

**UNIVERZITA PARDUBICE  
DOPRAVNÍ FAKULTA JANA PERNERA**

**DIPLOMOVÁ PRÁCE**

**2011**

**Bc. Václav Vlasák**

**Univerzita Pardubice**  
**Dopravní fakulta Jana Pernera**

**Přijímač referenčního UWB signálu**

**Bc. Václav Vlasák**

**Diplomová práce**

**2011**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Václav VLASÁK**  
Osobní číslo: **D09775**  
Studijní program: **N3708 Dopravní inženýrství a spoje**  
Studijní obor: **Dopravní infrastruktura-Elektrotechnická zařízení**  
Název tématu: **Přijímač referenčního UWB signálu**  
Zadávající katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

### Z á s a d y p r o v y p r a c o v á n í :

Navrhněte a realizujte přijímač referenčního UWB signálu. Synchronizaci časování příjmu vůči UWB vysílači proveďte pomocí principů fázového - časového závěsu. Verifikaci správného nastavení fázového závěsu proveďte pomocí zobrazení úrovně přijímaného signálu na LCD displeji.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

[1] REISENZAHN, A., BUCHEGGER, T., KAINEDER, G., DISKUS, CH.: Phase-Synchronization in UWB Receivers with Sampling Phase Detectors, Microwave Symposium Digest, 2005.

[2] LIU, L., FANG, G.: A Novel UWB Sampling Receiver and Its Applications for Impulse GPR Systems, IEEE Geoscience and Remote Sensing Letters., vol. 7, no. 4, Oct 2010.

[3] ABUASAKER, S., KOMPA, G.: A High Sensitive Receiver for Baseband Pulse Microwave Radar Sensor Using Hybrid Technology, Department of High Frequency Engineering, University of Kassel.

Vedoucí diplomové práce:

**Ing. Jan Mrkvica, Ph.D.**  
Retia a.s. Pardubice

Datum zadání diplomové práce: **15. prosince 2010**

Termín odevzdání diplomové práce: **31. května 2011**



prof. Ing. Bohumil Culek, CSc.

děkan

L.S.



doc. Ing. Radovan Doleček, Ph.D.

vedoucí katedry

V Pardubicích dne 11. března 2011

## **Prohlášení**

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 20. 05. 2011

Bc. Václav Vlasák

## Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu diplomové práce Ing. Janu Mrkvicovi, Ph.D. za jeho pomoc s jakýmkoliv problémem, který se v průběhu práce objevil. Také za rady, připomínky a veškeré doplňující materiály týkající se dané problematiky.

Taktéž bych chtěl touto cestou poděkovat Ing. Martinovi Šafaříkovi, který mne vždy s ochotou vyslechl a se zájmem mi pomohl řešit, nebo konzultovat softwarovou část zařízení.

Dále bych chtěl poděkovat zejména firmě Retia, a.s. za finanční podporu a další prostředky, při samotné realizaci přijímače.

Velký dík též patří rodině za její podporu a Jiřímu Nováčkovi, který mi během studia rozšiřoval obzory programovacího jazyka C/C++.

## Anotace

Práce obsahuje popis návrhu a realizace přijímače referenčního UWB signálu založeného na principu časového závěsu a synchronního časování. Koncepce vychází z ekvivalentního vzorkování s využitím širokopásmového obvodu S/H, dále obsahuje rychlý A/D převodník a syntezátor DDS, který je ovládán hradlovým polem FPGA. Data potřebná k informaci o správné funkci zařízení jsou z FPGA posílána do mikroprocesoru typu ARM, který je pomocí LCD displeje zobrazuje uživateli.

První část práce přibližuje principy a specifické vlastnosti UWB komunikace, teoretické rozbory, popis a funkce jednotlivých prvků navržené struktury zařízení. V druhé části jsou uvedena praktická řešení dílčích částí přijímače, měření, popisy schémat zapojení a programů pro FPGA a mikroprocesor. Následné přílohy obsahují obrázky plošných spojů, výpisy programů a fotografie dokumentující celé zařízení.

**Klíčová slova:** Ekvivalentní vzorkování, stroboskopický puls, přímá číslicová syntéza, ARM, FPGA, A/D převodník

## Annotation

In this thesis a design receiver of UWB reference signal is presented, which is based on time locked loop and synchronous timing. The concept uses the equivalent time sampling with wideband S/H, circuit it contains a fast A/D converter and next synthesizer DDS, which is controlled by FPGA. For checking the proper function of the device, Specific data is sent from the FPGA to the ARM procesor type to verify the prope function of the devices. This data is displayed to the user on LCD display.

The first part of the work includes principles and specific properties of UWB commutation, theoretical analysis, description and functions of individual elements of device. Practical solution of the particular parts of receiver, measurement, hardware description and programs for FPGA and microprocessor are shown in the second part. In the appendix, there are PCB, listing program and photos of realized receiver.

**Index terms:** Equivalent of sampling, Gaussian pulse, direct digital synthesis, ARM, FPGA, A/D convertor.

# Obsah

<b>ÚVOD</b> .....	<b>10</b>
<b>1 TEORETICKÁ ČÁST</b> .....	<b>12</b>
1.1 DRUHY UWB KOMUNIKACE .....	12
1.2 SPECIFICKÉ VLASTNOSTI PULSNÍHO UWB SIGNÁLU .....	13
1.2.1 Kapacita komunikačního kanálu .....	14
1.2.2 Šíření signálu .....	16
1.2.3 Vliv UWB spektra na úzkopásmové rádiové služby .....	19
1.3 SYNCHRONIZOVANÁ PULSNÍ UWB KOMUNIKACE .....	19
1.4 PŘÍJEM UWB SIGNÁLU S ČASOVÝM ZÁVĚSEM .....	24
1.5 MOŽNOST ŘEŠENÍ ČASOVÉHO ZÁVĚSU .....	25
1.5.1 Princip DDS .....	25
1.5.2 Vlastnosti DDS .....	27
1.6 PROSTŘEDKY PRO ŘÍZENÍ ČASOVÉHO ZÁVĚSU .....	30
1.6.1 Procesory ARM .....	31
1.6.2 Obvody FPGA .....	32
1.7 VZORKOVÁNÍ UWB SIGNÁLU .....	33
1.7.1 Ekvivalentní vzorkování .....	33
1.7.2 Vzorkovací obvod .....	33
1.8 A/D PŘEVODNÍKY .....	36
1.8.1 Vybrané parametry A/D převodníků .....	36
<b>2 EXPERIMENTÁLNÍ ČÁST</b> .....	<b>37</b>
2.1 KONCEPCE ZAŘÍZENÍ .....	37
2.2 NÁVRH OBVODU PROCESORU .....	38
2.2.1 Parametry procesoru AT91SAM7X256 .....	38
2.2.2 Obvodové řešení .....	39
2.3 NÁVRH OBVODU FPGA .....	42
2.3.1 Parametry hradlového pole XC3S50 .....	42
2.3.2 Obvodové řešení .....	43
2.3.3 Zdroj hodinového signálu .....	45
2.4 NÁVRH OBVODU DDS .....	46
2.4.1 Parametry syntezátoru AD9852 .....	46
2.4.2 Obvodové řešení .....	47
2.4.3 Řízení časového posunu procesorem ARM .....	49
2.4.4 Řízení časového posunu obvodem FPGA .....	51
2.5 NÁVRH A MĚŘENÍ VZORKOVAČE .....	52
2.5.1 Obvodové řešení .....	52
2.5.2 Měření vzorkovače .....	53
2.6 NÁVRH OBVODU A/D PŘEVODNÍKU .....	59
2.6.1 Parametry převodníku AD9248-20 .....	59
2.6.2 Obvodové řešení .....	59
2.7 NÁVRH NAPÁJECÍHO ZDROJE .....	61
2.8 ŘEŠENÍ SOFTWAREVÉ ČÁSTI .....	63
2.8.1 Popis programu pro hradlové pole .....	63
2.8.2 Popis programu pro procesor .....	66
2.9 FUNKCE A MĚŘENÍ ZREALIZOVANÉHO PŘÍJÍMAČE .....	67
2.9.1 Funkce přijímače .....	67
2.9.2 Měření přijímače .....	68
<b>ZÁVĚR</b> .....	<b>71</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>73</b>



---

<b>SEZNAM TABULEK .....</b>	<b>75</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>76</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>78</b>
<b>SEZNAM ZKRATEK .....</b>	<b>79</b>

---

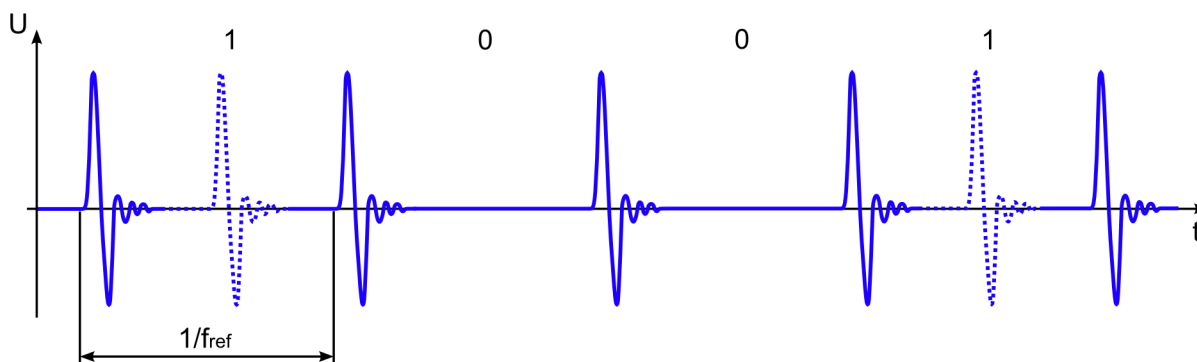
## Úvod

Cílem diplomové práce je návrh a realizace přijímače referenčního UWB signálu, který by využíval časový závěs pro synchronizaci časování příjmu vůči vysílači.

Hlavním důvodem, pro zvolení této struktury byla zkušenost se snadnou zarušitelností přijímače s přímou detekcí popsaného v mé bakalářské práci [1]. Uvedená koncepce vykazovala i přes značné zvýšení citlivosti zarušitelnost jakýmkoliv vysílaným signálem, a to i úzkopásmovým, který je propuštěn frekvenčním filtrem (pásmovou propustí) v přijímači.

Stejně jako princip přímé detekce i tento způsob zahrnuje nevýhody v podobě nízké dosahované citlivosti a malého dynamického rozsahu přijímače. Navíc je zde důležité provádět změnu časového posunu dostatečně rychle, a to i více než  $10^6$  za sekundu, což vychází z opakovací frekvence přijímaných pulsů. Odolnost proti zarušení širokopásmovým i úzkopásmovým signálem by měla být však neporovnatelně vyšší než u přijímače s přímou detekcí.

Tento fakt vychází mimo jiné i z principu samotného přijímání UWB signálu s referencí. Podle Obr. 1 je vysílán referenční puls s přesně danou opakovací (referenční) frekvencí a za ním je v definovaný čas přenášena informace modulací OOK (On-Of Keying). Vzhledem k této specifické vlastnosti se vyhodnocení přijímané informace omezuje na velmi krátký časový úsek, který je sledován, a díky tomu pravděpodobnost přijetí jakýchkoliv rušivých signálů je mnohem menší, než u předešlého typu přijímače. Z uvedeného je také zřejmé, že nejdůležitějším požadavkem pro přenos dat touto metodou je správná a spolehlivá detekce a následná synchronizace referenčního UWB signálu.



Obr. 1: Přijem UWB signálu s referencí

Přijímač referenčního UWB signálu bude sestaven z několika funkčních bloků. Vstupní blok bude zajišťovat ekvivalentní vzorkování s obvodem S/H, jehož okamžik odběru vzorků bude řízen časovým posunem, který bude řešen přímou číslicovou syntézou DDS. Kvůli požadavku dosažení co možná nejvyšší dynamiky změny časového posunu bude obvod DDS řízen rychlým hradlovým polem FPGA. Tento prvek bude také načítat a zpracovávat data z A/D převodníku, který je připojen na mezifrekvenční výstup vzorkovače. Zjišťované specifické hodnoty měřeného UWB signálu budou z FPGA posílány do procesoru typu ARM, který je bude pro kontrolu správné detekce referenčního signálu zobrazovat pomocí LCD displeje uživateli. Procesor bude také sloužit ke konfiguraci hradlového pole.

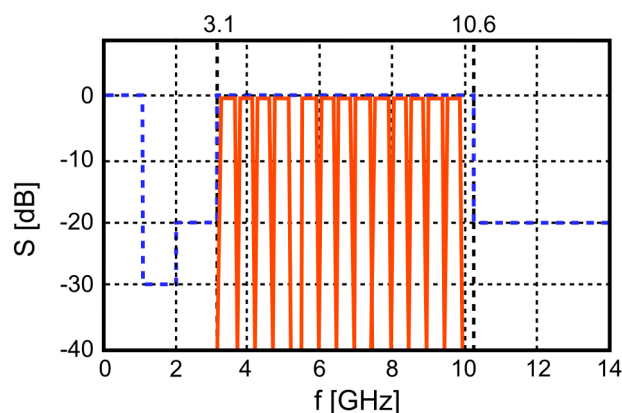
# 1 Teoretická část

## 1.1 Druhy UWB komunikace

Ultraširokopásmový signál je nejčastěji definován jako signál o šířce frekvenčního pásma větší než 500 MHz, nebo větší než 20% vzhledem ke střední frekvenci [2].

Možností jak dosáhnout velké šířky pásma může být několik – např. vysílání na několika nosných frekvencích zároveň tzv. UWB OFDM nebo vysílání krátkých pulsů v čase, což je označováno jako DS-UWB.

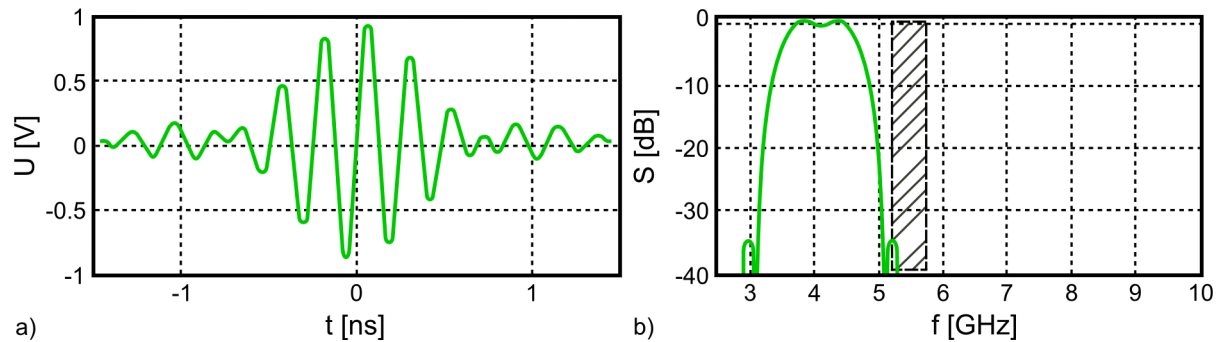
První způsob vychází ze známé a již několik let zpracované technologie OFDM, která využívá běžný úzkopásmový přístup, ale ve velkém počtu kanálů. U této technologie je povolené frekvenční spektrum vyplněno několika základními pásmy o šířce 528 MHz a každé pásmo je dále rozděleno na 128 přenosových pásem o šířce 4 MHz, přičemž se mezi jednotlivými pásmy přeskakuje. Stanovená kmitočtová pásma jsou uvedena na Obr. 1. 1, který vychází z [2]. Tato metoda však obsahuje všechny nevýhodné vlastnosti běžné úzkopásmové komunikace, jako jsou interference, různé druhy úniků atd., ale po technologické stránce je velmi dobře zpracována. Vyznačuje se velmi vysokou rychlostí a v případě, že je nějaká část spektra rušena nebo obsazena nějakou jinou službou, mohou se pásma v této oblasti vynechávat. Díky této vlastnosti tedy nemusí docházet k vzájemnému rušení různých rádiových služeb.



Obr. 1. 1: Obsazení frekvenčního spektra technologií UWB OFDM

Dalším způsobem pro získání velké šířky pásma je technologie DS-UWB, která využívá k přenosu dat krátké pulsy, jejichž typický tvar je uveden na Obr. 1. 2,

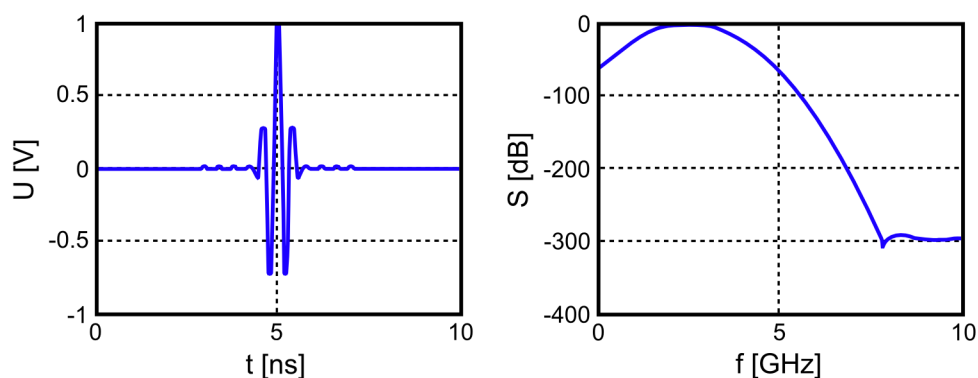
který vychází z [2]. Tento tvar pulsu přímo určuje spektrum vysílání. Modulace dat může být prováděna časovým posunem pulsu, změnou amplitudy nebo fáze vysílaného pulsu. V porovnání s UWB OFDM se vyznačuje mnoha výhodami, jako je např. větší přenosová rychlost. Velkou nevýhodou je však velmi problematická detekce krátkých UWB pulsů. Nikde v literatuře není prozatím uvedeno jak efektivně UWB pulsy přijímat na větší vzdálenosti.



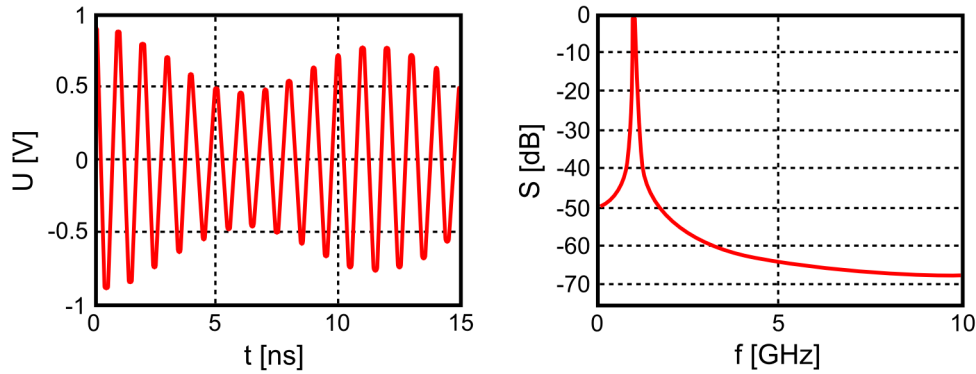
Obr. 1. 2: Technologie DS-UWB: a) tvar pulsu; b) normované spektrum

## 1.2 Specifické vlastnosti pulsního UWB signálu

Z uvedeného vyplývá, že technologie DS-UWB využívá impuls krátký v čase, ale široký ve frekvenční oblasti, což je zobrazeno na Obr. 1. 3. Běžný úzkopásmový signál je však v čase dlouhý a ve frekvenční oblasti úzký viz Obr. 1. 4. Uvedené obrázky vycházejí z [3].



Obr. 1. 3 Charakter UWB pulsu v časové a frekvenční oblasti



**Obr. 1. 4: Charakter úzkopásmového signálu v časové a frekvenční oblasti**

Úzkým pulsem je možné podle vztahu (1.1) z [4] dosáhnout vysoké mezní rozlišovací schopnosti v dálce  $\Delta R$ .

$$\Delta R = \frac{c\tau_p}{2}, \quad (1.1)$$

kde:  $c$ ..... rychlost šíření světla

$\tau_p$  ..... šířka pulsu

Nespornou výhodou UWB signálu je jeho specifické šíření v obecném prostředí. Jelikož je puls velmi krátký v čase, lze z přijatého časového průběhu signálu rozeznat různé cíle a odrazy. V případě jakékoliv změny ve sledované oblasti, jako je např. změna pozice člověka, dochází ke změně přijímaného signálu.

Jak již bylo uvedeno výše, hlavní nevýhodou UWB signálu jsou problémy s přijímáním a následným zpracováním krátkých pulsů. Obvyklé řešení je realizováno pomocí korelátorů nebo širokopásmového vzorkování.

### 1.2.1 Kapacita komunikačního kanálu

Maximální teoretická hodnota kapacity přenosového kanálu při nulové chybovosti BER lze určit z Shannon–Hartleyova vztahu [5]

$$C = B \cdot \log_2 \left( 1 + \frac{S}{N} \right), \quad (1.2)$$

kde:  $C$ ..... maximální přenosová kapacita

$B$ ..... šířka frekvenčního pásma

$S$ ..... střední úroveň signálu

$N$ ..... střední úroveň aditivního bílého Gaussovského šumu

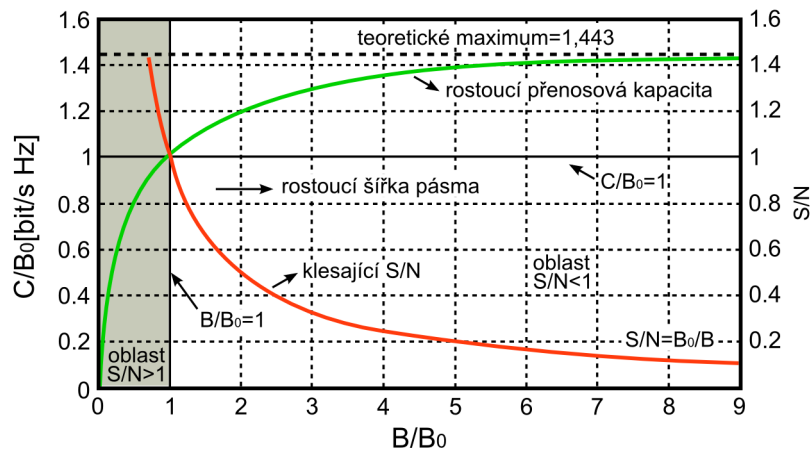
Pokud se ze vztahu (1.2) vyjádří výkon šumu  $N$  jako součin spektrální výkonové šumové hustoty  $N_0$  a šířky pásma  $B$ , tedy  $N = N_0 B$ , a zavede-li se do něho poměrná šířka pásma  $B_0 = S / N_0$ , lze vztah (1.2) upravit do normovaného tvaru

$$\frac{C}{B_0} = \frac{B}{B_0} \log_2 \left( 1 + \frac{B_0}{B} \right). \quad (1.3)$$

V tomto tvaru je normovaná přenosová kapacita  $C / B_0$  vyjádřena již jako funkce jedné proměnné, a to poměrné šířky pásma  $B / B_0$ . Tato funkce je uvedena na Obr. 1. 5, který vychází z [5]. Zde je také zobrazena závislost poměru  $S / N$  na poměrné šířce pásma, pro kterou z rovnosti posledních členů relací (1.2) a (1.3) vyplývá vztah

$$\frac{S}{N} = \frac{1}{B / B_0}. \quad (1.4)$$

Normovaná šířka pásma  $B / B_0$  odpovídá stavu, kdy se výkon signálu  $S$  rovná výkonu šumu  $N$ . Svislice vedená bodem  $B / B_0 = 1$  na vodorovné ose potom dělí celý graf na dvě části. Levá (šedá) část odpovídá klasickým radiokomunikačním systémům, které pracují při poměru  $S / N$  podstatně větším než jedna, přičemž jejich normovaná přenosová kapacita je hluboko pod dosažitelným maximem  $C / B_0 = 1,443$ . Pravá část odpovídá širokopásmovým radiokomunikačním službám, kdy hodnota  $S / N < 1$  a signál je tedy nižší než úroveň šumu. Z obrázku je zřejmé, že větší kapacity signálu lze získat dvěma odlišnými přístupy - zvýšením úrovně vysílaného signálu, což se běžně provádí, nebo zvýšením šířky vysílaného pásma. Realizace komunikačního systému, kdy  $B / B_0 > 1$  a  $S / N < 1$  je obecně podstatně složitější.



Obr. 1. 5: Závislost normované přenosové kapacity a poměru S/N radiového komunikačního systému na normované šířce pásma

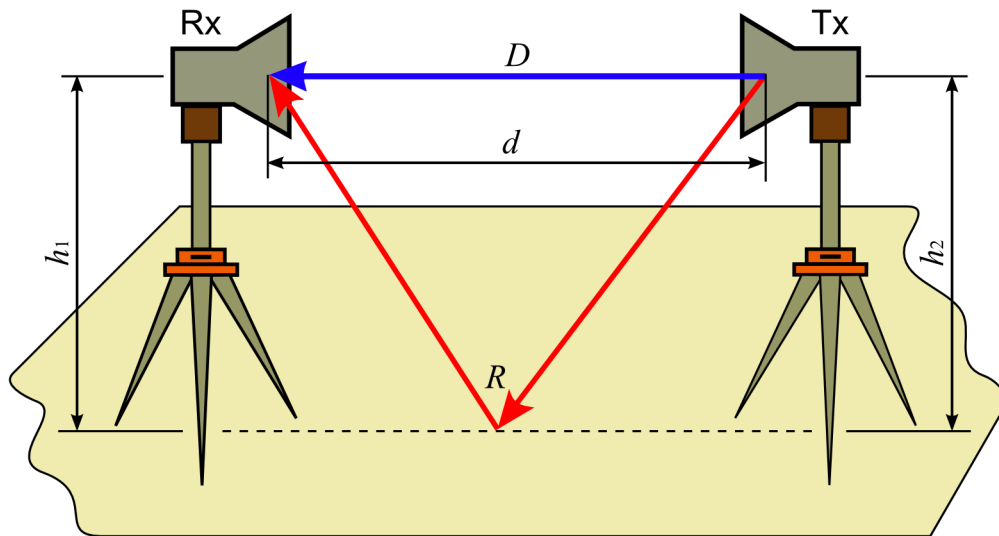
### 1.2.2 Šíření signálu

U běžných úzkopásmových služeb mohou vznikat stojaté vlny způsobené sčítáním a odrazy dopadající a odražené vlny. Tento jev je však odlišný v případě UWB pulsů. Pulsy jsou krátké v čase a rychlost šíření signálu konečná. I přes velmi vysokou rychlost nastává situace, že v určitém bodě od rozhraní je možné sledovat nejprve puls šířící se k rozhraní a za určitý malý okamžik následně odražený od rozhraní. Lze tedy pozorovat dva časově posunuté pulsy, přičemž vzniká fázové sčítání signálu, v podstatě interference.

Šíření UWB signálu vykazuje proti šíření běžných úzkopásmových signálů řadu odlišností. Při šíření volným prostorem je výkon přijímaného signálu závislý na vzdálenosti  $r$  od vysílací antény úměrně  $1/r^2$ . V tomto případě není mezi UWB a úzkopásmovým signálem žádný rozdíl. Ten však přichází v případě šíření v poloprostoru omezeném povrchem země [2].

Pokud se uvažuje šíření úzkopásmového signálu s frekvencí  $f$  mezi dvěma anténami nad vodivou rovinou, k určení úrovně signálu v místě příjmu lze využít tzv. dvoupraskový model, který předpokládá existenci vlny přímé a vlny odražené od země. Geometrické rozložení této situace je zobrazeno na Obr. 1. 6.





Obr. 1. 6: Dvoupaprskový model šíření nad rovinnou zemí

Délka přímé trasy v závislosti na vodorovné vzdálenosti mezi anténami  $d$  je

$$D = \sqrt{d^2 + (h_1 - h_2)^2}, \quad (1.12)$$

kde:  $h_1, h_2, \dots$  výšky antén

Délka trasy odražené vlny je následně

$$R = \sqrt{d^2 + (h_1 + h_2)^2}. \quad (1.13)$$

Intenzita elektrického pole v místě příjmu je pro vlny s horizontální polarizací součtem intenzit přímého a odraženého pole podle vztahu

$$|\mathbf{E}| = |\mathbf{E}_1 + \mathbf{E}_2| \propto \frac{1}{D} e^{-jkD} + \Gamma \cdot \frac{1}{R} e^{-jKR}, \quad (1.14)$$

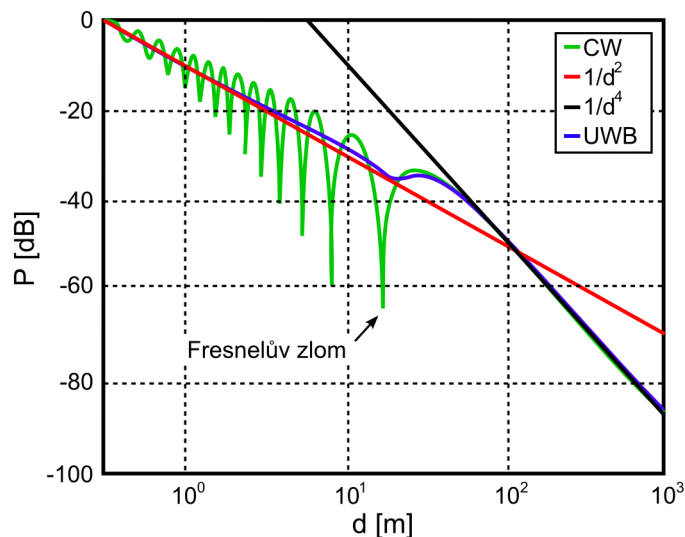
kde:  $k, \dots$  konstanta šíření ( $k = 2\pi f / c$ )

$\Gamma, \dots$  komplexní činitel odrazu země ( $\Gamma = -1$  pro případ dokonale vodivé plochy)

Při přijímání úzkopásmového signálu (CW) je jeho časovým průběhem harmonická funkce. Sčítání přímého a odraženého signálu představuje sčítání dvou vzájemně fázově posunutých harmonických funkcí. Pro libovolnou vzdálenost antén  $d$  dochází v místě příjmu ke sčítání přímé a odražené vlny, což se projevuje výskytem interferenčních jevů. V oblasti nejmenších  $d$ , kde je rozdíl délek tras

přímého a odraženého signálu největší (a větší než  $\lambda/2$ ), úroveň výkonu rychle osciluje mezi oblými maximy a ostrými minimy. Tyto interferenční jevy v podobě oscilací pokračují s rostoucí vzdáleností  $d$  až do bodu  $d_F = 4h_1h_2/\lambda$ , kde rozdíl délek tras obou signálů klesne na  $\lambda/2$ . Tento bod bývá nazýván tzv. Fresnelovým zlomem a projevuje se minimem přenosu. Pro vzdálenosti antén větší než tento zlom je rozdíl délek menší než  $\lambda/2$  a dále klesá teoreticky až k nule. Přijímaný výkon se proto přibližuje dalšímu minimu úměrně s  $1/d^4$ . Tuto skutečnost lze tématicky přiblížit na Obr. 1. 7, který vychází z [2].

Širokopásmové UWB pulsy jsou na rozdíl od úzkopásmových signálů v místě příjmu přítomny jen krátký časový okamžik. Odražený signál je v tomto případě zpožděná kopie přijímaného pulsu, avšak s opačnou polaritou. Pro větší rozdíl délek tras přímého a odraženého signálu než je šířka UWB pulsu, nedojde k časovému překryvu pulsů a tudíž ani k žádným interferenčním jevům. V této oblasti vzdáleností je pokles výkonu přijatého signálu úměrný  $1/d^2$ . V oblasti, kde dochází k částečnému překryvu pulsů, může nastat snížení nebo zvýšení přijatého výkonu vzhledem k šíření volným prostorem, nedochází však k interferenčním jevům jako u úzkopásmového signálu.



Obr. 1. 7: Závislost přijatého výkonu na vzdálenosti antén

Výše popsané vlastnosti jsou pro UWB signál výhodou, avšak nevýhodná je změna tvaru pulsu při průchodu stavebními materiály a také při šíření různými odrazy od povrchu země a budov. Při sčítání pulsu procházejícího přímou cestou a časově

posunutými, amplitudově zeslabenými kopiemi pulsů přicházejícími různými odrazy a lomy v materiálu jsou změny tvaru pulsů největší. Tato skutečnost způsobuje, že je nemožné odhadnout a použít pro detekci pulsů přes překážky nějaký obecný tvar přijímaného pulsů, a tudíž v tomto případě není možné využít k detekci kolerátory.

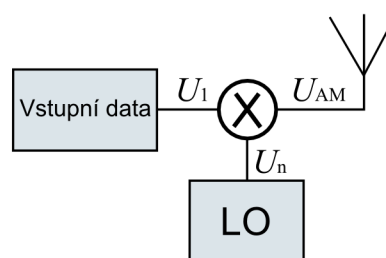
### 1.2.3 Vliv UWB spektra na úzkopásmové rádiové služby

Doposud uvedené výhody pulsní UWB technologie lze doplnit o další pozitivní vlastnost - i když může být okamžitá vysílaná energie pulsů dosti velká, tak nemusí ovlivňovat ostatní rádiové služby. Tento fakt je způsoben dvěma důvody. První odůvodnění je, že vysílaná energie je rozprostřena do širokého frekvenčního pásma, a tudíž na konkrétní frekvenci se nachází jen malá část energie, která by rušila úzkopásmovou komunikaci. Druhým důvodem je snížení celkového vysílaného výkonu malou opakovací frekvencí vysílání pulsů. Pokud se snižuje opakovací frekvence vysílání pulsů, klesá i střední úroveň spektrální hustoty výkonu. Např. pro vysílání pulsů opakovací frekvencí 10 MHz s šířkou pulsů 1 ns je pouze jednou za 100 ns vyslán puls a zbylých 99 ns se nic nevysílá. Při modulaci OOK lze touto opakovací frekvencí dosáhnout přenosové rychlosti i 10 Mbit/s. UWB pulsy se díky tomu pro běžnou úzkopásmovou komunikaci tváří jako vzrůst úrovně šumu pozadí.

### 1.3 Synchronizovaná pulsní UWB komunikace

Z požadavků na přenos informace je důležitá spolehlivá detekce přijímaného signálu, ovšem s ohledem na výše popsané je zřejmé, že největší nevýhodou UWB komunikace je právě obtížná detekovatelnost, jejíž důvod lze vysvětlit pomocí částečné analogie s běžnou úzkopásmovou technologií uvedené v [6].

Na Obr. 1. 8 je uvedeno zjednodušené schéma úzkopásmového vysílače.



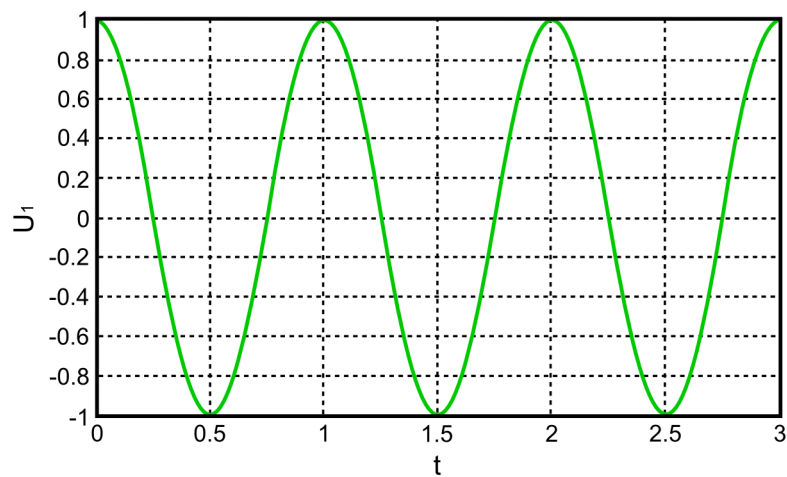
Obr. 1. 8: Blokové schéma vysílače úzkopásmového signálu

Z obrázku je zřejmé, že vstupní analogová data  $U_1$ , uvedená na Obr. 1. 9, vstupují na směšovač spolu s vysokofrekvenčním nosným signálem  $U_n$  lokálního oscilátoru LO, jehož průběh je uveden na Obr. 1. 10. Vstupní signál lze v tomto případě popsat vztahem

$$U_1(t) = |U_1| \cdot \cos(2\pi f_1 t + \varphi_1), \quad (1.15)$$

kde:  $f_1$  ..... kmitočet analogového signálu, který obsahuje data

$\varphi_1$  ..... fázový posun



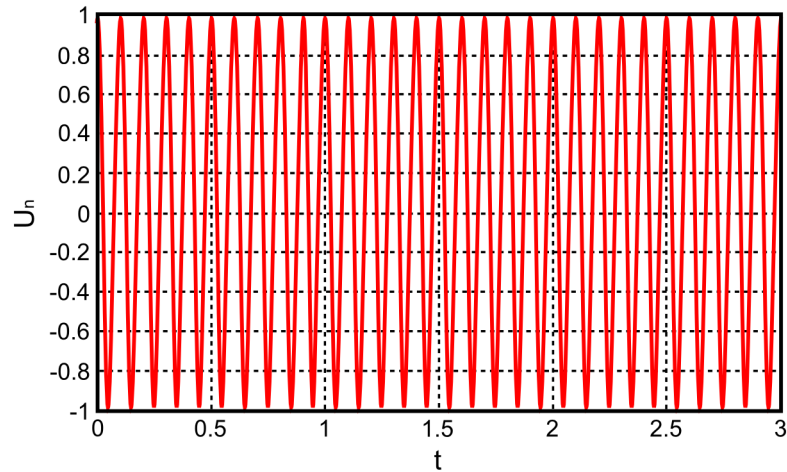
**Obr. 1. 9: Analogový přenášený signál**

Signál lokálního oscilátoru lze vyjádřit jako

$$U_n(t) = |U_n| \cdot \cos(2\pi f_n t + \varphi_n), \quad (1.16)$$

kde:  $f_n$  ..... kmitočet lokálního oscilátoru

$\varphi_n$  ..... fázový posun



Obr. 1. 10: Nosný signál

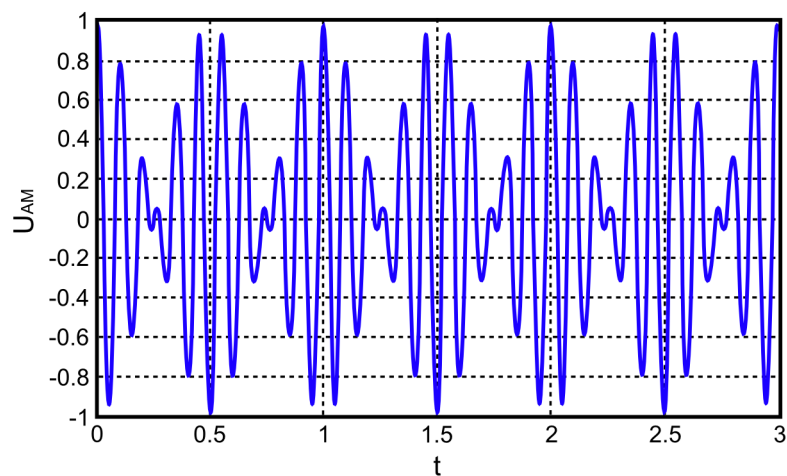
Směšovač na nelineárním prvku provádí násobení signálu podle vztahu

$$U_{AM}(t) = |U_1 \cdot U_n| \cdot \cos(2\pi f_1 t + \varphi_1) \cos(2\pi f_n t + \varphi_n). \quad (1.17)$$

Po úpravě přechází vztah (1.17) na

$$U_{AM}(t) = \frac{1}{2} |U_1 \cdot U_n| \cdot \left( \cos(2\pi t(f_1 - f_n) + \varphi_1 - \varphi_n) + \cos(2\pi t(f_1 + f_n) + \varphi_1 + \varphi_n) \right). \quad (1.18)$$

Výsledný signál po směšování, který je vysílán, je uveden na Obr. 1. 11.



Obr. 1. 11: Vysílaný vysokofrekvenční signál

Vysílaný signál je následně přijímán na přijímači vycházejícím ze zjednodušeného schématu na Obr. 1. 12. Zde opět dochází na směšovači k násobení se signálem místního lokálního oscilátoru, jež popisuje vztah

$$U_{nRx}(t) = |U_{nRx}| \cdot \cos(2\pi f_{nRx}t + \varphi_{nRx}), \quad (1.19)$$

kde:  $f_{nRx}$  ..... kmitočet místního lokálního oscilátoru

$\varphi_{nRx}$  ..... fázový posun

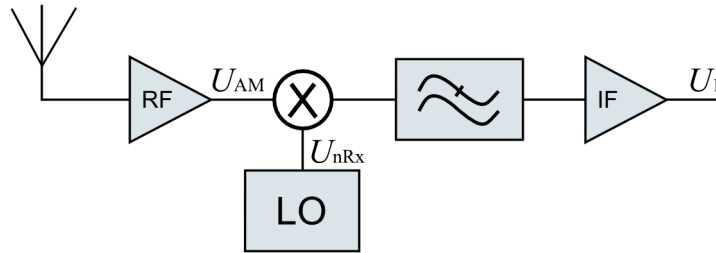
Výsledek směšování lze vyjádřit jako

$$U_{IF}(t) = \frac{1}{2} |U_1 \cdot U_n \cdot U_{nRx}| \cdot \left( \cos(2\pi t(f_1 - f_n) + \varphi_1 - \varphi_n) + \cos(2\pi t(f_1 + f_n) + \varphi_1 + \varphi_n) \right) \cdot \cos(2\pi t f_{nRx} + \varphi_{nRx}). \quad (1.20)$$

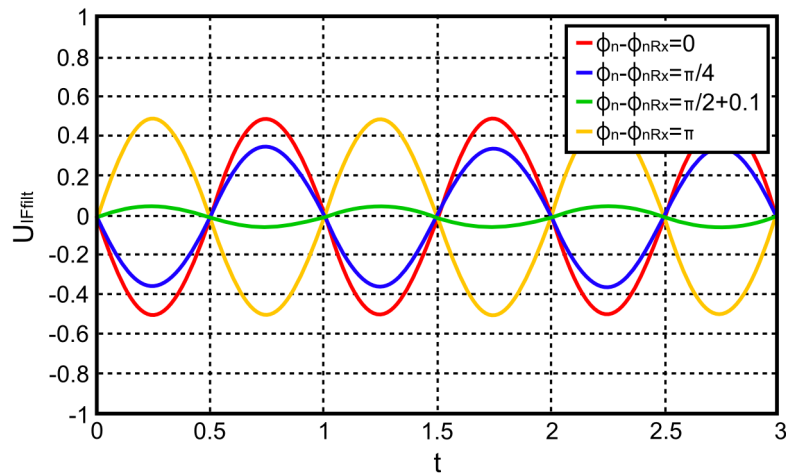
Po následné filtraci dolní propustí a předpokladu že  $f_n = f_{nRx}$ , vztah (1.20) přechází na

$$U_{IFfilt}(t) = \frac{1}{4} |U_1 \cdot U_n \cdot U_{nRx}| \cdot \left( \cos(2\pi t f_1 + \varphi_1 + \varphi_n - \varphi_{nRx}) + \cos(2\pi t f_1 + \varphi_1 - \varphi_n + \varphi_{nRx}) \right). \quad (1.21)$$

Ze vztahu vyplývá, že takto upravený signál neodpovídá vstupnímu signálu (1.15), a proto je nutné splnit podmínku  $\varphi_n - \varphi_{nRx} = 0$ , za které jsou průběhy obou signálů pouze rozdílné v amplitudě. V ostatních případech je signál popsán vztahem (1.21) vůči signálu vstupnímu menší nebo otočený, což je uvedeno na Obr. 1. 13. Tento problém lze odstranit rozšířením struktury přijímače o fázový závěs PLL, kterým je možné synchronizovat fázový rozdíl  $\varphi_n - \varphi_{nRx}$ .

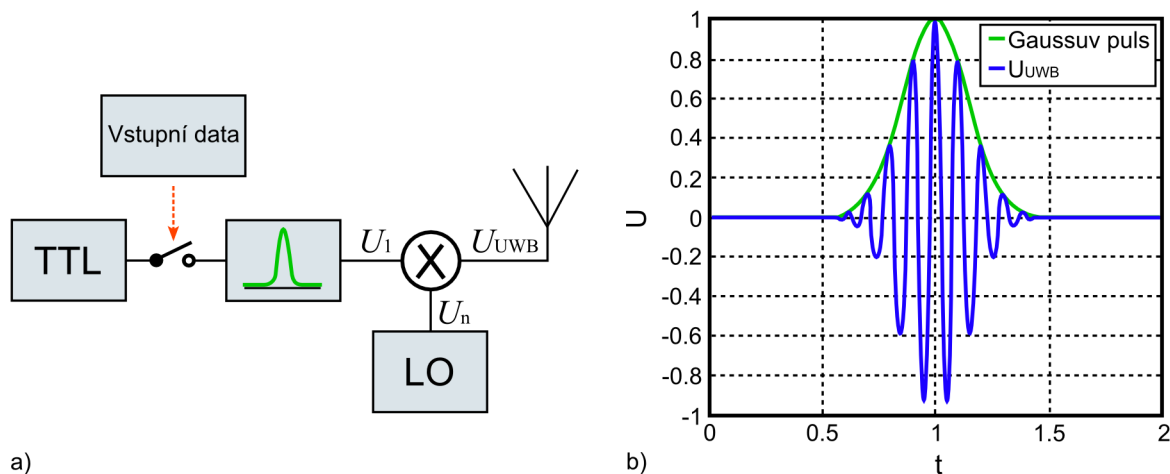


Obr. 1. 12: Blokové schéma úzkopásmového přijímače



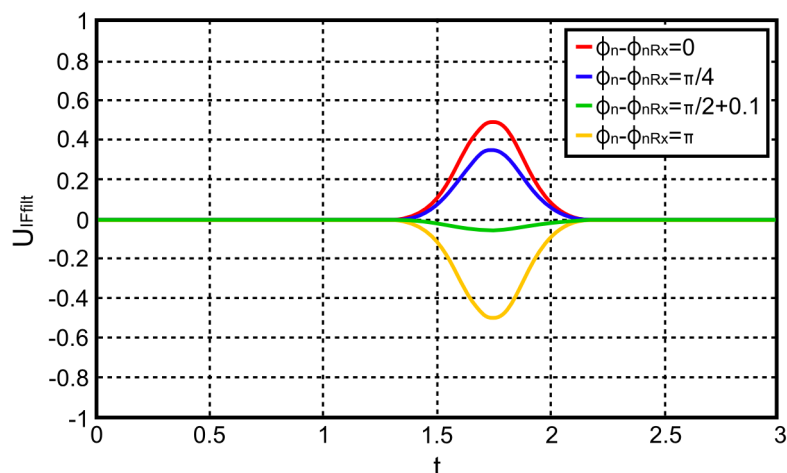
Obr. 1. 13: Filtrovaný signál na mezifrekvenci

Přijímač s fázovým závěsem je možné najít v mnoha variantách prakticky ve všech současných bezdrátových přenosech. Hlavní výhodou této koncepce je vysoká dosahovaná citlivost, dynamický rozsah a navíc jednoduchá realizace. Z těchto důvodů je snaha realizovat UWB přenos dat stejným způsobem. Vysílač UWB pulsů lze obvodově uspořádat obdobně podle uvedeného na Obr. 1. 14a. Poté vysílaná data klíčí budící obdélníkový TTL signál, který následně generuje puls Gaussova tvaru, jenž se směšuje s nosným kmitočtem a následně je vysílán ve specifickém tvaru, zobrazeném modrou křivkou na Obr. 1. 14b.



Obr. 1. 14: Vysílač UWB: a) blokové schéma; b) tvar pulsu na nosném kmitočtu

Příjem UWB signálu probíhá na přijímači shodné konstrukce podle Obr. 1. 12. Demodulovaný signál za směšovačem a IF zesilovačem má shodný tvar jako puls generovaný ve vysílači, ale vyskytují se u něho podobné problémy s rozdílem fází lokálního oscilátoru vysílače a přijímače, což je uvedeno na Obr. 1. 15. Vzhledem k této skutečnosti není možné fázový závěs použít, jelikož se nosný kmitočet nachází v průběhu signálu po velice krátkou dobu a fázový závěs za tak krátkou dobu (obvykle přibližně 1ns) nestihne zareagovat. Danou skutečnost lze ohodnotit jako neznalost pozice pulsu v čase, přičemž další problém se týká samotné detekce základního Gaussova pulsu.



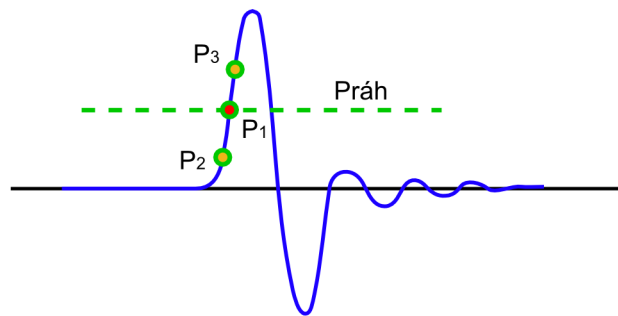
Obr. 1. 15: Demodulovaný Gaussův puls na straně přijímače

#### 1.4 Příjem UWB signálu s časovým závěsem

Z výše popsaného je zřejmé, že strukturu přijímače s fázovým závěsem nelze prozatím kvůli uvedeným důvodům využít. Z toho důvodu je snahou hledat řešení



detekce UWB signálu např. pomocí principu, který využívá časového závěsu. Ten spočívá v nastavení detekčního prahu v měřeném UWB pulsu a následnou změnou časové pozice širokopásmového obvodu S/H se udržuje detekční práh na stejné úrovni. Tato situace je uvedena na Obr. 1. 16.



Obr. 1. 16: Princip časového závěsu

Pokud se puls posune vzhledem časování přijímače do bodu  $P_2$ , měřená hodnota v následném A/D převodníku se zmenší. Naopak pokud se časování přijímače nachází v bodě  $P_3$ , je měřená hodnota větší než práh. Laděním časování se průběžně upravuje vhodná hodnota zpoždění hodinového signálu lokálního oscilátoru vůči přijímanému pulsu. Pokud se tedy přeladěním dosáhlo bodu  $P_2$ , zpoždění se v dalším kroku zvětší, naopak pro situaci v bodě  $P_3$  se zpoždění sníží. V případě, že velikost zpoždění odpovídá bodu  $P_1$ , žádná změna nenastává.

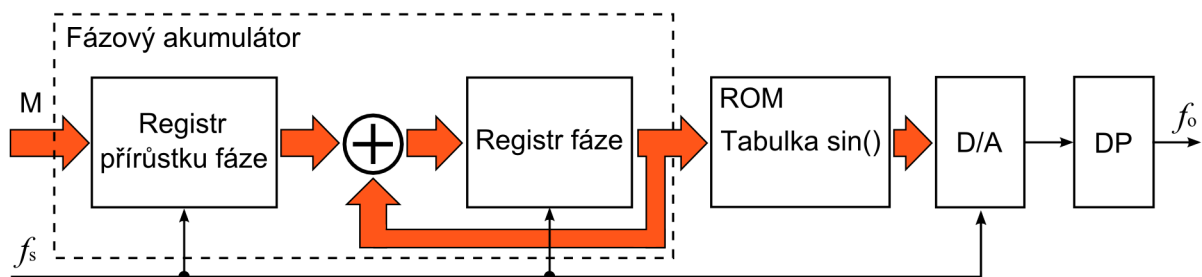
## 1.5 Možnost řešení časového závěsu

Z principu časového závěsu je zřejmé, že nutností pro tuto koncepci je schopnost posouvat UWB pulsy v čase. Časový posun pulsů lze řešit např. programovatelným zpožděním. Krok zpoždění těchto součástek se pohybuje od 20 ps do 2 ns s různým počtem pozic. Nevýhodou těchto obvodů je však neschopnost měnit jejich krok v průběhu pracovní činnosti, jelikož jej mají přesně definovaný. Výhodnější vlastnosti lze nalézt u obvodů přímé číslicové syntézy DDS, u níž je možné krok měnit v širokém rozmezí, a to v průběhu činnosti. Při nastavování zpoždění lze navíc pokrýt celou periodu posouváných pulsů.

### 1.5.1 Princip DDS

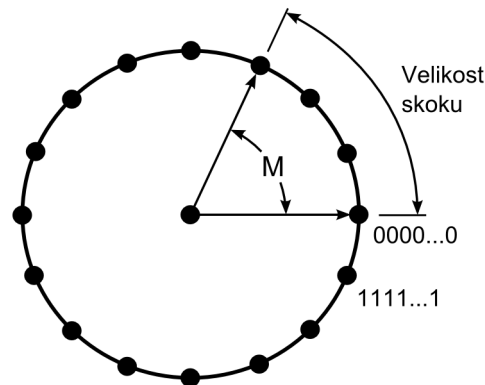
Princip přímé kmitočtové syntézy uvedený v [7] je způsob vytvoření frekvenčně a fázově modulovaného signálu pomocí stabilního referenčního signálu

o konstantní frekvenci. Pro vytvoření výstupního signálu tohoto charakteru je možné použít analogové syntezátory, ty jsou však v současné době nahrazovány integrovanými obvody pro číslicové zpracování signálu. Jemnost ladění výstupní frekvence referenčního signálu je dána hodnotou  $M$ . Toto řídicí slovo označováno taktéž jako Tuning Word Width je jedním z parametrů DDS obvodu a pohybuje se v rozmezí 24 až 48 bitů. Na Obr. 1. 17 je uvedeno blokové schéma obvodu DDS. Tento obvod obsahuje akumulátor fáze, do kterého se přivádí již uvedené řídicí slovo  $M$  o délce  $N$  bitů. Akumulátor se skládá z registru přírůstku fáze, sčítačky a registru fáze. Při každém hodinovém impulsu referenčního signálu akumulátor fáze zvýší svůj obsah o hodnotu  $M$ , což se děje až do okamžiku jeho přetečení. V akumulátoru zůstane zbytek po přetečení a celý cyklus může stejným způsobem pokračovat dále. Výstup akumulátoru je veden do paměti ROM, kde je uložena tabulka funkce sinus. Výstup z paměti je pak přiveden na převodník D/A, na jehož výstupu je připojen analogový filtr dolní propust.



Obr. 1. 17: Blokové schéma DDS

Princip funkce akumulátoru fáze je možné lépe uvést na kruhovém diagramu, který je uveden na Obr. 1. 18. Velikost zvyšování fáze a taktéž výstupní frekvence je ovlivněna hodnotou ladícího slova  $M$ . Se zvyšující hodnotou  $M$  je vyšší frekvence a s ní i zkreslení výstupního signálu, jelikož sinusový signál je na výstupu složen z menšího počtu vzorků na jednu periodu.



Obr. 1. 18: Kruhový diagram s hodnotou fázového skoku  $M$

Na  $N$ -bitový akumulátor fáze (ve většině DDS systémech je  $N$  obvykle 24, 28, 32 a 48), existuje možných  $2^N$  fázových bodů. Digitální slovo fázového registru  $M$ , představuje stupeň akumulátoru fáze zvýšeného při každém hodinovém cyklu. Je-li  $f_s$  frekvence referenčního signálu, pak lze frekvenci výstupního sinusového průběhu  $f_o$  vyjádřit dle následujícího vztahu označovaného jako rovnice ladění

$$f_o = \frac{M \cdot f_s}{2^N}. \quad (1.22)$$

Ze vztahu (1.22) vyplývá, že pokud bude ladící slovo  $M$  rovno jedné, bude se jednat o minimální výstupní kmitočty. Jelikož je DDS obvod se vzorkováním, maximální výstupní frekvence je podle Nyquistova vzorkovacího teorému omezena na  $0,5 f_s$ .

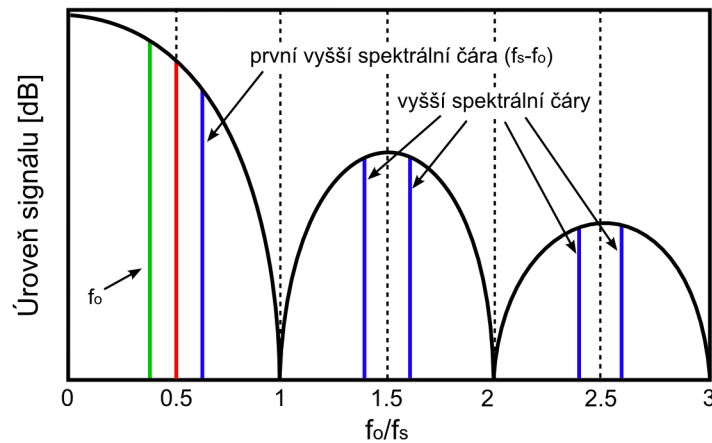
Vzhledem k možnostem nastavení časového zpoždění je minimální krok definován podle vztahu

$$\Delta t_{\min} = \frac{1}{f_o \cdot N}. \quad (1.23)$$

## 1.5.2 Vlastnosti DDS

### • Spektrální vlastnosti

Jelikož je analogový výstup obvodu DDS vytvářen D/A převodníkem, má výstupní signál spektrum odpovídající vzorkování druhého druhu a obálka vzniklých spektrálních složek je dána funkcí  $\sin(x)/x$ , což je uvedeno na Obr. 1. 19.



Obr. 1. 19: Spektrum výstupního kmitočtu DDS před filtrem

Z obrázku vyplývá, že vzniká teoreticky nekonečný počet spektrálních čar s amplitudou, která klesá dle funkce  $\sin(x)/x$ . Základní spektrum signálu je v rozsahu 0 a  $0,5, f_o/f_s$  kde nastává pokles amplitudy o 4 dB proti  $f = 0$ . Při generování signálu  $f_o$  (první spektrální čára) obecně platí, že další obrazové spektrální čáry jsou na frekvencích

$$f_s - f_o, f_s + f_o, 2f_s - f_o, 2f_s + f_o, 3f_s - f_o \text{ atd,} \quad (1.24)$$

Pro odstranění nežádoucích spektrálních čar se obvykle používá filtr dolní propust se šířkou pásma 0 až  $0,5 f_s$  (ve skutečnosti však 0 až  $0,4 f_s$ , jelikož výstupní frekvence vyšší než  $0,4 f_s$ , se kvůli špatné šumové kvalitě a spektrální vlastnosti výstupního signálu většinou nepoužívají).

### • Kvantizační šum

Ve spektru se ovšem objevují i další rušivé spektrální složky. Je to např. kvantizační šum způsobený konečným počtem bitů A/D převodníku. Poměr výkonu výstupního signálu a kvantizačního šumu  $SNR$  je dán vztahem.

$$SNR = 1,76 + 6,02 \cdot D, \quad (1.25)$$

kde:  $D$  ..... počet bitů D/A převodníku

V případě doplnění DDS o filtr dolní propusti omezující pásmo na 40%  $f_s$  se poměr signálu ku šumu zlepšil o 3,98 dB a tudíž vztah (1.25) přechází na

$$SNR = 5,74 + 6,02 \cdot D. \quad (1.26)$$

- **Vliv redukované tabulky sin funkce**

Každá adresa v tabulce odpovídá fázi bodu sinusového průběhu od 0 do 360°. Tabulka obsahuje digitální informaci o amplitudě pro jeden kompletní cyklus sinusového průběhu. V reálném systému DDS však nejsou všechny bity z akumulátoru fáze přenášeny do tabulky, ale jsou zkráceny pouze na prvních 12 až 16 MSB bitů. Tím se snižuje velikost tabulky a hlavně její kapacita paměti. Pro určité generované výstupní frekvence vycházející z určitého  $M$  tímto dochází k zaokrouhlení čísla akumulátoru fáze a vzniká chybový signál pilového průběhu, jenž vytváří rušivé spektrální čáry. Maximální amplituda první spektrální čáry je dána vztahem

$$SP_m = -6,02 \cdot P, \quad (1.27)$$

kde:  $P$  ..... počet bitů přivedených do převodníku fáze/amplituda.

Výsledek uvedeného vztahu je vztažen k amplitudě signálu  $f_o$ . Tento rušivý signál má tu vlastnost, že frekvence spektrálních čar závisí na přivedeném čísle a spektrální čára tak může padnout i do základní spektrální oblasti, ve které nemůže být odfiltrována zmíněnou dolní propustí. Spektrální čára s nejvyšší amplitudou  $f_r$  se označuje za základní a lze ji vyjádřit dle vztahu

$$f_r = f_s \left( ETW / B^2 \right), \quad (1.28)$$

kde:  $B$  ..... počet ignorovaných bitů

$ETW$  ... ekvivalentní ladící slovo, vypočtené dle

$$ETW = 2^B - T, \quad (1.29)$$

kde:  $T$  ..... originální ladící slovo

- **Fázový šum**

Pro dosažení maximální spektrální čistoty a tím i nízké hodnoty fázového šumu je nutné zajistit frekvenční čistotu systémového časování použitého k řízení

DDS, což je z výše popsaného referenční kmitočet  $f_s$ . Fázový šum se ze vstupu redukuje na výstup o hodnotu podle vztahu

$$20 \log \left( \frac{f_o}{f_s} \right). \quad (1.30)$$

- ***Další rušivé vlivy***

Kvalitu výstupního signálu mimo již uvedené mohou ovlivnit také rušení vznikající při přepínání bitů D/A převodníku, pronikání hodinového signálu  $f_s$  a zejména nelinearita převodníku, díky které vznikají vyšší harmonické. Díky vzorkování se jakákoliv harmonická vyšší než  $0,5 f_s$  projeví jako odraz v základním pásmu.

## ***1.6 Prostředky pro řízení časového závěsu***

Struktura přijímače s časovým závěsem už z principu požaduje provádět změnu zpoždění co možná nejrychleji. Pokud by byla rychlost změny nedostatečná, přijímaný puls by byl vzorkován širokopásmovým obvodem S/H nepřesně a správné zachycení časovým závěsem by nebylo možné. Tudíž i následná informace, která by měla přijít v přesně definovaný okamžik po příchodu referenčního signálu, by nemusela být správně přijata. Na rychlosti změny časového posunu tedy závisí spolehlivost komunikace.

Jelikož se díky svým možnostem jeví přímá číslicová syntéza jako nevhodnější varianta pro časový posun, je důležité obvody DDS co nejefektivnějším způsobem využít. Řešení se nachází v mikroprocesorech a hradlových polích.

Procesory se sekvenčním zpracováním dat jsou však vzhledem k výpočetnímu výkonu nevhodné. Implementace složitých operací do jediné instrukce, využívána u procesorů s architekturou CISC, sice vede ke zrychlení běhu programu, ale o to více se zvyšuje složitost programového řadiče procesoru, s kterým se snižuje maximální taktovací frekvence. Naopak minimalizace množství možných instrukcí zjednodušuje programový řadič a umožňuje rychlejší taktování procesoru [8]. V tomto případě se jedná o procesory s architekturou RISC, kterou se vyznačují procesory ARM.

O programovatelné logické obvody s obecnější strukturou, než jaká je u procesorů, se jedná v tom případě, pokud jsou složeny z jednoduchých programovatelných bloků, schopných realizovat logickou funkci. Tyto buňky jsou uspořádány v pravidelné struktuře (poli), přičemž programováním funkce jednotlivých buněk a jejich vzájemným propojením je možné realizovat složité kombinační funkce. Tyto PLD (Programmable Logic Devices) obvody lze rozdělit na SPLD (Simple Programmable Logical Devices), CPLD (Complex Programmable Logical Devices) a FPGA (Field Programmable Gate Array). SPLD obvody jsou schopny realizovat pouze jednodušší funkce. Obvody CPLD vycházejí z implementací několika SPLD a navíc obsahují programovatelnou propojovací strukturu. Největší flexibilitu a efektivitu implementace, spolu s dosahováním vyšších taktovacích kmitočtů, představují obvody FPGA, které jsou složeny z jednoduchých základních logických buněk, mnohem menších, než jsou u CPLD.

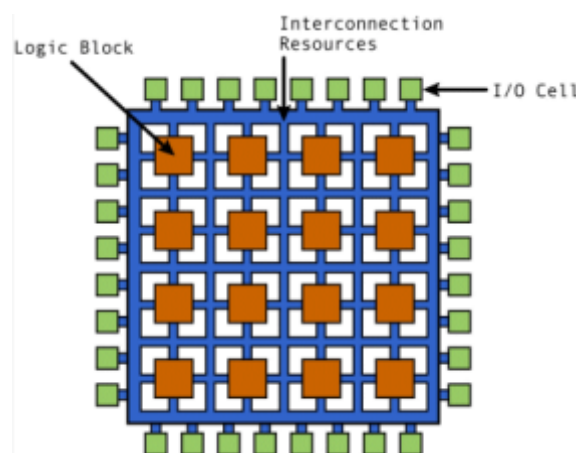
### 1.6.1 Procesory ARM

Z výše naznačeného plyne, že architektura procesoru ARM je založena na redukovaném instrukčním souboru (RISC). Soubor tohoto typu má vysokou instrukční výkonnost a rychlou reálnou odezvu na přerušení, protože jsou instrukce vykonávány přímo hardwarem a nikoliv mikrokódem. Jelikož procesor obsahuje menší počet hardwarových instrukcí, jeho návrh je do jisté míry jednodušší. Za výhody RISCových procesorů se považuje i větší počet univerzálně použitelných registrů. Tyto procesory využívají Von Neumannovu architekturu, což znamená, že data i program jsou v jednom adresovém prostoru. Data se tedy nejprve musí načíst z paměti do registrů, poté se zpracují a opět se uloží zpět do paměti. Dále procesory využívají jednotnou délku instrukcí (44 základních o délce 32b) a některé procesory také disponují druhou 16-bitovou tzv. Thumb sadou. Tato sada ušetří 35 - 40% paměti ve srovnání s 32-bitovým instrukčním souborem, aniž by se výrazně snížil výkon. Procesory také využívají sdílené registry, různé pracovní módy procesoru a pipelining.

Procesory ARM jsou rozděleny podle jádra od počátku vývoje na ARM6, ARM7, ARM9, ARM10, ARM11, ARM CORTEX 3A a další. Jednotlivé varianty se mezi sebou liší maximální taktovací frekvencí, velikostí vnitřní paměti, množstvím periférií a počtem I/O pinů.

## 1.6.2 Obvody FPGA

Díky nejobecnější struktuře, jsou obvody typu FPGA nejkompaktnějšími programovatelnými obvody, které jsou schopny pracovat i na frekvencích stovek MHz. Obecná architektura obvodů FPGA je uvedena na Obr. 1. 20, z kterého lze pozorovat, že základ těchto obvodů tvoří programovatelné logické bloky PLB (Programmable Logic Block), programovatelné propojení PIB (Programmable Interconnection Block) a konfigurovatelné vstupně/výstupní bloky IOB (Input/Output Block).



Obr. 1. 20: Architektura FPGA [9]

Každá logická buňka se skládá z paměťového bloku tvořeného D klopným obvodem, který lze konfigurovat buď jako typ Flip-Flop reagující na náběžnou hranu, nebo Latch, jenž reaguje na potenciálovou hladinu. Mimo to každá buňka obsahuje přenosovou kontrolní logiku pro zrychlení přenosu při provádění aritmetických operací [10]. Některé buňky obsahují tabulku požadované čtyřvstupové funkce LUT, které jsou realizovány jako konfigurovatelné paměti RAM. Obvody, které tuto tabulku neobsahují, jsou programovatelné pouze jednou.

Propojení PIB tvoří matici spojů, která umožňuje propojení logických bloků mezi sebou a také bloků I/O. Typy propojení se liší podle délky a způsobu rozmístění v FPGA.

Uvedené I/O bloky tvoří rozhraní mezi bloky PLB a výstupními piny FPGA. Více těchto bloků dohromady tvoří tzv. banky. Jednotlivé bloky obsahují programovatelné vstupní a výstupní buffery, programovací zpoždovací registry, multiplexory a D klopné obvody.



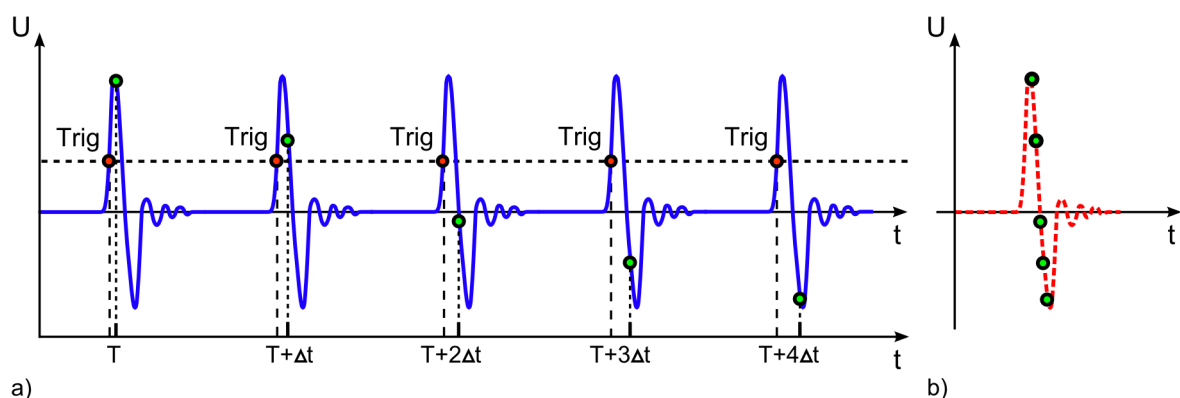
Takto tvořené programovatelné logické obvody dokáží díky své struktuře zpracovávat velmi rychlé datové toky a umožňují snadnou implementaci paralelních algoritmů, čímž lze dosáhnout mnohem vyšších výpočetních výkonů než u mikroprocesorů

## 1.7 Vzorkování UWB signálu

S využitím výše uvedených obvodů lze velmi efektivně řídit obvod DDS, který je schopen v širokém rozsahu měnit časový posun generovaného výstupního kmitočtu. Posun v čase určitého signálu je důležitý pro realizaci širokopásmového vzorkovacího obvodu, jenž je založen na principu ekvivalentního vzorkování.

### 1.7.1 Ekvivalentní vzorkování

Ekvivalentní způsob vzorkování je vhodný pouze pro periodické průběhy a lze jím zobrazit i mikrovlnné signály, které by byly jinak nezobrazitelné. Rekonstrukce signálu je provedena odečtením jednoho bodu měřeného vysokofrekvenčního průběhu při jeho každé periodě. Vysokofrekvenční signál se opakuje s přesně danou opakovací frekvencí a místo odečítání se posouvá v čase o přesně definovaný krok  $\Delta t$  po detekci spouštěcí podmínky (Trig). Každý bod periody musí být odečítán v co nejkratším čase, což umožňují obvody typu S/H. Princip ekvivalentního vzorkování a rekonstruovaný signál je zobrazen na Obr. 1. 21.



Obr. 1. 21: Princip ekvivalentního vzorkování: a) měřený VF signál; b) rekonstruovaný signál

### 1.7.2 Vzorkovací obvod

Pro co nejpřesnější rekonstrukci měřeného signálu je nutné, aby byl každý bod periody při ekvivalentním vzorkování odečítán v co nejkratším čase. Proto je zapotřebí využít obvody, které jsou schopny tuto podmínku zaručit. Řešením jsou

obvody typu S/H, které využívají k odběru vzorků v co nejkratším čase stroboskopické pulsy.

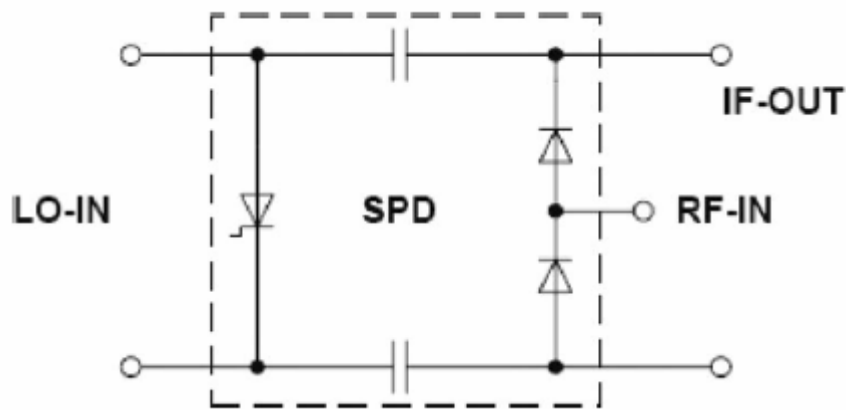
- **Stroboskopické pulsy**

Specifickou vlastností stroboskopických pulsů je jejich co nejužší šířka, která zaručuje otevření S/H obvodu po co nejkratší čas. Šířka takovýchto pulsů se proto pohybuje v desítkách až stovkách ps. Dalším parametrem stroboskopických pulsů je dosažení maximální možné amplitudy, na kterou je kladen hlavní požadavek, aby byla dostatečná k otevření Shottky diody v obvodu S/H. Tvar stroboskopického pulsu v ideálním případě kopíruje Gaussův puls, ovšem reálný puls obsahuje po odeznění zřetelné zákmity, které ho jednak rozšiřují v čase, ale také zhoršují vlastnosti vzorkovacího obvodu. Na ten jsou kladeny nejpřísnější požadavky, jelikož jako jediný rozhoduje o kvalitě a širokopásmovosti na mezifrekvenci.

Stroboskopický puls se může vytvářet přímé z TTL signálu, a to s využitím diod SRD (Step Recovery Diode), jenž generují puls na principu velmi rychlé změny impedance. Generátory s SRD dosahují vysokých amplitud stroboskopických pulsů, avšak jejich návrh je kvůli budicí obvodům značně složitý. Jednodušší variantou vytvoření pulsu na obdobném principu rychlé změny impedance je využití vysokofrekvenčních tranzistorů přivedených do stavu saturace. Tímto způsobem je dosahováno nižších amplitud, avšak návrh generátorů stroboskopických pulsů je levnější oproti koncepci s diodami SRD. Podrobnější pojednání o této problematice je uvedeno v [11].

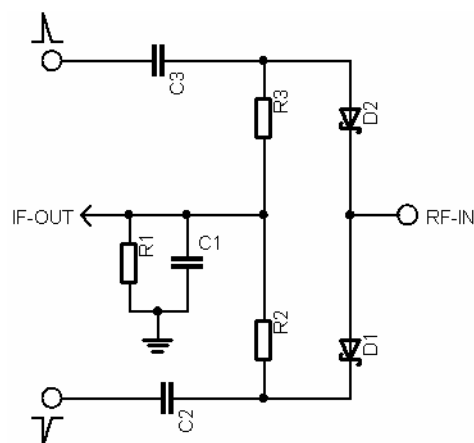
- **Obvod S/H**

Vzorkovacím obvodem, který využívá stroboskopické pulsy, může být např. SPD (Sampling Phase detector) z [12], jenž je uveden na Obr. 1. 22. U této koncepce je na oba vstupy LO-IN přiveden sinusový průběh, na každém vstupu fázově otočený o  $180^\circ$ . Na vstup RF-IN se přivede měřený vysokofrekvenční signál a z výstupu IF-OUT se odvádí mezifrekvenční výstupní signál. Díky buzení obvodu sinusovým signálem se stroboskopické pulsy k otevření S/H obvodu vytvářejí na SRD diodě.



Obr. 1. 22: Obvod SPD [12]

Jelikož není snahou využít jako budící signál sinusový průběh, ale již připravené stroboskopické pulsy vytvořené externím generátorem, lze pak výše uvedený obvod upravit dle Obr. 1. 23. Na vstupy LO-IN tedy přicházejí stroboskopické pulsy opačné polaroty přes kondenzátor nízké kapacity na můstek, který tvoří rezistory  $R2$ ,  $R3$  a Shottky diody  $D1$ ,  $D2$ . Pokud jsou přítomny stroboskopické pulsy, průběhy napětí na diodách jsou stejné a můstek je vyvážen. Pokud se v tomto okamžiku objeví i malá hodnota napětí na vstupu RF-IN, můstek se rozbíjí. Příchozími stroboskopickými pulsy je tedy obvod S/H otevřen na velmi krátkou dobu, po kterou se odečítá hodnota vstupního vysokofrekvenčního napětí. Rezistor  $R1$  a kondenzátor  $C1$  pak slouží k prodloužení získaného signálu v čase na měřitelnou úroveň. Mezifrekvenční signál z výstupu IF-OUT následně přichází přes impedanční oddělení a zesílení na vstup A/D převodníku.



Obr. 1. 23: Vzorkovací obvod buzený stroboskopickými pulsy [11]

## 1.8 A/D převodníky

Převodníky A/D jsou zařízení, která slouží k převodu analogového signálu na signál číslicový. Analogový signál, který je převáděn, je obvykle napětí a po převodu na digitální tvar je definován hodnotou odpovídající danému počtu bitů, kterou převodník disponuje.

Převod vstupní analogové veličiny na digitální je prováděn ve dvou fázích. Nejprve se vstupní signál periodicky vzorkuje vzorkovací frekvencí (vzorkovací rychlost), čímž se získává sled úzkých impulsů, jejichž amplituda odpovídá analogovému signálu v příslušných časových okamžicích. Při vzorkování musí být splněn již uváděný Nyquistův teorém, aby nedocházelo k aliasingu, tedy ke ztrátě informace v důsledku překrytí spekter dvou sousedních period. Následně se získané amplitudy kvantují a tím dostávají číslicový tvar, jelikož je jednotlivým vzorkům přiřazena diskrétní hodnota.

### 1.8.1 Vybrané parametry A/D převodníků

- **Rychlost převodu**

Obvykle bývá shodná se vzorkovací rychlostí, jelikož rychlost vzorkování vyplývá z nejkratší možné doby převodu.

- **Rozlišovací schopnost**

Rozlišovací schopnost, jíž daný převodník disponuje, je určena počtem úrovní, na které je rozdělen rozsah dovoleného vstupního napětí. Se zvyšující se rozlišovací schopností však klesá rychlost převodu.

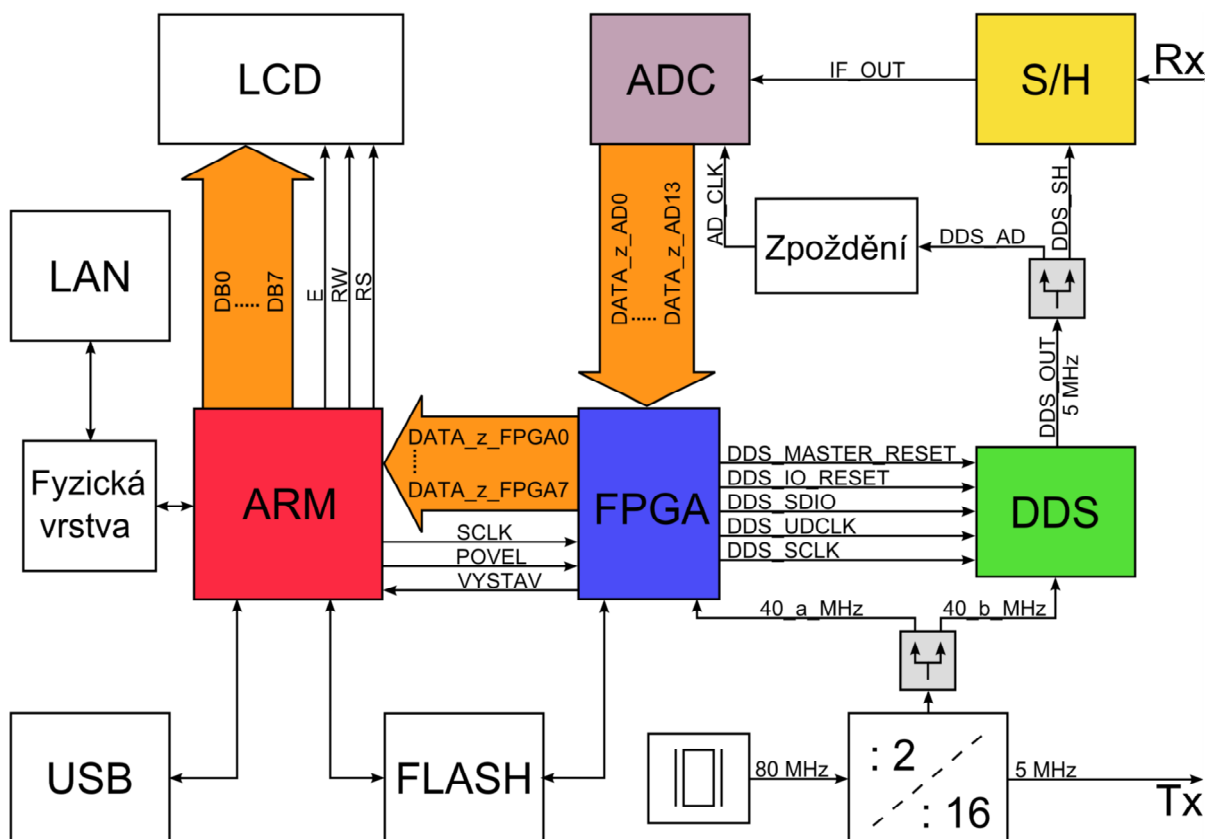
- **Kvantizační chyba**

Kvantizační chyba nebo také kvantizační šum vzniká konečným počtem bitů při kvantování vstupního signálu do  $2^N$  úrovní (kde  $N$  je počet bitů, jimiž převodník disponuje). Vzniklá chyba může dosahovat hodnoty, která je rovna  $\pm 0,5$  LSB bitu výstupního slova. Jedná se prakticky o rozdíl mezi vstupním a výstupním signálem kvantizačního obvodu.

## 2 Experimentální část

### 2.1 Koncepce zařízení

Struktura celého zařízení je zobrazena na Obr. 2. 1.



Obr. 2. 1: Blokové schéma přijímače

Hlavním řídicím obvodem je procesor typu ARM, který konfiguruje hradlové pole FPGA a zobrazuje změřená data na LCD, informující uživatele o správné detekci referenčního UWB signálu. Konfigurace procesoru je prováděna přes integrované rozhraní USB, přičemž FPGA je programováno přes fyzickou vrstvu LAN s pomocí externí paměti FLASH. Obvod FPGA řídí syntezeátor DDS, přijímá data od A/D převodníku a posílá do procesoru naměřené hodnoty charakterizující pozici přijímaného UWB pulsu v čase. Spolu s DDS pracuje na hodinovém kmitočtu 40 MHz, získaným děličkou z 80 MHz krystalu. Děličkou je také vytvořen kmitočet 5 MHz. Ten je využit pro generování referenční frekvence (výstup Tx), která slouží jako vstupní TTL signál pro generátor UWB pulsů při testování zařízení. Výstupní vysokofrekvenční referenční signál z generátoru je přiváděn na širokopásmový vzorkovací obvod (vstup Rx). Z výše uvedeného vyplývá, že přijímač je naladěn pro

přijem referenčního kmitočtu 5 MHz a to obvodem DDS, který je na tuto frekvenci nastaven a vykonává na ní časový posun pro vzorkovací obvod. Stejně tak převodník A/D pracuje s tímto kmitočtem jako s hodinovým.

## **2.2 Návrh obvodu procesoru**

Procesor v zařízení zastává hlavní komunikační člen s počítačem při programování softwaru. Dále slouží pro konfiguraci hradlového pole FPGA a ovládání s ním spjatých komponent - externí FLASH paměť a fyzická vrstva Ethernetu. S ohledem na samotnou funkci přijímače procesor ovládá LCD displej, který zobrazuje specifické hodnoty získané z detekovaného referenčního UWB signálu. Řízení displeje pomocí hradlového pole FPGA se ukázalo jako zbytečně náročné a programově zdouhavé. Proto jsou naměřené hodnoty vysokou rychlostí zpracovány v FPGA a následně posílány do procesoru, díky kterému jsou efektivněji pomocí LCD displeje zobrazeny.

Pro přijímač byl vybrán procesor AT91SAM7X256, který se proti procesoru použitým v [13] liší velikostmi pamětí, rozhraním LAN a počtem uživatelských I/O pinů.

### **2.2.1 Parametry procesoru AT91SAM7X256**

Jak bylo výše uvedeno, jedná se o obvod, který spadá do rodiny ARM7 a řady TDMI (T - Thumb architecture extension, D - Debug extension, M - enhanced Multiplier, I - Embedded ICE macrocell extension). Procesor podporuje dvě instrukční sady. První sadou je soubor výkonných 32-bitových ARM instrukcí a druhou podporovanou sadou jsou komprimované 16-bitové Thumb instrukce.

Dále disponuje integrovanou pamětí FLASH o velikosti 256 kB, která je organizovaná jako 1024 stránek po 256 bajtech a lze k ní přistupovat rychlostí 30 MHz. Procesor také obsahuje paměť SRAM, jejíž velikost je 64 kB a lze k ní přistupovat maximální rychlostí, kterou je procesor taktován. Maximální taktovací kmitočet je 55 MHz, přičemž lze dosahovat výpočetního výkonu až 0,9 MIPS/MHz. Zaváděcí program je obsažen v paměti ROM.

Externím rozhraním procesoru pro programování je integrovaná jednotka USB 2.0, která je schopná pracovat v režimu Full Speed s rychlostí 12 Mb/s. Procesor lze

mimo USB programovat i pomocí obsažené jednotky JTAG. Další integrované externí rozhraní jsou Ethernet, sběrnice CAN, SPI, I2S a TWI. Pro generování signálu PWM je zde obsažena i 16-bitová čtyřkanálová jednotka.

Mikroprocesor dále obsahuje 8-kanálový 10-bitový A/D převodník, periférii Watchdog, kontroléry pamětí, přerušení a další viz [13]. V procesoru jsou taktéž obsaženy tři 16-bitové čítače/časovače.

### 2.2.2 Obvodové řešení

Schéma realizovaného obvodu je uvedeno na Obr. 2. 2. Na napájecí vstupy procesoru je přivedeno napětí 3,3 V, které je blokováno u každého napájecího pinu keramickými kondenzátory. Napěťový výstup VDDOUT slouží pro napájení jádra procesoru a PLL, a to napětím 1,85 V, které je získáno interním regulátorem z napájecího napětí 3,3 V.

Procesor je taktován krystalem 18,432 MHz na pinech XIN a XOUT, který jen dán požadavkem jednotky USB pro využití funkce Full Speed. Dalším požadavkem pro realizaci této funkce je hodinový kmitočet UDPCCK o hodnotě 48 MHz, jenž je generován z PLL. Toto nastavení se provádí na pinu PLLRC. Samotný USB konektor je připojen na piny DDM a DDP.

Programování v tomto obvodu probíhá již jednodušším způsobem, než v případě procesoru použitým v [13], kde se prováděl zdlouhavý postup s pinem TST pro nahrání Bootloaderu. V tomto případě stačí pro programování pouze resetovat pin ERASE, a to jen na několik ms. Programová paměť je tímto okamžitě připravena pro naprogramování procesoru. Pro kontrolu úspěšného nahrání programu slouží *LED1* přivedená na I/O pin PA2.

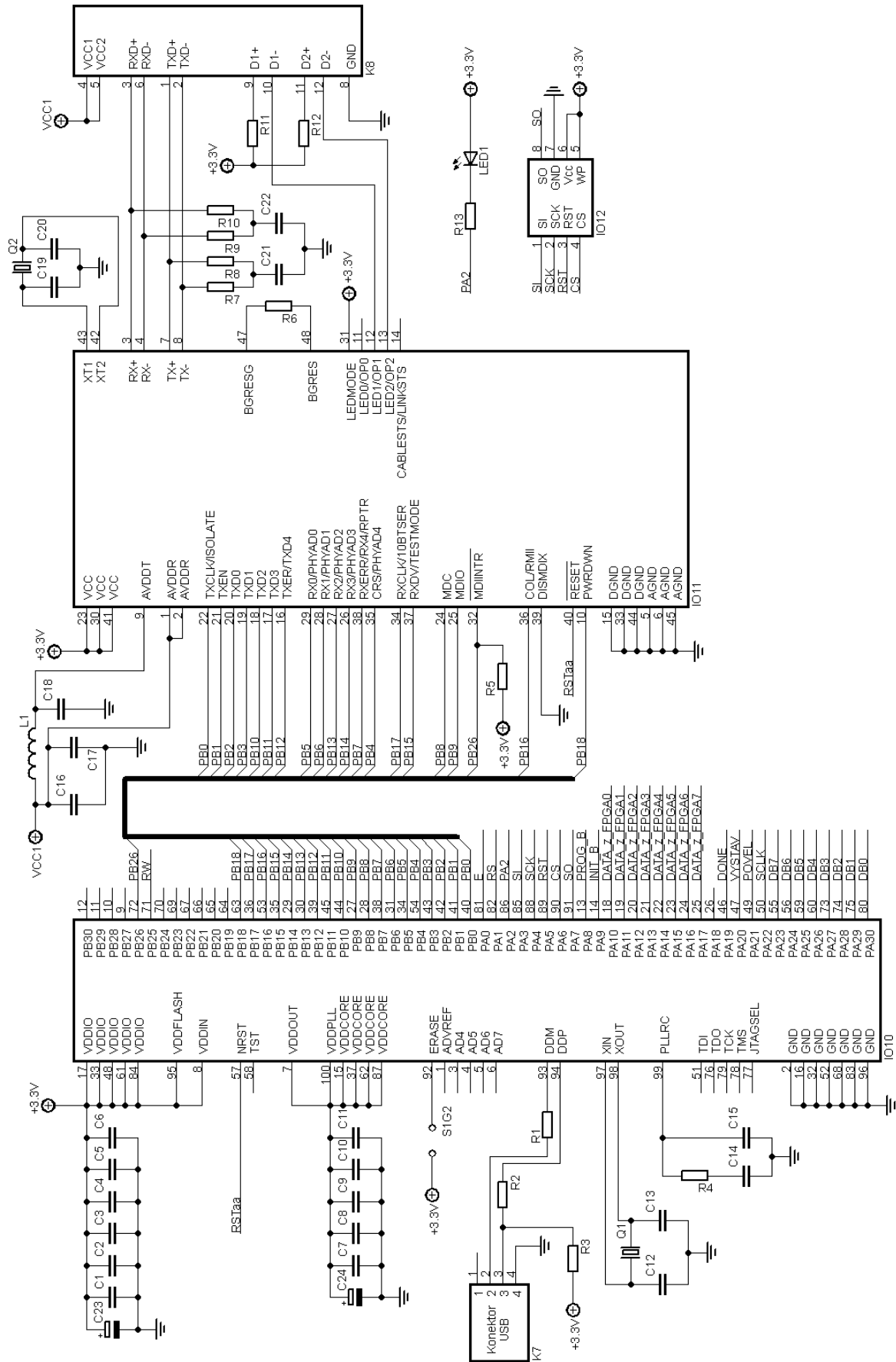
K procesoru je připojena externí paměť FLASH AT45DB021B, která slouží pro uložení konfiguračních dat pro FPGA. Její ovládací piny jsou připojeny na I/O piny dle schématu. Pro programování hradlového pole je k procesoru připojena fyzická vrstva DM9161, k níž je přiveden klasickým způsobem konektor LAN. Fyzická vrstva je taktována krystalem o hodnotě 25 MHz na pinech XT1 a XT2, což umožňuje rychlost komunikace 100 Mbps. Řídící piny fyzické vrstvy jsou přivedeny na většinu I/O pinů PB.

Uživatelské I/O piny označené jako DATA\_z\_FPGA0 - DATA\_z\_FPGA7 slouží pro příjem naměřených dat z FPGA. Pro komunikaci zde slouží piny označené jako POVEL a SCLK. Pin VYSTAV signalizuje připravenost FPGA k posílání dat.

K I/O portům procesoru označeným jako DB0 - DB7, E, RS a RW je připojen LCD displej CM1640 přes 16ti-pinový datový konektor. Jedná se o modul, který mimo displej obsahuje i svoji desku plošných spojů. Displej disponuje čtyřmi řádky, přičemž každý má 16 znaků. Pomocí pinů DB0 - DB7 lze nastavit požadovaný znak nebo určitou funkci displeje. Pin RW slouží k výběru operací čtení nebo zápisu, E k potvrzení a RS k výběru mezi instrukcí nebo daty. Různé kombinace využitých pinů pak mají specifické funkce, jako je např. zobrazení kurzoru, jeho posun při výpisu znaku a další. Modul displeje je uchycen na pomocnou desku plošných spojů viz příloha 5.

Hodnoty součástek pro realizaci obvodu procesoru jsou uvedeny v Tab. 2. 1.





Obr. 2. 2: Schéma zapojení procesoru

<i>IO10</i>	AT91SAM7X256
<i>IO11</i>	DM9161
<i>IO12</i>	AT45DB021D
<i>K8</i>	7499011222
<i>LED1</i>	HSMG-C670
<i>L1</i>	BLM41PG600SH1
<i>Q1</i> [MHz]	18,432
<i>Q2</i> [MHz]	25
<i>C1-C10, C16, C17, C18, C21, C22</i> [nF]	100
<i>C23, C24</i> [μF]	100
<i>C12, C13</i> [pF]	27
<i>C14</i> [nF]	10
<i>C15</i> [nF]	1
<i>C19, C20</i> [pF]	22
<i>R1, R2</i> [Ω]	27
<i>R3, R4</i> [kΩ]	1,5
<i>R5</i> [kΩ]	10
<i>R6</i> [kΩ]	6,8
<i>R7 - R10</i> [Ω]	51
<i>R11, R12</i> [kΩ]	1
<i>R13</i> [Ω]	270

Tab. 2. 1: Hodnoty součástek realizovaného obvodu procesoru

## 2.3 Návrh obvodu FPGA

Hradlové pole v přijímači vykonává hlavní logickou funkci, jelikož zpracovává změřená data z A/D převodníku. Přitom řídí časový posun obvodu DDS a zároveň odesílá zpracovaná data do procesoru. Tyto operace se provádí v co nekratším možném čase a byl pro ně vybrán obvod FPGA XC3S50 v pouzdře TQ144.

### 2.3.1 Parametry hradlového pole XC3S50

Tento obvod spadá do rodiny SPARTAN III a je vyráběn firmou Xilinx, která navíc poskytuje volně stažitelné vývojové prostředí ISE pro realizaci potřebných aplikací.

FPGA obsahuje 32-bitový softwarový RISC procesor MicroBlaze a 8-bitový softwarový kontrolér PicoBlaze. Jejich použití je však náročné, proto je v zařízení využíván zmíněný procesor ARM7.

Hradlové pole stejně jako procesor ARM obsahuje rozhraní pro konfiguraci pomocí JTAG jednotky.

Obvod obsahuje 4 bloky RAM, přičemž každá má svoji násobičku. Dále využívá 192 programovatelných logických bloků PLB a tabulky LUT mají celkovou kapacitu 12,288 kb. Pro ukládání dat slouží blokovaná paměť RAM, která disponuje kapacitou 73,728 kb.

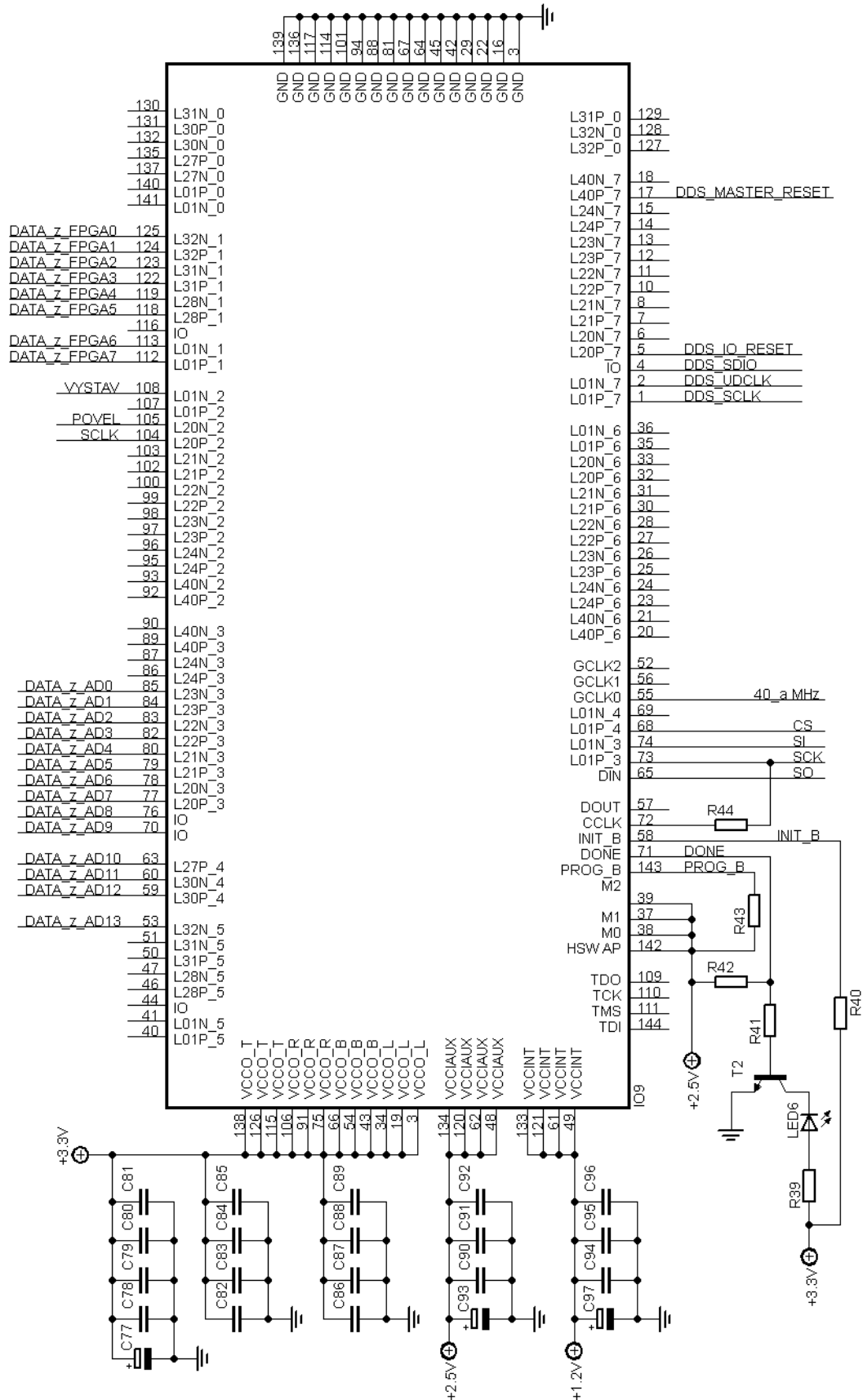
FPGA v použitém pouzdře TQ144 poskytuje 97 uživatelských pinů, které jsou uspořádány do osmi skupin (bank). Jednotlivé banky obsahují průměrně 15 pinů, přičemž Každá skupina obsahuje vždy dva piny GND a 2 napájecí piny VCCO.

### 2.3.2 Obvodové řešení

Schéma obvodového řešení je zobrazeno na Obr. 2. 3. FPGA vyžaduje 3 druhy napájení, a to 3,3 V (piny VCCOX), 2,5 V (piny VCCAUX) a 1,2 V (piny VCCINT). Všechny napájecí piny těchto různých napětí jsou blokovány kondenzátory dle schématu. Hodnoty použitých součástek jsou uvedeny v Tab. 2. 2.

Zdroj taktovacího signálu 40 MHz, který je popsán v následující kapitole, je přiveden na pin 55. Na 14 pinů FPGA jsou paralelně přiváděny data z převodníku pod označením DATA\_z\_AD0 - DATA\_z\_AD13 přes 20-pinový datový konektor. Dále jsou tímto způsobem na piny hradlového pole přivedeny signály pro konfiguraci a řízení obvodu DDS pod označením DDS\_MASTER\_RESET, DDS\_IO\_RESET, DDS\_UDCLK, DDS\_SDIO, DDS\_SCL. Požadovaná změřená data se z FPGA posílají do procesoru přes piny DATA\_z\_FPGA0 - DATA\_z\_FPGA7, přičemž komunikace mezi obvody je řešena pomocí pinů SLCK a POVEL. Pin VYSTAV signalizuje připravenost FPGA k posílání dat.

Programování FPGA se provádí pomocí vstupu PROG\_B, datového vstupu DIN a hodinového vstupu CCLK. Programování se inicializuje pinem PROG\_B. To je krátce potvrzeno log. 0 na výstupu INIT\_B, v případě chyby je INIT\_B trvale nastaven do nízké úrovně. Následně se nahrají konfigurační data z externí paměti FLASH přes CCLK a DIN. Pokud proběhlo programování bez chyby, rozsvítí se *LED1* na pinu DONE.



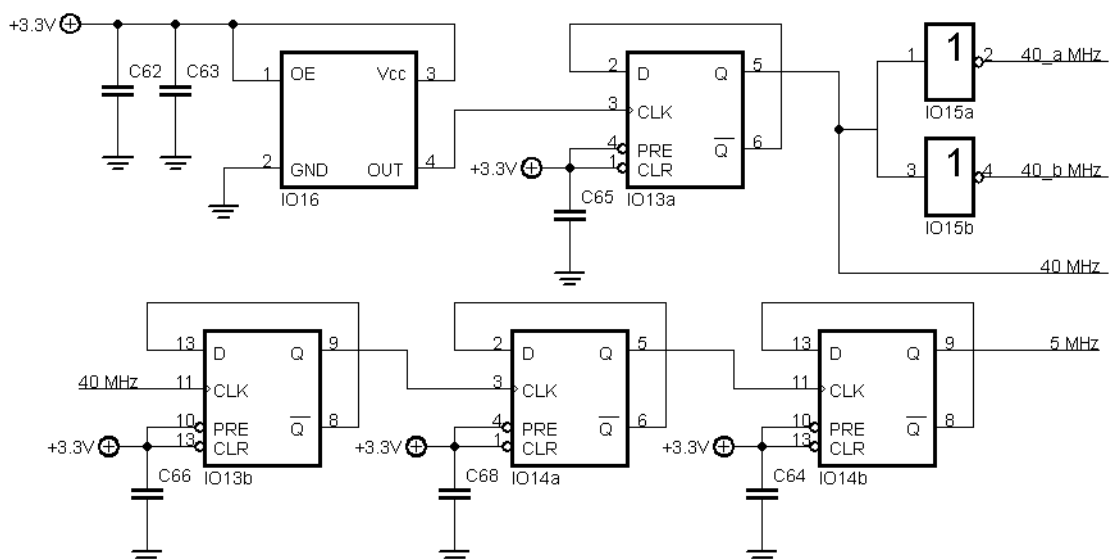
Obr. 2. 3: Schéma zapojení hradlového pole FPGA

<i>IO9</i>	XC3S50-4TQG144
<i>T2</i>	BC848C
<i>LED6</i>	HSMG-C670
<i>C79 - C96</i> [nF]	100
<i>C77, C93, C97</i> [μF]	10
<i>R39</i> [Ω]	470
<i>R40, R41, R42, R43</i> [kΩ]	10
<i>R44</i> [Ω]	68

Tab. 2. 2: Hodnoty součástek realizovaného obvodu FPGA

### 2.3.3 Zdroj hodinového signálu

V předešlém pojednání bylo uvedeno, že je FPGA taktováno kmitočtem 40 MHz. Ve schématu na Obr. 2. 4 je zobrazeno konkrétní řešení zdroje hodinového signálu. Požadovaný kmitočet je realizován z 80 MHz krystalu EG-2002CADCH. K vytvoření polovičního kmitočtu je použito klasického zapojení D klopného obvodu 74AC74. Ten je následně rozdělen hradly 74HC04 na dva signály. Signál 40\_a\_MHz slouží pro taktování obvodu FPGA a signál 40\_b\_MHz je použit pro syntezátor DDS. Dále je nutné vyvést pro dočasnou synchronizaci přijímače signál 5 MHz, který slouží jako vstup pro generátor UWB pulsů. Proto je výstup 40 MHz dále přiveden na druhý obvod 74AC74 a následně ještě na dvojici obvodů 74HC74. Takto získaný kmitočet o hodnotě 5 MHz je vyveden pomocí SMA konektorů a propojovacího VF kabelu na čelní panel přijímače označený popiskem Tx (pro funkci přijímače však nemá využití a používá se pouze k testovacím a ladícím účelům). Hodnoty součástek zdroje hodinového signálu jsou v Tab. 2. 3.



Obr. 2. 4: Schéma zapojení zdroje hodinového signálu

<i>IO12</i>	EG-2002CADCH
<i>IO13a, IO13b</i>	74AC74
<i>IO14a, IO14b</i>	74HC74
<i>IO15a, IO15b</i>	74HC04

Tab. 2. 3: Hodnoty součástek realizovaného zdroje hodinového signálu

## 2.4 Návrh obvodu DDS

Z teoretické části vyplývá, že pro změnu časového posunu je nejvhodnější použít syntezátor DDS. Pro realizaci byl použit na základě předchozích zkušeností [13] syntezátor AD9852 od firmy Analog Devices.

### 2.4.1 Parametry syntezátoru AD9852

Obvod AD9852 se vyznačuje vysokou vnitřní rychlostí a výkonným D/A převodníkem, který tvoří digitálně programovatelnou, syntezátorovou funkci a vytváří programovatelný sinusový (kosinusový) výstup, jenž má stabilní frekvenci, fázi i amplitudu.

Struktura obvodu umožňuje generování výstupních signálů o frekvenci až 150 MHz. Výstupní sinusový signál je možné převést pomocí vnitřního ultrarychlého komparátoru na obdélníkový průběh. Obvod obsahuje dvojici 14-bitových programovatelných registrů fáze a 48-bitových programovatelných registrů frekvence. Registry fáze mimo jiné umožňují binární PSK (BPSK) a pro vyšší PSK lze pro změny fáze využít I/O rozhraní. Obvod také obsahuje dva 12-bitové sinusové D/A převodníky. Pro kompenzaci závislosti  $\text{sinc}(x)$  na výstupech D/A převodníků je možné zvolit funkci inverzního sinc filtru, který koriguje pokles amplitudy. 12-bitový kontrolní D/A převodník při konfiguraci s komparátorem usnadňuje ovládání statické střídavy pro aplikace s vysokými rychlostmi hodinových signálů. Obvod dále obsahuje 12-bitovou digitální násobičku, která umožňuje přesné řízení amplitudy výstupního signálu ze sinusového D/A převodníku. Možné jsou také realizace AM, PM, FM, a to i ve tvaru klíčování.

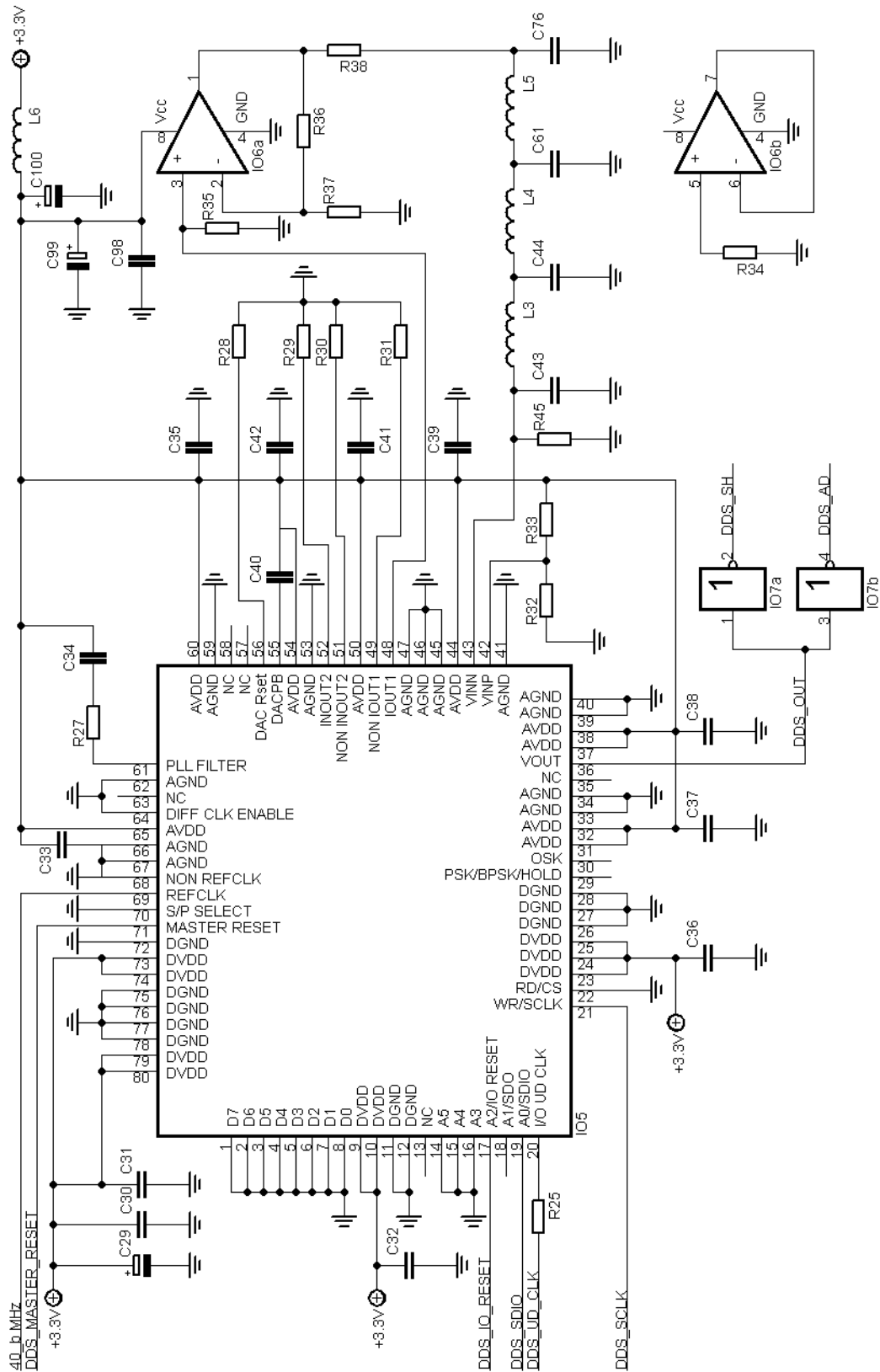
Významnou funkcí obvodu je schopnost násobit referenční kmitočet (REFCLK), tedy frekvenci přivedeného hodinového signálu, a to 4 až 20 krát. Při použití krystalu např. s hodnotou 15 MHz, lze tento kmitočet vynásobit maximálně na 300 MHz, což je maximální vnitřní hodinová frekvence obvodu. Samotná násobička je řešena vnitřním obvodem PLL.

Mezi další funkce obvodu patří lineární/nelineární rozmítání frekvence (FM chirp) nebo možnost programovatelné rampy amplitudy a frekvence při amplitudovém a frekvenčním klíčování.

Pro programování obvodu lze využít dvě řídicí rozhraní, a to buď sériové 10 MHz nebo 8-bitové paralelní 100 MHz.

#### **2.4.2 Obvodové řešení**

Schéma realizovaného obvodu je uvedeno na Obr. 2. 5. Obvod je napájen z napájecího zdroje zařízení napětovou hladinou 3,3 V, přičemž všechny napájecí vstupy jsou filtrovány příslušnými kondenzátory. Hodinový kmitočet obvodu 40 MHz je přiveden z děličky na pin REFCLK. Jelikož se obvod programuje přes sériové rozhraní, jsou na vstupní piny obvodu FPGA vyvedeny dle schématu piny DDS\_MASTER\_RESET, DDS\_IO\_RESET, DDS\_UDCLK, DDS\_SDIO, DDS\_SCL pomocí 20-pinového datového konektoru. Datové piny syntezátoru D0 až D7 pro paralelní programování jsou přivedeny na zem, stejně tak pin S/P SELECT pro vybrání sériového režimu programování. RC filtr pro PLL je umístěn na pinu 61. Komplementární výstup kosinového převodníku a oba výstupy kontrolního D/A převodníku jsou přes 50  $\Omega$  připojeny na zem. Unipolární výstup z kosinusového D/A převodníku (pin 48) je přiveden na operační zesilovač, jehož výstup je zesílen, upraven pasivním filtrem a přiveden na invertující vstup vysokorychlostního komparátoru (pin 43). Na neinvertující vstup komparátoru (pin 42) je přivedeno pomocí odporového děliče  $R_{32}$ ,  $R_{33}$  napětí 673 mV. Výstup z AD9852 DDS\_OUT je rozdělen a přiveden na 2 hradla obvodu 74HC04. Negované výstupy jsou následně vedeny přes SMA konektory na vstup A/D převodníku (DDS\_AD) a také na vstup generátoru stroboskopických pulsů (DDS\_SH) ve vzorkovači. Hradla jsou zde z důvodu impedančního oddělení, aby případných 50  $\Omega$  na vstupu výše zmíněných obvodů nezatěžovalo komparátor DDS. Hodnoty použitých součástek pro realizaci obvodu jsou uvedeny v Tab. 2. 4.



Obr. 2. 5: Schéma zapojení syntezátoru DDS



<i>IO5</i>	AD9852
<i>IO6a , IO6b</i>	AD8042
<i>IO7a , IO7b</i>	74HC04
<i>L3 , L4 , L5</i> [nH]	470
<i>L6</i>	BLM41PG600SH1
<i>C30 - C33 , C35 - C42 , C98</i> [nF]	100
<i>C29 , C99 , C100</i> [μF]	10
<i>C34</i> [nF]	10
<i>C43 , C76</i> [pF]	470
<i>C44 , C61</i> [pF]	56
<i>R25 , R32 , R36</i> [kΩ]	1
<i>R27</i> [kΩ]	1,2
<i>R33</i> [kΩ]	3,9
<i>R28 - R31 , R35</i> [Ω]	51
<i>R37</i> [Ω]	510
<i>R38</i> [Ω]	15

Tab. 2. 4: Hodnoty součástek realizovaného obvodu DDS

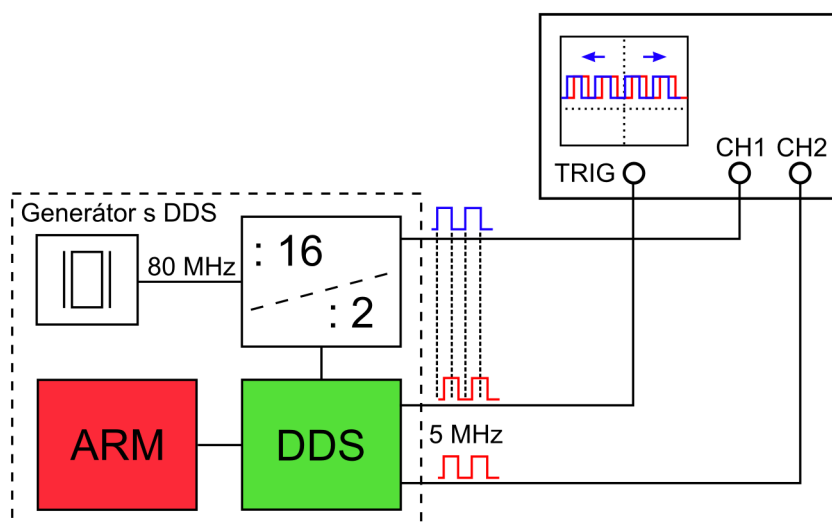
### 2.4.3 Řízení časového posunu procesorem ARM

Pro měření časového posunu byl využit generátor s DDS realizovaný v [13]. Generátor obsahuje již zmíněný procesor AT91SAM7S64. Obvody generátoru DDS jsou v [13] a v této práci totožné. Jelikož nebyl v generátoru připraven pevný kmitočet 5 MHz, byl pro kontrolu časového posunutí doplněn o externí děličku, která požadovaný kmitočet tvořila z 80 MHz krystalu. Jelikož byla dělička ke krystalu připájena standardním vodičem, průběh jejího výstupního kmitočtu obsahoval značně zákmity.

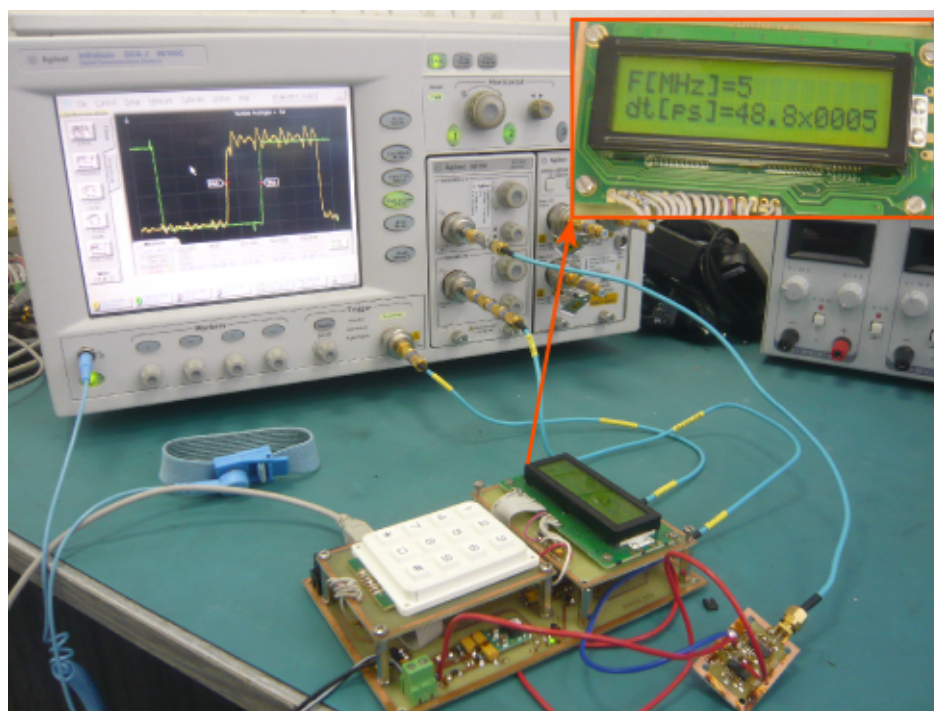
Kvůli realizaci časového posunu pomocí daného generátoru musela být provedena změna softwarové části zařízení. Generátor byl realizován pro ladění frekvence bez možnosti změny časového zpoždění. Proto byl přeprogramován tak, aby generoval výstupní frekvenci 5 MHz a pomocí klávesnice, jež sloužila k nastavení frekvence, bylo možné nastavovat velikost kroku (zpoždění) 12,2 ps, 48,8 ps, 195,2 ps atd. dle vztahu (1.23). Nastavení zpoždění buď probíhalo automaticky ve smyčce s opětovným přičítáním (odčítáním) daného kroku nebo bylo možné přejít k ručnímu ladění. S ohledem na to, že se vždy při zapnutí generátoru časový posun mezi frekvencí z DDS a kmitočtem děličky lišil, bylo automatické posouvání zpoždění využito (s vyšším krokem např. 292,8 ps) k nalezení pozice, kde se průběh z DDS a děličky překrýval, popřípadě byl v těsné blízkosti. Pro přesné překrytí obou kmitočtů se následně zjemnil krok např. na nejvyšší rozlišení (pro

kmitočet 5 MHz) 12,2 ps a bylo možné sledovat, že se při takřka přesném překrytí obou pulsů, po chvíli začaly v čase vůči sobě vzdalovat díky teplotním driftům součástek.

Blokové schéma měření časového posunu řízeného procesorem je na Obr. 2. 6. Důvodem pro řízení posunu takto upraveným generátorem bylo následné měření a ladění vzorkovače. Řízení časového posunu pomocí FPGA by bylo pro tento účel z hlediska ovládání nevhodné. Fotografie z měření zobrazuje Obr. 2. 7.



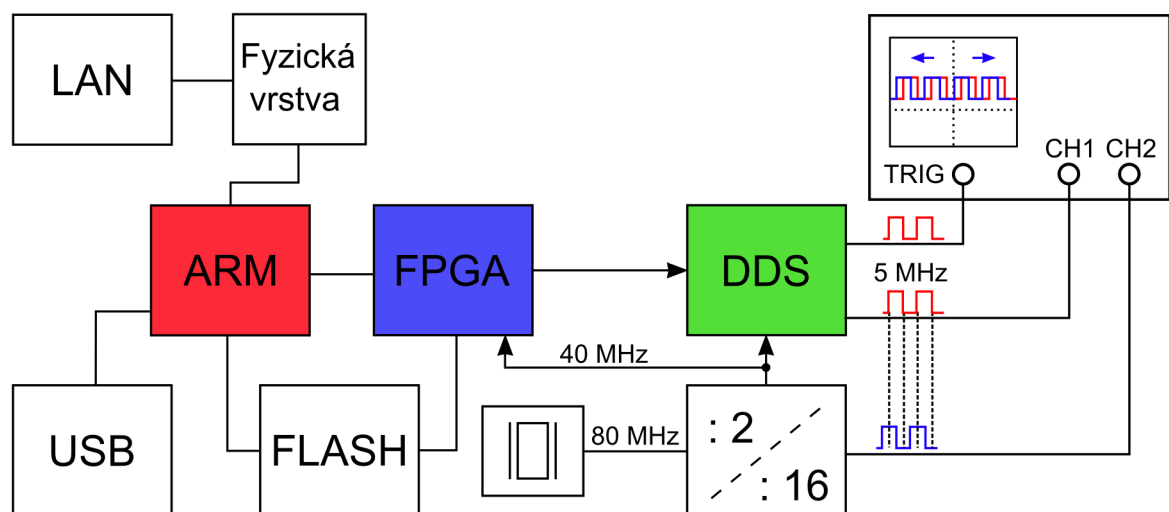
Obr. 2. 6: Blokové schéma měření časového posunu řízeného procesorem



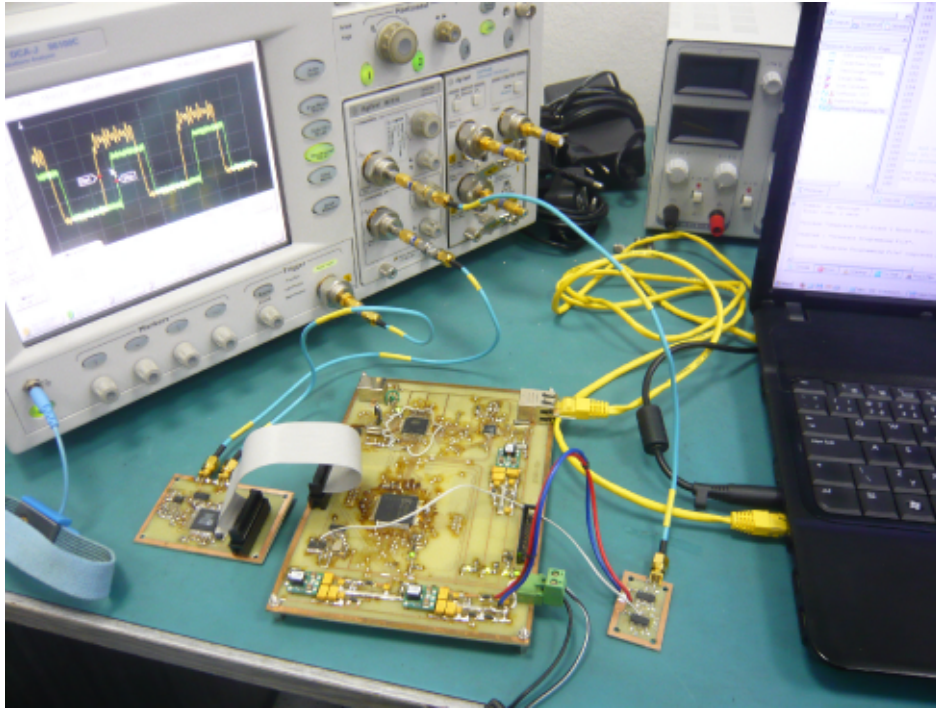
Obr. 2. 7: Měření časového posunu řízeného procesorem

### 2.4.4 Řízení časového posunu obvodem FPGA

V tomto případě byl obvod DDS řízen hradlovým polem FPGA. Programování obvodu se uskutečňovalo v jazyce VHDL, tudíž byl žádán odlišný přístup, než v předchozím případě, kde se proces změny časového posunu odehrával v programovacím jazyce C. V tomto kroku byl obvod DDS řízen tak, že automaticky měnil časový posun s nejmenším možným krokem, tedy 12,2 ps. Navíc se tato změna postupně prováděla se zvyšující se rychlostí, přičemž byla snaha nastavit minimální nutnou dobu pro nastavení DDS. Důvodem byl požadavek maximální rychlosti změny časového posunu pro následné použití při vzorkování UWB signálu. Při zkrácení čekacích smyček na nejmenší možné minimum však už nebylo možné na osciloskopu Agilent 86100C sledovat změnu posunu kmitočtu z DDS oproti průběhu z děličky. Proto byl k této struktuře připojen zrealizovaný vzorkovač. Ten následně potvrdil funkci obvodu DDS při minimálním čase potřebném ke změně zpoždění. Na obrazovce osciloskopu byl v tomto případě zhlédnutelný zákmit amplitudy stejnosměrné složky (viz Obr. 2. 13 v následující kapitole), který se periodicky opakoval a potvrdil tím správnou funkci detekce měřeného UWB pulsu. Takto připravený softwarový základ pro řízení DDS byl nadále rozšířen o algoritmus hledání specifických hodnot potřebných pro fázový závěs, příjem dat z A/D převodníku a komunikaci s procesorem ARM. Blokové schéma pro měření časového posunu řízeného hradlovým polem FPGA je uvedeno na Obr. 2. 8. Fotografie z průběhu měření je následně zobrazena na Obr. 2. 9.



Obr. 2. 8: Blokové schéma měření časového posunu řízeného pomocí FPGA

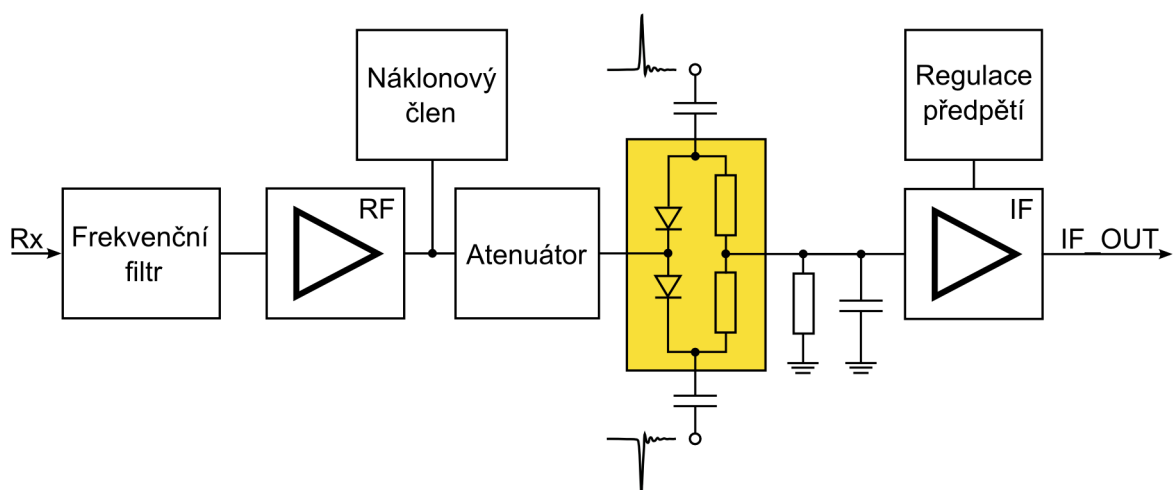


Obr. 2. 9: Měření časového posunu řízeného hradlovým polem FPGA

## 2.5 Návrh a měření vzorkovače

### 2.5.1 Obvodové řešení

Jelikož jsou na vzorkovač kladeny ty nejpřísnější požadavky, byla snaha jej navrhnout a naladit, takovým způsobem, aby vykazoval co nejlepší vlastnosti. Zapojení vychází z blokového schématu, které je zobrazeno na Obr. 2. 10.



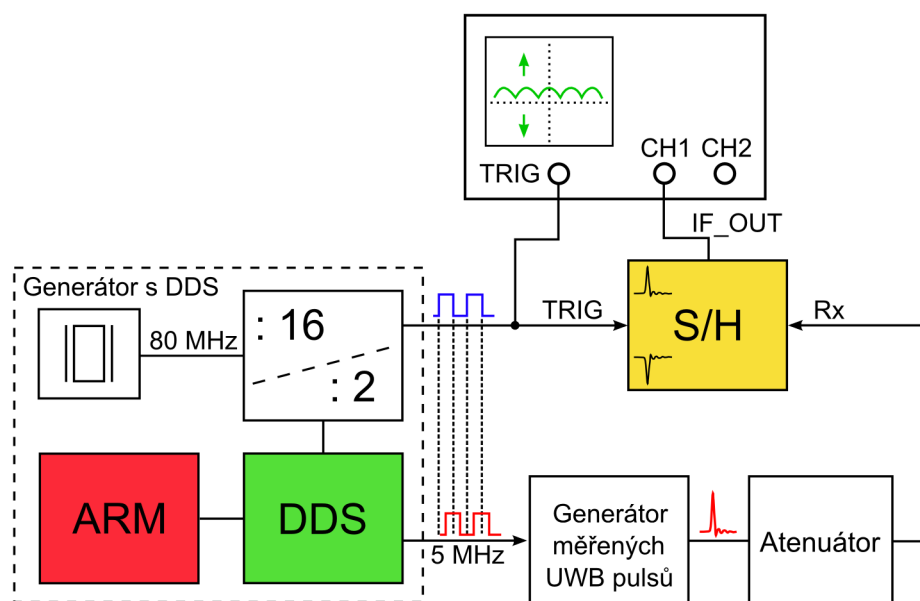
Obr. 2. 10: Blokové schéma vzorkovače

Z obrázku je patrné, že vysokofrekvenční referenční signál přichází na frekvenční filtr, jehož úkolem je omezit pásma GSM. Jedná se o keramický filtr, který

propouští pouze frekvence vyšší než 1220 MHz. Dále je signál přiveden na vysokofrekvenční monolitický zesilovač ERA-2SM, jenž byl použit v přijímači s přímou detekcí [1]. Tento zesilovač se aplikuje do struktury 50  $\Omega$  mikropáskového vedení stejně jako jemu předřazený frekvenční filtr. Pro snížení VF ztrát energie celého vzorkovače byl zvolen substrát AD450 od firmy Arlon. Vypočítaná šířka 50  $\Omega$  mikropáskového vedení je pro tento substrát 1,4 mm [1]. Pro dosažení vyrovnaného zisku, lineární fáze a tedy i vyrovnaného skupinového zpoždění je použit náklonový člen, přes který je zesilovač napájen. Následný keramický atenuátor omezuje vysoký zisk zesilovače a zlepšuje vstupní přizpůsobení obvodu S/H. Signál dále přichází na balanční zapojení dvou Shottkyho diod, na které jsou přiváděny stroboskopické pulsy opačných polarit. Zde je, dle popisu v teoretické části práce, získáno malé napětí, které je následným RC členem prodlouženo v čase. Mezifrekvenční zesilovač IF slouží jako impedanční přizpůsobení a také získaný signál zesiluje. Regulace předpětí slouží k nastavení velikosti stejnosměrné složky získaného IF signálu pro A/D převodník, jehož vstupní rozsah je 0 – 2 V.

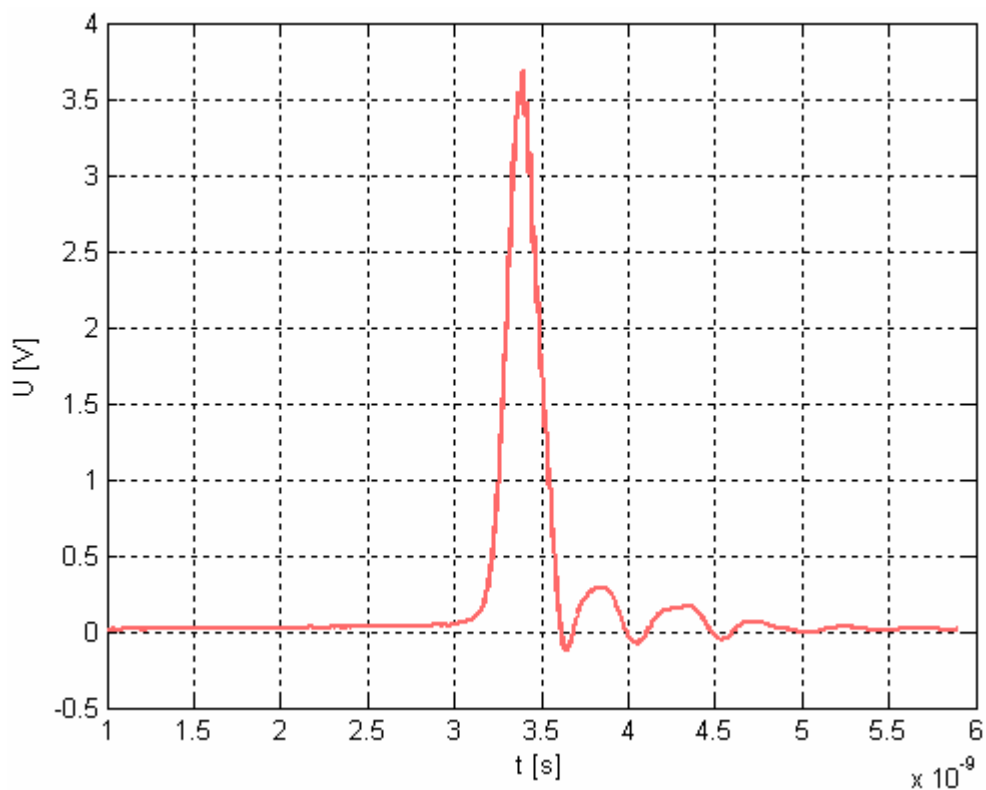
### 2.5.2 Měření vzorkovače

Blokové schéma pro měření vzorkovače je uvedeno na Obr. 2. 11. Pro měření byl využit již zmíněný generátor s DDS s upravenou (v předchozí kapitole popsanou) funkcí.



Obr. 2. 11: Blokové schéma měření vzorkovače

Z obrázku je zřejmé, že výstupní kmitočet z DDS (DDS\_SH viz Obr. 2. 5) je přiveden na generátor UWB pulsů. Tento generátor vychází ze zapojení prezentovaném v [11]. Proti uvedené verzi byl však nadále upravován a nyní stroboskopický puls dosahuje o 1,2 V vyšší amplitudy, a to při stejné šířce pulsu cca 400 ps. Generovaný puls je uveden na Obr. 2. 12 a následně slouží jako měřený referenční UWB signál. Totožný generátor je taktéž realizován v obvodu vzorkovače, přičemž způsob vytvoření stroboskopického pulsu opačné polarity je s drobnými úpravami obdobný, jako v [11]. Úprava se týkala především změny hodnot pasivních součástek v daném obvodu. Tím bylo docíleno menšího zpoždění mezi oběma komplementárními pulsy při zachování jejich amplitudy a šířky pulsu.



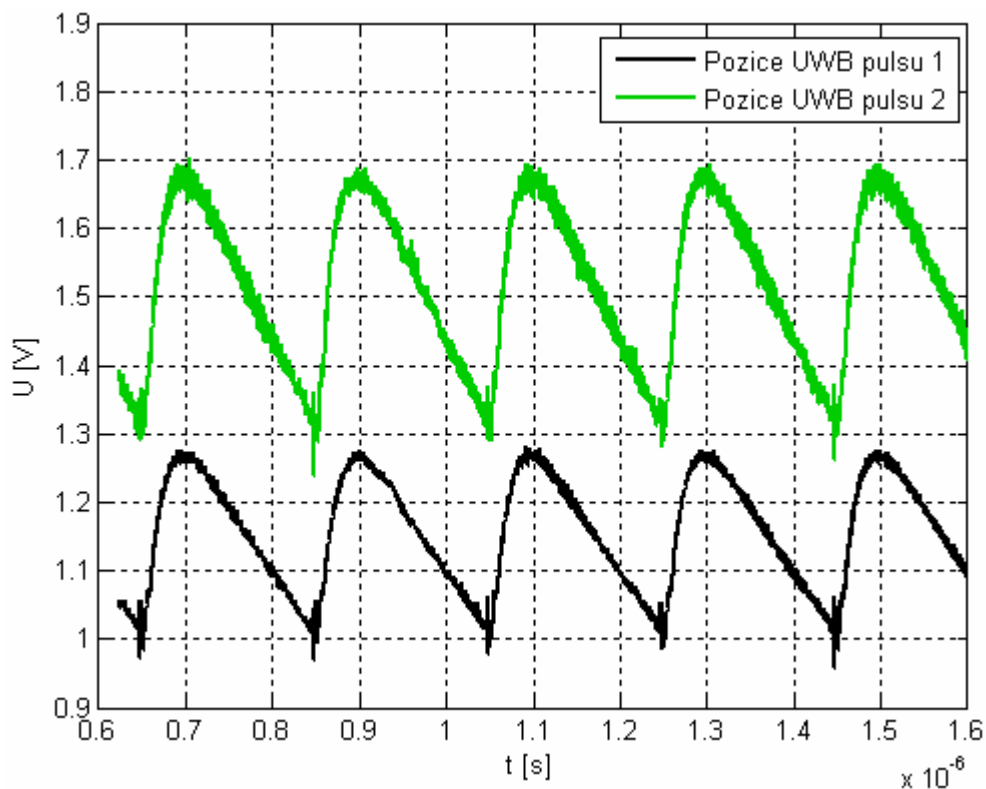
**Obr. 2. 12: Měřený UWB puls**

Jelikož je vzorkovač měřen přímou cestou pomocí VF kabelů a ne přes anténu, pro kterou je navržen, je nutné snížit amplitudu měřeného pulsu na vhodnou úroveň. Z tohoto důvodu je do cesty signálu vložen attenuátor. Při měření byly testovány různé hodnoty útlumu, aby byla zjištěna hranice, po kterou je díky vysoké amplitudě měřeného UWB pulsu vzorkovač přebuzen.

Vhodně utlumený měřený signál dále přichází na vysokofrekvenční vstup vzorkovače. Mezifrekvenční výstup vzorkovače je přiveden na měřicí kanál osciloskopu.

Aby bylo možné ověřit a následně změřit funkci vzorkovače, tedy na jeho výstupu pozorovat navzorkovaný UWB signál, z děličky je vyveden konstantní kmitočet pro trigger osciloskopu. Tento TTL signál také slouží jako buzení obvodu pro generování stroboskopických pulsů ve vzorkovači.

Při spuštění měřicího procesu lze na osciloskopu sledovat specifický průběh výstupního mezifrekvenčního signálu vzorkovače. Průběh odpovídá hodnotě poměru amplitudy odebraného vzorku měřeného signálu v daném časovém okamžiku vzorkování a naladěné stejnosměrné složce, kterou lze měnit odporovým trimrem v části vzorkovače nazvané regulátor předpětí. Na Obr. 2. 13 je uveden výstup vzorkovače ve dvou okamžicích, které odpovídají různým časovým posunům přivedeného UWB pulsu. Stejnosměrná složka je v tomto i následujících případech naladěna na střední hodnotu cca 1 V .



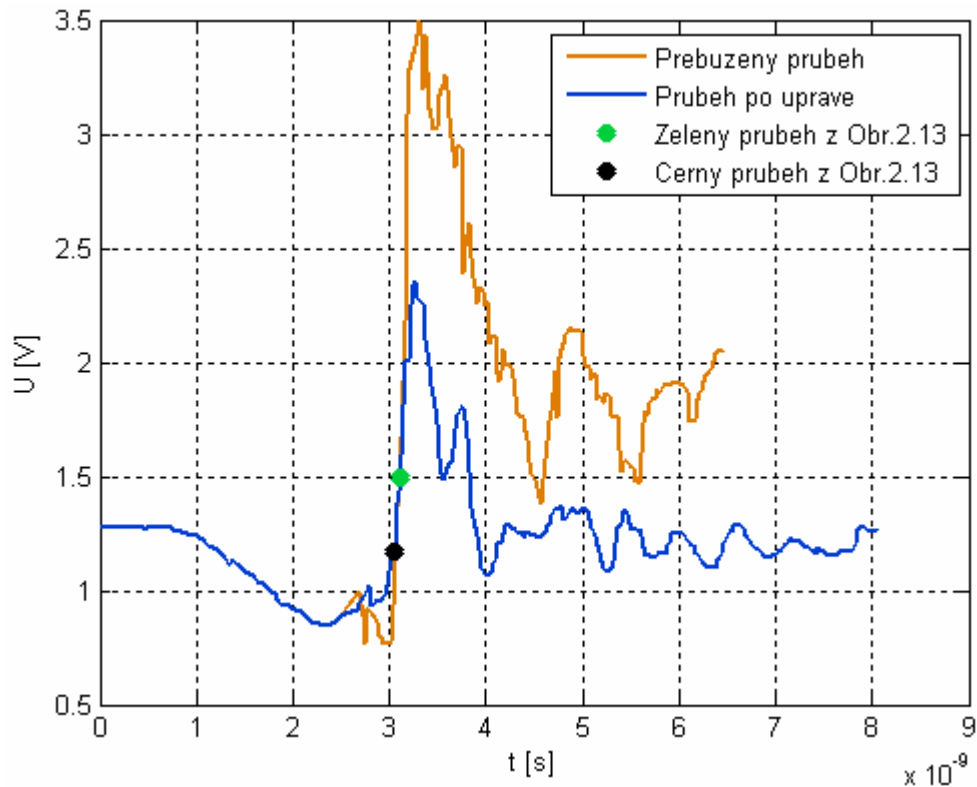
Obr. 2. 13: Výstup vzorkovače při různých posunech UWB pulsu

Při postupné změně časového posunu se posouvá pozice měřeného UWB pulsu v čase, přičemž vzorkovač je otevírán stroboskopickými pulsy stále ve stejný okamžik. Tím lze navzorkovat celý průběh měřeného UWB pulsu. Nejvyšší přesnost vzorkování lze docílit s krokem 12,2 ps. Jelikož samotný puls disponuje šířkou cca 400 ps, následné základy doznívají cca 1,5 ns a připočteme-li ještě alespoň 1 ns před a za pulsem, je nutné z celkové měřené doby 4 ns přes 300 krát změnit pozici časového posunu. To znamená zvolit na klávesnici generátoru s DDS časový posun o jeden daný krok, zaktualizovat průměrování na osciloskopu (samotné průměrování s hodnotou 256 trvá osciloskopu Agilent 86100C cca 15 s), uložit aktuální průběh (viz Obr. 2. 13) a to vše provést alespoň 300 krát pro jedno měření.

Z uvedených faktů vyplývá, že měření je časově náročné, proto byl k měření využit krok 24,4 ps, díky kterému lze časový úsek 4 ns pokrýt polovičním počtem kroků. Průběh celého navzorkovaného UWB pulsu s tímto nastavením je uveden na Obr. 2. 14 (oranžový průběh). Z tohoto průběhu je vidět zřetelné přebuzení vzorkovače, což se projevuje pomalým klesáním sestupné hrany navzorkovaného pulsu a tím i prodloužením samotné šířky pulsu. Prodloužení pulsu je však také způsobeno velmi pomalým měřením, které probíhalo několik hodin, a měřený puls se přitom celou dobu díky teplotnímu driftu pohyboval v čase. Tento jev byl posílen velmi pomalými změnami časového zpoždění, které nebyly navíc prováděny ve stejných časových intervalech. Z obrázku je také patrné, že kvůli přebuzení vzorkovače se po odeznění pulsu udržuje hodnota amplitudy na vysoké hodnotě (cca 1,8 V). Ta klesá velice pomalu během celé periody až do doby, kdy na vzorkovač přijde náběžná hrana pulsu následné periody.

Přebuzení vzorkovače bylo následně kompenzováno zvyšováním útlumu na vstupu vzorkovače. To však nepřineslo po několika měření žádně zlepšující se výsledky, a proto bylo provedeno snížení zesílení mezifrekvenčního zesilovače. Po této úpravě bylo také sníženo průměrování v osciloskopu na poloviční hodnotu (128) kvůli docílení vyšší rychlosti změny časového posunu. S ohledem na kontrolu eliminace přebuzení bylo odebráno přes 300 vzorků a tím byla při kroku 24,4 ps pokryta časová oblast 8 ns. Změřený průběh s upraveným zesílením mezifrekvenčního zesilovače je uveden na Obr. 2. 14 (modrý průběh).

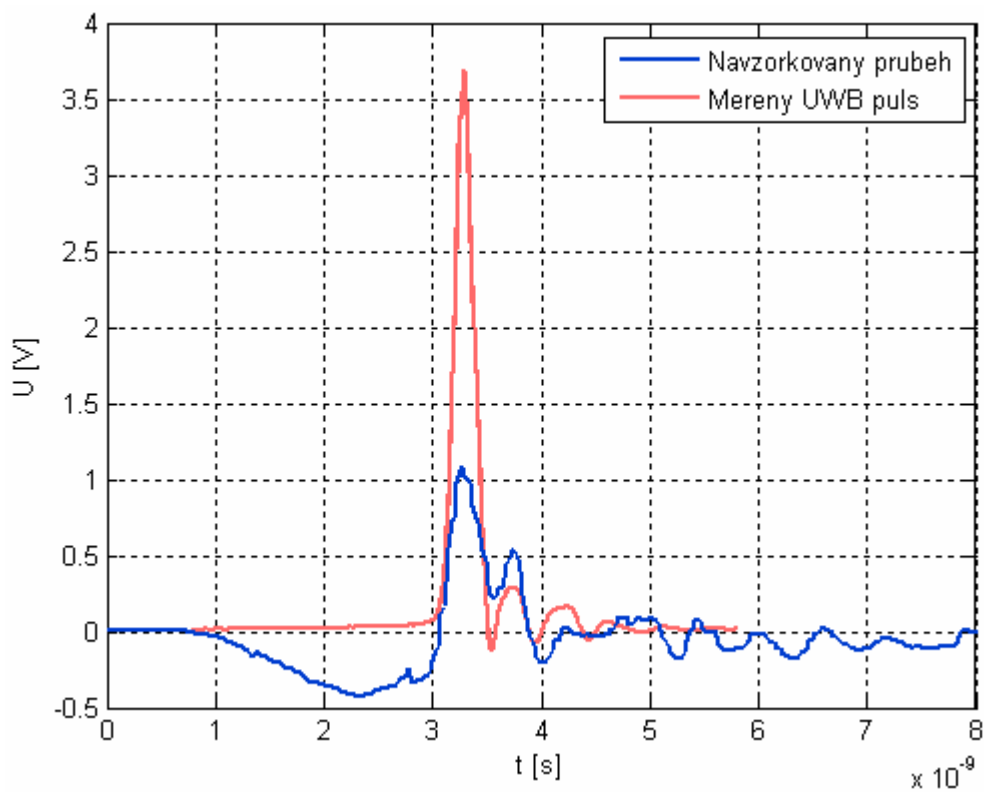




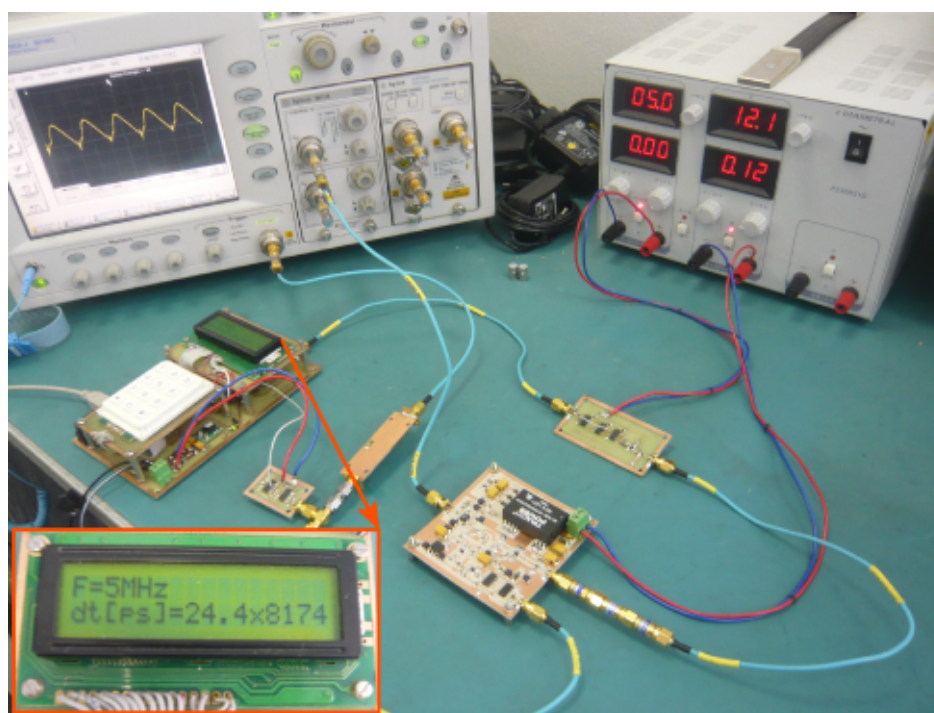
**Obr. 2. 14: Navzorkované průběhy UWB pulsu**

Z obrázku je patrné, že k přebuzení již nedochází, jelikož pokles amplitudy na sestupné hraně a po odeznění pulsu klesá nesrovnatelně rychleji než v předchozím případě. Při tomto měření bylo také snahou provádět změnu časového zpoždění, díky zkrácení doby průměrování, vyšší a přitom konstantní rychlostí.

Pro porovnání měřeného pulsu a navzorkovaného průběhu, je na následujícím Obr. 2. 15 snížena ss složka změřeného průběhu na hodnotu 0 V. Z obrázku vyplývá, že změřený puls vcelku odpovídá průběhu přijímaného UWB pulsu. Výrazný záporný zákmit je s největší pravděpodobností způsoben velice nekvalitním kmitočtem z děličky. Ten má totiž právě na náběžné hraně velice kmitavý charakter. Důvod realizace takto nevhodné provizorní děličky je uveden v kapitole (2.4.3). Fotografie z průběhu měření vzorkovače je následně uvedena na Obr. 2. 16.



Obr. 2. 15: Porovnání měřeného a navzorkovaného UWB pulsu



Obr. 2. 16: Měření vzorkovače

## 2.6 Návrh obvodu A/D převodníku

Pro převod analogového výstupního napětí mezifrekvenčního výstupu vzorkovače na digitální signál přicházející na hradlové pole FPGA byl vybrán obvod AD9248 od firmy Analog Devices.

### 2.6.1 Parametry převodníku AD9248-20

Tento dvoukanálový A/D převodník disponuje vzorkovací rychlostí až 20 MSPS. Oba kanály obsahují vlastní S/H zesilovač, přičemž každý kanál může pracovat na různé frekvenci. Vstupy převodníku jsou diferenciální, ale kanály mohou pracovat i v režimu proti zemi tzv. Single-Ended.

Maximální vstupní rozsah měřitelného napětí dosahuje 2 V, přičemž je tato hodnota uvažovaná mezi diferenciálními vstupy daného kanálu. Pokud se jedná o režim proti zemi, tato hodnota je stanovena na poloviční hodnotu, tedy 1 V. Integrovaná napěťová reference převodníku je 0,5 V, je zde možnost připojení externí reference, která může být maximálně 2 V.

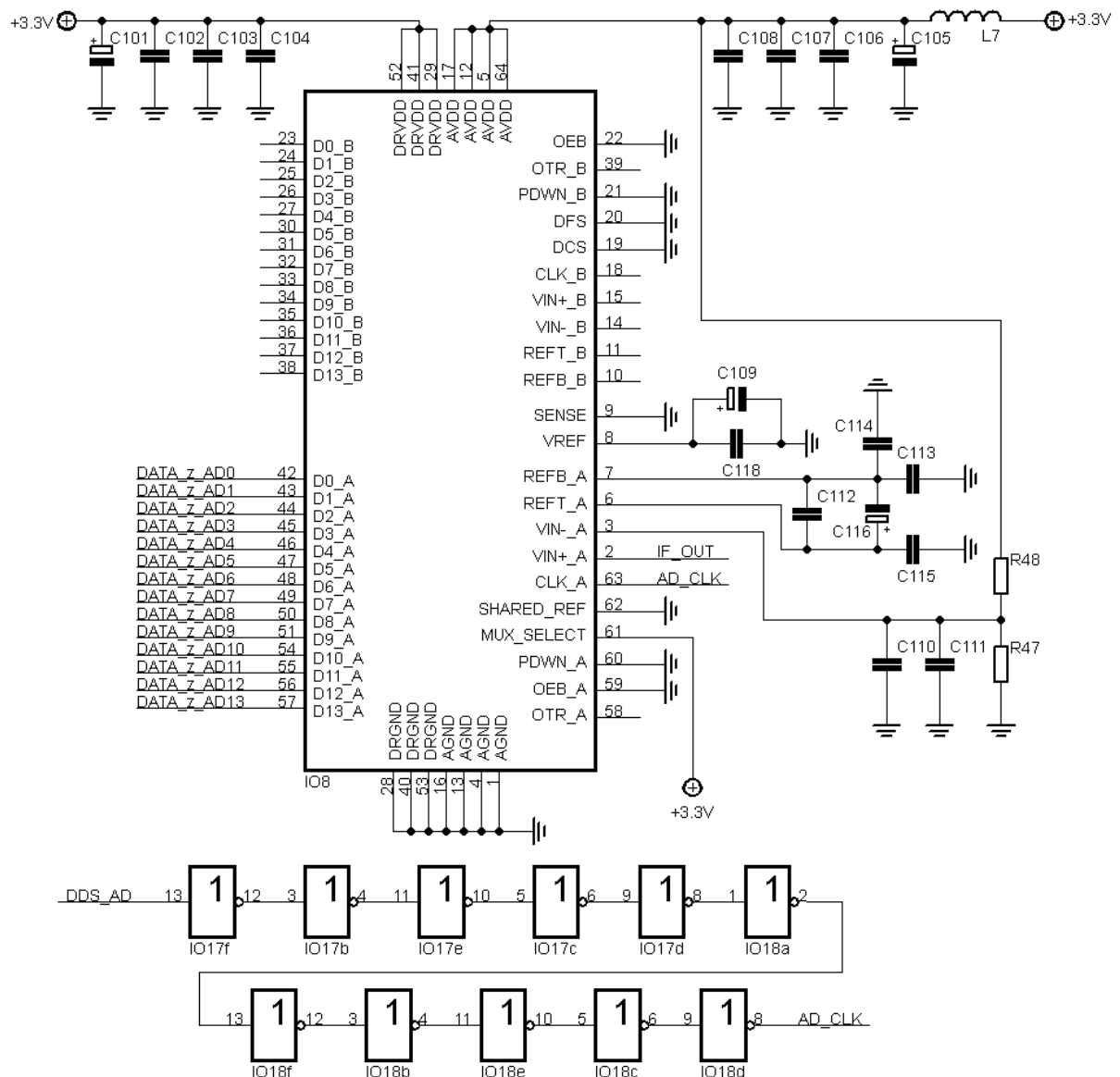
Rozlišovací schopnost převodníku je 14 bitů a výstupní tvar digitálního signálu může být buď klasický binární nebo ve dvojkovém doplňku. Každý kanál tedy obsahuje 14 samostatných výstupních pinů, přičemž lze využít i režim multiplexu. V tomto případě se na jednom výstupním portu střídají data obou kanálů.

Převodník obsahuje navíc obvody pro detekci a korelaci chyb, které slouží pro zpřesnění A/D převodu. Dále také využívá principu pipeline pro zrychlení převodu.

### 2.6.2 Obvodové řešení

Jelikož se jedná o obvod CMOS, je napájen napětím 3,3 V z napěťového zdroje zařízení. Analogové i digitální napájecí vstupy převodníku jsou filtrovány a blokovány danými kapacitami dle schématu uvedeném na Obr. 2. 17. Pro získání specifických informací vzorkovaného signálu postačí pouze jeden kanál převodníku, a proto jsou na vstupy FPGA vyvedeny pouze výstupy kanálu A (DATA\_z\_AD0 – DATA\_z\_AD13) pomocí 20-pinového datového konektoru. Zmíněná integrovaná reference převodníku je na svém výstupu VREF opatřena kapacitami a uzemněným pinem SENCE se potom nastavuje vnitřní reference na hodnotu 1 V. Referenční piny REFB a REFT kanálu A jsou přes kapacitu o hodnotě 100 nF propojeny a následně

blokovány uvedenou kombinací kondenzátorů. Toto nastavení definuje vstupní rozsah 0 – 2 V. Signál z mezifrekvenčního výstupu vzorkovače IF\_OUT je přiveden pomocí SMA konektoru na diferenciální vstup převodníku VIN+\_A. Hodinový signál pro řízení převodníku je získáván z výstupu syntezátoru DDS (DDS\_AD). Tento kmitočet přichází na propojení hradel 74HC04, což způsobuje zpoždění a zaručuje vybuzení převodníku pro odečet vzorků z měřeného mezifrekvenčního výstupu vzorkovače. Na hodinový vstup převodníku CLK\_A tedy přichází upravený kmitočet AD\_CLK o hodnotě 5 MHz. Hodnoty použitých součástek pro realizaci obvodu A/D převodníku jsou v Tab. 2. 5.



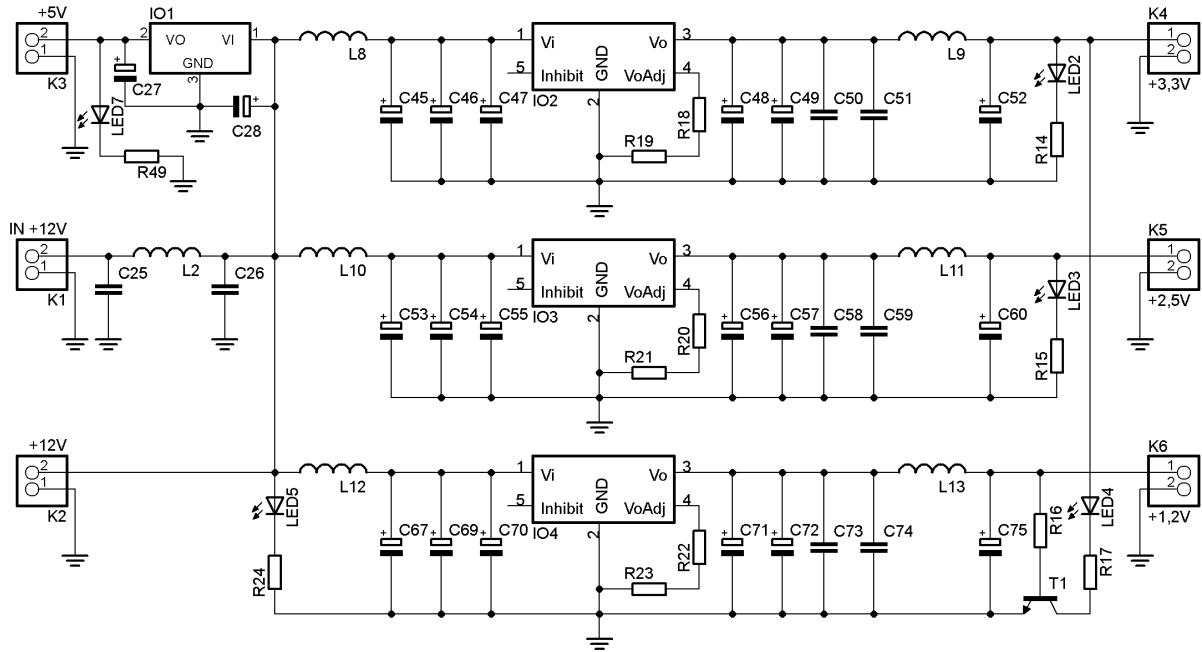
Obr. 2. 17: Schéma zapojení A/D převodníku

<i>IO8</i>	AD9248BSTZ-20
<i>IO17b - IO17f, IO18a - IO18f</i>	74HC04
<i>L7</i>	BLM41PG600SH1
<i>C102 - C104, C106 - C108, C112 - C115, C118</i> [nF]	100
<i>C101, C105, C109, C116</i> [μF]	10
<i>C110</i> [μF]	1
<i>C111</i> [pF]	22
<i>R47</i> [kΩ]	4,7
<i>R48</i> [kΩ]	10

Tab. 2. 5: Hodnoty součástek realizovaného obvodu A/D převodníku

## 2.7 Návrh napájecího zdroje

Napájecí napětí přijímače je 12 V DC, které je na napájecí zdroj zařízení přivedeno standardním napájecím konektorem typu 2,5 mm jack situovaným na panelu zadní strany přijímače. Napájecí zdroj zařízení vychází z požadavku hned několika napěťových hladin. Jelikož jsou navíc v přijímači použity integrované obvody, které vyžadují vysokou stabilitu napájecího napětí, není vhodné použít standardní stabilizátory. Proto jsou pro napájení procesoru a FPGA využity měniče napětí PTH08080 uvedené jako *IO2, IO3, IO4* ve schématu zapojení napájecího zdroje na Obr. 2. 18. Procesor vyžaduje pro své napájecí vstupy napětí 3,3 V a FPGA mimo tuto napěťovou hladinu vyžaduje navíc i 2,5 V a 1,2 V. Vstupy a výstupy měničů jsou ošetřeny LC filtry a jejich požadovaná výstupní napětí jsou nastavena pomocí rezistorů na pinu VoAdj. Mimo uvedené napěťové hladiny je ze vstupního napájecího napětí také stabilizováno napětí 5 V pro napájení LCD displeje. Samotných 12 V, je využito pro napájení vzorkovače, kde je pro symetrické napájení operačního IF zesilovače upraveno na +/- 5 V. Všechny napájecí napětí zařízení jsou vyvedeny na svorkovnici CLL 5,08/2 a obsahují kontrolní LED. Hodnoty použitých součástek pro realizaci napájecího zdroje jsou uvedeny v Tab. 2. 6.



Obr. 2. 18: Schéma zapojení napájecího zdroje

<i>IO1</i>	NCV7805
<i>IO2, IO3, IO4</i>	PTH08080
<i>K1 - K6</i>	CLL 5,08/2
<i>LED2 - LED5, LED7</i>	HSMG-C670
<i>T1</i>	BC847C
<i>L2, L8 - L13</i>	BLM41PG600SH1
<i>C25, C26, C37 [nF]</i>	100
<i>C27, C28, C45, C49, C52, C53, C57, C60, C67, C72, C75 [μF]</i>	10
<i>C46, C48, C54, C69, C71 [μF]</i>	100
<i>C47, C55, C70 [μF]</i>	470
<i>C50, C58, C73 [nF]</i>	10
<i>C51, C59, C74 [pF]</i>	10
<i>R14 [Ω]</i>	270
<i>R15, R17 [Ω]</i>	180
<i>R16 [Ω]</i>	330
<i>R18 [kΩ]</i>	1,8
<i>R19 [Ω]</i>	68
<i>R20 [kΩ]</i>	3,3
<i>R21, R49 [Ω]</i>	470
<i>R22 [kΩ]</i>	27
<i>R23 [Ω]</i>	390
<i>R24 [kΩ]</i>	1,2

Tab. 2. 6: Hodnoty součástek realizovaného zdroje napětí

## 2.8 Řešení softwarové části

### 2.8.1 Popis programu pro hradlové pole

Program pro FPGA je napsán v jazyce VHDL pomocí již zmíněného volně dostupného vývojového prostředí XILINX ISE 9.2i. Program je koncipován jako stavový automat, což je vhodné z důvodu požadované maximální rychlosti, ale také kvůli vlastnostem VHDL procesů, které jsou v programu použity. Výpis programu pro FPGA je uveden v příloze 12.

V první fázi je vytvořen projekt, do kterého jsou nejprve vloženy potřebné knihovny. Dále je založena entita „REFUWB“, která obsahuje definice vstupních a výstupních portů a pinů. První definovaný vstup *CLK* je hodinový kmitočet 40 MHz, který je získáván z výše popsané děličky kmitočtu. Dalším vstupním pinem *SCLK* je řízena komunikace z procesoru do FPGA a vstup *POVEL* určuje směr komunikace mezi procesorem a FPGA. Vstupní port *DATA\_z\_AD* slouží pro čtení dat z A/D převodníku. Výstup *VYSTAV* je použit k signalizaci, že byly nalezeny hodnoty hledaných parametrů a FPGA je připraveno je vysílat do procesoru. Výstupním pinem *DDS\_MASTER\_RESET* se navazuje spojení se syntezátorem DDS a výstupem *DDS\_IO\_RESET* se provádí resetování jeho I/O pinů. Nastavení sériové komunikace a samotný přenos dat z FPGA do DDS je řešen výstupy *DDS\_SCLK*, *DDS\_UDCLK* a *DDS\_SDIO*. Pro přenos hledaných parametrů z FPGA do procesoru slouží 8-bitový výstup *DATA\_z\_FPGA*.

Následně je vytvořena architektura „ZAVES“, která popisuje vytvořenou entitu „REFUWB“. Zde jsou obsaženy tzv. signály, které představují vnitřní propojení. Signály *max1*, *min1*, *pozmin1*, *pozmax1*, *max2*, *min2*, *pozmin2*, *pozmax2* slouží pro ukládání hledaných charakteristik měřeného signálu. Představují tedy změřenou hodnotu amplitudy maxima a minima navzorkovaného UWB signálu a jejich hodnotu časového posunu, ve kterém byly nalezeny. Dále je zde vytvořen datový typ *banky*, jehož signál *banka* určuje, zdali bude vyhledávána první čtveřice proměnných *max1*, *min1*, *pozmin1*, *pozmax1* nebo druhá. Další datový typ *stavy* obsahuje signály *stavDDS*, *stav*, *stav0*, *stav1* a *stav2*. Signály *vensignal\_F* a *vensignal\_P* obsahují hodnoty pro nastavení frekvence a časového posunu, které má obvod DDS vykonat. Pro jejich přenos do syntezátoru slouží signály *programovací\_vektor*, *fazovaniDDS* a *fazovaniDDS2*. Signál *POVEL\_POM* slouží pro výběr čtveřice proměnných, které

budou posílány do procesoru. V neposlední řadě jsou zde definovány signály *citac* a *count*.

Následná výkonná část je tvořena dvěma procesy, které reagují na změnu odlišných vstupních signálů. Oba procesy jsou proto vykonávány současně.

- **Proces 1**

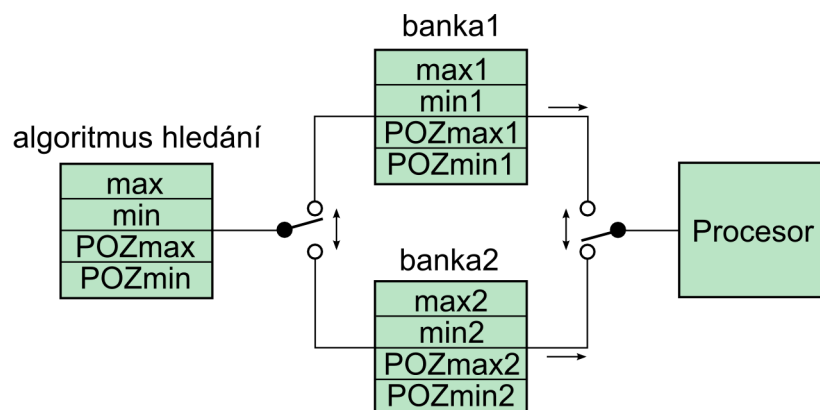
Tento proces reaguje na náběžnou hranu signálu *CLK*. Jeho funkce je rozdělena do tří částí. První část se provede, pokud signál *stav* nabývá hodnoty *s0*, na kterou je při prvním spuštění inicializován. V této části je naprogramována výstupní frekvence DDS obvodu na hodnotu 5 MHz, která je inicializována signálem *vensignal\_F*. Postup naprogramování syntezátoru je řešen obdobným způsobem jako v realizovaném generátoru s DDS v [13], kde jsou také uvedeny konkrétní parametry a požadavky pro jeho nastavení. Tato část procesu se nastaví pouze jednou, a proto k ní již není nadále přistupováno.

K vykonání druhé části procesu dochází po nastavení výstupního kmitočtu. Tato část se tedy provede, pokud signál *stav* nabývá hodnoty *s1*. V tomto případě je nastaven časový posun na uloženou hodnotu v signálu *vensignal\_P* (při prvním spuštění 0). Naprogramování syntezátoru je provedeno obdobným způsobem jako naprogramování výstupní frekvence, avšak s tím rozdílem, že zde již nefiguruje navázání komunikace pinem *DDS\_MASTER\_RESET*. K této části procesu je přistupováno neustále, což je uvedeno v následujícím odstavci.

Poslední třetí částí procesu je algoritmus hledání požadovaných proměnných a jejich ukládání v závislosti na signálu *banka* podle jeho nastavené hodnoty. Hledání se tedy provádí pro dvě čtveřice proměnných podle toho, jaké aktuální hodnoty nabývá signál *banka*. V prvním kroku po spuštění je inicializován signál *banka* na hodnotu *banka1*. V tomto případě se budou zpracovávat hodnoty proměnných *max1*, *min1*, *pozmax1* a *pozmin1*. Ty jsou inicializovány mimo proměnnou *min1* na nulu. Proměnné *min1* je přiřazena maximální hodnota daná jeho 14-bitovou velikostí, tedy 16 383. Takto nastavené proměnné jsou následně porovnány s hodnotou z A/D převodníku, která je se získávána z portu *DATA\_z\_AD*. Pokud tedy při daném časovém posunu nabývá proměnná *max1* menší hodnoty než přijímaná data *DATA\_z\_AD*, hodnota z převodníku je uložena do proměnné *max1* a odpovídající



časový posun je uložen do proměnné *pozmax1*. Pokud tato podmínka splněna není, hodnota v proměnné *max1* zůstává nezměněná a taktéž hodnota v proměnné *pozmax1*. Totožná situace nastává pro proměnnou *min1*, ale hledání probíhá opačným směrem, tedy k získání minima měřeného signálu. Po provedení těchto podmínek je inkrementována hodnota časového posunu *vensignal\_P*. Pokud *vensignal\_P* nenabývá maxima, tedy 16 383, proběhne skok na druhou část procesu, provede se časový posun a výše uvedené se opakuje. Pokud však *vensignal\_P* nabývá maximální hodnoty, znamená to, že byla prohledána celá perioda přijímaného signálu a v daných proměnných jsou uloženy hodnoty maxim, minim a jejich časových pozic. V této fázi je pin *VYSTAV* uveden do log. 1 a to za podmínky, že všechna předešlá data (v tomto případě čtveřice proměnných *max2*, *min2*, *pozmax2* a *pozmin2*) byla odeslána do procesoru. Tento stav znázorňuje Obr. 2. 19 přepínači mezi zobrazenými bloky.



Obr. 2. 19: Schéma třetí části procesu 1

Následně jsou proměnné *max2*, *min2*, *pozmax2* a *pozmin2* a hodnota časového posunu nastaveny do výchozích hodnot a pro další hledání je do signálu *banka* uložena hodnota *banka2*. Mezitím se do procesoru posílají hodnoty proměnných *max1*, *min1*, *pozmax1* a *pozmin1*.

## • Proces 2

Tento proces reaguje na náběžnou hranu signálu *SCLK*, který je generován procesorem v závislosti na vyvolání funkce pro příjem bajtu každé proměnné z FPGA. Pro příjem každé čtveřice proměnných z FPGA, je z procesoru vystaven pin *POVEL* na log. 0. Po přijetí celé dané čtveřice je pin *POVEL* procesorem vystaven do log. 1. Tento stav nuluje signál *citac*, jenž postupnou inkrementací rozhoduje o

tom, který bajt dané proměnné se bude do procesoru v daném kroku posílat. Výběr posílané čtveřice proměnných je řešen signálem *POVEL\_POM*. Ten je vždy po odeslání celé čtveřice proměnných nastaven do takové logické úrovně, jenž splňuje podmínku pro následné vybrání čtveřice, která nebyla právě posílána. S ohledem na to, že jsou měřená data 14-bitová a port *DATA\_z\_FPGA* 8-bitový, posílají se do procesoru po bajtech, přičemž horní bajt každé proměnné je doplněn nulami. Po odeslání celé čtveřice je signál *stavDDS* nastaven na hodnotu *s0*, čímž je možné v procesu 1 pin *VYSTAV* uvést do log. 1 a díky tomu posílat do procesu následující čtveřici proměnných.

### 2.8.2 Popis programu pro procesor

Program pro procesor je napsán v jazyce C ve volném vývojovém prostředí Programmers Notepad. Pro překlad programu byl použit kompilátor WinARM. Výpis části programu je uveden v příloze 13.

Tato část programu obsahuje pouze výběr funkcí, které charakterizují samotnou funkci zařízení a jsou definovány v externím hlavičkovém souboru. Zde jsou nejdříve uvedené potřebné knihovny a dále jsou definovány názvy použitých pinů procesoru. Samotné piny a jejich nastavení je definováno v hlavičkovém souboru *board.h*, který definuje fyzické prvky na DPS, vnitřní prvky procesoru a další. Piny procesoru označené jako *RS*, *RW*, *E*, *DB0* až *DB7* slouží pro nastavení LCD displeje. Piny *SCLK*, *POVEL* a *VYSTAV* představují již výše zmíněné funkce ohledně komunikace s hradlovým polem FPGA. Následně jsou deklarovány potřebné řetězce a proměnné, které jsou v daných funkcích používány.

V hlavní funkci programu *main()* je nejdříve vyvolána funkce *LCD\_init()*, která inicializuje LCD displej. Následně je volána funkce *LCD\_head()*, která vypisuje hlavičkový text na danou adresu LCD displeje pro každý zobrazovaný parametr. Poté je volána funkce *Prevod()*, která tvoří hlavní výpočetní funkci.

Funkce *Prevod()* pracuje v nekonečné smyčce. Nejdříve je zjišťován stav pinu *VYSTAV*. Pokud je pin *VYSTAV* v log. 1, což znamená, že je FPGA připraveno posílat data, vystaví se pin *POVEL* do log. 0 a 8 krát se vyvolá funkce *Prijimej\_BYTE*. Její vrácené hodnoty se ukládají do proměnných *maxH*, *maxL*, *minH*, *minL*, *pozmaxH*, *pozmaxL*, *pozminH*, *pozminL* přesně tak, jak byly poslány z FPGA.

V uvedených proměnných jsou tedy uloženy hodnoty horních a dolních bajtů přijaté proměnné z FPGA. Poté je pin *POVEL* vystaven do log. 1. Přijaté 2 bajty dané proměnné jsou následně sloučeny do 16-bitové proměnné, která je poté funkcí *sprintf()* převedena na řetězec. Poté je volána funkce *LCD\_write\_znak()*, která nejdříve nastavuje pozici na LCD displeji, kam má být daná hodnota vypsána. Dále je tato hodnota na již definovanou pozici LCD vypsána pomocí funkce *LCD\_write\_string()*. Za danou hodnotu jsou dále vypsány prázdné znaky, aby případně prepisovaly předešlou číselnou hodnotu, která byla ciferně delší než aktuálně zobrazená. To se projevuje hlavně u zjištěného minima, jelikož je dvou i tříciferné. Tento postup se opakuje pro všechny přijímané proměnné.

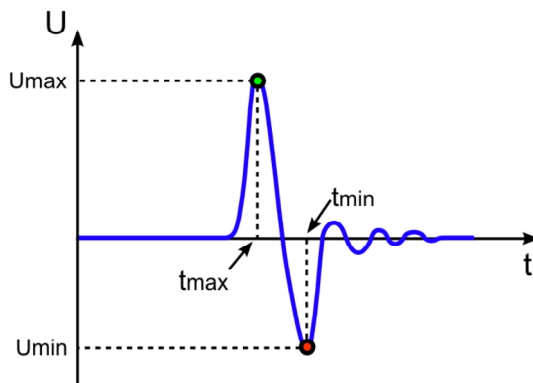
Funkce pro příjem posílaného bajtu z FPGA je uvedena pod názvem *Prijimej\_BYTE()*. Touto funkcí je generován hodinový signál *SCLK*, kterým je v programu pro FPGA řízen proces 2. Po vystavení pinu *SCLK* do log. 1 je načítán celý bajt z FPGA a nakonec vrácena jeho hodnota.

Pro výpis na displej a jeho konfiguraci slouží dvouparametrová funkce *LCD\_write\_znak()*. Pro výpis na displej je parametr *print* nastaven na hodnotu log. 1, pro zadání adresy nebo konfiguraci LCD displeje je pak hodnota tohoto parametru nastavena na log. 0. Pro výpis celého řetězce na displej slouží funkce *LCD\_write\_string()*.

## **2.9 Funkce a měření zrealizovaného přijímače**

### **2.9.1 Funkce přijímače**

Z popisu programů vyplývá, že přijímač neustále sleduje přijímaný UWB signál, přičemž vyhledává jeho maximální amplitudu, minimální amplitudu a jejich pozice v čase. Z těchto hodnot lze odvodit správnou funkci detekce přijímaného UWB signálu. Spolehlivá detekce je totiž stěžejní záležitostí samotné UWB komunikace. Princip ověření detekce UWB pulsů je uveden na Obr. 2. 20.

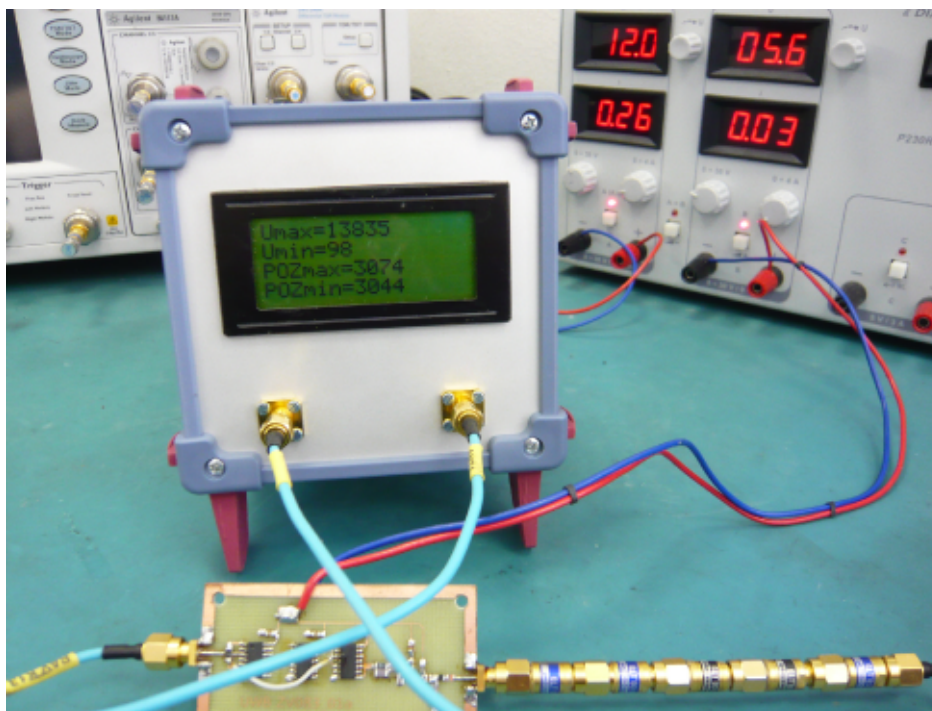


**Obr. 2. 20: Princip ověření správné detekce UWB signálu**

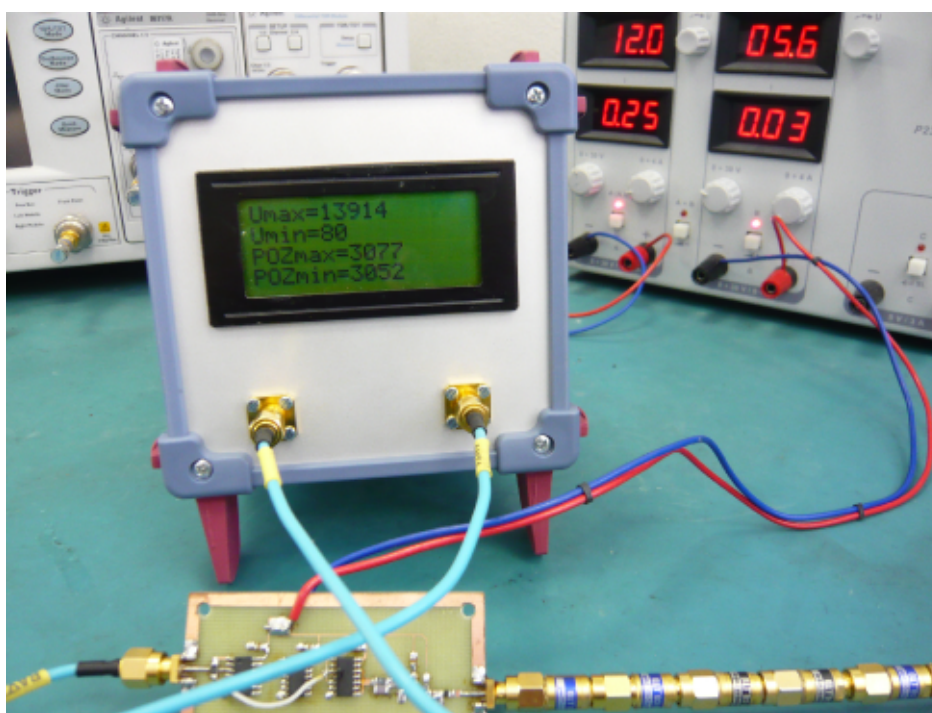
Z obrázku vyplývá, že hledaná pozice maximální amplitudy UWB pulsu je označena jako  $U_{max}$  a minimální hodnota  $U_{min}$ . Pozice těchto amplitud v čase jsou označeny jako  $t_{max}$  a  $t_{min}$ . Pokud by byl UWB signál spolehlivě detekován, dané amplitudy a jejich pozice v čase by se měly v průběhu opakovacího cyklu jejich vyhledávání jen nepatrně lišit. To lze ověřit při následném měření přijímače.

### 2.9.2 Měření přijímače

Na fotografiích zobrazených na Obr. 2. 21 a Obr. 2. 22 je uvedeno měření přijímače ve dvou okamžicích. Z tohoto pohledu je k pravému SMA konektoru Tx připojen generátor, který je buzen konstantním TTL signálem z děličky o kmitočtu 5 MHz. Výstup generátoru je utlumen řadou atenuátorů o celkové hodnotě 27 dB. UWB signál vstupuje do přijímače z tohoto pohledu přes levý SMA konektor Rx. Po uvedení generátoru a přijímače do provozu připojením k napájecímu napětí se na LCD displeji nejprve vypíše hlavičkový text měřených parametrů a následně jsou opakovaně za sebou vypisovány naměřené hodnoty. Na displeji jsou tedy zobrazovány maximální hodnoty amplitudy - parametr  $U_{max}$ , minimální amplitudy  $U_{min}$  a jejich pozice posunu v čase  $POZ_{max}$  a  $POZ_{min}$ . Konkrétní hodnoty zobrazované na displeji ukazují pouze číslicovou hodnotu naměřeného vzorku v desítkové soustavě. Přesné hodnoty změřených amplitud napětí a jejich hodnoty časových posunů, kterým odpovídají než byly nalezeny, nejsou pro kontrolu správné funkce důležité.



Obr. 2. 21: Měření přijímače - okamžik 1



Obr. 2. 22: Měření přijímače - okamžik 2

Hodnoty změřených parametrů v daných okamžicích jsou uvedeny v Tab. 2. 7.

	Okamžik 1	Okamžik 2
$U_{\max}$	13835	13914
$U_{\min}$	98	80
$POZ_{\max}$	3074	3077
$POZ_{\min}$	3044	3052

Tab. 2. 7: Hodnoty sledovaných parametrů ve dvou okamžicích

Z uvedené tabulky je možné pozorovat, že změřené parametry UWB signálu se jen nepatrně liší, což splňuje výše uvedenou podmínku. Správnou funkci zařízení tedy potvrzuje fakt, že pokud se zvýšila hodnota  $POZ_{\max}$ , následoval ji také parametr  $POZ_{\min}$ . To znamená, že je detekovaný puls v čase více zpožděn než puls, jehož parametry byly zobrazeny v předchozím kroku. Dále je také možné sledovat, že při rostoucí hodnotě parametru  $U_{\max}$ , klesá parametr  $U_{\min}$ . Změna amplitudy generovaného pulsu je způsobena teplotním driftem v daných časových okamžicích. Pokud měřený puls nabývá vyšší amplitudy maxima, je pravděpodobné, že nabývá nižší amplitudy minima z důvodu zvyšujících se zákrmitů.

## Závěr

Cílem této práce byl návrh a realizace přijímače referenčního UWB signálu. Přijímač měl řešit synchronizaci časování příjmu vůči UWB vysílači pomocí časového závěsu a pro kontrolu jeho nastavení měl na LCD displeji zobrazit úroveň přijímaného signálu.

Zařízení vychází z požadavku zlepšit vlastnosti detekce UWB pulsů a eliminovat snadnou zarušitelnost přijímače s přímou detekcí, který byl realizován v mé bakalářské práci. Proto byl zvolen naprosto odlišný způsob přijímání UWB signálu. Stěžejní část nynějšího přístupu je vzorkování měřeného UWB signálu, které je realizováno širokopásmovým obvodem S/H. Jeho otevření po minimální časový úsek je řešeno pomocí stroboskopických pulsů, které dosahují amplitudy vyšší než 3,5 V a jejich šířka pulsu je přitom pouhých cca 400 ps. Vzhledem k jednoduchosti zapojení realizovaného generátoru stroboskopických pulsů jsou docílené parametry pulsu více než dostačující. Jelikož byly na vzorkovač kladeny přísné požadavky, jeho realizace a ladění bylo nejnáročnější částí celého zařízení a trvalo několik týdnů, které byly provázeny mnoha problémy. Pro realizaci posouvání pulsu v čase, což vyžaduje princip časového závěsu, byl využit syntezátor DDS AD9852. S tímto obvodem byl realizován ročníkový projekt I, ovšem syntezátor zde plnil jinou funkci. Řízení časového posunu bylo nejprve realizováno pomocí procesoru AT91SAM7S64. Díky této fázi bylo možné v rámci možností změřit vlastnosti vzorkovače. Z důvodu požadované maximální rychlosti změny časového posunu bylo pro řízení syntezátoru DDS použito hradlové pole FPGA XC3S50. Tento krok s sebou přinesl odlišný přístup programování. Místo programovacího jazyku C, který je používán u procesorů, obvody FPGA využívají programovací jazyk VHDL. Ten se vyznačuje odlišnými postupy a jinou stavbou programu. Jelikož FPGA umožňuje vykonávat požadované procesy paralelně, je možné v jednom procesu hledat potřebná data přijímaného signálu a zároveň v druhém procesu již nalezená data posílat do procesoru pro jejich zobrazení na LCD displeji.

Přínosem této práce je způsob detekování UWB signálu. Díky vlastnostem přijímání s referencí a principu časového závěsu se tento způsob jeví jako velice efektivní. Proto bude přijímač následně doplněn o druhý kanál A/D převodníku pro vyhodnocení datového UWB signálu, který bude přijímán v přesně stanovený

okamžik od referenčního signálu. Pro tento způsob přenosu informace bude použita modulace OOK, kterou lze dosáhnout na daném kmitočtu přenosové rychlosti i 5 Mbit/s. V případě požadavku vyšší přenosové rychlosti, již není velký problém zrealizovanou strukturu navrhnout např. pro 10 MHz.



## Seznam použité literatury

- [1] Vlasák, V.: UWB přijímač s přímou detekcí pulsů, Bakalářská práce, Univerzita Pardubice, Dopravní fakulta Jana Pernera, 2009. 71 s.
- [2] Siwiak, K., Mckeowen, D.: *Ultra-wideband radio technology*, John Wiley & sons 2004, ISBN 0-470-85931-1.
- [3] Mrkvica, J., Dvořák, D., Jeřábek, J., Šikl, R.: *UWB radary a senzory*, Seminář ČES: Moderní technologie v mikrovlnných aplikacích, 2007.
- [4] Bezoušek, P., Šedivý, P.: *Radarová technika*, Nakladatelství ČVUT, 2004, ISBN 80-86056-47-3.
- [5] Rumánek, J.: *Nové metody kanálového kódování pro družicovou komunikaci*, Disertační práce, VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 97 s.
- [6] Pozar, D., M.: *Microwave Engineering Third Edition*, John Wiley & sons 2004, ISBN 978-0-471-44878-5.
- [7] ANALOG DEVICES inc., [online]. c2011, aktualizováno 20.4.2011. Dostupné z: <[http://www.analog.com/static/imported-iles/tech\\_docs/MixedSignal\\_Sect6.pdf](http://www.analog.com/static/imported-iles/tech_docs/MixedSignal_Sect6.pdf)>.
- [8] Kubíček, M.: *Modelování a implementace subsystémů komunikačního řetězce v obvodech*, Disertační práce, VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 101 s.
- [9] EETIMES, [online]. c2011, aktualizováno 20.4.2011. Dostupné z: <<http://www.eetimes.com/design/programmable-logic/4014815/All-about-FPGAs>>.
- [10] Šafařík, M.: *Počítačem řízený vzorkovací osciloskop s FPGA*, Diplomová práce, Univerzita Pardubice, Dopravní fakulta Jana Pernera, 2009. 86 s.
- [11] Vlasák, V.: *UWB generátory stroboskopických pulsů*, Ročníkový projekt II, Univerzita Pardubice, Dopravní fakulta Jana Pernera, 2011. 21 s.

- [12] Abuasaker, S., Kompa, G.: A High Sensitive Receiver for Baseband pulse Microwave Radar Sensor Using Hybrid Technology, Department of High Frequency Engineering, University of Kassel.
- [13] Vlasák, V.: Generátor 20 MHz s využitím DDS, Ročníkový projekt I, Univerzita Pardubice, Dopravní fakulta Jana Pernera, 2010. 43 s.

## Seznam tabulek

Tab. 2. 1: Hodnoty součástek realizovaného obvodu procesoru .....	42
Tab. 2. 2: Hodnoty součástek realizovaného obvodu FPGA .....	45
Tab. 2. 3: Hodnoty součástek realizovaného zdroje hodinového signálu .....	46
Tab. 2. 4: Hodnoty součástek realizovaného obvodu DDS.....	49
Tab. 2. 5: Hodnoty součástek realizovaného obvodu A/D převodníku .....	61
Tab. 2. 6: Hodnoty součástek realizovaného zdroje napětí .....	62
Tab. 2. 7: Hodnoty sledovaných parametrů ve dvou okamžicích.....	70

## Seznam obrázků

Obr. 1: Příjem UWB signálu s referencí.....	11
Obr. 1. 1: Obsazení frekvenčního spektra technologií UWB OFDM.....	12
Obr. 1. 2: Technologie DS-UWB: a) tvar pulsu; b) normované spektrum .....	13
Obr. 1. 3 Charakter UWB pulsu v časové a frekvenční oblasti .....	13
Obr. 1. 4: Charakter úzkopásmového signálu v časové a frekvenční oblasti.....	14
Obr. 1. 5: Závislost normované přenosové kapacity a poměru S/N radiového komunikačního systému na normované šířce pásma .....	16
Obr. 1. 6: Dvoupaprskový model šíření nad rovinnou zemí .....	17
Obr. 1. 7: Závislost přijatého výkonu na vzdálenosti antén.....	18
Obr. 1. 8: Blokové schéma vysílače úzkopásmového signálu .....	19
Obr. 1. 9: Analogový přenášený signál.....	20
Obr. 1. 10: Nosný signál .....	21
Obr. 1. 11: Vysílaný vysokofrekvenční signál .....	21
Obr. 1. 12: Blokové schéma úzkopásmového přijímače .....	23
Obr. 1. 13: Filtrovaný signál na mezifrekvenci .....	23
Obr. 1. 14: Vysílač UWB: a) blokové schéma; b) tvar pulsu na nosném kmitočtu ....	24
Obr. 1. 15: Demodulovaný Gaussův puls na straně přijímače.....	24
Obr. 1. 16: Princip časového závěsu .....	25
Obr. 1. 17: Blokové schéma DDS .....	26
Obr. 1. 18: Kruhový diagram s hodnotou fázového skoku M.....	27
Obr. 1. 19: Spektrum výstupního kmitočtu DDS před filtrem .....	28
Obr. 1. 20: Architektura FPGA [9].....	32
Obr. 1. 21: Princip ekvivalentního vzorkování: a) měřený VF signál; b) rekonstruovaný signál .....	33
Obr. 1. 22: Obvod SPD [12].....	35

Obr. 1. 23: Vzorkovací obvod buzený stroboskopickými pulsy [11] .....	35
Obr. 2. 1: Blokové schéma přijímače .....	37
Obr. 2. 2: Schéma zapojení procesoru .....	41
Obr. 2. 3: Schéma zapojení hradlového pole FPGA .....	44
Obr. 2. 4: Schéma zapojení zdroje hodinového signálu .....	45
Obr. 2. 5: Schéma zapojení syntezátoru DDS .....	48
Obr. 2. 6: Blokové schéma měření časového posunu řízeného procesorem .....	50
Obr. 2. 7: Měření časového posunu řízeného procesorem .....	50
Obr. 2. 8: Blokové schéma měření časového posunu řízeného pomocí FPGA .....	51
Obr. 2. 9: Měření časového posunu řízeného hradlovým polem FPGA .....	52
Obr. 2. 10: Blokové schéma vzorkovače .....	52
Obr. 2. 11: Blokové schéma měření vzorkovače .....	53
Obr. 2. 12: Měřený UWB puls .....	54
Obr. 2. 13: Výstup vzorkovače při různých posunech UWB pulsu .....	55
Obr. 2. 14: Navzorkované průběhy UWB pulsu .....	57
Obr. 2. 15: Porovnání měřeného a navzorkovaného UWB pulsu .....	58
Obr. 2. 16: Měření vzorkovače .....	58
Obr. 2. 17: Schéma zapojení A/D převodníku .....	60
Obr. 2. 18: Schéma zapojení napájecího zdroje .....	62
Obr. 2. 19: Schéma třetí části procesu 1 .....	65
Obr. 2. 20: Princip ověření správné detekce UWB signálu .....	68
Obr. 2. 21: Měření přijímače - okamžik 1 .....	69
Obr. 2. 22: Měření přijímače - okamžik 2 .....	69

## Seznam příloh

Příloha 1: Deska plošných spojů – Napájecí zdroj

Příloha 2: Deska plošných spojů – Vzorkovač

Příloha 3: Deska plošných spojů – Základní deska

Příloha 4: Deska plošných spojů – DDS a A/D převodník

Příloha 5: Deska plošných spojů – Obvod LCD displeje

Příloha 6: Deska plošných spojů – Montážní deska

Příloha 7: Zrealizovaný návrh – Napájecí zdroj a vzorkovač

Příloha 8: Zrealizovaný návrh – Základní deska

Příloha 9: Zrealizovaný návrh – DDS a A/D převodník

Příloha 10: Zrealizovaný návrh – Montážní prvky

Příloha 11: Vnější zpracování UWB přijímače

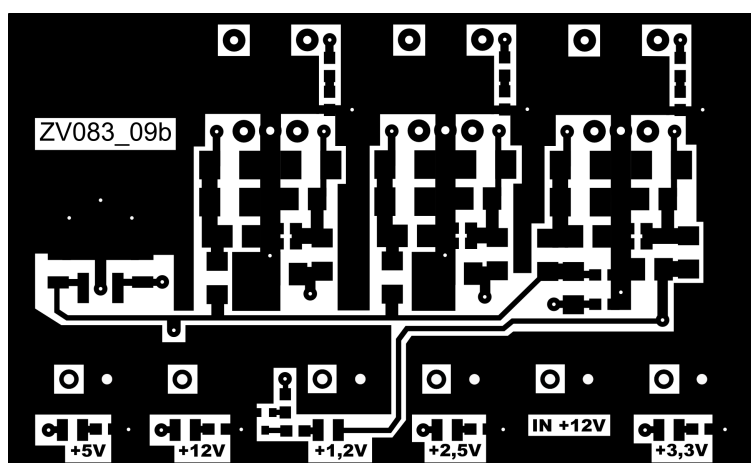
Příloha 12: Výpis programu pro FPGA

Příloha 13: Výpis části programu pro procesor

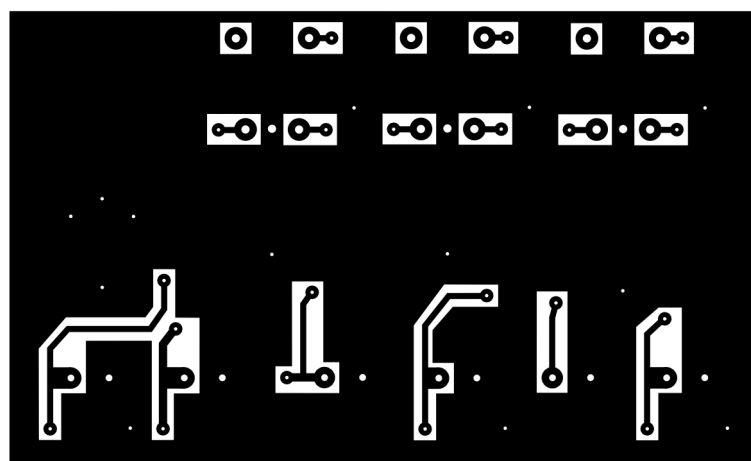
## Seznam zkratek

A/D	Analogově-digitální	(Analog to digital)
ADC	Analogově digitální převodník	(Analog to digital Converter)
ARM	Typ rychlého procesoru	(Advanced Reduced instr. set comp. Machine)
BER	Bitová chybovost	(Bit Error Rate)
CISC	Komplexní instrukční soubor	(Complex Instruction Set Computer)
CPLD	Komplexní program. log. obvod	(Complex Programmable Logic Device)
CW	Spojité signál	(Continuous Wave)
D/A	Digitálně-analogový	(Digital to Analog)
DDS	Přímá číslicová syntéza	(Direct Digital Synthesys)
FPGA	Hradlové pole	(Field Programmable Gate Array)
GND	Zem	(Ground)
IF	Mezifrekvenční signál	(Intermediate Frequency)
JTAG	Komunikační jednotka	(Join Test Action Group)
LAN	Místní síť	(Local Area Network)
LCD	Displej z tekutých krystalů	(Liquid Crystal Display)
LO	Lokální oscilátor	(Local oscilator)
LSB	Nejméně významný bit	(the Least Significant Bit)
LUT	Tabulka pro zadávání funkcí	(Look-Up Table)
MIPS	Milion instrukcí za sekundu	(Million Instruction Per Second)
MSB	Nejvíce významný bit	(the Most Significant Bit)
OOK	Způsob klíčování	(On-Off Keying)
PLB	Programovatelný logický blok	(Configurable Logic Block)
PLL	Fázový závěs	(Phase Locked Loop)
PSK	Klíčování fázovým posuvem	(Phase-Shift Keying)
PWM	Pulsně-šířková modulace	(Pulse Width Moduation)
RAM	Paměť s náhodným přístupem	(Random Access Memory)
RF	Vysokofrekvenční signál	(Radio Frequency)
RISC	Redukovaný instrukční soubor	(Reduced Instruction Set Computer)
S/H	Vzorkovací obvod	(Sample and Hole)
SPI	Sériové komunikační rozhraní	(Serial Pheripetial Interface)
TWI	Dvou vodičové rozhraní	(Two Wires Interface)
USART	Sériová komunikační jednotka	(Univer. Synch. & Asynch. Rec. & Trans.)
USB	Univerzální sériové rozhraní PC	(Universal Serial Bus)
UWB	Ultra širokopásmové	(Ultra-wide band)

**Deska plošných spojů – Napájecí zdroj**



(VRSTVA TOP 100 x 61 mm)

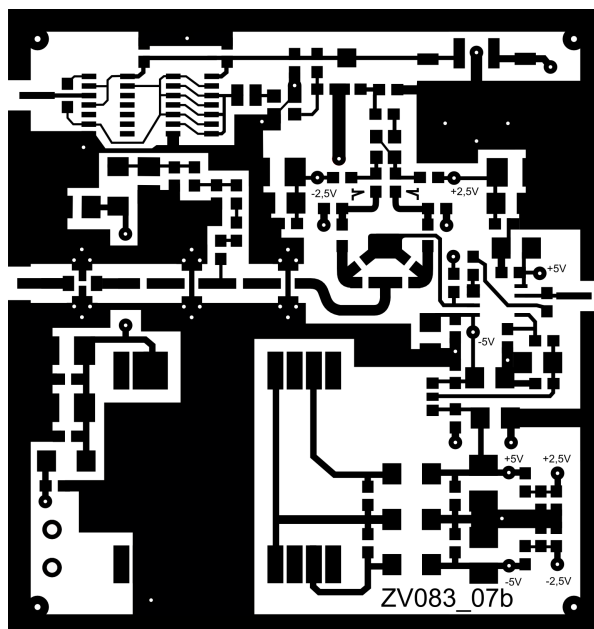


(VRSTVA BOTTOM 100 x 61 mm)

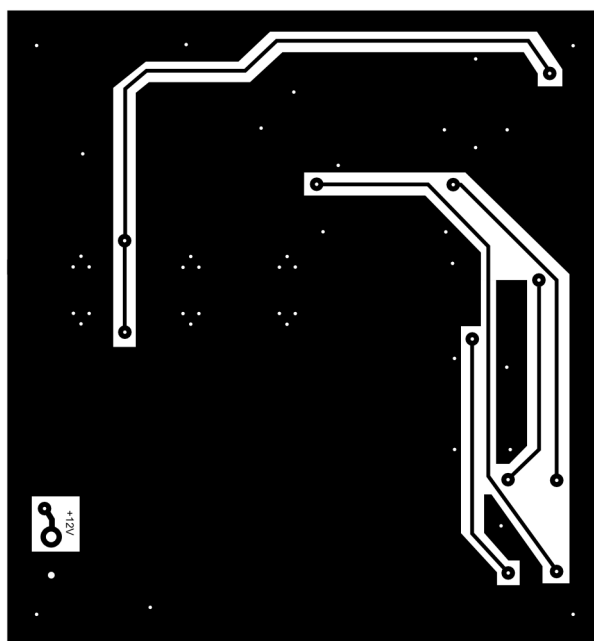
---



**Deska plošných spojů – Vzorkovač**



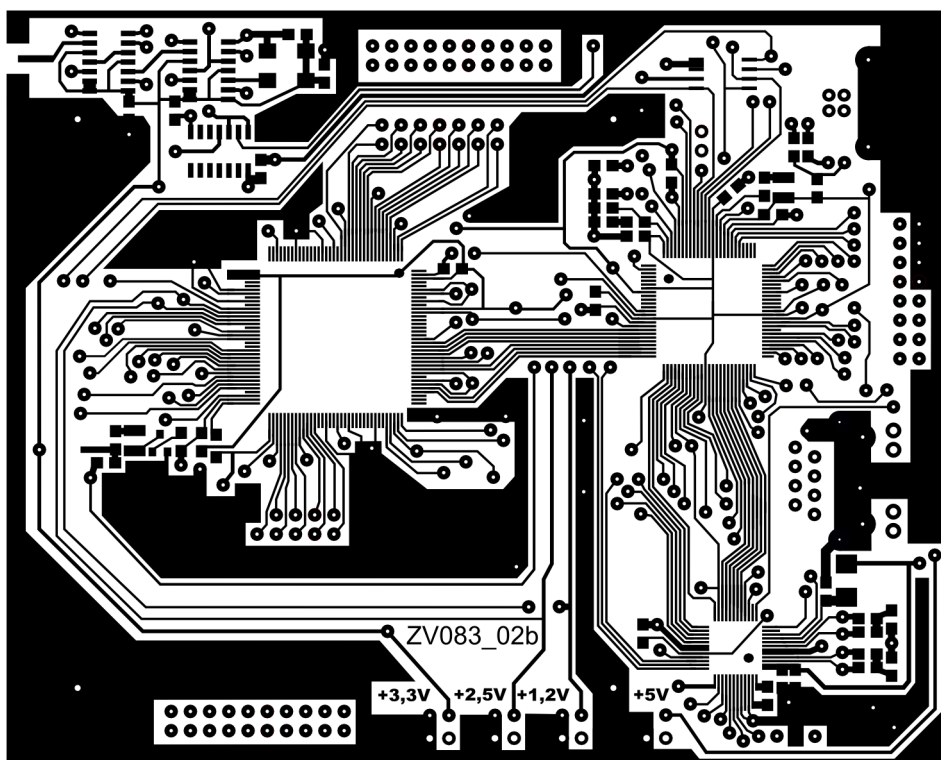
(VRSTVA TOP 79 x 83 mm)



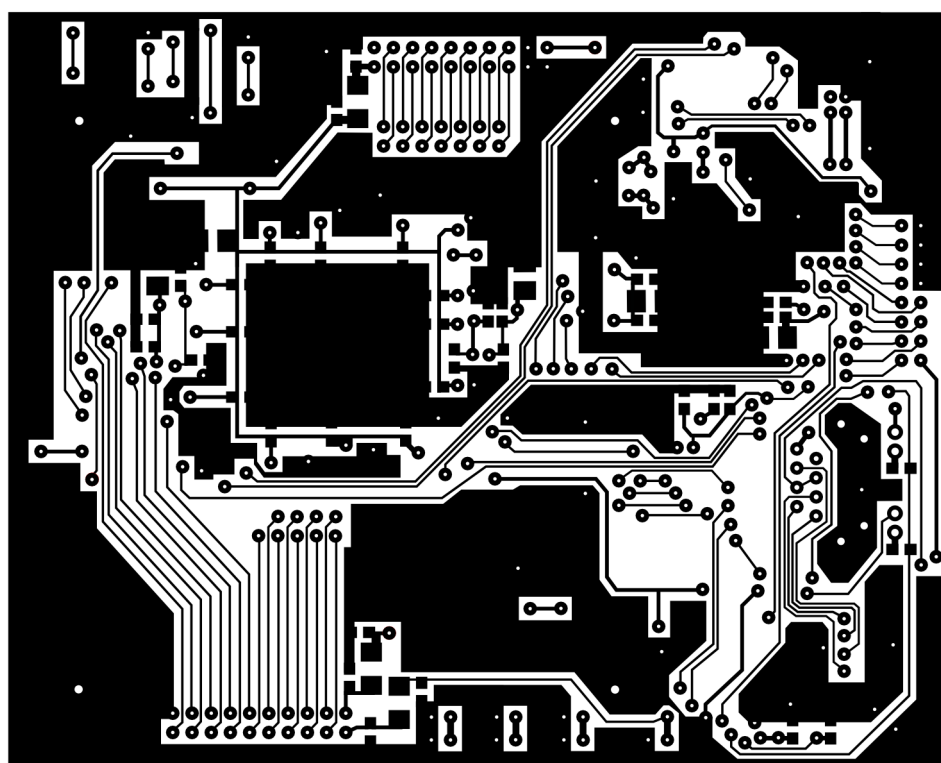
(VRSTVA BOTTOM 79 x 83 mm)

---

**Deska plošných spojů – Základní deska**



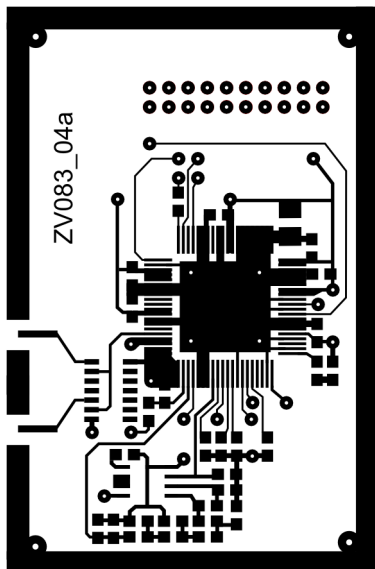
(VRSTVA TOP 120 x 100 mm)



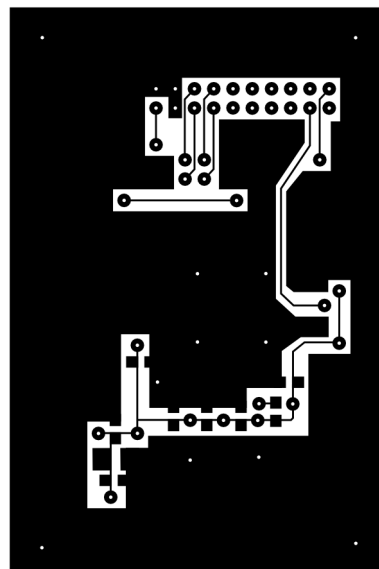
(VRSTVA BOTTOM 120 x 100 mm)

**Deska plošných spojů – DDS a A/D převodník**

- **Obvod DDS**

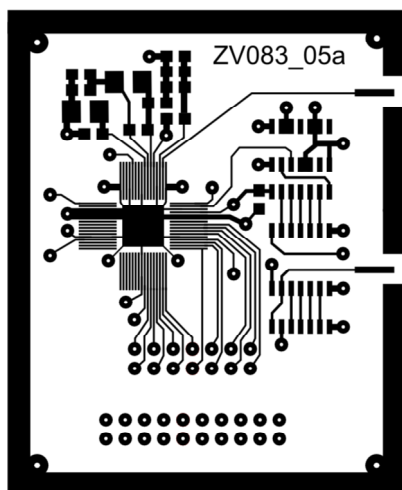


(VRSTVA TOP 49 x 75 mm)

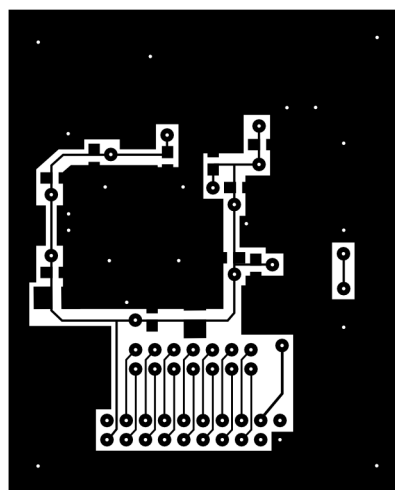


(VRSTVA BOTTOM 49 x 75 mm)

- **Obvod A/D převodníku**



(VRSTVA TOP 53 x 64 mm)



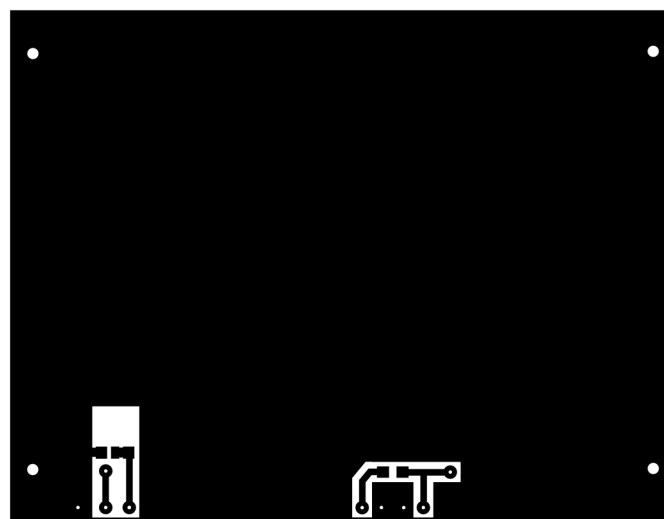
(VRSTVA BOTTOM 53 x 64 mm)

---

***Deska plošných spojů – Obvod LCD displeje***

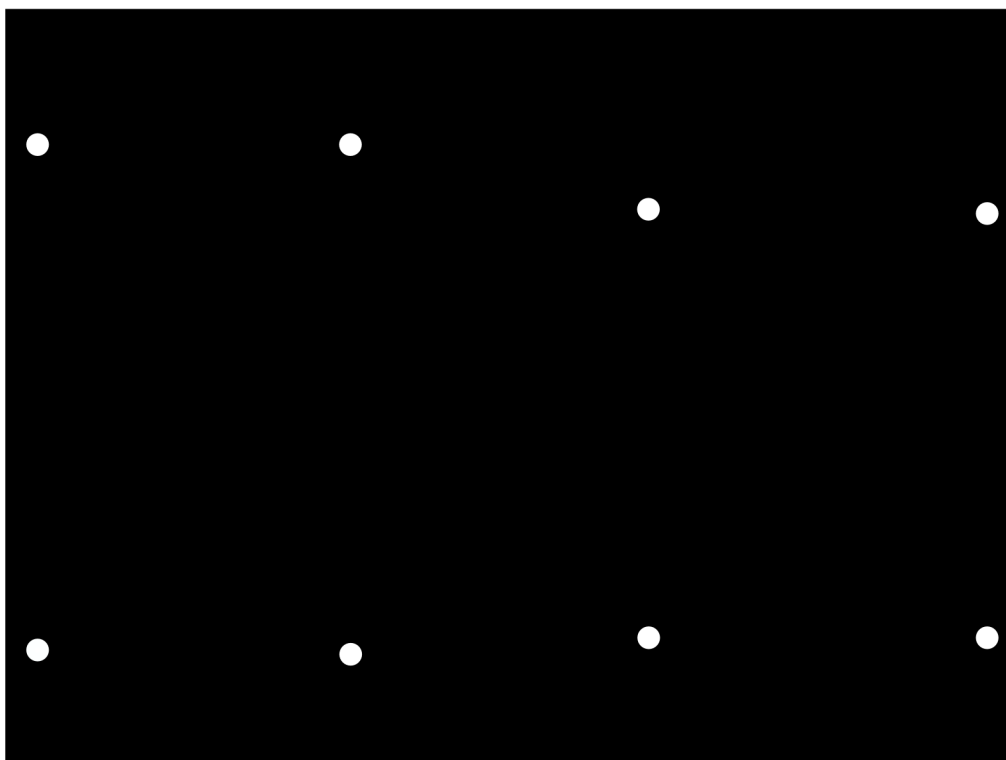


(VRSTVA TOP 88 x 69 mm)

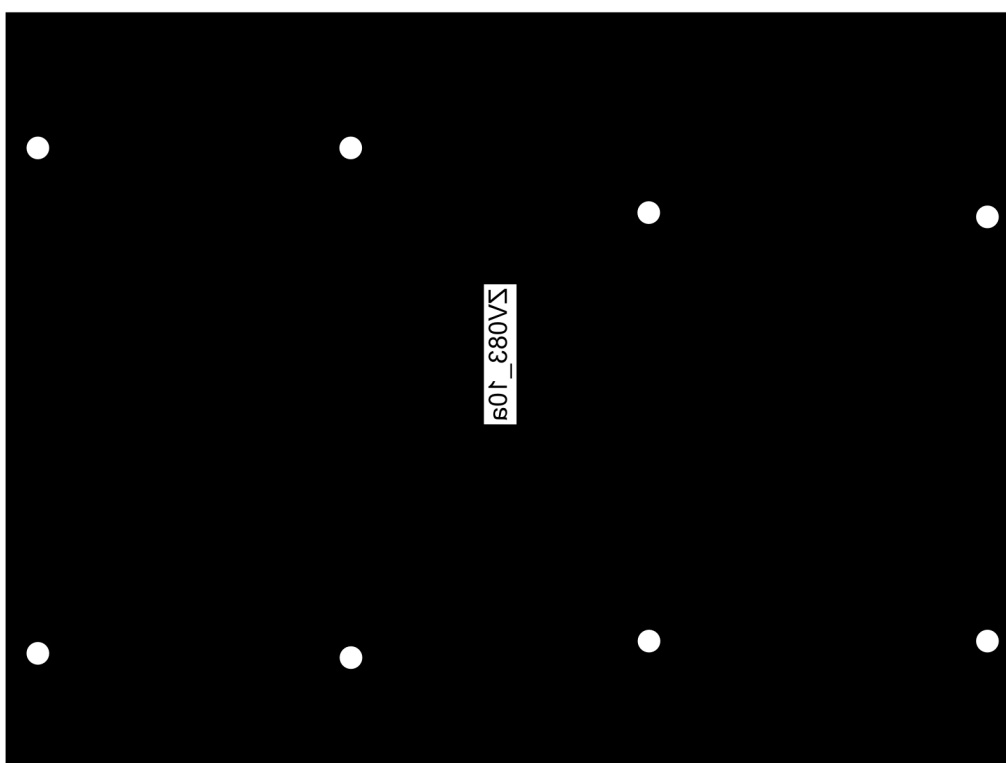


(VRSTVA BOTTOM 88 x 69 mm)

***Deska plošných spojů – Montážní deska***



(VRSTVA TOP 134 x 100 mm)

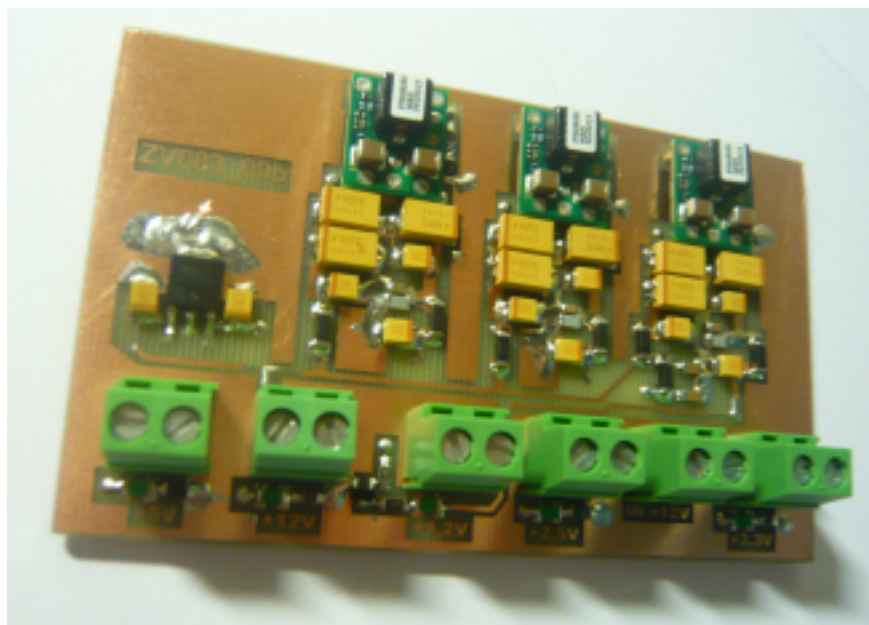


(VRSTVA BOTTOM 134 x 100 mm)

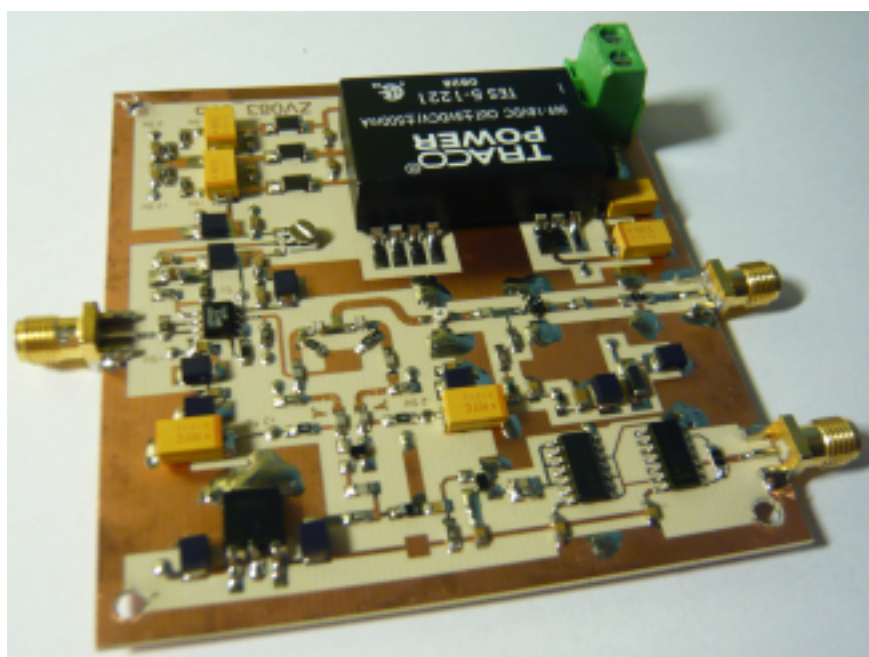
---

**Zrealizovaný návrh – Napájecí zdroj a vzorkovač**

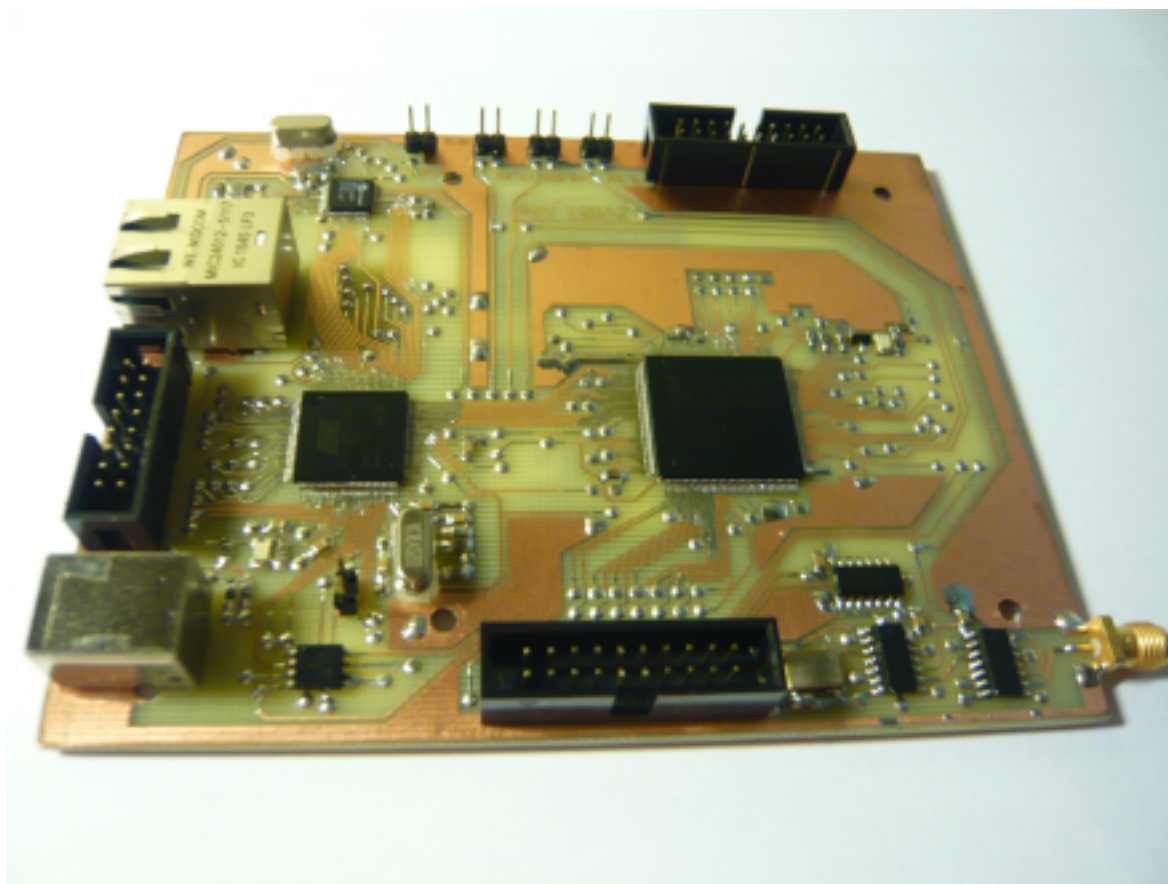
- **Napájecí zdroj**



- **Vzorkovač**

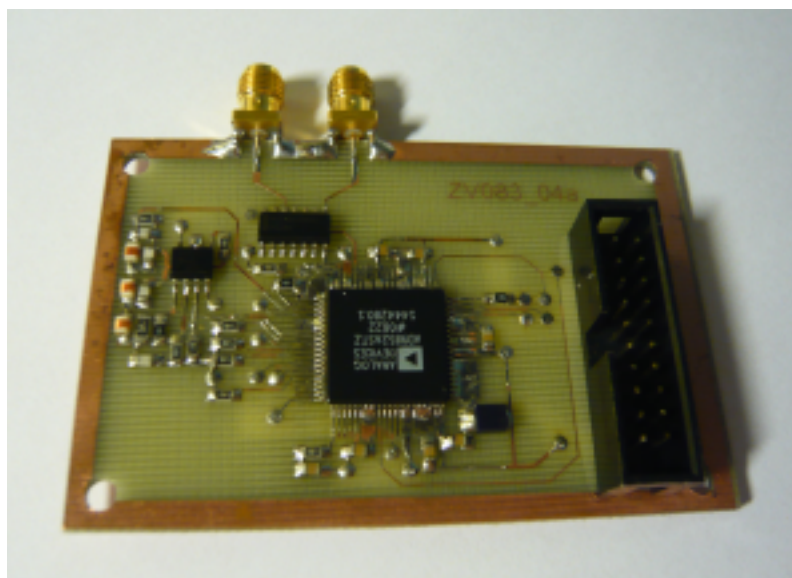


***Zrealizovaný návrh – Základní deska***

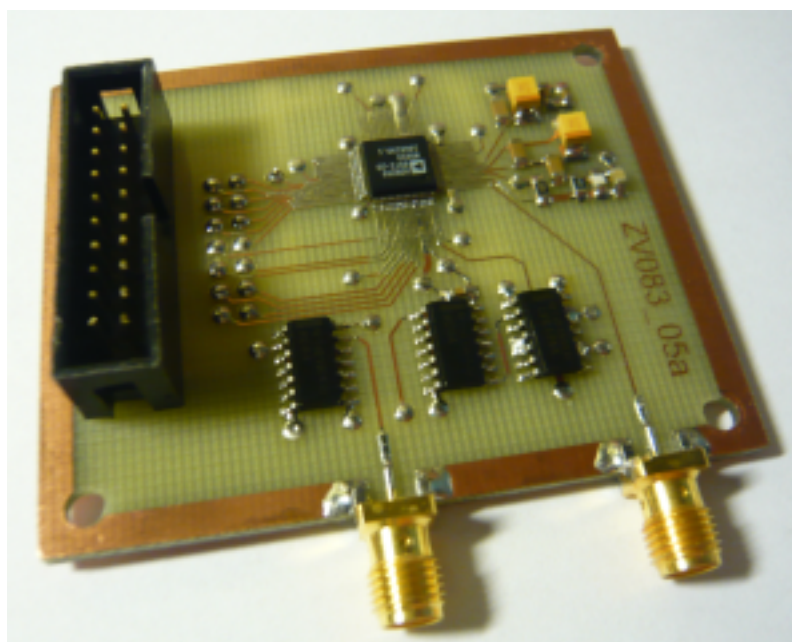


**Zrealizovaný návrh – DDS a A/D převodník**

- **Obvod DDS**



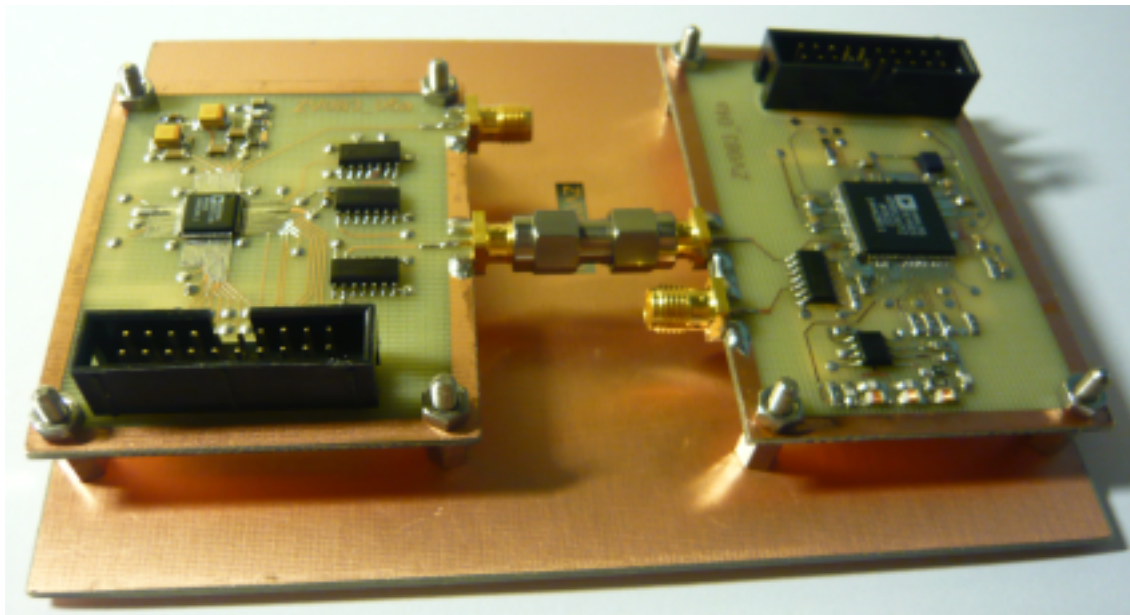
- **Obvod A/D převodníku**



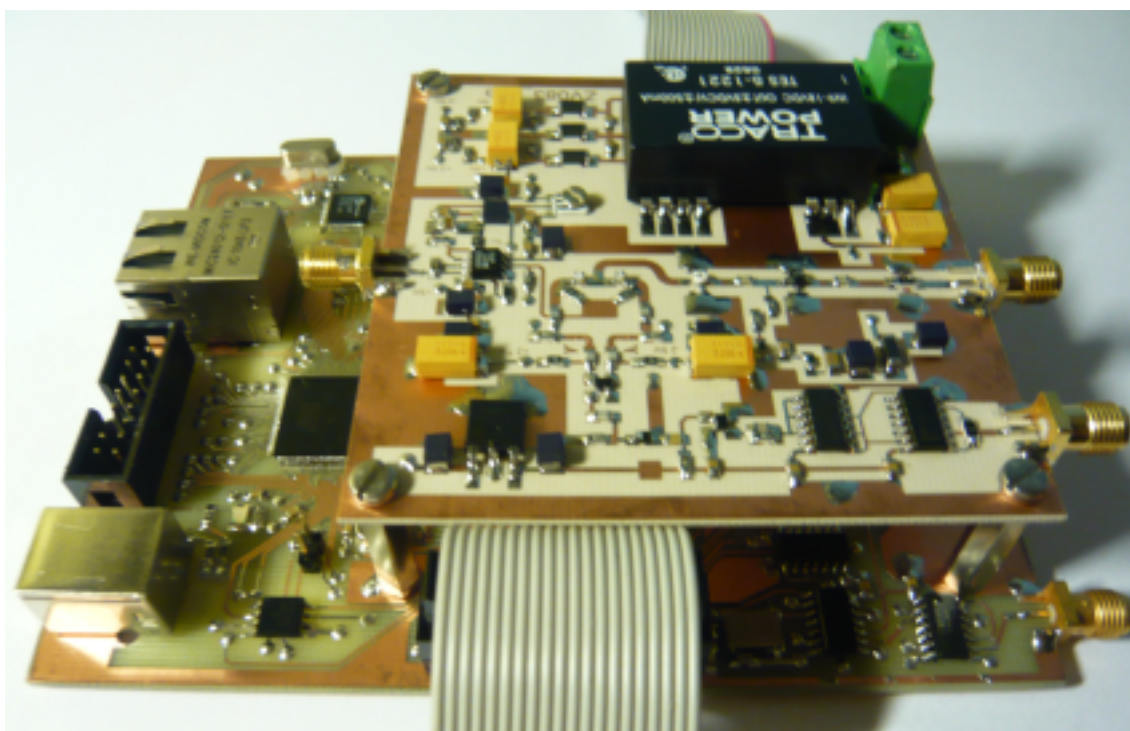


**Zrealizovaný návrh – Montážní prvky**

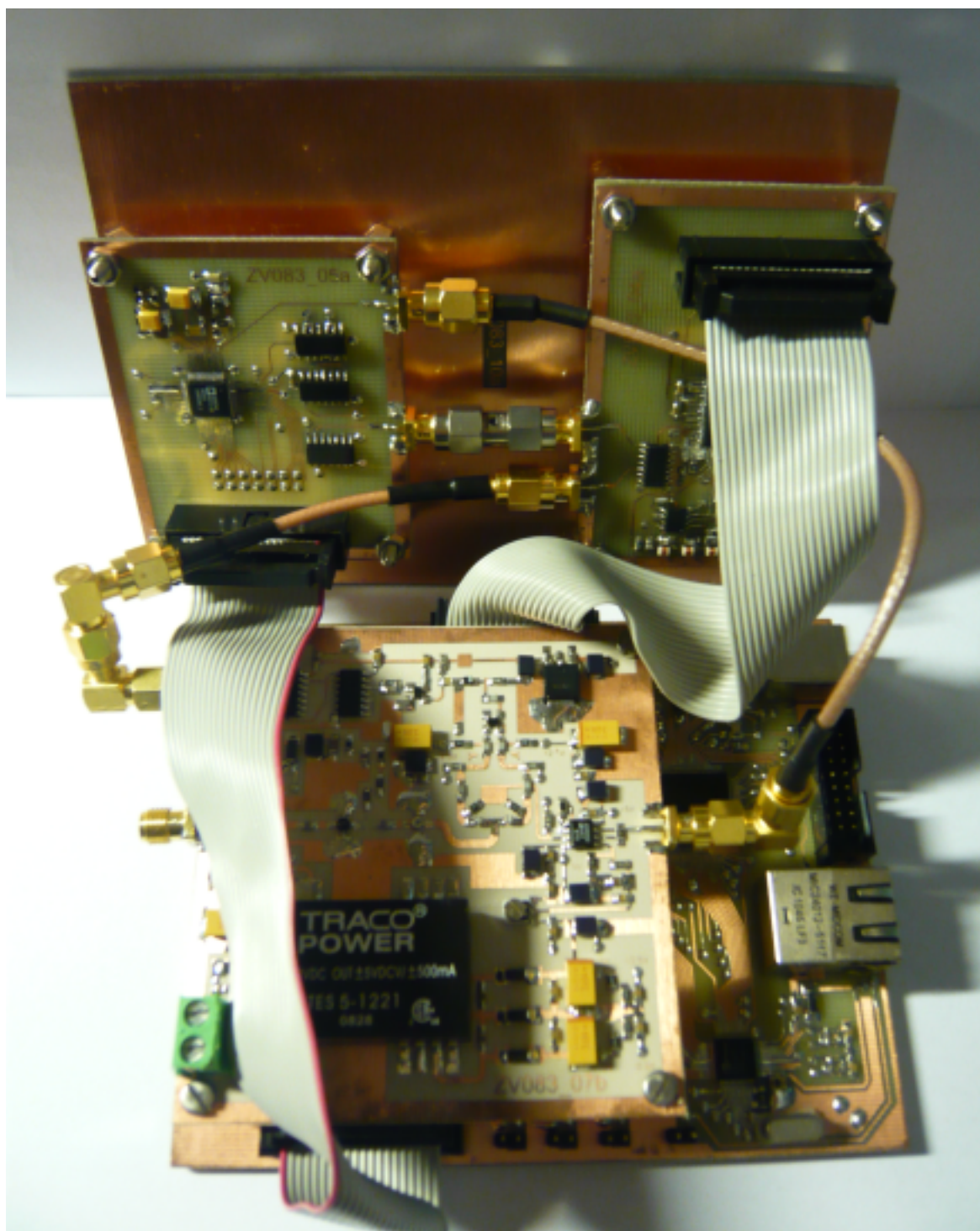
- **DDS a A/D převodník na montážní desce**



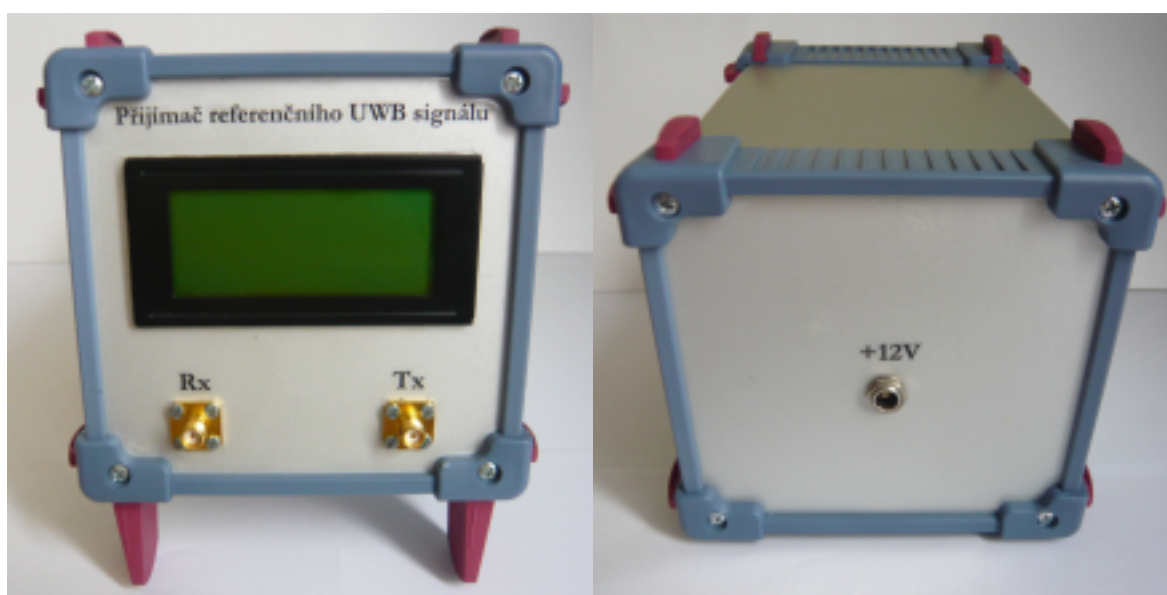
- **Základní deska se vzorkovačem**



- **Sestavení zařízení**



**Vnější zpracování UWB přijímače**



**Výpis programu pro FPGA**

```
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
use ieee.std_logic_arith.ALL;
use ieee.std_logic_unsigned.ALL;

entity REFUWB is
Port (
    CLK : in std_logic;
    DDS_IO_RESET : out std_logic;
    DDS_MASTER_RESET : out std_logic;
    DDS_SDIO : out std_logic;
    DDS_UDCLK : out std_logic;
    DDS_SCLK : out std_logic;
    SCLK: in std_logic;
    POVEL: in std_logic;
    VYSTAV: out std_logic;
    DATA_z_AD: in std_logic_vector (13 downto 0);
    DATA_z_FPGA: out std_logic_vector (7 downto 0)
);
end REF;
architecture ZAVES of REFUWB is
    signal programovaci_vektor : std_logic_vector(1 downto 0):="00";
    signal count : std_logic_vector(1 downto 0):="00";
    signal vensignal_F : std_logic_vector(95 downto 0):=x"07002000400220000000000000";
    signal fazovaniDDS : std_logic_vector(7 downto 0):="00000000";
    type stav is (s0,s1,s2,s3,s4,s5,s6);
    signal stav, stav0, stav1, stav2, stavDDS: stav :=s0;
    signal fazovaniDDS2 : std_logic_vector(7 downto 0):="00000000";
    signal vensignal_P : std_logic_vector(23 downto 0):=x"000000";
    signal POVEL_POM: std_logic:= '0';
    signal citac: std_logic_vector (2 downto 0):="000";
    type banky is (banka1,banka2);
    signal banka: banky :=banka1;
    signal max1: std_logic_vector (13 downto 0):="0000000000000000";
    signal min1: std_logic_vector (13 downto 0):="1111111111111111";
    signal pozmax1: std_logic_vector (13 downto 0):="0000000000000000";
    signal pozmin1: std_logic_vector (13 downto 0):="0000000000000000";
    signal max2: std_logic_vector (13 downto 0):="0000000000000000";
    signal min2: std_logic_vector (13 downto 0):="1111111111111111";
    signal pozmax2: std_logic_vector (13 downto 0):="0000000000000000";
    signal pozmin2: std_logic_vector (13 downto 0):="0000000000000000";

    begin
    process (CLK)
    begin
    if ((CLK='1')and(CLK'event)) then
        count<=count+1;
        if (stavDDS=s2) then
            VYSTAV<='0';
        end if;
    end if;
    end process;
end ZAVES;
```

```
end if;
if (count="01") then
  if (stav=s0) then
    if (stav0=s0) then
      DDS_MASTER_RESET<='1';
      DDS_UDCLK <='0';
      stav0<=s1;
    end if;
    if (stav0=s1) then
      DDS_MASTER_RESET<='0';
      stav0<=s2;
    end if;
    if (stav0=s2) then
      DDS_IO_RESET<='1';
      stav0<=s3;
    end if;
    if (stav0=s3) then
      DDS_IO_RESET<='0';
      stav0<=s4;
    end if;
    if (stav0=s4) then
      fazovaniDDS<=fazovaniDDS+1;
      case fazovaniDDS(0) is
        when '0' => programovaci_vektor(0)<=vensignal_F(95);
          programovaci_vektor(1)<='0';
        when others => programovaci_vektor(1)<='1';
          vensignal_F<=vensignal_F(94 downto 0) & vensignal_F(95);
      end case;
      if (fazovaniDDS(7 downto 1)=95) then
        stav0<=s5;
        programovaci_vektor<="00";
      end if;
    end if;
  end if;
  if (stav0=s5) then
    DDS_UDCLK<='1';
    stav0<=s6;
  end if;
  if (stav0=s6) then
    DDS_UDCLK<='0';
    stav<=s1;
    stav1<=s0;
  end if;
end if;
end if;
if (stav=s1) then
  fazovaniDDS2<="00000000";
  if (stav1=s0) then
    DDS_IO_RESET<='1';
    stav1<=s1;
  end if;
  if (stav1=s1) then
    DDS_IO_RESET<='0';
    stav1<=s2;
```

```
end if;
if (stav1=s2) then
    fazovaniDDS2<=fazovaniDDS2+1;
    case fazovaniDDS2(0) is
        when '0' => programovaci_vektor(0)<=vensignal_P(23);
            programovaci_vektor(1)<='0';
        when others => programovaci_vektor(1)<='1';
            vensignal_P<=vensignal_P(22 downto 0) & vensignal_P(23);
    end case;
    if (fazovaniDDS2(5 downto 1)=23) then
        stav1<=s3;
        programovaci_vektor<="00";
        vensignal_P<=vensignal_P(22 downto 0) & vensignal_P(23);
    end if;
end if;
if (stav1=s3) then
    DDS_UDCLK<='1';
    stav1<=s4;
end if;
if (stav1=s4) then
    DDS_UDCLK<='0';
    stav<=s2;
    stav2<=s0;
end if;
end if;
if (stav=s2) then
    if (stav2=s0) then
        if (stavDDS=s2) then
            VYSTAV<='0';
        end if;
        if (banka=banka1) then
            if (max1<DATA_z_AD) then
                max1<=DATA_z_AD;
                pozmax1<=vensignal_P(13 downto 0);
            end if;
            if (min1>DATA_z_AD) then
                min1<=DATA_z_AD;
                pozmin1<=vensignal_P(13 downto 0);
            end if;
            vensignal_P<=vensignal_P+1;
            if (vensignal_P(13 downto 0)="111111111111") then
                if (stavDDS=s0) then
                    VYSTAV<='1';
                    vensignal_P(13 downto 0)<="00000000000000";
                    banka<=banka2;
                    max2<="00000000000000";
                    min2<="11111111111111";
                    pozmax2<="00000000000000";
                    pozmin2<="00000000000000";
                else
                    vensignal_P(13 downto 0)="11111111111110";
                end if;
            end if;
        end if;
    end if;
end if;
```

```

        end if;
        stav<=s1;
        stav1<=s0;
    end if;
    if (banka=banka2) then
        if (max2<DATA_z_AD) then
            max2<=DATA_z_AD;
            pozmax2<=vensignal_P(13 downto 0);
        end if;
        if (min2>DATA_z_AD) then
            min2<=DATA_z_AD;
            pozmin2<=vensignal_P(13 downto 0);
        end if;
        vensignal_P<=vensignal_P+1;
        if (vensignal_P(13 downto 0)="11111111111111") then
            if (stavDDS=s0) then
                VYSTAV<='1';
                vensignal_P(13 downto 0)<="00000000000000";
                max1<="00000000000000";
                min1<="11111111111111";
                pozmax1<="00000000000000";
                pozmin1<="00000000000000";
                banka<=banka1;
            else
                vensignal_P(13 downto 0)="11111111111110;
            end if;
        end if;
        stav<=s1;
        stav1<=s0;
    end if;
end if;
end if;
end if;
end if;
end process;
process (SCLK)
begin
    if ((SCLK='1')and(SCLK'event)) then
        stavDDS<=s2;
        if (POVEL='0') then
            citac<=citac+1;
            if (POVEL_POM='0') then
                if (citac="000") then
                    DATA_z_FPGA <="00" & max1(13 downto 8);
                end if;
                if (citac="001") then
                    DATA_z_FPGA <=max1(7 downto 0);
                end if;
                if (citac="010") then
                    DATA_z_FPGA <="00" & min1(13 downto 8);
                end if;
                if (citac="011") then

```

```
        DATA_z_FPGA <=min1(7 downto 0);
    end if;
    if (citac="100") then
        DATA_z_FPGA <="00" & pozmax1(13 downto 8);
    end if;
    if (citac="101") then
        DATA_z_FPGA <=pozmax1(7 downto 0);
    end if;
    if (citac="110") then
        DATA_z_FPGA <="00" & pozmin1(13 downto 8);
    end if;
    if (citac="111") then
        DATA_z_FPGA <=pozmin1(7 downto 0);
        stavDDS<=s0;
        POVEL_POM<='1';
    end if;
end if;
if (POVEL_POM='1') then
    if (citac="000") then
        DATA_z_FPGA <="00" & max2(13 downto 8);
    end if;
    if (citac="001") then
        DATA_z_FPGA <=max2(7 downto 0);
    end if;
    if (citac="010") then
        DATA_z_FPGA <="00" & min2(13 downto 8);
    end if;
    if (citac="011") then
        DATA_z_FPGA <=min2(7 downto 0);
    end if;
    if (citac="100") then
        DATA_z_FPGA <="00" & pozmax2(13 downto 8);
    end if;
    if (citac="101") then
        DATA_z_FPGA <=pozmax2(7 downto 0);
    end if;
    if (citac="110") then
        DATA_z_FPGA <="00" & pozmin2(13 downto 8);
    end if;
    if (citac="111") then
        DATA_z_FPGA <=pozmin2(7 downto 0);
        stavDDS<=s0;
        POVEL_POM<='0';
    end if;
end if;
end if;
if (POVEL='1') then
    citac<="000";
end if;
end if;
end process;
```



## Příloha 12

```
DDS_SDIO<=programovací_vektor(0);  
DDS_SCLK<=programovací_vektor(1);  
  
end architecture ZAVES;
```

**Výpis části programu pro procesor**

```
#include "emacif.h"
#include <string.h>
#include <emac/emac.h>
#include <board.h>
#include <pio/pio.h>

#define RS 30
#define RW 29
#define E 28
#define DB0 20
#define DB1 21
#define DB2 22
#define DB3 23
#define DB4 24
#define DB5 25
#define DB6 26
#define DB7 27
#define SCLK 18
#define VYSTAV 17
#define POVEL 19
#define LED 8

char LCD_max[6],LCD_min[6],LCD_pozmax[6],LCD_pozmin[6];
char Umax[6]="Umax=",Umin[6]="Umin=",Pozmax[8]="POZmax=",Pozmin[8]="POZmin=";
char maxH,maxL,minH,minL,pozmaxH,pozmaxL,pozminH,pozminL;
static u16_t max,min,pozmax,pozmin;

void LCD_init (void)
{
    PIN_Set(E);
    LCD_write_znak(0,0x38);
    LCD_write_znak(0,0x0C);
    LCD_write_znak(0,0x06);
}
void LCD_head (void)
{
    LCD_write_znak(0,0x80);
    LCD_write_string(Umax);
    LCD_write_znak(0,0xC0);
    LCD_write_string(Umin);
    LCD_write_znak(0,0x90);
    LCD_write_string(Pozmax);
    LCD_write_znak(0,0xD0);
    LCD_write_string(Pozmin);
}
void Prevod (void)
{
    while(1)
    {
        int VYSTAV_POM=0;
```

---

```
VYSTAV_POM=PIN_Get(VYSTAV);
if (VYSTAV_POM==1)
{
    PIN_Clear(POVEL);
    maxH=Prijimej_BYTE();
    maxL=Prijimej_BYTE();
    minH=Prijimej_BYTE();
    minL=Prijimej_BYTE();
    pozmaxH=Prijimej_BYTE();
    pozmaxL=Prijimej_BYTE();
    pozminH=Prijimej_BYTE();
    pozminL=Prijimej_BYTE();
    PIN_Set(POVEL);

    max=maxH;
    max=max<<8;
    max|=maxL;
    min=minH;
    min=min<<8;
    min|=minL;
    pozmax=pozmaxH;
    pozmax=pozmax<<8;
    pozmax|=pozmaxL;
    pozmin=pozminH;
    pozmin=pozmin<<8;
    pozmin|=pozminL;

    sprintf(LCD_max, "%i", max);
    sprintf(LCD_min, "%i", min);
    sprintf(LCD_pozmax, "%i", pozmax);
    sprintf(LCD_pozmin, "%i", pozmin);

    LCD_write_znak(0,0x85);
    LCD_write_string(LCD_max);
    LCD_write_znak(1, ' ');
    LCD_write_znak(1, ' ');
    LCD_write_znak(0,0xC5);
    LCD_write_string(LCD_min);
    LCD_write_znak(1, ' ');
    LCD_write_znak(1, ' ');
    LCD_write_znak(0,0x97);
    LCD_write_string(LCD_pozmax);
    LCD_write_znak(1, ' ');
    LCD_write_znak(1, ' ');
    LCD_write_znak(0,0xD7);
    LCD_write_string(LCD_pozmin);
    LCD_write_znak(1, ' ');
    LCD_write_znak(1, ' ');
}
}
}
char Prijimej_BYTE(void)
```

```
{
    char hodnota=0, poc=1;
    int i,pom=0;
    PIN_Set(SCLK);

    for (i=0; i<8; i++)
    {
        pom=PIN_Get(9+i);// od MSB dolu
        if (pom>0)
        {
            hodnota=hodnota+0x80/poc;
            pom=0;
        }
        poc=poc*2;
    }
    PIN_Clear(SCLK);
    return hodnota;
}

void LCD_write_znak (char print, char byte)
{
    PIN_Clear(DB0);
    PIN_Clear(DB1);
    PIN_Clear(DB2);
    PIN_Clear(DB3);
    PIN_Clear(DB4);
    PIN_Clear(DB5);
    PIN_Clear(DB6);
    PIN_Clear(DB7);
    PIN_Clear(RW);
    PIN_Clear(RS);
    Sleep(20);
    if (print) PIN_Set(RS);
    if (byte & 0x01) PIN_Set(DB0);
    if (byte & 0x02) PIN_Set(DB1);
    if (byte & 0x04) PIN_Set(DB2);
    if (byte & 0x08) PIN_Set(DB3);
    if (byte & 0x10) PIN_Set(DB4);
    if (byte & 0x20) PIN_Set(DB5);
    if (byte & 0x40) PIN_Set(DB6);
    if (byte & 0x80) PIN_Set(DB7);
    Sleep(20);
    PIN_Clear(E);
    Sleep(20);
    PIN_Set(E);
    Sleep(20);
}

void LCD_write_string(char *str)
{
    while (*str) LCD_write_znak(1,*(str++));
}
}
```

---