

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Jan Vykydal



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIOENGINEERING

DIGITÁLNÍ ZPRACOVÁNÍ SIGNÁLU POMOCÍ PROGRAMOVATELNÝCH HRADLOVÝCH POLÍ

DIGITAL SIGNAL PROCESSING USING FIELD PROGRAMMABLE GATE ARRAYS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Vykydal

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. Zdeněk Kolka

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Jan Vykydal

ID: 186240

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Digitální zpracování signálu pomocí programovatelných hradlových polí

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je realizace platformy pro digitalizaci a zpracování signálu s využitím FPGA a dodatečných obvodů s aplikací v kvantových experimentech. Na základě studia literatury navrhnete vhodný způsob převodu snímaného signálu do digitální oblasti (komparátory, multikanálové analyzátoři či analogově digitální převodníky) a jeho efektivní přenos do FPGA s důrazem na rychlost a rozlišení vzorkování vstupního signálu, flexibilitu digitálního zpracování a minimalizaci celkové latence. Typickým vstupním signálem jsou pulsy s dobou trvání od 20ns a amplitudu 0,1 až 2V. Dále navrhnete implementaci základních bloků pro zpracování digitalizovaného signálu v FPGA.

Na bázi navržené platformy realizujte zařízení k analýze spektra gamma záření na základě vyhodnocení pulzů ze scintilačních detektorů jaderného záření. Aplikace má amplitudy třídít do histogramu a odesílat na příkaz do PC. V rámci testování student ověří parametry zařízení na uměle generovaných i reálných pulzech.

DOPORUČENÁ LITERATURA:

[1] R. Woods, J. McAllister, G. Lightbody, Y. Yi, FPGA-based Implementation of Signal Processing Systems, druhé vydání, Wiley 2017.

[2] Photomultiplier tubes - basics and applications, Hamamatsu photonics K.K. 2017, [cit. 10.1.2020]. Dostupné z: https://www.hamamatsu.com/resources/pdf/etd/PMT_handbook_v3aE.pdf

Termín zadání: 3.2.2020

Termín odevzdání: 6.8.2020

Vedoucí práce: prof. Dr. Ing. Zdeněk Kolka

Konzultant: Mgr. Michal Dudka, RNDr. Miroslav Ježek, Ph.D.

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se věnuje návrhu, realizaci a testování zařízení, které provádí spektrální analýzu gama záření na základě vyhodnocování pulzů ze scintilačního detektoru. Ty jsou po zesílení zdigitalizovány a jejich další zpracování probíhá číslicově v hradlovém poli, což umožňuje snadnou případnou budoucí modifikaci funkce vyvinutého zařízení.

Po úvodu do problematiky spektroskopie gama záření se zaměřením na jeho detekci se práce věnuje vývoji hardwaru multikanálového analyzátoru, jehož jednotlivé části jsou postupně rozebrány. Dále je popsán vývoj číslicového systému pro zpracování signálu v programovatelném hradlovém poli. Následuje rozbor firmwaru řídicího mikrokontroléru a textového protokolu pro ovládání zařízení. Na závěr jsou diskutovány výsledky práce zejména testovacího měření gama záření.

KLÍČOVÁ SLOVA

spektrální analýza, gama záření, scintilační detektor, programovatelné hradlové pole, digitální zpracování signálu

ABSTRACT

This thesis deals with the design, implementation and testing of an equipment that performs spectral analysis of a gamma radiation, based on the evaluation of pulses from a scintillation detector. The pulses are pre-amplified and digitized, and their further processing takes place numerically in an FPGA, which allows a simple modification of the function of the developed device.

After an introduction to the issue of gamma radiation spectroscopy with a focus on its detection, the thesis is devoted to the development of a multichannel analyzer hardware, whose individual parts are further described later. Next, the development of a digital signal processing system in the FPGA is described. Followed by an analysis of a microcontroller firmware, and a text protocol for the controlling of the device. Finally, the results of the work are discussed, with a focus on the test measurement of gamma radiation.

KEYWORDS

spectral analysis, gamma radiation, scintillation detector, field programmable gate array, digital signal processing

VYKYDAL, Jan. *Digitální zpracování signálu pomocí programovatelných hradlových polí*. Brno, 2020, 85 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce: prof. Dr. Ing. Zdeněk Kolka

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Digitální zpracování signálu pomocí programovatelných hradlových polí“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval konzultantovi diplomové práce Mgr. Michalu Dudkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Také bych rád poděkoval Mgr. Aleši Stejskalovi za věcné připomínky k fyzikální části práce. Další poděkování patří vedoucímu práce prof. Dr. Ing. Zdeňkovi Kolkovi za dohled nad textovou částí práce. Velký dík patří i Bc. Filipu Kubečkovi za korekturu textu a mým nejbližším za podporu.

Obsah

Úvod	12
1 Spektroskopie gama záření	13
1.1 Ionizující záření	13
1.2 Detektory ionizujícího záření	13
1.3 Příprava signálu pro multikanálovou analýzu	14
1.4 Multikanálová analýza	15
2 Návrh hardware	16
2.1 Blokový popis zařízení	16
2.2 Analogově-digitální převodník	16
2.2.1 Požadované parametry	17
2.2.2 Integrovaný obvod AD9214	17
2.2.3 Vlastnosti integrovaného obvodu AD9214	18
2.2.4 Aplikace integrovaného obvodu AD9214	18
2.3 Vstupní nízkošumový zesilovač	19
2.3.1 Požadované parametry vstupního zesilovače	19
2.3.2 Zesilovač s integrovaným obvodem ADA4930	20
2.3.3 Návrh zesilovače s obvodem ADA4930	21
2.4 Operační paměť s náhodným přístupem	23
2.4.1 Požadované parametry	23
2.4.2 Paměťový čip IS61WV12816EDBL-10TLI	24
2.5 Programovatelné hradlové pole	24
2.5.1 Požadované parametry	24
2.5.2 Hradlové pole XC7S15	25
2.5.3 Konfigurace XC7S15	25
2.5.4 Blokování napájení XC7S15	26
2.6 Mikrokontrolér	27
2.6.1 Požadované parametry	27
2.6.2 Mikrokontrolér STM32H750VBT6	27
2.7 Napájecí systém	28
2.7.1 Požadavky na stabilizátory napětí	28
2.7.2 Stabilizátor NCP59744	29
2.7.3 Řízení napájecího systému	30
2.7.4 Integrovaný sekvencer napájení LM3880	31
2.7.5 Ochranné prvky	32
2.7.6 Odrušovací filtr	32

3	Systém v hradlovém poli	34
3.1	ADC driver	34
3.2	SRAM driver	35
3.3	SPI driver	37
3.3.1	Popis sběrnice SPI	37
3.3.2	Realizace SPI slave driveru	38
3.4	Resetovací obvod	39
3.5	Zpracování signálu	40
3.5.1	Detekce pulzů	40
3.5.2	Měření amplitudy pulzů	40
3.5.3	Měření šířky pulzů	41
3.5.4	Realizace DSP jednotky	41
3.6	Řídící jednotka	42
3.6.1	SPI příkazy	42
3.6.2	Realizace řídicí jednotky	44
4	Firmware	46
4.1	Popis firmware	46
4.1.1	Inicializace	46
4.1.2	Superloop	46
4.1.3	SPI driver	46
4.1.4	SPI protokol	47
4.1.5	FTDI driver	47
4.1.6	FTDI protokol	47
4.1.7	Clock driver	47
4.1.8	Callback při příjmu znaku jednotkou UART	47
4.2	Příkazy pro komunikaci s PC	47
5	Výsledky	50
5.1	Testování šumu systému	50
5.2	Testování digitalizace signálu z detektoru	50
5.3	Ovládací aplikace	51
5.4	Ověřovací měření spektra	52
	Závěr	55
	Literatura	56
	Seznam symbolů, veličin a zkratk	58
	Seznam příloh	59

A	Přiložená schémata zapojení	60
B	Přiložené osazovací plány	69
C	Přiložený výkres plošného spoje	72
D	Bloková schémata číslicového systému v hradlovém poli	77
E	Fotografie zařízení	80
F	Zdrojové soubory	85

Seznam obrázků

1.1	Blokové schéma multikanálové analýzy gama záření.	14
1.2	Blokové schéma scintilačního detektoru.	14
1.3	Schématické znázornění získání napětí ze scintilačního detektoru.	15
2.1	Blokové schéma multikanálového analyzátoru.	16
2.2	Blokové schéma AD9214 převzato z [1].	17
2.3	Voltampérová charakteristika TLV431, převzato z [11].	19
2.4	Schéma pro výpočty nízko-šumového zesilovače.	20
2.5	Odsimulovaná závislost U_{IN+} a U_{IN-} na vstupním napětí LNA.	23
2.6	Odsimulovaná modulová přenosová charakteristika LNA.	24
2.7	Blokové schéma napájecího systému.	28
2.8	Blokové schéma obvodu NCP59744.	30
2.9	Blokové schéma obvodu LM3880, převzato z [10].	31
2.10	Časové průběhy sekvencí LM3880MF-1xx, převzato z [10].	32
3.1	Blokové schéma systému v FPGA.	34
3.2	Časování ADC AD9214, převzato z [1].	35
3.3	Entita navrženého ADC driveru.	36
3.4	Entita navrženého SRAM driveru.	37
3.5	Entita navrženého SPI driveru.	39
3.6	Ukázka jednoho komunikačního rámce SPI.	39
3.7	Entita navrženého resetovacího obvodu.	40
3.8	Ukázka měření pulzů nad daty ze scintilačního detektoru.	41
3.9	Entita navržené DSP jednotky.	42
3.10	SPI příkaz STOP.	42
3.11	SPI příkaz GET STATE.	43
3.12	SPI příkaz MEMORY ERASE.	43
3.13	SPI příkaz MEMORY READ.	43
3.14	SPI příkaz MEASUREMENT START.	44
3.15	SPI příkaz SET RISING LEVEL.	44
3.16	SPI příkaz SET FALLING LEVEL.	44
3.17	Entita řídicí jednotky.	45
5.1	Histogram hodnot vzorků měřených na výstupu ADC driveru.	50
5.2	Zachycené pulzy z detektoru na výstupu ADC driveru v FPGA.	51
5.3	Ukázka GUI vyvinuté aplikace.	52
5.4	Aparatura pro spektroskopii gama záření.	53
5.5	Změřená spektra gama záření.	54
A.1	Schéma zapojení – strana 1. Blokové schéma.	61
A.2	Schéma zapojení – strana 2. Napájení.	62

A.3	Schéma zapojení – strana 3. MCU.	63
A.4	Schéma zapojení – strana 4. USB.	64
A.5	Schéma zapojení – strana 5. LNA.	65
A.6	Schéma zapojení – strana 6. SRAM.	66
A.7	Schéma zapojení – strana 7. ADC.	67
A.8	Schéma zapojení – strana 8. FPGA.	68
B.1	Osazovací plán digitizéru – pohled na stranu součástek.	70
B.2	Osazovací plán digitizéru – pohled na stranu spojů.	71
C.1	Plošný spoj – vrstva L1. Signály.	73
C.2	Plošný spoj – vrstva L2. Rozlitaná zem.	74
C.3	Plošný spoj – vrstva L3. Napájení.	75
C.4	Plošný spoj – vrstva L4. Signály.	76
D.1	Blokové schéma systému v hradlovém poli.	78
D.2	Blokové schéma bloku DSP v hradlovém poli.	79
E.1	Fotografie prototypů modulů a vývojového kitu s FPGA.	81
E.2	Fotografie zařízení, pohled na stranu součástek.	82
E.3	Fotografie zařízení, pohled na stranu spojů.	83
E.4	Fotografie zařízení, pohled na konektory.	84

Seznam tabulek

2.1	Konfigurace maximální diferenciální hodnoty vstupního napětí a výstupního formátu dat analogově digitálního převodníku AD9214.	18
2.2	Parametry navrženého zesilovače s ADA4930.	20
2.3	Konfigurační možnosti FPGA Spartan-7, převzato z [12].	25
2.4	Chování pinu PUDC_B [12].	26
2.5	Doporučené blokovací kondenzátory pro XC7S15 v pouzdře FTBG196 [13].	26
2.6	Přehled teoretických maximálních hodnot proudů v jednotlivých větvích.	29
3.1	Časování AD9214 hodnoty jsou brány jako nejhorší případ, převzato z [1].	35
3.2	Vstupně-výstupní rozhraní IS61WV12816EDBLL [5].	36
3.3	Přehled signálů SPI sběrnice.	37
3.4	Přehled módů SPI sběrnice.	38
3.5	Stavy zařízení.	43

Úvod

Multikanálová analýza (někdy označovaná jako mnohokanálová) je spektroskopická metoda určená k měření spektra ionizujícího záření. Tato metoda je založena na analyzování velikostí amplitud z proporcionálních detektorů ionizujícího záření a jejich statistickém zpracování formou histogramu.

Jaderná spektroskopie nachází uplatnění v mnoha vědních odvětvích, například ve fyzice při zkoumání struktury jader atomů, v geologii k dataci stáří radiokarbidovou metodou či ve zdravotnictví při výpočetní tomografii (CT).

Zařízení, jehož vývojem se zabývá tato práce, bude sloužit ke spektroskopii gama záření v laboratoři experimentální fyziky. Vyvinuté zařízení zpracovává signál ze scintilačního detektoru upravený transimpedančním zesilovačem. Tento signál je ve vyvinutém zařízení převeden na diferenční nízkošumovým zesilovačem pro buzení analogově-digitálního převodníku, kterým je zdigitalizován. Další zpracování signálu je kompletně číslicové, realizované v programovatelném hradlovém poli, díky čemuž je možné provádět úpravy zpracování signálu bez zásahu do hardwaru.

1 Spektroskopie gama záření

Spektroskopie gama záření, jadernými fyziky označovaná jako multikanálová analýza, je metoda používaná k získání spektra ionizujícího záření. Tato metoda má široké uplatnění například ve zjišťování původu záření či filtraci, pro zpracování určité části spektra. Jelikož je ale za touto metodou velmi obsáhlá část kvantové mechaniky, nebudeme zabíhat do detailů, koneckonců cílem práce je návrh elektronického zařízení, nikoliv částicová fyzika.

1.1 Ionizující záření

Rozlišujeme základní tři typy ionizujícího záření, a to α , β a γ . Vzhledem k tomu, že multikanálovou analýzu provádíme v této práci pouze pro gama záření, omezíme se převážně na jeho popis.

Gama záření je vysokoenergetické elektromagnetické záření vznikající při štěpení atomových jader, případně při interakci atomového jádra s částicemi. Často vzniká při radioaktivním rozpadu atomového jádra doprovázeného vyzářením α nebo β částice. Také vzniká při anihilaci, tedy při zániku elektronu a pozitronu při srážce. Vlastnostmi je srovnatelné se zářením rentgenovým, ačkoliv to má odlišný původ, vzniká při excitaci elektronů v atomovém obalu. Jak rentgenové, tak i gama záření jsou tvořeny na rozdíl od záření α a β vysokoenergetickými fotony. Gama záření má také největší pronikavost ve srovnání se zářením α , které lze odstínit listem papíru, či zářením β , na jehož odstínění je již zapotřebí několik milimetrů široký kovový plech [9]. Naproti tomu k odstínění γ záření je potřeba několik metrů betonu, přičemž potřebná šířka stínění závisí na konkrétní energii částic.

1.2 Detektory ionizujícího záření

K získání spektra ionizujícího záření je třeba převést záření na elektrický signál, který může být elektronicky zpracováván a vyhodnocen. K tomuto účelu se používají detektory. Detektory můžeme rozdělit do dvou kategorií, a to na detektory nespektrometrické a spektrometrické (proporcionální) [8].

Nespektrometrické detektory poskytují pouze informaci o detekci částice a pro multikanálovou analýzu nemají příliš praktické využití. Z proporcionálních detektorů lze získat i informaci o energii detekované částice. Mezi proporcionální detektory se řadí scintilační, plynové a polovodičové detektory.

Scintilační detektory slouží převážně k detekci γ záření. Oproti polovodičovým a plynovým detektorům vynikají při detekci vysokoenergetického γ záření s účinností dosahující od desítek do sta procent, v závislosti na energii detekovaných částic [8].

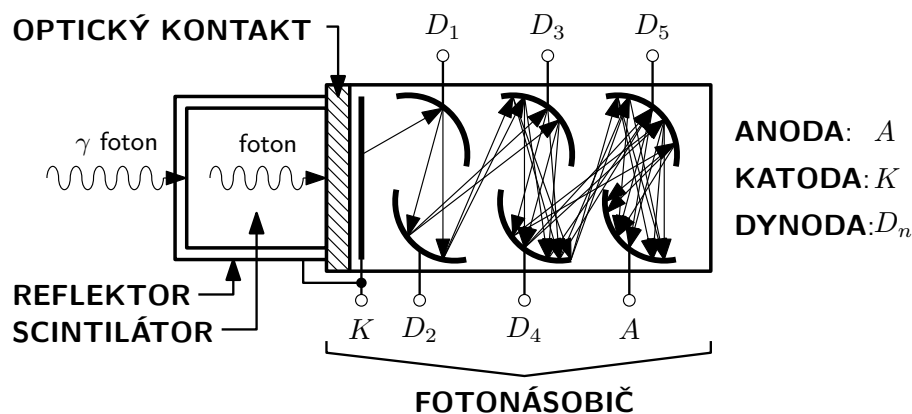
1.3 Příprava signálu pro multikanálovou analýzu

Gama záření je vyzařováno zářičem, například izotopem kobaltu ^{57}Co , rozpadajícím se na izotop železa ^{57}Fe . Ze zářiče vyletují γ fotony o různých definovaných energiích, závislých na typu zářiče. Tyto fotony dopadají do scintilačního detektoru. Pokud dojde ke srážce fotonu s elektrony ve scintilačním krystalu, tak jim γ foton předá svou energii. Tím jsou elektrony excitovány do vyšších energetických slupek atomového obalu. Po krátkém čase dojde k deexcitaci. Elektrony se zbaví energie, která jim byla interakcí s γ fotonem předána tím, že vyzáří foton s vlnovou délkou viditelného světla a vrátí se na nižší energetickou hladinu. Vlnová délka vyzářeného fotonu odpovídá energii, kterou elektron uvolnil.



Obr. 1.1: Blokové schéma multikanálové analýzy gama záření.

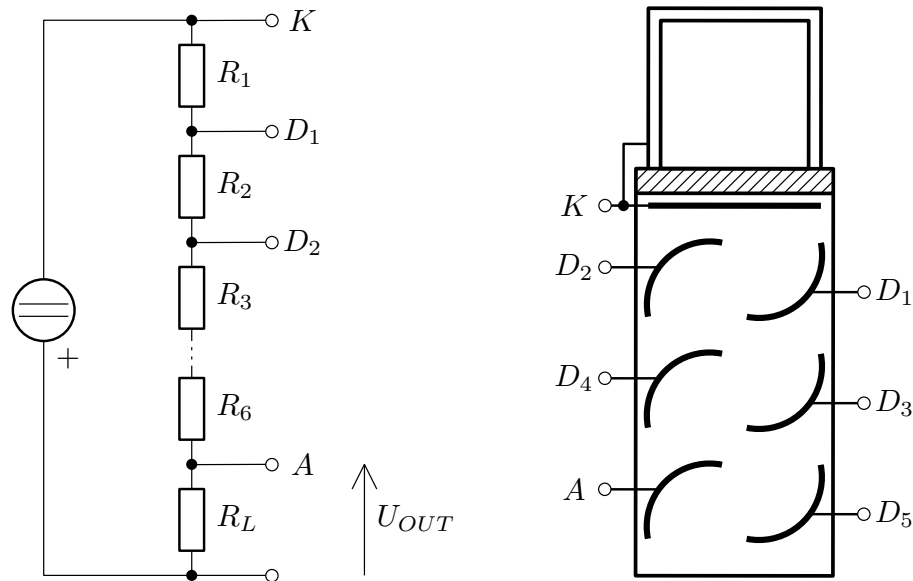
Díky tomu, že fotony vyzařované ze scintilačního krystalu mají větší vlnovou délku než γ fotony, není tak těžké je zachytit na katodě, na které dochází k fotoelektrickému jevu. Pokud se scintilačním krystalem interaguje jeden γ foton, tak po výše popsaném procesu z něj vyletí několik fotonů s vlnovou délkou odpovídající viditelnému záření, tedy s nižší energií. Součet energií všech vystupujících fotonů je přímo úměrný energii vstupujícího γ fotonu.



Obr. 1.2: Blokové schéma scintilačního detektoru.

Fotony vylétající ze scintilačního krystalu, prolétají optickým kontaktem a dopadají na katodu fotonásobiče, na kterou je přivedeno záporné napětí. Fotony předávají svou energii elektronům ve fotokatodě. Ty z ní díky získané energii vylétají a dále jsou urychlovány elektrickým polem, které je způsobeno vysokým napětím mezi katodou, dynodami a anodou. Při dopadu elektronů na dynody dochází k sekundární

emisi elektronů, při níž každý elektron vyrazí dalších několik elektronů. Díky tomuto procesu dosahuje fotonásobič obvykle zesílení 10^5 až 10^9 , podle konstrukce [3].



Obr. 1.3: Schématické znázornění získání napětí ze scintilačního detektoru.

Na anodě se tedy hromadí záporný náboj. Tento nahromaděný náboj můžeme například připojením rezistoru k anodě převést na elektrické napětí, jehož časový průběh nám poskytuje informace o energii γ fotonu. Signál je poté ještě zesílen zesilovačem, který slouží i k impedančnímu přizpůsobení, a poté je již přiveden na vstup zařízení, jehož vývojem se zabývá tato práce.

1.4 Multikanálová analýza

Multikanálová analýza je metoda, která slouží k získání spektra ionizujícího záření. Je založena na analýze velikosti amplitud pulzů na výstupu proporcionálního detektoru. Čím větší je amplituda pulzu, tím větší byla energie detekované částice.

Multikanálový analyzátor měří velikosti amplitud pulzů a následně na základě změřených hodnot amplitud inkrementuje příslušné kanálové čítače. Výsledkem měření je tedy histogram četností amplitud, který odpovídá spektru analyzovaného záření.

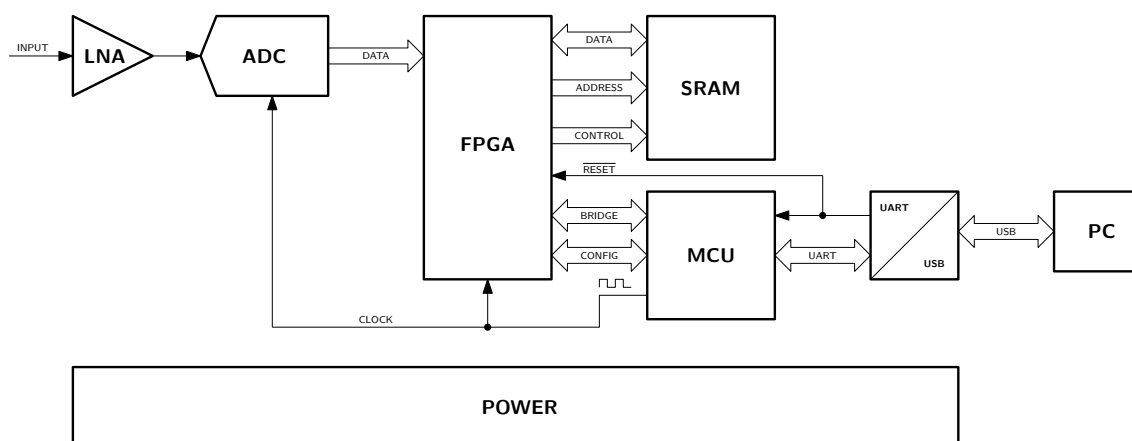
2 Návrh hardware

Tato kapitola se věnuje popisu a návrhu elektronické části multikanálového analyzátoru. Postupně budou diskutovány jednotlivé části zařízení. Základním požadavkem na hardware vyvíjeného zařízení je jeho schopnost digitalizovat analogový signál a umožnit jeho zpracování.

2.1 Blokový popis zařízení

Vstupní signál je nejprve zesílen a převeden na diferenciální pomocí nízkošumového zesilovače (LNA). Následně je zdigitalizován pomocí analogově-digitálního převodníku (ADC). Digitální signál je poté již zpracováván v hradlovém poli (FPGA). Změřená data jsou předána mikrokontroléru (MCU), který je prostřednictvím převodníku UART / USB odešle do osobního počítače (PC).

FPGA i ADC jsou taktovány společným hodinovým signálem, ovládaným MCU. K dočasnému uchovávání zpracovávaného signálu je k FPGA připojena statická operační paměť s náhodným přístupem (SRAM).



Obr. 2.1: Blokové schéma multikanálového analyzátoru.

2.2 Analogově-digitální převodník

Analogově digitální převodník je jednou z klíčových součástí multikanálového analyzátoru. Jeho vlastnosti mají značný vliv na výsledné parametry celého zařízení. Při jeho výběru bylo nutné udělat kompromis mezi vzorkovací frekvencí, počtem bitů, komunikačním rozhraním, odstupem signálu od šumu a cenou.

2.2.1 Požadované parametry

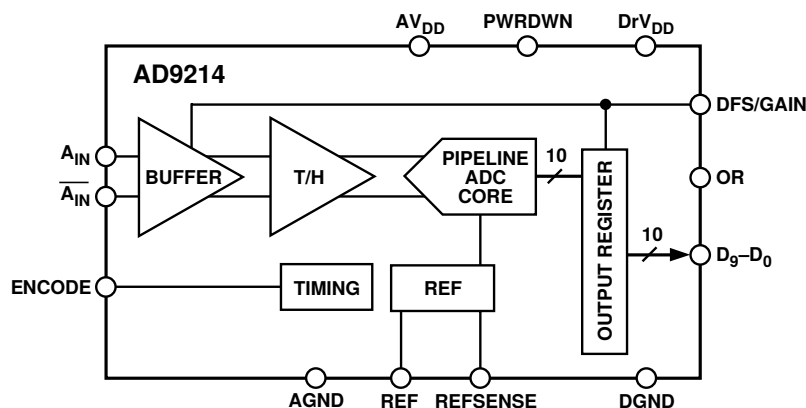
Z komunikačních rozhraní přicházelo v úvahu LVDS, případně paralelní rozhraní. Další rozhraní používaná u ADC, jako například SPI nebo I²C, jsou pro očekávaný datový tok z ADC příliš pomalá. Pro první verzi bylo výhodné využít paralelní rozhraní, jelikož umožňuje snadnější ladění problémů logickým analyzátozem.

Pro multikanálovou analýzu je vhodné mít dostatečné bitové rozlišení, protože umožňuje přesnější měření. Po analýze vstupního signálu jsem se rozhodl, že desetibitové rozlišení je pro tuto aplikaci dostatečné.

Vzorkovací frekvence převodníku musí splňovat vzorkovací teorém. Pro její stanovení byl opět nutný rozbor vstupního signálu. Signál z detektoru byl proto změřen osciloskopem a poté filtrován číslicovou dolní propustí a následně decimován. Výsledkem těchto testů byl závěr, že i při 80 MSPS mají pulzy dostatečný počet vzorků pro analýzu. Průměrná šířka pulzu z provedeného měření byla 960 ns. Avšak v zadání je požadavek na detekování pulzů od šířky 20 ns. Po konzultaci se zadavatelem se ukázalo, že pulzy z detektoru jsou před vstupem do vyvíjené aplikace tvarovány pomocí integrátoru a minimální šířka pulzu 20 ns je značně nadsazená. Proto jsem použil nejrychlejší verzi integrovaného obvodu AD9214 taktovaným na 100 MHz, což je dostatečná vzorkovací frekvence pro tuto aplikaci.

2.2.2 Integrovaný obvod AD9214

Z dostupných převodníků na trhu jsem nakonec vybral integrovaný obvod AD9214. Tento obvod je dostupný ve třech garantovaných minimálních nejvyšších rychlostech vzorkování, a to 68/80/105 MSPS. Tato rozmanitost se jevila jako výhodná, jelikož na první testy se může osadit levnější varianta a ve finálním zařízení se bez změny schématu použije jeho nejrychlejší verze. Jedná se o převodník s pipeline architekturou tvořenou pěti zřetěženými jednotkami.



Obr. 2.2: Blokové schéma AD9214 převzato z [1].

2.2.3 Vlastnosti integrovaného obvodu AD9214

AD9214 je vybaven vstupním oddělovacím zesilovačem s nastavitelným zesílením, které se nastavuje podle maximální špičkové hodnoty napětí vstupního signálu. Na výběr jsou dvě maximální úrovně vstupního signálu, a to $1 V_{PP}$ a $2 V_{PP}$. Úroveň se nastavuje pomocí pinu DFS/GAIN, který zároveň slouží k výběru kódování. Nastavení je shrnuto v tabulce 2.1.

Integrovaný obvod dále obsahuje vzorkovací obvod, jehož výstup je přiveden do samotného pipeline jádra převodníku. Pipeline jádro využívá referenční napětí, které se může použít buď z interní reference, případně se může na pin REF připojit reference externí. O tom, která reference bude použita, rozhoduje napětí přivedené na pin REFSEANSE. Pokud je pin uzemněn, využívá se reference interní, pokud je připojen k AVDD, využívá se reference externí. Jmenovitá hodnota interní reference je 1,25 V. Jestli bychom chtěli využít referenčního napětí, je vyžadována tatáž hodnota s tolerancí $\pm 5\%$.

Tab. 2.1: Konfigurace maximální diferenciální hodnoty vstupního napětí a výstupního formátu dat analogově digitálního převodníku AD9214.

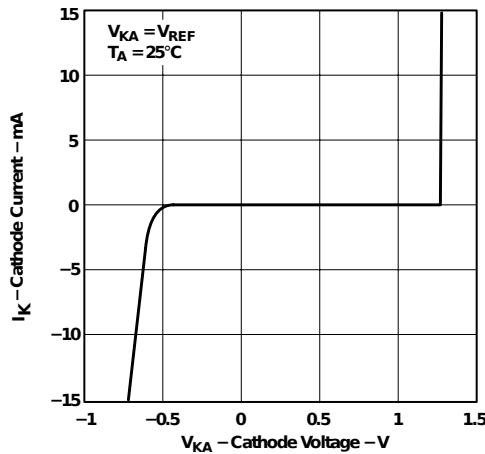
Zapojení DFS/GAIN	Maximální vstupní napětí	Formát dat
AGND	$1 V_{PP}$	binární
AVDD	$1 V_{PP}$	dvojkový doplněk
REF	$2 V_{PP}$	dvojkový doplněk
plovoucí	$2 V_{PP}$	binární

Výstupem převodníku je desetibitové číslo a signál OR, který nás informuje o překročení maximálního rozsahu převodníku. Z převodníku jsou paralelně vyvedeny signály využívající logické úrovně LVCMOS33.

2.2.4 Aplikace integrovaného obvodu AD9214

Abych docílil většího odstupu signálu od šumu, používám převodník v režimu největšího vstupního diferenciálního napětí, tedy $2 V_{PP}$ s binárním kódováním a interní referencí, taktovaný na 100 MHz.

ADC jsem doplnil o externí referenci, abych docílil větší stability referenčního napětí. Navíc převodník taktovaný na maximální frekvenci se zahřívá, proto bylo příhodné umístit referenci od převodníku dále. Jako reference byla použita TLV431. Tato reference poskytuje při vhodném nastavení pracovního bodu stabilní napětí 1,24 V, což je v toleranci zvoleného ADC.



Obr. 2.3: Voltampérová charakteristika TLV431, převzato z [11].

U reference bylo nutné spočítat hodnotu předřadného rezistoru, aby se pracovní bod nacházel ve stabilní části. Po prozkoumání voltampérové charakteristiky 2.3 jsem se rozhodl nastavit pracovní proud na 2 mA. Tím pádem hodnota předřadného rezistoru byla spočítána na 1 k Ω .

$$R = \frac{U_{CC} - U_{REF}}{I} = \frac{3,3 - 1,24}{2 \cdot 10^{-3}} \doteq \underline{\underline{1 \text{ k}\Omega}} \quad (2.1)$$

Aby se zabránilo interferencím odražených signálů od vstupů FPGA ¹, byly digitální výstupy doplněny o sériové rezistory o hodnotě 47 Ω . Šířka spojů byla pomocí integrovaného KiCad kalkulátoru spočítána tak, aby impedance mezi zemní vrstvou a digitálním spojem byla 50 Ω . Díky tomu sériové rezistory pohlcují odrazy od vstupů FPGA. Výsledné schéma je součástí příloh A.7.

2.3 Vstupní nízkošumový zesilovač

Vstupní nízko-šumový zesilovač je prvním členem v řetězci zpracování signálu, viz obrázek 2.1. Proto je důležité, aby do signálu přidával co nejméně šumu, jelikož by tento šum procházel celým řetězcem zpracování signálu a znehodnocoval měřená data. Jeho hlavní úlohou je budít analogově-digitální převodník.

2.3.1 Požadované parametry vstupního zesilovače

Vybraný převodník AD9214 vyžaduje mít diferenciální vstup, ačkoliv vstupní signál je nesymetrický [1]. Zesilovač tedy musí disponovat diferenciálním výstupem.

Vstupní signál je přiváděn pomocí 50 Ω koaxiálního vedení. K eliminaci odrazů na vstupu zesilovače je důležité, aby měl 50 Ω vstupní odpor.

¹Vybrané FPGA nemá možnost vnitřního zakončení do 50 Ω .

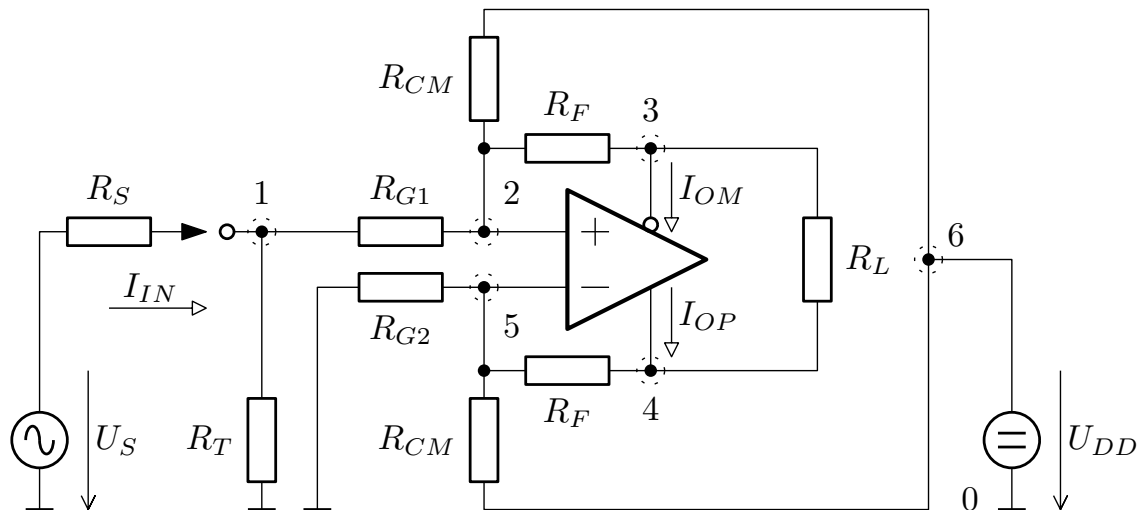
Výstup zesilovače by měl dosahovat maximálně $2 V_{PP}$, jelikož je to maximální vstupní napětí zvoleného *Analog-to-digital converter* (ADC). Dalším parametrem je dostatečná šířka pásma pro zpracování pulzního signálu, která již byla diskutována v návrhu ADC.

2.3.2 Zesilovač s integrovaným obvodem ADA4930

Na základě výše uvedených požadavků jsem zvolil integrovaný zesilovač ADA4930. Použil jsem jej v referenčním zapojení s asymetrickým napájením 3,3 V pro buzení ADC. Kompletní schéma je součástí příloh A.5.

Tab. 2.2: Parametry navrženého zesilovače s ADA4930.

parametr	hodnota
maximální vstupní napětí	$2 V_{PP}$
napěťový zisk	1
vstupní odpor	50Ω
šířka pásma	1,2 GHz
napájecí napětí	3,3 V



Obr. 2.4: Schéma pro výpočty nízko-šumového zesilovače.

Pro střídavou analýzu obvodu jej můžeme popsat následující soustavou rovnic 2.2, která vychází z modifikované metody uzlových napětí. Přitom je důležité si uvědomit, že při střídavé analýze se napěťové zdroje zkratují a proudové rozpojí, tedy uzel 6 je v tomto případě totožný s referenčním uzlem 0. Nejprve vytvoříme klasickou

admitanční matici a poté do ní doplníme rovnice popisující operační zesilovač s diferenciálním výstupem. Obvod zdroje signálu do rovnic nezahrneme, jelikož bychom získali místo hodnoty vstupního odporu hodnotu paralelní kombinace odporu zdroje a vstupního odporu zesilovače.

$$\mathbf{Q} \cdot \mathbf{M}^T = \mathbf{V} \quad (2.2)$$

Kde \mathbf{M} , \mathbf{Q} a \mathbf{V} lze vyjádřit jako:

$$\mathbf{M} = \begin{pmatrix} G_T + G_{G1} & -G_{G1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -G_{G1} & G_F + G_{CM} + G_{G1} & -G_F & 0 & 0 & 0 & 0 & 0 \\ 0 & -G_F & G_F + G_L & -G_L & 0 & 0 & 1 & 0 \\ 0 & 0 & -G_L & G_F + G_L & -G_F & G_F + G_{CM} + G_{G2} & 0 & -1 \\ 0 & 0 & 0 & -G_F & G_F + G_{CM} + G_{G2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{Q} = (U_{10} \ U_{20} \ U_{30} \ U_{40} \ U_{50} \ I_{OP} \ I_{OM}) , \quad \mathbf{V} = (I_{IN} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) .$$

Soustava rovnic popisuje zapojení s ideálním operačním zesilovačem, které je buzeno proudem vtékajícím do uzlu 1. Z ní již jednoduše pomocí determinantů můžeme vyjádřit uzlová napětí potřebná pro další výpočty.

$$U_{10} = \frac{\Delta_{10}}{\Delta}, U_{30} = \frac{\Delta_{30}}{\Delta}, U_{40} = \frac{\Delta_{40}}{\Delta}, U_{DIFF} = U_{40} - U_{30} \quad (2.3)$$

Ze vstupního proudu a napětí odpovídající uzlovému napětí U_{10} dále můžeme odvodit vztah pro vstupní odpor 2.4. Výsledný vztah je příliš dlouhý, proto zde není uveden.

$$R_{IN} = \frac{U_{IN}}{I_{IN}} = \frac{U_{10}}{I_{IN}} \quad (2.4)$$

Z poměru výstupního diferenčního napětí ku napětí vstupnímu získáme napěťový přenos zesilovače 2.5.

$$K_u = \frac{U_{DIFF}}{U_{10}} = \frac{2R_F (R_F R_{CM} + R_F R_{G2} + R_{CM} R_{G2})}{R_F R_{CM} R_{G1} + R_F R_{CM} R_{G2} + 2R_F R_{G1} R_{G2} + 2R_{CM} R_{G1} R_{G2}} \quad (2.5)$$

2.3.3 Návrh zesilovače s obvodem ADA4930

Na základě rovnic uvedených v podkapitole 2.3.2 jsem vytvořil v programovacím jazyce Python skript pro návrh hodnot rezistorů. Tyto rovnice předpokládají, že se zesilovač chová jako ideální, tedy má nekonečně velké zesílení nezávislé na frekvenci, nekonečně velké vstupní odpory a nulové odpory výstupní.

Dalším krokem bylo navrhnout hodnotu napětí na pinu VO_{CM}. Hodnota tohoto napětí umožňuje nastavit úroveň, vůči které jsou vztaženy diferenciální výstupy zesilovače, tento vztah popisuje rovnice 2.6. Toto napětí nemá pro střídavou analýzu význam, a proto tento pin není ani ve schématu 2.4, ze kterého se odvozovalo zesílení a vstupní odpor. Hodnota tohoto napětí závisí na použitém ADC. Jeho technická dokumentace uvádí, že typická hodnota tohoto napětí je jedna třetina napájecího napětí ADC [1]. Napájecí napětí analogových obvodů na této desce je 3,3 V. Abych omezil diverzitu druhů součástek, rozhodl jsem se jako zdroj tohoto napětí použít dělič s doporučeným poměrem, vytvořený ze tří rezistorů stejné hodnoty a pro větší stabilitu opatřený na výstupu kondenzátorem. Hodnotu rezistorů jsem zvolil s ohledem na dostatečný proud děličem, jeho stabilitu při zatížení a také s ohledem na to, abych využil hodnoty rezistorů, které jsou již v obvodu zastoupeny.

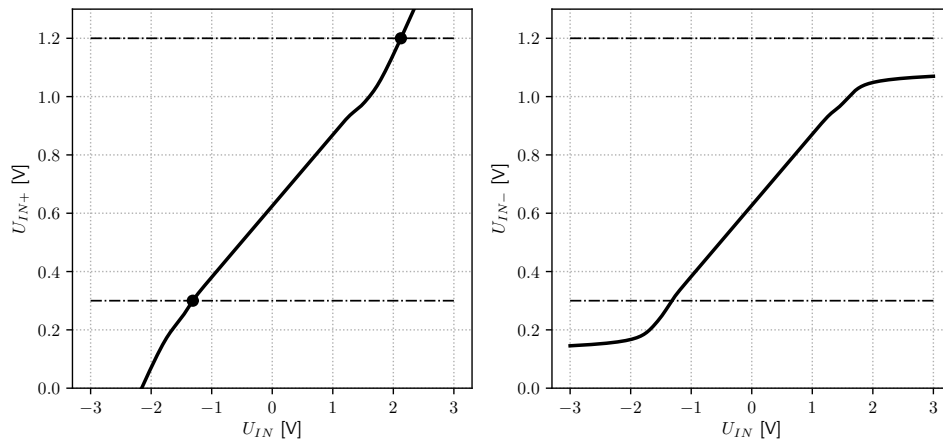
$$U_{VOCM} = \frac{U_{OUT+} - U_{OUT-}}{2} \quad (2.6)$$

Zapojení vytvořené pomocí výpočtů založených na ideálním zesilovači bylo nutné odsimulovat a doladit jeho parametry tak, aby korektně fungovalo s reálným zesilovačem. K tomuto účelu byl použit zdarma dostupný program LTspice. Simulace byla založena na modelu poskytnutém výrobcem integrovaného zesilovače.

Prvním krokem po ověření funkce obvodu bylo doladění hodnoty rezistorů R_{CM} tak, aby zapojení zvládalo pracovat i při záporném vstupním napětí. Tato odolnost je žádoucí, jelikož zesilovač, který upravuje signál z detektoru, má střídavou vazbu a při detekování více pulzů se jeho klidové napětí může posouvat až do hodnoty $-0,5$ V. Rezistory R_{CM} přivádí na vstupy zesilovače kladné předpětí. Pro zajištění korektní činnosti zesilovače je nutné, aby na jeho invertujícím a neinvertujícím vstupu bylo napětí v intervalu 0,3 V až 1,2 V [2]. V grafu 2.5 jsem tyto hodnoty zvýraznil čerchovanou čarou. Jak je z něj patrné, zesilovač tuto podmínku splňuje pro vstupní napětí v intervalu $-1,3$ V až 2,1 V při hodnotě $R_{CM} = 4,7$ k Ω .

Změna R_{CM} ovlivnila hodnotu vstupního odporu, proto dalším krokem bylo doladění hodnoty rezistoru R_T tak, aby hodnota vstupního odporu byla 50 Ω . Toho bylo docíleno sledováním poklesu vstupního napětí na polovinu v uzlu 1 ve schématu 2.4, přičemž hodnota rezistoru R_S byla 50 Ω .

Následovala střídavá analýza. Pro simulaci byl diferenciální výstup zatížen rezistorem o jmenovité hodnotě 1 k Ω . Na obrázku 2.6 je zobrazená odsimulovaná modulová přenosová charakteristika se zvýrazněným poklesem o 3 dB. Odsimulované výsledky odpovídají přenosové charakteristice uvedené v technické dokumentaci k tomuto obvodu [2].



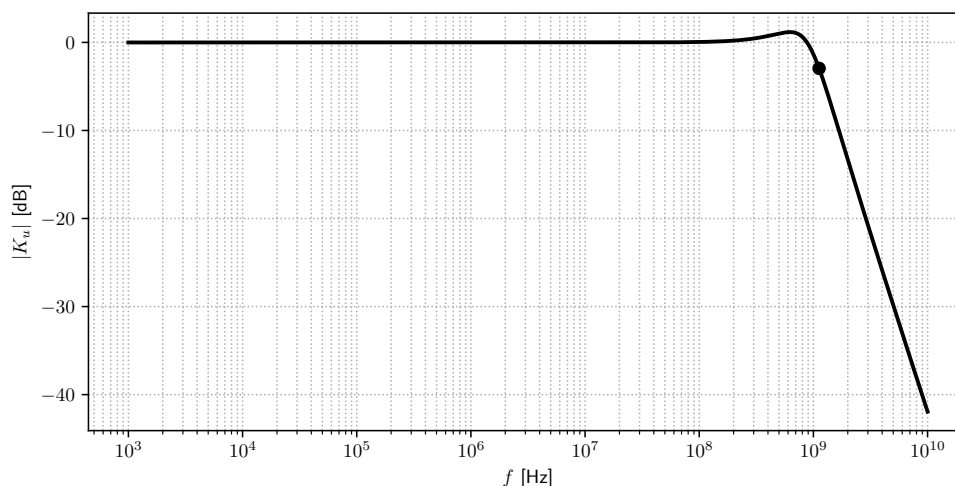
Obr. 2.5: Odsimulovaná závislost U_{IN+} a U_{IN-} na vstupním napětí LNA.

2.4 Operační paměť s náhodným přístupem

Operační paměť je v multikanálovém analyzátoru využívána k ukládání zpracovávaných dat před odesláním do mikrokontroléru, případně k ukládání většího objemu dat, se kterým se dále pracuje. Tato paměť je volatilní, tedy po odpojení napájení je její obsah ztracen. Operační paměti jsou dnes dostupné ve dvou odlišných typech, a to statické a dynamické. Statické paměti si po celou dobu připojení napájecího napětí pamatují svůj obsah. Naproti tomu obsah paměti dynamické je třeba neustále obnovovat, nebo bude zapomenut. Díky této vlastnosti jsou dynamické paměti konstrukčně jednodušší, zabírají méně místa na čipu a jsou levnější, na druhou stranu statická paměť má snadnější ovládání.

2.4.1 Požadované parametry

Multikanálová analýza vyžaduje poměrně velké množství paměti odpovídající $C2^N$ bitů, kde N je počet bitů ADC a C je počet bitů čítače četností. Pokud bychom uvažovali desetibitový ADC a šestnáctibitový čítač, tak bychom potřebovali minimálně 16 384 b. Požadavky na velikost paměti tedy exponenciálně rostou s počtem bitů ADC. Pro desetibitový převodník by sice stačila integrovaná BRAM v FPGA, ale pro možnost používat integrovaný logický analyzátor, a především proto, aby navržená deska měla dostatek paměti pro budoucí funkce, je třeba vybrat externí paměť s dostatečnou rezervou. Dalším parametrem je rychlost paměti, ta je v případě multikanálové analýzy dána především frekvencí, se kterou jsou generovány analyzované pulzy.



Obr. 2.6: Odsimulovaná modulová přenosová charakteristika LNA.

2.4.2 Paměťový čip IS61WV12816EDBLL-10TLI

Na základě výše uvedených požadavků jsem si vybral integrovaný obvod označený IS61WV12816EDBLL-10TLI, jedná se o paměť typu SRAM, která je asynchronní, nemá tedy žádný hodinový signál. Vybraná paměť disponuje kapacitou 2 Mib organizovanou $16 \cdot 128$ kib. Šestnáctibitová datová sběrnice má tu výhodu, že v jednom taktu lze vyčíst celý kanálový čítač. Přístupová doba do paměti je 10 ns. Pokud by bylo třeba přístupovou dobu zkrátit, vyrábí se i o něco dražší verze s přístupovým časem o 2 ns kratším.

2.5 Programovatelné hradlové pole

Programovatelné hradlové pole (FPGA) je integrovaný obvod určený pro realizování operací, pro které se dosud nevyrábějí specializované integrované obvody. Také se může používat k prototypům obvodů, než se začnou sériově vyrábět. Použití FPGA má význam tehdy, když klasické mikrokontroléry již nejsou dostatečně rychlé, případně by bylo obtížné na nich zaručit požadavky na přesné časování aplikace.

2.5.1 Požadované parametry

Je třeba vzít v úvahu, že FPGA bude vyčítat data z ADC, zpracovávat je, mezivýsledky ukládat do SRAM a také komunikovat s MCU. Největší datový tok je očekáván právě z ADC. Vybraný analogově digitální převodník AD9214 může být osazen ve variantě se vzorkovací frekvencí až 105 MSPS. Tato frekvence určuje minimální frekvenci, na které musí být schopné FPGA pracovat. Vzhledem k časování

AD9214 bylo nutné fázově posunout hodiny uvnitř FPGA, proto bylo nezbytné, aby vybrané FPGA disponovalo jednotkou pro správu hodinového signálu, která tohle nastavení podporuje. Dalším požadavkem byl dostatečný počet IOB pro připojení ADC, SRAM a komunikačního rozhraní s MCU. Posledním kritériem byla cena hradlového pole.

2.5.2 Hradlové pole XC7S15

S ohledem na výše uvedená kritéria bylo vybráno hradlové pole Spartan-7 od společnosti Xilinx. Konkrétně se jedná o typ XC7S15 v pouzdře FTBGA196. K urychlení vývoje jsem si pořídil vývojovou desku od společnosti Qmtech, osazenou právě tímto hradlovým polem, jelikož k tomuto kitu poskytuje i kompletní dokumentaci [7].

Toto hradlové pole disponuje 8 000 šestivstupými LUT, pro realizaci logických funkcí nebo distribuované paměti (150 kib), 16 000 klopnými obvody, 100 vstupně-výstupními porty, 20 DSP jednotkami, které dokáží najednou násobit až $25 \cdot 18$ bitů. Dále obsahuje dvě jednotky MMCM a 360 kiB blokové operační paměti.

2.5.3 Konfigurace XC7S15

FPGA má možnost být nakonfigurováno prostřednictvím několika rozhraní, shrnutých v tabulce 2.3. Volba rozhraní je provedena nastavením logické úrovně pinů M2, M1 a M0. Tyto piny mají interní pull-up rezistory. JTAG je možné použít ke konfiguraci FPGA, i pokud je nastaveno rozhraní jiné, což je chování výhodné zejména při vývoji.

Tab. 2.3: Konfigurační možnosti FPGA Spartan-7, převzato z [12].

Konfigurační mód	M[2:0]	Šířka sběrnice	Chování pinu CCLK
Master Serial	000	x1	výstup
Master SPI	001	x1, x2, x4	výstup
Master BPI	010	x8, x16	výstup
Master SelectMAP	100	x8, x16	výstup
JTAG	101	x1	nevyužitý
Slave SelectMAP	110	x8, x16, x32	vstup
Slave Serial	111	x1	vstup

Vzhledem k malému počtu vodičů jsem se rozhodl použít rozhraní SPI se šířkou sběrnice x4. Dalším krokem bylo zjistit paměťové nároky na uložení konfigurace pro zvolené FPGA. Minimální velikost paměti pro XC7S15 je dle výrobce 8 Mib [12].

Z vývojového prostředí Vivado jsem vygeneroval seznam podporovaných pamětí. Tyto paměti jsem vyhledal na stránkách prodejců elektronických součástek a na základě zjištěné minimální kapacity paměti, rozhraní, pracovního napětí, ceny a dostupnosti jsem nakonec vybral paměť IS25LP064A.

Po zapnutí a během nahrávání konfigurace do FPGA je možné vybrat, v jakém stavu se mají nacházet vstupně-výstupní piny. Tato volba se provádí nastavením logické úrovně na pinu PUDC_B. Chování FPGA v závislosti na logické úrovni na tomto pinu je shrnuto v tabulce 2.4. Během inicializace se tato úroveň nesmí měnit. Pro vyvíjenou aplikaci jsem zvolil použít interní pull-up rezistory, abych měl definovanou úroveň na vstupně-výstupních pinech.

Tab. 2.4: Chování pinu PUDC_B [12].

Logická úroveň	Funkce
0	interní pull-up
1	vysoká impedance

K ovládní konfiguračního rozhraní slouží pin PROGRAM_B, díky němuž je možné vynutit inicializace FPGA. Pin DONE informuje o dokončení inicializace a pin INIT_B signalizuje chybu vzniklou při pokusu o inicializaci. Tyto signály jsem propojil s řídicím MCU, aby mohl proces inicializace FPGA kontrolovat.

2.5.4 Blokování napájení XC7S15

Sedmá generace hradlových polí od firmy Xilinx, do které spadá právě vybraný čip XC7S15, má výrobcem předepsané požadavky na blokování napájecích napětí [13]. Pro korektní funkci čipu je vhodné se touto specifikací řídit, i když výrobce uvádí, že čipy mohou fungovat i s jinými blokovacími kondenzátory, než uvádí. Seznam doporučených blokovacích kondenzátorů je uveden v tabulce 2.5.

Tab. 2.5: Doporučené blokovací kondenzátory pro XC7S15 v pouzdře FTBG196 [13].

V _{CCINT}			V _{CCBRAM}			V _{CCAUX}			V _{CCO}		
100 μF	4,7 μF	470 nF	47 μF	4,7 μF	470 nF	47 μF	4,7 μF	470 nF	47 μF	4,7 μF	470 nF
1	2	2	1	1	1	1	1	2	1	2	4

Z důvodu snížení počtu druhů součástek byl doporučený kondenzátor 100 μF nahrazen paralelní kombinací kondenzátorů 47 μF. Dále výrobce doporučuje použití

keramických kondenzátorů s nízkou hodnotou ekvivalentního sériového odporu, což bylo dodrženo.

2.6 Mikrokontrolér

Mikrokontrolér (MCU) je integrovaný obvod obsahující integrovaný procesor a periferie v jednom čipu. Mikrokontrolér vykonává program, proto je vhodný na provádění nejrůznějších algoritmů, avšak je to již hotový elektronický obvod, který umožňuje maximálně konfigurovat své periferie a měnit program. To jej předurčuje převážně k vykonávání časově méně kritických operací než například zpracování velkých datových toků, v nichž excelují převážně hradlová pole a zákaznické obvody.

MCU bude mít za úkol prostřednictvím sběrnice SPI ovládat systém v hradlovém poli, zdroj hodinového signálu pro FPGA a ADC. Také bude zajišťovat inicializaci FPGA a také bude komunikovat s počítačem prostřednictvím UART / USB převodníku.

Ve vyvíjené aplikaci by MCU mohl být teoreticky nahrazen soft core procesorem v hradlovém poli. Vzhledem k tomu, že zadavatel práce požadoval oddělení MCU od FPGA, aby bylo možné modifikovat firmware bez znalosti FPGA, jsem musel použít MCU.

2.6.1 Požadované parametry

Přáním zadavatele práce bylo, aby MCU byl značně naddimenzován z důvodu budoucích modifikací a také, aby mikrokontrolér byl z rodiny STM32, z důvodu, že tyto mikrokontroléry jsou hojně používány i v dalších projektech zadavatele.

Mezi technické požadavky na MCU patřila podpora SPI pro komunikaci s FPGA, UART včetně hardwarové signalizace pro komunikaci s počítačem prostřednictvím UART / USB převodníku.

2.6.2 Mikrokontrolér STM32H750VBT6

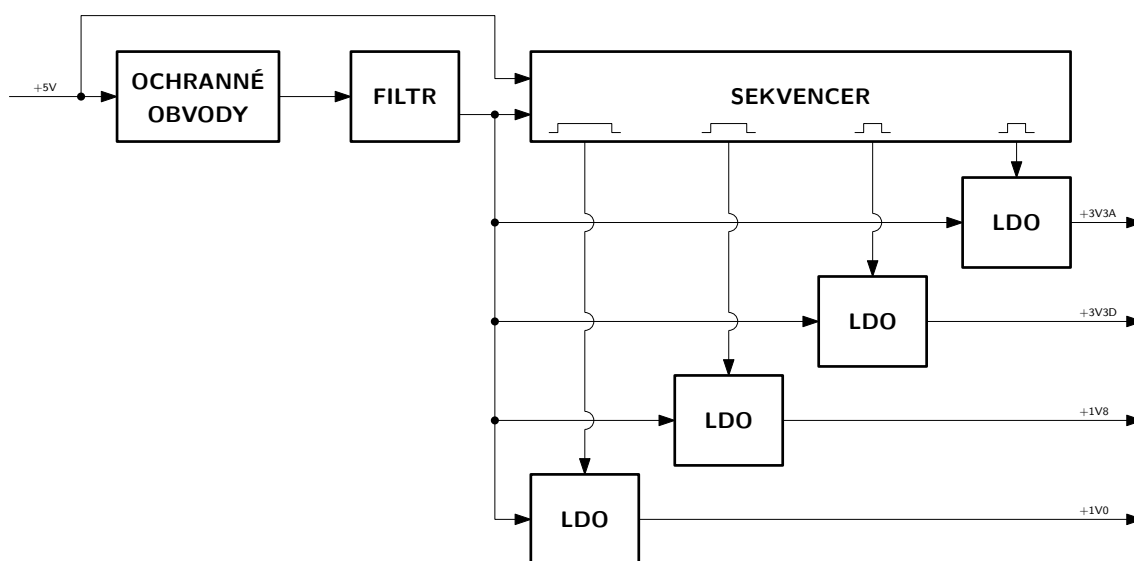
Vybral jsem mikrokontrolér STM32H750VBT6, který splňuje všechny požadované parametry. Je založen na ARM jádru Cortex-M7. Jde tedy o 32b mikrokontrolér s redukovanou instrukční sadou. Je možné jej taktovat až na frekvenci 480 MHz. Disponuje flash pamětí s kapacitou 128 kiB a pamětí SRAM o velikosti 1 MiB.

Pro programování MCU je na desce konektor rozhraní SWD, rozšířeném o pin SWO. Díky tomu lze snadno posílat ladící informace.

2.7 Napájecí systém

Napájecí systém má za úkol dodávat elektrickou energii komponentům navrhovaného zařízení v požadované kvalitě. Navržený systém se skládá ze tří typů bloků, které budou dále rozebrány.

Blokové schéma navrženého systému je na obrázku 2.7. Celé zařízení je napájeno napětím 5 V. Na vstupu napájení jsou ochranné prvky proti přepolarizování a přepětí. Následuje filtr, který má za úkol odfiltrovat případné rušení z napájecího zdroje. Požadovaná napětí jsou regulována pomocí nízkoubytkových lineárních stabilizátorů (LDO). Napájecí systém je řízený sekvencerem, který na základě hodnoty vstupního napětí ovládá jednotlivé stabilizátory.



Obr. 2.7: Blokové schéma napájecího systému.

2.7.1 Požadavky na stabilizátory napětí

V zařízení je požadováno několik různých napěťových větví, a to 3,3 V pro analogové obvody a 3,3 V pro obvody digitální. Hradlové pole navíc vyžaduje další dvě interní napětí, a to 1,8 V a 1,0 V. Pro odlišení analogového napětí od digitálního budou dále označovat analogové napětí jako +3V3A a napětí digitální jako +3V3D.

V tabulce 2.6 jsou uvedeny maximální hodnoty proudů, při plném vytížení všech obvodů. Výsledný odebíraný proud bude nižší, avšak při výběru LDO je vhodné uvažovat právě tyto proudy, ať je LDO vybráno s dostatečnou výkonovou rezervou.

Dalším parametrem, který musí LDO splňovat, je schopnost pracovat s velkou zatěžovací kapacitou, danou značným počtem blokových kondenzátorů pro FPGA. LDO musí umožnit nastavovat výstupní napětí, aby stačil jeden typ stabilizátoru.

Tab. 2.6: Přehled teoretických maximálních hodnot proudů v jednotlivých větvích.

Větev	I [mA]
+3V3A	80
+3V3D	777
+1V8	50
+1V0	217

Také musí disponovat pinem enable, aby bylo možné tento obvod použít se sekven-
cerem napájecího napětí.

2.7.2 Stabilizátor NCP59744

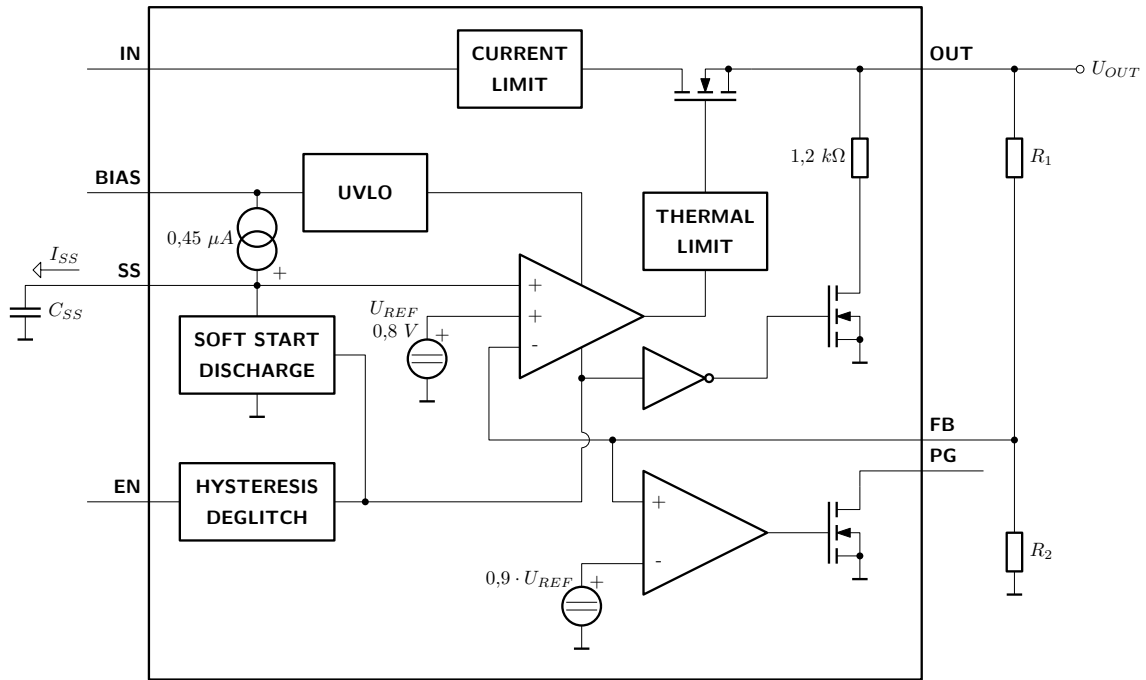
Po porovnání dostupných LDO na trhu, jsem převážně díky ceně vybral integrovaný obvod firmy ON Semiconductor NCP59744. Tento obvod dovoluje nastavovat vý-
stupní napětí v rozsahu od 0,8 V do 3,6 V a poskytuje maximální výstupní proud 3 A. Maximálním vstupním napětím stabilizátoru je 5,5 V. Minimální úbytek na stabilizátoru při maximálním zatížení činí 95 mV. Disponuje také funkcí programo-
vatelného soft start (postupný náběh zdroje). Navíc je tento obvod přímo určený k napájení obvodů FPGA. Stabilizátor má integrovanou referenci o jmenovitém na-
pětí 0,8 V, od jejíž hodnoty se odvíjí většina výpočtů souvisejících s tímto stabili-
zátorem.

Vzhledem k tomu, že stabilizátor umožňuje stabilizovat nízké napětí s relativně
malým úbytkem, je vybaven dvěma napěťovými vstupy. Pin IN slouží k přívodu
napětí, které bude stabilizováno, a pin BIAS k přivedení napájecího napětí pro
stabilizátor. Aby obvod korektně pracoval, je nutné, aby na pin BIAS bylo přivedeno
napětí minimálně 2,2 V.

Funkce stabilizátoru je rozebrána na blokovém schématu 2.8. Nastavení výstup-
ního napětí se provádí nastavením dělicího poměru děliče složeného z rezistorů R_1 a
 R_2 . LDO sleduje úbytek napětí na R_2 a pomocí regulace výstupního napětí se snaží
docílit, aby odpovídal hodnotě referenčního napětí. Toto chování popisuje rovnice
2.7.

$$U_{OUT} = U_{REF} \frac{U_{R_2}}{R_2} (R_1 + R_2) = U_{REF} \left(1 + \frac{R_1}{R_2} \right) = 0,8 \text{ V} \left(1 + \frac{R_1}{R_2} \right) \quad (2.7)$$

V technické dokumentaci k NCP59744 je uveden seznam předpočítaných re-
zistorů pro často používaná výstupní napětí [6]. Doporučené hodnoty jsme se rozhodli
využít, jelikož obsahovaly i mnou požadovaná výstupní napětí.



Obr. 2.8: Blokové schéma obvodu NCP59744.

Soft start funkce je založena na nabíjení kondenzátoru proudovým zdrojem C_{SS} , čímž na něm roste napětí. Tento proces popisuje rovnice 2.8.

$$u_{C_{SS}}(t) = \frac{1}{C_{SS}} \int i_{C_{SS}}(t) dt = \frac{I_{SS}}{C_{SS}} \int dt \quad (2.8)$$

S rostoucím napětím na C_{SS} se zvyšuje i výstupní napětí stabilizátoru. Až napětí na C_{SS} dosáhne hodnoty napěťové reference, výstupní napětí bude rovno hodnotě nastavené děličem.

$$u_{C_{SS}}(t_{SS}) = U_{REF} = \frac{I_{SS}}{C_{SS}} \int_0^{t_{SS}} dt = \frac{I_{SS}}{C_{SS}} t_{SS} \Rightarrow t_{SS} = \frac{U_{REF}}{I_{SS}} C_{SS}$$

$$t_{SS} = \frac{0,8 \text{ V}}{0,45 \mu\text{A}} C_{SS} \quad (2.9)$$

Jako hodnotu C_{SS} jsem použil 1 nF, jelikož je v zapojení již využita a náběhový čas zdroje s ní činí 1,8 ms, což splňuje požadavky FPGA [14].

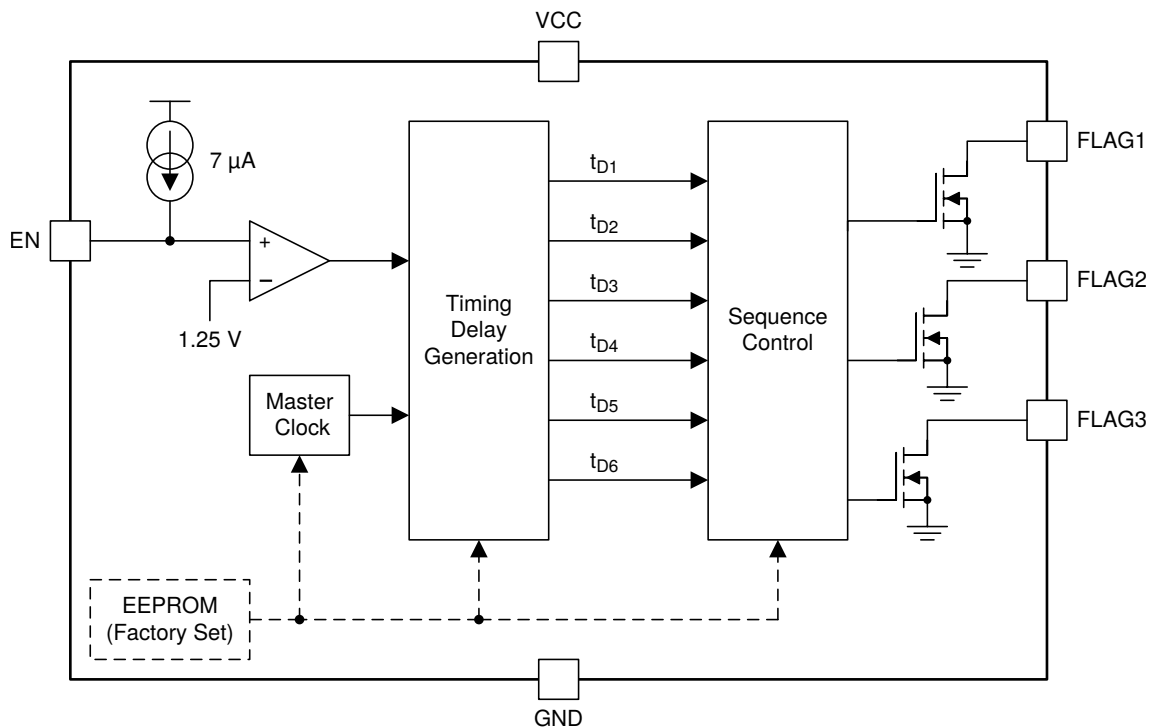
2.7.3 Řízení napájecího systému

V technické dokumentaci k FPGA [14] se doporučuje postupně spínat jednotlivé napájecí napětí po zapnutí napájení a při vypnutí provést vypínání v opačném pořadí. Doporučené pořadí spínání je následující: V_{CCINT} , V_{CCBRAM} , V_{CCAUX} a nakonec V_{CCO} . Ve vyvíjeném zařízení používají větve V_{CCINT} a V_{CCBRAM} společné napájení

označené jako +1V0. Napětí V_{CCAUX} je napájeno větví +1V8 a napětí V_{CC0} větví +3V3D, kterou jsou napájeny i ostatní digitální obvody. V zařízení je ještě jedna napájecí větev označovaná jako +3V3A pro napájení analogových obvodů, která je řízena stejným signálem jako větev +3V3D.

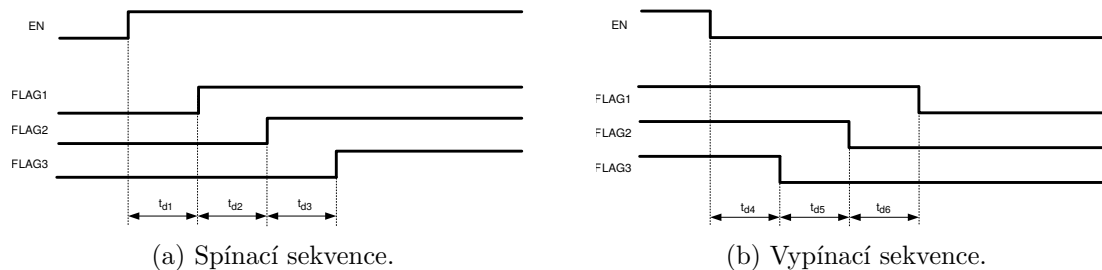
2.7.4 Integrovaný sekvencer napájení LM3880

Na základě výše uvedených požadavků na řízení napájení jsem vybral integrovaný obvod LM3880, který je určený přímo k sekvencování napájecího napětí FPGA obvodů. Pracuje od napětí 2,7 V do 5,5 V. Obvod generuje tři řídicí signály označené FLAG1–FLAG3. Jak je vidět na blokovém schématu 2.9, tyto signály jsou typu otevřený kolektor, proto je třeba pro definování kladné úrovně obvod rozšířit o pull-up rezistory. Sekvence generované tímto obvodem jsou naprogramovány do EEPROM již při výrobě a není je možné uživatelsky přepsat. Obvod se však vyrábí v mnoha konfiguracích, takže je možné si vybrat požadovanou sekvenci.



Obr. 2.9: Blokové schéma obvodu LM3880, převzato z [10].

Pokud napětí na pinu EN překročí referenční napětí 1,27 V, dojde k vygenerování spínací sekvence. Aby bylo možné vygenerovat i vypínací sekvenci, je třeba zajistit, aby po odpojení napájení měl obvod dostatek energie ke korektnímu vypnutí. To zajišťují ochranné obvody a filtr. Pokud dojde ke změně napětí na pinu EN v průběhu vykonávání sekvence, je nejprve prováděná sekvence dokončena a až poté se na



Obr. 2.10: Časové průběhy sekvencí LM3880MF-1xx, převzato z [10].

změnu zareaguje. Generované sekvence jsou zobrazeny na obrázku 2.10. Všechny zakótované časy pro vybranou variantu obvodu LM3880MF-1AA jsou dlouhé 10 ms.

Na blokovém schématu 2.7 můžeme pozorovat, že sekvencer snímá napájecí napětí na vstupu před ochrannými obvody. Díky tomu je možné včas reagovat na jeho změnu. Napájení je snímáno pomocí děliče napětí, který umožňuje nastavit prahovou úroveň, na kterou bude zařízení reagovat.

2.7.5 Ochranné prvky

Mezi ochranné prvky použité v napájecím systému patří ochrana proti přepolarizování napájecího napětí a ochrana proti přepětí.

Ochrana proti přepolarizování je realizována sériovým vložením křemíkové diody do hlavní napájecí větve. Na této diodě navíc dochází k úbytku napětí, díky čemuž se snižuje zátěž stabilizátorů. Navíc tato dioda umožňuje detekovat změny napájecího napětí předtím, než se tato změna projeví změnou napětí na výstupu filtru.

Ochrana proti přepětí je realizována výkonou TVS diodou. Tato dioda, díky své voltampérové charakteristice omezuje vstupní napětí pro LDO na 5,2 V, čímž chrání systém proti krátkodobému přepětí.

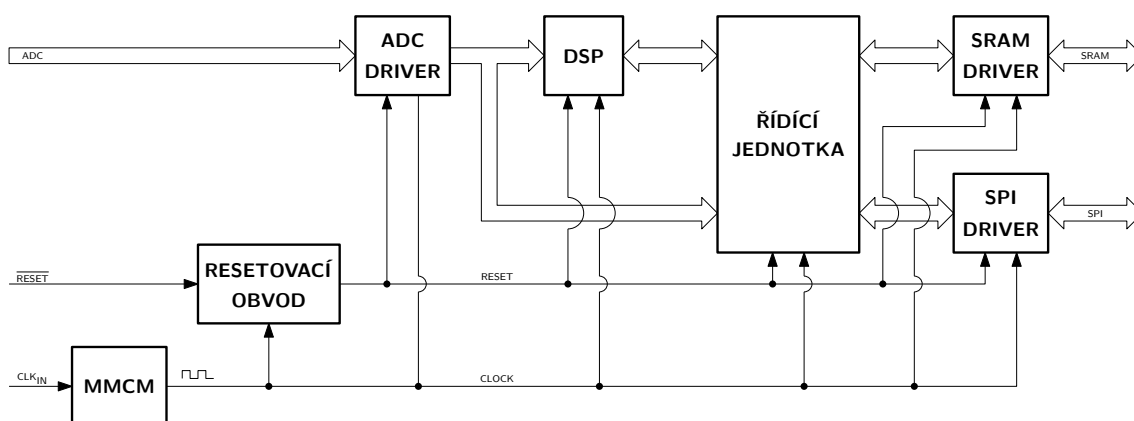
2.7.6 Odrušovací filtr

K potlačení případného šumu ze zdroje napájení je zařízení vybaveno LC filtrem. Ten je navržen jako dolní propust, realizovaná dvojitým Π článkem. Jelikož se reálné kondenzátory od určité frekvence začínají chovat jako cívky, u kterých dochází k opačnému jevu, způsobeného parazitními vlastnostmi součástek, jsou kondenzátory složeny z kondenzátorů různých hodnot, čímž je zajištěna korektní funkce v širším frekvenčním rozsahu. Z téhož důvodu jsou v obvodu použity dvě odlišné cívky. Pro pokrytí nižších frekvencí je využita drátová tlumivka a pro frekvence vyšší je využita feritová perlička.

Kromě potlačení rušení ze zdroje má filtr ještě jeden účel, jímž je akumulace energie. Díky tomu po odpojení napájecího napájení může být zařízení korektně vypnuto.

3 Systém v hradlovém poli

Tato kapitola se věnuje návrhu číslicového systému v hradlovém poli. Systém je popisován jazykem VHDL, který je určený převážně pro popis a simulaci číslicových systémů. Cílem vyvíjeného systému je detekovat pulzy v datech z analogově-digitálního převodníku a u těchto detekovaných pulzů změřit jejich amplitudu. Z informací o amplitudách poté systém v reálném čase vytváří histogram.



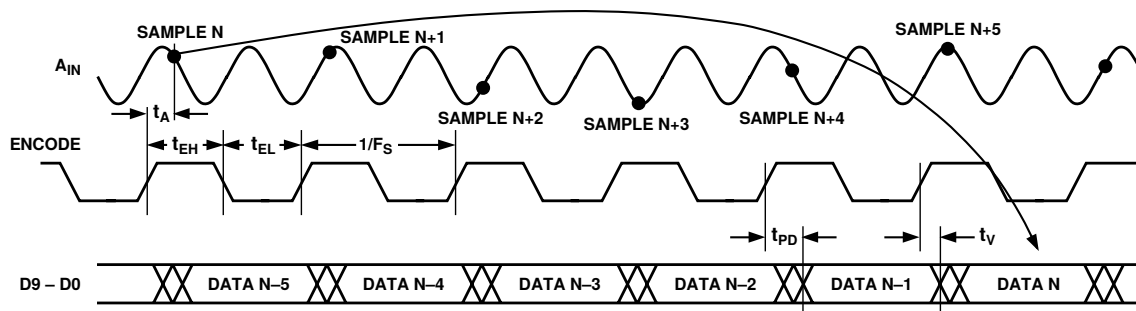
Obr. 3.1: Blokové schéma systému v FPGA.

Vzorkovaný signál s ADC je zapisován do záchytných registrů ADC driveru. Přesné časování této operace zajišťuje jednotka MMCM, která generuje systémové hodiny, jenž jsou fázově posunuty oproti vstupnímu hodinovému signálu. Jednotka MMCM je IP blok vyvinutý firmou Xilinx, z toho důvodu zde není více rozebrán. Signál z ADC driveru je analyzován v bloku DSP, který provádí analýzu signálu. Řídící jednotka konfiguruje blok DSP a ze získaných dat vytváří histogram, také ovládá celý systém v FGPA. SRAM driver slouží k ovládání paměti SRAM, ve které je uložen histogram. SPI driver zajišťuje komunikaci s mikrokontrolérem.

3.1 ADC driver

Časování zvoleného převodníku AD9214 je zobrazené na obrázku 3.2. Systémový oscilátor, který je zdrojem hodinového signálu jak pro ADC, tak i pro FPGA generuje signál s frekvencí 100 MHz, tedy jedna perioda odpovídá 10 ns.

K zajištění korektního časování pro vyčítání hodnot s ADC jsem musel posunout fázi hodinového signálu uvnitř FPGA. K tomuto účelu jsem použil MMCM jednotku. Nastavení korektní hodnoty jsem provedl experimentálně. Na vstup ADC byl přiveden harmonický signál a pomocí integrovaného logického analyzátoru (ILA) v FPGA jsem vyhodnocoval, jestli je vyčítání správně načasováno. Špatné načasování se projevovalo špičkami v harmonickém signálu.



Obr. 3.2: Časování ADC AD9214, převzato z [1].

Tab. 3.1: Časování AD9214 hodnoty jsou brány jako nejhorší případ, převzato z [1].

Označení	Hodnota	Popis
t_{PD}	6 ns	Output Propagation Delay
t_V	3 ns	Output Valid Time

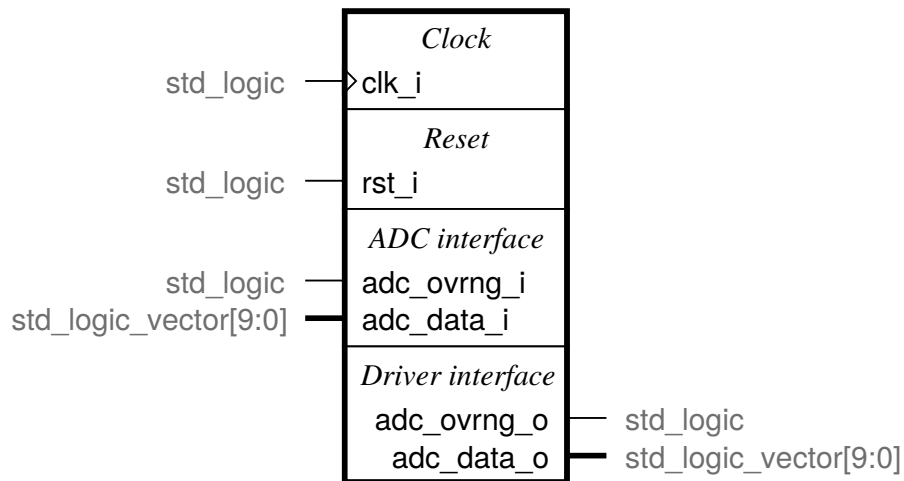
ADC driver na náběžnou hranu hodinového signálu do 10b registru ukládá vzorky dat z ADC přivedené na port `adc_data_i` pro další zpracování. Zároveň na tutéž hranu hodinového signálu do 1b registru ukládá hodnotu ADC pinu OR, která je přivedena k portu `adc_ovrng_i`, jenž nese informaci o překročení rozsahu ADC. Výstupy registrů jsou vyvedeny na výstupní porty.

3.2 SRAM driver

SRAM driver se stará o zápis dat do operační paměti a jejich čtení z ní. Použitá paměť je statická, asynchronní a paralelní. Tomu odpovídá i způsob, jakým se paměť ovládá.

Datová sběrnice je obousměrná, nechtěl jsem riskovat poškození hardwaru chybnou implementací, a proto jsem si obstaral od výrobce simulační model paměti. Napřed jsem celý blok odsimuloval a až po jeho odladění jej otestoval na fyzickém hardwaru.

Signály \overline{CE} , \overline{LB} a \overline{UB} jsou stále aktivované, díky čemuž je paměť neustále v pohotovostním režimu. Vzhledem k tomu, že je paměť asynchronní a nemá žádný výstup informující o validnosti výstupních dat, případně dokončení zápisu, rozhodl jsem se rozdělit všechny operace s pamětí na dva taktů hodinového signálu. Tím jsem docílil korektního časování vyčítání. V případě čtení v prvním taktu nastavím adresu, ze které chci číst, a aktivuji signál \overline{OE} . Ve druhém taktu si již přečtu stabilní data. V případě zápisu je postup obdobný. V prvním taktu nejprve nastavím adresu a



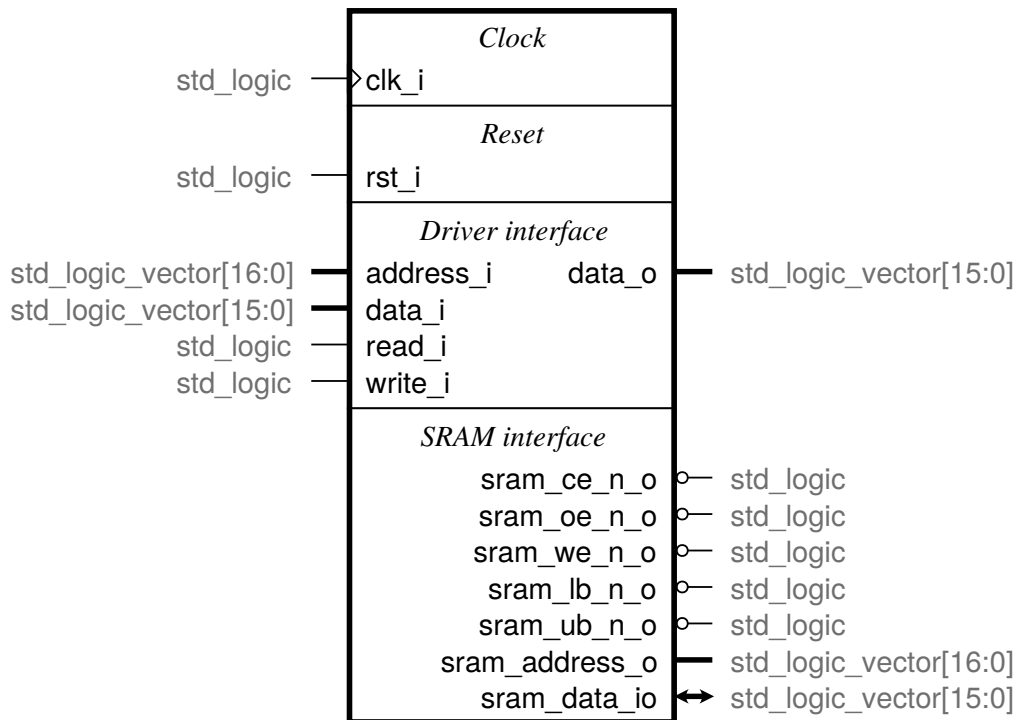
Obr. 3.3: Entita navrženého ADC driveru.

Tab. 3.2: Vstupně-výstupní rozhraní IS61WV12816EDBL [5].

Pin	Funkce
A0-A16	Adresní sběrnice.
IO0-IO15	Obousměrná datová sběrnice.
$\overline{\text{CE}}$	Aktivace čipu.
$\overline{\text{OE}}$	Povolení výstupu.
$\overline{\text{WE}}$	Povolení zápisu.
$\overline{\text{LB}}$	Ovládání LSB (IO0-IO7).
$\overline{\text{UB}}$	Ovládání MSB (IO8-IO15).

požadovaná data, v taktu druhém aktivuji signál $\overline{\text{WE}}$ a provedu zápis dat. Vzhledem k tomuto rozdělení operací s pamětí do dvou taktů dosahují přenosové rychlosti mezi FPGA a SRAM 800 Mbps.

Ovládání SRAM driveru je realizováno pomocí ovládacího rozhraní. Adresa, jejíž obsah má být přepsán či přečten, je nastavena na port `address_i`. Data určená k zápisu jsou přivedena na port `data_i`. Poté je aktivováním signálu `write_i` proveden zápis. Pro přečtení obsahu paměti je třeba aktivovat signál `read_i`, který způsobí, že obsah paměti z požadované vstupní adresy je zapsán na výstupním portu `data_o`. Pokud budou aktivovány signály `write_i` a `read_i` současně, bude upřednostněna operace zápisu.



Obr. 3.4: Entita navrženého SRAM driveru.

3.3 SPI driver

Tato podkapitola se věnuje popisu sběrnice SPI a navrženého rozhraní slave zařízení.

3.3.1 Popis sběrnice SPI

SPI je sériová sběrnice, která umožňuje plně duplexní synchronní komunikaci, kterou řídí jedno zařízení označované jako master, ostatní zařízení na sběrnici jsou označována jako slave. Kompletní implementace SPI komunikace využívá čtyř signálních vodičů, jež jsou popsány v tabulce 3.3. Vodičů pro výběr slave zařízení může být použito více v závislosti na jejich počtu.

Tab. 3.3: Přehled signálů SPI sběrnice.

Signál	Funkce
MOSI	Datový signál generovaný master zařízením.
MISO	Datový signál generovaný slave zařízením.
SCK	Synchronizační hodinový signál.
NSS	Výběr slave zařízení aktivní v logické nule.

Specifikace SPI sběrnice využívá čtyři komunikační módy, které rozlišuje podle konfigurace hodinového signálu, u něhož se nastavuje klidová úroveň a fáze. Nastavení klidové úrovně se označuje jako CPOL (Clock Polarity), pro značení fáze je zavedena zkratka CPHA (Clock Phase). Módy jsou shrnuty v tabulce 3.4.

$$\text{Klidová logická úroveň hodinového signálu je } \begin{cases} 0, & \text{pokud CPOL} = 0 \\ 1, & \text{pokud CPOL} = 1 \end{cases}$$

$$\text{Data jsou vyčítána na } \begin{cases} \text{liché hraně (1, 3, 5, \dots),} & \text{pokud CPHA} = 0 \\ \text{sudé hraně (2, 4, 6, \dots),} & \text{pokud CPHA} = 1 \end{cases}$$

Tab. 3.4: Přehled módů SPI sběrnice.

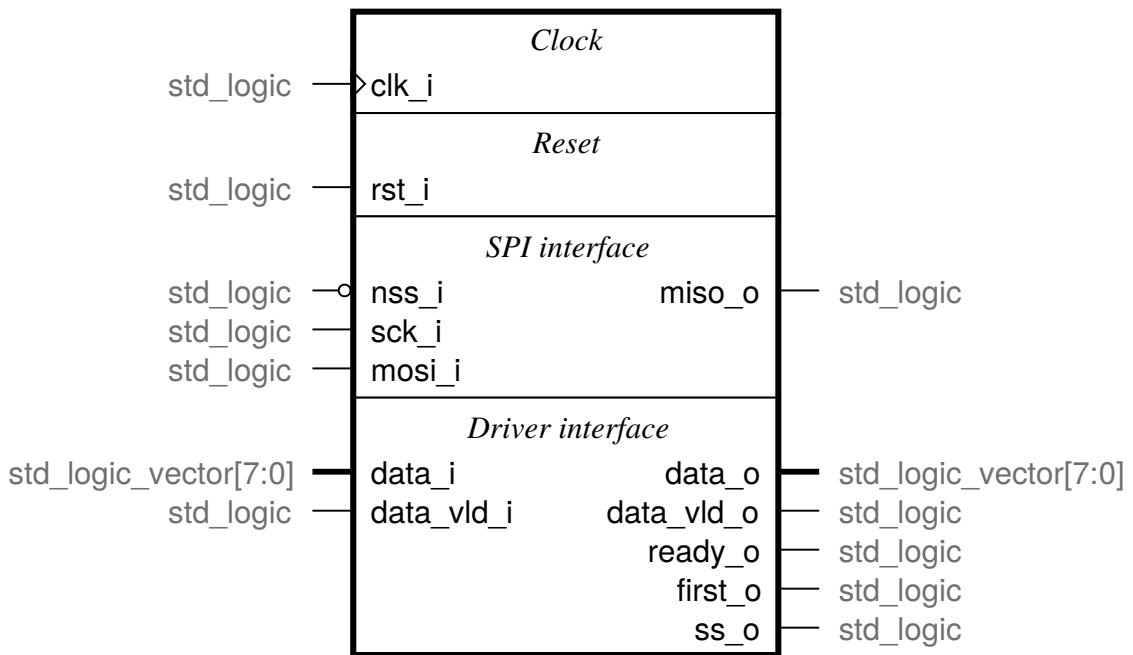
Mód	CPOL	CPHA	Popis
0	0	0	Klidová úroveň 0, data jsou platná na náběžné hraně.
1	0	1	Klidová úroveň 0, data jsou platná na sestupné hraně.
2	1	0	Klidová úroveň 1, data jsou platná na sestupné hraně.
3	1	1	Klidová úroveň 1, data jsou platná na náběžné hraně.

3.3.2 Realizace SPI slave driveru

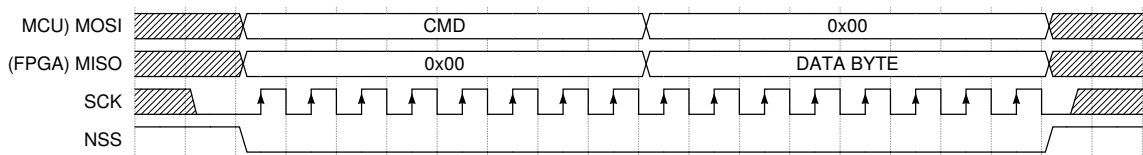
SPI se v této aplikaci využívá pro komunikaci mezi FPGA a MCU, kde MCU figuruje jako master, tedy v FPGA jsem implementoval slave. Moje implementace SPI pracuje v módu 0.

Vyvinutý SPI driver se ovládá pomocí skupiny signálů Driver interface. Pokud je jednotka aktivovaná pomocí signálu `nss_i`, přijímá vysílaná data. Příjem dat je realizován jako posuvný registr, proto lze pozorovat, jak se při příjmu na portu `data_o` postupně posouvají přijímané bity. Pokud dojde ke korektnímu příjmu celého bajtu, aktivuje se na jeden takt hodinového signálu signál `data_vld_o`, který informuje o tom, že na portu `data_o` jsou dostupná korektní data. Pokud je zároveň přijatý bajt prvním bajtem rámce, aktivuje se i signál `first_o`. Jako rámec se rozumí přenos dat, během kterého je stále aktivní signál `nss_i`.

Vysílání je založeno na podobném principu jako příjem. Je rovněž realizováno posuvným registrem. Pokud je výstupní signál `ready_o` aktivní, signalizuje, že vysílací buffer je prázdný. Pokud je v tento okamžik signálem `data_vld_i` nastaven příznak validnosti vstupních dat, jsou data přítomná na portu `data_i` zapsána do vysílacího bufferu, ze kterého jsou následně přesunuta do vysílacího posuvného registru.



Obr. 3.5: Entita navrženého SPI driveru.



Obr. 3.6: Ukázka jednoho komunikačního rámce SPI.

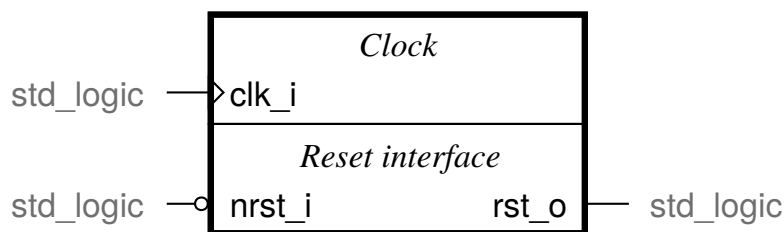
Celý SPI driver je synchronní obvod reagující na náběžnou hranu hodinového signálu. Pro její korektní činnost po inicializaci FPGA je nutné jej resetovat. K tomuto účelu je vybaven vstupem pro synchronní reset.

Vstupní signály rozhraní SPI je nutné synchronizovat se systémovými hodinami, aby se zamezilo vzniku metastabilit. To je zajištěno uvnitř této jednotky vzorkováním vstupních signálů SPI klopnými obvody typu D aktivními na náběžnou hranu systémového hodinového signálu.

Výstupní signál `ss_o` informuje o tom, že slave je aktivován signálem `nss_i`.

3.4 Resetovací obvod

Aby bylo po inicializaci FPGA v definovaném stavu, je třeba nastavit registry na výchozí požadované hodnoty. Tento úkon by sice bylo možné provést v definicích výchozích hodnot signálů, ale mým záměrem bylo, aby bylo možné tento stav způsobit i jiným způsobem než reinitializací FPGA.



Obr. 3.7: Entita navrženého resetovacího obvodu.

Z toho důvodu jsem se rozhodl vytvořit resetovací obvod. Tento obvod resetuje systém v hradlovém poli po inicializaci a také pokud je detekován na pinu `nrst_i` signál ke kompletnímu resetu zařízení. Výstupem resetovacího obvodu je signál `rst_o`, což je synchronní reset aktivní v logické úrovni jedna.

3.5 Zpracování signálu

Cílem zpracování signálu je detekce pulzů a následné měření jejich amplitud. Pro tento účel jsem navrhl jednotku DSP (Digital Signal Processing), která se skládá z několika bloků, které budou v této podkapitole popsány.

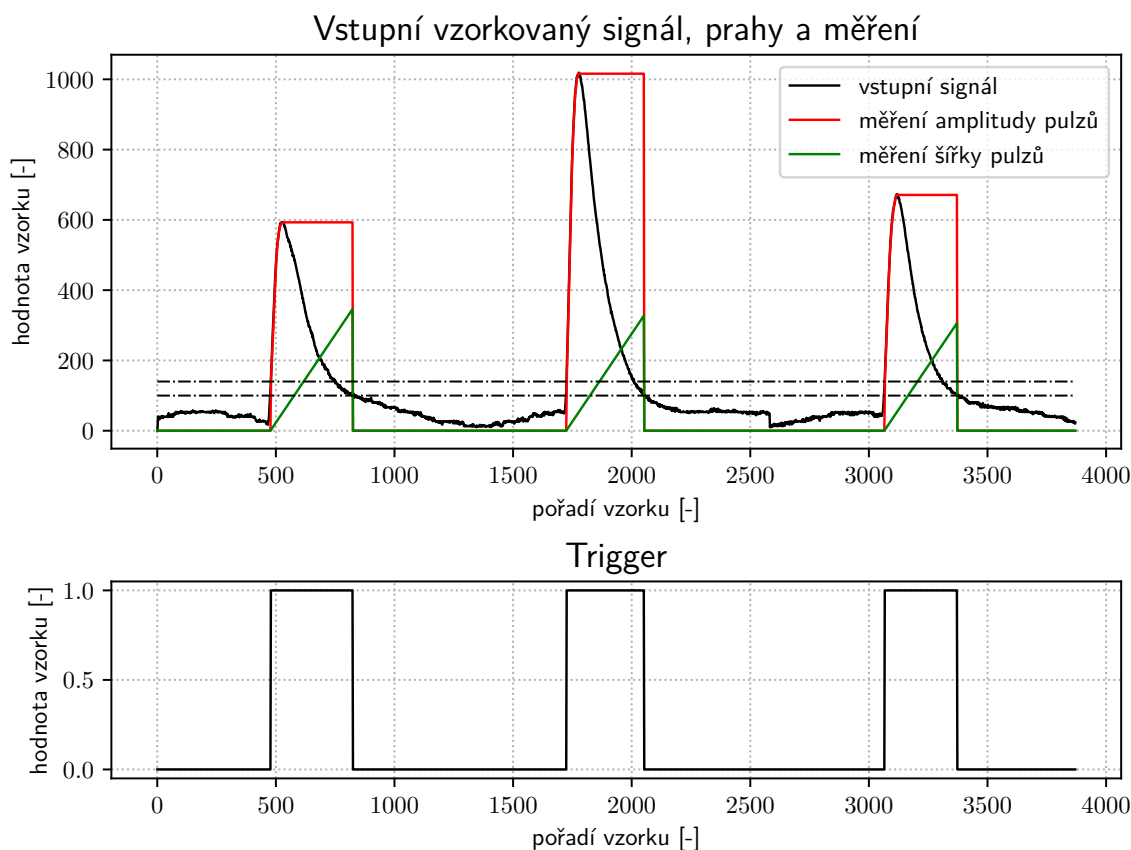
3.5.1 Detekce pulzů

K detekci pulzů využívám digitální komparátor s hysterezí. Výstup z detektoru se poté využívá jako trigger pro měření délek pulzů a amplitud. Výhodou této metody je značná odolnost proti šumu v signálu. Nevýhodou je, že pulzy pod komparačními úrovněmi nebudou detekovány. Testoval jsem i využití derivačního filtru, ale v porovnání s výše zmíněnou metodou byl mnohem méně spolehlivý.

Na obrázku 3.8 je zobrazeno testování detekce pomocí skriptu napsaného v jazyce Python. Signál je vytvořen složením tří samostatně změřených pulzů pomocí digitálního osciloskopu, následně upraveného dolní propustí a decimátorem tak, aby odpovídal signálu vzorkovanému 80 MSPS. Komparační úrovně jsou znázorněny čerchovanou čarou.

3.5.2 Měření amplitudy pulzů

Velikost amplitudy je určena jako aritmetický průměr dvou vzorků s nejvyšší hodnotou. Tato metoda se dobře implementuje v hradlovém poli, protože lze realizovat jako bitový posun součtu vzorků. Navíc by měla dosahovat přesnějšího výsledku než



Obr. 3.8: Ukázka měření pulzů nad daty ze scintilačního detektoru.

pouhého měření maxima, protože pomocí ní se alespoň částečně mohou eliminovat chyby měření. Měření probíhá, dokud je aktivní trigger signál, poté je výsledek zapsán na výstupní port a pracovní registry jsou nastaveny na výchozí hodnoty.

3.5.3 Měření šířky pulzů

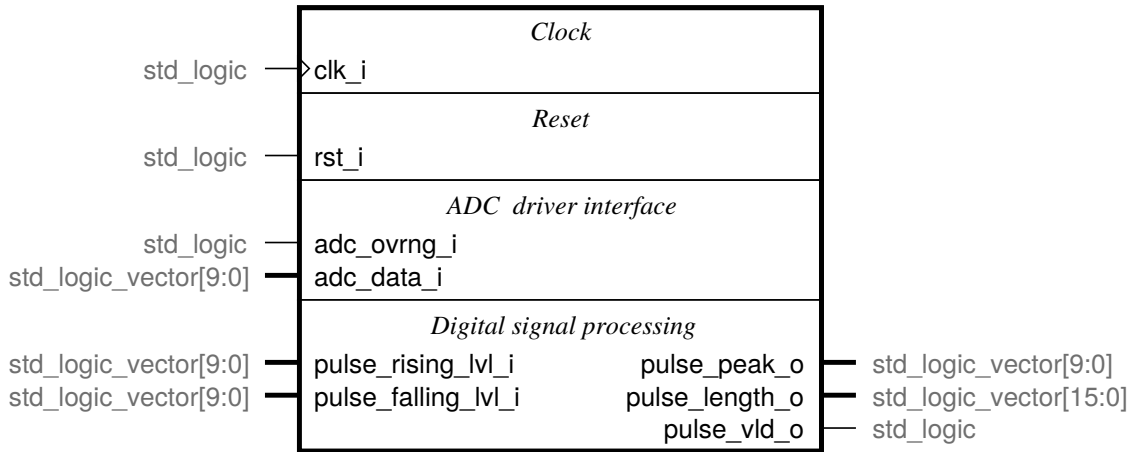
Pro určení šířky pulzu je využíván trigger signál generovaný detekcí pulzů. Vzhledem k tomu, že znám frekvenci systémového oscilátoru, tak mohou měření šířky pulzů realizovat jako čítač a počet pulzů poté přepočítat na čas. Čítání probíhá, dokud je aktivní trigger, poté je výsledek zapsán do výstupního portu a čítač je vynulován.

Pokud by bylo nutné zvětšit časové rozlišení, stačí pomocí jednotky MMCM generovat rychlejší hodinový signál pro čítač.

3.5.4 Realizace DSP jednotky

Entita vytvořené DSP jednotky je znázorněna na obrázku 3.9. Přivádí se do ní vzorky signálu z ADC driveru. Dalšími vstupními signály jsou `pulse_rising_lvl_i`

a `pulse_falling_lvl_i`, kterými se nastavují prahové úrovně komparátoru, který generuje trigger signál. Na port `pulse_peak_o` se zapisuje změřená amplituda pulzu. Informace o délce pulzu je dostupná na portu `pulse_length_o`. Signál `pulse_vld_o` informuje o validnosti změřených dat.



Obr. 3.9: Entita navržené DSP jednotky.

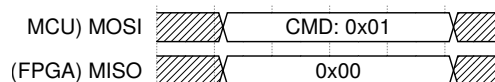
3.6 Řídící jednotka

Systém v hradlovém poli se skládá z několika bloků, které je třeba ovládat a předávat mezi nimi data, a právě k tomuto účelu jsem vyvinul řídicí jednotku. Ta reaguje na příkazy z mikrokontroléru, který je pomocí SPI propojený s FPGA, a podle nich řídí a konfiguruje ostatní části systému.

3.6.1 SPI příkazy

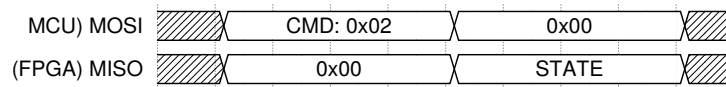
Tato podkapitola popisuje příkazy používané k ovládání systému v FPGA z řídicího MCU prostřednictvím sběrnice SPI.

STOP Přeruší právě vykonávaný příkaz a uvede zařízení do režimu čekání na příkaz nový. Může se použít například k pozastavení měření.



Obr. 3.10: SPI příkaz STOP.

GET STATE Vrátil stav zařízení zakódovaný do jednoho bajtu a přitom neovlivní právě vykonávaný příkaz.



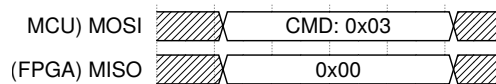
Obr. 3.11: SPI příkaz GET STATE.

Jednotlivé stavy jsou popsány v následující tabulce.

Tab. 3.5: Stavy zařízení.

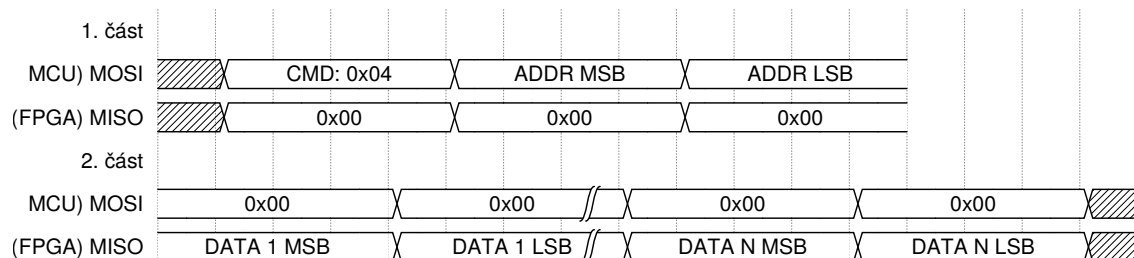
Kód stavu	Stav
0x00	Zařízení je připraveno.
0x01	Čtení paměti.
0x02	Mazání paměti.
0x03	Probíhající měření.

MEMORY ERASE Zapiše na všechny adresy paměti SRAM nuly. Využívá se zejména k vynulování kanálových čítačů před zahájením měření.



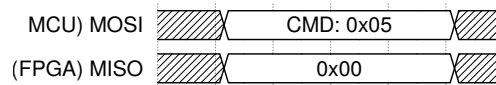
Obr. 3.12: SPI příkaz MEMORY ERASE.

MEMORY READ Příkaz zahájí čtení obsahu paměti SRAM. Jeho parametry jsou dva bajty, ve kterých je zakódovaná počáteční adresa čtení. Data jsou čtena tak dlouho, dokud master generuje signál SCK nebo není ukončen rámeček.



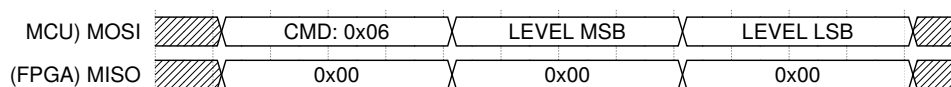
Obr. 3.13: SPI příkaz MEMORY READ.

MEASUREMENT START Příkaz zahájí měření, tedy začne z dat z DSP jednotky vytvářet histogram. Pokud by hrozilo přetečení histogramu, je měření považováno za dokončené a zařízení se vrátí do stavu, ve kterém očekává další příkaz.



Obr. 3.14: SPI příkaz MEASUREMENT START.

SET RISING LEVEL Nastaví práh pro náběžnou hranu pulzu. Tato hodnota se používá v jednotce DSP při detekci pulzů. Hodnota prahu je dána rovnicí 4.1.



Obr. 3.15: SPI příkaz SET RISING LEVEL.

SET FALLING LEVEL Nastaví práh pro sestupnou hranu pulzu. Tato hodnota se používá v jednotce DSP při detekci pulzů. Hodnota prahu je dána rovnicí 4.1.

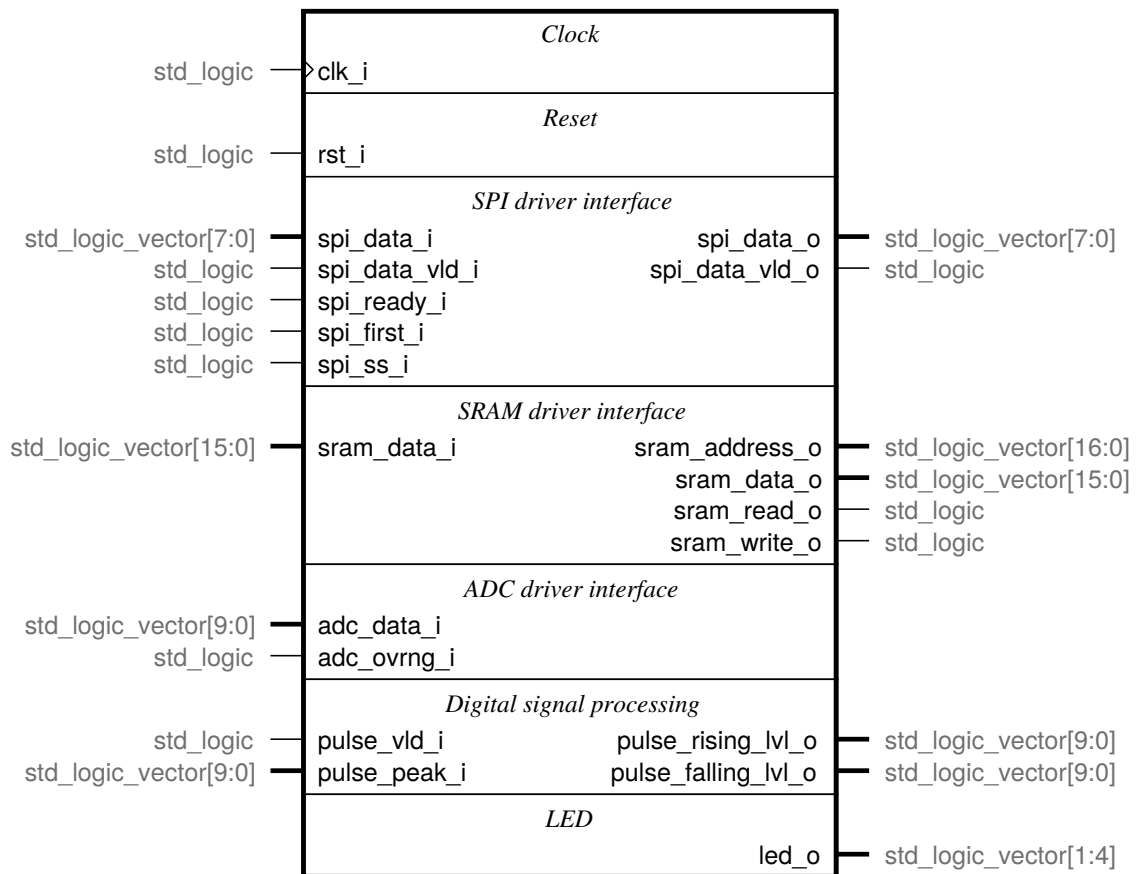


Obr. 3.16: SPI příkaz SET FALLING LEVEL.

3.6.2 Realizace řídicí jednotky

Realizace řídicí jednotky je založena na třech stavových automatech. První stavový automat zpracovává po bajtech přijímaná data z SPI driveru a třídí je dle přijatého pořadí do třech kategorií, na první, druhý a další. Druhý stavový automat na základě přijatých dat a jim přiděleného pořadí od prvního stavového automatu dekoduje přijatý příkaz a zkontroluje, zdali má správný počet argumentů. Pokud vyhodnotí příkaz jako platný, předá jej třetímu stavovému automatu, nebo pokud se jedná o příkazy konfigurace DSP jednotky, je příkaz rovnou vykonán.

Třetí stavový automat ovládá celé měření, spravuje stavový registr, generuje řídicí signály pro SPI driver, pro posílání odpovědí, vytváří histogramy z měřených hodnot, ovládá SRAM a ADC drivery. Po resetu čeká na korektní příkaz od druhého stavového automatu.



Obr. 3.17: Entita řídicí jednotky.

Zajímavostí na realizaci třetího stavového automatu je, že dva jeho stavy fungují obdobně jako podprogramy. Stavový automat si v jednom registru uchovává informaci o aktuálním stavu a ve druhém registru uchovává stav, který bude následovat po skoku do jednoho ze dvou speciálních stavů. Speciální stavy provádějí čtení a zápis do SRAM ovládaním SRAM driveru, díky čemuž je možné v různých stavech tohoto stavového automatu provést operaci s pamětí a určit, jaký stav bude po této operaci následovat, což velmi zjednodušuje výsledný stavový automat.

Řídicí jednotka také ovládá LED, které v raných fázích vývoje byly používány k ladění chyb, nyní je jejich funkce signalizovat obsluze zařízení překročení rozsahu ADC.

4 Firmware

Pod pojmem firmware (FW) se v této práci rozumí program vykonávaný MCU. Úkolem FW je přijímat a vyhodnocovat příkazy z PC a na jejich základě komunikovat s FPGA a posílat zpět do PC odpovědi na ně.

FW je vyvíjený pro MCU STM32H750 a využívá oficiálních knihoven HAL od společnosti STMicroelectronics. Programován je v jazyce C, dodržuje specifikaci C99. FW je kompilovatelný v ARM GCC a při vývoji byla využívána verze 9.3.1.

4.1 Popis firmware

Tato podkapitola se věnuje popisu firmware používaného pro ovládání vyvinuté elektroniky.

4.1.1 Inicializace

Poté co je inicializována knihovna HAL, dojde ke spuštění funkce inicializace zařízení. Ta nejprve aktivuje zdroj hodinového signálu pro FPGA, následuje inicializace SPI driveru a FTDI driveru. Poté je provedeno blokující čekání na dokončení inicializace FPGA. Následuje inicializace polí pro příjem a vysílání SPI. Nakonec jsou po SPI odeslány příkazy nastavující výchozí prahové úrovně pro DSP jednotku a je spuštěna hlavní programová smyčka.

4.1.2 Superloop

Hlavní programová nekonečná smyčka neboli superloop čeká na nastavení vlajky signalizace nového příkazu, které se provádí v obsluze přerušení od jednotky UART. Po detekci nového příkazu příslušnou vlajku deaktivuje a vykoná požadovaný příkaz. Následně se celý cyklus opakuje.

4.1.3 SPI driver

SPI driver je vyvinutá knihovna pro zapouzdření ovládání SPI. Pro její funkčnost je nezbytné ji inicializovat. Driver využívá nově definovaného datového typu pro uložení konfigurace SPI. Po inicializaci umožňuje používat funkce pro plně duplexní přenos dat s využitím DMA (Direct memory access), dále zpřístupňuje funkci zjišťující zaneprázdněnost jednotky SPI a funkci pro blokující čekání na dostupnost jednotky SPI.

4.1.4 SPI protokol

SPI protokol je knihovna, ve které jsou implementovány jednotlivé příkazy specifikované v 3.6.1. Každý příkaz je naprogramován jako samostatná funkce.

4.1.5 FTDI driver

FTDI driver je knihovna určená pro zapouzdření řízení přenosu dat mezi MCU a PC. MCU komunikuje s PC pomocí rozhraní UART, které je čipem FT231XS od firmy FTDI převedeno na rozhraní USB. Knihovnu je třeba před použitím inicializovat. Během inicializace je nutné nastavit, která jednotka UART bude použita a která funkce má být zavolána při příjmu dat. Driver si zadané hodnoty uloží do vlastního datového typu, ve kterém má uloženou konfiguraci. Po inicializaci je možné používat funkce pro přenos dat využívající systém přerušení.

4.1.6 FTDI protokol

FTDI protokol je knihovna, ve které jsou implementovány jednotlivé odpovědi na příkazy specifikované v 4.2. Každý příkaz je naprogramován jako samostatná funkce.

4.1.7 Clock driver

Clock driver slouží k ovládání zdroje hodinového signálu pro FPGA. Umožňuje aktivovat a deaktivovat krystalový oscilátor.

4.1.8 Callback při příjmu znaku jednotkou UART

Tato funkce je předána pomocí ukazatele inicializační funkci FTDI driveru. Funkce detekuje v přijímaných datech příkazy a pokud se jedná o platný příkaz, uloží jej včetně jeho parametrů do vyhrazených globálních proměnných a nastaví vlajku signalizující nový příkaz.

4.2 Příkazy pro komunikaci s PC

Pro komunikaci mezi MCU a PC se využívá textový protokol. Jednotka UART v MCU je nakonfigurována na přenosovou rychlost 921 600 Bdps, přenášené rámce mají délku 8 b, nevyžívají paritu a mají jeden stop bit.

Komunikační protokol je z důvodu uživatelské přívětivosti na žádost zadavatele kompletně textový. Odpovědi jsou zakončeny symbolem EOL (End of line), jenž se využívá i jako oddělovač, pokud odpověď vrací více dat. Jako EOL je možné použít dvojici znaků `\r\n` (návrat vozíku a nový řádek), případně pouze znak `\n`.

V ukázkových výpisech bude symbol EOL pro přehlednost umístěn mezi lomené závorky: <EOL>. Pro názorné odlišení příkazu od odpovědi zařízení, budou příkazy uvozeny znakem \$. Tento znak není součástí příkazu, jde pouze rozlišovací označení.

GET STATE Vrátí aktuální stav zařízení, přitom neovlivní právě vykonávaný příkaz.

Výpis 4.1: Ukázka příkazu GET STATE.

```
$GET STATE<EOL>  
OK<EOL>  
<STATE><EOL>
```

<STATE> může nabývat těchto hodnot

{	READY : zařízení připraveno
	READ : čtení paměti
	ERASE : mazání paměti
	MEASUREMENT : měření spektra

MEASUREMENT START Zahájí měření. Lze použít i pro pokračování v měření, které bylo zastaveno.

Výpis 4.2: Ukázka příkazu MEASUREMENT START.

```
$MEASUREMENT START<EOL>  
OK<EOL>
```

MEASUREMENT STOP Zastaví právě probíhající měření. Po zpracování tohoto příkazu se zařízení nachází ve stavu READY, tedy čeká na nový příkaz.

Výpis 4.3: Ukázka příkazu MEASUREMENT STOP.

```
$MEASUREMENT STOP<EOL>  
OK<EOL>
```

MEMORY ERASE Zapíše do každé buňky SRAM hodnotu nula, čímž dojde k vynulování kanálových čítačů.

Výpis 4.4: Ukázka příkazu MEMORY ERASE.

```
$MEMORY ERASE<EOL>  
OK<EOL>
```

MEMORY READ Přečte obsah paměti. Příkaz vrátí hodnoty všech 1 024 čítačů kanálů. <CHANNEL n > je textová reprezentace hodnoty kanálového čítače v desítkové soustavě, kde n odpovídá číslu kanálu.

Výpis 4.5: Ukázka příkazu MEMORY READ.

```
$MEMORY READ<EOL>
OK<EOL>
DATA BEGIN<EOL>
<CHANNEL 1>
...
<CHANNEL 1024>
DATA END<EOL>
```

SET RISING LEVEL Nastaví práh pro náběžnou hranu pulzu. Tato hodnota se používá v jednotce DSP při detekci pulzů. Hodnota prahu <VALUE> pro rozsah $2 V_{PP}$ se vypočítá následovně:

$$\langle \text{VALUE} \rangle = 2^9(U + 1) \quad (4.1)$$

Kde U je prahové napětí ve voltech.

Výpis 4.6: Ukázka příkazu SET RISING LEVEL.

```
$SET RISING LEVEL <VALUE><EOL>
OK<EOL>
```

SET FALLING LEVEL Nastaví práh pro sestupnou hranu pulzu. Tato hodnota se používá v jednotce DSP při detekci pulzů. Hodnota prahu <VALUE> je dána rovnicí [4.1](#).

Výpis 4.7: Ukázka příkazu SET FALLING LEVEL.

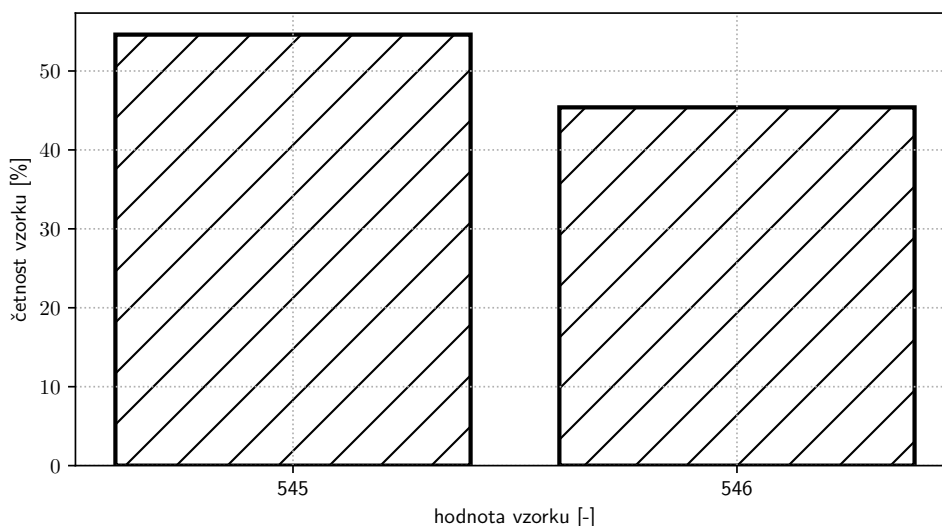
```
$SET FALLING LEVEL <VALUE><EOL>
OK<EOL>
```

5 Výsledky

V této kapitole okomentuji některé výsledky měření, která jsem během vývoje multikanálového analyzátoru prováděl. Dále se tato kapitola věnuje popisu ovládací aplikace.

5.1 Testování šumu systému

K získání základních informací o šumu analogové části systému jsem provedl následující test. Na vstupní SMA konektor jsem přišrouboval 50Ω terminátor, přes který se propojil vstupní signál se zemí zařízení. Systém v FPGA jsem rozšířil o integrovaný logický analyzátor (ILA), který snímal výstup ADC driveru. Ze získaných dat z ILA (16 384 vzorků) jsem vytvořil histogram, zobrazený na obrázku 5.1.



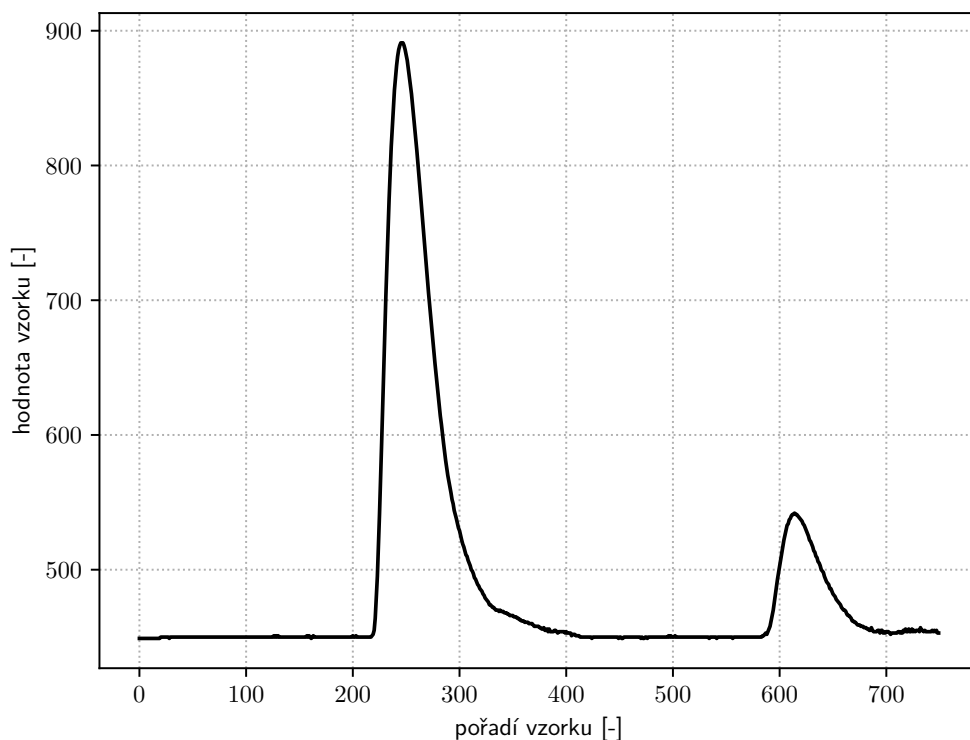
Obr. 5.1: Histogram hodnot vzorků měřených na výstupu ADC driveru.

V histogramu jsou zastoupeny dvě hodnoty, což vypovídá o tom, že šum systému ve výše specifikovaných podmínkách odpovídá jednomu LSB (nejnižšímu bitu). Dále je z histogramu patrný mírný offset, který je způsobený chováním LNA a také napěťovým úbytkem na terminátoru.

5.2 Testování digitalizace signálu z detektoru

Pro ověření, že celý analogový systém funguje korektně a hradlové pole zpracovává validní data, jsem se rozhodl provést test digitalizace signálu z detektoru. Na

vstup mnou vyvíjeného zařízení jsem připojil výstup z transimpedančního zesilovače, který upravuje signál z fotonásobiče. Do systému v FPGA jsem vložil IP blok ILA (Integrated Logic Analyzer), který zaznamenává výstupní signál z ADC driveru. Před scintilační detektor byl umístěn zářič ^{57}Co , ze kterého vyletují γ fotony s dominantním zastoupením energií (6 keV, 14 keV a 122 keV). Na obrázku 5.2 lze pozorovat získaný signál na výstupu ADC driveru. Systémové hodiny byly při tomto experimentu generovány z krystalového oscilátoru, který poskytuje signál s frekvencí 100 MHz. Jak je z obrázku patrné, pulzy jsou široké přibližně sto vzorků, což poskytuje dostatečné množství dat pro jejich zpracování.

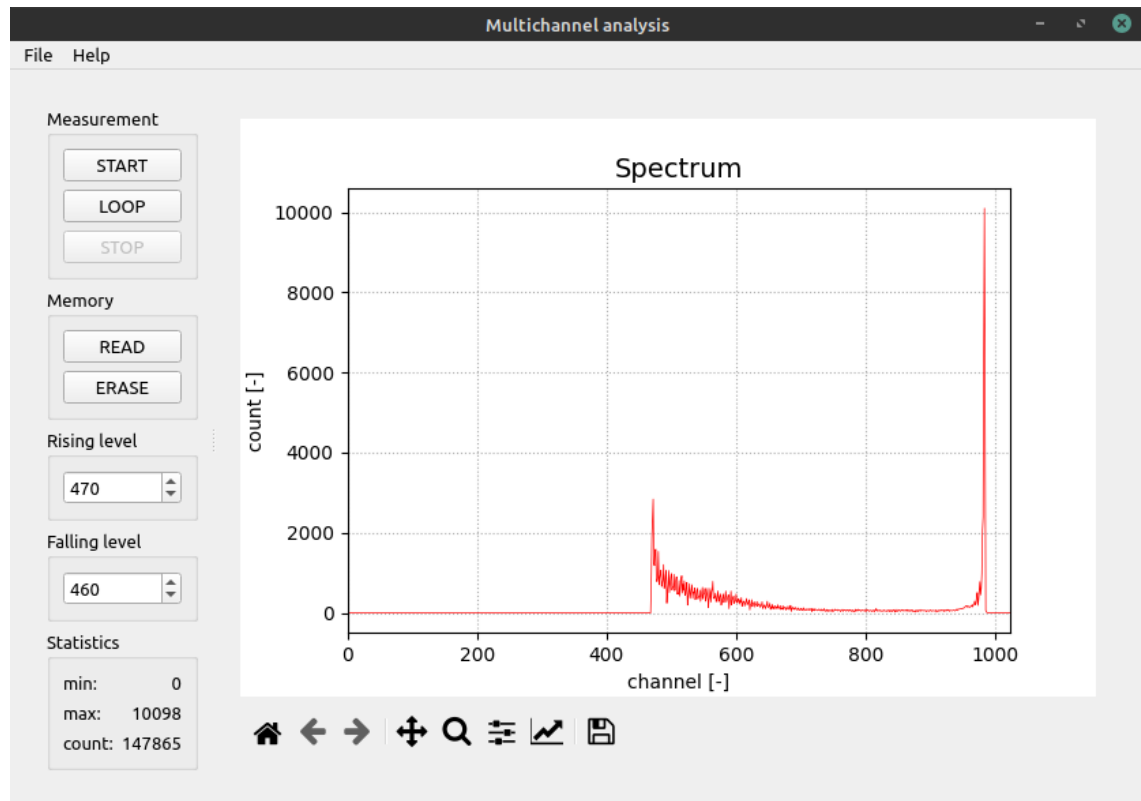


Obr. 5.2: Zachycené pulzy z detektoru na výstupu ADC driveru v FPGA.

5.3 Ovládací aplikace

Pro testování a zjednodušení obsluhy zařízení jsem vyvinul ovládací aplikaci. Díky tomu, že je napsána v programovacím jazyce Python, je multiplatformní. Umožňuje ovládat multikanálový analyzátor, zobrazovat průběh měření, exportovat a importovat změřená data ve formátu CSV. K realizaci GUI jsou využity knihovny PySide2 a matplotlib. Knihovna PySide2 poskytuje python API (Application Programming

Interface) pro knihovnu QT. K vytvoření widgetu grafu jsem využil knihovnu matplotlib, k ovládání virtuální sériové linky knihovnu pyserial a pro zpracování, export a import dat knihovnu numpy. Vyvinutá aplikace obsahuje i knihovnu, ve které je implementován komunikační protokol s multikanálovým analyzátozem. Tu je možné využít pro skriptování měření.

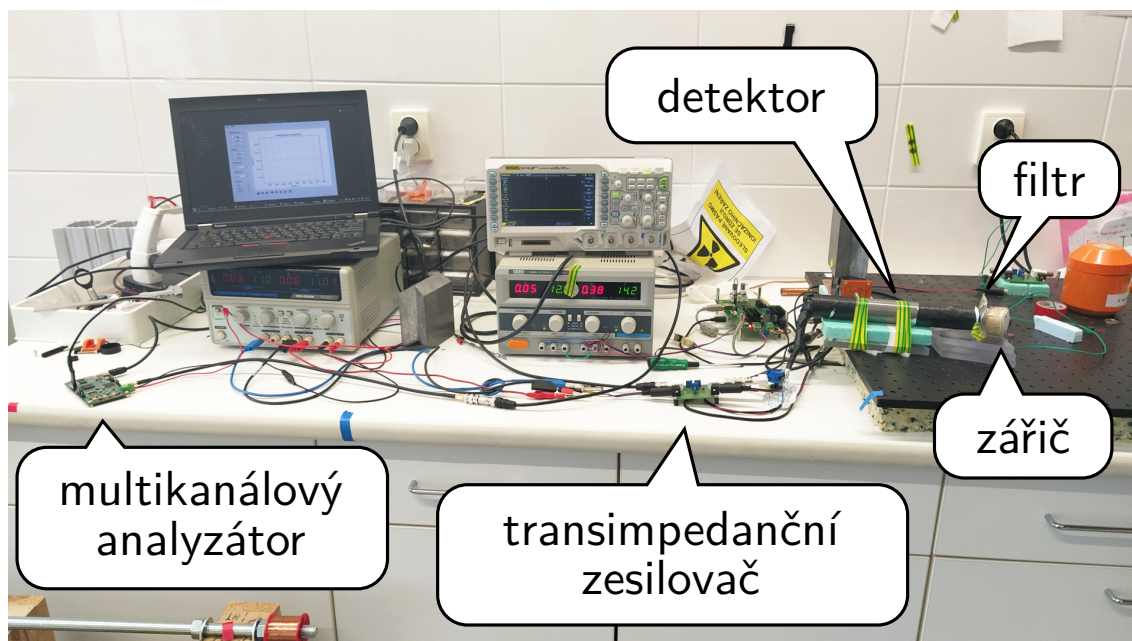


Obr. 5.3: Ukázka GUI vyvinuté aplikace.

5.4 Ověřovací měření spektra

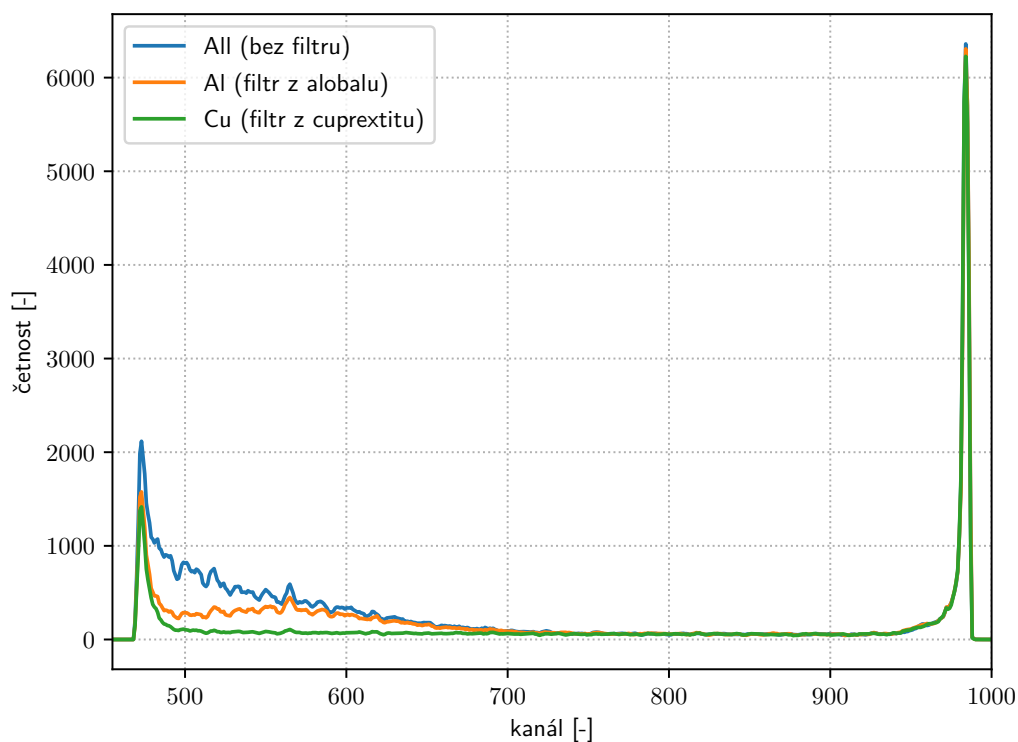
Vyvíjeným multikanálovým analyzátozem bylo provedeno testovací měření spektra γ záření v laboratoři experimentální fyziky Univerzity Palackého v Olomouci. Jako zářič byl použit kobalt ^{57}Co rozpadající se na železo ^{57}Fe . Při tomto rozpadu ze zářiče vyletují γ fotony s dominantním zastoupením energií (6 keV, 14 keV a 122 keV). Zářič byl umístěn před scintilační detektor s fotonásobičem R6095, který pro svá měření sestavili pracovníci laboratoře [4].

Výstup detektoru byl upraven transimpedančním zesilovačem s integrovaným tvarovačem tak, aby ve spektru bylo možné pozorovat γ fotony o energii 6 keV a 14 keV, které jsou zajímavé pro spektroskopické účely. Tento signál byl přiveden na vstup vyvíjeného multikanálového analyzátozem. Sestavená aparatura je na zachycena obrázku 5.4.



Obr. 5.4: Aparatura pro spektroskopii gama záření.

Nejprve bylo změřeno celé (nefiltrované) spektrum zářiče, v grafu označené jako All. Následně byl mezi zářič a detektor vložen filtr z hliníkové fólie (alobalu) o tloušťce $200\ \mu\text{m}$, který zabraňuje průchodu 6keV fotonům. Získané spektrum je v grafu označeno Al. Dále byl před zářič vložen měděný filtr (dvouvrstvá cuprexitová destička o celkové tloušťce mědi $70\ \mu\text{m}$), který zabraňuje průchodu jak 6keV , tak 14keV fotonů. Změřené spektrum je v grafu označeno Cu. Získaná spektra jsou zobrazena na obrázku 5.5.



Obr. 5.5: Změřená spektra gama záření.

V pravé části spektra se nachází špička, která je způsobena pulzy v saturaci, které pochází od detekce částic s vysokou energií převážně od γ fotonů o energii 122 keV. Tyto pulzy nejsou v tomto případě pro spektroskopii podstatné.

Nejdůležitější částí multikanálové analýzy je oblast mezi kanály 500 až 700, v níž jsou detekovány γ fotony o energii 14 keV, které jsou důležité pro danou spektroskopickou metodu. To, že se jedná opravdu o fotony s příslušnou energií, je zřetelné ze změny tvaru spektra při použití jednotlivých filtrů, které stíní vybrané energie γ záření. Špička, kterou lze pozorovat mezi kanály 450 až 500, je způsobena radiačním pozadím v laboratoři, šumem fotonásobiče a vstupní elektroniky.

Závěr

Cílem práce bylo vyvinout multikanálový analyzátor použitelný pro měření spektra gama záření. Nejprve jsem v rámci semestrální práce vyrobil dva testovací moduly, a to modul digitizéru obsahující LNA a ADC a modul paměti osazený SRAM. Moduly jsem otestoval, byly funkční, ale modul digitizéru vykazoval značný šum. Ten byl způsobený převážně konstrukcí na dvouvrstevném plošném spoji a také snahou o oddělení digitální a signální země. Oddělení navzdory předpokladu nezlepšovalo vlastnosti zařízení, proto jsem se rozhodl u prototypu země opět spojit pomocí vysokofrekvenčních lanek, což šum zredukovalo.

Po oživení hardwaru jsem propojil vyvinuté moduly s vývojovým kitem osazeným FPGA. Vytvořil jsem několik VHDL modulů pro otestování funkce digitizéru. To umožnilo odesílat digitalizovaná data do osobního počítače pro podrobnější vyhodnocení vlastností vyvíjeného přístroje. Také jsem ve skriptovacím jazyce Python vytvořil několik skriptů pro zpracování dat z digitizéru.

V rámci samotné diplomové práce jsem navrhl schéma a čtyřvrstvý plošný spoj multikanálového analyzátoru, který kromě vylepšené verze elektroniky ze semestrální práce obsahoval i FPGA Spartan-7 v BGA pouzdře a další obvody.

Následně jsem se zaměřil na vývoj systému v hradlovém poli, který jsem popsal jazykem VHDL. Kromě popisu systému jsem vytvořil i soubory pro jeho simulaci, což umožnilo jeho kompletní testování.

Po vyrobení testovacího prototypu a jeho oživení jsem upravil systém v hradlovém poli tak, aby fungoval na nově vyvinuté desce. Bylo nezbytné upravit časování vyčítání dat z ADC, abych zajistil jejich korektní čtení.

Dále jsem naprogramoval firmware pro MCU, který komunikuje s FPGA a PC. Firmware umožňuje ovládat měření a vyčítat změřená spektra. Komunikace s PC probíhá pomocí textového protokolu, přenášeného prostřednictvím rozhraní UART. Pro komunikaci mezi MCU a FPGA jsem navrhl binární protokol, kterým se komunikuje po sběrnici SPI.

Vyvinuta byla rovněž aplikace umožňující ovládat multikanálový analyzátor. Ta usnadňuje obsluhu vytvořeného zařízení a dovoluje ukládat změřená data do osobního počítače.

Zařízení bylo ověřeno měřeními pulzů generovaných z generátoru i ze scintilačního detektoru. Měření otestovala použitelnost zařízení pro spektroskopii gama záření. Získaná spektra odpovídají teoretickým předpokladům, avšak pro finální nasazení přístroje bude nezbytné udělat jeho podrobnější charakterizaci.

Do budoucna by bylo vhodné doplnit vstupní invertující pin LNA kladným předpětím, aby bylo možné rozšířit použitelné rozlišení zařízení pro aktuální měřící aparaturu.

Literatura

- [1] Analog Devices: 10-Bit, 65/80/105 MSPS 3 V A/D Converter. Technická Zpráva AD9214, Analog Devices, únor 2002.
URL <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9214.pdf>
- [2] Analog Devices: Ultralow Noise Drivers for Low Voltage ADCs. Technická Zpráva ADA4930-1/ADA4930-2, Analog Devices, květen 2017, rev. C.
URL https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4930-1_4930-2.pdf
- [3] Gerndt, J.: *Detektory ionizujícího záření*. V Praze: České vysoké učení technické, 2011, ISBN 978-80-0104-710-1.
- [4] Hamamatsu: Photomultiplier tubes. Technická Zpráva PMT_TPMZ0002E, Hamamatsu, únor 2016.
URL https://www.hamamatsu.com/resources/pdf/etd/PMT_TPMZ0002E.pdf
- [5] ISSI: 128K x 16 HIGH SPEED ASYNCHRONOUS CMOS STATIC RAM WITH ECC. Technická Zpráva IS61WV12816EDBLL, ISSI, říjen 2011.
URL <http://www.issi.com/WW/pdf/61-64WV12816EDBLL.pdf>
- [6] ON Semiconductor: Linear Regulator – Dual-Rail, Very Low-Dropout, Programmable Soft-Start 3.0 A. Technická Zpráva NCP59744, ON Semiconductor, listopad 2019.
URL <https://www.onsemi.com/pub/Collateral/NCP59744-D.PDF>
- [7] Qmtech: XC7S15-1FTGB196-CORE-BOARD. Technická zpráva, Qmtech, říjen 2018.
URL <http://www.chinaqmtech.com/filedownload/18056>
- [8] Stejskal, A.: Tvarová filtrace signálu v detekci jaderného záření [online]. 2017 [cit. 2020-07-13].
URL <https://theses.cz/id/bjhxkr>
- [9] Stockley, C.: *Velká ilustrovaná encyklopedie : fyzika, chemie, biologie*. Havlíčkův Brod: Fragment, 2003, ISBN 80-7200-810-2.
- [10] Texas Instruments: LM3880 Three-Rail Simple Power Sequencer. Technická Zpráva LM3880, Texas Instruments, listopad 2018.
URL <https://www.onsemi.com/pub/Collateral/NCP59744-D.PDF>

- [11] Texas Instruments: TLV431x Low-Voltage Adjustable Precision Shunt Regulator. Technická Zpráva TLV431, Texas Instruments, květen 2018.
URL <https://www.ti.com/lit/ds/symlink/tlv431.pdf>

- [12] Xilinx: 7 Series FPGAs Configuration. Technická Zpráva UG470, Xilinx, srpen 2018.
URL https://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf

- [13] Xilinx: 7 Series FPGAs PCB Design Guide. Technická Zpráva UG483, Xilinx, květen 2019.
URL https://www.xilinx.com/support/documentation/user_guides/ug483_7Series_PCB.pdf

- [14] Xilinx: Spartan-7 FPGAs Data Sheet: DC and AC Switching Characteristics. Technická Zpráva DS189, Xilinx, březen 2019.
URL https://www.xilinx.com/support/documentation/data_sheets/ds189-spartan-7-data-sheet.pdf

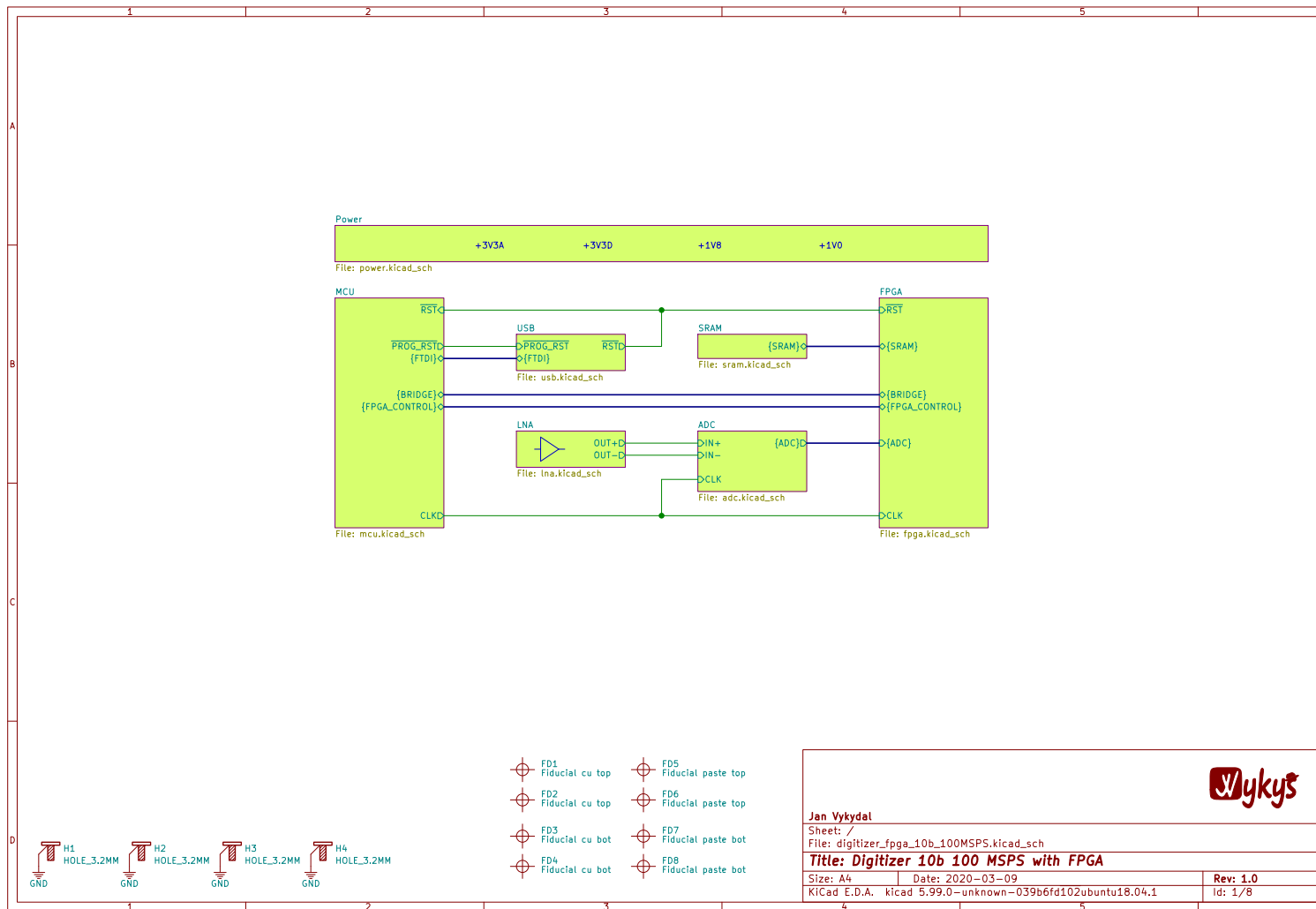
Seznam symbolů, veličin a zkratek

FPGA	Field-programmable gate array
LNA	Low-noise amplifier
ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
LPF	Low-pass filter
UART	Universal asynchronous receiver-transmitter
USB	Universal Serial Bus
SRAM	Static random-access memory
MSB	Most significant bit
LSB	Least significant bit
LVDS	Low-voltage differential signaling
IOB	Input/Output Block
PLL	Phase-Locked Loop
MMCM	Mixed-Mode Clock Manager
SPI	Serial Peripheral Interface
JTAG	Joint Test Action Group
LUT	Lookup table
DSP	Digital signal processing
IP	Intellectual Property
MCU	Microcontroller Unit
PC	Personal Computer
HW	Hardware
FW	Firmware
SW	Software

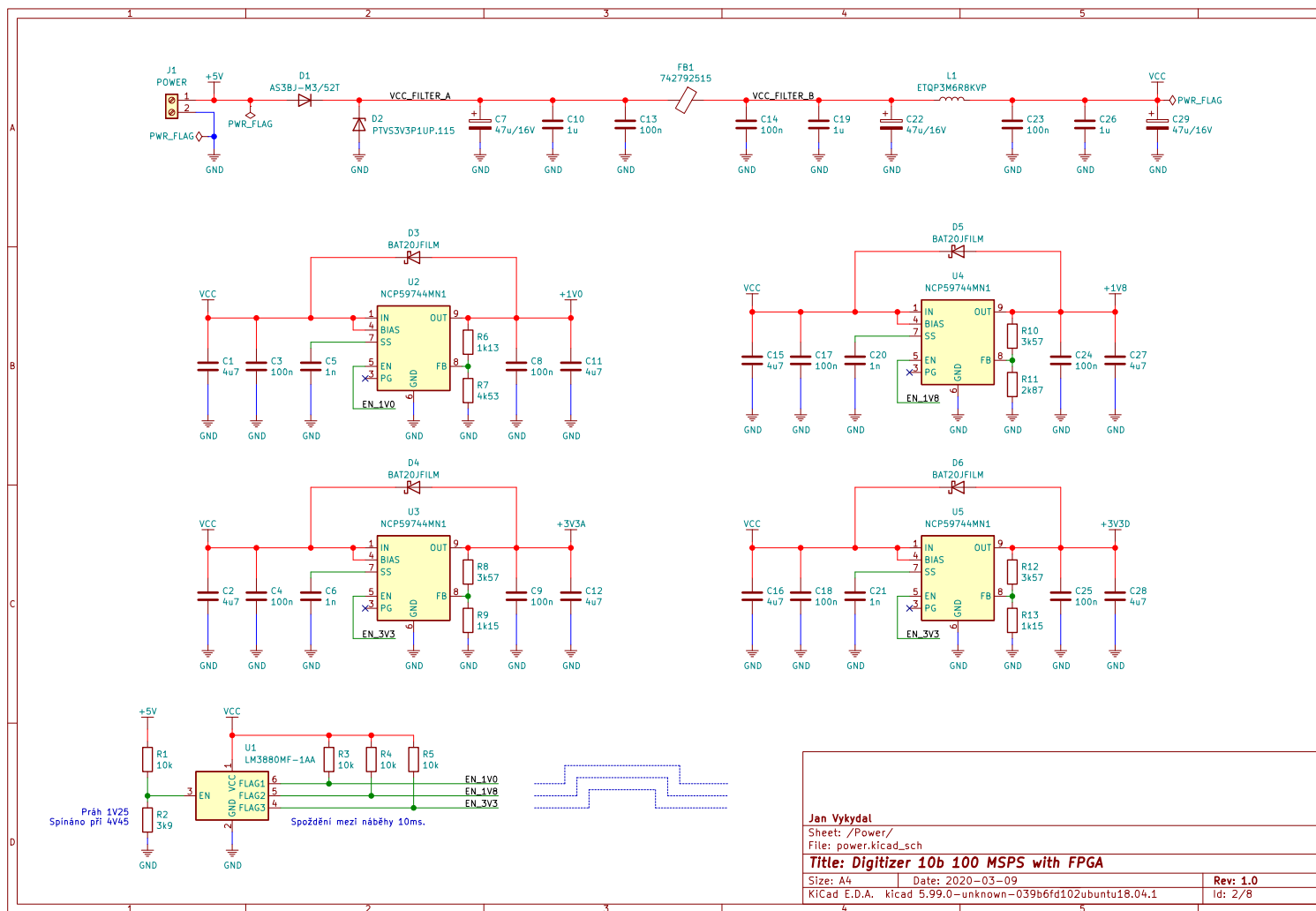
Seznam příloh

A	Přiložená schémata zapojení	60
B	Přiložené osazovací plány	69
C	Přiložený výkres plošného spoje	72
D	Bloková schémata číslicového systému v hradlovém poli	77
E	Fotografie zařízení	80
F	Zdrojové soubory	85

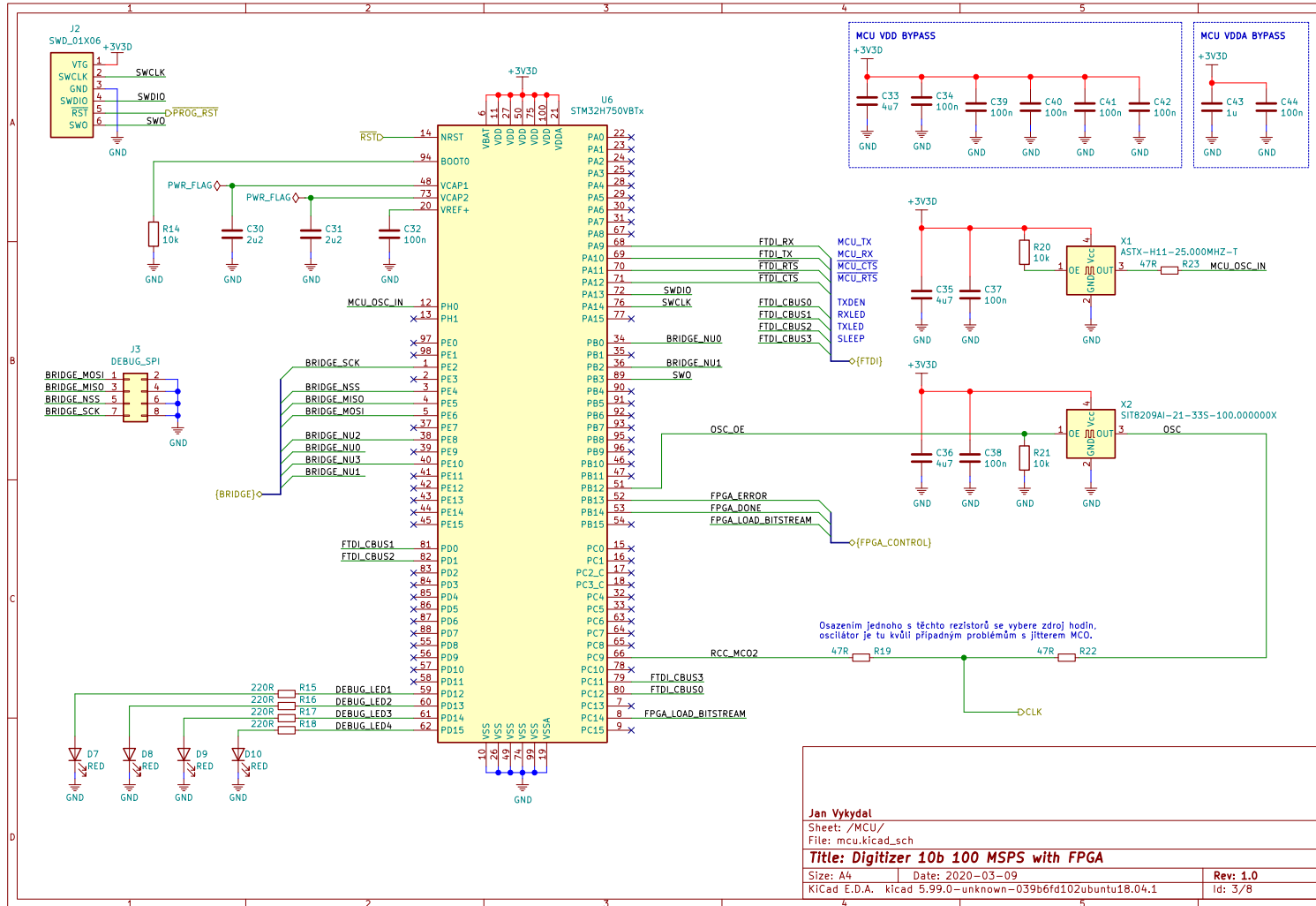
A Přiložená schémata zapojení



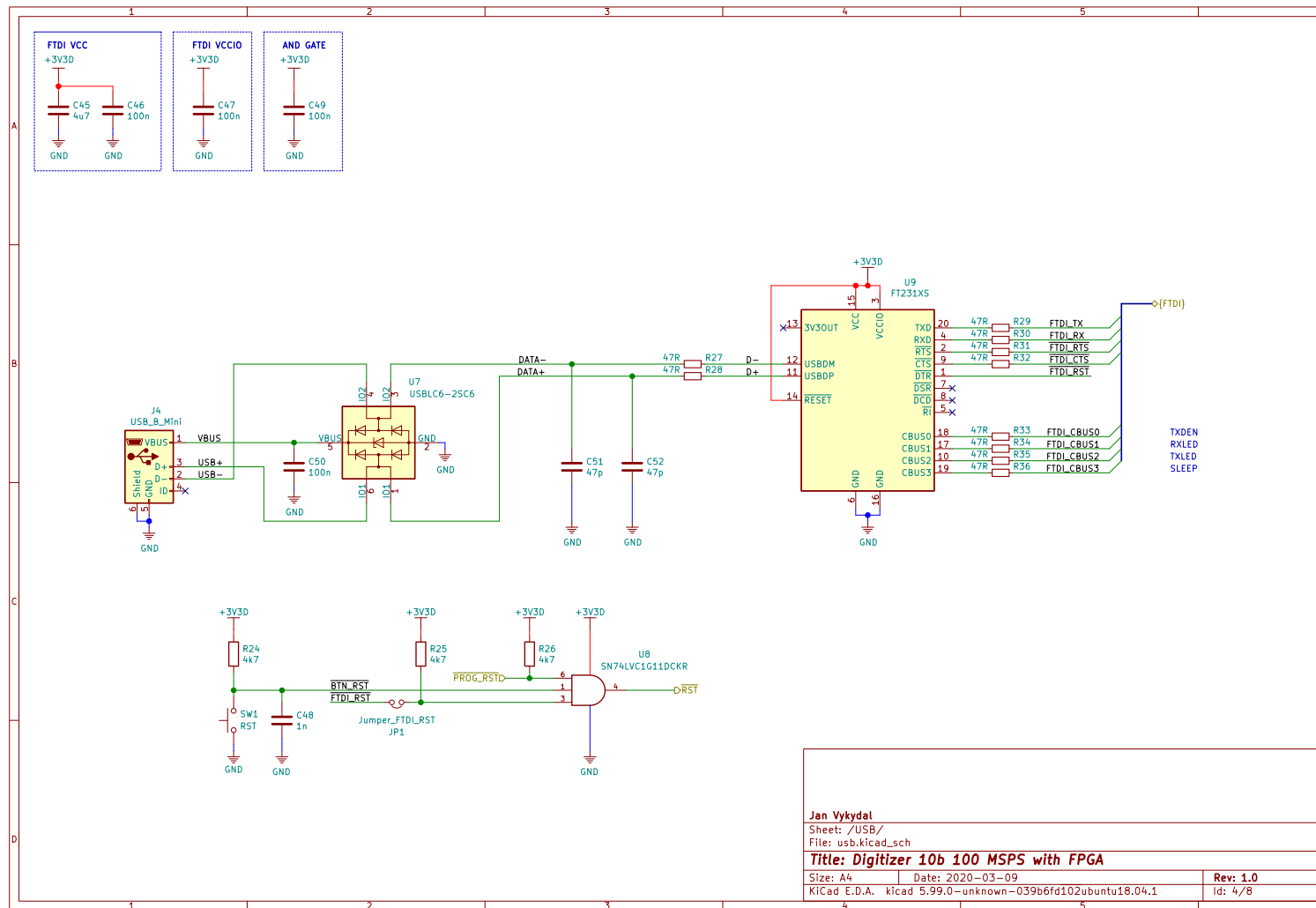
Obr. A.1: Schéma zapojení – strana 1. Blokové schéma.



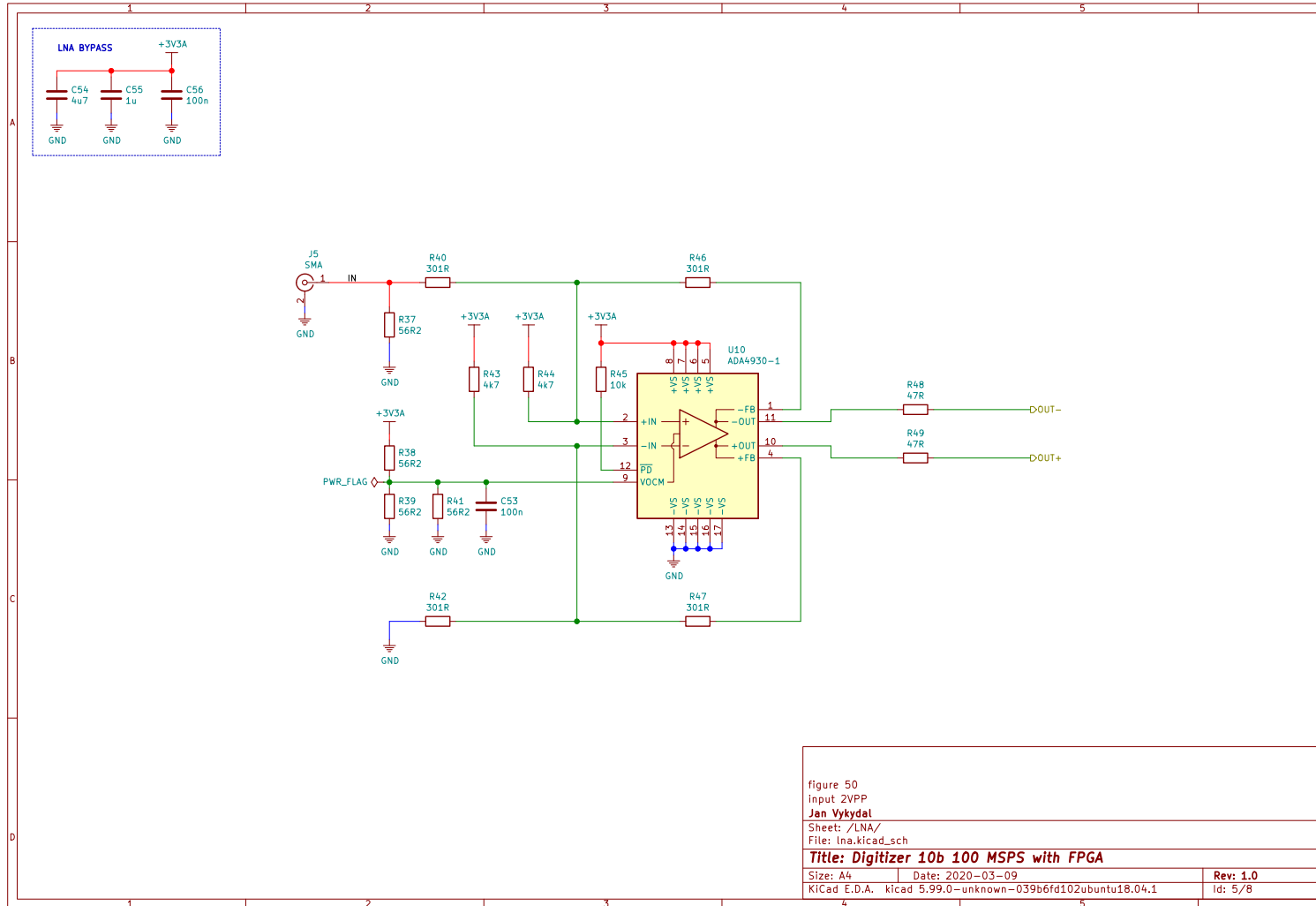
Obr. A.2: Schéma zapojení – strana 2. Napájení.



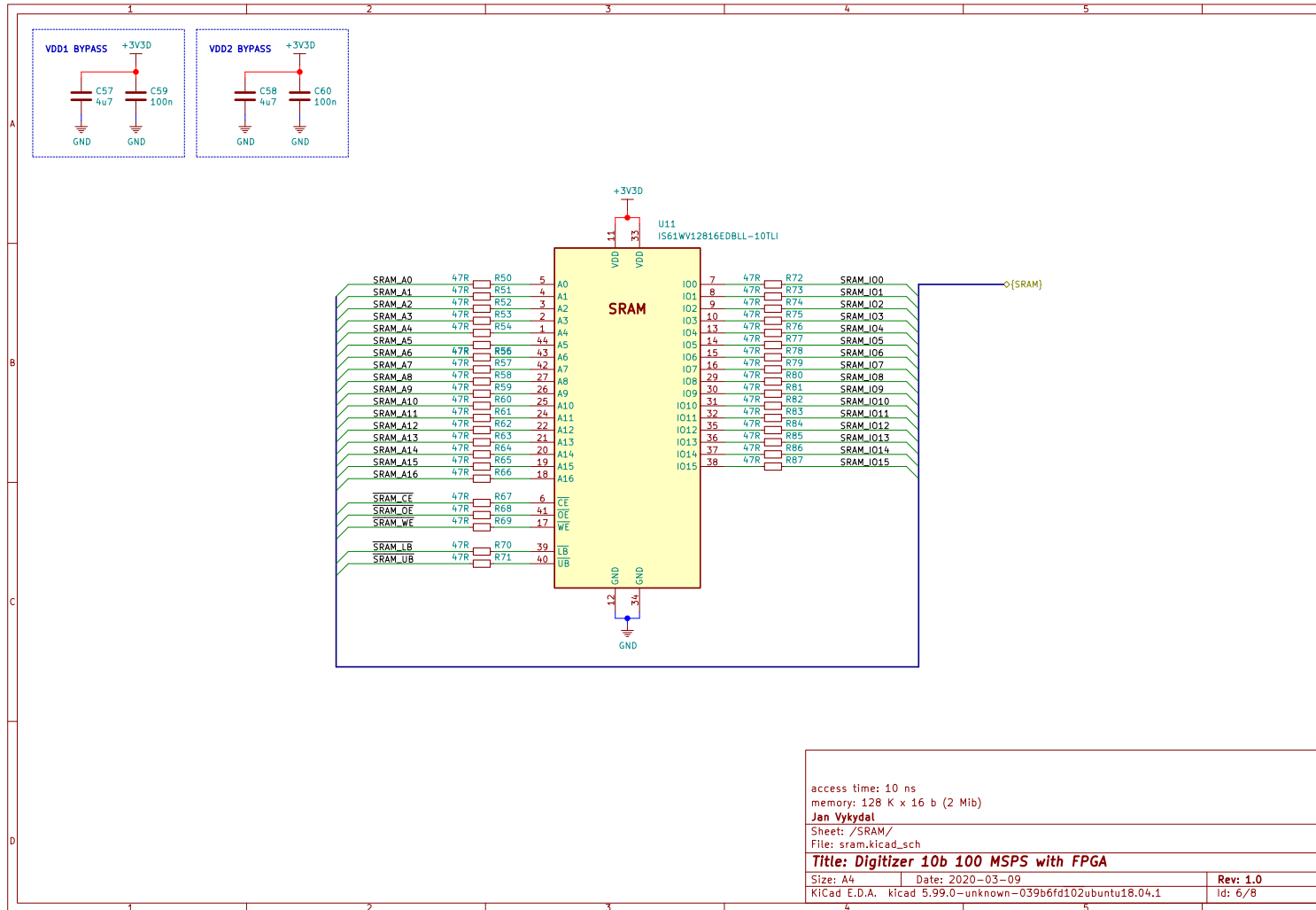
Obr. A.3: Schéma zapojení – strana 3. MCU.



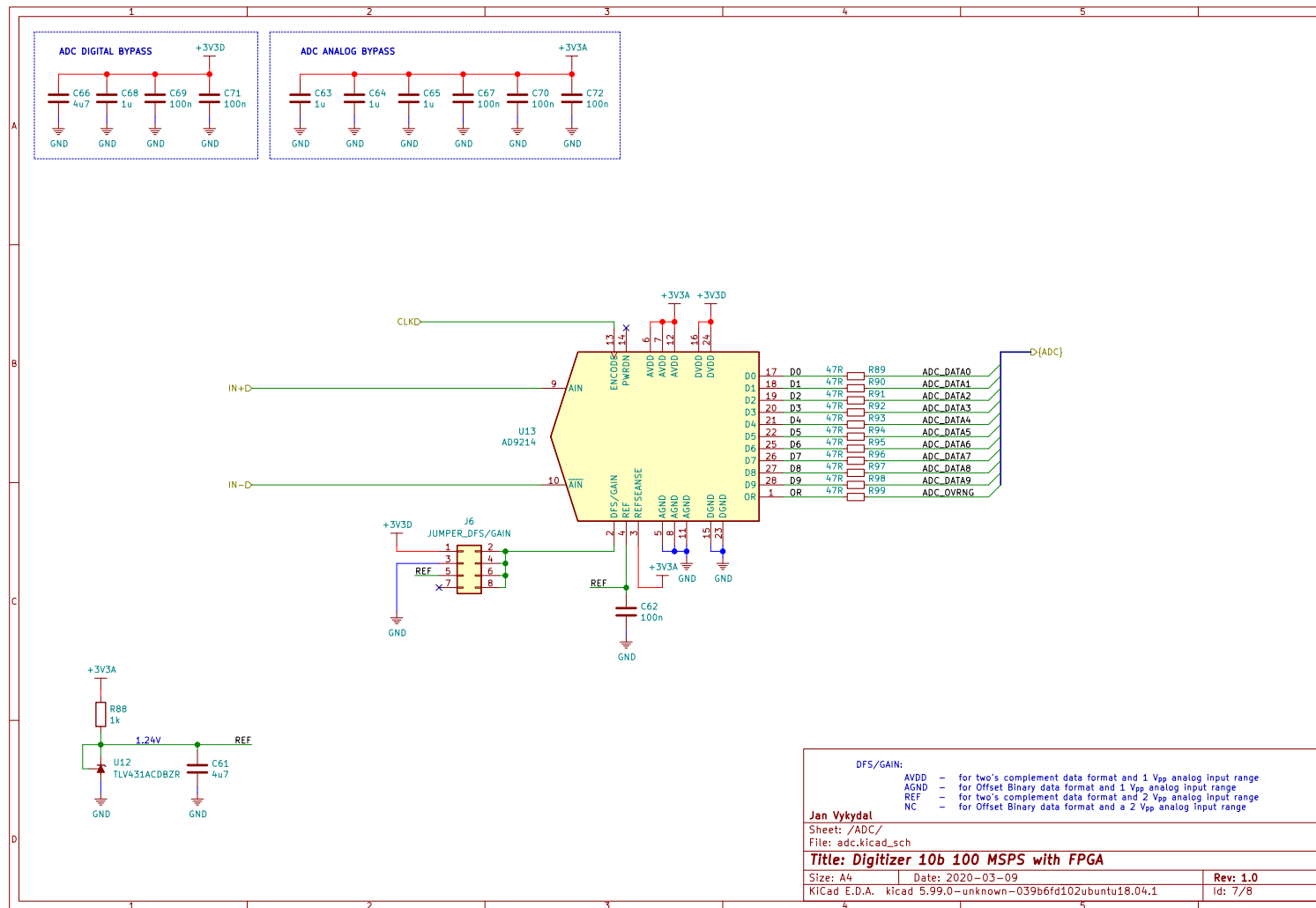
Obr. A.4: Schéma zapojení – strana 4. USB.



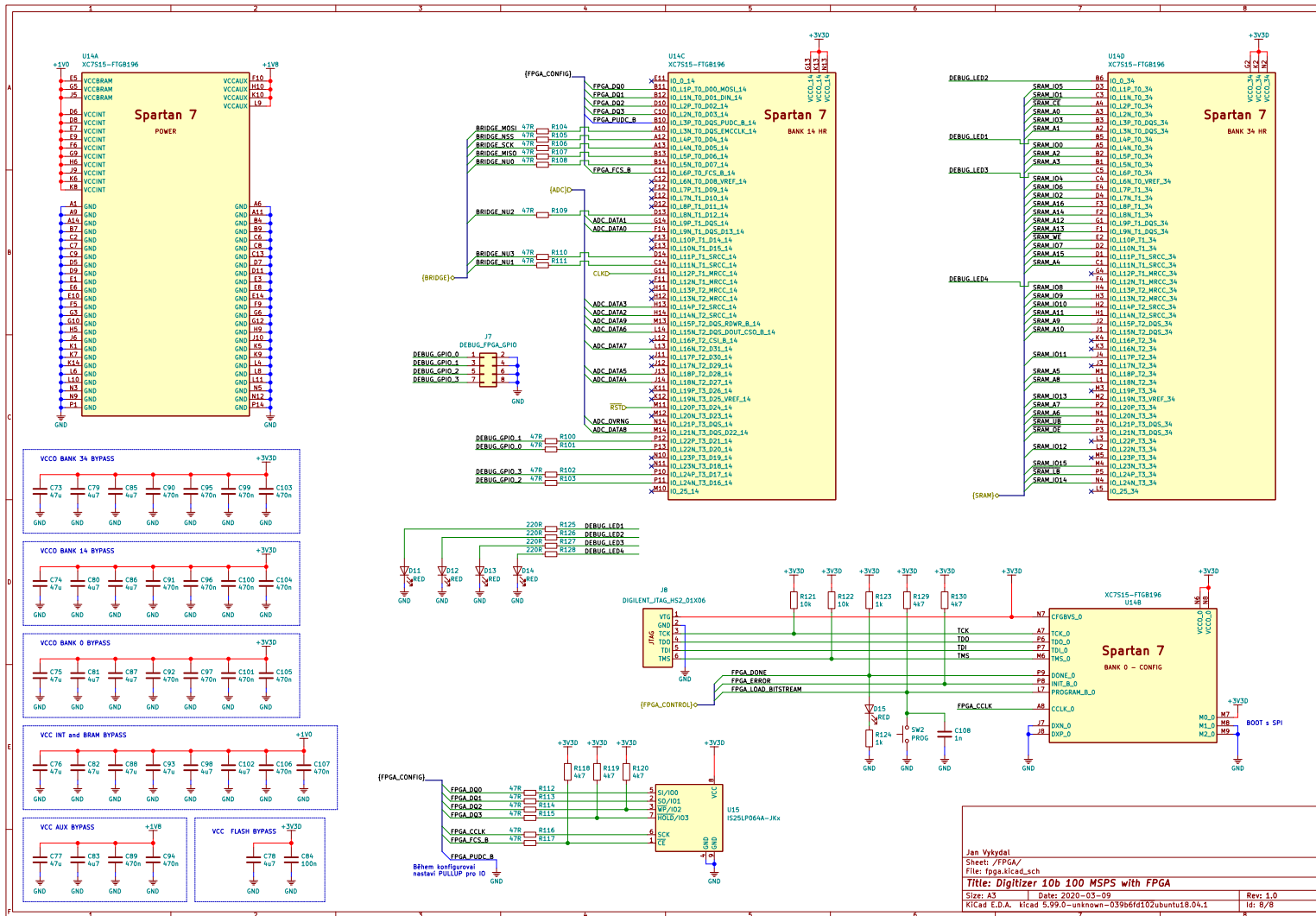
Obr. A.5: Schéma zapojení – strana 5. LNA.



Obr. A.6: Schéma zapojení – strana 6. SRAM.



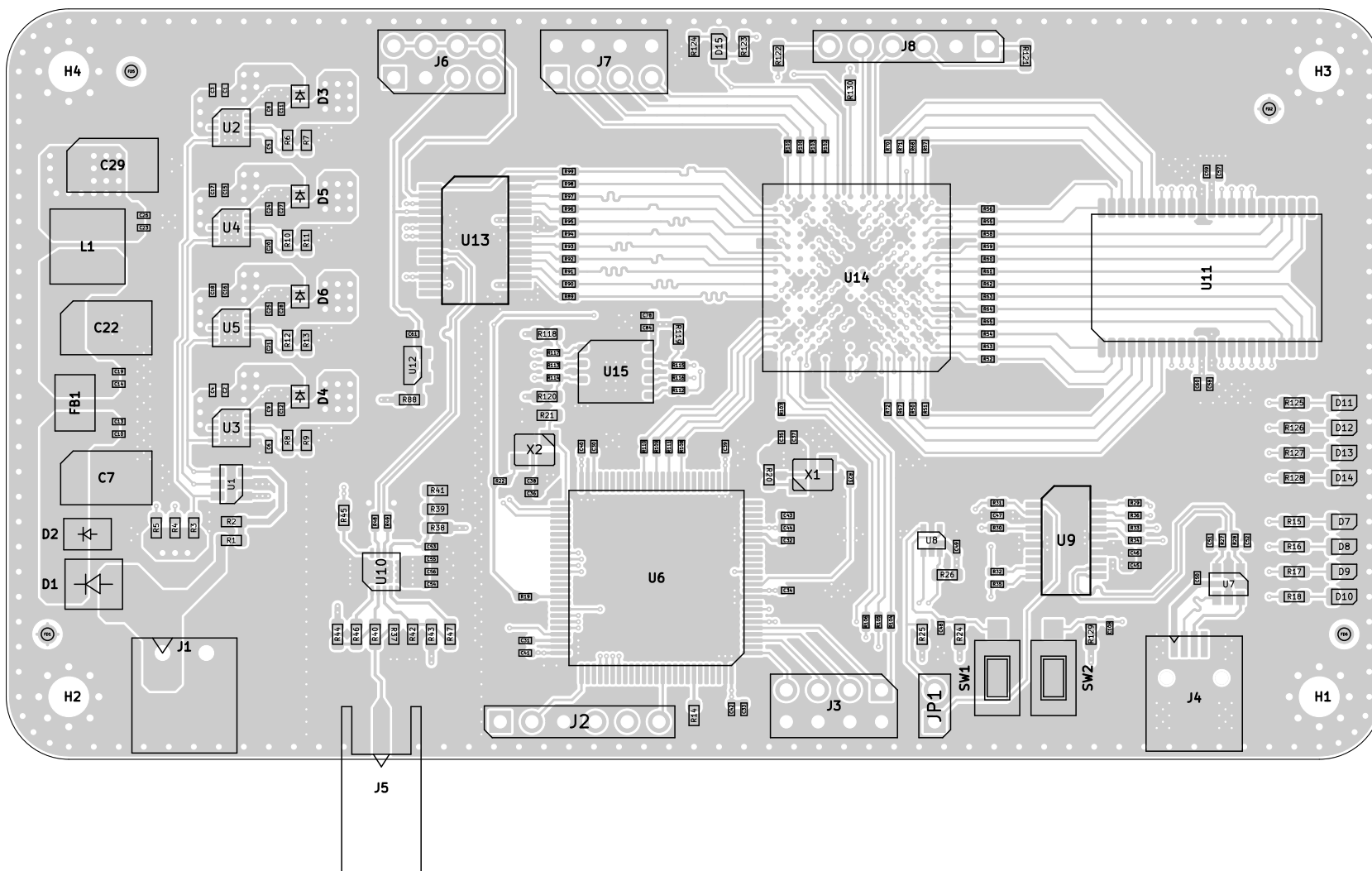
Obr. A.7: Schéma zapojení – strana 7. ADC.



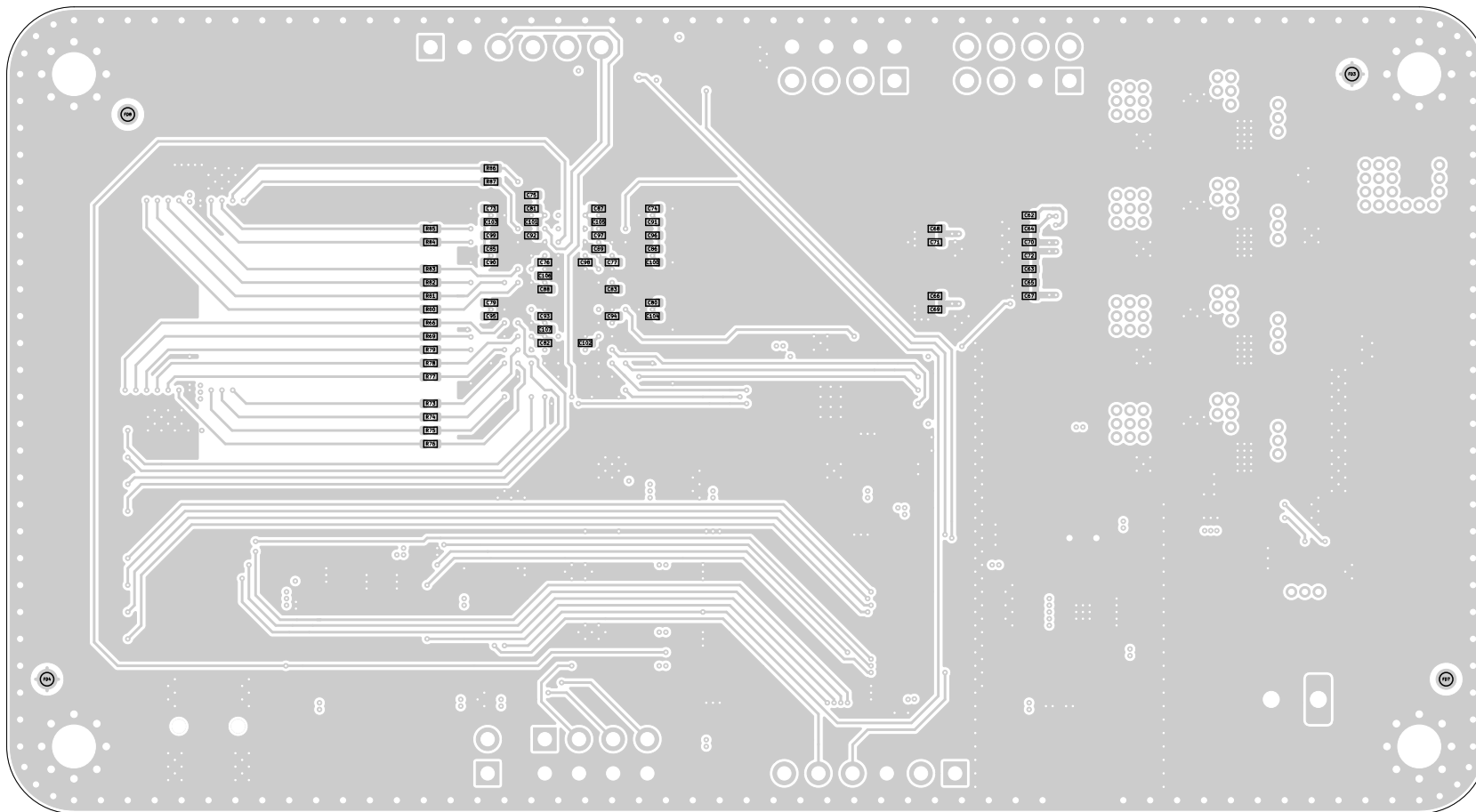
Obr. A.8: Schéma zapojení – strana 8. FPGA.

Jan Vytýdal
 Sheet: FPGA/1
 File: fpga.kicad_sch
Title: Digitizer 10b 100 MSPS with FPGA
 Size: A3 Date: 2020-03-09 Rev: 1.0
 KiCad E.D.A. Kicad 5.99.0-unknown-03966f02ubuntu18.04.1 18: 8/8

B Přiložené osazovací plány

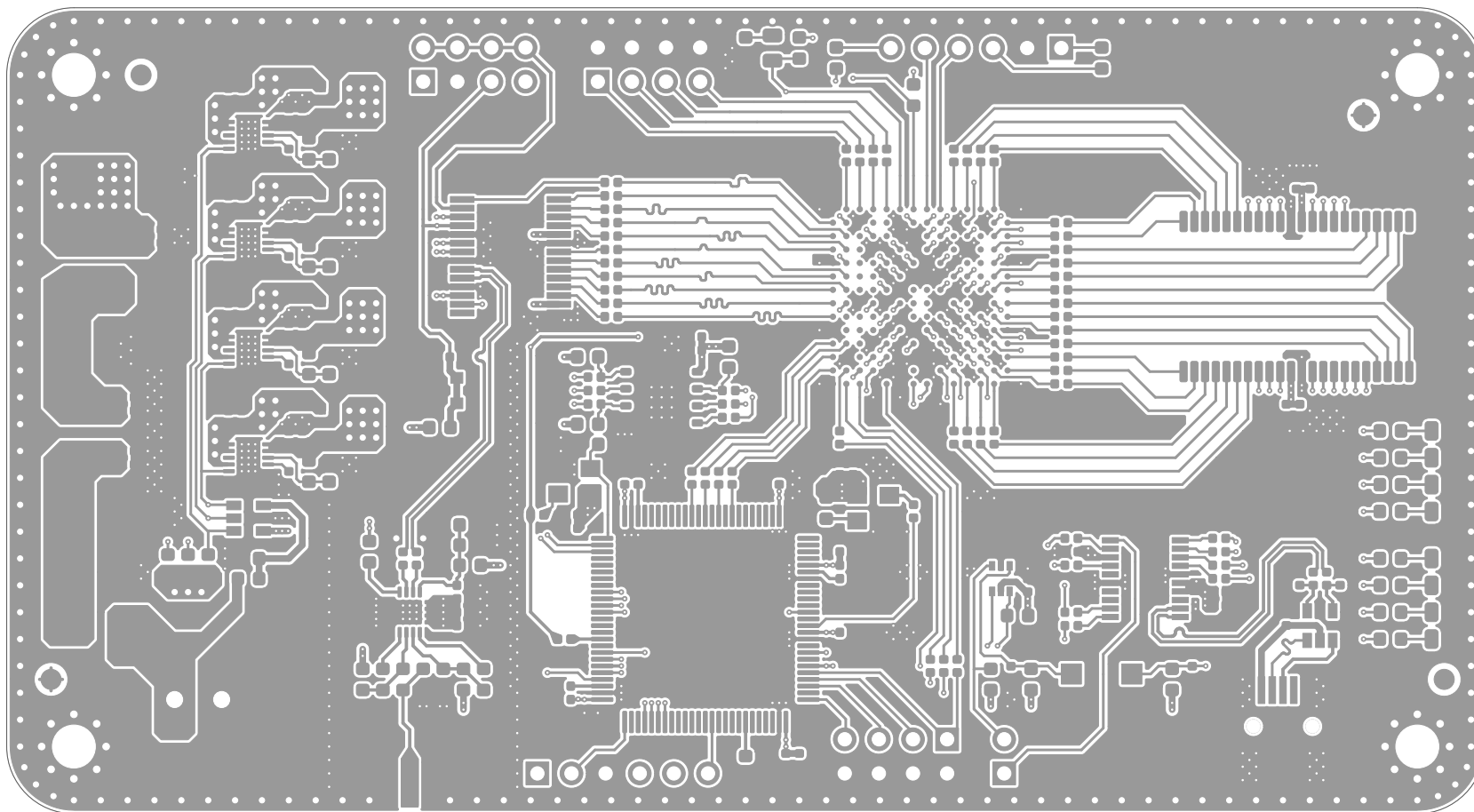


Obr. B.1: Osazovací plán digitizéru – pohled na stranu součástek.

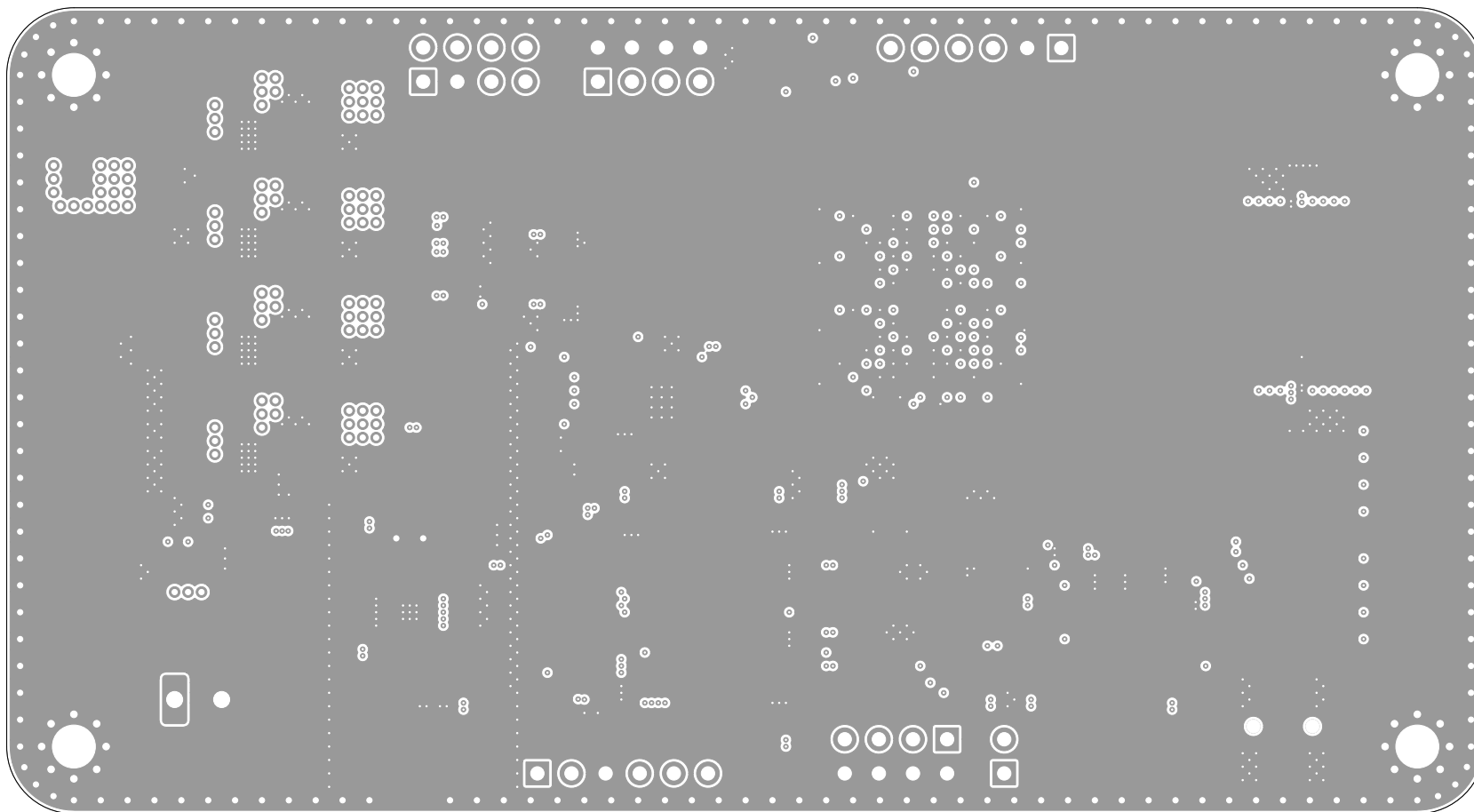


Obr. B.2: Osazovací plán digitizéru – pohled na stranu spojů.

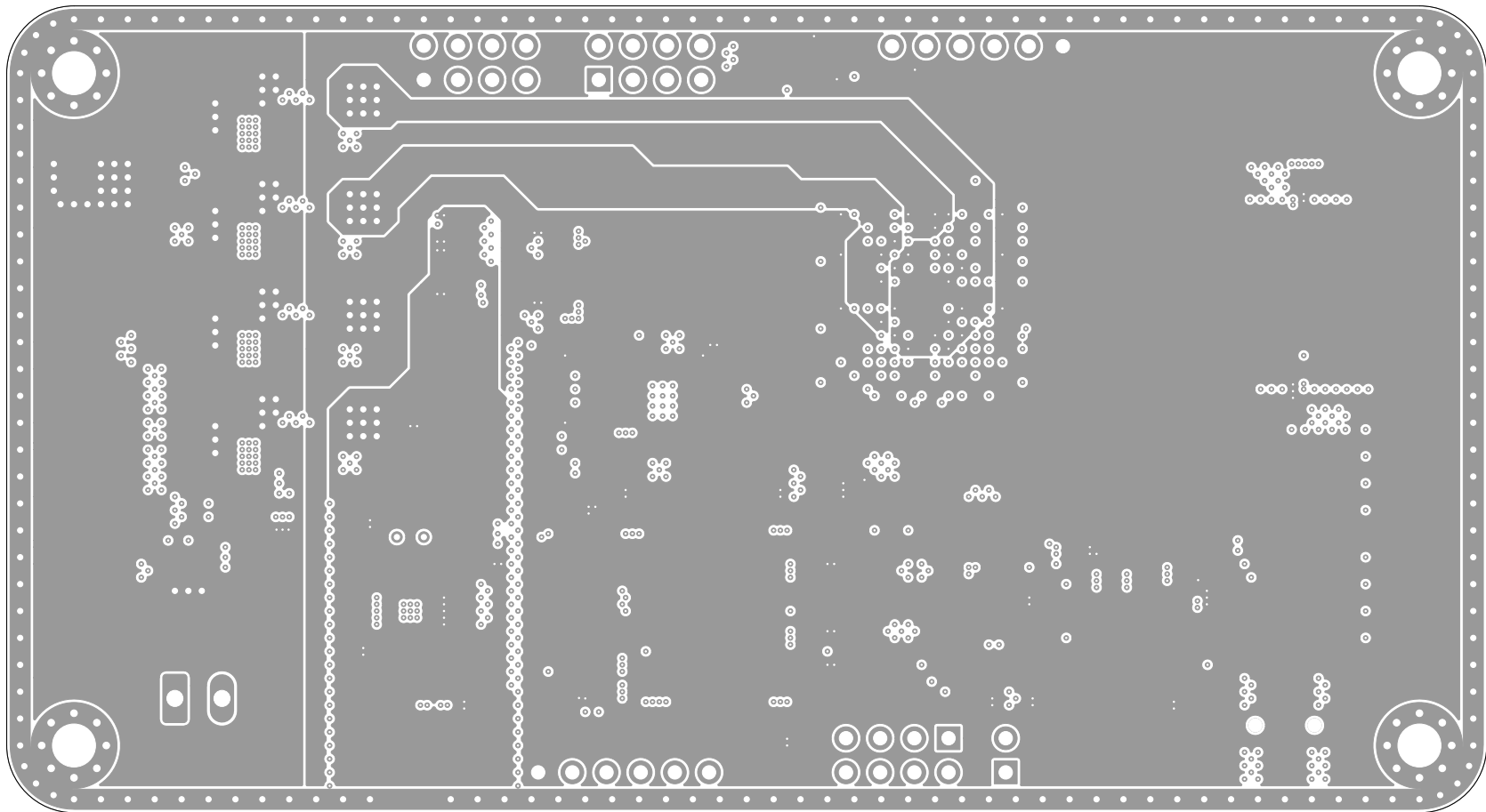
C Přiložený výkres plošného spoje



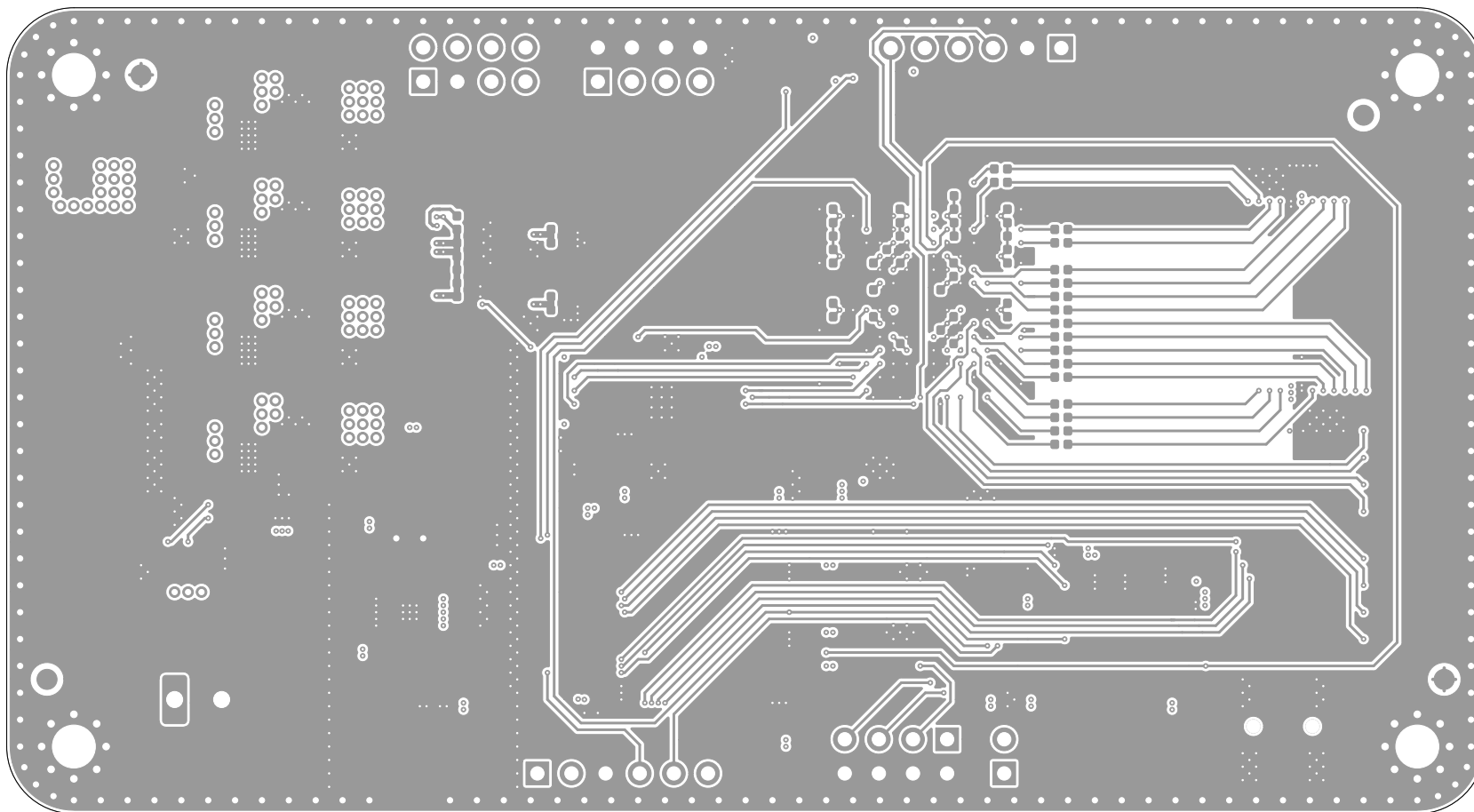
Obr. C.1: Plošný spoj – vrstva L1. Signály.



Obr. C.2: Plošný spoj – vrstva L2. Rozlitá zem.



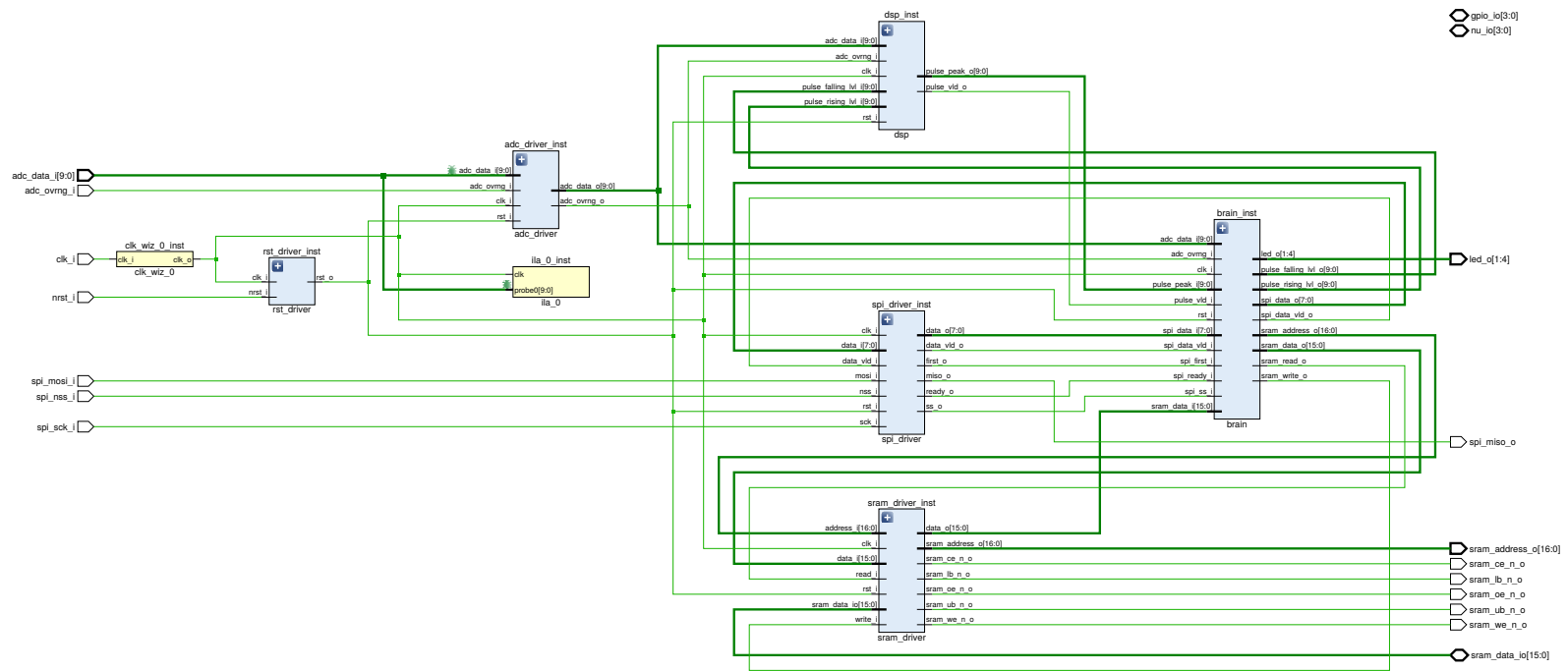
Obr. C.3: Plošný spoj – vrstva L3. Napájení.



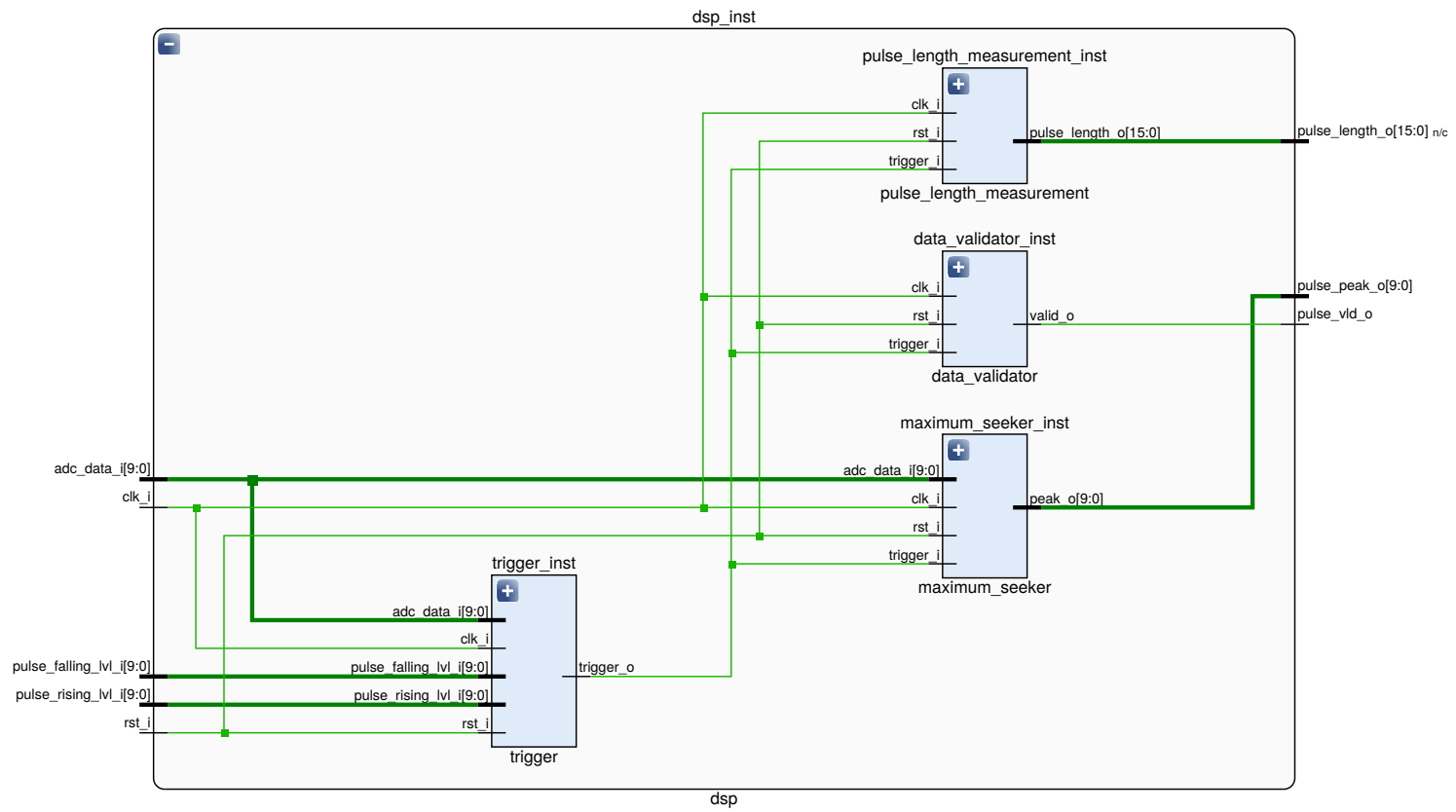
Obr. C.4: Plošný spoj – vrstva L4. Signály.

D Bloková schémata číslicového systému v hradlovém poli

Přiložená schémata byla vygenerována z nejvyššího VHDL modulu top.vhdl a modulu dsp.vhdl ve vývojovém prostředí Vivado.

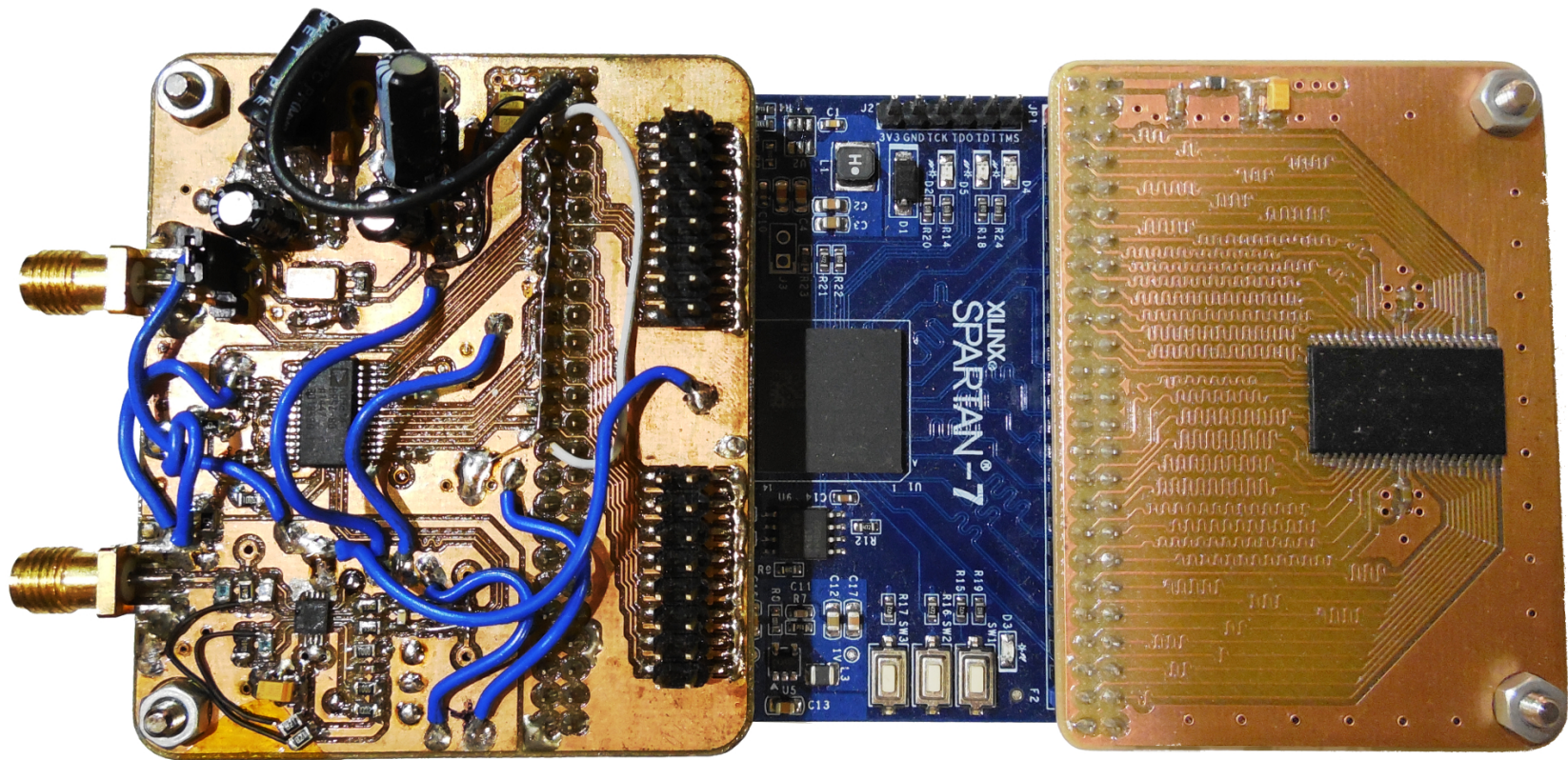


Obr. D.1: Blokové schéma systému v hradlovém poli.

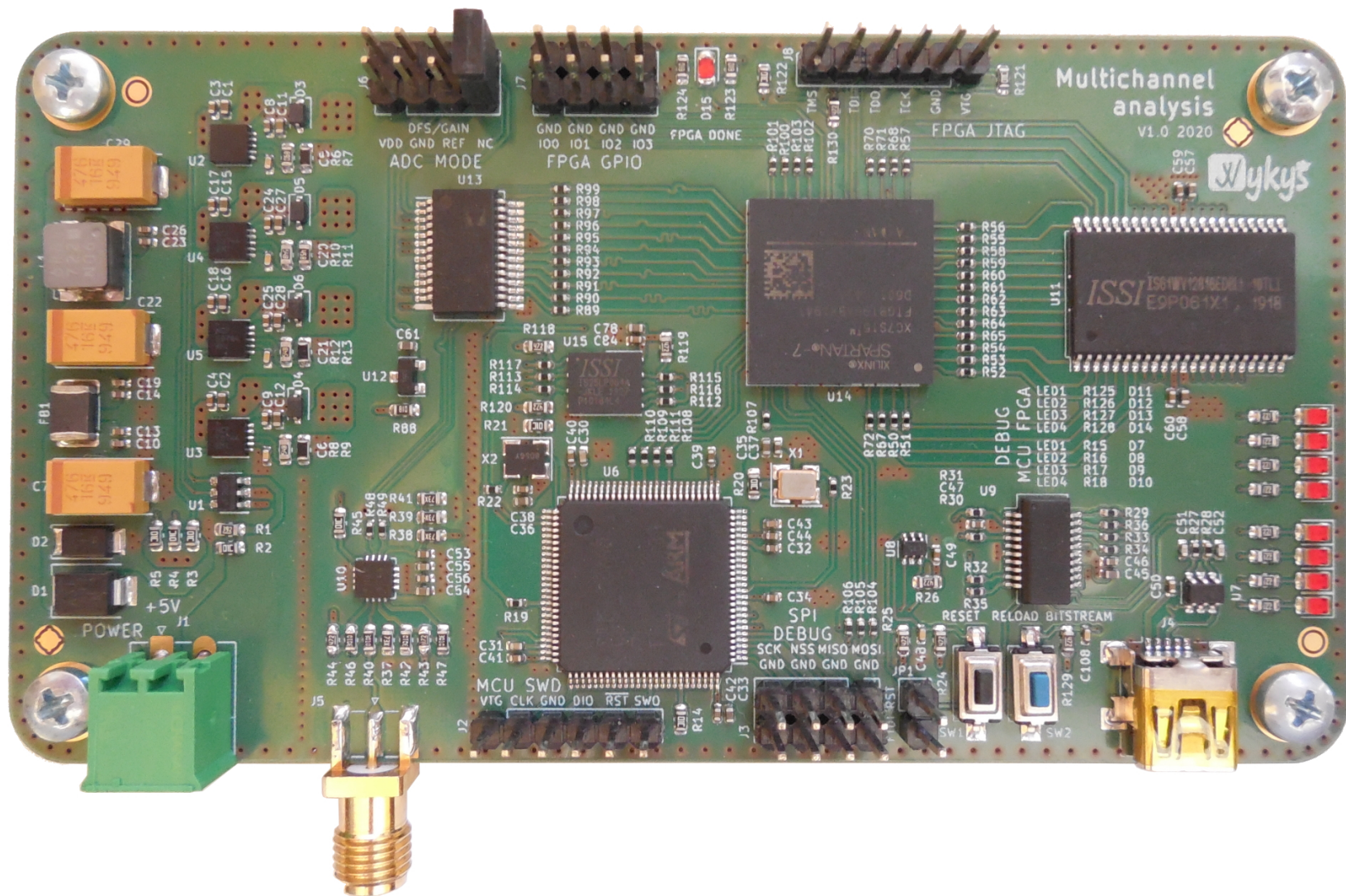


Obr. D.2: Blokové schéma bloku DSP v hradlovém poli.

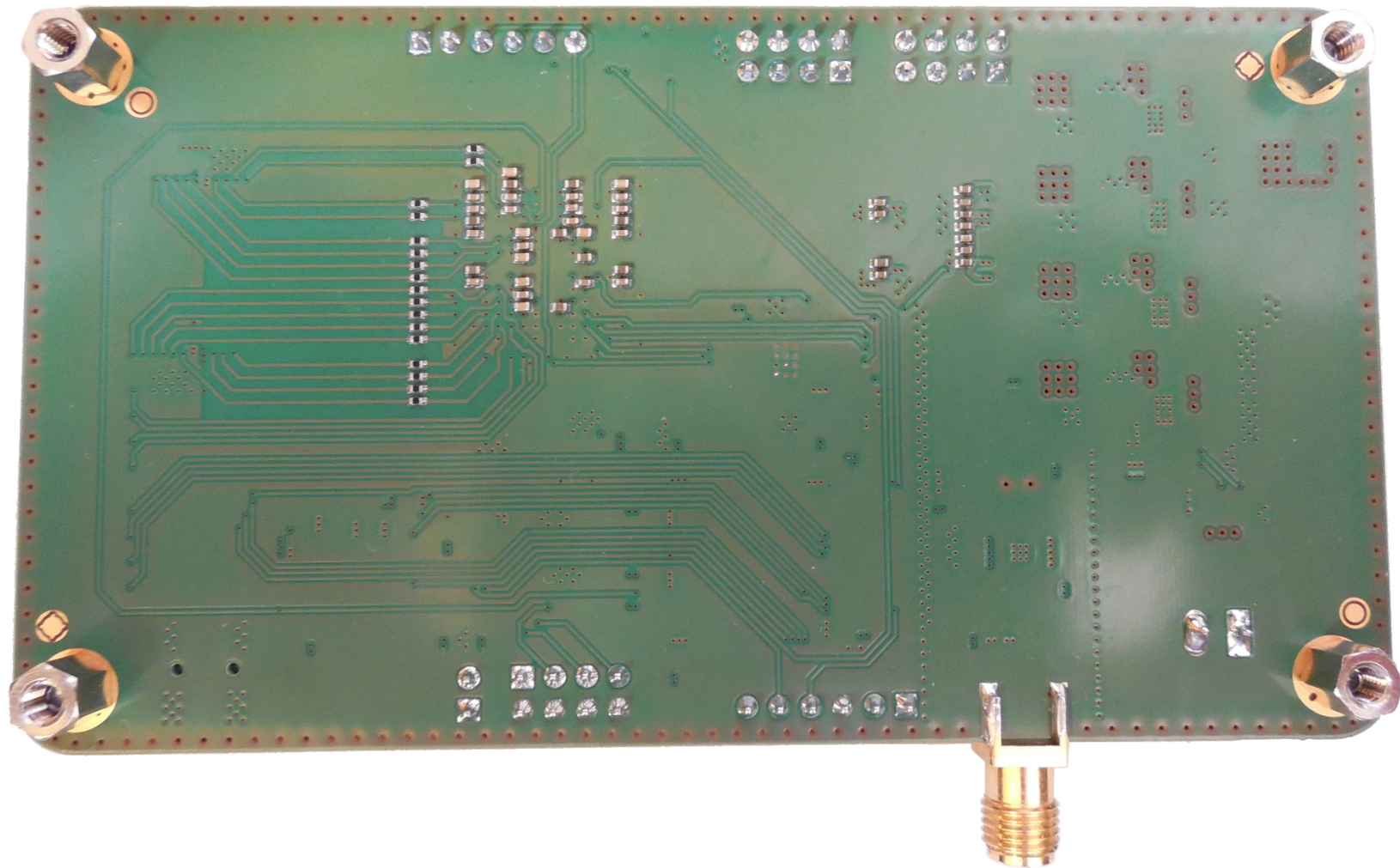
E Fotografie zařízení



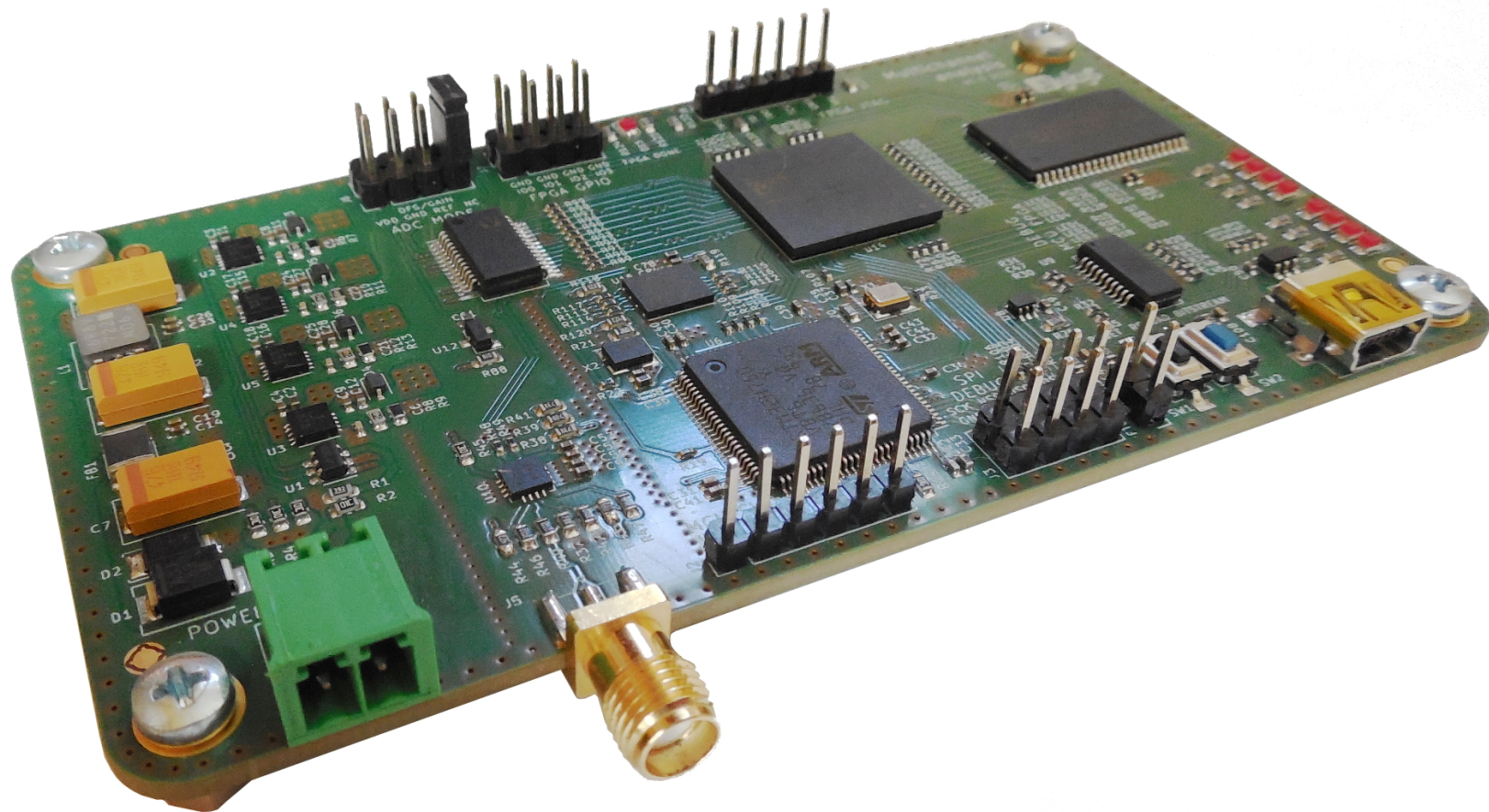
Obr. E.1: Fotografie prototypů modulů a vývojového kitu s FPGA.



Obr. E.2: Fotografie zařízení, pohled na stranu součástek.



Obr. E.3: Fotografie zařízení, pohled na stranu spojů.



Obr. E.4: Fotografie zařízení, pohled na konektory.

F Zdrojové soubory

Vzhledem k tomu, že velikost příloh je větší, než dovoluje informační systém, přikládám zde odkazy na jednotlivé git repozitáře hostované na <https://github.com>:

- [KiCad projekt](#)
- [Vivado VHDL projekt](#)
- [Firmware pro MCU](#)