



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

**Modulární výuková platforma pro oblast vestavěných
systémů a číslicových obvodů**

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Pavel Koupý

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Václav Šimek

BRNO 2021

Zadání diplomové práce



24116

Student: **Koupý Pavel, Bc.**
Program: Informační technologie
Obor: Počítačové a vestavěné systémy
Název: **Modulární výuková platforma pro oblast vestavěných systémů a číslicových obvodů**
Modular Educational Platform for Embedded Systems and Digital Circuits Domain
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se architekturou a obecnými principy použití rekonfigurovatelných obvodů FPGA a moderními mikrokontroléry na bázi jádra ARM Cortex-M.
2. Zpracujte stručný přehled aktuálně dostupných kitů vhodných pro výukové účely, jejichž součást tvoří mikrokontrolér nebo obvod FPGA.
3. Na základě poznatků z bodu 2) zadání navrhněte koncepci modulární výukové platformy tvořené dvěma samostatnými deskami. Klíčovým prvkem první z nich bude MCU s jádrem ARM Cortex-M, v druhém případě pak FPGA.
4. Výchozí koncepci z bodu 3) zadání doplňte o propojovací systém, který umožní vzájemnou kombinaci funkcionality obou desek.
5. Zvolte vhodné komponenty a následně vytvořte schéma zapojení, které na obvodové úrovni realizuje navrženou koncepci modulární výukové platformy.
6. V návrhovém editoru připravte motivy obou desek plošných spojů dle obvodového zapojení z bodu 5) zadání. Následně proveděte jejich fyzickou realizaci a oživení.
7. Implementujte základní obslužný firmware pro MCU s jádrem ARM Cortex-M a v jazyce VHDL základní hardwarovou infrastrukturu pro obvod FPGA.
8. Vhodným způsobem demonstrujte funkčnost výsledného řešení. Zhodnoťte dosažené výsledky, diskutujte možnosti dalšího rozvoje a případná vylepšení.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 5 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 19. května 2021

Datum schválení: 30. října 2020

Abstrakt

Cílem práce je návrh a realizace dvou desek plošných spojů realizující výukové platformy, které se budou sestávat ze dvou samostatných desek plošných spojů s ARM MCU a programovatelným hradlovým polem FPGA, které budou propojitelné a vhodně doplněné o periferie. Tyto platformy budou vytvořeny na základě rozboru aktuálních řešení výukových a vývojových plafotrem a následně demonstrovány na konkrétních příkladech. Hlavním přínosem této práce by mohla být obměna fakultního vybavení a jeho zjednodušení pro studenty. Zároveň je kladen důraz na větší transparentnost celého řešení, aby nebylo pro začínajícího studenta příliš komplikované se seznámit s moderními mikrokontroléry a programovatelnými hradlovými poli a mohl jednotlivé jednodušší celky propojovat do složitějších, kde lze jednotlivé desky použít jako samostatné fungující celky a jejich propojení poskytne výpočetně silnější a zároveň složitější zařízení pro pokročilého uživatele.

Abstract

The aim of the work is the design and implementation of two circuit boards delivering learning platforms, which will consist of two separate circuit boards with ARM MCU and a programmable FPGA gate array that will be interconnectable and appropriately complemented by peripherals. These platforms will be developed by analysing current teaching and development platform solutions and then demonstrating on practical examples. The main benefit of this work should be update and simplification of existing equipment. At the same time, there is an emphasis on greater transparency of the whole solution, so that it is not too complicated for an aspiring student to familiarise himself with modern micro-controllers and programmable gate arrays and can link the simpler units into more complex ones, where the individual boards can be used as separate working units and their interconnection will provide a computationally stronger yet more complex device.

Klíčová slova

Vývojová platforma, ARM Cortex-M, FPGA, Arduino kompatibilní, CMSIS-DAP integrované ladící rozhraní

Keywords

Development board, ARM Cortex-M, FPGA, Arduino compatible, CMSIS-DAP onboard debug interface

Citace

KOUPÝ, Pavel. *Modulární výuková platforma pro oblast vestavěných systémů a číslicových obvodů*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek.

Modulární výuková platforma pro oblast vestavěných systémů a číslicových obvodů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Koupý

30.6.2021

Poděkování

Rád bych poděkoval svému vedoucímu práce panu Ing. Václavu Šimkovi za odborný dohled, vedení, konzultace, cenné rady a zprostředkování součástek a desek plošných spojů. Déle bych rád poděkoval firmě CANLAB s.r.o. a všem jejím zaměstnancům a vedoucím, kteří mi poskytnuli prostředky a nástroje pro osazení a oživení desek plošných spojů a konkrétně Ing. Davidu Španělovi a Mgr. Vítězslavu Rejdovi za cenné rady při oživování desek a v neposlední řadě také Oldřichu Koprívovi za pomoc s osazením klíčových součástek.

Obsah

1 Úvod	4
2 Metodika návrhu desek plošných spojů	5
2.1 Návrh DPS	5
2.2 Realizace DPS.....	6
3 Architektura programovatelných hradlových polí	8
4 Architektura ARM Cortex-M	14
5 Přehled dostupných platforem	18
5.1 FITKit	18
5.1.1 FITkit 2.0	19
5.1.2 FITkit 3.0 – Minerva	19
5.2 Raspberry Pi.....	19
5.3 TE0726-03 (ZynqBerry).....	20
5.4 Spartan Edge Accelerator Board	20
5.5 Arty.....	21
5.6 FRDM-K64F.....	22
5.7 STM32F4VE.....	22
5.8 Arduino	23
6 Návrh a technická realizace platforem	25
6.1 Platforma s ARM MCU	26
6.1.1 Mikrokontrolér	26
6.1.2 Ladící a programovací obvod.....	26
6.1.3 Napájecí obvody	28
6.1.4 Periferie.....	28
6.1.5 Rozvržení vývodů DPS.....	29
6.2 Platforma s FPGA	31
6.2.1 Obvod FPGA.....	31
6.2.2 Ladící a konfigurační obvod	31
6.2.3 Napájecí obvod.....	32
6.2.4 Periferie.....	32
6.2.5 Rozvržení vývodů DPS.....	33
6.3 Propojení platforem.....	34
6.4 Vytvoření schémat zapojení	35

6.5 Metodika	35
6.6 Návrh motivů	36
6.6.1 Technologické postupy	36
6.6.2 Specifikace výrobce.....	36
6.6.3 Vrstvy a jejich složení.....	37
6.6.4 Prokovy	37
6.6.5 Rozmístění součástek.....	37
6.6.6 Routování	37
6.7 Výrobní podklady.....	39
7 Osazení desek plošných spojů a jejich oživení	40
8 Programové vybavení pro platformu s MCU	42
8.1 Vývojové prostředí	42
8.2 Základní popis prostředí	42
8.3 Vytvoření projektu.....	43
8.4 Překlad projektu	45
8.5 Nahrání aplikace.....	46
8.6 Ladění	47
8.6.1 OpenOCD.....	47
8.6.2 PyOCD.....	49
8.6.3 Firmware ladícího adaptéru	49
9 Programové vybavení pro platformu s FPGA	52
9.1 Vývojové prostředí	52
9.2 Základní popis prostředí	53
9.3 Vytvoření projektu.....	55
9.4 Simulace	55
9.5 Syntéza konfiguračního řetězce pro FPGA	56
9.6 Nahrání konfigurace	57
9.7 Ladění	59
10 Demonstrace	60
10.1 Příklad signalizační LED diody a tlačítka	61
10.1.1 Platforma s MCU.....	61
10.1.2 Platforma s FPGA.....	62
10.2 Příklad I ² C rozhraní pro konfiguraci FPGA	63
10.1.1 Platforma s MCU.....	63
10.1.2 Platforma s FPGA.....	64
10.3 Příklad s rozšiřujícím modulem	65

10.3.1 Platforma s MCU.....	66
10.3.2 Platforma s FPGA.....	67
10.4 Příklad ovládání servomotoru	68
10.4.1 Platforma s MCU.....	69
10.4.2 Platforma s FPGA.....	70
11 Závěr	71
Literatura	72
A Schéma zapojení platformy s MCU	76
B Schéma zapojení platformy s FPGA	83
C Obsah přiloženého paměťového média	89

Kapitola 1

Úvod

Modulární výuková platforma, tedy v tomto případně dvě samostatné platformy jedna s mikrokontrolérem s ARM Cortex-M a druhá s programovatelným hradlovým polem nebo též FPGA, které bude možné propojit za účelem programování FPGA za pomocí MCU a případně další komunikace jako například výměna výsledků nebo stav řízení stavového automatu modelovaného v FPGA. V rámci této práce se tak nejprve zmíníme o technologických a návrhových postupech při vytváření schémat zapojení a návrhu motivů desek plošných spojů (DPS), tedy především ty postupy, které nám umožní následně navrhnout požadované platformy se základními principy fungování architektur FPGA čipů a mikrokontroléra ARM Cortex-M. Po teoretickém úvodu se podíváme na již existující konkrétní řešení a prozkoumáme jak existující komerční výukové platformy, tak platformy navržené na naši fakultě. Komerčních řešení je nepřeberné množství a uvedené desky jsou tak jen úzkým průřezem těch nejpopulárnějších platforem. Výukové platformy se zabudovaným MCU nebo FPGA navržených na naši fakultě je také velké množství, ale zmíníme především platformu FITkit[30]. Po získání přehledu dostupných řešení a prozkoumání jejich schématických zapojení se pokusíme navrhnout vlastní platformu podobnou FITkitu, avšak odděleně desku plošných spojů s FPGA a desku s mikrokontrolérem včetně možnosti jejich dodatečného propojení. V rámci návrhu se budeme věnovat konkrétním součástkám a budeme chtít vytvořit samostatně fungující desku s mikrokontrolérem případně FPGA jejichž součástí bude též ladící a programovací adaptér, který umožní automatizované programování a ladění za pomocí vývojových prostředí dodávaných výrobci daných součástek. V rámci návrhu se budeme v této práci zabývat také samotnou výrobou, osazením a oživením obou platforem. Okrajově si shrneme problémy návrhu motivu desek plošných spojů a následně si něco řekneme i k osazení a oživení. Dle vytvořeného návrh můžeme pracovat s již konkrétními nástroji pro vývoj na obou platformách a nechybí tak ani úvod do vývojových prostředí k těmto platformám. Na konec si ukážeme jednoduché demonstrační příklady práce s platformami, abychom tak ověřily funkčnost a demonstrovali práci s platformami.

Práce je tedy dělena na několik kapitol zabývajících se rozbořem metodik návrhu desek plošných spojů (Kapitola 2), základní architekturou FPGA (Kapitola 3) a mikrokontroléru s jádrem ARM Cortex-M a rozbořem aktuálních řešení, který najdeme v kapitole 5. Následuje kapitola 6 návrh konceptu platforem s ARM MCU a FPGA. Další podkapitoly se věnují vlastnímu schématickému návrhu, přípravě motivu DPS a výrobní dokumentaci. Kapitola 7 se zaobírá fyzickou realizaci DPS, osazením a oživení. Následuje představení nástrojů v kapitolách 8 a 9. Práce je zakončena demonstrací v kapitole 10 a závěrem v kapitole 11, který sumarizuje práci a věnuje se nedostatkům, možným vylepšením a nápadům k dalšímu vývoji.

Kapitola 2

Metodika návrhu desek plošných spojů

V kontextu této práce stručně shrneme zásady a obecnou metodiku spojenou s návrhem schémat obvodů, a výrobou desky plošných spojů (dále jen DPS). Níže jsou uvedeny jen základní pojmy a definice pro zorientování v tématu a budeme se soustředit především na desky plošných spojů pro povrchovou montáž vývodových i nevývodových součástek (dále jen SMD), protože tento typ desky by měl být konečným výsledkem praktické části této práce. Těchto pouček a metodiky se poté drží celá tato práce. Metodiky nejsou strikně definované a vychází z předmětu NAV [1], dokumentace k programu Altium Designer [26][27] a prací studentů [3].

2.1 Návrh DPS

Návrh DPS pomocí softwarového návrháře lze shrnout do šesti kroků a jedná se o iterativní proces, kdy každý další krok stavíme na tom předešlém. Tato metodika je inspirována generickým pracovním postupem návrhu v populárních softwarových nástrojích jako Eagle, Altium Designer, či Pads.

První krok je hrubá idea funkcionality požadovaného obvodu a identifikace jednotlivých obvodových celků. Též se zabýváme výběrem klíčových součástí systému jako například mikrokontrolér či napájecí zdroj a pak tuto hrubou ideu obvodu lze znázornit pomocí blokového schématu s vybranými klíčovými součástkami jako jednotlivými bloky.

Druhý krok je převod blokového schématu na konkrétní schéma zapojení na obvodové úrovni. V tomto kroku je nutné specifikovat zbylé součástky systému a jejich zapojení. Na této úrovni lze provést simulaci jednoduchých částí obvodu jako je například napájení. Je vhodné se též zmínit o přístupu k schématickému návrhu obvodu při dělení DPS na více schémat. Přístupy jsou dva a to plochý (flat) návrh nebo návrh hierarchický. Oba přístupy budou blíže vysvětleny v Kapitola 6.

Třetí krok se týká návrhu motivu desky plošných spojů a rozmístění součástek na desce nebo též layout desky a vedení spojů. V případě složitějších desek je vhodná simulace chování jednotlivých signálů, většina dostupných aplikací pro návrh umožňuje určitý stupeň simulace obvodu či jeho validaci. V této části neexistuje jedna správná nebo přímá cesta návrhu desky, tedy převodu z modelu schématu na model motivu DPS, ale je možné se držet obecně daných pouček vycházejících z elektromagnetické kompatibility obvodu nebo postupy či příklady zapojení vycházející z konkrétních technických listů k daným součástkám[5][15].

Tyto tři kroky jsou zásadní k výrobě plošného spoje, další tři kroky jsou důležité z pohledu vyrobiteľnosti, kdy se chceme přesvědčit a ověřit si funkčnost našeho návrhu a následně danou desku realizovat, osadit a oživit. Tedy kroky jsou následující:

Ve čtvrtém kroku ověříme rozměrovou kompatibilitu, provedeme poslední kontrolu součástek a jejich propojení a pokud máme vytvořené 3D modely pro součástky můžeme si výsledný obvod vizualizovat s osazenými součástkami a provést vizuální kontrolu.

Pátý krok je vytvořením dokumentace pro výrobu. Tyto dokumenty generované obvykle aplikací pro návrh, umožňuje vytvořit soubory, které lze poslat komerčnímu výrobcovi DPS a on dle těchto podkladů vytvoří nabídku na výrobu desky nebo je použít pro vlastnoruční výrobu spoje. Aplikačně vytvořená dokumentace obvykle obsahuje motiv desky, pokovené

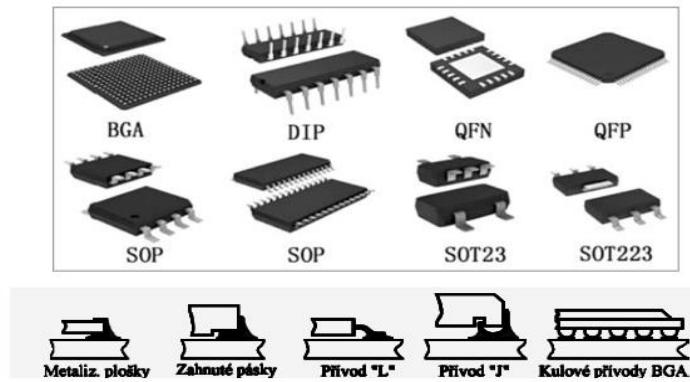
otvory, povrchové úpravy a také například tabulky se součástkami a odkazy na zvolené dodavatele součástek, je tak možné objednat vše potřebné od jednoho dodavatele v jedné objednávce, a to velmi zjednoduší a zrychluje výrobu konečného produktu. Šestý krok zahrnuje osazení, zapájení jednotlivých součástek, očistění a lakování. Následuje oživení a ověření funkcionality desky.

2.2 Realizace DPS

Vedle metodiky pro návrh schématu existují pravidla pro umístění součástek v závislosti na EMC a logickém uspořádání do celků. Též se používají krom základní jednotek pro délku dle soustavy SI i angloamerickou soustavu[7]. Nejčastěji používané jednotky[17] z obou těchto soustav jsou:

- 1 mil = 0,025 mm = 0,001 palců
- 1 mm = 39,37 mil
- 1 palec = 2,54 cm

Typy pouzder a vývodů součástek pro povrchovou, ty nejpoužívanější jsou vyznačeny na obrázku 2.1.V rámci práce jsou použity především pouzdra QFP, QFN a SOP.



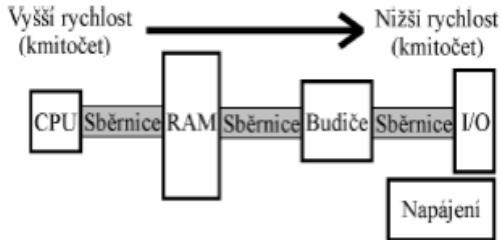
Obrázek 2.1: Dostupné typy pouzder a vývodů

Třídy přesnosti definují základní parametry objektů na plošných spojích, objektem je myšleno vodivá cesta, pájecí ploška či díry v plošném spoji a k nim se vztahující parametry jako minimální šířka spojů, minimální průměr vrtaného otvoru a další. Tyto hodnoty obvykle udává zvolený výrobce desky.

Elektromagnetická kompatibilita (EMC) je upravována především normami z rady CSN EN 61000 či CSN IEC 1000 [4]. Nicméně v této práci se zabýváme tvorbou výukové platformy do laboratorního prostředí, je tedy možné částečně přehlédnout elektromagnetickou kompatibilitu, ale zase je třeba dodržet určitá základní pravidla, aby deska byla plně funkční a vlastní rušení osazených součástek nebylo omezující pro chod jiných součástek na desce. Základní pravidla jsou následující:

- Minimalizace hodnot proudu – volba vhodných obvodů z hlediska impedance
- Minimalizace proudových smyček čili délky spojů – vhodné rozmístění komponent, zemnění, napájení a správné blokování napájení pomocí kondenzátorů.
- Minimalizace kmitočtového spektra – nevolit součástky se zbytečně vysokým kmitočtem
- Filtrace a ochrany napájení a I/O svorek – ochrana před ESD a přechodovými jevy
- Stínění – potlačení vyzařování a rušení

Rozmístění součástek se odvíjí od kmitočtu součástek a je vhodné oddělit co největší vzdáleností součástky s vysokým a nízkým kmitočtem.

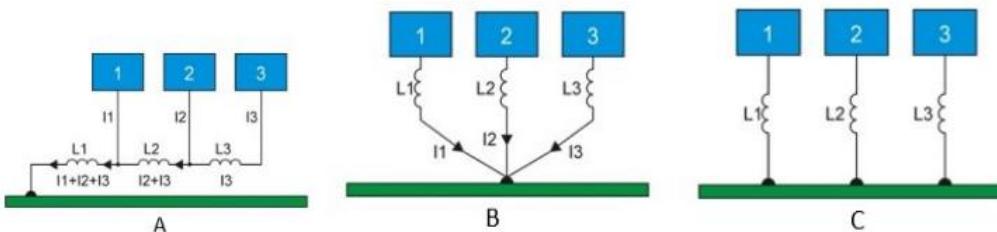


Obrázek 2.2: Umístění komponent vzhledem k EMC¹

V rámci metodiky návrhu je vhodné si detailněji popsát možnosti napájení obvodů plošných spojů. Základními prvky v této oblasti jsou regulátory a stabilizátory napětí, které převádí vstupní napětí napájecího konektoru na vhodná napětí pro danou aplikaci. Existují již hotová řešení na čipu, který tak poskytuje kompletní správu napájení [33]. Takový obvod je možné obvykle programovat, či různě měnit chování napěťových výstupů. Toto řešení je vhodné u složitějších desek plošných spojů, kde je nutné vést více napěťových vedení s různými parametry.

Při zemnění obvodu rozlišujeme dva základní druhy připojení zemnících vývodů¹ součástek ke společnému potenciálu:

- Jednobodové, lze dělit dále na sériové a paralelní
- Vícebodové



Obrázek 2.3: Způsoby zemnění obvodu¹

Vícebodové zemnění je vhodné pro vysokofrekvenční aplikace jako je i číslicový obvod, kde je vhodné cestu od zemnícího pinu k vodivé ploše společného potenciálu udělat co nejkratší. Stejně tak postupujeme i napájecích cest. Důležitým prvkem pro ochranu obvodu jsou blokovací kondenzátory. Ty slouží jako ochrana před proudovými špičkami v obvodu a vysokofrekvenčním rušením. Existují tři druhy blokovacích kondenzátorů [6]:

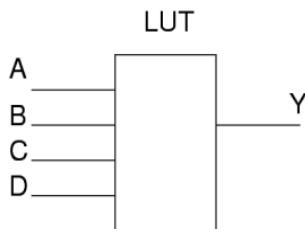
- **Filtracní** (Bypassing) – slouží jako širokopásmový filtr, který snižuje vliv indukčnosti přívodů od napájecího zdroje, je-li součástí desky.
- **Lokální** (Decoupling) – slouží jako lokální krátkodobý zdroj energie a redukuje impulsní výkyvy proudů, které by jinak ovlivnily celou desku. Umisťují se co nejbliže integrovanému obvodu.
- **Skupinové** (Bulk) – slouží jako zdroj energie pro současně nabíjení několika kapacitních zátěží.

¹ https://moodle.fel.cvut.cz/pluginfile.php/92743/mod_resource/content/3/Navrhova_pravidla.pdf

Kapitola 3

Architektura programovatelných hradlových polí

FPGA (Field Programable Gate Array) je polovodičová technologie [2], která se sestává převážně z pole konfigurovatelných logických buněk (CLB) a vestavěných bloků jako je Block RAM, násobička či ARM procesor. Funkce je dána vnitřní konfigurací uložené obvykle v SRAM či flash paměti. Tuto konfiguraci lze libovolně měnit a je možná i částečná rekonfigurace části čipu. CLB buňky se dále dělí na menší logické elementy zvané SLICE. Element SLICE se skládá z funkčního generátoru jako je LUT, Block RAM nebo SRL. Element obsahuje též registry/latch, multiplexory, „carry“ logiku a další pomocnou logiku



Obrázek 3.1 LUT schématická značení

pro aritmetiku. LUT neboli Look-Up Table je základní logické hradlo potažmo funkční generátor. Toto hradlo má N-bitový vstup a 1bitový výstup. Umožňuje generovat libovolnou binární funkci N proměnných. Je možné tedy realizovat jakékoli funkce vyhledáním výstupu pro příslušné vstupy v look-up tabulce.

Klasická FPGA jsou založena na paměti SRAM, kterou je možné libovolně přepisovat. Tato paměť je volatilní a je nutné při ztrátě napájení znova nahrávat obsah paměti, v tomto případě konfigurační řetězec. Pro programování SRAM či interní flash paměti je možné použít JTAG adaptér a na desce pouze vyvést JTAG konektor pro připojení. Tento způsob programování je obvykle sdílen napříč všemi výrobci FPGA čipů, ale konkrétních možnostech si povíme v dalších kapitolách.

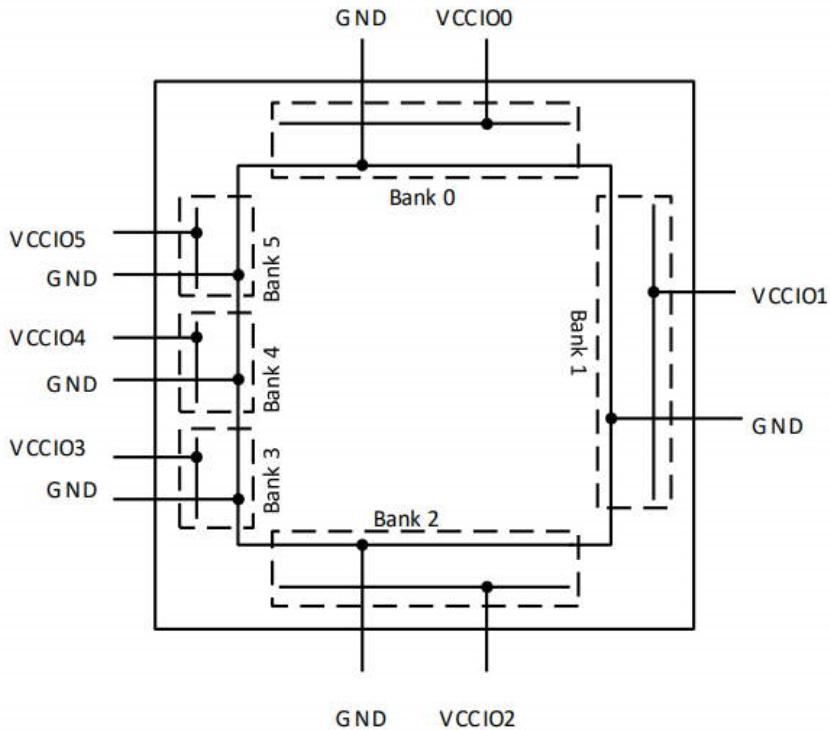
Dále budeme popisovat specificky architekturu FPGA ze série MachXO3D [5] od firmy Lattice Semiconductor, ale opět se bude jednat o nějaké obecné principy a součásti či jejich obměny najdeme jistě u všech FPGA čipů, jen má pro ně každý výrobce své vlastní pojmenování.

Čipy ze série MachXO3D obsahují pole logických bloků obklopených programovatelnými I/O bloky. Navíc tato série čipů obsahuje také sysCLOCK PLL, které jsou schopny generovat různé hodinové signály, sysMEM Embedded Block RAM nebo také EBR bloky, specializovaný bezpečnostní blok ESB též Embedded Security Block poskytující například šifrovací algoritmus AES 128/256, generátor náhodných čísel nebo modul pro elliptickou kryptografii. Kromě různých funkčních bloků poskytují tyto čipy též I²C a SPI kontroléry v rámci Embedded Functional Blocks (EFB)[9].

Logickými bloky, ze kterých se skládá FPGA máme na mysli PFU nebo též Programmable Functional Unit. Ty jsou spolu s EBR bloky řazeny do dvourozměrné mřížky po sloupcích a řádcích. Každý řádek obsahuje logické nebo EBR bloky. Na okrajích čipu najdeme do sysIO Bank řazené PIO buňky určené pro ovládání vstupních/výstupních vývodů

čipu. Krom tohoto pole bloků je na čipu také interní flash paměť a zmiňovaný bezpečnostní blok či blok pro generování hodinového signálu.

PFU bloky jsou základem pro logické, aritmetické operátory dále pak pro RAM, ROM a registry. Každý PFU blok se sestává ze čtyř vnitřně propojených slice elementů. Každý slice obsahuje dvě LUT a dva registry a celkově je z každého PFU bloku vede 25 výstupů a má 53 vstupů. PIO (Programmable Input/Output) používá pro svou funkci I/O buffer v rámci této série označované jako sysI/O buffer, ten podporuje mnoho standardů pro klasická či diferenciální rozhraní. Některé standardy pro I/O bufferu je možné vidět v tabulce níže.



Obrázek 3.2: Schéma MachXO3D čipů z pohledu Bank a jejich napájení (z.: [5])

O slice elementech jsme se již obecně zmínili, dále si uvedeme detailněji jak vypadá jednotlivý slice element a jaké jsou operační režimy jednotlivých elementů. Registry uvnitř elementu lze nakonfigurovat na citlivost na kladnou/zápornou logickou úroveň či hranu hodinového signálu. Každý slice obsahuje 15 routovacích vstupů, a navíc jeden pro interní vstup (FCI) z vedlejší PFU. Stejně tak pro výstup máme 5 výstupů, a navíc jeden pro interní výstup (FCO) do další PFU. Operační režimy jsou Logic, Ripple, RAM a ROM.

V režim Logic jsou LUT v elementech nakonfigurovány jako 4vstupé kombinatorické look-up tabulky v rámci jednoho elementu je možné realizovat až 5vstupé tabulky nebo též LUT5 a další více vstupové tabulky je možné vytvořit pomocí více slice elementů.

Režim Ripple poskytuje implementaci aritmetických funkcí. Jsou dostupné následující funkce:

- 2bitové sčítání a odčítání
- 2bitová inkrementace a dekrementace
- Různé čítače
- Násobička
- Komparátory

V tomto režimu je podporována aritmetika s přenosem z vyššího řádu pomocí zmíněných

vnitřních FCI a FCO přípojů.

RAM režim poskytuje 16x4bitový distribuovaný single RAM port (SPR). Každý PFU může poskytnout první dva slice elementy jako 16x1bitovou paměť, třetí element v PFU je použit pro adresové a kontrolní signály.

ROM režim poskytuje před programovanou paměť, data jsou nahrávána při konfiguraci PFU.

Hodinová distribuční síť obsahuje osm hodinových vstupů (PCLK [T, C] [Bank]_[2..0]). Tyto hodinové vstupy, některé z nich diferenciální, mohou řídit hodinové signály uvnitř čipu a nejsou-li využity pro vstup hodin mohou sloužit jako obecné I/O piny. Hodinové signály jsou rozvedeny skrze celý čip a jsou zdrojem hodin pro PFU, EBR.

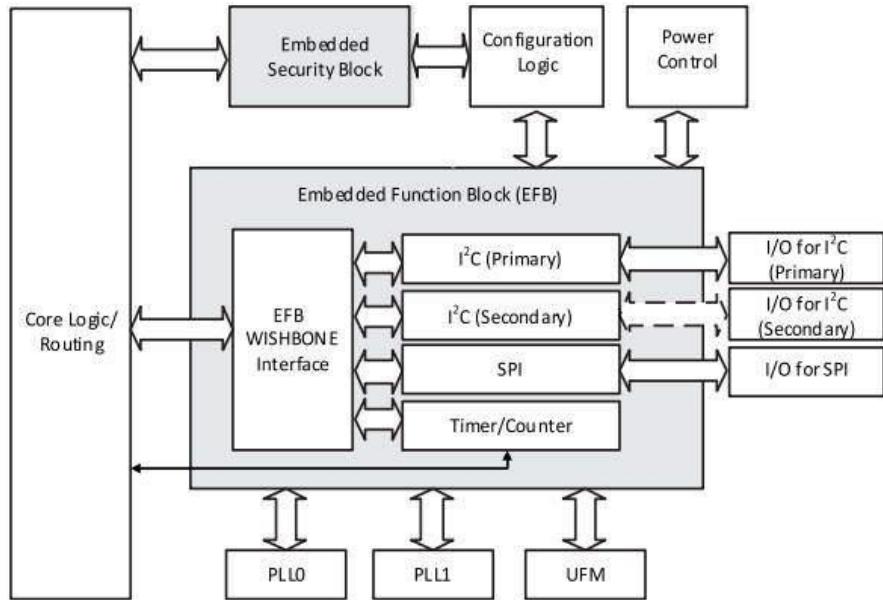
	V_{CCIO}				
Standard	3,3 V	2,5 V	1,8 V	1,5 V	1,2 V
LVTTL	●	—	—	—	○
LVCMOS33	●	—	—	—	○
LVCMOS25	—	●	—	—	○
I3C33	●	○	○	○	○
LVDS	●	●	○	○	○
MIPI	●	●	○	○	○
— - omezená funkcionalita					

Tabulka 3.1: Podporované standardy I/O bufferu a standardem podporované napětí

Další vestavěné bloky jsou poskytovány jako tzv. pevně vestavěné funkce (Hardened Control Functions). U této série čipů je to I²C, SPI kontrolér, časovače/čítače a User Flash Memory (UFM) spadající pod zmíněné EFB a také bezpečnostní ESB moduly. Tyto bloky jsou přístupné skrze Wishbone sběrnici a dále si ukážeme jejich blokové schéma, které si také krátce popíšeme.

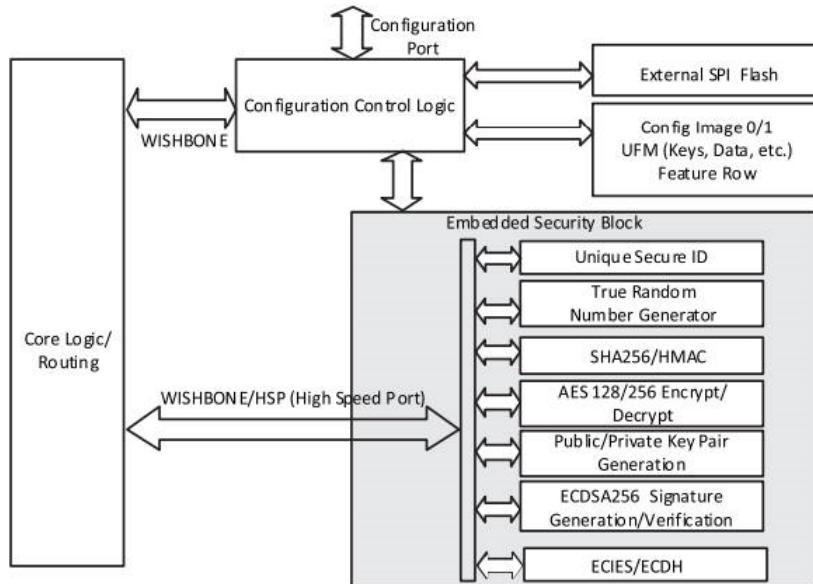
Wishbone je open source počítačová sběrnice a je navržena pro komunikaci jednotlivých a plně nezávislých IP bloků[38] v různých topologiích. Nejčastěji jsou to topologie typu křížový přepínač, sdílená sběrnice či zapojení za sebe tzv. data flow. Standard pro Wishbone sběrnici poskytuje především popis logických signálů a nespecifikuje tedy topologii nebo hodinový signál či logické napěťové úrovně. Tyto parametry jsou již záležitostí konkrétní implementace [14]. Uvedeme si též signály a způsob, kterým je možné propojit některý z IP bloků z uživatelem navrženým obvodem [23]. Na obrázku 3.6 vidíme dané signály pro wishbone sběrnici. Konkrétní popis jednotlivých signálů je možné najít v dokumentaci k vestavěným funkcím FPGA [9]. Pro připojení je tedy nutné implementovat v rámci uživatelského obvodu a logiky Wishbone Master kontrolér či stavový automat, který bude řídit sběrnici, adresovat odpovídající registr v EFB bloku a především uskutečňovat čtení či zápis.

EFB je funkční blok poskytující skrze Wishbone sběrnici dvě rozhraní I²C sběrnice, SPI sběrnici, časovače a čítače, přístup k UFM sektorům UFM0-3, a vlastně celé interní Flash paměti, tedy i CG0, CG1 a Feature Row.



Obrázek 3.3: *Embedded Function Block* blokové schéma (z.: [8])

Další dostupný blok je ESB s bezpečnostními prvky pro kryptografii a ověřování identity. Stejně jako ostatní funkční bloky je zpřístupněn také pomocí Wishbone sběrnice. Na obrázku



Obrázek 3.4: *Embedded Security Block* blokové schéma (z.: [8])

3.4 je možné vidět jednotlivé bezpečnostní moduly ESB bloku jako AES128/256, SHA256/HMAC a další.

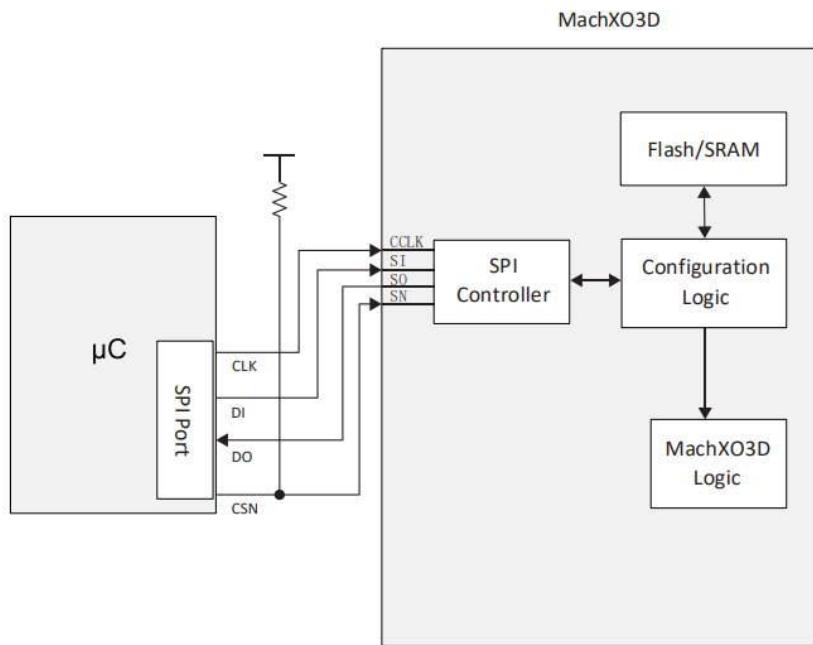
Pro nahrání konfiguračního řetězce existuje několik možností, jedna možnost pomocí JTAG adaptéra již byla naznačena. Dále existuje možnost konfigurace pomocí I²C a SPI

rozhraní, které si v této kapitole či kapitole s demonstrací přiblížíme blíže. Krom těchto zmíněných existují obecně tyto možnosti a jim odpovídající hw porty[8]:

- 1149.1 JTAG
- Self-download (interní flash paměť)
- Master/Slave SPI
- Dual Boot
- I²C
- WISHBONE sběrnice

K programování pomocí těchto rozhraní slouží sysConfig sada programovacích instrukcí. Tyto příkazy si přiblížíme v kapitole demonstrace, kde se budeme věnovat konfiguraci FPGA čipu právě pomocí platformy s mikrokontrolérem navržené v této práci.

Na obrázcích níže můžeme vidět zapojení SPI sběrnice, kde mikrokontrolér vykonává roli Mastera a FPGA roli Slave. Signály pro SPI sběrnici jsou CCLK (hodinový signál), SI (Master out/Slave in), SO (Master in/Slave out) a SN je povolovací signál pro SPI kontrolér. Dále si pomocí obrázku uvedeme složení vnitřní flash paměti, abychom se později mohli

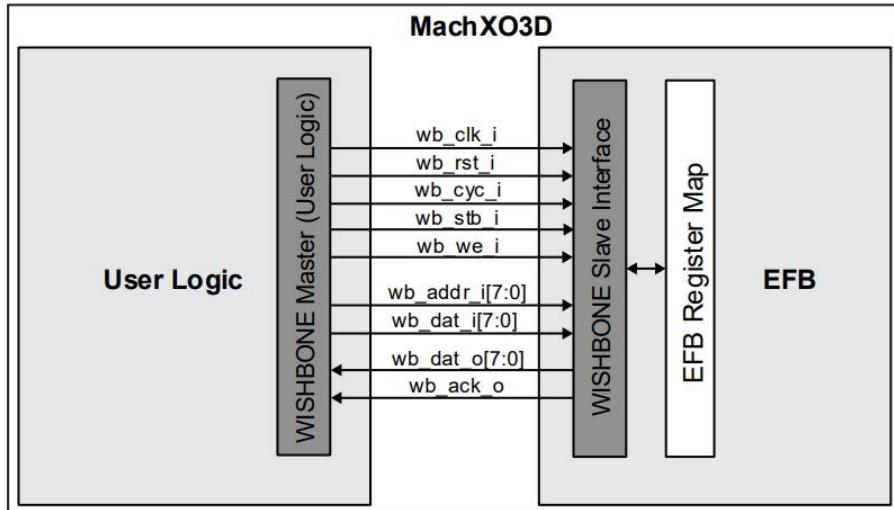


Obrázek 3.5: Příklad připojení Master-Slave pro konfiguraci skrze SPI sběrnici (z.: [8])

odkazovat na jednotlivé sektory.

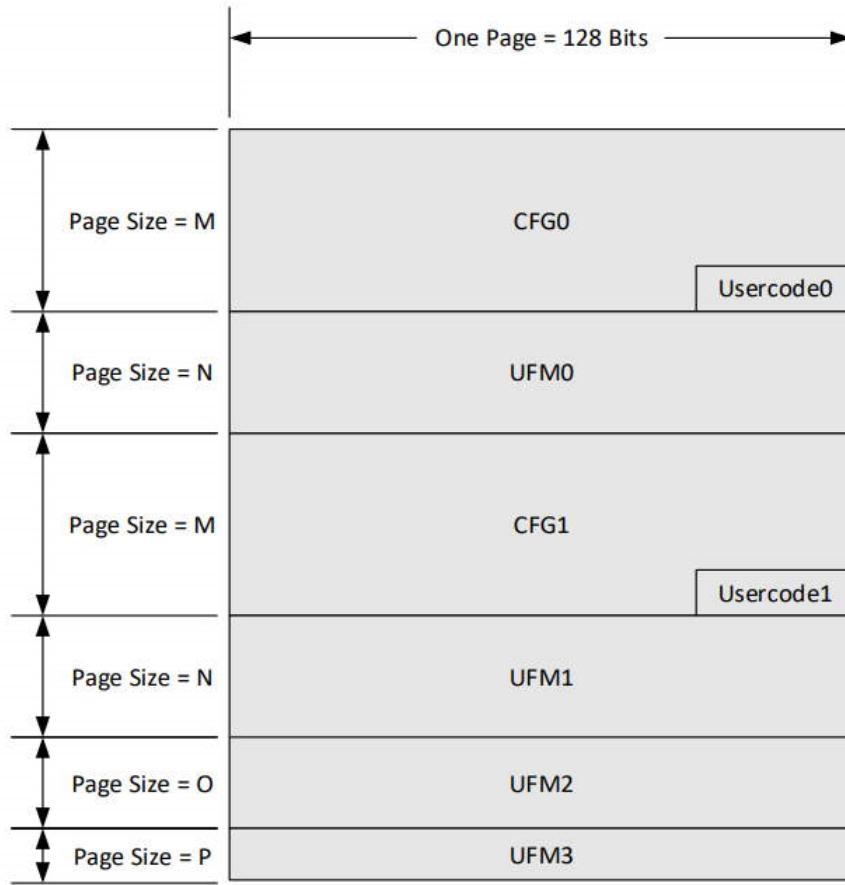
Interní flash paměť je dělena na konfigurační část CFGx a uživatelsky dostupnou paměť UFMx. CFG sektory obsahují primární a sekundární konfigurační řetězec. Velikost konfiguračního řetězce obvykle nepřekročuje velikost CFG sektorů, ale může se stát, že tuto velikost přeteče a pro tento případ jsou za CFG sektory umístěny bloky s uživatelskými sektory a do těchto a pouze těchto přímo sousedících sektorů můžou konfigurace CFG0 případně CFG1 přesáhnout. Schéma paměti je na obrázku 3.7.

Feature row je samostatně programovatelná a mazatelná část interní flash paměti. Umožnuje kontrolovat dostupnost jednotlivých parametrů a napevnou vyvedených kontrolérů sběrnic či ladícího rozhraní a také dedikované konfigurační piny jako je PROGRAMN, INITN a DONE.



Obrázek 3.6: Schéma připojení EFB bloku k uživatelské logice (z.: [8])

K již částečně popsanému konfiguračnímu režimu FPGA patří také další režimy Offline, Transparent, User. Mezi těmito režimy je možné přecházet v rámci konfigurace, kdy je například nutné pro rekonfiguraci běžícího zařízení přejít z User režimu, do režimu Offline, aby byla zneplatněna aktuální logika a zápis do CFG neměnil běžící konfiguraci.



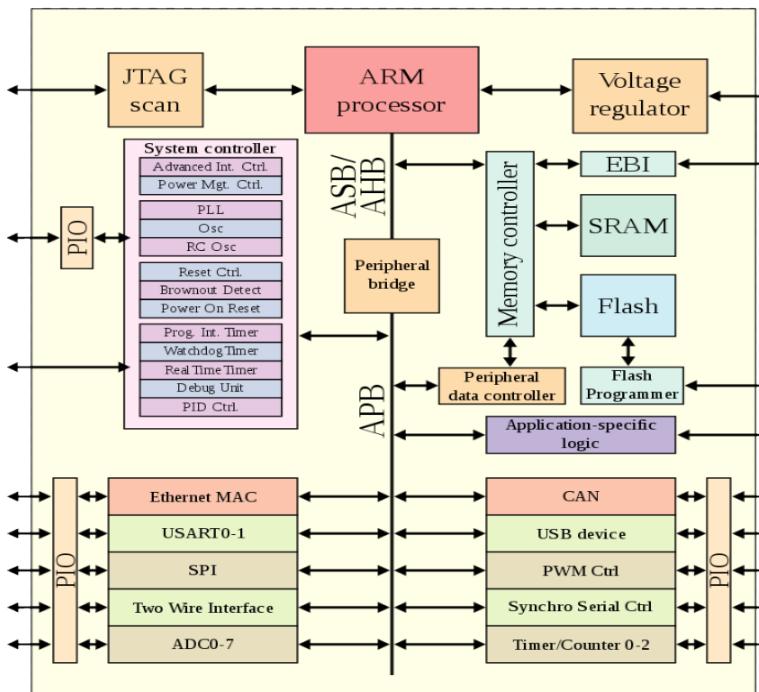
Obrázek 3.7: Mapa sektorů vnitřní flash paměti (z.: [8])

Kapitola 4

Architektura ARM Cortex-M

Obecně procesory ARM Cortex-M je rodina 32 a 64bitových RISCových procesorových jader s von Neumannovou či Harvardskou architekturou, registrovaných u společnosti Arm Holdings. Tyto procesory jsou cílené na použití v mikrokontrolérech, ASIC či FPGA. Firma Arm Holdings procesory jako takové již nevyrábí, ale soustřeďuje pouze na jejich návrh, aktuálně je dostupná verze ARMv8[10]. Takový to návrh si výrobce zakoupí a poté může ARM jádro integrovat do svého návrhu. Procesorové jádro lze simulovat i na FPGA, tzv. soft-core, návrh jádra ARM Cortex-M1 je dokonce uzpůsoben pro tento účel. Dalšími zástupci této rodiny procesorů jsou Cortex-M0, M0+, M3, M4 a M7. Mezi těmito zástupci lze najít shodné funkční bloky či oblasti, které nás budou zajímat i v rámci této práce. První tři oblasti boudou popsány podrobněji dále a poslední oblast bude zmíněna jen velice stručně, protože není pro tuto práci podstatný:

1. Základní programovací model
2. Řadič vektorů vnořených přerušení (Nested Vectored Interrupt Controller – NVIC)
3. Ladící a programovací techniky
4. Podporu OS funkcionalit a definice spánkových režimů



Obrázek 4.1: Blokové schéma Cortex-M jádra. (z.: [10])

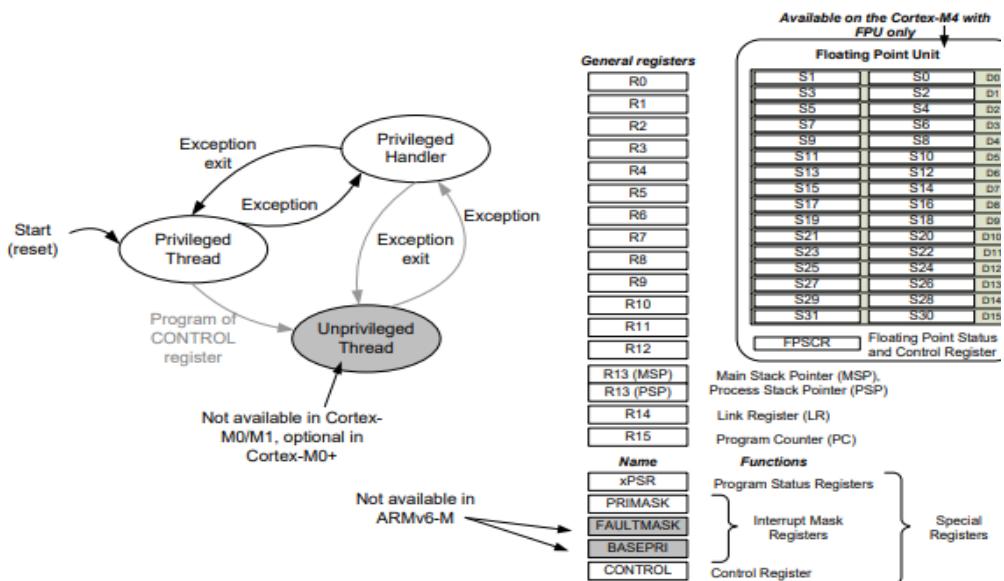
Instrukční sada se skládá z množiny instrukcí zvaných Thumb. Kompletní sada instrukcí je dosud obsažná a je dále rozšířena v novější technologií Thumb-2 a různé Cortex-M procesory podporují jen podmnožinu celé sady instrukcí.

AMBA je volně dostupný, a otevřený standard pro připojení a správu funkčních IP bloků v system-on-chip (SoC) modulech. Specifikace pro AMBA technologie je bezplatná a platformě nezávislá.

Standard se postupně rozrůstal o různé vylepšené sběrnice se specifickým určením. První verze tohoto standardu definovala sběrnice ASB pro propojení náročných komponent a APB pro připojení pomalejších periferií. Standard se od první verze vyvinul v početnou rodinu sběrnic, jako již zmiňované ASB nahradila sběrnice AHB a přibyly sběrnice AXI či její rozšíření ACE, které umožňuje tzv. big.LITTLE zpracování tedy spolupráci dvou jader, jednoho výkonnějšího a druhého úspornějšího, ale méně výkonného jádra.

Základní programovací model je silně konzistentní v rámci všech členů rodiny Cortex-M. Na obrázku 4.2 jsou vyobrazeny základní obecné registry R0-R12. Registry R13–R15 slouží jako ukazatele na zásobník, linkovací registr a programový čítač.

Speciální registry: stavový xPSR, registry k maskování přerušení PRIMASK, FAULTMASK, BASEPRI a kontrolní registr CONTROL. Registry pro práci v FPU S1 až S30 a stavový registr FPSCR jsou dostupné pouze na procesoru Cortex-M4 s FPU. Dále jsou na obrázku vyobrazeny operační režimy procesoru privilegovaný/neprivilegovaný Thread a Handler režim. Operační módy ukazují hlavní oepra4ní režim Thread a řešení výjimek

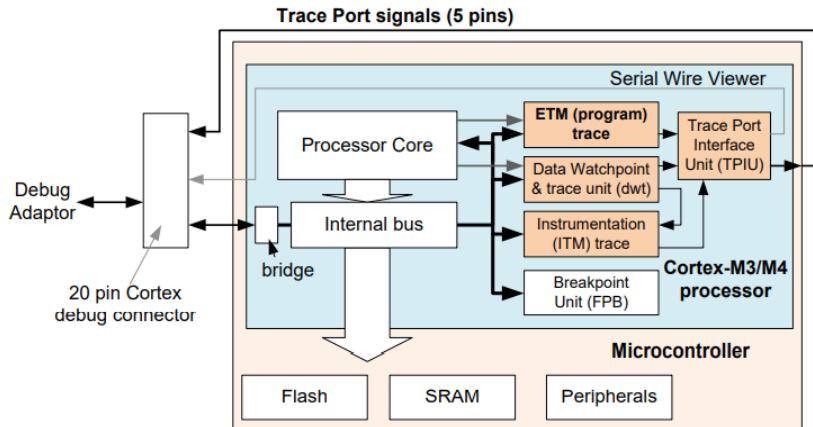


Obrázek 4.2: Základní registry a přechodový diagram operačních režimů
(z.: [10])

pomocí přechodu do módu Handler, kde je obsloužena výjimka a vykonávání se vrací do původního režimu Thread a to jak v privilegovém, tak i neprivilegovém režimu.[11]

Ladící a programovací techniky, které si popíšeme vychází z procesorů ARM jsou tedy založeny na technologii ARM CoreSight. Jedná se o vysoce škálovatelnou architekturu, které podporuje ladění více procesorových systému. Ladící konektor či adaptér umístěný přímo na desce s cílovým procesorem umožňuje:

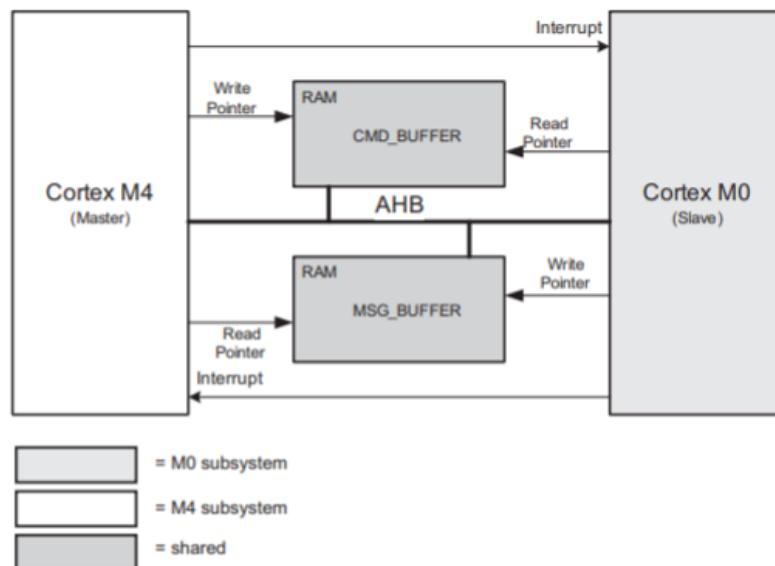
- Přístup k registrům kontrolujících ladění a trasování
- Přístup k paměti procesoru, a to i v případě běhu procesoru, tzv. přístup on-the-fly
- Přístup ke kontrolním registrům, a to i v případě zastavení procesoru
- Přístup k trasovacímu logu z MTB (Micro Trace Buffer)
- Programování Flash



Obrázek 4.3: Schéma propojení ladící a trasovací architektury ARM a konektoru (s 20 vývody), který je standardním řešením, když chceme vyvést i trasovací signály. (z.: [10])

Cortex-M procesory umožňují připojení pomocí JTAG protokolu nebo sériového protokolu SWD pro ladění. Trasováním ladící adaptér sbírá informace o vykonávání programu s minimálním zpožděním při jeho vykonávání. Dostupné trasovací informace jsou generovány z ETM, DWT či ITM jednotek. Informace zahrnuje informace o vykonávání programu, datové trasování, trasování událostí, profilování a další. Jak bylo zmíněno ladění nebo trasování je možné pomocí vyvedeného konektoru na desce plošných spojů a externím ladícím adaptérem nebo adaptérem integrovaným přímo na desce. V této práci se budeme zabývat především druhým řešením a dále v rámci návrhu budou zmíněny konkrétní adaptéry, které jsou volně dostupné formou open source projektů a umožňují jednoduchou integraci do navrhované platformy.

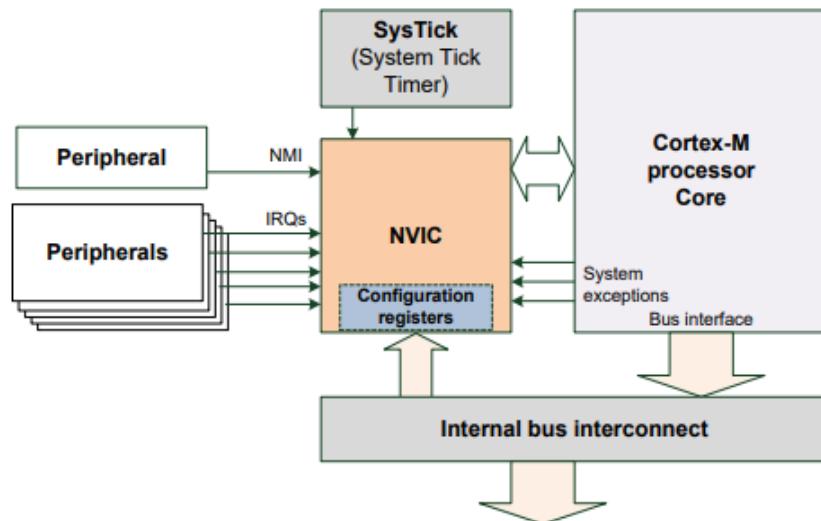
Řadič vektorů vnořených přerušení (NVIC) je řadič či kontrolér zpracovávající přerušení, takže jej dále alespoň stručně popíšeme. Všechny procesory ARM Cortex sdílí stejný model výjimek pro tento řadič, který umožňuje především konfigurovatelné



Obrázek 4.4: Schéma IPC Cortex-M jader mikrokontroléru LPC4327 skrze sdílenou RAM (z.: [5])

zpracování přerušení s podporou až 240 přerušení s 256 úrovněmi priority. Dále zpracování výjimek s nízkou latencí a zpracování přerušení a částečně správu napájení

Interní procesorovou komunikaci (IPC) jsme si naznačili výše a dále popíšeme podrobněji komunikaci mezi jednotlivými procesory v rámci série mikrokontroléru LPC43xx. Koprocesor a procesor v mikrokontroléru komunikují pomocí sdílené paměti a přerušení, oba procesory jsou připojeny k hlavní AHB matici. Komunikace mezi procesory v režimu Master-Slave, kde v roli master je hlavní procesor ARM Cortex-M4, dále je M4, začíná inicializací systému ARM Cortex-M0, dále jen M0, především fronty zpráv (MSG_BUFFER) a fronty příkazů (CMD_BUFFER). Fronta je definována čtyřmi registry: počáteční adresa fronty, adresa konce fronty, ukazatel čtení (Read Pointer) a ukazatel zápisu (Write Pointer). Komunikace probíhá vkládáním zpráv od M4 jádra do CMD_BUFFER na adresu ukazatele zápisu, který je následovně inkrementován a M0 na základě přerušení od M4 vyčte zapsaný příkaz a změní hodnotu registru čtecího ukazatele. Pří zapsání odpovědi do MSG_BUFFERu a po vyvolání přerušení od M0 vyčítá M4 zapsané výsledky, operace s ukazateli je obdobná jako při vkládání příkazu od M4 jádra. Na této úrovni nejsou žádné ochranné mechanismy, a proces vkládající zprávu do jedné z výše uvedených front se musí ve vlastní režii ujistit, že fronta není plná. V rámci komunikace není zajištěno ani žádné explicitní hlášení chyb a předpokládá se, že na příkaz od M4 odpoví M0 vždy[16].



Obrázek 4.5: Blokové schéma zapojení řadiče přerušení[10]

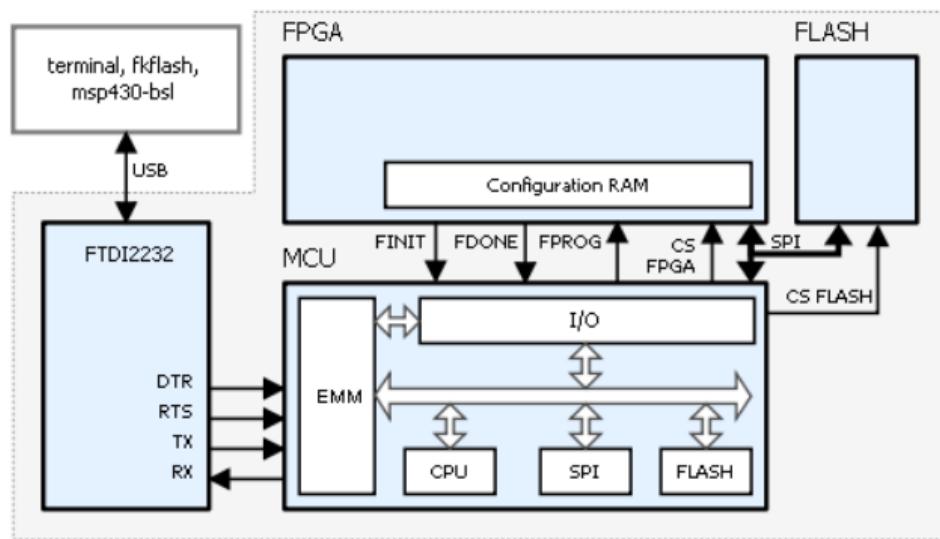
Kapitola 5

Přehled dostupných platforem

Pro lepší představu, jak by měla výsledná výuková platforma vypadat bude dále uveden stručný přehled nejpopulárnějších výukových a vývojových platforem. V rámci uvedených platforem jsou dostupná i schémata jejich zapojení a sloužily jako odrazový můstek pro návrh výukové platformy, kterou se zabývá tato práce.

5.1 FITKit

Platforma FITkit je interní výukovou platformou naší fakulty. Všechny verze kombinují MCU a FPGA. První verze platformy obsahuje 16bitový mikrokontrolér MSP430F168 a hradlové pole Spartan 3 XC3S50-4PQ208C, ke které je připojena DRAM paměť o velikosti 64 MB. Mikrokontrolér obsahuje vnitřní flash paměť o velikosti 48KB a 2 KB RAM paměti. Kromě kombinace FPGA a MCU nabízí platforma velké množství periferií již jako součást desky. Na desce plošných spojů najdeme PS2 konektor, RS232 konektor, audio rozhraní, VGA pro připojení displeje a sadu vstupně/výstupních pinů. K desce jsou též připojeny moduly klávesnice a řádkového LCD displeje. Většina periferií je připojena k FPGA, protože mají velké množství vývodů a jedná se především o výukovou platformu, který má uživateli pomoci pochopit FPGA technologii a obsluha periferií je vhodná pro výuku programování ve VHDL, kterým se v FPGA čipu popisuje obvod, který na něm bude realizován. Komunikace mezi MCU a FPGA se realizuje přes SPI. Existuje i několik revizí této platformy, které byly realizovány a studenti s nimi běžně pracují, a to revize 1.2 a 2.0[30].



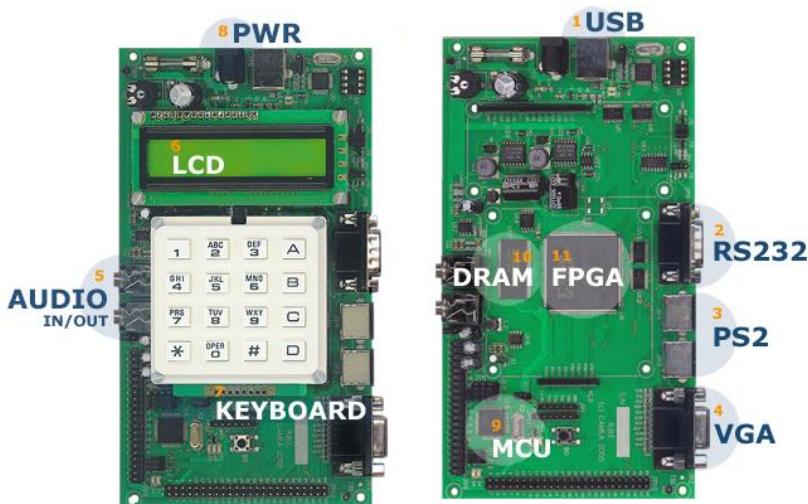
Obrázek 5.1: Schéma zapojení interních součástí FITkitu a propojení s PC (z.: [30])

Na obrázku výše je vyznačeno blokové schéma platformy FITkit. Blokové schéma napovídá, jak vypadá komunikace na desce. FPGA jsou spojeny pomocí SPI a signálových vodičů pro řízení programovaní, a by byla umožněna konfigurace z MCU. Jak FPGA, tak MCU lze programovat standardním způsobem přes JTAG konektor, ale je též umožněno programovat přes dvoukanálový USB převodník. Procesor přes USB převodník vystavuje svojí sériovou

linku, které se počítači jeví jako virtuální COM port nebo je možné použít knihovny FTD2XX pro komunikaci od výrobce převodníku.

5.1.1 FITkit 2.0

Novější verze platformy obsahuje výkonnější mikrokontrolér MSP430F2616, který obsahuje 92KB flash a 8 kB RAM. Revize 2.0 obsahuje nový konfigurovatelný kodek TLV320AIC23B a přibyla EEPROM paměť v FT232 obvodu pro identifikaci jednotlivých platforem. Krom samotné platformy, existují ještě přídavné moduly rozšiřující platformu o 10/100 Ethernet a bezdrátovou komunikaci v pásmu 2,4GHz[30].



Obrázek 5.2: Obrázek FITkitu s popisem součástí

5.1.2 FITkit 3.0 – Minerva

Tato platforma je posledním počinem naší fakulty a stal jednou z hlavních inspirací pro tuto práci. Platforma je založena na mikrokontroléru Freescale Kinetis K60 s jádrem ARM Cortex-M4, dále disponuje programovatelným hradlovým polem od firmy Xilinx typu Spartan-6. Oba čipy jsou doplněny o rozšiřující rozhraní. Mikrokontrolér obsluhuje Ethernet, USB I/O a slot micro-SD karty. Na hradlové pole je připojen HDMI výstup, který je vlastně na straně pole implementován jako DVI rozhraní s převodníkem na HDMI výstup. Dále je v k hradlovému poli připojen audio vstup/výstup. FPGA je tak vhodnější připojit ke komponentám s náročnějším požadavkem na přesun většího množství dat, případně akceleraci operací nad témito Audio/Video komponentami. Platforma je vybavena USB kontrolérem FT2232R. Napájení je řešeno integrovaným obvodem pro správu napájení TPS65251, který poskytuje při typické konfiguraci tři výstupy 3,3 V (2 A), 1,8V (2 A), 1.2 V (3 A). Vstupní napětí může být v rozmezí od 4,5 V do 18 V.

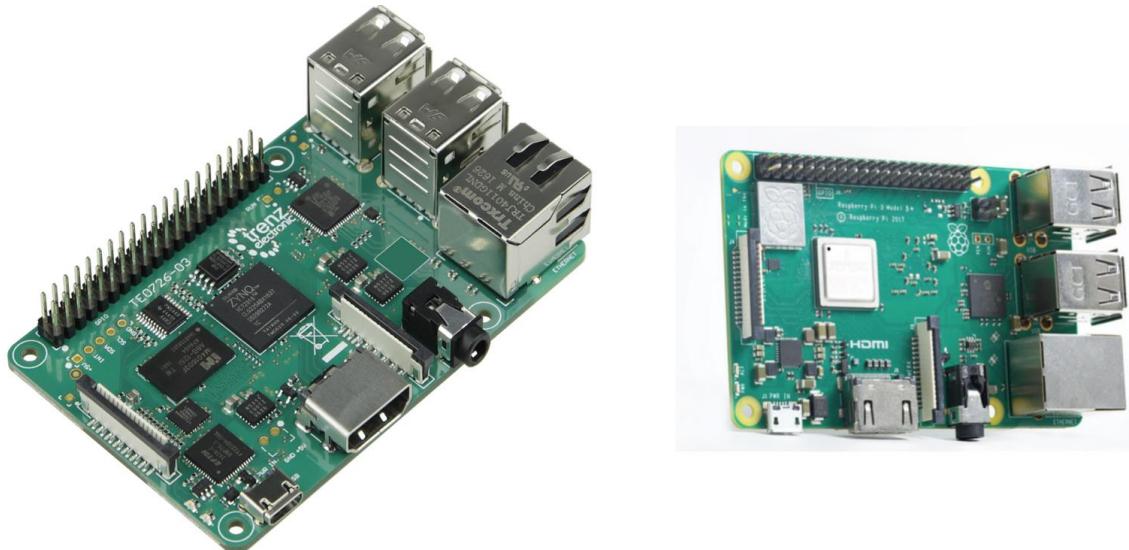
5.2 Raspberry Pi

Tato platforma[39] je vybavena již pokročilejším Cortex-A procesorem, které komplexitou přesahuje rámec této práce. Jedná se však pravděpodobně o jednu z nejpoužívanějších výukových a vývojových open source platforem. Existuje již čtvrtá verze této desky a mnoho dalších Raspberry PI platforem vyvinutých na základě této desky a daného procesoru, jsou to například model Zero, nebo výpočetní modul SoC, který lze pomocí DDR-SODIMM konektoru připojit do vlastní základní desky plošných spojů. Ve verzi 3 je na desce osazen

64bitový čtyř jádrový procesor BCM2837 pracující na frekvenci až 1,2 GHz od firmy Broadcom, ke kterému je připojeno 1 GB RAM. Jedná se tak vlastně o System-on-Chip řešení, protože na čipu je kromě CPU ARM Cortex-A53, resp. Cortex-A72 vyšších verzí Raspberry Pi verze 3, i GPU Videocore IV 400Mhz a mezi těmito dvěma jednotkami sdílená operační paměť typu SRAM. Deska poskytuje bezdrátovou komunikaci pomocí čipu BCM43438, dále je na desce obvod LAN9512, který slouží jako USB hub a Ethernet kontrolér. Krom těchto komunikačních rozhraní lze připojit k Raspberry Pi i HDMI nebo DSI displej. Deska poskytuje i audio výstup a připojení kamery přes CSI-2 sběrnici.

5.3 TE0726-03 (ZynqBerry)

Platforma TE0726-03² je vyrobena jako form-factor Raspberry Pi 2 a na první pohled nejsou odlišitelné, ale na této desce se místo čistě ARM procesoru nachází FPGA čip od firmy Xilinx. Čip Zynq-7010 kombinuje v jednou pouzdro dvoujádrový procesor ARM Cortex-A9 a programovatelné hradlové pole. Tato deska umožňuje běh linuxové distribuce obdobně jako Raspberry Pi, ale zároveň umožňuje určitý stupeň volnosti v podobě programovatelné logických obvodů, který Raspberry Pi chybí. Tato deska je jiným pohledem na kombinaci klasického procesového jádra ARM a FPGA. Na desce se nalézá 512 MB DDR3L SDRAM, 16 MB Flash paměť. Periferie na desce jsou obdobné jako na Raspberry Pi, tedy LAN9514 rozšiřuje desku o USB hub a 10/100 Ethernet, dále jsou na desce HDMI konektor, DSI konektor, CSI-2 konektor, klasický Raspberry Pi kompatibilní header a Micro-USB konektor



Obrázek 5.3: TE0726-03 (vlevo), Raspberry Pi 3 B+

umožňující nejen napájení, ale také sériovou komunikaci UART a ladění ARM procesoru a FPGA přes USB-JTAG přemostění.

5.4 Spartan Edge Accelerator Board

Tato deska je určena jako rozšiřující modul pro Arduino, ale funguje i samostatně. Jedná se o zajímavý koncept blízký žádanému řešení. Deska obsahuje FPGA XC7S15-1FTGB196C, které je možné použít samostatně nebo v kombinaci s Arduinem. Vedle FPGA obsahuje

² <https://docs.rs-online.com/e24d/0900766b8165dca1.pdf>

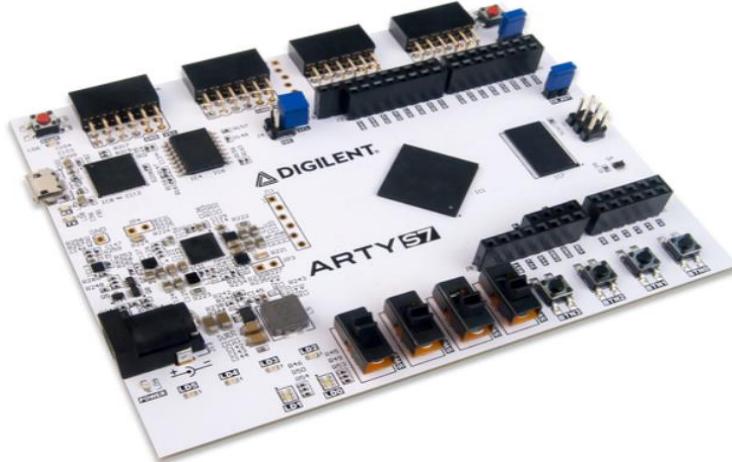
platforma také mikrokontrolér ESP32 vhodný pro IoT a bezdrátové aplikace.



Obrázek 5.4: Spartan ECB modul

5.5 Arty

Další z řady vývojových desek s Arduino kompatibilními vývody od firmy Diligent. Konkrétně deska Arty S7 je osazena FPGA XC7S50-1CSGA324C od Xilinx. Existuje několik verzí této desky s jiným typem FPGA nebo případně i verze s čipem řady Zynq od Xilinx. Deska Arty S7 je určena k připojení velkého množství periferií pomocí 4 konektorů Pmod s 2x6 piny na boku desky. Další verze desky obsahují například HDMI či Ethernet



Obrázek 5.5: Arty S7

rozhraní. Osazena je kromě Quad-SPI flash paměti také rychlá DDR3L paměť s různou velikostí. Programování je možné z Quad-SPI Flash případně může být FPGA programováno za pomocí USB-JTAG převodníku přes FTDI FT2232HQ. Napájení platformy je možné přes USB a pro energeticky náročnější periferie připojené přes Pmod konektor je možné použít externího stejnosměrného napájení o napětí 7-15 V poskytující proud alespoň 1 A, zde zmíníme programovatelný obvod pro správu napětí DA9062, který obsahuje čtyři snižovače napětí (buck-converter) a 4 regulátory napětí s nízkým úbytkem

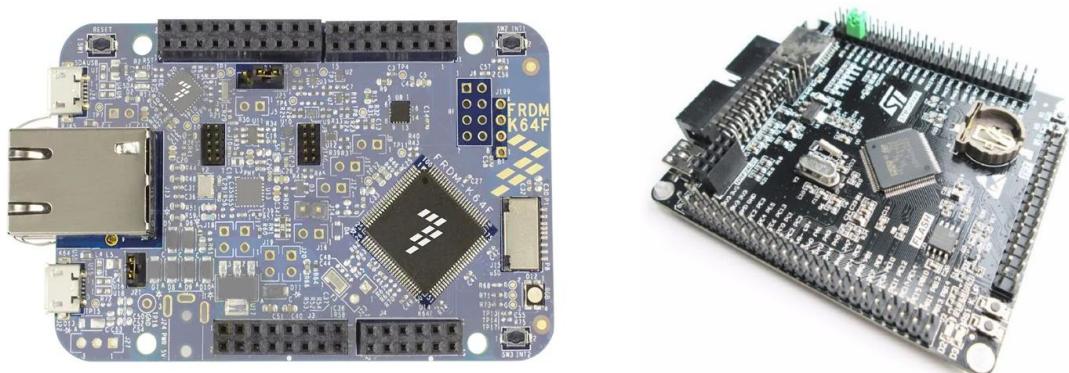
(LDO), a který zvládne pokrýt zařízení s až 8,5 A odběrem. Napájecí obvod je programovatelný přes I²C sběrnici.[34]

5.6 FRDM-K64F

Platforma³ je určená k vývoji a prototypování. Platforma je osazena mikrokontrolérem MK64FN1M0VLL12 od NXP. Tento MCU je založen na procesorovém jádru ARM Cortex-M4 a běží na frekvenci maximálně do 120 MHz a konkrétně je osazen čip s 1 MB flash paměti a 256 KB RAM. K tomu je mikrokontrolér vybaven kontroléry pro Ethernet, full-speed USB a SD hosta. Deska je form-factor Arduino R3 a lze připojit rozšiřující moduly pro Arduino. Na desce dále nalezneme šesti-osý akcelerometr a magnetometr FXOS8700CQ a tří barevnou led diodu nebo Ethernet konektor. Platforma obsahuje ladící adaptér přímo na desce, konkrétně OpenSDAv2. Adaptér je open source řešení poskytující CMSIS-DAP ladící rozhraní. Obvod tak nabízí jednoduchý způsob pro sériovou komunikaci, programování Flash paměti a ladění.

5.7 STM32F4VE

Jednoduchá vývojová platforma, která je založena na mikrokontroléru STM32F407VET6 s



Obrázek 5.6: FRDM-K64F (vlevo), STM32F4VE

ARM Cortex-M4F, tedy jádro s Floating Point Unit. Deska nabízí také externí SPI flash paměť o velikosti 16 Mb, micro-SD slot a baterii pro napájení RTC obvodu v mikrokontroléru. Tato deska neobsahuje integrovaný ladící a programovací adaptér, je tedy nutné použít externí adaptér jako je např. ST-Link, který je možné připojit k 20pinovému JTAG/SWD konektoru na desce. V rámci výuky to není ideální, ale zjednoduší to návrh desky. Deska je osazena velkým množstvím pinů, které mohou sloužit jako klasické digitální I/O, případně jsou na tyto piny vyvedeny sběrnice jako I2C, CAN, SPI nebo je možné generovat PWM signál. Na jednu sadu pinů jsou také vyvedeny piny FMS kontroléru procesoru, která slouží k řízení externích pamětí jako jsou NAND nebo NOT FLASH paměti nebo jako v tomto případě je tato sada pinů určena k řízení TFT LCD displeje.

³ <https://www.nxp.com>

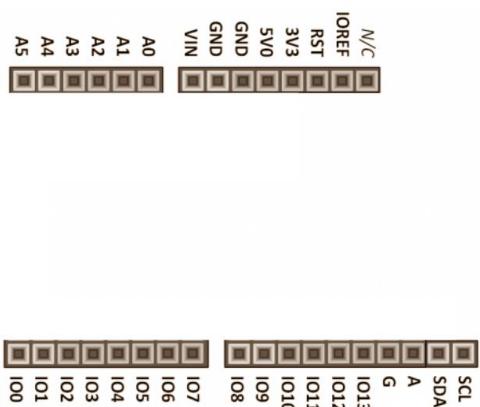
5.8 Arduino

S nadsázkou lze říci, že tato platforma se stala skoro standardem, který otevřel mnohým začátečníkům a nadšencům svět vestavěných zařízení a v době psaní této práce je komunita okolo Arduina velmi rozsáhlá, a tak existuje i velké množství rozšiřujících modulů, tzv. „shieldů“, které umožňují rozšíření této bare-metal desky o takřka jakoukoliv periferii a je tedy nasnadě je využít i v této práci, protože to otevírá, za předpokladu dodržení vývodové kompatibility a napěťových omezení, obrovské možnosti rozšíření. Andruino UNO obsahuje 8bitový mikrokontrolér ATmega328P, který je zasazený ve dvouřadé dutinové patici a umožnuje jeho jednoduchou výměnu při zničení. Tento mikrokontrolér je jednoduchý RISCový mikrokontrolér v této verzi umožňující 14 digitálních vstupů/výstupů, ze kterých 6 umí PWM výstup, a 6 analogových vstupů. Mikrokontroler běží na 16 MHz a obsahuje v sobě 32 KB FLASH paměti, 2 KB SRAM a 1 KB EEPROM. Deska má operační napětí 5 V, ale vzhledem k určení pro úplné začátečníky je hlavní napájecí vstup dimenzován na 6 až 20 V, ze kterých je napětí konvertováno na 5 V a deska má též 3,3 V napájecí okruh.



Obrázek 5.7: Arduino Uno (vlevo), Arduino Zero

Arduino Zero je form-factor Andruinu UNO, ale místo mikrokontroléru Atmega328P je na desce osazen 32bitový mikrokontrolér Atmel SAMD21, který obsahuje jádro ARM Cortex-M0+. Mikrokontrolér běží maximálně na frekvenci 48 Mhz a na čipu nalezneme také 256 KB Flash paměti a 32 KB SRAM. Deska obsahuje také ladící adaptér integrovaný na desce



Obrázek 5.8: Schéma vývodů Arduino Uno

a jedná se o proprietární adaptér EDBG od výrobce MCU. Vzhledem k použitému mikrokontroléru je pracovní napětí desky 3,3V a vstupní i výstupní piny nejsou tolerantní k napětí vyššímu, než je dané pracovní napětí 3,3V a vyšší napětí, které by pro klasické Arduino UNO s Atmega328P nebyl problém, by tuto desku mohlo poškodit.

Vývody kompatibilní s tímto rozvržením jsou důležitou součástí práce, protože bude možné připojit rozšiřující moduly pro platformu Arduino. V tomto uskupení vývodů jsou vyvedeny napěťové linky pro 3,3 V a 5 V, referenční napětí logického obvodu IOREF, RESET pin. Dále 6 analogových vstupů A0-A6, digitální I/O IO0-IO13, IO0(RX) a IO1 (TX) slouží u platformy také pro sériovou komunikaci. SCL, SDA slouží pro připojení I²C periferií. V rámci zapojení je možné použít desítky až stovky dostupných modulů.

Kapitola 6

Návrh a technická realizace platforem

V rámci předchozího přehledu bude navržen koncept dvou samostatných výukových platforem čili dvou desek plošných spojů, které budou uživateli či studentovi umožňovat jednoduché seznámení s platformou obsahující ARM mikrokontrolér nebo s platformou používající programovatelné hradlové pole. Cílem je však využít kombinace těchto dvou samostatných platforem jako v případě platforem FITkit. Je tedy zapotřebí kromě interního propojení obou desek uvažovat i jejich propojení a vzájemnou komunikaci, aby byla umožněna výměna dat mezi mikrokontrolérem a FPGA.

Návrh obou platforem vychází z přehledu platforem výše uvedených a dalších především open source projektů. Dále jsou v textu uvedeny hlavní myšlenky koncepce obou desek a jejich propojení. Tyto koncepce byly vytvořeny na základě práce s některými zmíněnými platformami a postřehy z uživatelské přívětivosti těchto platforem promítnuto do koncepce obou platforem.

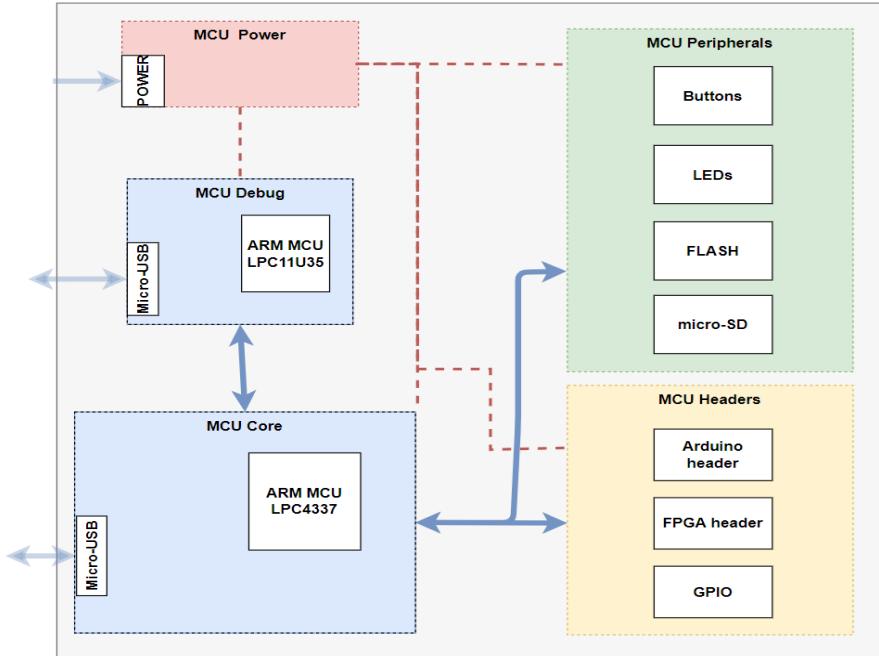
Obě platformy by měly umožňovat rychlé oživení a začínajícímu uživateli tak umožnit snadnou orientaci v obou platformách. Z tohoto pohledu je vhodné integrovat u obou desek programovací a ladící obvody, aby nebylo nutné složitě zapojovat externí moduly, obdobně jako u většiny populárních výukových platforem. Tato abstrakce vývojového a ladícího HW, obvykle v proprietární podobě od výrobce daného čipu nebo v podobě volně dostupné platforemy integrované přímo na desce je klíčová pro odstranění prvotní překážky u začínajícího programátora mikrokontrolérů a programovatelných hradlových polí, touto překážkou je nákup často nákladných vývojových prostředků od výrobců daných čipů.

Koncepce je zaměřena na maximální jednoduchost obou desek. Nicméně je kladen důraz i na použitelnost. V případě zkušeného uživatele je nasnadě umožnit komunikaci mezi platformami či doplnit další periferie jako jsou moduly pro měření neelektrických veličin, zobrazovací moduly či další na trhu již dostupná rozšíření pro již zaběhnuté platformy s širokou podporou open source komunity a společností stojících za výše uvedenými platformami. Obvykle jsou tyto rozšiřující moduly. Tyto rozšiřující moduly se pomocí kompatibilních vývodů propojují a rozšiřují funkcionality jednoduché základní platformy.

Platforma s ARM mikrokontrolérem bude obsahovat SWD/JTAG ladící adaptér, napájecí obvody, hodinový obvod, flash paměti, USB host/device, a především vhodně umístěné vývody pro připojení Arduino kompatibilních modulů a druhé platformy s FPGA. Dále bude tedy popsáno nejprve blokové schéma platformy s mikrokontrolérem s podrobnějším popisem hlavních součástek.

Obdobně je tomu pro platformu s FPGA, kde opětovně na základě blokového schématu jsou zvoleny hlavní součástky a tím nastíněny možnosti obou platforem. Platforma s FPGA by měla sloužit podobným způsobem jako na platformě FITkit sloužit k připojení komplexnějších periferií či jako akcelerátor výpočtů, které by na MCU trvaly příliš dlouhou dobu nebo je vhodné je implementovat pomocí paralelního HW zpracování, které je FPGA vlastní. Konkrétně by se měla platforma s FPGA sestávat JTAG ladícího a konfiguračního adaptéru, napájecích a hodinových obvodů, a opět vhodně umístěné vývody pro připojení Arduino kompatibilních modulů nebo platformy s mikrokontrolérem.

6.1 Platforma s ARM MCU



Obrázek 6.1: Blokové schéma navrhované platformy s MCU

6.1.1 Mikrokontrolér

Vybírat lze z obrovského množství mikrokontrolerů s nepřebernou paletou periferií a rozhraní. Pro tento projekt byl vybrán mikrokontrolér LPC4337 od firmy NXP. Tento mikrokontrolér je založen na jádru ARM Cortex-M4F a obsahuje též koprocesor s jádrem ARM Cortex-M0, který je vhodný k obsluze periferií a umožňuje ulevit zátěži hlavnímu procesorovému jádru. Jádro Cortex-M4F obsahuje jednotku pro výpočet v pohyblivé řádové čárce (FPU). Obě jádra běží na frekvencích do 204 MHz. K tomu nalezneme ve zvoleném 144 pinovém pouzdro typu LQFP také 1 MB flash paměti a 136 kB SRAM paměti. Mikrokontrolér obsahuje velké množství sběrnic jako I2C, I2S, SPI, SSI, UART nebo USB. Vestavěn na čipu je také External Memory Controller (EMC) podporující SDRAM, ROM, NOR flash či SRAM zařízení, do kterého lze zařadit i FPGA čip. Použití EMC je jedním z uvažovaných způsobů propojení obou desek a umožňuje přenést až 16 bitů na jeden hodinový cyklus, detaily jsou uvedeny v kapitole „Propojení platforem“. Pro ladění a programování obou jader Cortex-M4 i Cortex-M0 je možné využít JTAG či SWD protokolu, pro ladění pomocí integrovaného ladícího adaptéru budeme používat primárně JTAG rozhraní, a na desce bude vyveden 10pinový konektor JTAG/SWD pro připojení externího programátoru pro oba protokoly. Mikrokontrolér bude připojen na 12 MHz krystal jako externí zdroj hodin.

6.1.2 Ladící a programovací obvod

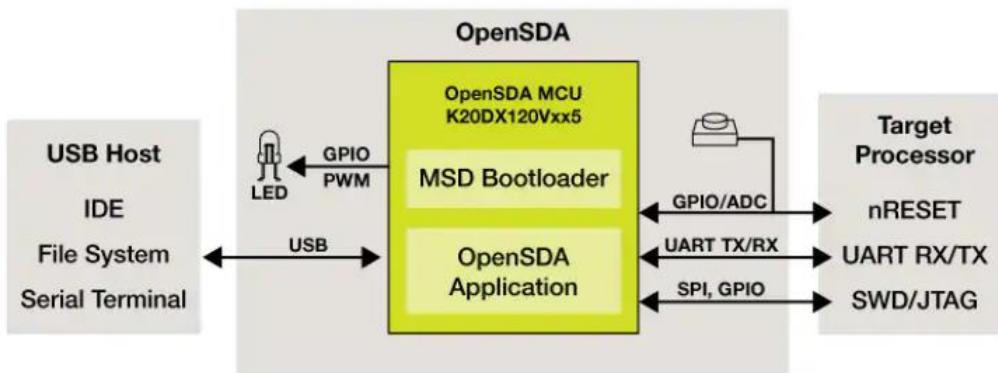
K ladění a nahrání firmwaru je obvykle zapotřebí nějakého proprietárního adaptéru/sondy, obvykle poskytovaný výrobcem daného MCU či FPGA. To však není příliš uživatelský přístup, kdy musí uživatel hned pracovat se samotnou deskou, a ještě programovacím a ladícím modulem, který se obvykle připojuje přes JTAG nebo SWD rozhraní. To však není obvyklé nativní rozhraní u běžných PC. Je tedy vhodné adaptér začlenit do návrhu obou

desek, obdobně je tomu na vývojových a výukových platforem, ze kterých bylo vycházeno při návrhu desky.

6.1.2.1 DAPLink

Původně projekt OpenSDA je sériový ladící adaptér, který je vestavěn do několika vývojových desek od NXP. Poskytuje tak most mezi počítačem uživatele (USB hostem) a cílovým mikrokontrolérem. Adaptér lze tak použít k ladění, programování a sériové komunikaci přes USB kabel.

DAPLink je nová open-source implementace mbed bootloaderu a standardního ladícího



Obrázek 6.2: Blokové schéma komunikace mezi USB hostem (PC) a cílovým procesorem

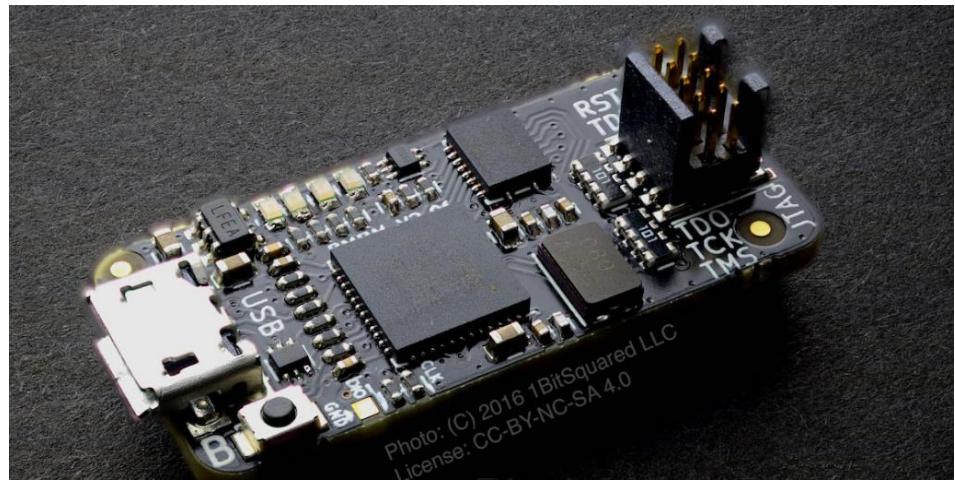
rozhraní umožňující programování a ladění aplikací pro ARM Cortex-M. To čemu dále budeme říkat rozhraní firmwaru DAPLink běží na sekundárním MCU, konkrétně LPC11U35, které je pomocí rozhraní JTAG nebo SWD připojeno na cílový mikrokontrolér. DAPLink umožňuje CMSIS-DAP ladění skrze sériový port a drag-and-drop programování. Je tak podporován všemi IDE, které podporují CMSIS-DAP protokol pro ladění. Návrh zapojení je volně dostupný k použití a existují příklady již naklonovaných aplikací s dostupnými schématy zapojení celých desek nebo jen hlavního rozhraní sond. Pro tento projekt byla použito schématu adaptéru SWDAP.

6.1.2.2 IBDAP

Další možnou implementací CMSIS-DAP ladící sondy je projekt IBDAP, který umožnuje ladění pomocí JTAG/SWD protokolu a je kompatibilní se stávajícím zapojením DAPLink s LPC11U35. Podporuje krokování v kódu, breakpointy a programování cílového mikrokontroléru.

6.1.2.3 BlackMagic adaptér

Krátce uvedeme další příklad open source projektu ladící sondy, postaveném na mikrokontroléru STM32F103CBUX od firmy STMicroelectronics. Obdobně jako u DAPLinku umožňuje sonda ladění a programování za pomoci JTAG nebo SWD protokolu. Jedná se o konkrétní sondu a pokud by byla sonda implementována na desce platformy JTAG konektor by nebyl nutný, ale vždy je vhodný, protože pokud není funkční sonda je možné připojit externí adaptér. Plně podporuje ladící funkcionality jako watchpointy, breakpointy ve Flash paměti a její programování, prohlížení pamětí a registrů.



Obrázek 6.3: BlackMagic adaptér

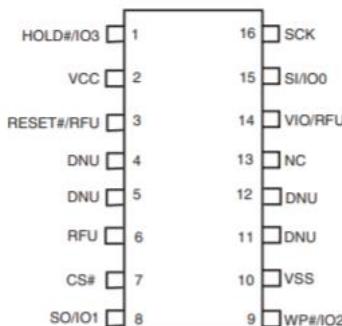
6.1.3 Napájecí obvody

Napájení u desky s mikrokontrolérem je realizováno přes dva USB porty pro ladění a I/O mikrokontroléru. Na desce je pro napájení z USB konektoru použit pevně nastavený LDO regulátor TPS77733DR s výstupním napětím 3,3V a maximálním výstupním proudem 750 mA. Tento regulátor napájí mikrokontrolér LPC4337 a měl by mít dostatečnou rezervu pro napájení jak samotných jader, tak případných I/O vývodů. Vzhledem k možnosti připojit Arduino kompatibilní modul, je na desce též 5V výstup, který je spojen s přímo s napájením USB konektoru. Součástí skupiny Arduino vývodů je i vývod pro vstupní napětí, který je připojen do výše zmíněného regulátoru přes odpojovací propojku, stejně tak je propojkou odděleno napájení z obou USB portů.

Na desce je též třeba napájet mikrokontrolér určený k ladění a programování LPC11U35. Tento obvod je napájen odlišným USB konektorem a k jeho napájení je použit slabší LDO regulátor TC1017 poskytující napětí 3,3V a maximální proud 150 mA, který je použit ve schématu adaptéru SWDAP. Je-li použit USB konektor pro I/O k LPC4337 není ladící adaptér s LPC11U35 napájen. Je-li použit USB konektor pro ladění a programování je třeba napájet oba mikrokontroléry. K USB portu pro ladění jsou tedy připojeny dva LDO regulátory TC1017 a TPS77733DR.

6.1.4 Periferie

Pro platformu byla vybrána flash paměť S25FL128SAGMFI001. Jedná se o paměť velikosti

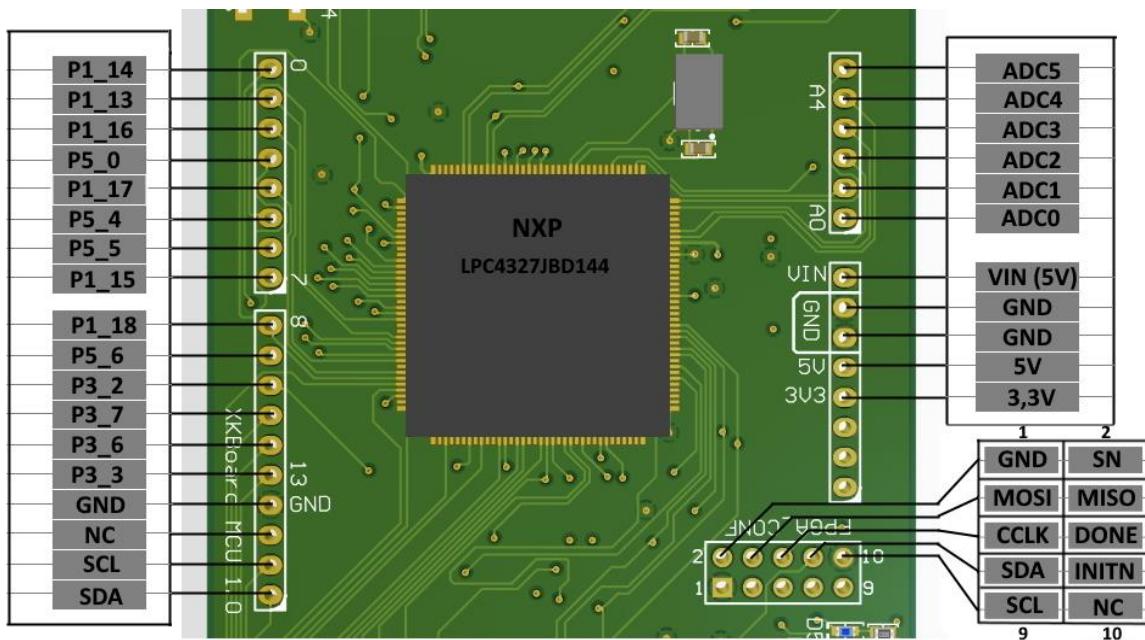


Obrázek 6.4: Blokové schéma S25FL128SAGMFI001

128 Mb. Paměť umožnuje komunikaci pomocí čtyřnásobné SPI sběrnice, tzv. Quad-SPI, s přenosovou rychlosí až 52 MB/s. Operační napětí je 3,0 V, ale paměť umožnuje připojení maximální napětí 3,6 V, a proto bude paměť napájena rozvodem 3,3V. Paměť bude k mikrokontroléru připojena přes SPIFI rozhraní na pinech P3_3 až P3_8. Z této paměti je možné bootovat mikrokontrolér či programovat FPGA pomocí uložené předem vytvořené a uložené konfigurace či konfigurací. Platforma bude také osazena LED diodami a tlačítky pro ověření základní funkcionality pomocí demonstračních příkladů.

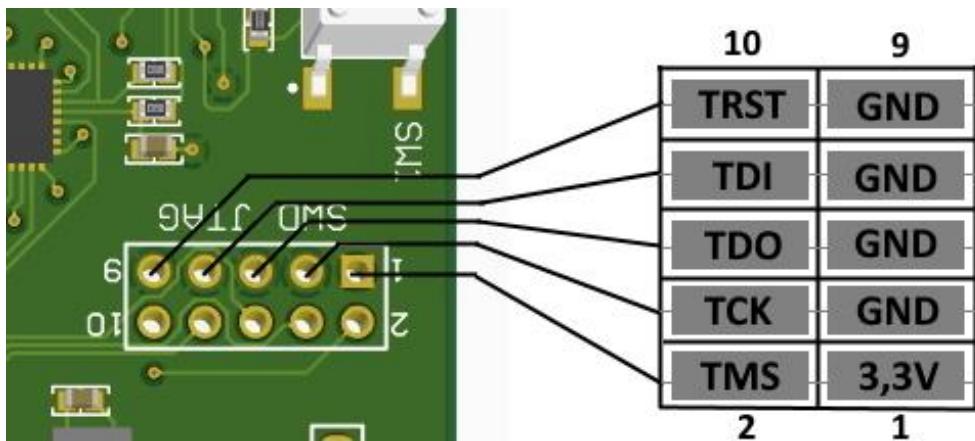
6.1.5 Rozvržení vývodů DPS

Hlavními vývody jsou vývody pro Arduino kompatibilní moduly. Vzhledem k použitému mikrokontroléru je na vývody možné přivést logický signál o maximálním napětí 3,3V jinak hrozí poškození mikrokontroléru. Obrázek 6.5 ukazuje rozvržení vývodů pro Arduino moduly a pro připojení platformy s FPGA. Piny vyvedené pro FPGA_CONF nejsou přednastaveny tímto způsobem a je nutné je manuálně nakonfigurovat a inicializovat před vykonáváním jakékoliv výměny dat, jinak mohou vykonávat jinou funkci a mohou nabývat funkcí určených výrobcem a jsou tak označeny především z praktického hlediska.



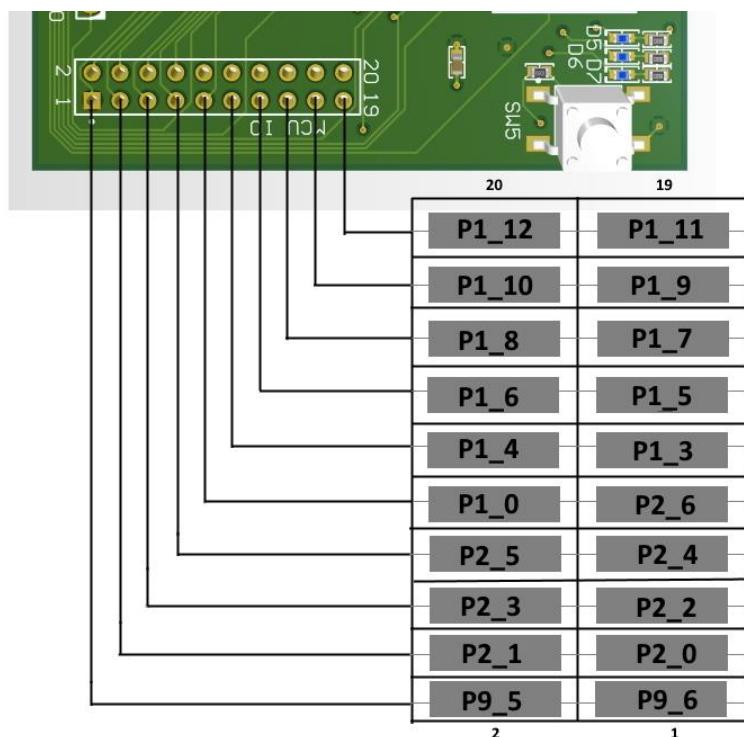
Obrázek 6.5: Arduino kompatibilní vývody a vývody pro konfiguraci/komunikaci s FPGA

Následující obrázek popisuje rozvržení vývodů označených jako MCU_IO a jedná se jen o další jinak nevyužité vývody mikrokontroléru. Na desce je vyveden konektor pro ladění mikrokontroléru v případě nefunkčnosti ladícího adaptéru osazeného na DPS a je možné platformu i z této skupiny vývodů i napájet.



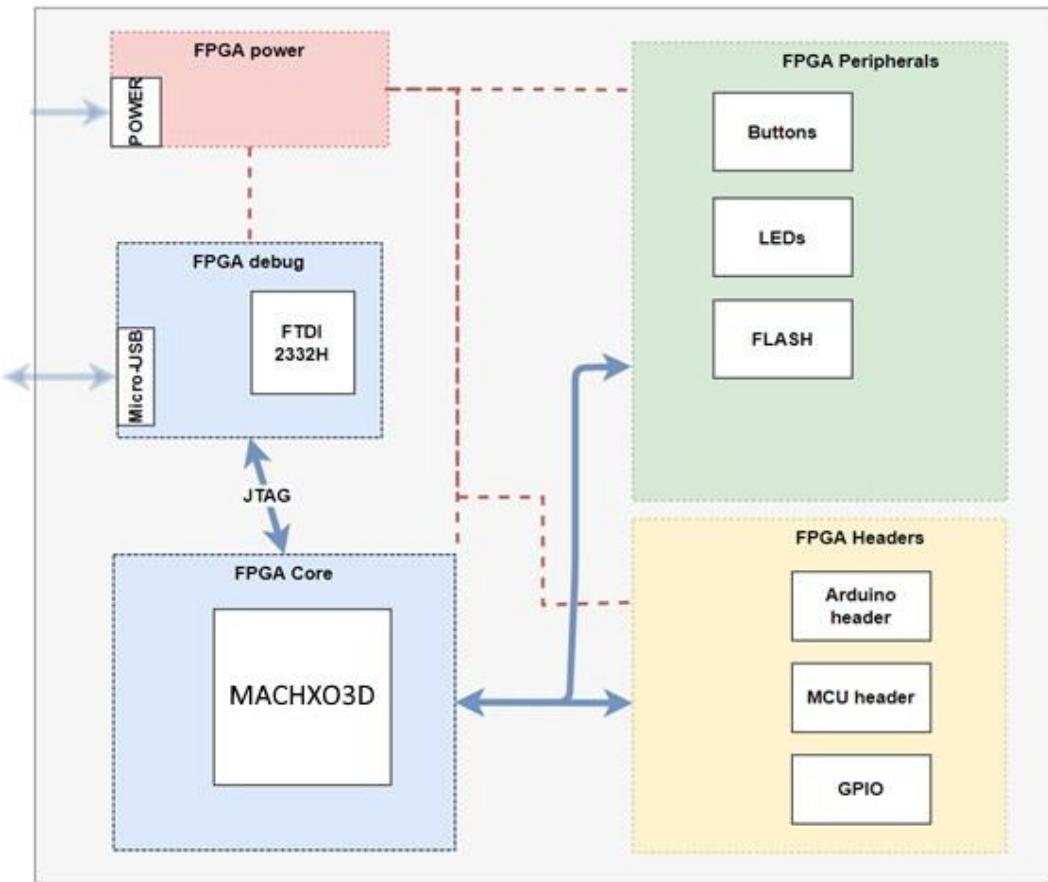
Obrázek 6.6: Vývody pro ladění mikrokontroléru LPC4327 (SWD/JTAG)

Tyto vývody jsou defaultně nastaveny na tyto signály, aby bylo možné vždy komunikovat s MCU, alespoň jedním rozhraním, pokud jsou tedy například nedopatřením programově přenastaveny ostatní komunikační rozhraní je stále možnost komunikovat alespoň přes ladící rozhraní JTAG



Obrázek 6.7: Vývody mikrokontroléru LPC4327 (MCU_IO)

6.2 Platforma s FPGA



Obrázek 6.8: Blokové schéma navrhované platformy s FPGA

Tato platforma vychází částečně z FITkitu Minerva, Spartan Edge Accelerator a Arty S7 platformy, avšak místo hradlového pole od firmy Xilinx je čip od firmy Lattice Semiconductor, který umožnuje použití pro konfiguraci čipu volně dostupné prostředky oproti čipům od firmy Xilinx, která poskytuje vlastní proprietární řešení, které nemá otevřený návrh a vynucuje použití výrobcem dodávaných a6ízení pro konfiguraci nebo různých ad-hoc přístupů jako .

6.2.1 Obvod FPGA

Pro realizaci bylo vybráno FPGA ze série MachXO3D od firmy Lattice Semiconductor. Konkrétně jde o model LCMXO3D-9400HC-5SG72C. Jedná se o FPGA s 9400 logickými prvky, vnitřní FLASH a SRAM pamětí. Pouzdro QFN s EPAD vývodem a 72 vývody poskytuje 58 I/O. Operační napětí je v rozmezí 2,5-3,3 V shodné pro jádro i I/O Banky[15]. Některá specifika této platformy jsme si již uvedli v rámci Kapitola 3 anebo se jim budeme ještě v dalších kapitolách věnovat.

6.2.2 Ladící a konfigurační obvod

Hlavní způsoby konfigurace FPGA, která je nutná vždy po připojení napájení k obvodu, slouží interní flash paměť nebo se FPGA konfiguruje pomocí platformy s ARM MCU

propojením obou platforem přes dedikovaný konektor. Kromě samotné konfigurace může být také zapotřebí ladění obvodu, u platformy s FPGA mluvíme o logickém analyzátoru Reveal. K tomuto účelu poskytuje FPGA klasický JTAG protokol a na desce bude vyveden JTAG konektor. Nicméně, jak již bylo uvedeno, chceme mít možnost FPGA ladit bez nutnosti externího adaptéra. Pro tento účel byl zvolen obvod FT2232HL od firmy FTDI. Tento USB převodník poskytuje dva nezávisle programovatelné kanály A a B. Oba kanály poskytují následující režimy převodu USB 2.0 High Speed na:

- UART (RS232/RS422/RS485)
- FIFO
- MPSSE (Multi-Protocol Synchronous Serial Engine) – JTAG, I²C, SPI a bit-bang

Jde nám, tedy především o JTAG funkcionalitu poskytovanou MPSSE na portu A[35]. Platforma s FPGA bude obsahovat flash paměť, ve které bude možné uložit konfigurační řetězec. Krom toho má FPGA i interní flash paměť, které je primárním zdrojem konfiguračního řetězce.

6.2.3 Napájecí obvod

Napájení celé desky je realizováno jedním LDO regulátorem NCP1117DT33T5G, který reguluje vstupní napětí 5 Voltů na výstupní napětí 3,3 V a proud 1 A. V rámci DPS je napájení 3,3 V rozvedeno do vnitřních vrstev DPS, ke kterým pak jen stačí přivést požadovaný prokov.

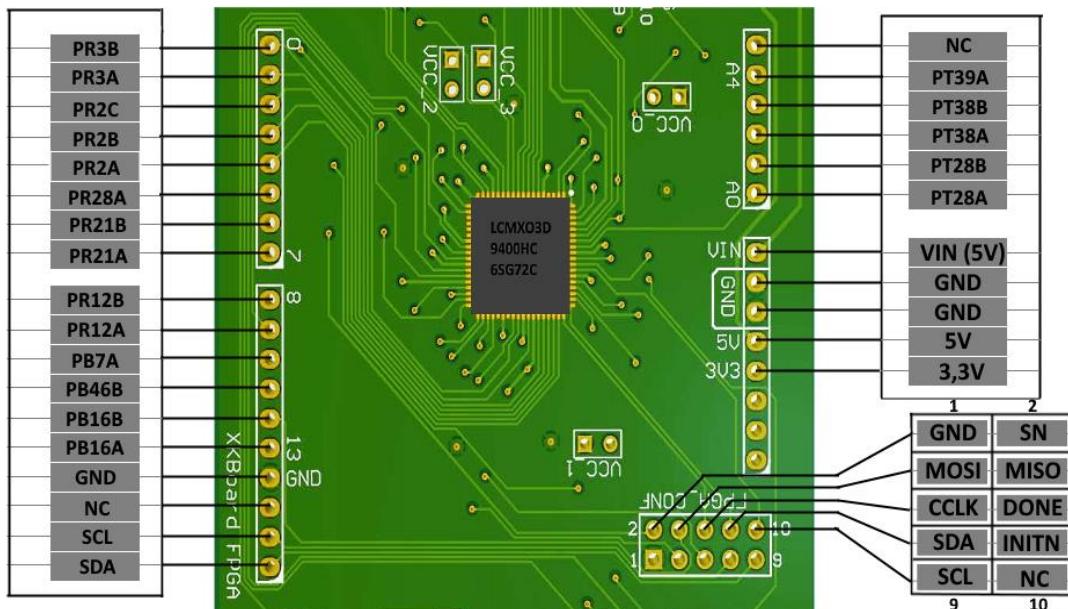
6.2.4 Periferie

Flash paměť S25FL128SAGMFI001 je na této desce použita pro uchování konfigurace FPGA mezi jednotlivými napájecími cykly, kdy je SRAM paměť FPGA smazána. Paměť může také sloužit k uložení uživatelských dat či například sdílet tuto paměť s připojeným mikrokontrolérem, vše již záleží na implementaci obvodu v FPGA. Paměť je připojena přes SPI rozhraní.

Dále platforma bude obsahovat LED diody a tlačítka připojená k programovatelným vývodům FPGA. Tyto součástky jsou připojeny přes rezistory a jejich ovládání a vývodu budou uvedeny v rámci demonstrace.

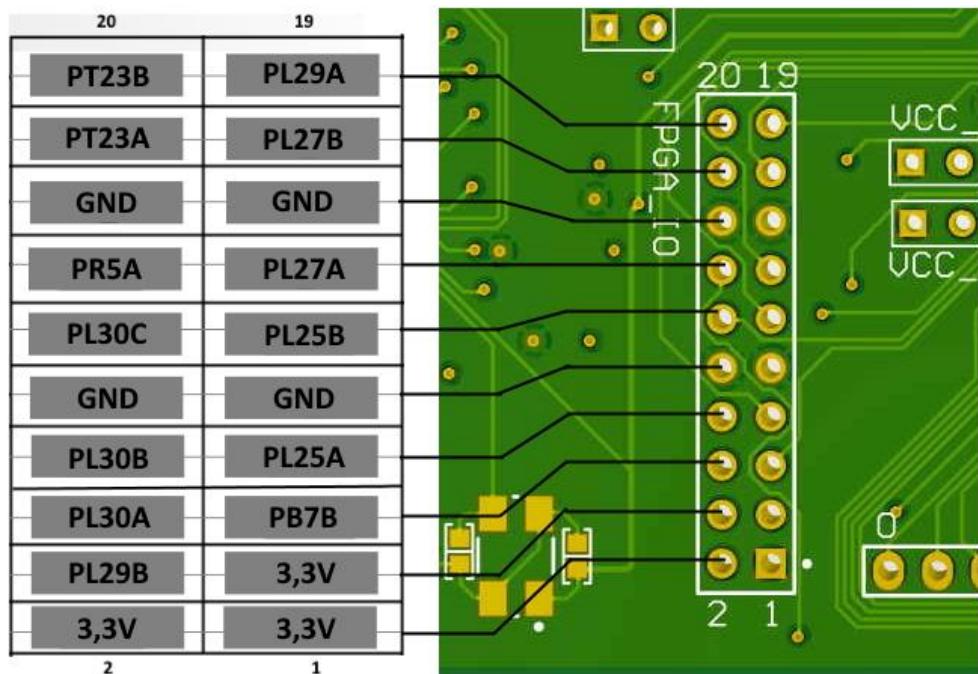
6.2.5 Rozvržení vývodů DPS

Rozvržení vývodů platformy s FPGA je podobný platformě s ARM mikrokontrolérem. I tato

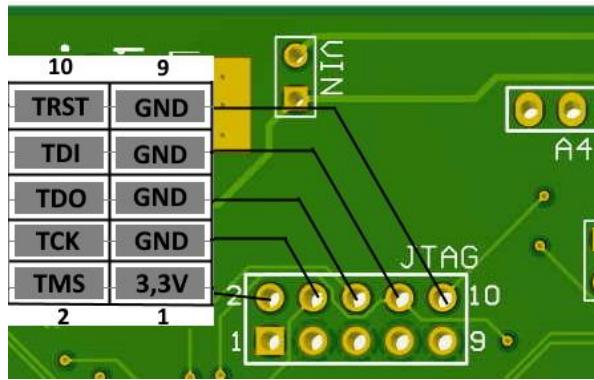


Obrázek 6.9: Arduino kompatibilní vývody a vývody pro komunikaci s MCU

platforma má omezení na maximální logické napětí 3,3V. FPGA konektor, zde má podobu pinů zespodu desky, aby bylo možné piny zasunout do dutinové lišty na druhé desce plošných spojů s mikrokontrolérem. Kromě vstupně/výstupních pinů bude na desce vyveden konektor pro použití externího JTAG adaptéru.



Obrázek 6.11: Přebytečné vývody FPGA označené jako FPGA_IO



Obrázek 6.10: Vývody JTAG rozhraní

6.3 Propojení platforem

Jedním z cílů této práce je zkombinovat obecnou funkcionalitu FPGA a ARM MCU, které nabízí nepřeberné množství již vyladěných periferií a nemusí být implementovány navíc na FPGA. Propojení obou platforem je možné pomocí vyvedené I²C sběrnice nebo SPI sběrnice. Obě tyto sběrnice je možné použít zároveň ke konfiguraci FPGA při jeho inicializaci, a poté ke komunikaci s FPGA. Konfigurační proces je přeprogramovanou rutinou po resetu či v průběhu běhu zařízení a vyžaduje implementaci požadovaných příkazů na straně MCU[8]. Následná komunikace obou desek již vyžaduje implementaci obvykle stavového automatu komunikačního protokolu používající I²C nebo SPI sběrnici. Podrobně se tomuto tématu budeme věnovat v kapitole s demonstračními příklady, kde si dále specifikujeme postupy programování a poté samotný příklad.

6.4 Vytvoření schémat zapojení

Schéma zapojení a jeho návrh vychází z funkčních návrhů ladících nástrojů popsaných dále a jsou doplněny o požadovaný cílový mikrokontrolér nebo FPGA.

Pro vytvoření schémat zapojení a následovně celého projektu je použito programu Altium Designer, který umožňuje vytvoření všech nezbytných podkladů k výrobě desky plošných spojů. Též umožnuje vyhledávat součástky od nejznámějších a největších dodavatelů součástek jako Farnell, Mouser, TME, ale o tom podrobněji dále. Následující text je zaměřen především na Altium Designer, a i přes to, že tento nástroj sdílí s konkurenčními nástroji určitou podobnost danou přirozeným přístupem a potřebami návrhu desek plošných spojů, je v detailech odlišný.

Jednou z výhod Altium Designer je možnost použití knihoven se součástkami a motivem pájecích ploch součástek. V projektu je využito hned několik volně dostupných knihoven či obecně zdrojů, ze kterých jsou čerpány podklady pro všechny součástky použité v této práci. Prvním zdrojem je databázová knihovna Celestial Altium Library, která je pravidelně aktualizovaná a je v ní tak možné najít podklady pro nejnovější součástky. Převážná většina součástek je tak z tohoto zdroje a dalšími zdrojem je online databáze SnapEDA, která umožnuje volně stáhnout nepřeberné množství schématických symbolů a motivů pájecích ploch vytvořených lidmi podporujících tento webovou databázi. Dobrým zdrojem podkladů jsou také distributoři součástek, především Mouser na svých stránkách nabízí schémata i motivy pro součástky, které prodávají. Pro stažení podkladů existuje v Altium Designeru i plugin, který automaticky stáhne schéma i motiv a součástku automaticky umístí na právě otevřené schéma. Je vhodné si zkonto rovat, zda schéma i motiv odpovídají a jsou správně namapovány piny schématu i motivu mezi sebou, protože některé součástky jsou automaticky generované a některé nemusí odpovídat aktuální revizi skutečně zakoupené součástky.

6.5 Metodika

Flat návrh je přístup, při kterém schéma organizujeme vše v jedné rovině bez hierarchie a další hlubší organizace nad jednotlivými schématy, v nejjednodušší podobě si můžeme představit jeden schematický list o velikosti A4, na který se vejde celý obvod a není tedy nutná žádná hierarchie. Tento způsob je tedy vhodný u menších obvodových zapojení. Dále se tímto přístupem zabývat nebudeme.

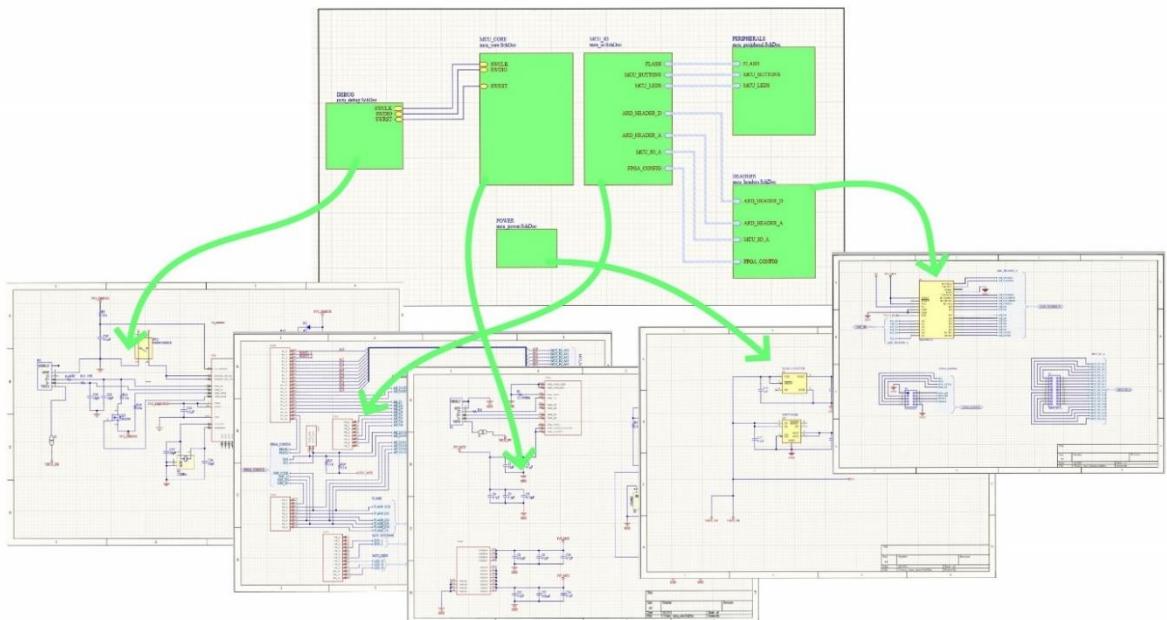
V této práci se zaměříme hierarchický návrh. Je to způsob sofistikovanější, a především umožnuje jednodušší orientaci v návrhu i při tvorbě motivu desky. Tento způsob se vyznačuje užitím speciálního schématu se symbolickým znázorněním všech ostatních schémat v roli součástek a jejich vzájemného propojení v rámci celku.

Pro tento způsob organizace schématu je vhodné si popsat jakým způsobem je možné propojit součástky v rámci jednoho schématického nákresu i mezi více schématickými nákresy jaké prvky umožnuje Altium Designer použít. Existují dvě možnosti, fyzické a logické propojení pinů součástek v rámci jednoho schématu. Fyzický způsob je natažením spoje mezi jednotlivými piny. Způsobem druhým je logické propojení, které umožnuje pin součástky pojmenovat za použití propojovacího prvku tzv. „Net label“, tedy pojmenování, a piny se stejným pojmenováním považuje editor za taktéž propojené. Dalšími propojovacími prvky používanými především pro propojení mezi více schématickými nákresy jsou sběrnice, port, signální svazek a napájecí port.

Sběrnice je propojovací prvek, který umožňuje plošné spoje sdružovat do jedné sběrnice. Tento přístup umožnuje seskupovat např. paralelní datové vodiče a podobné. Propojovací prvky v Altium Designer mají generickou podobu konektoru, jejich vstupů a vedení mezi jednotlivými konektory at' už se jedná o svazek či sběrnici.

Dalším i v této práci využívaného typu logického sdružení vodičů je svazek, který umožnuje sdružit jak různorodé signály, a tak i samotné sběrnice. Opět má obdobnou strukturu vstup konektoru, konektor a vedení mezi konektory.

Porty umožňují propojení mezi jednotlivými schématickými nákresy na základě propojení na vyšší úrovni v hierarchickém přístupu k návrhu, ale i ve flat přístupu. Propojení je definováno na základě jména portu. Podobným prvkem jsou i napájecí porty, které umožní globalizovat plošný spoj na základě přípojného bodu se symbolem napájení či např. země. Opět je důležité pojmenování napájecího portu, na jehož základě poté editor propojí všechny stejně pojmenované porty do jednoho plošného spoje.



Obrázek 6.12: Hierarchický návrh (Platforma s MCU)

6.6 Návrh motivů

Vlastní návrh motivů se řídí schématickým znázorněním, tedy máme-li například blokovací kondenzátory umístěné ve schématu blízko čipu budou stejně umístěny i v motivu. Dalším hlediskem, kterým se návrh motivu řídí je elektromagnetická kompatibilita součástek a jednotlivých celků v rámci schématických nákresů.

6.6.1 Technologické postupy

Pro výrobu desek byl využit prototypový POOL servis od firmy Gatema. Tato služba umožní výrobu prototypu desky až do 6 vrstev. Tato služba má specifické výrobní podmínku oproti klasickému zadání do výroby a vychází z nich i specifikace pro výrobu, které je třeba dodržet pro bezproblémové vyrobení desek plošných spojů. V rámci výrobního procesu čekají DPS kromě vyleptání cest v měděné fólii také vrtání otvorů, mechanické opracování desky jako řezání a frézování, různé čistící procesy, nanášení krycích a ochranných vrstev a mnohé další.

6.6.2 Specifikace výrobce

Pro desky v tomto projektu bylo využito POOL servisu[36] a ten specifikuje následující

parametry pro formát dat, obrys desky, signálové vrstvy, nepájivou masku a vrtání. Pro naše platformy je použit 4vrstvá DPS o tloušťce 1,6 mm.

6.6.3 Vrstvy a jejich složení

Jednotlivé vrstvy DPS jsou nejčastěji z vyrobeny z laminátu, tedy skelné tkaniny zalité epoxidovou pryskyřicí. Tato vrstva laminátu je pokryta měděnou fólií, do které je následně chemickou cestou přes masku odebrána měď, aby zůstaly jen zamaskované cesty a vznikl požadovaný obvod. V závislosti na použití a složitosti desky je možné zvolit různé tloušťky. Obvykle je tento materiál označován FR1-FR5, kde FR4 je nejobvyklejší a v rámci například čínských výrobců DPS i nejlevnější variantou.

6.6.4 Prokovy

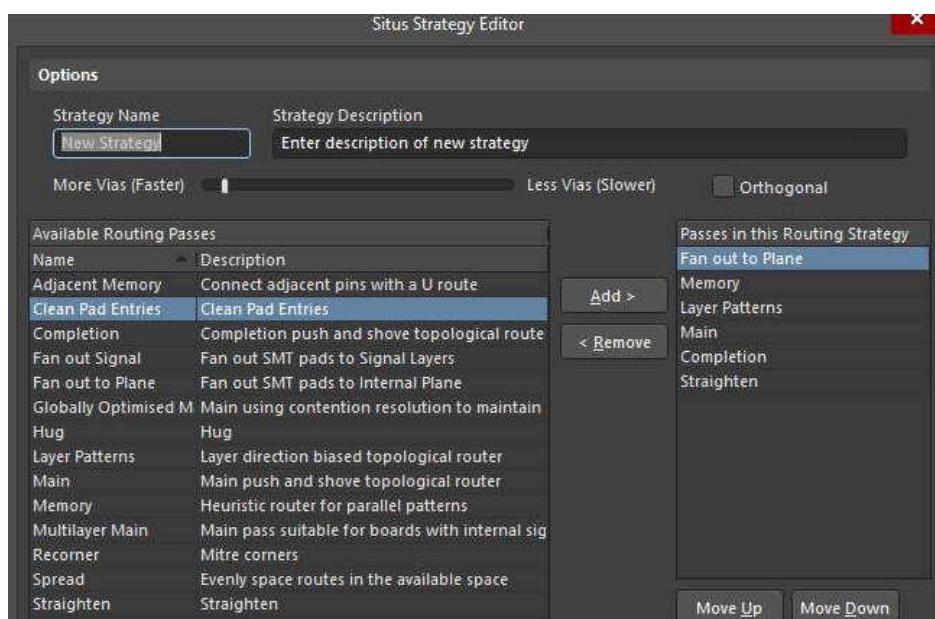
Prokovy jsou vrtané prokové otvory skrze desku plošného spoje. Umožnuje vést spoj skrze desku a např. tak překročit vedení jiných spojů, přes které nelze na jedné vrstvě obejít nebo by cesta byla zbytečně dlouhá. Tento postup též využijeme nemáme-li dostatek prostoru pro vedení cest v okolí čipu nebo v případě více vrstv desky může jako u obou realizovaných desek i jako napájecí prokovy do napájecích polygonů a rovin ve vnitřních vrstvách desky.

6.6.5 Rozmístění součástek

Jak již bylo naznačeno, umístění součástek na desce podléhá několika kritériím, především elektromagnetické kompatibilitě a vzájemné vzdálenosti komponent. Dále bereme v potaz samotné schématické nákresy, tedy relativní vzdálenosti a pozice komponent ve schématu. Díky nástrojům pro práci mezi schématem a motivem cross-selection a cross-probe je možné ze schématického návrhu postupně uchopit jednotlivé součástky, tedy jejich motivy a přímo je umísťovat na konečnou pozici na desce například v relativní pozici, tak jak jsou umístěny ve schématu.

6.6.6 Routování

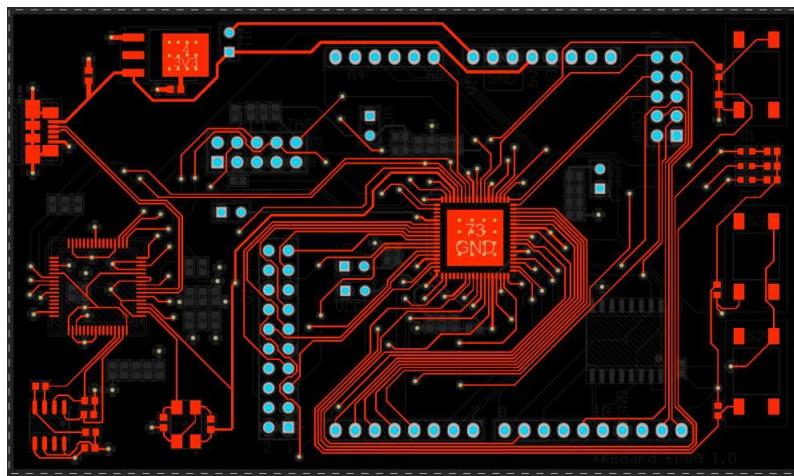
Vedení spojů mezi součástkami je vlastně tím důvodem proč DPS vytváříme a v případě vedení spojů mezi jednotlivými komponenty řešíme jakým způsobem je co nejlépe propojit,



Obrázek 6.13: Strategie pro automatický router plošných spojů

jedná se vlastně o specializovaný grafový problém. Pro vedení plošných spojů existuje mnoho algoritmů, které používají automatické routery. Následně si krátce popíšeme, jak tyto nástroje pracují a jaký je algoritmický přístup k řešení vedení plošných spojů.

Problém vedení vodivých cest (routování) na desce je možné řešit v moderních návrhových aplikacích tzv. autoroutery. Autorouter kombinuje obvykle několik různých algoritmu pro pokládání cest plošných spojů a sám postupuje podle předvolené strategie. Tyto strategie jsou použity v různých počtech iterací, aby bylo dosáhнуto strategií vytyčených cílů a parametrů. Strategie se řídí faktory jako počet prokovů, rozptyl spojů a různá zohlednění pro vstup cest do pájecí plochy nebo zohlednění vícevrstvé desky a mnohé další.



Obrázek 6.14: Motiv platformy s FPGA s výrazněnou vrchní vrstvou desky

Obecně lze problém charakterizovat následovně: Pro danou plochu (DPS) a fixní počet vrstev najdi takový vzor vedení vertikálních (prokovů) a horizontálních spojů, který dle schématu propojí odpovídající kontaktní plochy výstupů jednotlivých součástek.

Pro lepší představu o komplexitě tohoto problému si představme ten nejjednodušší problém, a to propojení dvou součástek na jedné vrstvě bez jakýchkoliv překážek a dalších pravidel, také tzv. „Steiner tree“ problém je NP-těžký problém, a to pouze pokud je možné vést cestu pod jakýmkoliv úhlem. Omezíme-li vedení cesty pouze horizontálně či vertikálně je zmíněný problém i NP-úplný [19]. Z tohoto důvodu se u algoritmů pro automatické routování spokojíme s nějakým optimálním řešením, které hledáme s použitím vhodně zvolené heuristiky, kde vyměníme nejlepší řešení, avšak ve výpočetně neakceptovatelně dlouhém čase, za uspokojivé řešení ve výpočetně krátkém čase. Jaké řešení je uspokojivé a jaký čas je výpočetně krátký je na návrháři desky a případně dalších externalitách jako ekonomická stránka či další technologické požadavky jako například vedení spojů DDR pamětí či analogové vedení kde, již mohou automatické routery selhávat.

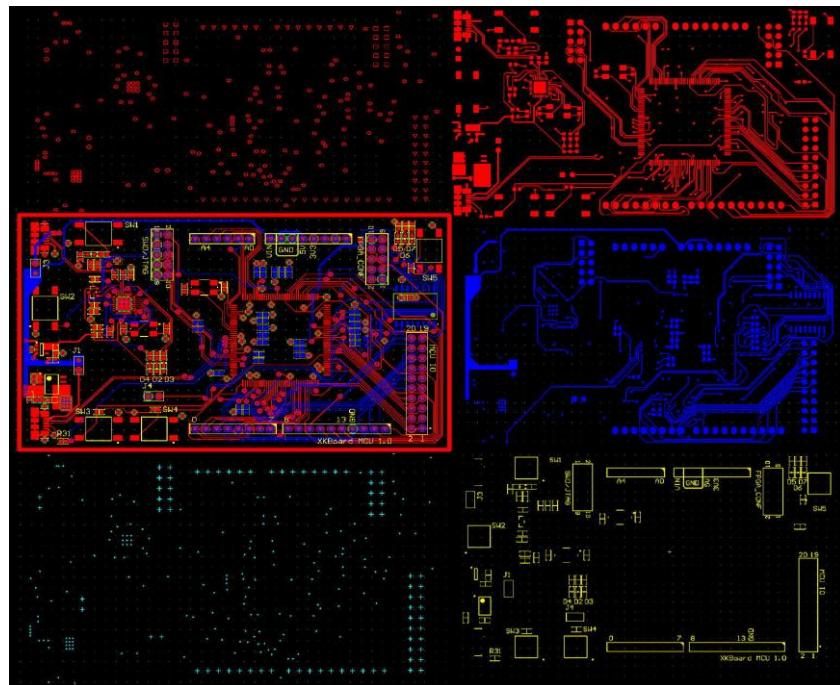
Existují také různé úrovně abstrakce nad routovacím procesem, a to globální nebo detailní routování. Globální routování slouží rozvrhnutí k vedení různých sběrnic, napájecích spojů a dalších signálů, tak aby nebylo při detailním vedení spojů příliš křížení a také, aby spoje byly vedeny v rámci elektromagnetické kompatibility, ale nezaměřuje se na konkrétní vedení jednotlivých cest. Detailní vedení spojů je opakem, tedy zabývá se vedením konkrétních cest mezi jednotlivými pájecími plochami v hranicích daných globálním

návrhem vedení.

Na obrázku 6.14 je vidět motiv vrchní strana motiva desky platformy s FPGA a v rámci návrháře Altium Designer je možné si desku plošných spojů vizualizovat ve 3D, ale to pouze v případě, že máme pro dané komponenty vytvořené 3D modely. Jedná se pouze o modelovou vizualizaci, ale při velkém množství komponent může odhalit různé problémy s umístěním.

6.7 Výrobní podklady

Po fázi iteracích vývoje zabývajícím se návrhem desky a vedením spojů je pro skutečnou realizaci vygenerovat, pomocí nástrojů poskytovaných daným vývojářem, výrobní dokumentaci, která bude následně předána výrobci desek plošných spojů. My se budeme zabývat především výrobní podklady pro POOL servis. Pro tento typ rychlé prototypové výroby lze exportovat výrobní dokumentaci ve formátech GERBER, GERBER X2 a mnoha dalších, ale tento formát byl použit pro tuto práci. Na obrázku 6.15 můžeme vidět jednotlivé vrstvy generované ve formátu GERBER. Představují vlastně základ masky pro leptání vodivých vrstev, které budou následně slisovány v jednu a jednotlivé vrstvy budou provrtány a pokoveny, aby vytvořily vertikální vodivé cesty.



Obrázek 6.15: Jednotlivé vrstvy platformy s MCU v GERBER
formátu pro výrobu DPS

Kapitola 7

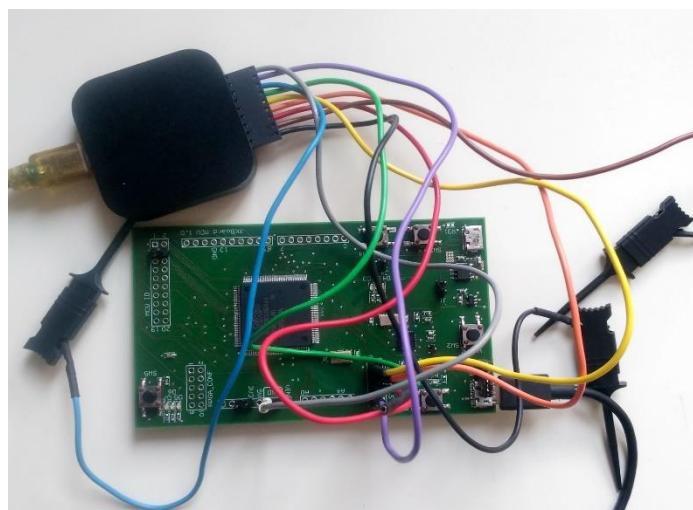
Osazení desek plošných spojů a jejich oživení

V rámci práce se také krátce vyjádříme k osazení a následnému oživení desek. Obě desky byly navrhovány, tak aby je bylo možné osadit ručně bez osazovacího automatu. Byla tedy využita konstrukční třída 0603 [41] ještě stále umožňuje ruční osazení a ostatní součástky jsou typu LQFP a QFN a na jejich osazení stačí obyčejná pájecí stanice s úzkým špičatým hrotom a případně hrotom s plochým hrotom pro rozvírání cínu přes vývody LQFP pouzder. Pro QFN s odhalenou pájecí plochou pro odvod tepla je vhodná horkovzdušná pájecí stanice a pájení za pomocí pájecí pasty.

V průběhu práce nastaly výpadky v součástkách a je tak oproti původnímu záměru LPC4337 použita nižší řada mikrokontroléra LPC43XX, a to konkárně LPC4327, která je takřka identická s řadou LPC4337, ale obsahuje menší flash paměť. Osazena byla nejprve spodní strana s blokovacími kondenzátory a FLASH pamětí u obou desek. Vrchní strana desky byla osazována spolu s oživováním jednotlivých komponent.

Oživovací proces spolu s osazovacím probíhal směrem od podporných obvodů k mikrokontrolérovi. Při oživení desky s MCU bylo výhodné nejprve osadit ladící obvod, který by měl být samostatně funkční. Jeho funkčnost lze ověřit připojením MicroUSB konektoru a desku připojit k PC. Ve Windows se mikrokontrolér ladícího obvodu LPC11U35 objeví jako diskové zařízení, na které je možné nahrát požadovaný firmware v binární podobě.

Je-li ladící obvod funkční je možné osadit hlavní mikrokontrolér a napájecí obvody, které napájí vnitřní vrstvu 3,3 voltů a napájí hlavní mikrokontrolér. Poté se můžeme pokusit se pomocí ladících nástrojů a ladícího obvodu s využitím SWD protokolu spojit s hlavním mikrokontrolérem. Nicméně funkčnost mikrokontroléra je možné ověřit až po nahrání firmware do mikrokontroléra ladícího obvodu. Detailně se ladícím nástrojům a nahrání konkárního firmware pro mikrokontrolér ladícího obvodu a hlavní mikrokontrolér věnuje část o demonstraci funkčnosti obou obvodů



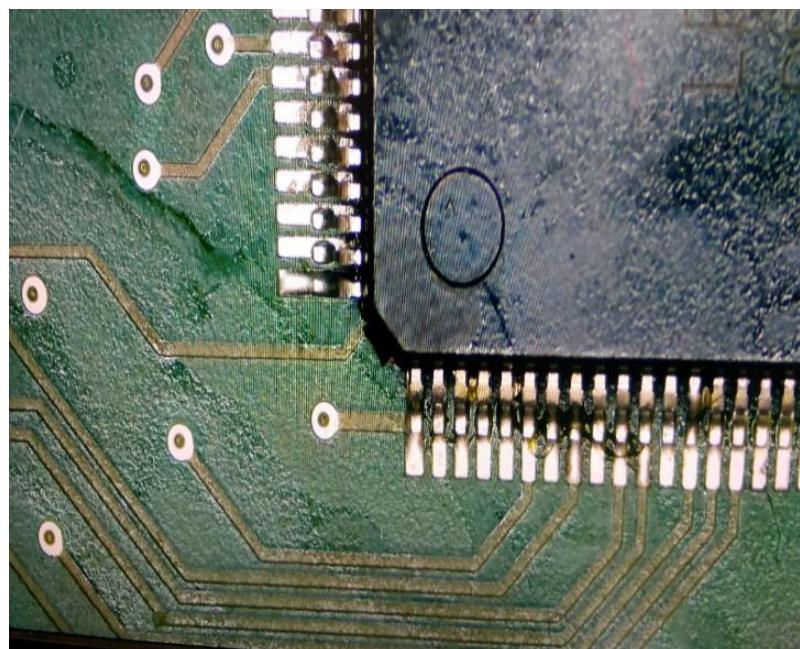
Obrázek 7.1: Oživování platformy s pomocí logického analyzátoru Saleae

Osazení a oživení desky s FPGA probíhalo obdobně. Nejdříve byl osazen ladící obvod FT2232HL. Pro ověření jeho funkčnosti na platformě Windows, opětovně připojíme pomocí

MicroUSB konektoru k PC. Je-li obvod v pořádku objeví se jako neznámé USB zařízení, pro další ověření funkčnosti je nutné doinstalovat ovladače od výrobce FTDI, které budou emulovat sériové zařízení po připojení k PC.

K oživení FPGA obvodu je zapotřebí propojit propojku označenou jako VCC_CORE a VCC_0, ale opět je pro ověření správné funkce FPGA nutná další příprava a konfigurace ladícího obvodu, které se budeme věnovat v části demonstrace funkčnosti.

Ruční osazování může skýtat několik problémů. Největším problémem je přesné pozicování součástek s velkým počtem vývodů jako je např. LPC4327, které má 144 vývodů a malé vyosení při přichycování součástky vyústí v okem neviditelné zkraty a nefunkčnost obvodu. Pro diagnostiku takového problému či například ověření funkčnosti krystalových oscilátorů apod., je možné použít osciloskopu případně analyzátoru logických úrovní. Pro ověření funkčnosti a ladění je v této práci použito Saleae logického analyzátoru a klasického multimetru. S těmito nástroji je možné ověřit komunikační protokoly jako JTAG nebo SPI a multimetrem lze ověřit napájecí obvody a případné zkraty.



Obrázek 7.2: Detail osazeného mikrokontroléru LPC4327

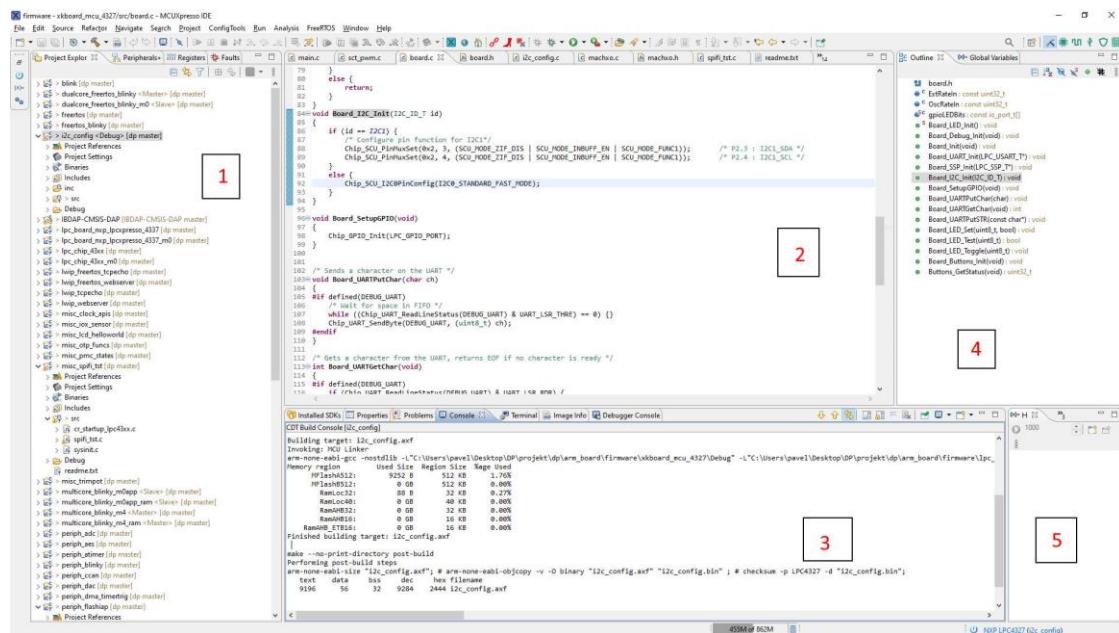
Kapitola 8

Programové vybavení pro platformu s MCU

V rámci této kapitoly se zmíníme o použitém softwarovém vybavení a potřebném firmwaru pro ladící adaptér a obvod s mikrokontrolérem LPC4327. Kapitola se obecně věnuje potřebnému softwarovému vybavení především vývojového prostředí MCUXpresso k MCU od NXP a krátce i pracovním postupům či tzv. workflow v tomto vývojovém prostředí.

8.1 Vývojové prostředí

MCUXpresso IDE je prostředí určené pro mikrokontroléry od výrobce NXP. Toto prostředí je kompatibilní s použitým ladícím obvodem a umožnuje hlavní mikrokontrolér LPC4327 programovat s automaticky generovanými podpůrnými soubory s konfigurací hodin, paměti a periferií a Makefile pro překlad pomocí překladače GNU Arm Embedded.



Obrázek 8.1: Představení jednotlivých částí prostředí MCUXpresso

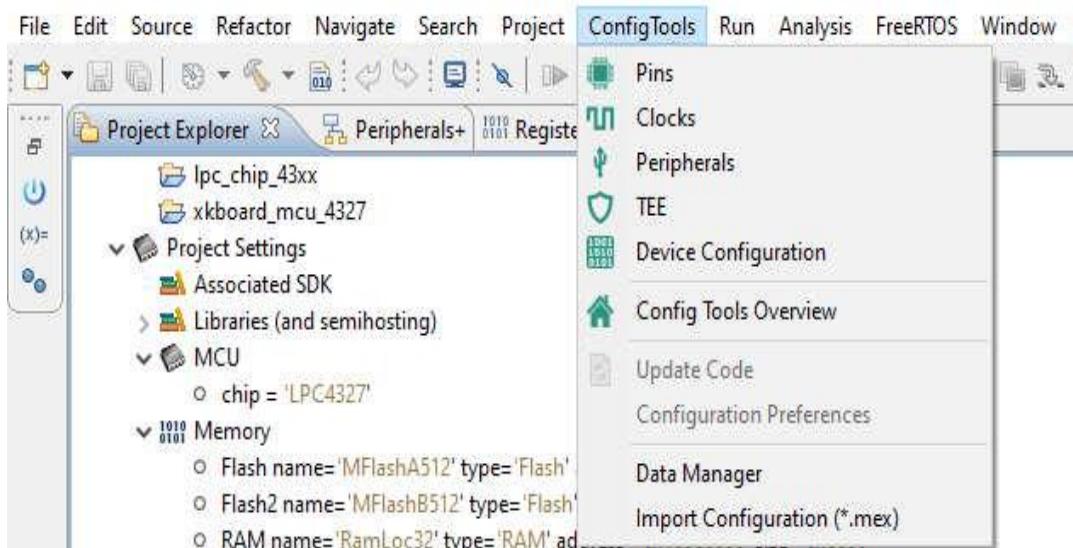
8.2 Základní popis prostředí

Jedná se o projektově orientované vývojové prostředí. Zběžně si popíšeme základní okno a jednotlivá vnořená okna vývojového editoru a nástroje pro konfiguraci čipu.

Složka projekty (workspace) je pomocí správce souborů vyobrazen v oblasti 1. V této oblasti se zobrazují kromě samotných souborů také reference na nastavení projektu. Oblastí s číslem 2 je samotný textový editor pro psaní kódu. V oblasti 3 se defaultně spouští okno logovací konzole, kam jsou vypisovány informace o překladu, ladění a další informace. Dalšími oblastmi defaultně nastaveného rozvržení editoru jsou v oblasti 4 různá informační

okna jako „Outline“ právě otevřeného zdrojového souboru nebo globální proměnné. To pokrývá základní rozvržení editoru, kromě těchto oblastí obsahuje prostředí také nástroje pro práci s vestavěnými zařízeními a vývojovými platformami od NXP.

Pro většinu vývojových desek od NXP existuje možnost stáhnout SDK pro danou desku, který popisuje pomocí konfiguračních souborů (soubory s příponou. mex) různá další specifika a je také zdrojovými soubory pro další nástroje, které MCUXpresso IDE poskytuje. Krátce si tyto nástroje představíme. V projektu samotném nebylo SDK pro desku vytvářeno, ale bylo použito projektu s implementací různých sběrnic a periferií, tak jsou na desce osazeny a zapojeny. Tento způsob vytváření projektu bez přidruženého SDK pouze pomocí projektu, který obsahuje definici rozhraní a implementaci funkcí použitelných na navrhované platformě s MCU.



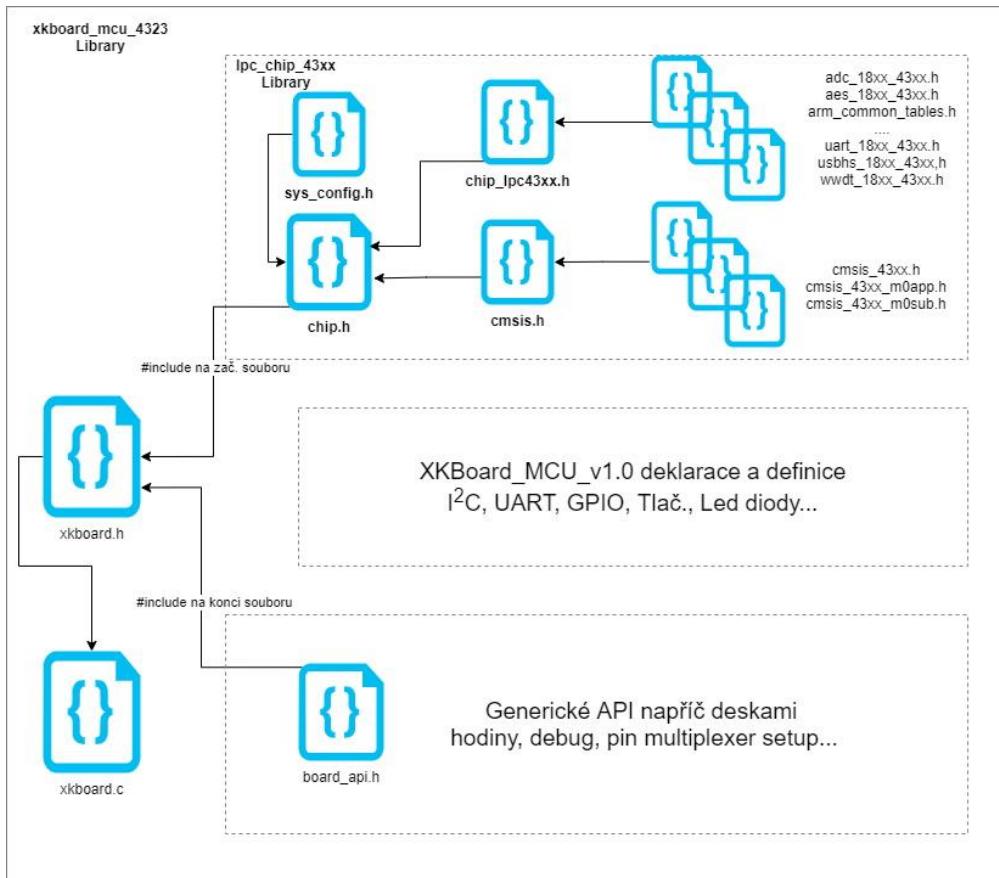
Obrázek 8.2: Nástroje pro správu SDK balíčků.

Mezi nástroje pro práci s SDK podporovanými deskami patří nástroj Pins pro práci a definici funkcí jednotlivých pinů. Poté nástroj pro správu a konfiguraci hodinových rozvodů – Clocks. Dále pak konfigurace periferií a samotného zařízení. Editor také podporuje práci s FreeRTOS operačním systémem a některé příklady v rámci knihovny LPCOpen pro LPC4337 poskytují i příklady práce s FreeRTOS.

8.3 Vytvoření projektu

Vytvoření nového projektu pro naši desku vychází z vytvoření nového LPCOpen C/C++ projektu, který bude používat projekt či knihovnu xkboard_mcu_4327, která rozšiřuje základní knihovnu lpc_chip_43xx pro programování mikrokontroléru LPC43xx od NXP. V rámci použití v IDE MCUXpresso je možné při vytváření nového projektu použít existujících projektů v daném pracovním adresáři či workspace, je-li součástí workspace nebo je importován skrze nástroje IDE. Při zvolení daného projektu se automaticky generuje statickou knihovnu, která je následně linkována s vaším nově vytvořeným projektem. Celý vývojový proces to velmi usnadňuje a urychluje.

Na obrázku níže je možné vidět přibližnou strukturu knihoven a hlavní strukturální části knihovny xkboard_mcu_4327.

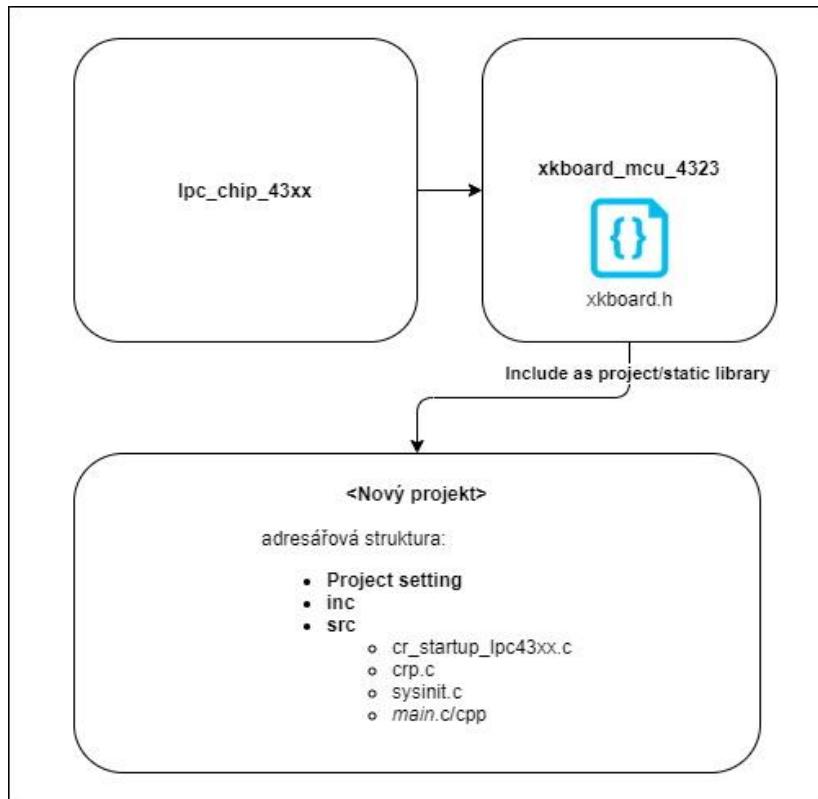


Obrázek 8.3: Schéma závislostí jednotlivých knihoven LPCOpen a knihovny navržené v tomto projektu xkboard_mcu_4327

Jak již bylo naznačeno pro desku s MCU byla výrobcem vytvořena programová abstrakce nad jednotlivými zápisy do konfiguračních registrů SCU a dalších, která umožnuje jednodušší programování a eliminuje nutnost inicializovat jednotlivé I/O piny a komunikační sběrnice či periferie skrze dané registry, která vyžaduje práci s bitovými hodnotami a jejich posuny a je náchylná k chybě. To vše je již automaticky přednastaveno vývojovým prostředím a je zmíněno pro použití bez IDE při překladu tzv. *bare-metal*, protože i v tomto případě je vhodné použít dostupných LPCOpen knihoven. Těchto zdrojových kódů a příkladu desky LPCXpresso4337 bylo využito pro vytvoření API pro naši prototypovou desku „XKBoard MCU v1.0“, aby nebylo nutné inicializovat periferie a jednotlivé piny a sběrnice pin po pinu. Generickým předpisem, který najdeme v souboru `board_api.h` je souborem základních funkcí, které by desky obecně měly poskytovat, aby byla jejich obsluha co nejjednodušší ať už při programování nebo ladění. Jedná se o inicializaci hodin a procesoru či nastavení multiplexoru pro jednotlivé vývody v rámci daného pouzdra a osazených součástek. Implementací zmíněných funkcí v souboru `board_api.h`, které není povinné implementovat, ale je to výhodné pro další práci s již použitelnou platformou, a ne pouze s deskou nespolupracujících součástek. Konkrétní implementací v případě naší platformy jsou v souborech `board.h` a `board.c`, tyto soubory obsahují konkrétní funkce pro inicializaci periferií dle specifikací prototypové desky, dle

daných vývodů a konkrétního modelu mikrokontroléru řady LPC43xx, u prototypové desky je osazen konkrétně LPC4327. Následně si popíšeme stručně popíšeme nově vytvořený projekt a možnosti nastavení konkrétního mikrokontroléru.

Obrázek 8.4 ukazuje základní adresářovou strukturu projektu, kde budou také dostupné nalinkované knihovny *xkboard_mcu_4327*.



Obrázek 8.4: Adresářová struktura nového projektu v MCUEpresso

8.4 Překlad projektu

Překlad aplikace v IDE MCUXpresso probíhá pomocí křížového překladače *arm-none-eabi-gcc* pomocí klasického Makefile souboru generovaného pomocí nástrojů IDE.

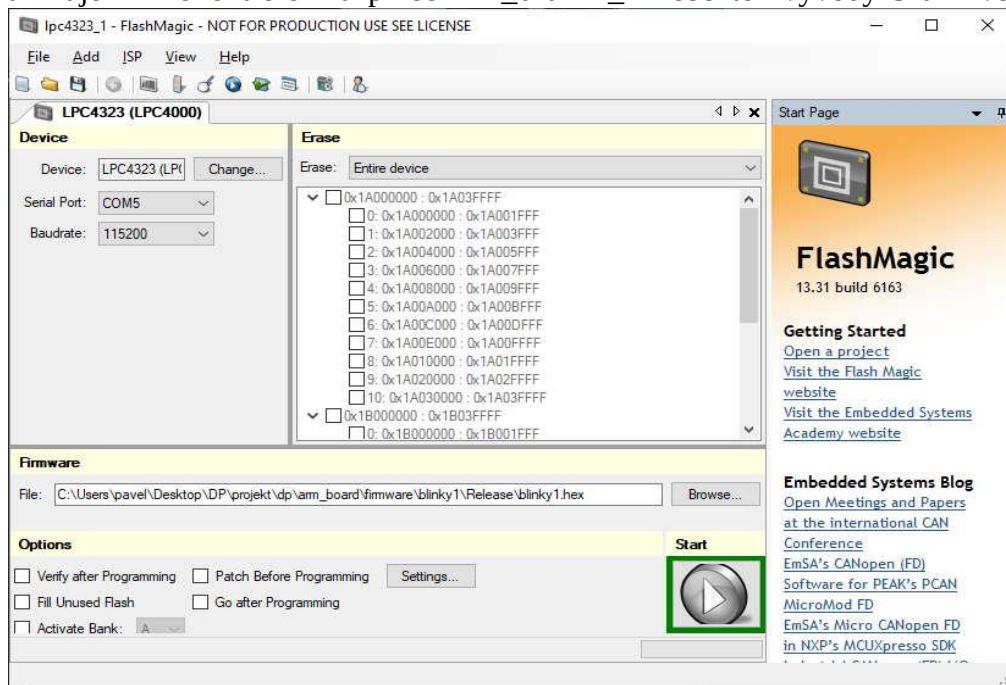
GNU ARM Embedded Toolchain je nástroj pro křížový překlad a obecně se jedná o předpřipravenou sadu nástrojů především pro překlad a ladění kódu v C, C++ a assembleru. Tato sada se zaměřuje především na 32bitové ARM Cortex-A, ARM Cortex-M a Cortex-R architektury. Sada obsahuje GNU kompilér, konkrétně GCC, dále obsahuje Binutils a GDB server. GDB nebo též GNU Project debugger umožnuje vidět co se děje „uvnitř“ jiného programu, který právě vykonáváme na daném mikrokontroléru. V počítači se spuštěn GDB server, ke kterému jsou je připojeno vestavěné zařízení a k tomuto serveru je možné se připojit obvykle pomocí daného vývojového prostředí nebo také přímo pomocí např. nástroje Telnet. Připojíme-li se pomocí terminálu a nástroje Telnet je nutné ladící sezení řídit pomocí textových příkazů, která odpovídají obvyklým ladícím ovládacím prvkům ve vývojovém prostředí. Tedy krokování kódu, zachytávání breakpointů a další různé ladící techniky jako např. výpis registrů nebo zásobníku. Tyto metody umožňují spolu s výpisy a případnými testy odladit aplikaci do detailů.

Boot. režim	P2_9	P2_8	P1_2	P1_1	Popis
USART0	L	L	L	L	UART ISP na vývodech P2_0 a P2_1
SPIFI	L	L	L	H	Quad SPI komp. připojení k osazené flash paměti
EMC 8-bit	L	L	H	L	Možnost paralelního rozhraní, dostupná i 16bitová šířka.
USB0	L	H	L	H	USB Konektor osazený na platformě u odporu označeného R31
SPI (SSP)	L	H	H	H	P3_3 (SSP0_SCK), P3_6 (SSP0_SSEL), P3_7 (SSP0_MISO) , P3_8 (SSP0_MOSI).
USART3	H	L	L	L	

Tabulka 8.1: Bootovací režimy

8.5 Nahrání aplikace

Nahrání aplikace je možné několika způsoby, je možné přes USB rozhraní tzv. drag-and-drop, tedy přetáhnutím požadovaného firmware v binární podobě. Je též možné nahrát firmware přes UART či USB, pokud mikrokontrolér při startu vstoupí do ISP režimu na daném rozhraní. Základní režim, do kterého LPC43xx vstoupí, pokud jsou v režimu bootovaní dedikované vývody v logické nule, do UART ISP režimu. V tomto režimu komunikuje mikrokontrolér na pinech P2_0 a P2_1 nebo též vývody 3 a 4 ve skupině



Obrázek 8.6: Utilita FlashMagic pro nahrávání firmware pomocí UART ISP režimu

MCU_IO. Pro vstup do USB ISP režimu je nutné osadit 12 MHz externí oscilátor a správnou kombinací log. úrovní bootovacích vývodů viz. tabulka 8.1.

8.6 Ladění

Použitý mikrokontrolér umožnuje ladění pomocí několika typů adaptérů, obecně jsou založeny na CMSIS-DAP protokolu a transportní vrstva využívá JTAG nebo SWD protokolu pro navázání spojení s ladícími porty mikrokontrolerů. V rámci této kapitoly si uvedeme nejpoužívanější ladící nástroje pro vestavěná zařízení a firmware pro ladící adaptér umístěný na desce platformy, se kterými budou ladící nástroje komunikovat.

8.6.1 OpenOCD

Volně odstupný projekt pro CMSIS kompatibilní ladění ARM mikrokontroléru, ale obecně se zaměřuje na ladění, programování a boundary-scan testování vestavěných zařízení na bázi mikrokontroléru. Nástroj umožnuje s pomocí malých hw modulů tzv. debug adapterů, které popisují správné zapojení signálů pro různé typy ladících či též trasnportních protokolů a konkrétního zapojení použitého adaptéra či sondy.

Projekt obsahuje velké množství konfigurací podporovaných adaptérů a cílových mikrokontrolérů. Konfigurace adaptéra popisuje tedy potřebné signály a zapojení. Obdobně existují také konfigurace pro cílové mikrokontroléry, které specifikují svoje parametry pro trasnportní protokoly či programování interní flash paměti, tedy počáteční adresy uživatelského prostoru a jeho velikosti. Nástroj umožnuje programování též externích flash pamětí a to CFI-kompatibilních NOR flash pamětí používající sadu instrukcí od Intelu či AMD/Spansion[22].

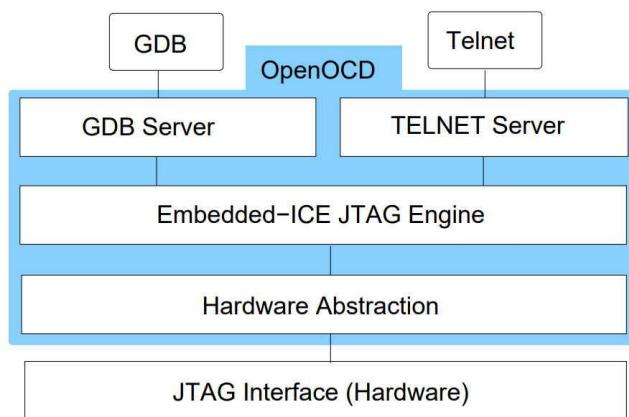
```

Open On-Chip Debugger 0.10.0
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
Info : If you need SWD support, flash KT-Link buffer from https://github.com/bharrisau/busblaster
and use dp_busblaster_kt-link.cfg instead
adapter speed: 5 kHz
jtag
adapter speed: 50 kHz
cortex_m reset_config vectreset
Info : clock speed 50 kHz
Info : JTAG tap: lpc4350.m4 tap/device found: 0x4ba00477 (mfg: 0x23b (ARM Ltd.), part: 0xba00
ver: 0x4)
Info : JTAG tap: lpc4350.m0 tap/device found: 0x0ba01477 (mfg: 0x23b (ARM Ltd.), part: 0xba01
ver: 0x0)
Info : lpc4350.m4: hardware has 6 breakpoints, 4 watchpoints
Info : lpc4350.m0: hardware has 2 breakpoints, 1 watchpoints

```

Obrázek 8.7: Výstup OpenOCD nástroje při spuštění ladění mikrokontroléru LPC4327

Podporovanými transportními protokoly jsou JTAG (IEEE 1149.1)[42] či SWD. Pro JTAG adaptér s FT2232HL existuje již konfigurace odbodného adaptéra se stejným zapojením jako je v této práci použitý externí modul s FT2232HL, konfigurační soubory a jejich struktura bude pospána dále. JTAG adaptér komunikuje s TAP (Test Access Port) každého jádra v mikrokontroléru. TAP umožňují zpracovat speciální instrukce a data pro ladění, tyto porty jsou v rámci MCU vzájemně propojeny. Jako příklad lze uvést LPC4327 použité v této práci, které obsahuje dva tyto porty pro jádro ARM Cortex-M4 a Cortex-M0. Pokud se nástroji podaří úspěšně najít a připojit se k nějakému TAP portu. V našem příkladu se podařilo s frekvencí 50 kHz připojit k dvoum TAP portům. Po úspěšném připojení k portům následuje spuštění GDB serveru s adresou localhost a defaultně na portu 4444. K tomuto ladícímu prostředí se dá připojit pomocí TELNET služby nebo pomocí některé z vybraných IDE jako například Eclipse a na něm založená prostředí, kterým je i MCUEpresso. Dále popsáné prostředí OpenOCD popsáno pomocí diagramu.



Obrázek 8.8: Architektura OpenOCD [22]

8.6.2 PyOCD

Jedná se další volně dostupný projekt, který umožnuje ladění ARM mikrokontrolérů za pomocí CMSIS-DAP protokolu a je kompatibilní se zvoleným ladícím obvodem desky s MCU. Je to modernější alternativa nástroje OpenOCD naspaná v jazyce python[37]. Nástroj umožnuje smazání vnitřní flash paměti a její opětovné nahrání novým programem. Základní instalace nepodporuje přímo LPC4327, ale pyOCD umožnuje rozšíření o požadovaný mikrokontrolér pomocí rozšiřujících MDK5 balíčků s definicemi pinů, hodin a periferií, které volně poskytuje firma Kiel. Tato softwarová platforma nebyla použita v rámci projektu, ale cílem je práce je též informovat o různých alternativách k použitým nástrojům v rámci možné alternativy pro případnou výuku.

8.6.3 Firmware ladícího adaptéru

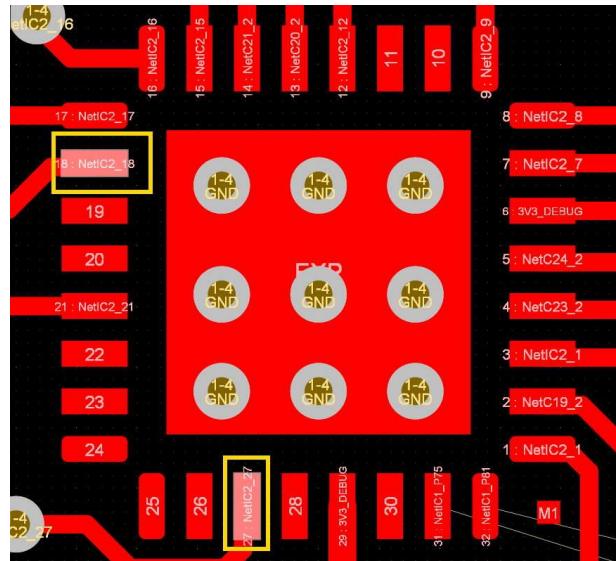
V rámci testování a oživování bylo vyzkoušeno několik různých firmwarů pro ladící obvod na desce MCU, ale kvůli chybě v prvním návrhu, kde nebyly vyvedeny vývody pro rozhraní JTAG nebylo možné tento obvod nakonec použít. Alternativní možností se stalo tedy připojení pomocí externího adaptéru s rozhraním JTAG založeném na FT2232HL, tato možnost se po různých testech ukázala jako nejpoužitelnější a obvod pro ladění u MCU vyžaduje revizi a případně zvolení jiného ladícího obvodu. V rámci této kapitoly tak popíšeme pár vybraných pokusů o zprovoznění implementovaného adaptéru na desce s LPC11U35 a následně funkční obdobu JTAG adaptéru podobnému tomu, který je implementovaný na desce s FPGA. Původní návrh počítal s použitím firmware DAPLink, pro který byla sonda na desce navržena, ale ta se ukázala v kombinaci s použitým LPC4327 jako nepoužitelná. Následoval pokus o doplnění JTAG rozhraní a použití firmwaru IBDAP, poté již následuje úspěšný pokus s použitím externí JTAG sondy s FT2232HL.

8.6.3.1 DAPLink

Zdrojové kódy firmwaru pro ladící obvod jsou volně dostupné jako DAPLink a aktuální verze v době psaní práce firmware 0254. Tento firmware tedy není možné použít z důvodu podpory pouze SWD protokolu. V průběhu pokusů s tímto projektem se ukázalo, že je možné firmware přeložit i s podporou JTAG rozhraní, ale v rámci dokumentace projektu nebylo zcela jasné zda, které piny odpovídají chybějícím signálům JTAG rozhraní a také bylo od tohoto postupu upuštěno.

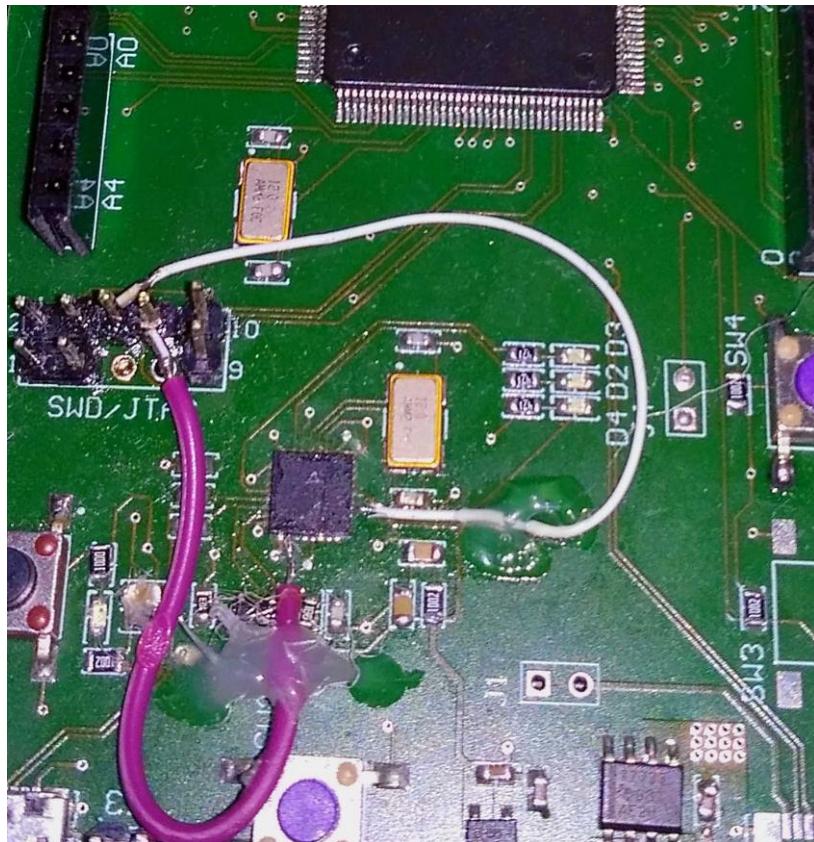
8.6.3.2 IBDAP

Z důvodu nemožnosti připojení ladící sondy pouze přes SWD protokol bez prvního nastavení přes JTAG protokol není možné, protože mikrokontrolér defauktně poskytuje pouze JTAG rozhraní a z něj je možné se přepnout do SWD protokolu, ale není možné použít čistě SWD sondu. Pro použití tohoto firmwaru je nutné vyvést dodatečně další dva piny pro TDI a TDO signály z LPC11U35, a i přes tuto snahu se nepovedlo zprovoznit daný čip jako ladící sondu. Problém se v průběhu práce nepovedlo přesně specifikovat, ale v rovině spekulací to mohlo být způsobeno odporem na daných signálových cestách JTAG rozhraní.



Obrázek 8.9: Výstupy TDO a TDI, které je třeba doplnit k původnímu SWD rozhraní

V rámci testování této možnosti byly na všechny signály přidány 33 Ohmové odpory, které jsou použity ve schématu v IBDAP adaptéru, a přesto se nepodařilo JTAG rozhraní zprovoznit ani připojené LED diody, které mají informovat o nějaké stavu sondy se



Obrázek 8.10: Pokus o doplnění JTAG rozhraní

nerozsvítily, a tedy je možné, že poskytnutý firmware nebyl opravdu funkční a při vlastním překladu tohoto projektu se objevilo tolik problémů, že od tohoto postupu bylo také odstoupeno. Obrázek ukazuje schematicky doplněné signály v motivu desky s MCU, ale

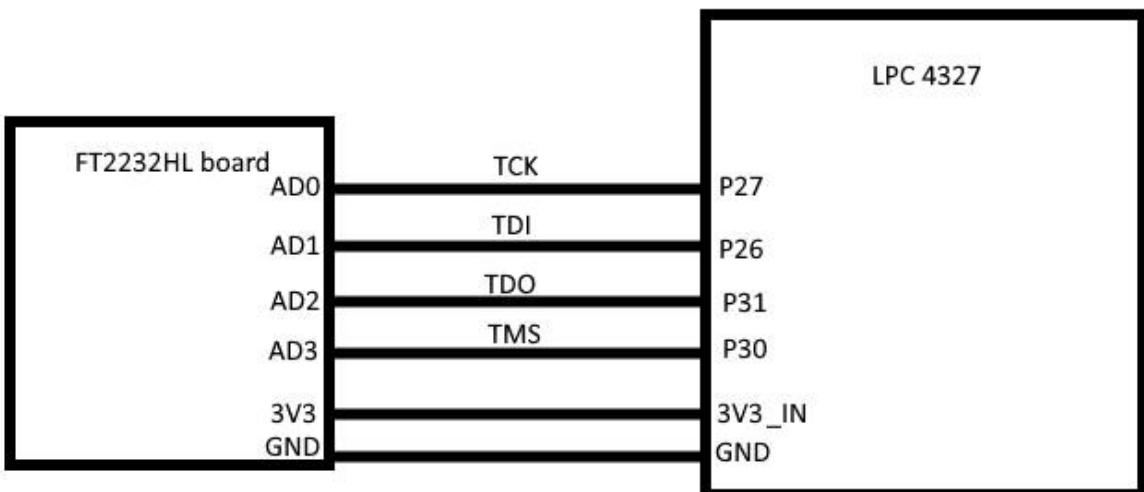
v rámci testování byly piny 18 a 27 vyvedeny pomocí drátků napájením na kontakty QFN pouzdra čipu. Tento pokus je zobrazen na obrázku 8.10. Další pokusy již popisují použití zmíněného externího adaptéru poskytujícího JTAG rozhraní.

8.6.3.3 BlackMagic

Firmware pro tuto sundu lze emulovat i na PC a vytvořit přemostění USB-JTAG rozhraní pomocí externího adaptéru FT2232HL se spuštěnou emulací BlackMagic sondy. Aplikaci lze přeložit změnou cíle hostovaného HW. Takto přeložená aplikace podporuje BMP, STLinkV2/3 nebo námi použitý FTDI-MPSSE adaptér. Tento způsob je možný, ale vzhledem k přímé podpoře FT2232HL v nástroji OpenOCD je to spíše nadbytečné řešení, ale v rámci experimentů bylo též funkční.

8.6.3.4 FT2232H USB-JTAG adaptér

V rámci ověření funkčnosti ladících obvodů hlavního MCU bylo jako náhradní možnosti použito externího obvodu s čipem FT2232HL od FTDI. Tento obvod byl zvolen jako jedna z možných náhrad nevyhovujícího návrhu DAPLink adaptéru. Jedná se o jednoduchou desku tzv. breakout board, osazený pouze FT2232HL, konfigurační EEPROM pamětí a krystalem pro generování hodinového signálu. Firmware zde není nutný, pouze počáteční konfigurace pomocí programu FT_PROG obdobně jako u obvodu s FPGA proto si zde uvedeme pouze schématické zapojení a nástroji FT_PROG se budeme věnovat dále v kapitole s programovým vybavením pro FPGA. Vývody jsou vyvedeny na desce s MCU



Obrázek 8.11: Schéma propojení FT2232HL pomocí JTAG rozhraní

včetně napájení v konektoru označeném jako SWD/JTAG. Tedy je možné napájet platformu s MCU i pomocí adaptéru s FT2232HL, který má vyvedeno napětí 3,3 V. V rámci použití s cílovým mikrokontrolérem je možné s úpravami použít existující konfigurační soubory pro OpenOCD prostředí. Pro adaptér je použit konfigurační soubor dp_bluster.cfg a pro cílový mikrokontrolér je použito konfiguračního souboru pro LPC4350, tedy lpc4350.cfg, který obsahuje definici dvou TAP portů pro obě jádra a poskytuje možnosti nastavit rychlosť a další parametry spojení.

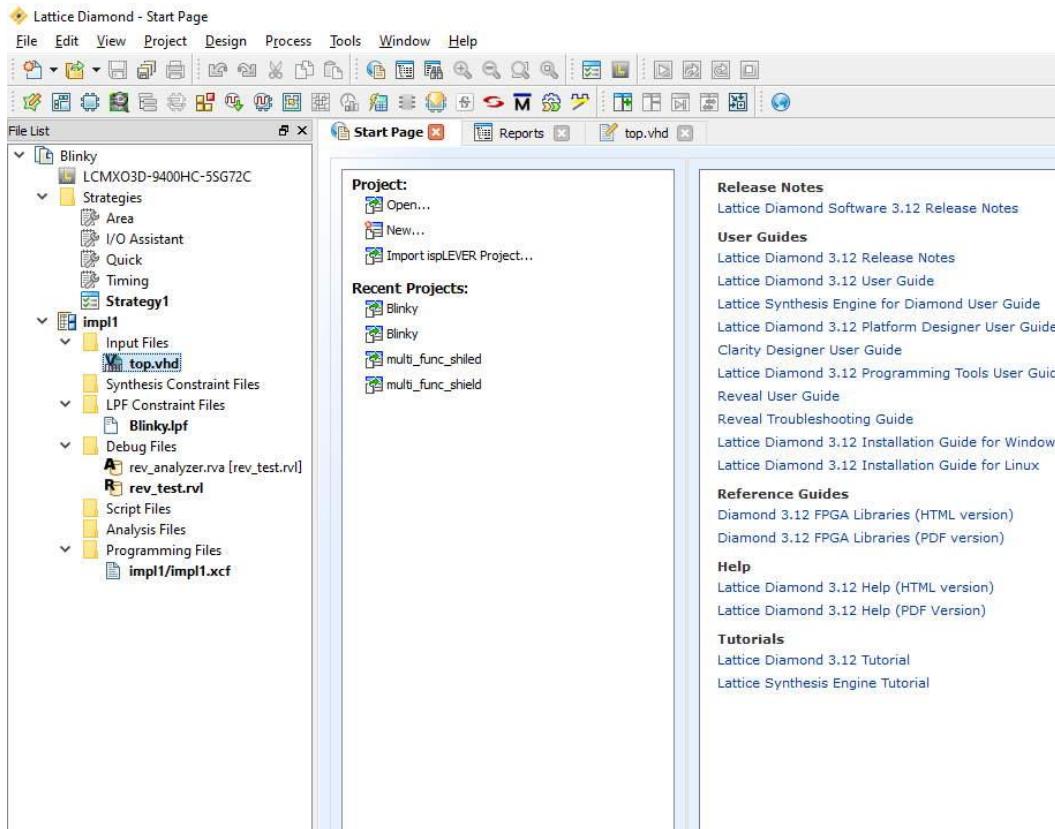
Kapitola 9

Programové vybavení pro platformu s FPGA

V podobném duchu jako jsme si uvedli nástroje pro programování, nahrávání a ladění aplikací pro platformu s mikrokontrolérem uvedeme i nástroje pro práci s deskou s FPGA. V rámci desky je osazen čip FT2232HL od FTDI, který slouží jako JTAG adaptér pro komunikaci s JTAG rozhraním v FPGA. S tímto čipem spojeným s PC hostem pomocí USB umí dále pracovat k FPGA dodávané vývojové prostředí. V této kapitole si představíme konfigurační nástroj pro FT2232HL, který je použit jako JTAG adaptér a dále samotnému vývojovému prostředí *Lattice Diamond*. Uvedeme si nezbytné kroky pro úspěšnou syntézu a nahrání konfiguračního řetězce a budeme se věnovat také ladění našeho návrhu pomocí logického analyzátoru *Reveal*.

9.1 Vývojové prostředí

Na rozdíl od překladu programu pro mikrokontrolér není takřka možné u syntézy konfiguračního řetězce pro FPGA zvolit přístup typu překladu bez IDE pouze pomocí LSE pro syntézu a překladového manažera jako je CMake nebo Automake. *Lattice Diamond* je kompletní vývojovou platformou, která umožní vývojáři kompletně navrhnout, napsat a od



Obrázek 9.1: Představení základního prostředí *Lattice Diamond*

simulovat návrh obvodu pro FPGA čip a následně pomocí dostupných utilit vytvořit

potřebný JEDEC soubor, tedy soubor s konfigurací, který je poté pomocí JTAG rozhraní zapsán do interní flash paměti FPGA nebo přímo do SRAM paměti. Vývojové prostředí si krátce představíme a ukážeme si základní „workflow“ a potřebné utility. Obecně se jedná o klasický projektově orientovaný návrhový editor formou podobná nástroji MCUXpresso z předchozí kapitoly.

9.2 Základní popis prostředí

Základem prostředí je tedy pole „file list“, kde najdeme seznam s jednotlivými soubory a konfigurací zvoleného čipu a omezující podmínky. Tedy projektově orientovaný editor s důrazem na oddělení návrhu v HDL, simulace a dalších analyzačních či konfiguračních souborů. Dalšími nástroji jsou různé pohledy na konfiguraci čipu (Views) nebo nástroj pro programování či logickou analýzu.

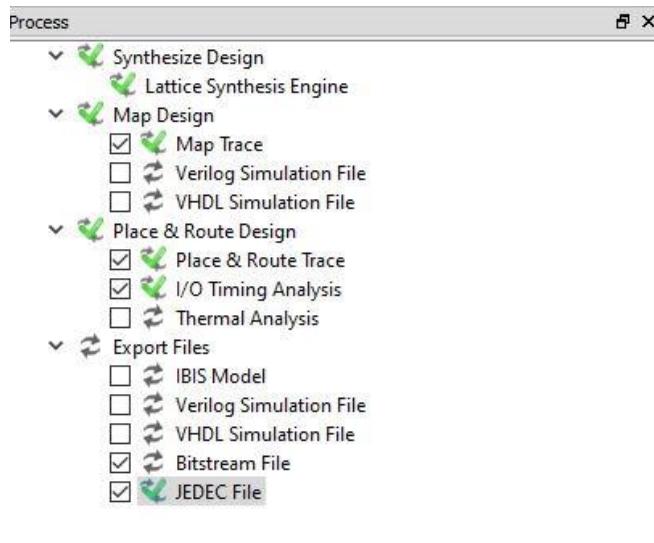
Views jsou základními stavebními bloky celého vývojového prostředí. Různé kontextově odlišená nástrojová okna především pro konfiguraci čipu jako je mapování vstupů a výstupů či časování. Views jsou ve volném překladu „pohledy“ a to v pravém slova smyslu, protože představují právě pohled na čip jako například „Spreadsheet view“, který poskytuje možnost přiřazení signálů jednotlivým vývodům FPGA čipu ve formě více úrovňové tabulky. V jednotlivých bunkách lze zvolit podrobnou konfiguraci daného portu či vývodu. Je možné zvolit např. pull mód, slew rate nebo typ I/O buffer standardu.

	Name	Group By	Pin	BANK	ANK_VCI	VREF	IO_TYPE	PULLMODE
1	All Ports	N/A	N/A	N/A	N/A	N/A		
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1.1	i_clock	N/A	69(69)	0(0)	Auto	N/A	LVCMOS33(LVCMOS33)	DOWN(DOWN)
1.1.2	i_enable	N/A	27(27)	2(2)	Auto	N/A	LVCMOS33(LVCMOS33)	DOWN(DOWN)
1.1.3	i_switch_1	N/A	43(43)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)	DOWN(DOWN)
1.1.4	i_switch_2	N/A	26(26)	2(2)	Auto	N/A	LVCMOS33(LVCMOS33)	DOWN(DOWN)
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	o_led_drive	N/A	30(30)	2(2)	Auto	N/A	LVCMOS33(LVCMOS33)	DOWN(DOWN)

Obrázek 9.2: Spreadsheet View

Dalšími užitečnými „pohledy“ jsou Timing Analysis View tedy časová analýza navrženého obvodu. Netlist View a především Netlist Analyzer umožnuje vidět obvod pomocí klasického obvodového schématu vytvořeného z čipu dostupných primitiv jako jsou sčítáčky, multiplexory či logické funkce. Dalšími pohledy jsou Floorplan view a Physical view, které jsou schopny schematicky či na úrovni fyzických obvodů zobrazit syntetizovaný návrh i s relativním umístěním k pouzdrou čipu. V rámci těchto nástrojů existuje i možnost cross-probe jako při návrhu plošných spojů v prostředí Altium Designer. Lze tak z jednoho pohledu identifikovat primitiva, cesty či obvody v jiné reprezentaci tedy pohledu či view.

Tzv. „Process flow“ je postupné volání nezbytných procesů k syntetizaci, mapování a generování konfiguračního souboru, k jeho správě slouží záložka s procesy, viz. obrázek 9.3. Prvním spouštěným procesem je vždy validace a syntéza HDL souborů a následně translace



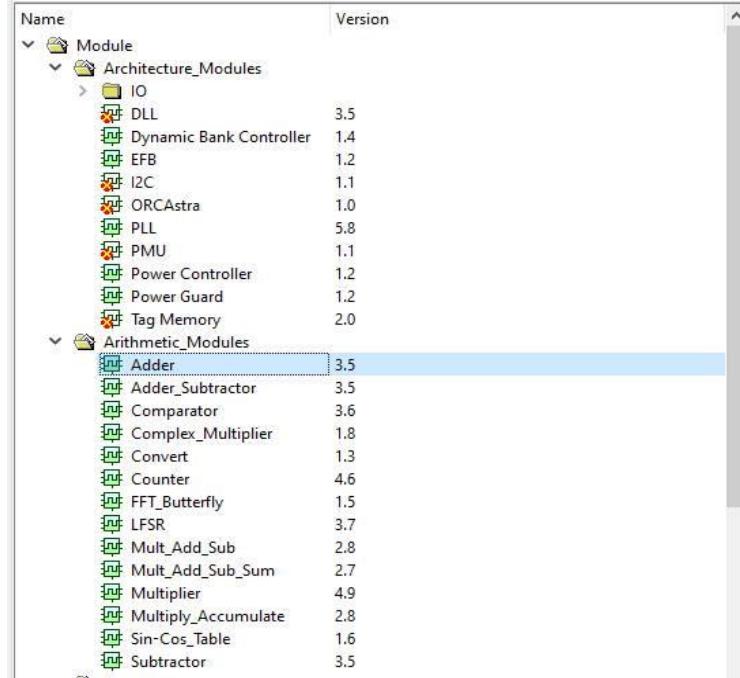
Obrázek 9.3: Okno editoru s procesy pro syntézu konfigurace

do NGD netlistu pomocí syntetizačního nástroje LSE, následuje routování uživatelem zvoleného mapování portů. To je možné pomocí např. Spreadsheet view nakonfigurovat vstupy a výstupy obvodu, tedy jejich mapování na vývody FPGA čipu. Posledním krokem procesu je generování výstupních souborů typu JEDEC nebo Bitstream v binární či hexadecimální podobě.

Type	Input Port	Signal Name	IO TYPE	Pin	Pad Name	Bank	Polarity
		i_switch_s2	LVCMS33	8	PL27A	3	P
				7	PL25B	3	N
				6	PL25A	3	P
				34	PB41B	2	N
				33	PB41A	2	P
				27	PB29B	2	N
				24	PB22B	2	N
				23	PB22A	2	P
				22	PB18B	2	N
				21	PB18A	2	P
				17	PB7B	2	N
				16	PB7A	2	P
				53	PR2A	1	P

Obrázek 9.4: Součást Spreadsheet view, okno pro přiřazení vývodů

IP bloky jsou předpřipravené konfigurovatelné části obvodu plnící nějakou specifickou úlohu[38]. Lattice Semiconductor poskytuje tyto části či bloky v rámci nástroje IPexpress, který nabízí početné množství funkčních bloků od sčítáček přes I2C rozhraní po různé FIFO fronty a složené funkční bloky jako je EFB nebo napájecí kontrolér.



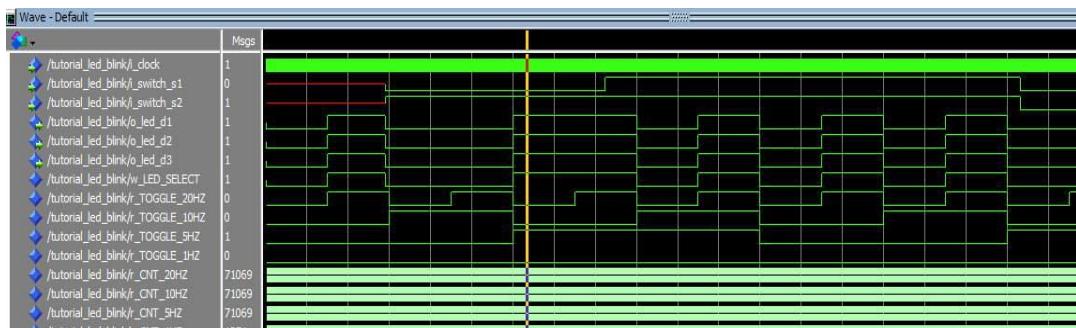
Obrázek 9.5: Nástroj IPexpress s předpřipravenými komponentami

9.3 Vytvoření projektu

K vytvoření nového projektu poskytuje prostředí průvodce (The New Project wizard). Tento průvodce umožní vytvořit základ funkčního projektu. Projekt se v prostředí Lattice Diamond sestává z HDL zdrojových souborů, LPF souborů s omezující podmínkami, EDIF netlist, omezující podmínky pro syntézu, soubory Reveal pro ladění a dalších. Všechny vyjmenované soubory krom vlastního návrhu ve VHDL nebo Verilogu nejsou povinné.

9.4 Simulace

Pro simulaci vytvořeného návrhu je možné použít simulačního průvodce, který vytvoří celý projekt pro spuštění v ModelSIM Lattice edition z vytvořeného návrhu. Nástroj ModelSim je součástí instalace Lattice Diamond.

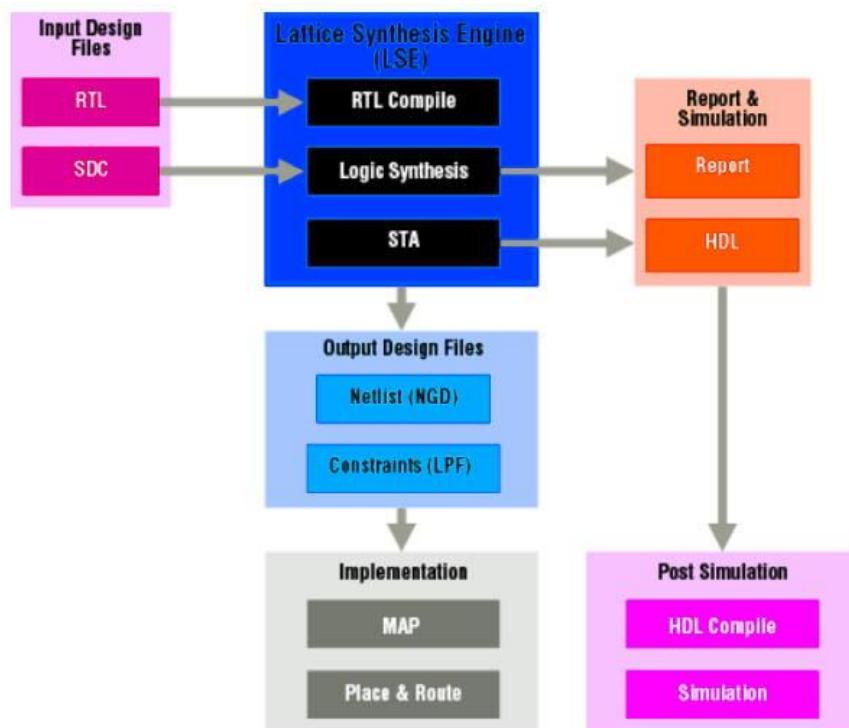


Obrázek 9.6: Simulace příkladu s LED diodami a tlačítky

9.5 Syntéza konfiguračního řetězce pro FPGA

Syntéza konfiguračního řetězce probíhá pomocí LSE (Lattice Synthesis Engine). Jedná se nástroj pro syntézu logických obvodů. Nástroj syntetizuje HDL návrh do netlist souborů vytvořených pomocí základních primitiv specifických pro čipy od firmy Lattice. LSE překládá popis obvodu na optimalizovaný obvod v cílovém FPGA, který je schopen rozeznat a implementovat obvodové prvky vyšší úrovně jako ROM, RAM, konečný stavový automat či různé aritmetické operátory. Na HDL návrh jsou při syntéze kladený určité cíle jako je časová či prostorová omezení. Po provedení optimalizačního procesu LSE vygeneruje netlist ve formátu NGD pro procesy Map Trace, a Place & Route Trace. To vše je provedeno automaticky v rámci prostředí a uživatelsky to nepředstavuje velkou překážku. LSE podporuje následující jazyky [44]:

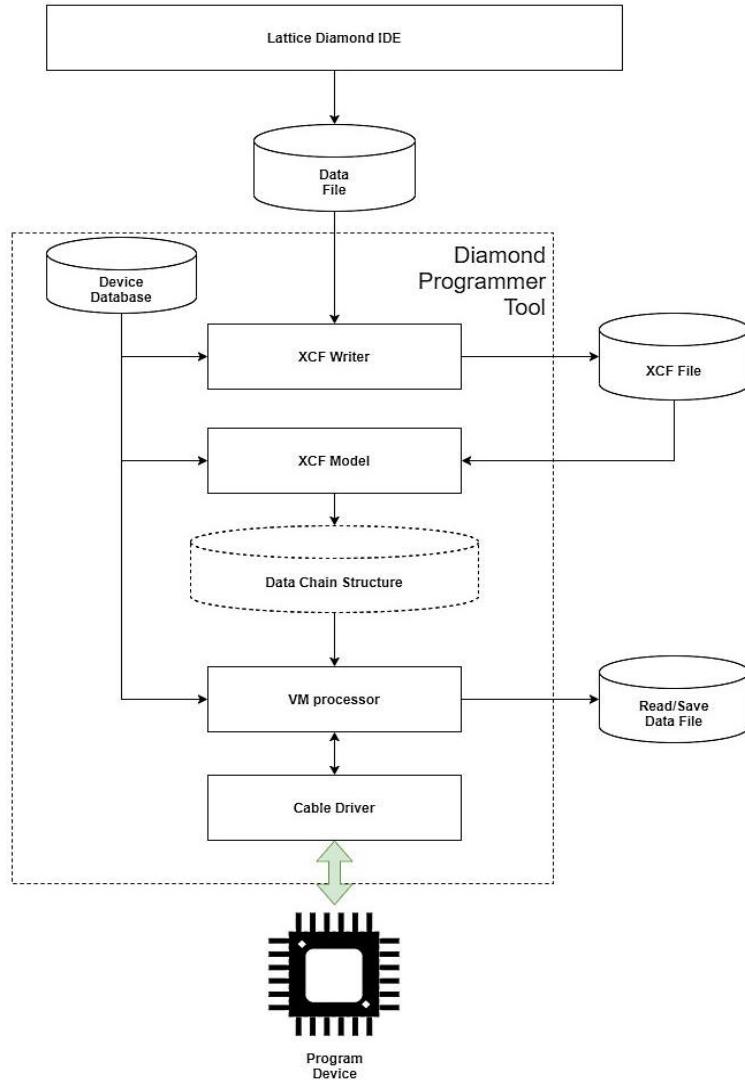
- VHDL 87/93 IEEE 1076 Std
 - *std.numeric_bit*
 - *std.numeric_std*
 - *ieee.std_logic_1164*
 - *ieee.std_logic_arith*
 - *ieee.std_logic_signed/unsigned*
 - *ieee.math_real*
- Verilog 95/2001 IEEE 1364 Std



Obrázek 9.7: Blokové schéma postupu syntézy konfiguračního řetězce
(z.: [44])

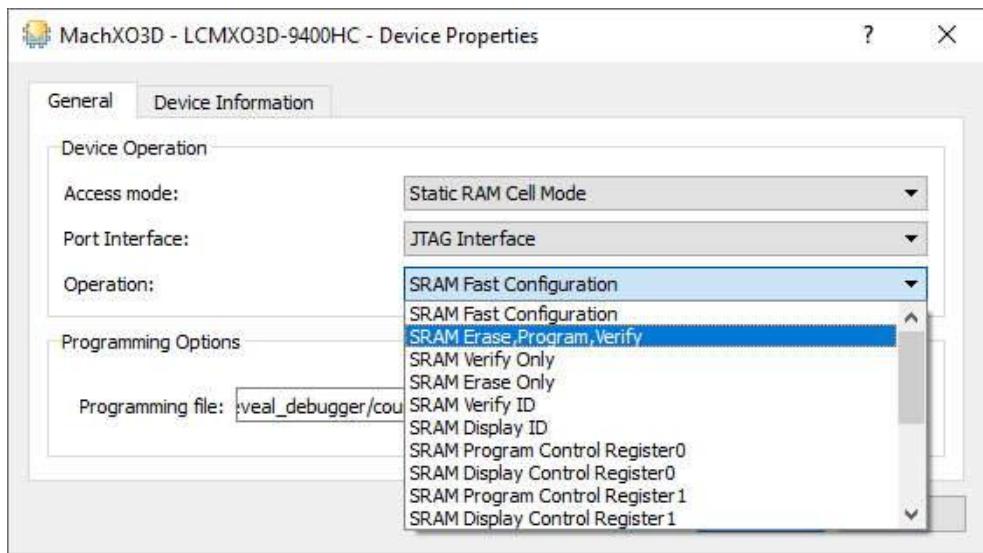
9.6 Nahrání konfigurace

Pro nahrání konfiguračního řetězce a jeho ověření na konkrétní platformě existuje několik možností a záleží také na osazených komponentech. Pomocí nástroje programátoru je možné konfigurovat přímo SRAM paměť v FPGA, tento způsob nahrání pouze do volatilní SRAM paměti neuchová konfigurační řetězec pod odpojení desky od napájení. Tato možnost je tedy



Obrázek 9.8: Schéma funkce programátoru konfigurace FPGA
(z.: [8])

spíše vhodná pro testování a ladění. Druhou a trvalou možností je nahrání do interní non-volatilej paměti FLASH FPGA, ze které bude po zapnutí desky nakonfigurována SRAM paměť, tento způsob konfigurace je též označován SDM (Self-Download Mode). A třetí obdobnou možností je nahrání konfigurace do externí flash paměti, ta je osazena, ale není implementována.



Obrázek 9.9: Okno programátoru a výběr přístupového/konfiguračního režimu

Máme-li syntetizovaný konfigurační řetězec je možné jej nahrát pomocí programovací utility v prostředí Lattice Diamond IDE. Obrázek 9.10 ukazuje jakým způsobem probíhá nahrání syntetizovaného řetězce. Programátor obsahuje databázi čipů poskytovaných výrobcem. Z této databáze je vybráno správné pouzdro, rychlosť a sériu použitého FPGA jak bylo ukázáno v kapitole o vytváření projektu. V rámci uložení konfigurace umožnuje vybrané FPGA uložit dva konfigurační řetězce označované CFG0 a CFG1. Pomocí příkazu „FLASH Erase,Program.Verify“. Tato rutina smaže data obsažená v interní flash paměti na místech CFG0 a CFG1

```

Programmer device database loaded
Starting: "pgr_project open "C:/lscc/diamond/3.12/examples/R

Starting: "pgr_program run"

INFO - Check configuration setup: Start.

INFO - JTAG Chain Verification. No Errors.

INFO - Check configuration setup: Successful.

INFO - Device1 LCMXO3D-9400HC: FLASH Erase,Program,Verify

        "Programming Flash CFG0"

Total erase time: 2640 ms.

Total erase time: 880 ms.

INFO - Operation Done. No errors.

INFO - Elapsed time: 00 min : 30 sec

INFO - Operation: successful.

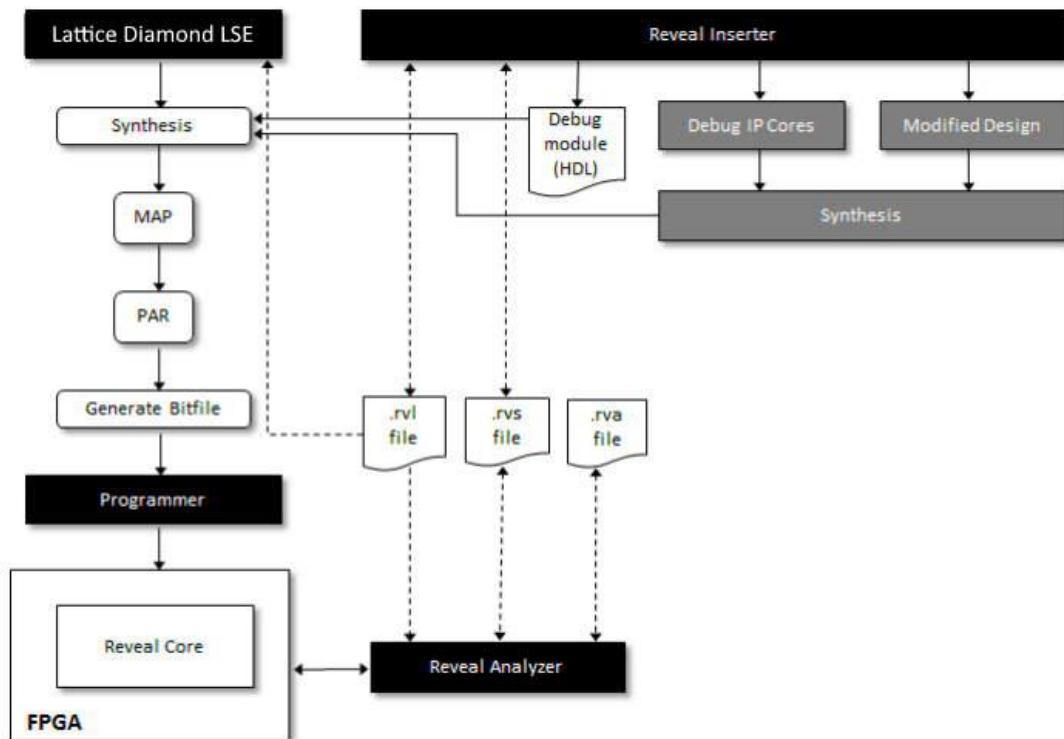
```

Obrázek 9.10: Výstup prostředí při úspěšné konfiguraci FLASH paměti FPGA

9.7 Ladění

Pro ladění obvodu vytvořeného v FPGA se nepoužívá stejných prostředků jako u mikrokontrolerů, kde chceme ladiť program, ale využívá se především logické analýzy signálů, která je schopna přes JTAG adaptér připojené USB host zařízení informovat o logické úrovni v návrhu použitých signálů. Analyzátor vyžaduje dodatečné obvodové zapojení, které je též realizováno v FPGA. K tému účelům slouží v prostředí *Lattice Diamond* nástroje *Reveal Inserter* a *Reveal Analyzer* [43]. JTAG adaptér realizovaný v FPGA komunikuje s JTAG rozhraním FT2232HL, a to zasílá informace z analyzátora přes USB rozhraní. *Reveal Inserter* je nástroj pro vložení a konfiguraci bloku obvodu s logickým analyzátorem.

Reveal Analyzer a obecně logické analyzátory jsou jednou z nejpoužívanějších metod ladění logických signálů vedoucích z daného čipu v čase. Tak dokážeme identifikovat chyby v námi navržených obvodech v rámci návrhu v FPGA, ale i v návaznosti na další externí obvody, ke kterém je FPGA připojeno. Pro analýzu signálů vyvedených z FPGA či vnitřních signálů je nutné tyto signály připojit k obvodu logického analyzátoru, který vzorkuje dané signály a pomocí JTAG adaptéra též realizovaného v FPGA informuje nástroj *Reveal Analyzer*, který umí získané vzorky vizualizovat na časové ose. Analyzátor umožnuje konfiguraci spouštěcích událostí (Trigger) skrze JTAG rozhraní připojeného přímo do jádra *Reveal analyzátoru* v obvodu. Data zachycená analyzátorem mohou být uložena ve formátu .vcd, který může být importován např. do programu ModelSim.



Obrázek 9.11: Schéma použití *Reveal analyzátoru* (z.: [43])

Kapitola 10

Demonstrace

Demonstraci funkčnosti provedeme naprogramováním cílového MCU či FPGA pomocí podpůrných obvodů nezávisle na sobě, tedy bez propojení platforem. Jako demonstrace bude sloužit různá obměna aplikace pro rozsvícení LED diod na deskách. Dále nás bude zajímat propojení obou desek pomocí skupiny vývodů označených na desce jako `FPGA_CONF`. Pomocí tohoto konektoru by mělo být možné nejen FPGA naprogramovat pomocí MCU, ale také komunikovat a přenosit data mezi oběma deskami pomocí I²C či SPI sběrnice. V posledním demonstračním příkladu se budeme zabývat použitím rozšiřujícího modulu pro Arduino platformu.

Demonstrační příklady pro LPC4327 vychází ze sady příkladů od výrobce mikrokontroléru⁴. Na základě knihoven pro čipy řady LPC43xx a knihovny pro navrhovanou desku, která je popsána výše je možné jednoduše používat prakticky všechny dostupné periferie, časovače a podpůrné obvody integrované přímo v mikrokontroléru.

Pro FPGA poskytuje výrobce různé vlastní demonstrační použití především výrobci specifických komponent. V rámci FPGA bylo využito příkladu použití EFB modulu a programování FPGA pomocí I²C sběrnice.

⁴ <https://www.nxp.com/design/microcontrollers-developer-resources/lpcopen-libraries-and-examples:LPC-OPEN-LIBRARIES>

10.1 Příklad signalizační LED diody a tlačítka

Klasický příklad pro demonstraci základní funkčnosti obvodu. Který prověří základní periferie jako LED diody a tlačítka.

10.1.1 Platforma s MCU

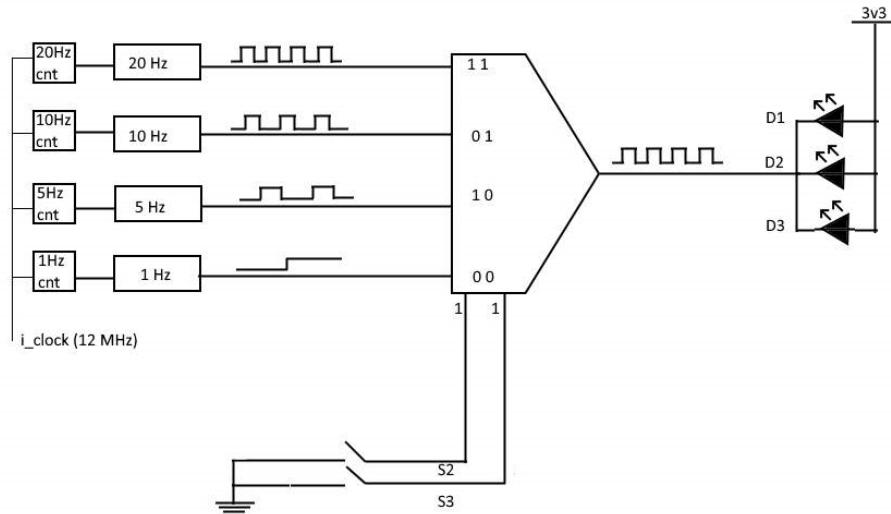
Za účelem naprogramování MCU za pomocí ladícího obvodu je zapotřebí nahrát správný firmware do MCU ladícího obvodu. Případně firmware nahrát pomocí externího adaptéru nebo s použitím sériové linky v ISP režimu. Deska s MCU poskytuje tři LED diody pro obecné použití připojené na vývodech P6_7, P6_9, P6_11. LED diody budou blikat s frekvencí 1 Hz. Příklad využívá také tlačítek osazených pod identifikátory SW3 a SW4. Ty jsou připojeny na piny P6_3 a P6_6. Tlačítka jsou použita pro změnu frekvence blikání. Jak již bylo naznačeno výše, adresář src, kde se nachází zdrojové soubory, obsahuje dané API pro DPS označenou jako „XKboard MCU 1.0“ tedy soubory *board.c* a *board.h* v podobě knihovny. Aplikaci přeloženou pomocí vývojového prostředí, křížového překladače a ladící konfigurace je možné testování pomocí GDB serveru spuštěného v rámci OpenOCD prostředí. Aplikace pro mikrokontrolér využívá funkce *Board_Init()*, která je volána vždy a obsahuje volání pro inicializaci ladícího výstupu na UART0, nastavení vývodů dostupných na desce na základní GPIO funkci. Inicializace LED diod a tlačítek. Rozblikání LED diod je možné např. pomocí pomocné funkce *Board_LED_Toggle(pin)*, která invertuje stav zadaného vývodu. V tomto případě stačí použít číslování 0-2.



Obrázek 10.1: Rozsvícení signalizačních LED diod na platformě s MCU

10.1.2 Platforma s FPGA

Příklad s blikáním osazeným LED diod v závislosti na stisknutých tlačítcích je na FPGA realizován jako obvod s čítači simulujícími různou frekvenci blikání a pomocí tlačítek a multiplexoru je volen zdroj blikání, který je přiveden na všechny tři osazené LED diody D1-D3. Obvod v FPGA je ilustrován na obrázku 10.2. Tento obvod je následně doplněn o



Obrázek 10.2: Schéma obvodu realizovaného v FPGA

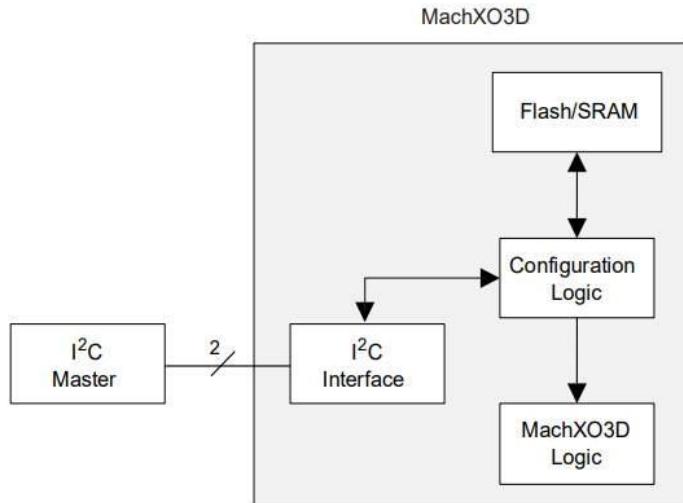
logický analyzátor Reveal, aby bylo možné sledovat jednotlivé signály použité v obvodu. Zachycená změna Reveal analyzátorem na obrázku 10.3 ukazuje moment, kdy bylo stisknuto tlačítko S1 a je zvolen výstup s 10 Hz frekvencí blikání.



Obrázek 10.3: Výstup z Reveal analyzátoru

10.2 Příklad I²C rozhraní pro konfiguraci FPGA

Konfigurace FPGA čipu je, jak již bylo zmíněno možná přes I²C v režimu Slave s pomocí desky s MCU v roli Mastera. Tento příklad ukazuje především implementaci na straně MCU, protože to je v roli Master zařízení a vyžaduje implementaci sady příkazů, které řídí přes zmíněnou sběrnici nahrání konfiguračního řetězce. Ze strany FGPA se jedná o konfigurační rutinu a je tedy nativně podporována komunikace pro nenakonfigurované FPGA přes tuto sběrnici pomocí sady zmíněných příkazů.



Obrázek 10.4: Schéma propojení platforem s FPGA a MCU v rámci konfigurace

10.1.1 Platforma s MCU

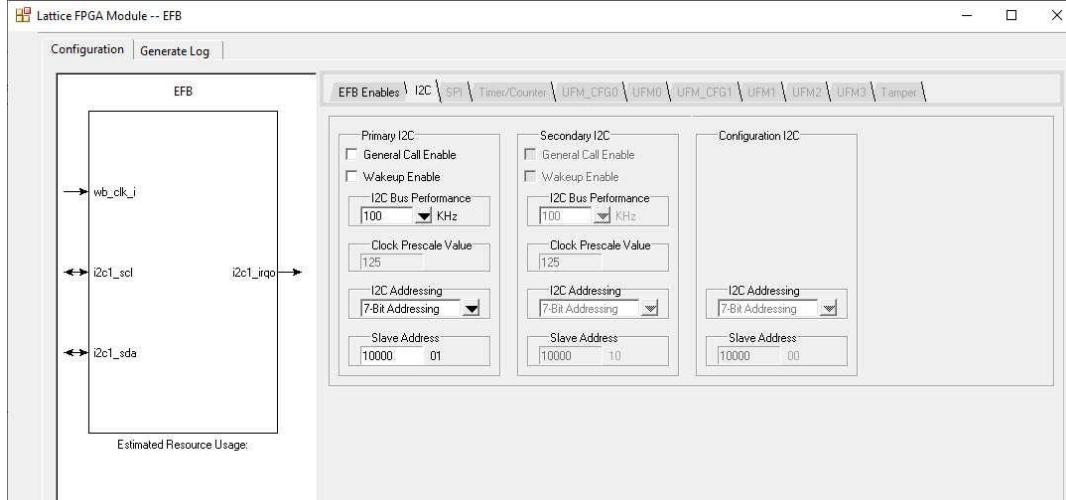
Mikrokontrolér musí jako Master zahájit komunikaci a pomocí sady programovacích instrukcí, které budou dále popsány, může získávat různé informace o zařízení a hlavní účelem je samotná konfigurace vnitřní FLASH paměti. Implementaci těchto příkazů je možné nalézt v souboru *machxo.c* a definice v hlavičkovém souboru *machxo.h*. Všechny funkce v těchto souborech slouží pouze k přiřazením a vytvořením datového bufferu k odeslání pomocí I²C zprávy. I²C Master funguje na principu přerušení, aby nemusel neustále sondovat, jestli nejsou dostupná nějaká data na lince. Soubor s hlavní smyčkou pokrývá samotnou komunikaci a odesílání zpráv FPGA jako slave zařízení.

V tomto příkladu si u zvoleného mikrokontroléru ukážeme použití NVIC kontroléru přerušení a inicializaci I²C sběrnice, pro konfiguraci, a sériové linky UART pro zobrazení výstupu z příkazů.

Nyní si uvedeme základní přehled použitých příkazů a jejich účel a příkazy potřebné pro různé úkony jako např. konfigurace FPGA čipu nahráním konfiguračního řetězce do interní flash paměti FPGA.

Na uvedených příkazech jsou založeny jednotlivé úkony s FPGA. Tedy například pro vypsání detailů o FPGA je možné použít metodu ze souboru s hlavní smyčkou, tedy *printDetails()* poskytující výpis ID zařízení, status zařízení nebo čist různé části paměti jako *Feature Row*.

Konfigurace je hlavním úkonem, který budeme po rozhrání požadovat. Obecně je zapotřebí, aby pro konfiguraci přešlo FPGA do offline režimu, kdy není vykonávána žádná logika, protože se mění obsah paměti SRAM či flash a tím i samotný realizovaný obvod. Tedy existuje příkaz `cmd_enableConfigOffline()`, který předvyplní buffer pro odeslání dat



Obrázek 10.6: Konfigurace EFB bloku v rámci nástroje IPxpress

požadovanými hodnotami. Poté je možné pomocí funkce `cmd_erase(flag)` a na základě flagu (`MACHXO_ERASE_CONFIG_FLASH | MACHXO_ERASE_UFM`) smazat jak CFG sektory, tak EFM sektory v interní flash paměti. Kontrolu dokončení je možné rozpoznat pomocí tzv. poolingu pomocí `cmd_isBusy()`, který čte příznakový bit a dle toho je možné zjistit konec mazání paměti. Poté je možné nahrát nový konfigurační soubor jehož zdroj může být externí flash paměť, nebo je možné nahrát binární soubor v hexadecimálním tvaru přes sériovou linku vytvořenou skrze USB-UART(RS232) adaptéru, přes různá další připojitelná média například pomocí USB rozhraní nebo SD-kartu pomocí rozšiřujícího Arduino modulu. Na obrázku níže je možné vidět výstup z čistě vymazaného zařízení. Náhrání do flash paměti FPGA není plně funkční kvůli nerealizovanému rozhraní k nějakému většímu paměťovému

Device ID:	0x212E3043
User Code:	0x00000000
Feature Row:	0x00000000000000000000
Feature Bits:	0xFFFF
Status:	0x00200000

Obrázek 10.5: Výstup funkce `printDetails()`

médium. Ale jak bylo zmíněno je možné například nahrát konfiguraci i přes UART0 z PC do MCU a z něj poté FPGA přes I²C, je to možnost zbytečně komplikovaná, ale též možná.

10.1.2 Platforma s FPGA

Aby bylo možné naprogramovat FPGA pomocí I²C je třeba splnit jednu z následujících podmínek. První podmínka je, že programujeme zařízení, kterému byla vymazána interní flash paměť a konkrétně sektory s konfigurací CFG0 a CFG1 (též nově osazený FPGA čip), a tedy zařízení nevstoupí do uživatelského režimu, protože neexistuje žádná funkční

konfigurace, se kterou by mohlo FPGA vstoupit do uživatelského režimu. Další možností je použití EFB bloku a nastavení v rámci vývojové prostředí Lattice Diamond. V pohledu Spreadsheet view>Global Preferences je nutné povolit položku I2C_PORT. Toto nastavení je prováděno automaticky při konfiguraci EFB bloku, ke kterému je zapotřebí připojit hodinový signál na port wb_clk_i, který by měl být alespoň 7,5násobkem použité taktovací frekvence I²C kanálu. V základním nastavení EFB bloku je pro I²C sběrnici nastaven hodinový signál 100 kHz a je použito 7bitotové adresování. Požadovaná adresa pro konfiguraci je přednastavena na „1000000“ v binární podobě v rámci adresování jsou důležité poslední dva byty, tedy pokud máme 7bitovou adresu „aaaaabb“, kde hodnota „a“ je uživatelsky konfigurovatelná a poslední dva „b“ byty jsou určeny pevně a jedná se o jednotlivé I²C kontroléry obsažené v FPGA tedy primární, sekundární („aaaaaa01“ a „aaaaaa10“) a třetí použitá adresa je právě ke konfiguraci zařízení („aaaaaa00“). V rámci použití EFB bloku je možné také s tímto blokem komunikovat a vytvořit tak spojení mezi tímto blokem a uživatelem definovanou logikou, který implementuje Wishbone Master rozhraní. Pro tento případ je nutné I²C v EFB bloku inicializovat pro práci jako primary či secondary user. To přidá bloku logiku Wishbone slave kontroléru, který je nutný pro komunikaci s EFB blokem potažmo i s ESB blokem. Pro realizaci Wishbone Master rozhraní, lze využít soft-core 8bitového mikrokontroléru Mico8 či 32bitového Mico32, bohužel není zvolená série MachXO3D přímo podporována, ale Mico8 jádro podporuje MachXO3L sérii. Jádro nebylo vyzkoušeno v rámci tohoto projektu, ale výrobce poskytuje celou sadu nástrojů pro křízový překlad programu pro tento mikrokontrolér s integrovaným Wishbone rozhraním, na který primárně odkazuje i dokumentace k sérii MachXO3D. Další možností obsluhy je vlastní implementace stavového automatu mastera Wishbone sběrnice. I přes zmínku o Wishbone rozhraní pro připojení uživatelské logiky k EFB bloku není tato část pro samotnou konfiguraci nutná, pro konfiguraci stačí pouze připojit specifikovaná hodinový signál a připojit FPGA fyzicky k I²C Masterovi.

10.3 Příklad s rozšiřujícím modulem

Na tomto příkladu si ukážeme základní použití rozšiřujícího modulu kompatibilním s Arduino vývody na DPS jak s MCU, tak s FPGA. Pro tuto demonstraci byl vybrán multifunkční modul ze sedmi-segmentovým displejem, bzučákem, signalizačními led a tlačítky nebo otočným potenciometrem. Funkčnost modulu opět ověříme jednoduchým rozblikáním LED diod v závislosti na stisknutých tlačítcích a také vypsáním nějakého textu na LED zobrazovač. Také je možné u platformy s MCU měřit napětí na otočném potenciometru.

Signalizační LED diody jsou připojeny na vývody 10–13 v rámci označení Arduino vývodů, a i další vývody si popíšeme v tomto značení. Bzučák osazený na modulu je připojen k vývodu 3. Jedná se o pasivní součástku a je tak nutné ton pro bzučák generovat např.

pomocí PWM signálu. Dalšími osazenou periferií je LCD zobrazovač, který je postaven ze čtyř sedmi-segmentových displejů a jsou řízeny dvojící obvodů 74HC595d. Jmenovaný obvod je posuvným registrem s přidaným paměťovým registrem a 3 stavovými výstupy. Tyto registry slouží k uložení dat o tom, které segmenty mají svítit, které zůstat zhasnuté. Registry jsou propojeny a je možné je programovat pomocí tří vývodů. Datový vývod vyveden na 8, hodinový signál – 7 a signál pro zahájení komunikace – 4. Komunikace probíhá tak, že signálem na vývodu 4 je zahájen zápis do registrů a poté se sériově zapisují data přes vývod 8. Příkaz se skládá ze dvou bajtů. První bajt určuje, které segmenty se mají rozsvítit či zhasnout a druhý bajt říká, který ze čtveřice sedmi segmentových displejů bude zvolen. V jednu chvíli je tak možné rozsvítit pouze jeden z displejů zobrazovače, ale pokud budeme nastavovat požadovaná písmena či čísla s dostatečnou rychlostí budou se zobrazovaná



Obrázek 10.7: Multifunkční rozšiřující modul

písmena jevit jako, že se zobrazují současně.

10.3.1 Platforma s MCU

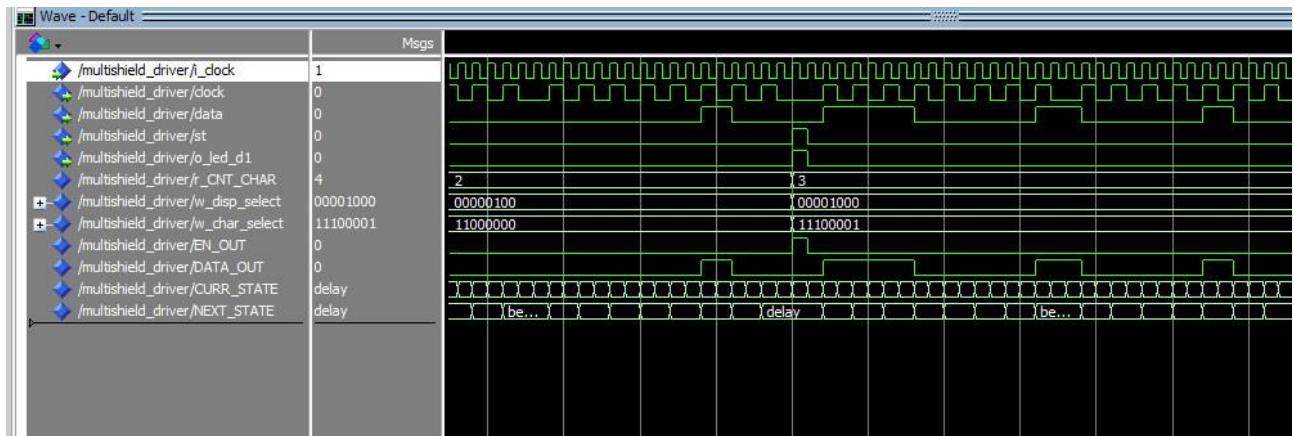
V rámci platformy s MCU jsou příklady s blikáním diod a snímáním tlačítek již opakováním předchozího příkladu a zaměříme se tedy hlavně na implementaci logiky pro LED zobrazovač.

Pro implementaci zobrazovače použijeme klasický GPIO funkcionality vývodů. Pro tento modul existuje implementace pro obvod Arduino a je možné tedy se částečně inspirovat touto implementací a budeme se držet i pojmenování různých funkcí použitých v Arduino knihovnách. Tou hlavní funkcí pro sériový zápis dat na datový výstup je `shiftOut()`⁵ funkce. Tato funkce v cyklu projde zadáný bajt, který je právě odesílán a bit po bitu jej zapíše na datový vodič a zároveň generuje sestupnou hranu hodinového signálu mezi jednotlivými byty, které se tak propíšou do registru. Zápis probíhá dle uvedeného předpisu výše. Je zahájen

⁵ <https://forum.arduino.cc/t/anyone-know-the-code-for-the-shiftout-function/45638>

změnou logické úrovně povolovacího signálu do logické nuly. Poté jsou pomocí funkce shiftOut() zapsána data a adresa pro jednotlivé displeje mezi každým písmenem změníme logickou hodnotu povolovacího signálu, abychom odělily jednotlivé dvojice hodnot data, adresy.

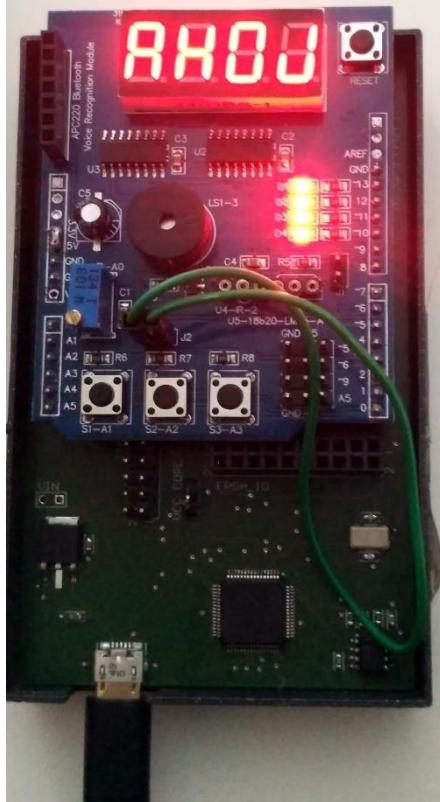
10.3.2 Platforma s FPGA



Obrázek 10.8: Simulace popsaného protokolu pro řízení registrů 74HC595d v FPGA

U FPGA si také ukážeme především obvod pro řízení LED zobrazovače, protože připojení tlačítek a LED diod jsme již věnovali v prvním příkladu. Pro kontrolní signály si vytvoříme výstupní porty, které potom přiřadíme vstupům ve skupině Arduino vývodů, jak je uvedeno u specifikace zobrazovače výše. Pro toto chování vytvoříme behaviorální popis a v něm proces, který bude na základě stavové proměnné CURR_STATE řídit zasílání zvoleného písmene a adresy sedmi segmentového displeje bit po bitu, tedy stav po stavu. Odeslání dvou osmi bitových hodnot je odděleno pomocí povolovacího signálu značeného v simulaci jako *st*. Každý stav pro odeslání jednotlivého bitu je rozdělen na dva stavy, kdy je v rámci inicializace příkazu a vystavení dat na vývod označený na obrázku jako *data* a generuje se změna stavu hodinového signálu *clock* v rámci zmíněného prvního a druhého kroku, kvůli dvěma krokům je třeba uložit následující stav potažmo bit, který má být odeslán, po vystavení dat jsou data nasunuta do registru se změnou hodin do logické nuly.

Pro výběr jednotlivých písmen a jejich adres slouží multiplexory, které na základě čtyř hodnotového čítače zvolí multiplexory jedno ze čtyř písmen z výsledného slova „AHOJ“ a příslušnou adresu.



Obrázek 10.9: Demonstrační použití multifunkčního rozšiřujícího modulu na platformě s FPGA

10.4 Příklad ovládání servomotorů

Tento příklad ukazuje možnost propojení obou navržených platform. Platforma s MCU bude snímat pomocí ADC převodníku hodnotu os tvořených potenciometry v joysticku a bude ji zasílat s pomocí jednoduchého sériového rozhraní, které je podobné tomu z předchozího příkladu se sedmi segmentovým zobrazovačem. V tomto případě se omezíme na zjednodušený příkaz pouze o 8 bitech, který bude obsahovat 3 bity počínaje MSB bitem představující adresu jednoho ze servomotorů, které pohybují s ramenem vytvořeným na 3D tiskárně, a pozici v rozmezí 0-180 stupňů s rozlišením 11 stupňů ($180/2^5=11$) ve spodních 5 bitech příkazu. Zmíněný 8bitový příkaz je poslan FPGA čipu přes uživatelsky specifikované vývody. Stejně jako v příkladu se zobrazovačem využijeme pouze tři signály a to datový, hodinový a povolovací.

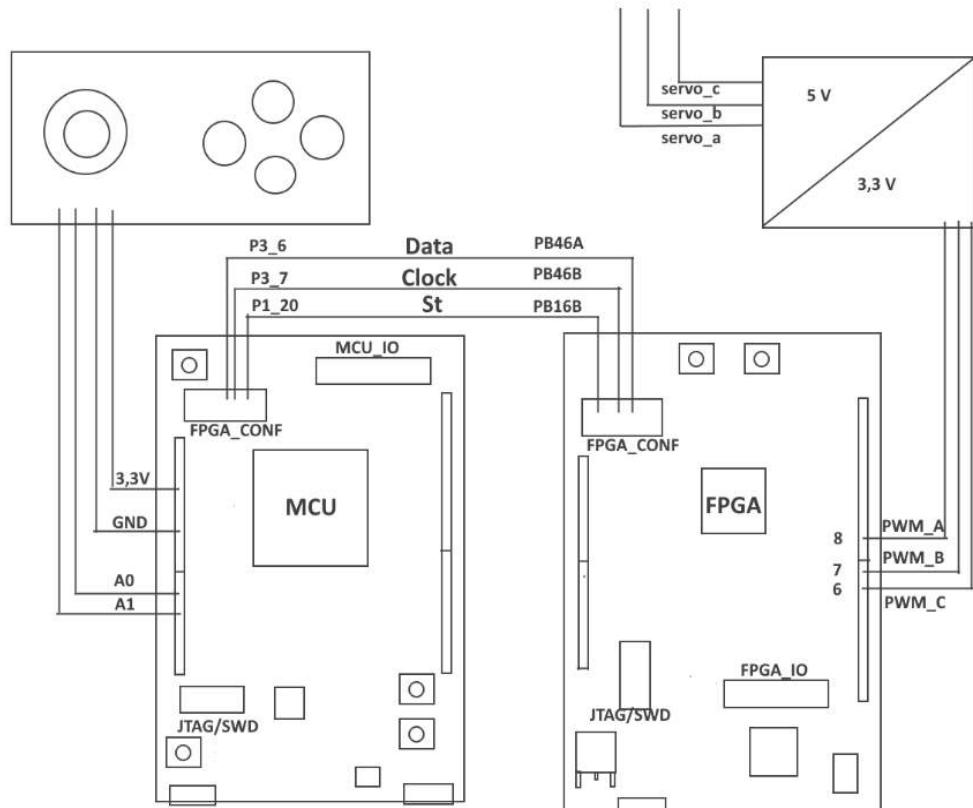
V tomto příkladu tedy je MCU v roli Master, který zasílá příkazy s nastavením PWM signálu FPGA jako Slave zařízení. V rámci MCU tedy čteme hodnoty z ADC převodníku z kanálu 0 a 1. Ty jsou poté odesílány do FPGA pomocí zmíněných 8bitových příkazů, které je zpracovává a mění střidu PWM signálů A-C, které řídí mechanismus ramene.

Zapojení využívá rozšiřujícího modulu pro Arduino s osazeným joystickem a tlačítka s označením „Joystick Shield V1.A“, který je připojen k platformě s MCU, protože FPGA nemá integrovaný ADC převodník. A FPGA je zase využito pro vytváření PWM signálů, protože se jedná o jednoduché čítače, tedy implementace je jednodušší než na MCU. Jedná se tak o vhodný příklad doplnění funkcionality obou platforem. Pro ovládání servomotorů typu MG995 je nutné minimální napětí 5 V a vyžaduje i PWM s těmito napěťovými

úrovněmi, avšak obě platformy mají operační napětí 3,3 V a nestačí na řízení servomotorů. Z tohoto důvodu je použito převodníku napěťových úrovní a externího 5 V napájení. Servomotory lze řídit pomocí střídy PWM signálu o frekvenci obvykle 50 Hz. Tato frekvence znamená periodu 20 milisekund, a pro řízení servomotoru je potřeba střída v rozmezí obvykle 0,5 – 1,5 milisekund. Krajiná hodnota znamená úplné natočení 0 či 180 stupňů. Natočení do středu tedy dosáhneme střídou okolo 1ms. Uvadíme přibližné údaje, protože existuje mnoho druhů napodobenin a je možné, že se hodnoty liší. Máme-li tedy zdrojový signál pro PWM čítače 64 kHz tak jeden tik hodin zabere 1/64000 sekund. PWM čítač má rozsah 0 až frekvencí danou maximální hodnotu střídy 20 milisekund, pro tyto hodnoty je nutný čítač od 0 do 1280. Tedy například pro středové natočení o se střídou 1ms je zapotřebí na PWM držet signál v logické jedničce po 64 tiků, které odpovídají 1/20 střídy. Schéma zapojení na obrázku 10.10 ukazuje kompletní propojení a mapování použitých vývodů.

10.4.1 Platforma s MCU

Z hlediska firmware pro LPC4327 je nutné inicializovat ADC převodník. V tomto příkladu používáme ADC0 a kanály 0 a 1. Ty jsou připojeny k vývodům A0 a A1 ve skupině vývodů kompatibilních s Arduino rozšiřujícími moduly. Pro vyčítání hodnot z registrů modulu ADC převodníku existuje několik možností jako například tzv. pooling hodnot, pomocí přerušení nebo DMA přenosem. V tomto příkladu je využito poolingu hodnoty na základě stavu



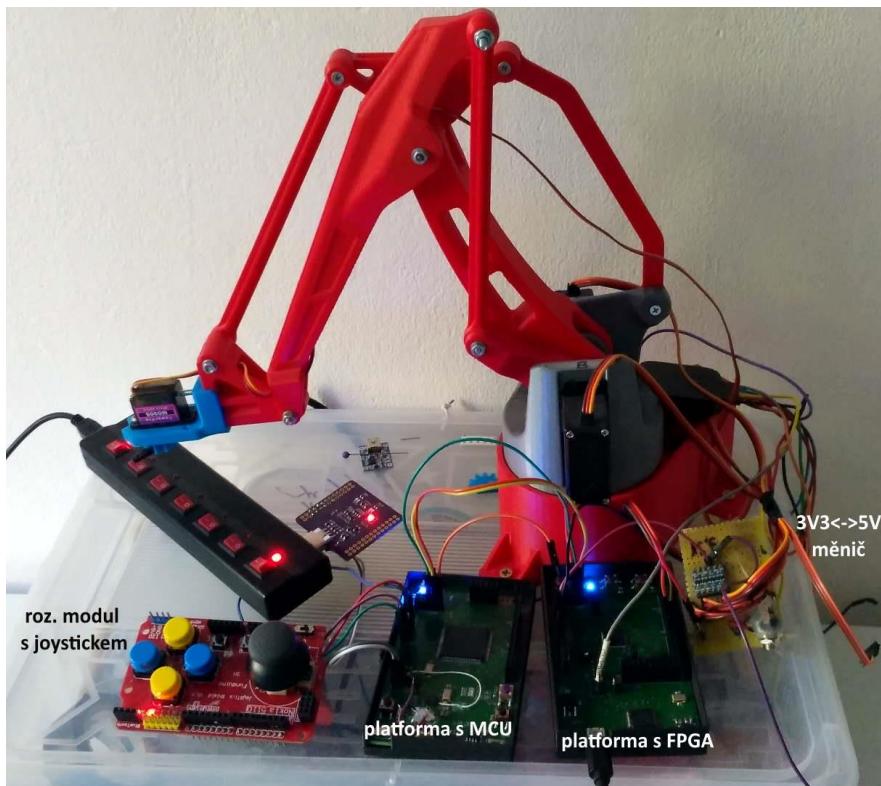
Obrázek 10.10: Schéma zapojení příkladu

překladu analogové hodnoty, v daném kanálu, na digitální hodnotu s 8 či 16bitovou přesností. Po inicializaci ADC převodníku a zavolání funkce `pool_ADC_val(axis)` je vyčtena hodnota osy joysticku, dle parametru axis, který určuje osu potažmo ADC kanál. Tato hodnota je 8bitová reprezentace napětí na potenciometru, a tedy rozlišení na jeden bit je zhruba 0,013

V. V rámci zjednodušení jsou nejvyšší 3 bity této hodnoty vynulovány a použity jako adresy pro jednotlivé servomotory. Znamená to značné zkreslení, ale vyzkoušení funkce by to mělo postačovat. Joystick má pouze dvě osy, ale rameno poskytuje tři osy volnosti a třetí osa je suplována tlačítka na modulu a čítačem, který tyto tlačítka zvyšuje či snižuje ve stejném rozmezí jako hodnoty z joysticku.

10.4.2 Platforma s FPGA

Obvod pro FPGA bude zahrnovat děličku vstupní frekvence externího oscilátoru s frekvencí 12 MHz a výstupní 64 kHz, čítač počítá do 187 (12 MHz/ 64 kHz). Výstupní frekvence děliče bude využito jako zdroje hodin pro PWM procesy, které generují požadované signály. Tyto procesy čítají do již zmíněné hodnoty 1280, a dle požadované střídy určené pozicí joysticku nebo tlačítka upravují dobu po, kterou má být signál pro servomotor v logické jedničce či nule. Příjem příkazu od MCU je řízen jedním procesem, který modeluje jednoduchý stavový automat pro příjem 8bitového příkazu. Na základě povolovacího signálu je spuštěn přenos příkazu a ten je následně zpracován a dle jeho obsahu je nastaven čítač, který určuje střídu PWM signálu daného servomotoru.



Obrázek 10.11: Příklad propojení obou platform za účelem ovládání servomotorů ramene pomocí joysticku

Kapitola 11

Závěr

Tato práce se zabývala návrhem a realizací dvou platforem myšleno desek plošných spojů. První platforma s ARM Cortex-M mikrokontrolérem včetně ladícího a programovacího adaptéru. Druhá pak s programovatelným hradlovým polem též FPGA a opět s adaptérem pro nahrávání a ladění konfigurace. Obě desky je možné propojit pomocí dedikovaného konektoru, ve kterém jsou vyvedeny sběrnice SPI a I²C, které umožní konfiguraci a komunikaci FPGA s pomocí mikrokontroléru. Konektor s dutinových vývodů označen jako FPGA_CONFIG je na obou platformách na stejném místě a je možné tak desky napojit jen pomocí správně orientovaných dutinek a hřebíkových kolíků. V rámci návrhu a výroby jsme se seznámili s základními metodami návrhu desek plošných spojů a následně si řekli základní informace o obou technologiích Cortex-M a FPGA od firmy Lattice Semiconductor. Poté co jsme desky oživily a opravil chyby v obou návrzích plošných spojů, které jsou částečně vyloženy v rámci předchozích kapitol, ale jednalo se především o chybně propojené vývody pro zvolení bootovacího režimu a špatně zvolený programovací a ladící adaptér u desky s mikrokontrolérem. U desky s FPGA se jednalo o špatně zapojený resetovací obvod, který držel obvod po dobu zkoušení v resetu. Po vyřešení problémů se zapojením desek, již bylo možné oživené desky programovat a vytvořit požadované demonstrační příklady. Pro obě desky jsme si uvedli nástroje, které jsou nutné pro vývoj aplikací na těchto platformách a základní „flow“ v těchto prostředích, umožňovalo nakonec vytvořit požadovaný binární soubor ať již s programem pro MCU nebo s konfigurací pro FPGA čip a jejich následné nahrání do dané platformy. Demonstrační příklad s blikáním s využitím vstupů z osazených tlačítek je obdobou „Hello world“ programu pro vestavěná zařízení. Tento příklad byl úspěšně přeložen a naprogramován pro obě platformy a sloužil i jako příklad pro vyzkoušení základních ladících technik jako GDB u mikrokontroléru či Reveal analyzátor u FPGA čipu. V rámci návrhu obvodu v FPGA je často výhodné nejprve spustit simulaci a poté až vyzkoušet obvod reálně v FPGA, protože i přes mnohé ochrany, které dnešní FPGA čipy poskytují je pořád možné zničit špatným návrhem a připojením k dalším obvodům zkratem zničit celý čip. Dalším demonstrace funkčnosti platforem se týkala jejich propojení, především pak konfigurací FPGA čipu. Tento příklad popisuje detailnější použití obou platforem, a to především použití výrobcem definovaných periferií či funkčních bloků. Posledním příkladem bylo využití rozšiřujících modulů pro Arduino a příklad implementace pomocí platformy s FPGA a také obdobně na platformě s MCU, kde díky většímu množství periferií je možné např. i převod analogového signálu na digitální a můžeme tak měřit například napětí na modulu osazeném potenciometru.

Celkově považuji výsledek práce za uspokojivý, ale návrh ladícího adaptéru na desce s mikrokontrolérem by potřeboval revizi či úplné nahrazení za modernější adaptér jako je například firmou NXP vyráběný MCULink, který plně podporuje JTAG rozhraní. Obvod s FPGA se zdá být relativně dobře navržen a zvolené FPGA je velmi přívětivé na používání. Vývody kompatibilní s Arduino rozšiřujícími moduly a obdobné zapojení vzhledem k funkcionalitě obvodů poskytuje obrovské možnosti rozšíření obou navržených platforem. Pro větší použitelnost by byla výhodná i možnost jednoduše vyzkoušet některé příklady pouze s pomocí nějakého online nástroje či prostředí bez nutnosti instalace vývojových prostředí a rozšíření sady demonstračních příkladů je určité práce do budoucna.

Literatura

- [1] Šimek V.: Návrh desek plošných spojů, přednáška. Fakulta informačních technologií VUT v Brně, 2020. [online] [cit. 13-10-2020] Dostupné z:
https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FNAVIT%2Flectures%2F05_NAV_PCB.pdf&cid=13424
- [2] Martínek T.: Technologie FPGA, přednáška. Fakulta informačních technologií VUT v Brně, 2020. [online] [cit. 10-11-2020] Dostupné z:
https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FINCIT%2Flectures%2Ftechnologie_fpga.pdf&cid=10318
- [3] Cirkovský J.: Metody návrhu desek plošných spojů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 53 s. [online] [cit. 18-11-2020] Dostupné z:
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7581
- [4] Svačina, J.: Elektromagnetická kompatibilita, skripta. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2002. 171 s. [online] [cit. 21-12-2020] Dostupné z:
https://home.pilsfree.net/fantom/FEL/EMC/EMC_skripta.pdf
- [5] Lattice Semiconductor: MachXO3D Family Data Sheet – FPGA-DS-02026. [online] [cit. 13-01-2021] Dostupné z:
https://www.latticesemi.com/view_document?document_id=52590
- [6] Záhlava V: Návrh DPS a EMC – blokování napájení. [online] [cit. 28-12-2020] Dostupné z:
https://moodle.fel.cvut.cz/pluginfile.php/125301/mod_resource/content/1/07_EMCA_blokovani.pdf
- [7] Lorenc M.: Využití základních jednotek. [online] [cit. 10-01-2021] Dostupné z:
<https://lorenc.info/3MA112/vyuziti-zakladnich-jednotek.htm>
- [8] Lattice Semiconductor: MachXO3D Programming and Configuration Usage Guide – FPGA-TN-02069. [online] [cit. 14-02-2021] Dostupné z:
https://www.latticesemi.com/view_document?document_id=52591
- [9] Lattice Semiconductor: Using Hardened Control Functions in MachXO3D Devices – FPGA-TN-02117. [online] [cit. 11-02-2021] Dostupné z:
https://www.latticesemi.com/view_document?document_id=52593
- [10] Yiu J.: Cortex-M for Beginners. [online] [cit. 01-03-2021] Dostupné z:
https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-00-52-96/White-Paper_-2D00_-Cortex_2D00_M-for-Beginners_-2D00_-2016_-2800_final-v3_2900_.pdf
- [11] Tišnovský P.: Architektura mikrořadičů s jádry ARM Cortex-m0 a ARM Cortex-m0. [online] [cit. 16-03-2021] Dostupné z: <https://www.root.cz/clanky/architektura-microradiici-s-jadry-arm-cortex-m0-a-arm-cortex-m0/>

mikroradicu-s-jadry-arm-cortex-m0-a-arm-cortex-m0/

- [12] Coleman Ch.: A deep dive into ARM Cortex-M debug interfaces. [online] [cit. 05-04-2021] Dostupné z: <https://interrupt.memfault.com/blog/a-deep-dive-into-arm-cortex-m-debug-interfaces>
- [13] Peterson Z.: Flat vs. Hierarchical Schematics : Why you need dynamic schematic compilation. [online] [cit. 08-04-2021] Dostupné z: <https://resources.altium.com/p/flat-vs-hierarchical-schematics-why-you-need-dynamic-schematic-compilation>
- [14] Yiu J.: ARM Cortex-M for Beginners. [online] [cit. 2-2-2021] Dostupné z: <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/white-paper-cortex-m-for-beginners-an-overview-of-the-arm-cortex-m-processor-family-and-comparison>
- [15] NXP Semiconductor: LPC435x/3x/2x/1x Product data sheet. [online] [cit. 12-10-2020] Dostupné z: https://www.nxp.com/docs/en/data-sheet/LPC435X_3X_2X_1X.pdf
- [16] NXP Semiconductor: LPC43xx/LPC43Sxx Arm Cortex-M4/M0 multi-core microcontroller – User Guide. [online] [cit. 12-10-2020] Dostupné z: <https://www.nxp.com/secured/assets/documents/en/user-guide/UM10503.pdf>
- [17] SOS Electronic: Design desky plošných spojů – Základní pravidla návrhu DPS (1.část). [online] [cit. 02-01-2021] Dostupné z: <https://vyvoj.hw.cz/design-desky-plosnych-spoju-zakladni-pravidla-navrhu-dps-1-cast.html>
- [18] SOS Electronic: Design desky plošných spojů – Jak správně rozmístit součástky na DPS (2.část). [online] [cit. 05-01-2021] Dostupné z: <https://vyvoj.hw.cz/design-desky-plosnych-spoju-jak-spravne-rozmistit-soucastky-na-dps-2-cast.html>
- [19] Wikipedia Contributors: Routing (electronic design automation). [online] [cit. 09-04-2021] Dostupné z: [https://en.wikipedia.org/wiki/Routing_\(electronic_design_automation\)](https://en.wikipedia.org/wiki/Routing_(electronic_design_automation))
- [20] Klauz M.: Automatické routování plošných spojů. [online] [cit. 16-04-2021] Dostupné z: <https://www.dps-az.cz/cad-cam-cae/id:5996/automaticke-routovani-plosnych-spoju>
- [21] NXP Semiconductor: MCUXpresso IDE User Guide. [online] [cit. 01-03-2021] Dostupné z: https://www.nxp.com/docs/en/user-guide/MCUXpresso_IDE_User_Guide.pdf
- [22] OpenOCD Contributors: What is OpenOCD. [online] [cit. 01-03-2021] Dostupné z: http://openocd.org/doc/html/About.html#What-is-OpenOCD_003f
- [23] Wikipedia Contributors: Wishbone (computer bus). [online] [cit. 01-03-2021] Dostupné z: [https://en.wikipedia.org/wiki/Wishbone_\(computer_bus\)](https://en.wikipedia.org/wiki/Wishbone_(computer_bus))
- [24] Lattice Semiconductor: Lattice Diamond 3.12 User Guide. [online] [cit. 05-03-2021] Dostupné z:

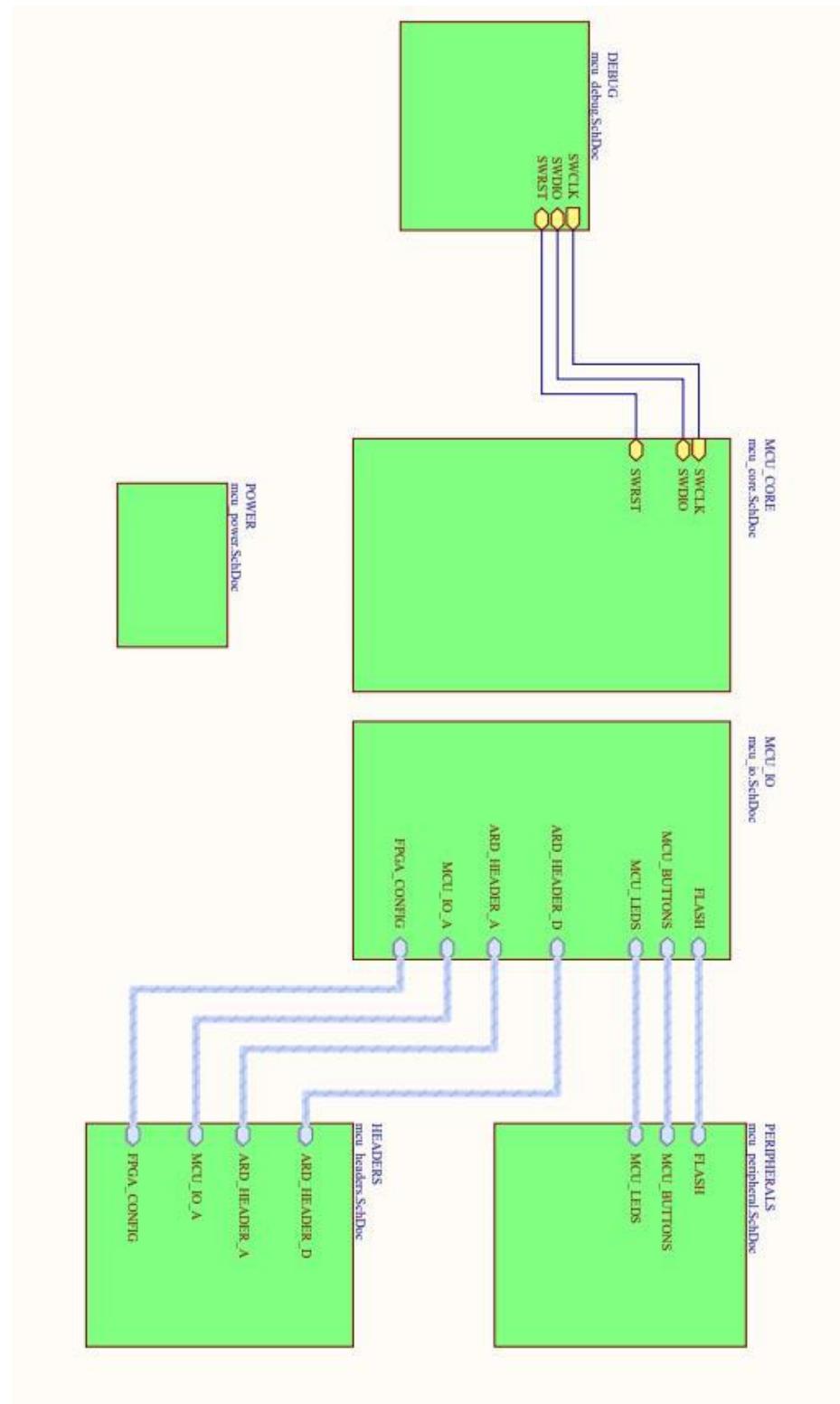
https://www.latticesemi.com/view_document?document_id=53077

- [25] Edwards S.A.: Writing VHDL for RTL Synthesis. [online] [cit. 11-02-2021]
Dostupné z:
<https://citeseervx.ist.psu.edu/viewdoc/download?doi=10.1.1.296.3496&rep=rep1&type=pdf>
- [26] Riege S.: Creating circuit connectivity in your schematics in Altium Designer. [online] [cit. 12-02-2021] Dostupné z:
<https://www.altium.com/documentation/altium-designer/creating-connectivity-ad?version=18.1>
- [27] Cullen T.: Understanding the Makeup of a printed circuit board. [online] [cit. 10-1-2021] Dostupné z: <https://www.altium.com/documentation/altium-designer/the-board-ad?version=18.1#!double-sided-board>
- [28] Rust Embedded: Overview of embedded programming and debugging. [online] [cit. 10-1-2021] Dostupné z: <https://docs.rust-embedded.org/debugonomicon/>
- [29] Panoš M.: Popis: Arduino Multi-function Shield. [online] [cit. 10-1-2021]
Dostupné z: <http://kabinet.fyzika.net/dilna/ARDUINO/funduino-popis.php>
- [30] Vašíček Z., Slaný K.: FITkit – úvod. [online] [cit. 10-1-2021] Dostupné z:
<http://merlin.fit.vutbr.cz/FITkit>
- [31] Farrell: LAttice MachXO2 Breakout Board HelloWorld with Diamond. [online] [cit. 10-1-2021] Dostupné z: http://www.farrellf.com/projects/hardware/2016-01-03_Lattice_MachXO2_Breakout_Board_HelloWorld/
- [32] Riege S.: Tutorial – A complete design walkthrough with Altium Designer. [online] [cit. 10-1-2021] Dostupné z:
<https://www.altium.com/documentation/altium-designer/from-idea-to-manufacture-driving-a-pcb-design-through-altium-designer?version=18.1>
- [33] Wikipedia Contributors: Power management integrated circuit. [online] [cit. 10-1-2021] Dostupné z:
https://en.wikipedia.org/wiki/Power_management_integrated_circuit
- [34] Diligent: Arty S7 Reference Manual, [online] [cit. 10-04-2021] Dostupné z:
<https://reference.digilentinc.com/reference/programmable-logic/arty-s7/reference-manual>
- [35] FTDI chip: FT2232H Datasheet. [online] [cit. 10-04-2021] Dostupné z:
https://ftdichip.com/wp-content/uploads/2020/07/DS_FT2232H.pdf
- [36] Gatema: Kritéria dat pro POOL servis. [online] [cit. 17-04-2021] Dostupné z:
<https://www.gatema.cz/file-link/kriteria-dat-pro-pool-servis.pdf>
- [37] pyOCD Contributors: pyOCD Documentation. [online] [cit. 10-01-2021]
Dostupné z: <https://github.com/pyocd/pyOCD>
- [38] Mozhikunnath R.: What is an IP (Intellectual Property) core in Semiconductors. [online] [cit. 17-12-2020] Dostupné z: <https://any silicon.com/ip-intellectual->

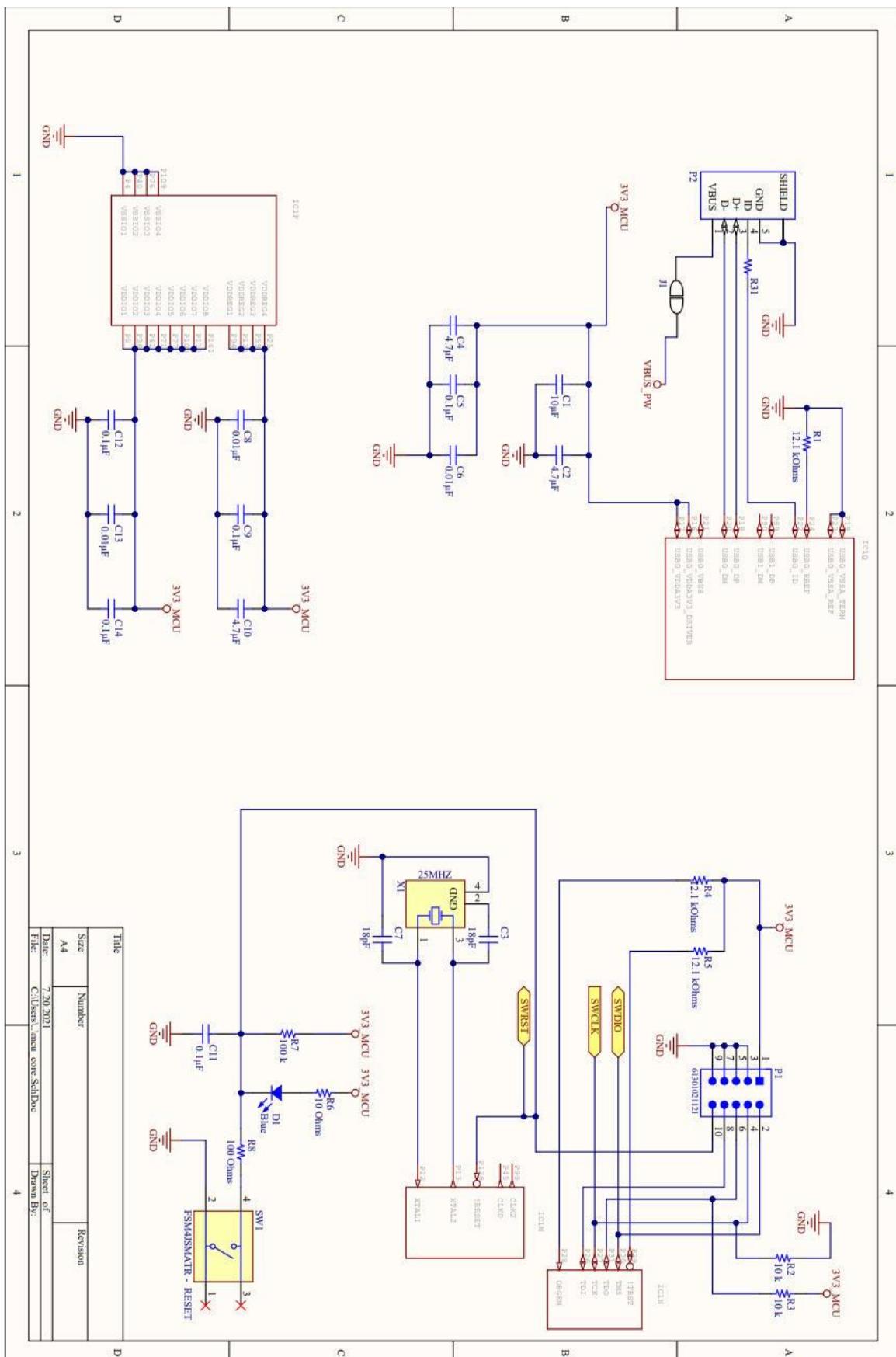
- property-core-semiconductors/
- [39] Raspberry Pi Foundation: Raspberry Pi 3 B+ Product brief. [online] [cit. 04-01-2021] Dostupné z: <https://datasheets.raspberrypi.org/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>
- [40] NXP Semiconductor: OpenSDA seriál and debug adapter. [online] [cit. 23-02-2021] Dostupné z: <https://www.nxp.com/design/microcontrollers-developer-resources/ides-for-kinetis-mcus/opensda-serial-and-debug-adapter:OPENSDA>
- [41] Kovář P.: Plošné spoje (PCB), přednáška. [online] [cit. 01-06-2021] Dostupné z: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=4469>
- [42] Texas Instruments: IEEE Std 1149.1 (JTAG) Testability Primer . [online] [cit. 10-01-2021] Dostupné z: <https://www.ti.com/lit/an/ssya002c/ssya002c.pdf>
- [43] Lattice Semiconductor: Reveal User Guide. [online] [cit. 05-06-2021] Dostupné z: https://www.latticesemi.com/view_document?document_id=52075
- [44] Lattice Semiconductor: Lattice Synthesis Engine (LSE). [online] [cit. 10-06-2021] Dostupné z:
<https://www.latticesemi.com/en/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond/LatticeSynthesisEngine>

Příloha A

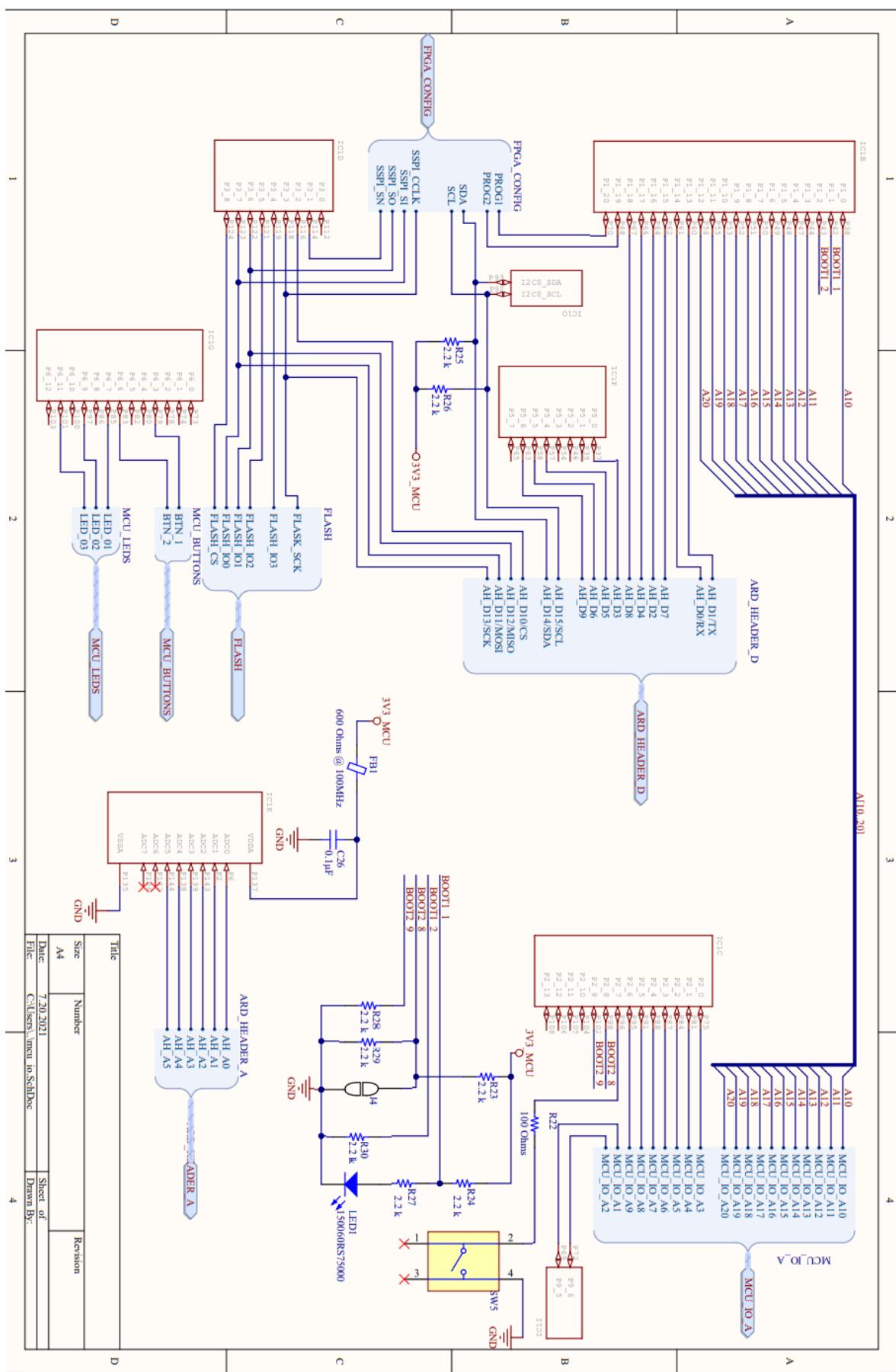
Schéma zapojení platformy s MCU



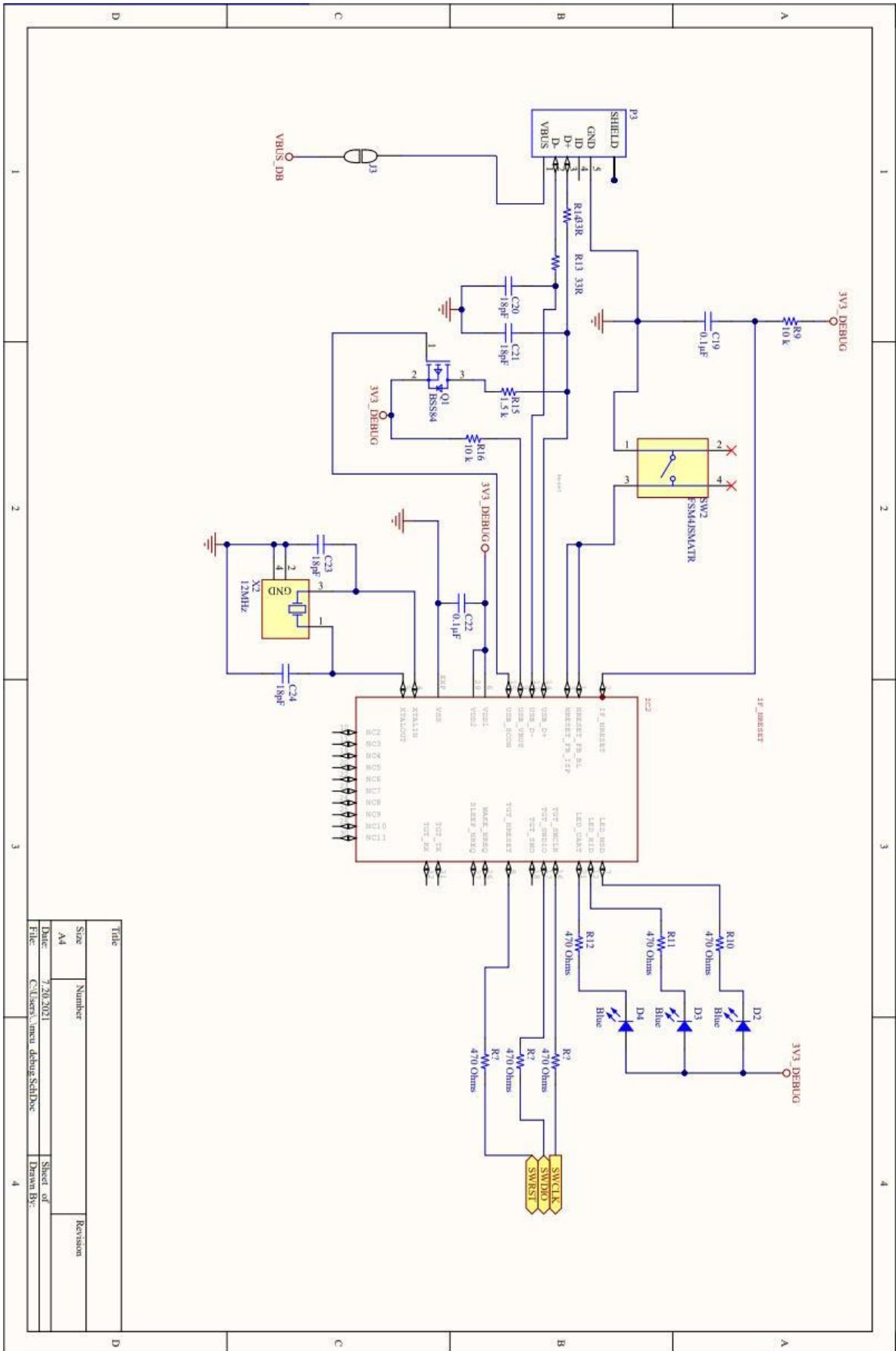
Obrázek A.1: Schématické propojení jednotlivých částí platformy s MCU



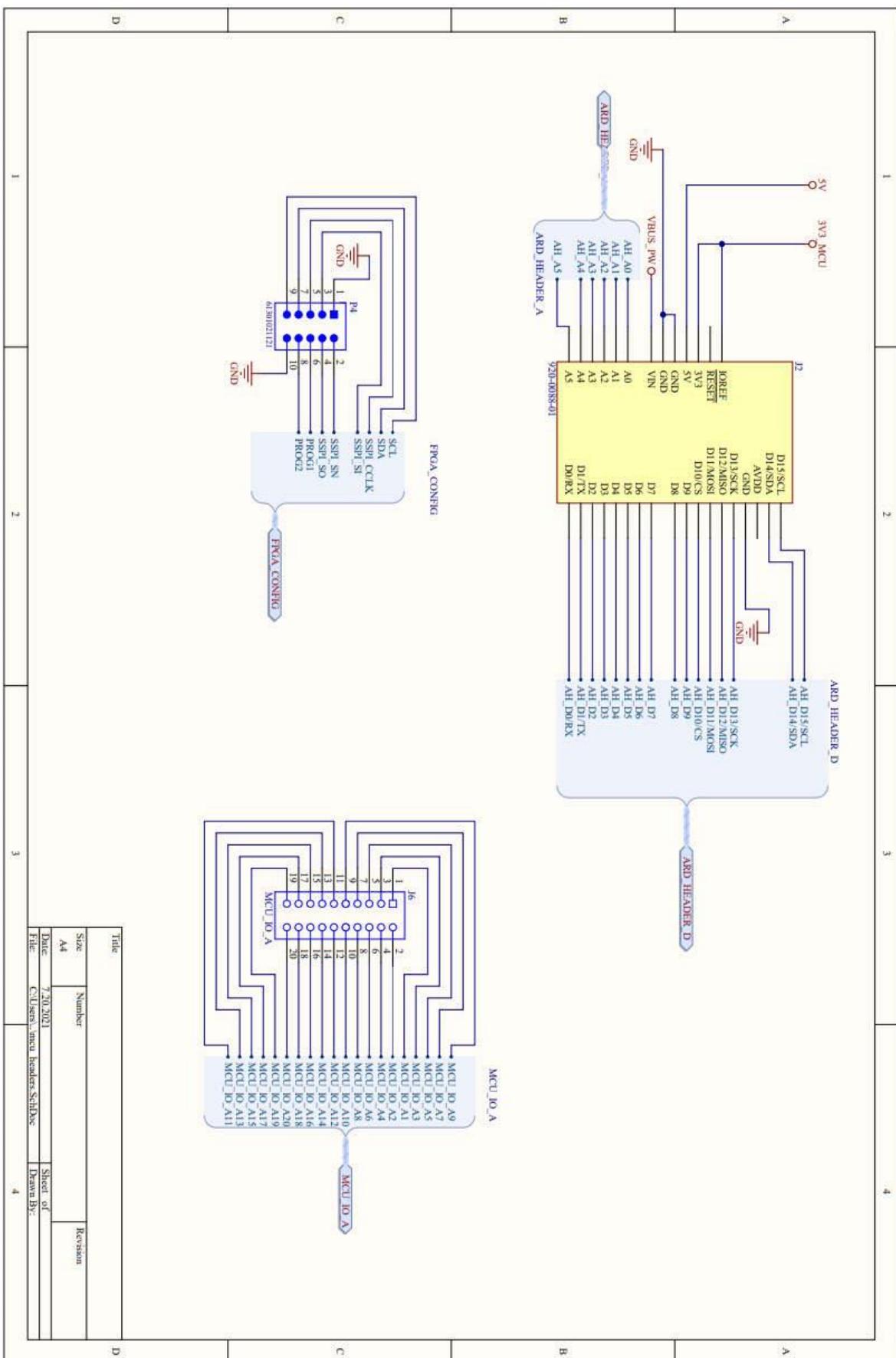
Obrázek A.2: Schéma obsahující zapojení hlavním podpůrných obvodů s rozhraní pro mikrokontrolér



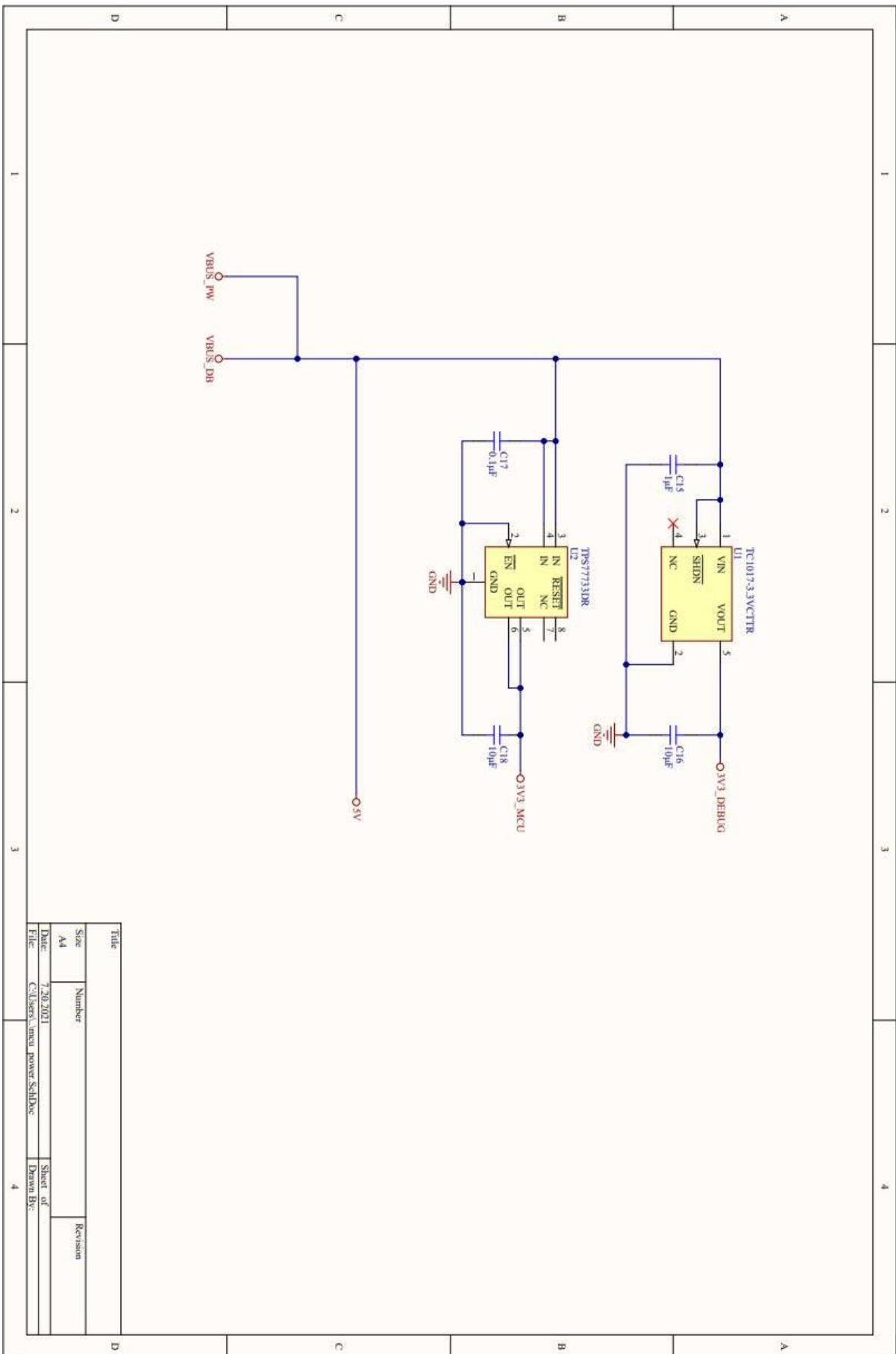
Obrázek A.3: Schéma obsahující zapojení různé sběrnice vývodů a jejich připojení k mikrokontroléru



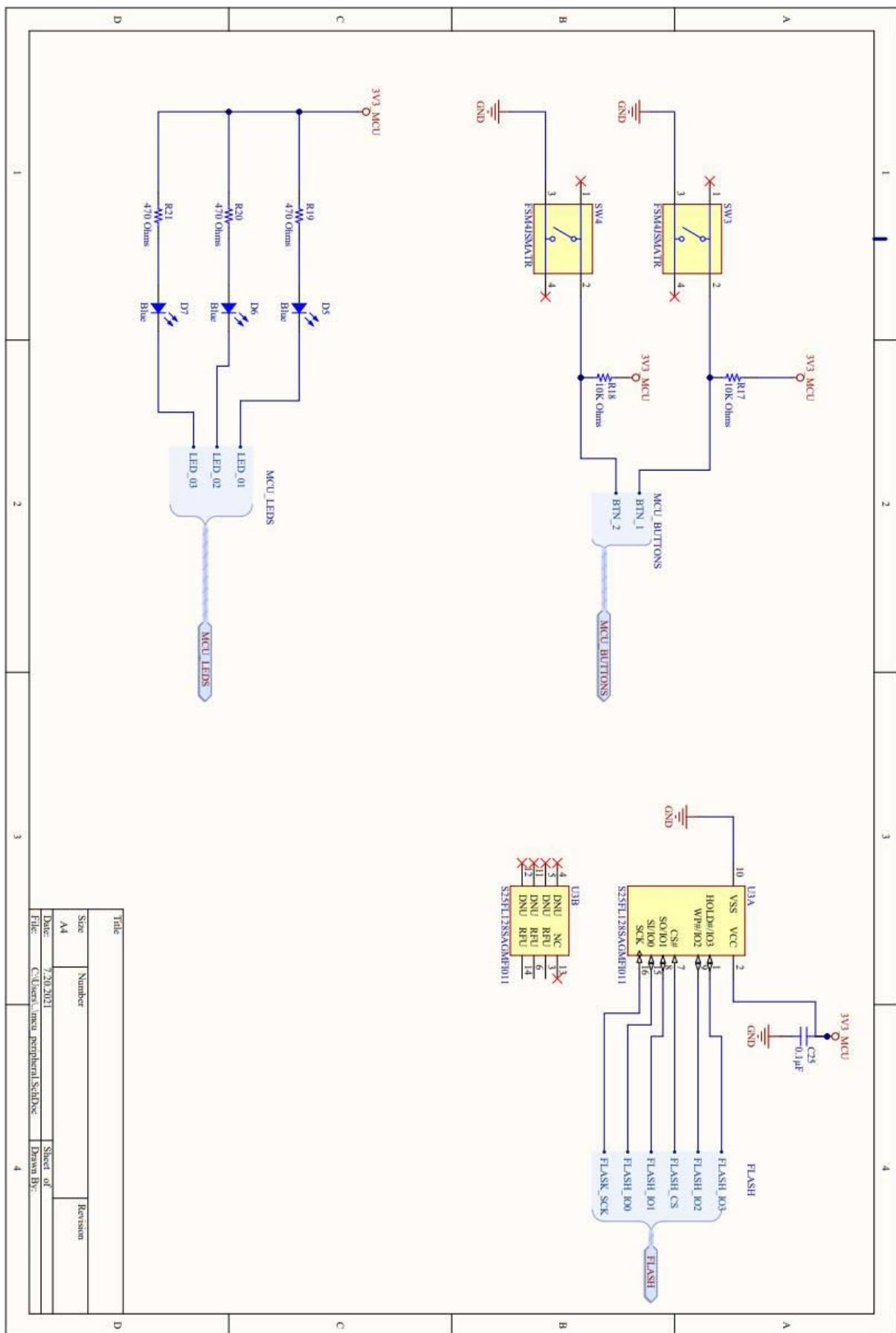
Obrázek A.4: Schéma zapojení ladícího adaptéru pro obvod s mikrokontrolérem



Obrázek A.5: Schéma zapojení vývodů na desce



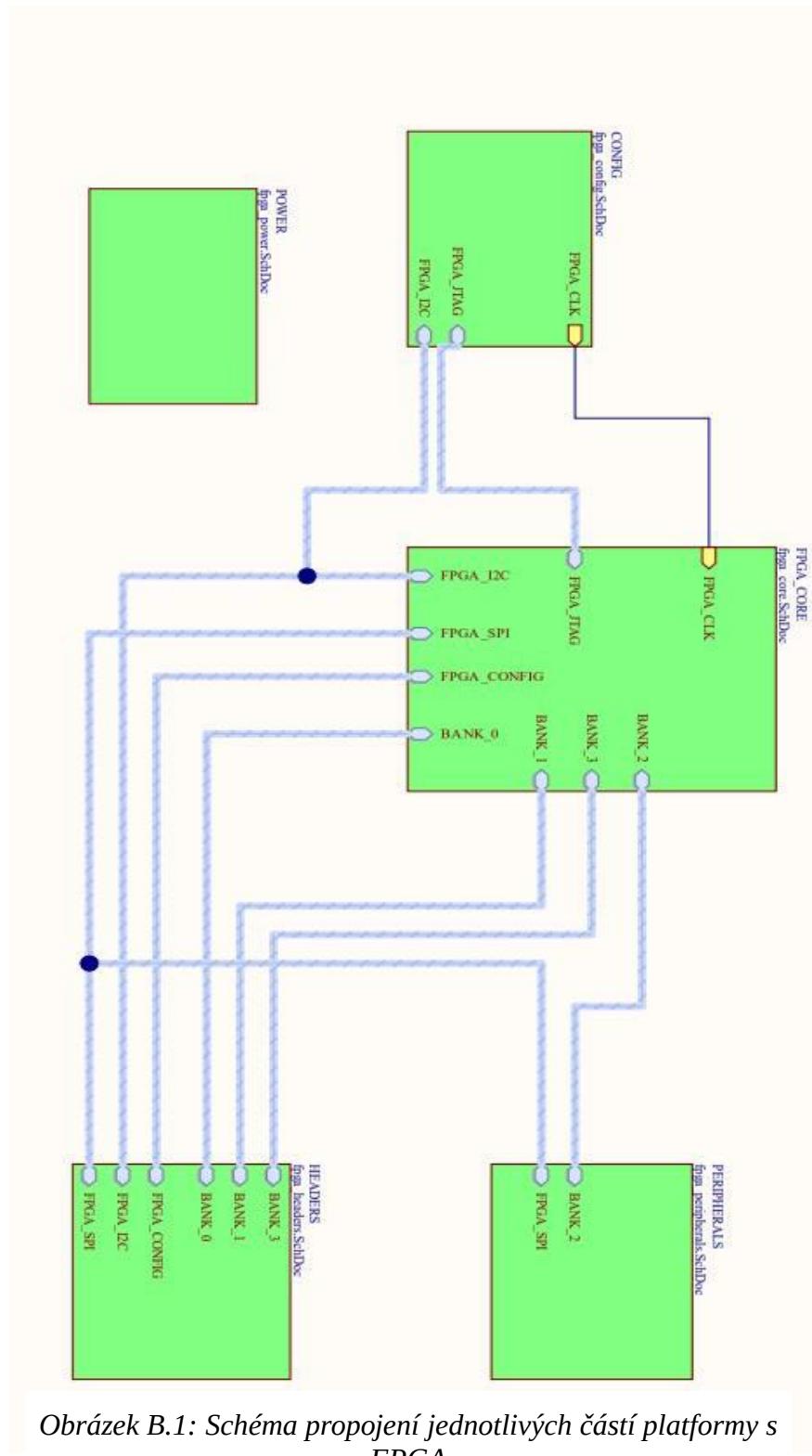
Obrázek A.6: Schéma napájení platformy



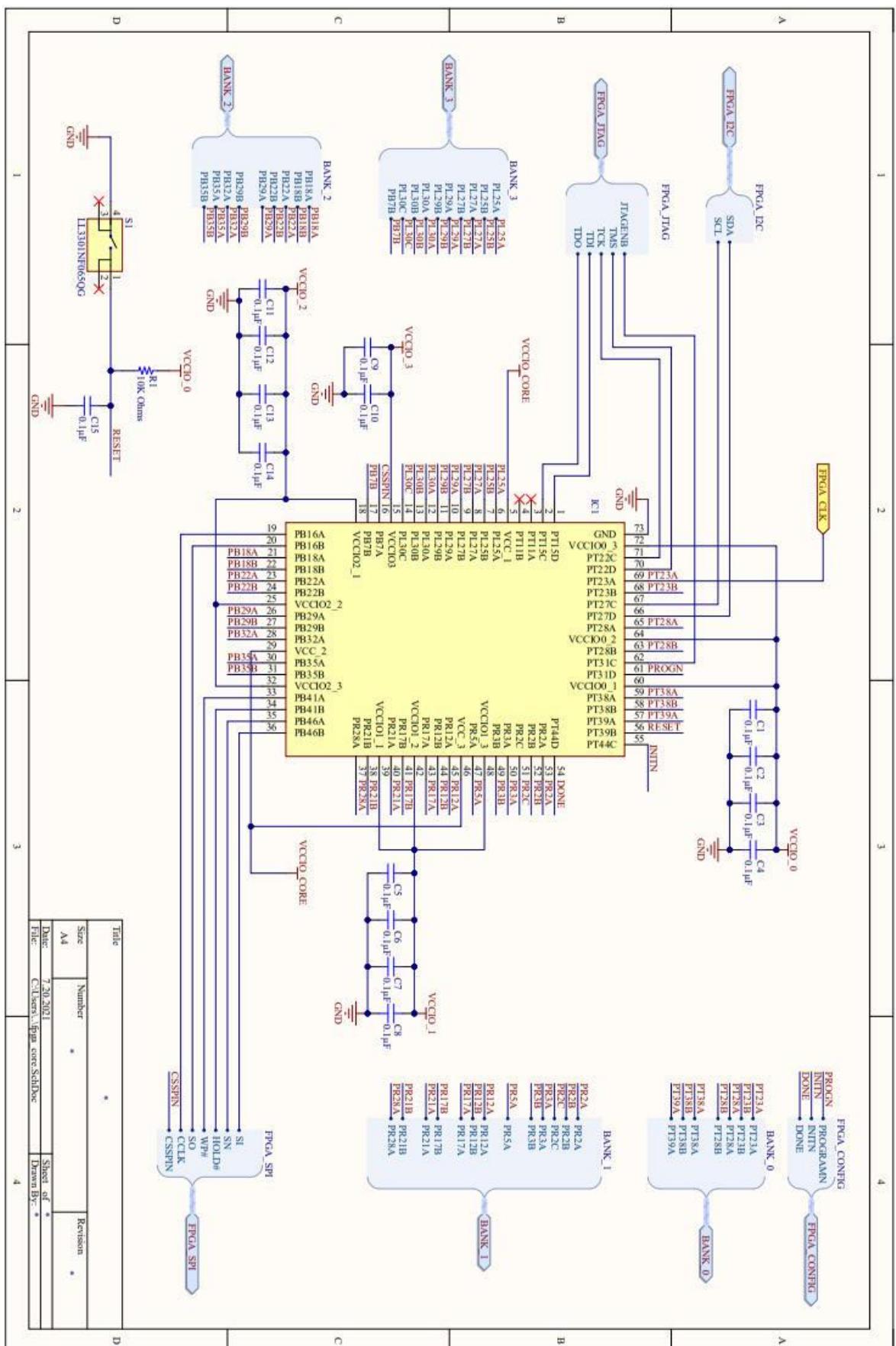
Obrázek A.7: Schéma zapojení periferií

Příloha B

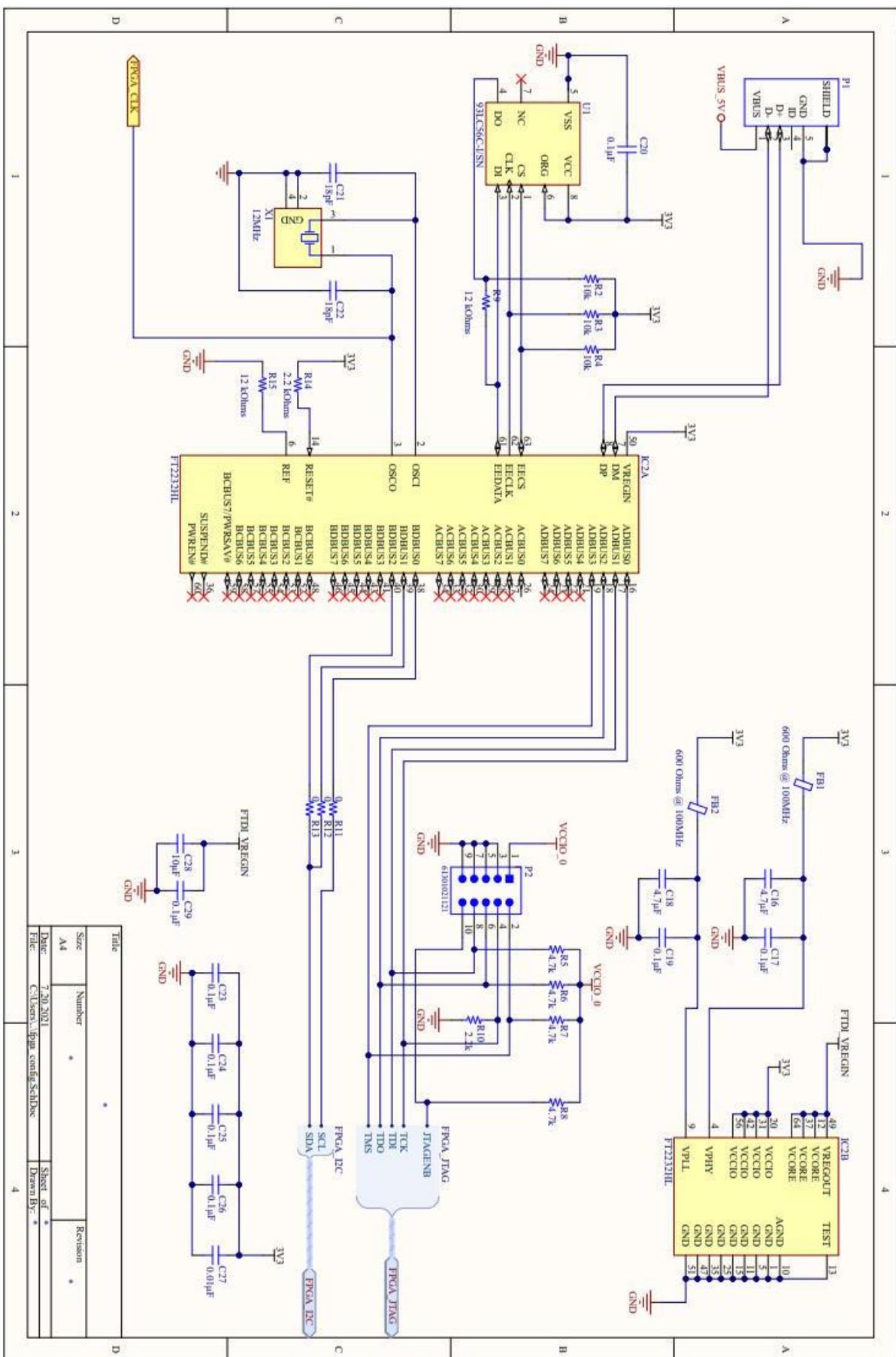
Schéma zapojení platformy s FPGA



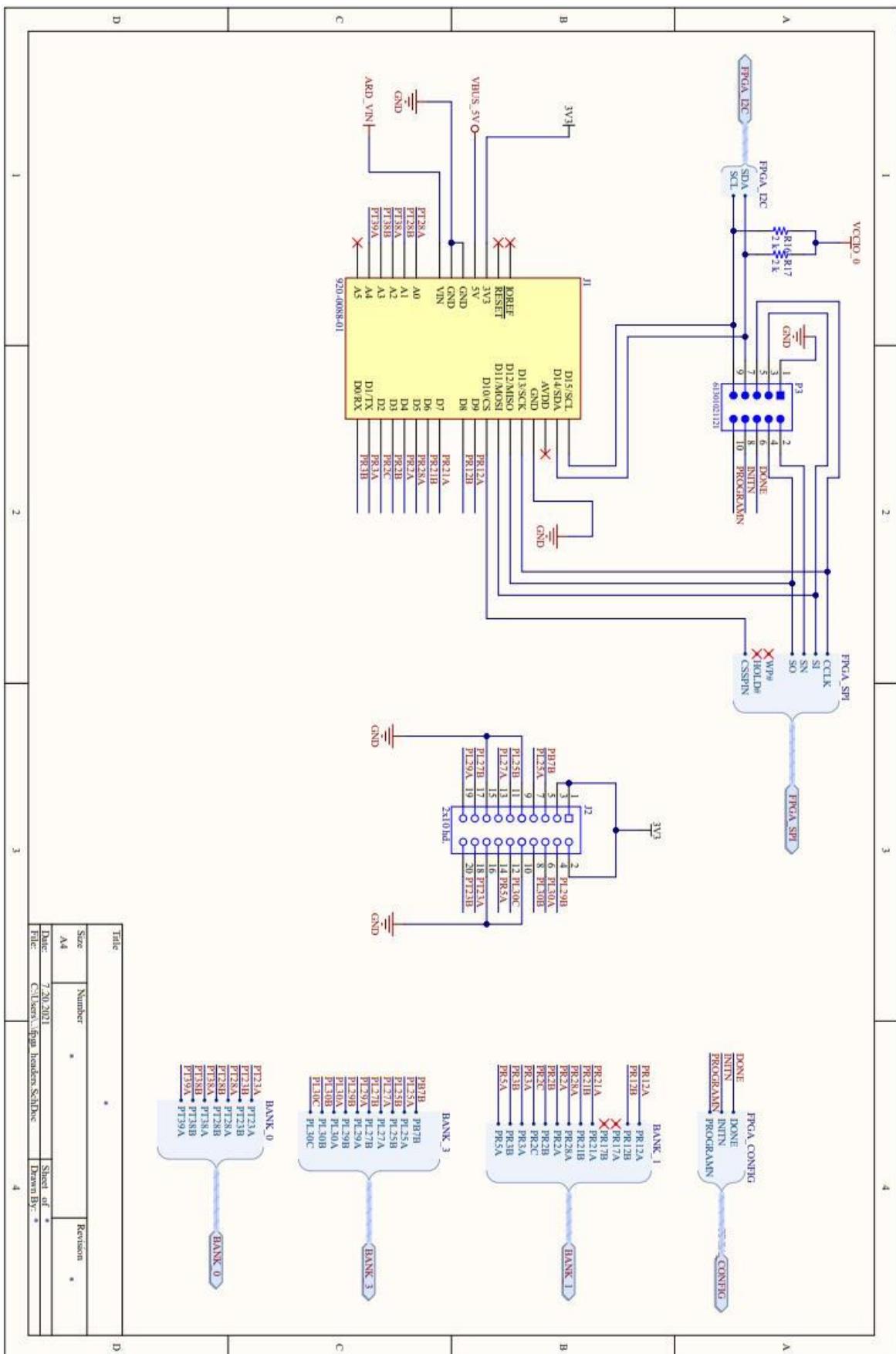
Obrázek B.1: Schéma propojení jednotlivých částí platformy s FPGA



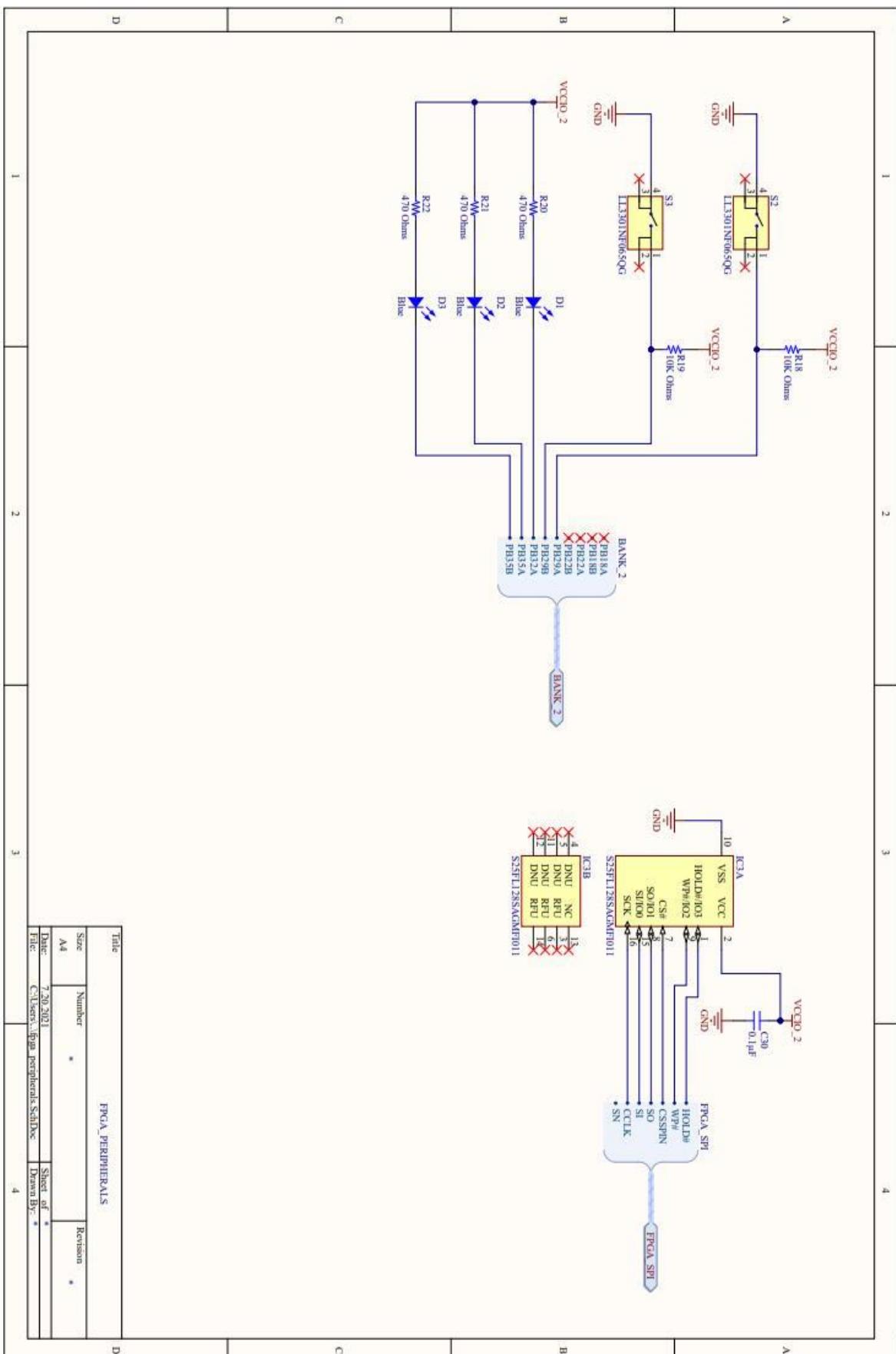
Obrázek B.2: Schéma zapojení signálů ze strany FPGA



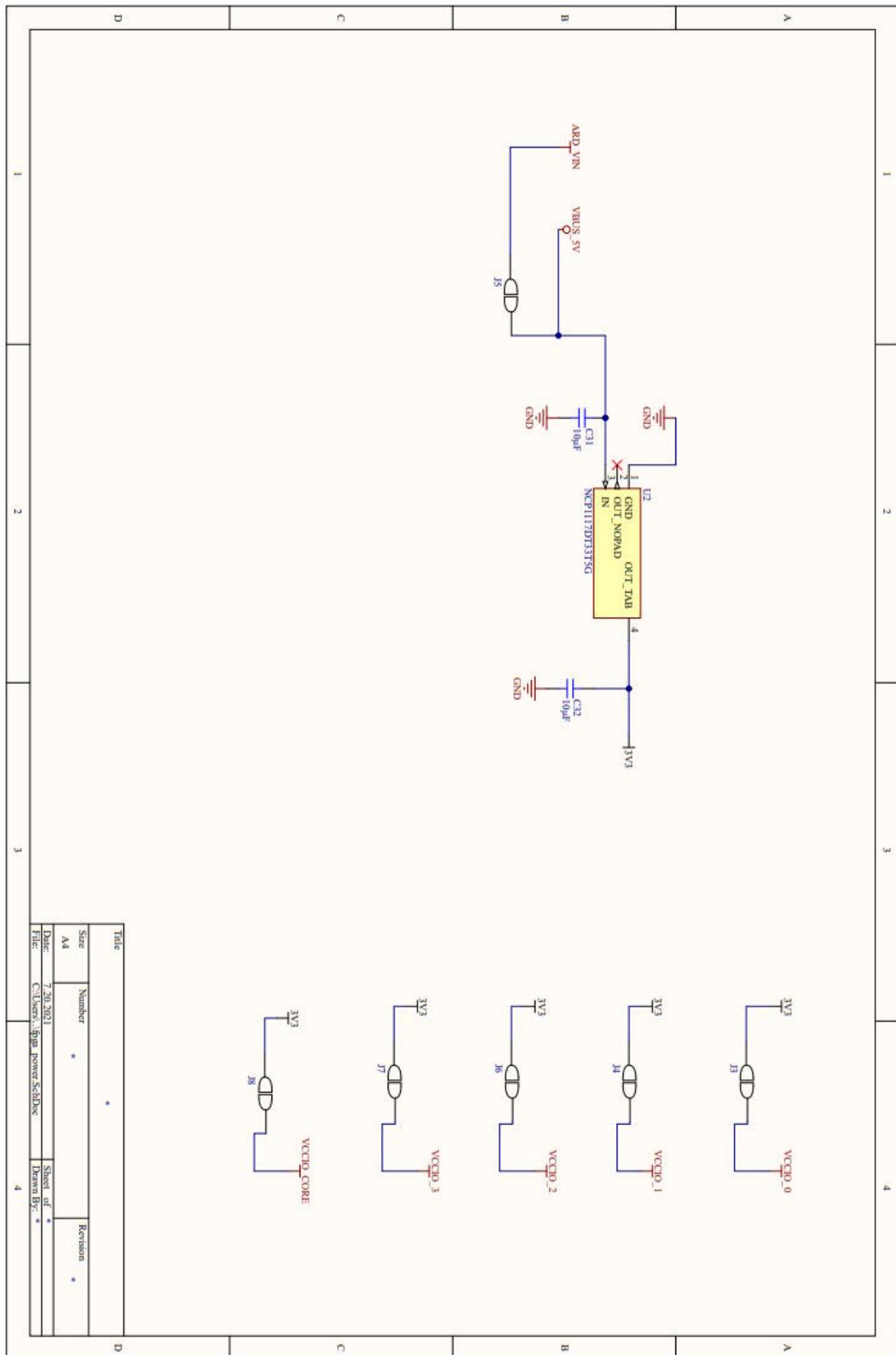
Obrázek B.3: Schéma konfiguračního a ladící obvodu FPGA



Obrázek B.4: Schéma zapojení jednotlivých skupin výstupů na desce



Obrázek B.5: Schéma zapojení periferií



Příloha C

Obsah přiloženého paměťového média

Vzhledem k rozsáhlější souborové struktuře si uvedeme a krátce komentujeme strukturu adresářů obsahující projekt a potažmo odevzdádaného DVD nosiče. Jednotlivé adresáře obsahují README soubor s popisem obsahu a případně dalšími pokyny.

```
-/xkboard_mcu
-----/firmware      #Projekty pro jednotlivé dem. příklady
-----/dps           #Obsahuje Altium Designer projekt
-----/case          #3D model v STL formátu pro tisk obalu plaforem
-----/schema.pdf    #PDF se schématem celé platformy
-/xkboard_fpga
-----/firmware
-----/dps
-----/case
-----/schema.pdf
-/technicka_zprava.pdf
-/technicka_zpraca.docx
-/images
```