant
    Dim BinVal As String
    Dim SysClk As Double

    'Get the current system clock
    SysClk = CDb1("500")

    'Initialize as a decimal value
    TWDecVal = CDec(0)

    'Correct number check
    If Not IsNumeric(txtChOutFreq0(Index).Text) Then
        txtChOutFreq0(Index).Text = "0"
    End If

    'Convert to a double
    OutFreqval = CDb1(txtChOutFreq0(Index).Text)

    If OutFreqval >= SysClk Then
        MsgBox "The output frequency can not exceed the system clock.",
            vbApplicationModal + vbInformation + vbOKOnly, "Output Frequency to
            High!!!"
        'Calculate max frequency given by System clock
        OutFreqval = DecTW2CarrierFreq((2 ^ OFTWBits) - 1, SysClk,
            OFTWBits)
    End If

    'Calculate the decimal tuning word ... Round value
    TWDecVal = CalculateDecTuningWord(OutFreqval, SysClk, OFTWBits)
    'Convert it to a double
    OutFreqval = CDb1(DecTW2CarrierFreq(TWDecVal, SysClk, OFTWBits))

    'Format and display the output frequency
    txtChOutFreq0(Index).Text = FormatOutFreq(OutFreqval, OFTWBits, SysClk)

    'Update the tag property
    txtChOutFreq0(Index).Tag = CStr(TWDecVal)

    'Update the register map
    BinVal = cbaseDec2Bin(txtChOutFreq0(Index).Tag, OFTWBits)
    EvBd.SetRegMapValue rm9959_NewRegMapVals, 4, BinVal,
    EvBd.GetChMaskVal(Index)

    cmdLoad.Enabled = True
    cmdLoad.SetFocus
End Sub

Private Sub txtStatus_Change()
    'Frame1_Click
End Sub

```

Příloha 4: Kód řídicí aplikace PLL syntezátoru

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
using FTD2XX_NET;

namespace AD4252
{
    public partial class Form1 : Form
    {
        private FTDI port = new FTDI();
        private uint count = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void InicializeSynthesizer()
        {
            int[] master = { 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1 };
            int[] rfControl = { 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
                               1, 0 };
            int[] rfrDivider = { 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1,
                                 0, 1, 0, 0, 0, 0, 1 };

            var status = port.OpenByDescription("FT232R USB UART");
            if (status == FTDI.FT_STATUS.FT_OK)
            {
                WriteRegister(master);
                WriteRegister(rfControl);
                WriteRegister(rfrDivider);
            }
            else
            {
                MessageBox.Show("USB device not found");
                Close();
            }
        }

        private void SetFrequency()
        {
            int[] rfNDivider = new int[24];
            int frequency = (int)numericUpDown1.Value / 10;
            int appendix = ((int)numericUpDown1.Value % 10) * 50;

            var a = IntToBoolEight(frequency);
            var b = IntToBoolTwelve(appendix);

            for (int i = 1; i < 9; i++)
            {
                rfNDivider[i] = a[i - 1];
            }
            for (int i = 9; i < 21; i++)

```

```

        {
            rfNDivider[i] = b[i - 9];
        }

WriteRegister(rfNDivider);
}

private int[] IntToBoolEight(int number)
{
    int[] array = new int[8];
    for (int i = 7; i >= 0; i--)
    {
        array[i] = number % 2;
        number /= 2;
    }
    return array;
}

private int[] IntToBoolTwelve(int number)
{
    int[] array = new int[12];
    for (int i = 11; i >= 0; i--)
    {
        array[i] = number % 2;
        number /= 2;
    }
    return array;
}

private void button1_Click(object sender, EventArgs e)
{
    button1.Enabled = false;
    button2.Enabled = false;
    SetFrequency();
    button1.Enabled = true;
    button2.Enabled = true;
}

private void WriteRegister(int[] data)
{
    for (int i = 0; i < data.Length; i++)
    {
        if(data[i] == 0)
        {
            port.SetDTR(false);
        }
        else
        {
            port.SetDTR(true);
        }

        Thread.Sleep(1);
        port.SetRTS(true);
        Thread.Sleep(1);
        port.SetRTS(false);
    }
    port.SetDTR(false);
    byte[] a = {0x00};
    uint numBytesWritten = 0;
    port.Write(a, 1, ref numBytesWritten);
    Thread.Sleep(200);
}

```

```

}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    port.Close();
}

private void Form1_Load(object sender, EventArgs e)
{
    InicializeSynthesizer();
}

private void button2_Click(object sender, EventArgs e)
{
    int[] master = { 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1 };
    if (count % 2 == 0)
    {
        button1.Enabled = false;
        button2.Enabled = false;
        WriteRegister(master);
        button2.Text = "Power Up";
        button2.Enabled = true;
    }
    else
    {
        master[5] = 0;
        button2.Enabled = false;
        WriteRegister(master);
        button1.Enabled = true;
        button2.Text = "Power Down";
        button2.Enabled = true;
    }
    count++;
}
}
}

```

Příloha 5: Kód knihovny pro řízení PLL syntezátoru z prostředí Agilent VEE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using FTD2XX_NET;
using System.Threading;
using System.Diagnostics;

namespace ADF4252VEE
{
    public class ADF4252
    {
        private FTDI port = new FTDI();

        public ADF4252()
        {
            InicializeSynthesizer();
        }

        private void InicializeSynthesizer()
        {
            int[] master = { 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1 };
            int[] rfControl = { 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
                               1, 0 };
            int[] rfRDivider = { 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1,
                                 0, 1, 0, 0, 0, 0, 0, 1 };

            port.OpenByDescription("FT232R USB UART");
            WriteRegister(master);
            WriteRegister(rfControl);
            WriteRegister(rfRDivider);
            port.Close();
        }

        public string SetFrequency(int frequency)
        {
            Thread.Sleep(1000);
            var status = port.OpenByDescription("FT232R USB UART");
            if (status == FTDI.FT_STATUS.FT_OK)
            {
                if (frequency > 2500 || frequency < 1600)
                {
                    port.Close();
                    return "error: out of range";
                }

                int[] rfNDivider = new int[24];
                int appendix = (frequency % 10) * 50;

                frequency = frequency / 10;

                var a = IntToBoolEight(frequency);
                var b = IntToBoolTwelve(appendix);
                for (int i = 1; i < 9; i++)
                {
                    rfNDivider[i] = a[i - 1];
                }

                for (int i = 9; i < 21; i++)
```

```

        {
            rfNDivider[i] = b[i - 9];
        }

        WriteRegister(rfNDivider);
        port.Close();
        return "OK";
    }
    else
    {
        return "error: USB device not found";
    }
}

public string SetFrequency(double frequency)
{
    return SetFrequency((int)frequency);
}

public string PowerDown()
{
    Thread.Sleep(1000);
    var status = port.OpenByDescription("FT232R USB UART");
    if (status == FTDI.FT_STATUS.FT_OK)
    {
        int[] master = { 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1 };
        WriteRegister(master);
        port.Close();
        return "OK";
    }
    else
    {
        return "error: USB device not found";
    }
}

public string PowerUp()
{
    Thread.Sleep(1000);
    var status = port.OpenByDescription("FT232R USB UART");
    if (status == FTDI.FT_STATUS.FT_OK)
    {
        int[] master = { 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1 };
        WriteRegister(master);
        port.Close();
        return "OK";
    }
    else
    {
        return "error: USB device not found";
    }
}

private int[] IntToBoolEight(int number)
{
    int[] array = new int[8];
    for (int i = 7; i >= 0; i--)
    {
        array[i] = number % 2;
        number /= 2;
    }
}

```

```

        return array;
    }

private int[] IntToBoolTwelve(int number)
{
    int[] array = new int[12];
    for (int i = 11; i >= 0; i--)
    {
        array[i] = number % 2;
        number /= 2;
    }
    return array;
}

private void WriteRegister(int[] data)
{
    for (int i = 0; i < data.Length; i++)
    {
        if (data[i] == 0)
        {
            port.SetDTR(false);
        }
        else
        {
            port.SetDTR(true);
        }

        Thread.Sleep(1);
        port.SetRTS(true);
        Thread.Sleep(1);
        port.SetRTS(false);
    }
    port.SetDTR(false);
    byte[] a = { 0x00 };
    uint numBytesWritten = 0;
    port.Write(a, 1, ref numBytesWritten);
    Thread.Sleep(200);
}
}
}

```

Příloha 6: Laboratorní úloha: Měření na PLL syntezátoru

Cílem laboratorní úlohy je seznámení se syntezátorem typu PLL, jeho řízení z prostředí Agilent VEE, automatizované měření výstupu pomocí spektrálního analyzátoru Rohde&Schwarz ZVL a srovnání změřených hodnot s nastavenými.

Zadání laboratorní úlohy

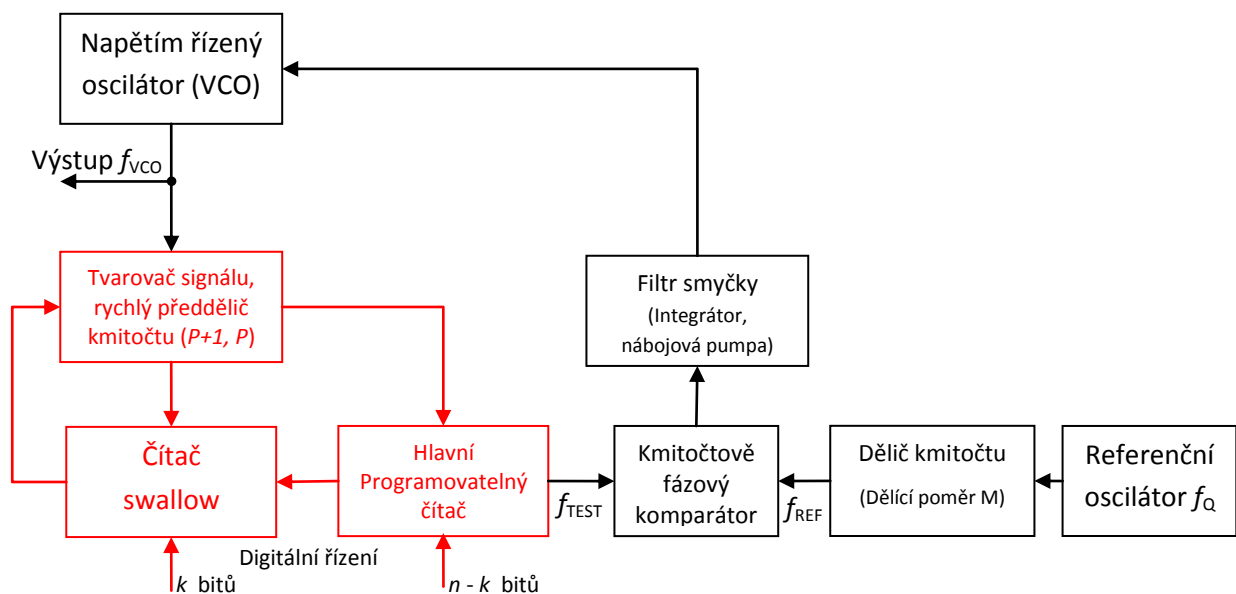
1. Seznamte se s principem PLL syntezátoru a způsobem jeho řízení z prostředí Agilent VEE.
2. Navrhněte program pro automatické přeladování syntezátoru a současně automatické měření jeho výstupu pomocí spektrálního analyzátoru.

Teoretický úvod

Kmitočtovými syntezátory jsou zdroje signálů (obvykle sinusového nebo pravouhlého průběhu) přesných kmitočtů, které mohou nabývat diskretních hodnot v určitém rozsahu. Na rozdíl od běžných oscilátorů nelze výstupní kmitočet měnit plynule, ale po obvykle ekvidistantních krocích.

PLL (Phase-Locked Loop) syntezátor s nepřímou syntézou využívá zpětné vazby realizované fázovým závěsem. Mezi základní stavební prvky patří směšovače, násobiče, děliče kmitočtu, napětím řízený oscilátor (VCO), programovatelný dělič kmitočtu, kmitočtový fázový detektor, atd.

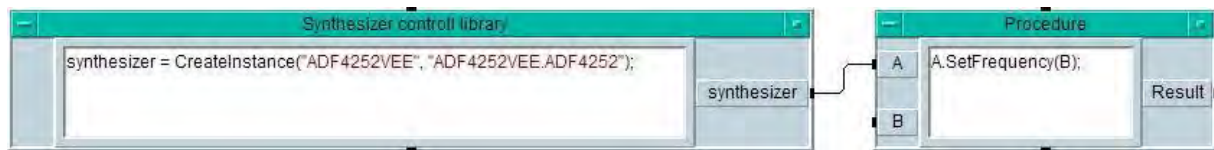
Dnes obvyklý je typ syntezátoru s řízeným předděličem, jehož blokové schéma je na obr. 1. Většina prvků je běžně integrována do jednoho čipu. Výjimkou bývá referenční oscilátor a VCO, který se volí v závislosti na dané aplikaci.



Obr. 1 PLL syntezátor s řízeným předděličem

V laboratorním přípravku je použit syntezátor Analog Devices ADF4252 a VCO s nastavitelným rozsahem $1600 \div 2500$ MHz. Řízení se provádí přes PC z prostředí Agilent VEE a to pomocí importované .NET knihovny. Knihovna nabízí uživateli funkci pro nastavení frekvence syntezátoru, softwarové vypnutí a opětovné zapnutí syntezátoru. Její inicializaci je

nutno provádět v bloku `FORMULA` a to způsobem, jak je uvedeno na obr. 2. Na stejném obrázku je vidět i postup při následném volání funkcí knihovny.



Obr. 2 Inicializace knihovny a následné volání její funkce

Funkce knihovny vrací hlášení o úspěchu provedené operace. V případě, že je operace úspěšná, je vráceno "OK". Hlášení o chybě jsou vrácena při zadání frekvence mimo povolený rozsah (1600 ÷ 2500 MHz) nebo při odpojeném syntežátoru.

Měření výstupu syntežátoru se provádí pomocí spektrálního analyzátoru Rohde&Schwarz ZVL, který se přes rozhraní LAN ovládá taktéž z prostředí Agilent VEE pomocí příkazů, které jsou všechny uvedeny v operačním manuálu. Základní potřebné příkazy pro tuto laboratorní úlohu jsou uvedeny dále v postupu měření.

Při měření vás bude zajímat průměrná frekvenční odchylka změřené a nastavené hodnoty přes zvolené spektrum měření a závislost výstupního výkonu na frekvenci. Závislost výstupního výkonu je dána čistě vlastnostmi použitého VCO a ideálně by měla být konstantní.

Postup měření

2) Program by měl automaticky přeladovat syntežátor v plném rozsahu frekvencí 1600 ÷ 2500 MHz. Pro toto přeladování použijte některý z nabízených cyklů. Inicializace řídicí knihovny syntežátoru a následné volání funkcí je znázorněno v teoretickém rozboru. Frekvenci zadávejte do funkce v celých MHz (bez jednotky). Úspěšné provedení volané funkce lze kontrolovat na základě vrácení příznaku "OK".

Pomocí analyzátoru Rohde&Schwarz ZVL změřte vždy výstupní frekvenci a výkonovou úroveň syntežátoru. Změřené hodnoty zpracujte tak, aby byla získána průměrná absolutní frekvenční odchylka přes všechna měření a graf závislosti výstupního výkonu na nastavené frekvenci. K řízení analyzátoru použijte následující příkazy zadávané do objektu DIRECT I/O:

- `*RST` → provede restart analyzátoru do defaultního nastavení
- `INST:NSEL 1` → přepne mód na spektrální analyzátor
- `FREQ:START xGHz` → nastaví počáteční frekvenci měření na x GHz
- `FREQ:STOP xGHz` → nastaví konečnou frekvenci měření na x GHz
- `SWE:POIN 401` → nastaví počet bodů měření na 401
- `DISP:WIND:TRAC:Y:RLEV xdBm` → nastaví referenční výkonovou úroveň na x dBm
- `CALC:MARK ON` → zapne marker 1
- `CALC:MARK:MAX:AUTO ON` → zapne funkci markeru pro kontinuální sledování maximálního výkonu v měřeném spektru
- `CALC:MARK:X?` → požadavek na vrácení frekvenční pozice markeru – tj. změření frekvence měřeného signálu ve spektru
- `CALC:MARK:Y?` → požadavek na vrácení výkonové pozice markeru – tj. změření výkonové úrovně měřeného signálu ve spektru

Po přepnutí módu analyzátoru je vhodné vložit čekací interval, protože se jedná o náročnější operaci a zamezí se tak možnosti problému kolize s následujícím příkazem.

Pro nalezení průměrné frekvenční odchylky je vhodné použít funkci `mean(x)` v objektu `FORMULA`. Do funkce musí vstupovat pole hodnot složené ze zjištěných odchylek frekvencí. K vytvoření pole použijte objekt `Collector`. Na vstup `XEQ` se přivádí řídicí signál inicializující konec příjmu vstupních dat. Po příchodu signálu je vytvořené pole odesláno v požadovaném tvaru na výstup.

Pro zobrazení grafu závislosti výstupního výkonu na frekvenci použijte objekt `X vs Y Plot`.

Požadovaný program může mít následující vývojový diagram:

1. Restart analyzátoru a jeho přepnutí do režimu spektrálního analyzátoru
2. Nastavení parametrů měření spektrálního analyzátoru
3. Inicializace řídicí knihovny syntezátoru
4. Nastavování frekvence v definovaném cyklu, současné měření parametrů výstupu, výpočet frekvenčních rozdílů a vynášení grafu závislosti výkonu na frekvenci
5. Nalezení a zobrazení průměrné frekvenční odchylky

Literatura

- [1] R&S ZVL *Vector Network Analyzer* [online]. Operating Manual [cit. 2009]. Dostupný z WWW: www2.rohde-schwarz.com.
- [2] HANUS, S.; SVAČINA, J. *Vysokofrekvenční a mikrovlnná technika*. Elektronické skriptum. Brno: FEKT VUT v Brně, 2002.