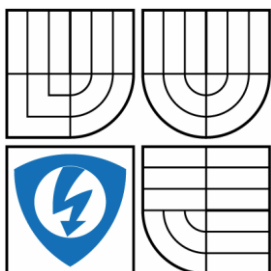


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

UNIVERZÁLNÍ MODUL S PŘIPOJENÍM NA ETHERNET

UNIVERSAL ETHERNET MODULE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

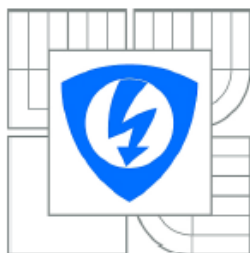
Bc. TOMÁŠ ŘEHÁČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FRANTIŠEK BURIAN

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Tomáš Řeháček
Ročník: 2

ID: 136580
Akademický rok: 2014/2015

NÁZEV TÉMATU:

Univerzální modul s připojením na ethernet

POKYNY PRO VYPRACOVÁNÍ:

Úkolem studenta v této semestrální práci je navrhnout univerzálně použitelné zařízení, integrující v sobě hradlové pole s mikroprocesorem, použitelné pro dálkové měření digitálních veličin. Zařízení komunikuje s počítačem pomocí sběrnice ethernet a protokolů IPv4.

1. Navrhněte a realizujte plošný spoj implementující výše uvedenou funkci.
2. Navrhněte a realizujte krabičku pro vytvořený plošný spoj.
3. Navrhněte základní programový kód pro použité programovatelné obvody pro otestování správného propojení obvodů na desce s plošnými spoji.
4. Navrhněte aplikační vrstvu komunikačního protokolu nad UDP pro spojení s PC.
5. Navrhněte aplikaci do počítače v jazyce C# vizualizující data z realizovaného zařízení.
6. Realizujte programový kód do všech programovatelných obvodů na desce dle zadání.

DOPORUČENÁ LITERATURA:

E, Spurgeon Charles. Ethernet: The definitive guide. Vyd. 1. Boston: O'Reilly, 2000, 497 s. ISBN 15-659-2660-9.

Termín zadání: 9.2.2015

Termín odevzdání: 18.5.2015

Vedoucí práce: Ing. František Burian

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Diplomová práce se zabývá vytvořením univerzálního modulu s připojením na ethernet a jeho použitím pro měření digitálních veličin. Hlavními součástmi modulu jsou hradlové pole typu FPGA, mikroprocesor, PHY čip a konektor RJ45. V práci je navržena deska plošných spojů v programu Eagle. Krabíčka pro desku je vytvořena v programu SketchUp a vytištěna na 3D tiskárně. Ve vytvořeném univerzálním modulu je implementován kód plnící funkci logického analyzátoru. Data naměřená analyzátozem jsou odesílána na ethernet pomocí UDP/IPv4 protokolu. Poslední částí práce je vytvoření aplikace v jazyce C# pro zobrazení naměřených průběhů v PC.

Klíčová slova

FPGA, STM32F4, Ethernet, MII, IPv4, UDP, logický analyzátor, 3D tisk

Abstract

This master's thesis deals with creating a universal module with Ethernet interface and its using for digital values measurement. The main module components are gate array FPGA, microprocessor, PHY chip and RJ45 connector. Printed circuit board is in this thesis designed in Eagle. The chase for board is drawn in SketchUp and printed on a 3D printer. There is a code implemented in created universal module that is performing the function of logic analyzer. Data measured by the analyzer are sent to the Ethernet using UDP/IPv4 protocol. The last part is creating an application in C# to display the measured waveforms on a PC.

Keywords

FPGA, STM32F4, Ethernet, MII, IPv4, UDP, logic analyzer, 3D print

Bibliografická citace:

ŘEHÁČEK, T. *Univerzální modul s připojením na ethernet*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 101s. Vedoucí diplomové práce byl Ing. František Burian.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Univerzální modul s připojením na ethernet jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **18. května 2015**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Františku Burianovi za odbornou pomoc, ochotu a vřelý přístup při konzultacích a za další cenné rady při zpracování mé diplomové práce. Děkuji také všem, kteří mě podporovali při vytváření práce.

V Brně dne: **18. května 2015**

.....
podpis autora

Obsah

1	Úvod	12
2	Hradlová pole	13
2.1	Rozdělení PLD	13
2.1.1	SPLD (Simple Programmable Logic Device)	13
2.1.2	CPLD (Complex Programmable Logic Device)	14
2.1.3	FPGA (Field Programmable Gate Array)	15
2.2	Programování (konfigurace) hradlových polí	17
2.2.1	Jazyk VHDL	17
3	Komunikační sběrnice	17
3.1	UART	18
3.1.1	Asynchronní sériová komunikace	18
3.2	USART	19
3.2.1	SPI	20
3.3	JTAG	21
3.4	Ethernet	22
3.4.1	MII	22
3.4.2	Ethernetový rámec	25
3.4.3	IPv4 protoko	26
3.4.4	CRC	28
4	Hardware uvažovaný při návrhu	32
4.1	Hradlové pole (FPGA) - Spartan-3A	32
4.1.1	Architektura Spartan-3A	33
4.2	Mikroprocesor – STM32F427VIT6	34
4.3	PHY čip – KSZ8051MLL	35
4.3.1	NAND Tree test	36
4.4	Konektor RJ45 – LMJTAB881243M-L	37
4.5	Spínané regulátory napětí – LM2595	37
4.6	Budiče 74AHC245	39
4.7	Programovací kabel JTAG HS2	39
4.8	Black Magic Probe	40
5	Návrh desky plošných spojů	40
5.1	Napájení	41
5.1.1	Postup pro výběr součástek	41
5.1.2	Vybrané součástky	43

5.1.3	Rozvod napájení.....	44
5.2	Mikroprocesor	45
5.3	Hradlové pole.....	47
5.4	PHY čip a konektor RJ45.....	48
5.5	Propojení mezi součástkami a okolím.....	50
5.5.1	Tlačítka a diody.....	50
5.5.2	PHY čip ↔ mikroprocesor.....	51
5.5.3	mikroprocesor ↔ FPGA	53
5.5.4	mikroprocesor ↔ okolí	54
5.5.5	FPGA ↔ okolí	55
6	Realizace desky plošných spojů	56
6.1	Výroba a popis desky jako celku	58
6.1.1	Osazení a oživení desky	61
6.2	Dodatečné změny na desce oproti schématu.....	62
7	Návrh krabičky pro desku	63
8	Software pro otestování funkčnosti desky.....	65
8.1	Testovací software pro mikroprocesor.....	65
8.1.1	Modul „funkce_spolecne“	66
8.1.2	Modul „test_desky“	66
8.2	Testovací software pro FPGA.....	68
8.3	Shrnutí testování desky s využitím programů pro mikroprocesor a FPGA	69
9	Realizace logického analyzátoru	71
9.1	Část analyzátoru implementovaná v FPGA	71
9.1.1	Generování signálů.....	72
9.1.2	Měření signálů a odesílání dat.....	74
9.1.3	Shrnutí návrhu v FPGA.....	77
9.2	Část analyzátoru implementovaná v mikroprocesoru	79
9.2.1	Modul „funkce_Analyzator“	80
9.2.2	Funkce a globální proměnné v hlavní části programu (main.c).....	81
9.2.3	Shrnutí návrhu v mikroprocesoru.....	84
9.3	Stručné shrnutí implementace analyzátoru na desce.....	85
9.4	Zobrazující aplikace v PC	87
9.4.1	Zobrazení naměřených průběhů a předávání dat.....	88
9.4.2	Ovládací prvky	89
10	Závěr.....	91

Seznam obrázků

Obrázek 2.1: Struktura obvodů SPLD [11].....	13
Obrázek 2.2: Součtové hradlo obvodu SPLD [11].....	14
Obrázek 2.3: Struktura obvodů CPLD [11]	15
Obrázek 2.4: Základní struktura obvodů FPGA [11].....	15
Obrázek 3.1: Asynchronní sériový přenos [21]	19
Obrázek 3.2: Komunikace SPI rozhraní [13].....	20
Obrázek 3.3: Schéma JTAG systému [14].....	21
Obrázek 3.4: Propojení MII rozhraní	23
Obrázek 3.5: Způsob přenosu dat přes MII rozhraní [3].....	24
Obrázek 4.1: Architektura rodiny Spartan-3A [4]	34
Obrázek 4.2: Blokové schéma mikroprocesorů STM32F427xx [5]	35
Obrázek 4.3: Blokové schéma pro zapojení PHY čipu do obvodu [8]	36
Obrázek 4.4: Schéma propojení vývodů konektoru RJ45 (mimo LED) [28]	37
Obrázek 4.5: Regulátor LM2595 v pouzdru TO220-5(NDH) [GME Electronic].....	37
Obrázek 4.6: Blokové schéma LM2595 [9].....	38
Obrázek 4.7: Schéma zapojení obvodu 74AHC245 [10].....	39
Obrázek 4.8: Programovací kabel JTAG HS-2 [7]	40
Obrázek 4.9: Uspořádání konektoru SWD.....	40
Obrázek 5.1: Schéma zapojení regulátoru LM2595 [9]	41
Obrázek 5.2: Graf pro výběr indukčnosti pro LM2595-ADJ [9]	42
Obrázek 5.3: Ochrana proti přepólování	44
Obrázek 5.4: Schéma pro regulátor 3,3V	45
Obrázek 5.5: SWD rozhraní pro STM32F427	46
Obrázek 5.6: Napájení STM32F427 s oddělovacími kondenzátory [5].....	46
Obrázek 5.7: Schéma zapojení konfiguračních signálů pro FPGA.....	48
Obrázek 5.8: Schéma napájení PHY (KSZ8051MLL) [8].....	49
Obrázek 5.9: Připojení krystalu k PHY [8].....	49
Obrázek 5.10: Schéma připojení tlačítek	51
Obrázek 5.11: Signály vyvedené z jedné strany mikroprocesoru	53
Obrázek 6.1: Rozvržení komponent na desce	57
Obrázek 6.2: Návrh desky v programu Eagle	57
Obrázek 6.3: 3D model desky	58

Obrázek 6.4: Vrchní strana desky s hodnotami součástek	59
Obrázek 6.5: Spodní strana desky s hodnotami součástek.....	59
Obrázek 6.6: Rozměry desky a rozmístění montážních otvorů.....	60
Obrázek 6.7: Popis a rozmístění pinů vedoucích z desky, LED diod a tlačítek.....	61
Obrázek 7.1: Spodní díl krabičky.....	63
Obrázek 7.2: Vrchní díl krabičky	63
Obrázek 7.3: Detail spojení vpředu a výřezy pro matky v horním dílu krabičky	64
Obrázek 7.4: Detail spojení vzadu	64
Obrázek 7.5: Náhled 3D tisku pro 20 vrstev.....	64
Obrázek 7.6: Složená krabička s deskou (zepředu).....	65
Obrázek 7.7: Složená krabička s deskou (zezadu)	65
Obrázek 8.1: Schéma vytvořené v ISE pro testování desky (propojení FPGA)	68
Obrázek 9.1: Zapojení DCM bloku.....	71
Obrázek 9.2: Nastavení směru a povolení budičů.....	72
Obrázek 9.3: Část pro generování signálů	72
Obrázek 9.4: Schéma pro měření a odesílání dat	74
Obrázek 9.5: Simulace funkce analyzátoru v FPGA.....	79
Obrázek 9.6: Zjednodušený vývojový diagram analyzátoru.....	86
Obrázek 9.7: Zobrazovací aplikace po spuštění (pokud nepřicházejí data)	87
Obrázek 9.8: Přiblížení průběhu z předchozího obrázku	88
Obrázek 9.9: Záznam z měření signálů generovaných hradlovým polem	89

Seznam tabulek

Tabulka 3.1: Ethernetový rámec	25
Tabulka 3.2: Formát protokolu IPv4	26
Tabulka 3.3: Formát protokolu UDP	28
Tabulka 3.4: První tabulka pro sestavení paralelního CRC	31
Tabulka 3.5: Druhá tabulka pro sestavení paralelního CRC	31
Tabulka 4.1: Maximální absolutní hodnoty napětí [4]	33
Tabulka 4.2: Pořadí pinů pro NAND Tree test [8]	36
Tabulka 4.3: Proudové odběry součástek [4][5][8]	38
Tabulka 4.4: Časové parametry pro budiče 74AHC245 [10]	39
Tabulka 5.1: Připojení SWD konektoru k mikroprocesoru	45
Tabulka 5.2: Připojení JTAG konektoru k hradlovému poli	48
Tabulka 5.3: Připojení tlačítek a LED diod	50
Tabulka 5.4: Propojení PHY čipu s mikroprocesorem	52
Tabulka 5.5: Propojení skupiny vodičů „CONFIG_FPGA“ mezi piny mikroprocesoru a hradlového pole	53
Tabulka 5.6: Propojení skupiny vodičů „FPGA“ mezi piny mikroprocesoru a hradlového pole	54
Tabulka 5.7: Vyvedené piny z mikroprocesoru a některé alternativní funkce	55
Tabulka 5.8: Propojení hradlového pole s budiči	56
Tabulka 5.9: Vyvedené signály z FPGA na pinovou lištu	56
Tabulka 6.1: Dodatečné připojení hodinového signálu do FPGA	62
Tabulka 8.1: Přehled propojení mezi piny zařízení a periferiemi v programech pro testování desky	70
Tabulka 9.1: Mezní hodnoty pro měření při využití počtu kanálů	78

1 ÚVOD

Úkolem této diplomové práce je vytvořit univerzální modul pro dálkové měření digitálních veličin.

Tento modul v sobě bude integrovat hradlové pole, mikroprocesor a bude komunikovat pomocí ethernetu. Součástí vytváření modulu bude návrh desky plošných spojů, výroba krabičky, osazení a oživení desky, sestavení firmware pro hradlové pole a mikroprocesor, navrhnutí komunikačního protokolu UDP a vytvoření aplikace v PC, která bude zobrazovat informace z měření veličin.

Při návrhu desky budou stručně popsány jednotlivé součástky vybrané pro realizaci univerzálního modulu. Modul bude vytvářen s ohledem na jeho širší využití. Hlavními součástkami na navržené desce budou spínané zdroje pro zajištění napájení, hradlové pole pro zpracování přicházejících dat, mikroprocesor pro řízení jednotlivých komponent, PHY čip a ethernetový konektor RJ45 pro komunikaci s PC. S ohledem na množství použitých součástek bude deska navržena jako čtyřvrstvá.

Pro univerzální modul musí být vyrobena krabička, ta bude navržena na míru podle vyrobené desky plošných spojů a vytištěna na 3D tiskárně. Bude navržen základní firmware, pomocí kterého bude možné ověřit správnost osazení a zapájení součástek.

Finální verze firmwaru musí umožňovat dálkové měření digitálních veličin, proto bude ve vytvořeném univerzálním modulu implementován firmware realizující funkci logického analyzátoru. Průběhy signálů budou měřeny hradlovým polem a naměřená data budou přes mikroprocesor odesílána pomocí UDP protokolu do počítače. V PC poběží aplikace napsaná v jazyce C#, která bude graficky interpretovat naměřená data.

2 HRADLOVÁ POLE

Hradlová pole jsou programovatelné logické obvody vhodné pro realizaci prakticky jakékoliv logické funkce (kombinační obvody, sekvenční obvody, stavové automaty). Oproti zákaznickým integrovaným obvodům (ASIC - Application Specific Integrated Circuits), které jsou vytvořeny na míru pro danou aplikaci, jsou hradlová pole rekonfigurovatelná a neslouží pouze jedinému účelu. Výhoda hradlových polí spočívá v implementování digitálního obvodu uvnitř pole s možností jeho jednoduché změny, nesoucí s sebou nízké ekonomické náklady.

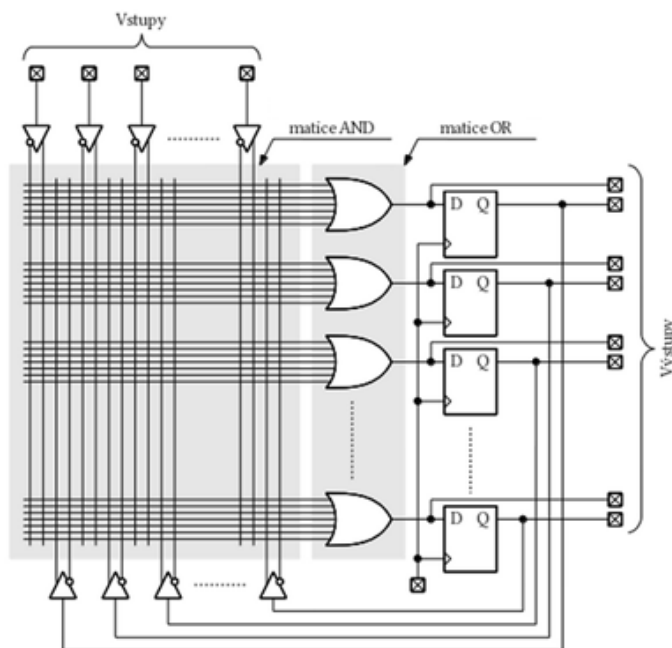
Tato kapitola slouží pouze jako přehled hradlových polí. Detailnější struktura konkrétního hradlového pole zvoleného pro vytvoření univerzálního modulu, bude uvedena v příslušné kapitole (4.1).

2.1 Rozdělení PLD

Programovatelné logické obvody (PLD - Programmable Logic Device) se dají rozdělit do tří skupin: klasické PLD (SPLD), komplexní PLD (CPLD) a obvody typu FPGA. Názvosloví není zcela jednotné a někdy jsou obvody typu FPGA brány jako samostatná skupina.

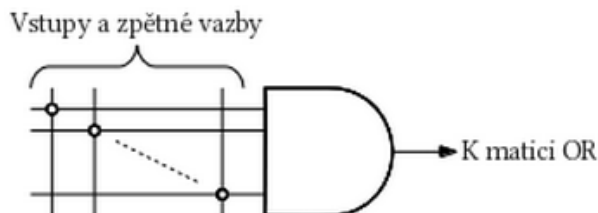
2.1.1 SPLD (Simple Programmable Logic Device) [11]

Jednoduché, nebo také klasické PLD (SPLD) jsou nejjednoduššími programovatelnými obvody, mají malý počet hradel (typicky několik stovek) a jejich charakteristická vnitřní struktura je na obrázku (Obrázek 2.1).



Obrázek 2.1: Struktura obvodů SPLD [11]

Ve schématu každá vodorovná čára v programovatelné matici AND představuje jedno součtové hradlo. Na vstupy každého hradla lze připojit kombinaci vstupních signálů, zpětných vazeb a jejich negací (počet vstupů hradla je omezen v závislosti na použité výrobní technologii). Zapojení jednoho součtového hradla je uvedeno na následujícím obrázku (Obrázek 2.2).



Obrázek 2.2: Součtové hradlo obvodu SPLD [11]

Pro možnost propojení vodičů v matici slouží programovatelné spínače, ty jsou na obrázku (Obrázek 2.2) představovány kružnicemi. Jejich realizace závisí na výrobní technologii obvodu (přepalovací kovová propojka, tranzistor řízený plovoucím hradlem, ...).

Podle možnosti programovatelnosti matic AND a OR obecné struktury SPLD z obrázku (Obrázek 2.1) se dají tyto obvody rozdělit do dvou podkategorií:

PAL (Programmable Array Logic)

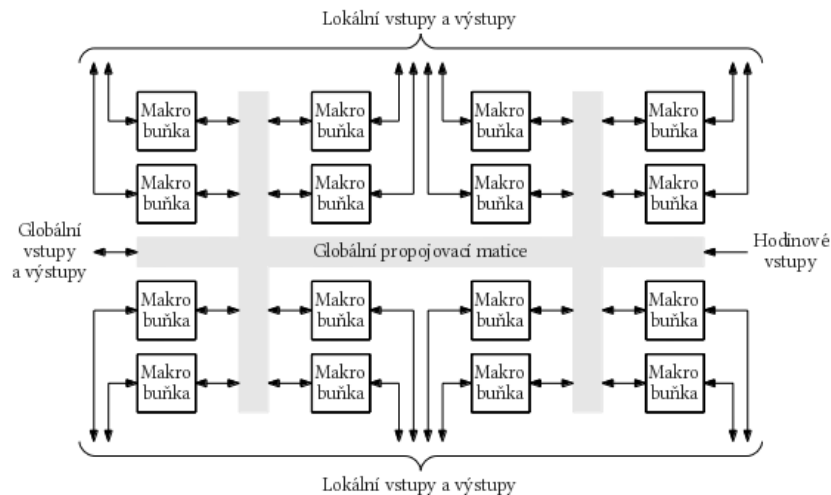
- mají programovatelnou pouze matici logických součinů (matici AND) a matice logických součtů (matice OR) je pevná
- do této podkategorie patří obvody typu PAL, GAL, PEEL a PALCE

PLA (Programmable Logic Array)

- mají programovatelnou jak matici logických součinů, tak matici logických součtů

2.1.2 CPLD (Complex Programmable Logic Device) [11]

Klasické PLD (SPLD) jsou univerzální obvodové prvky, avšak složitost implementovatelných funkcí je velmi omezená a je tedy možno s nimi realizovat pouze jednodušší funkce. Z tohoto důvodu došlo k integraci více PLD na jeden čip spolu s nutnými prostředky pro jejich vzájemné propojení, a tak vznikla hradlová pole typu CPLD (Complex Programmable Logic Device). Zjednodušená typická struktura obvodu CPLD je znázorněna na obrázku (Obrázek 2.3).



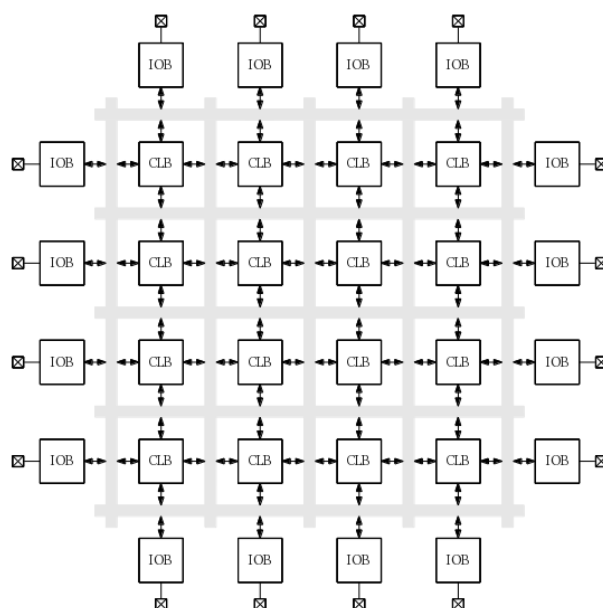
Obrázek 2.3: Struktura obvodů CPLD [11]

Realizace jednotlivých makrobuněk (macrocells) CPLD se liší v závislosti na použité technologii konkrétního výrobce (většinou vycházejí ze struktury PAL). Na čipu CPLD je integrována i non-volatilní paměť, takže po zapnutí napájení jsou tyto obvody schopny okamžitě začít plnit nakonfigurovanou funkci.

2.1.3 FPGA (Field Programmable Gate Array)

FPGA mají ze všech uvedených obvodů nejobecnější strukturu, obsahují největší množství využitelné logiky, a proto jsou i nejuniverzálnější. Zjednodušená typická struktura FPGA je uvedena na obrázku (Obrázek 2.4). [11]

Největšími výrobci těchto hradlových polí jsou Xilinx, Altera a Lattice Semiconductors. Jednotliví výrobci používají různá názvosloví, zde bude využito názvosloví přebrané od firmy Xilinx.



Obrázek 2.4: Základní struktura obvodů FPGA [11]

Hradlová pole FPGA se skládají z konfigurovatelných bloků CLB (configurable logic blocks), vstupně-výstupních buněk IOB (input-output blocks) a programovatelné propojovací matice. Potom záleží na jednotlivých výrobcích, jaké další prostředky do FPGA integrují. Typicky hradlová pole obsahují rychlá sériová vstupně-výstupní rozhraní, obvody synchronní blokové paměti, bloky pro realizaci aritmetických operací a bloky pro správu hodinových signálů DCM (digital clock manager). DCM bloky jsou například schopné násobit a dělit hodinovou frekvenci nebo nastavovat zpoždění. [12]

Obvody typu FPGA musejí být při zapnutí napájení (nebo po jeho výpadku) nakonfigurovány vždy znovu na rozdíl od hradlových polí typu CPLD.

2.1.3.1 CLB (configurable logic blocks - konfigurovatelné logické bloky) [11]

Jsou to základní bloky FPGA a umožňují vytvoření kombinačních i sekvenčních logických funkcí s využitím vlastní konfigurovatelné logiky. Tyto bloky se skládají z logických řezů (slices), které budou popsány dále. Bloky jsou mezi sebou spojeny propojovací maticí, jak je vidět na obrázku výše (Obrázek 2.4).

Každý logický řez běžně obsahuje 4 části:

- vyhledávací tabulku LUT (look-up table) – umožňuje realizaci kombinační funkce; je to v podstatě RAM paměť, jejíž obsah je dán konfigurací
- řetězec rychlého šíření přenosu – umožňuje vytváření aritmetických obvodů (sčítačky, násobičky); jeho vstupy a výstupy jsou propojeny se sousedními logickými bloky ve sloupci
- multiplexery – umožňují realizaci logických funkcí s více vstupy, než má jedna LUT tabulka (kombinací výstupů z několika LUT)
- registry – umožňují realizaci sekvenčních logických funkcí

2.1.3.2 IOB (input-output blocks – vstupně-výstupní bloky) [12]

Slouží k propojení hradlového pole s okolím. Všechny signály, jak přicházející z venku, tak vycházející z hradlového pole, jdou přes tyto bloky. IOB obsahují obvody pro ochranu vstupů a výstupů, impedanční přizpůsobení, zpožďovací linky, vstupní a výstupní registry, budiče a přijímače. Vstupy a výstupy podporují několik rozšířených standardů (LVTTTL, LVCMOS, ...). Ochranné obvody zajišťují logiku před poškozením vlivem elektromagnetického rušení nebo před nesprávným zapojením.

2.2 Programování (konfigurace) hradlových polí

U hradlových polí se pro jejich naprogramování používá výraz konfigurace. Výrobci poskytují nástroje pro syntézu návrhu a vytvoření konfiguračních souborů k jejich výrobkům většinou zdarma. Tyto nástroje jsou velice komplexní a umějí syntetizovat návrhy, vytvářet konfigurační soubory, simulovat chování obvodu a poskytují řadu optimalizačních nástrojů. Návrhy obvodu se dají vytvářet pomocí sestavení schématu nebo popsáním v jazyce HDL (VHDL – převážně Evropa, Verilog - Amerika), oba tyto přístupy se dají kombinovat.

2.2.1 Jazyk VHDL

Jazyk VHDL (VHSIC Hardware Description Language) začal vznikat v roce 1981 v rámci výzkumného projektu VHSIC (Very High Speed Integrated Circuits) ministerstva obrany Spojených států amerických. Cílem projektu VHSIC bylo vytvořit efektivní popis velmi rozsáhlých integrovaných obvodů, který by nebyl závislý na cílové technologii, neboť popis obvodů byl do té doby řešen pomocí vzájemně nekompatibilních nástrojů a jazyků. [2]

Základní definici jazyka vytvořili firmy IBM, Intermetrics a Texas Instruments, jazyk VHDL byl poprvé veřejně publikován v roce 1985. Další vývoj a standardizaci předalo ministerstvo obrany organizaci IEEE (Institute of Electrical and Electronics Engineers), která má každých 5 let jazyk revidovat. [2]

VHDL je tedy jazyk vysoké úrovně a slouží k popisu a simulaci rozsáhlých číslicových obvodů (pro popis hardware). Má bohaté vyjadřovací schopnosti a je značně nezávislý na konečné použité technologii (použití pro CPLD i FPGA).

V jazyce VHDL jsou vytvářeny moduly a složité systémy jsou zpravidla popisovány hierarchickou strukturou jednotlivých modulů. Modul je sestavený z deklarace entity, čímž jsou definovány porty modulu (vstupy a výstupy) a z popisu architektury, čímž je stanoveno jeho chování. Jazyk VHDL nerozlišuje velká a malá písmena v klíčových slovech a názvech identifikátorů. Na velikosti písmen však záleží u řetězců nebo znaků ve výčtových typech.

3 KOMUNIKAČNÍ SBĚRNICE

Tato kapitola popisuje některé standardy komunikačních sběrnic. Vytvořená deska pro univerzální modul disponuje možností komunikace podle všech standardů popsanych v této kapitole.

3.1 UART

UART (Universal Asynchronous Receiver Transmitter) je zkratka pro univerzální asynchronní vysílač a přijímač. Jedná se tedy o hardwarové zařízení pro sériovou asynchronní komunikaci, které v sobě má vysílač i přijímač. Každá UART jednotka má posuvný registr, který je základním prvkem pro převod paralelní informace na sériovou. Běžně využívá komunikační standardy RS-232, RS-422, RS-485. [22] Pro komunikaci mezi mikroprocesory a integrovanými obvody se nejvíce využívá UART zařízení s TTL nebo CMOS logikou, která má logickou úroveň nula reprezentovanou hodnotou napětí 0 V a logickou úroveň jedna podle zvoleného standardu.

Základní piny UART jednotky nezbytné pro komunikaci:

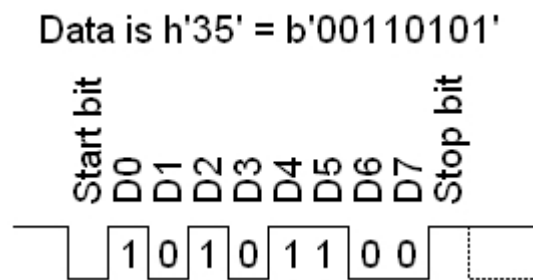
- TxD – pin pro odesílání dat
- RxD – pin pro příjem dat
- GND – propojení se zemí

3.1.1 Asynchronní sériová komunikace [18]

U asynchronního přenosu dat není přijímač a vysílač synchronizován pomocí společného hodinového signálu. Synchronizace obou zařízení probíhá pokaždé, když je vyslána sekvence bitů pomocí takzvaných start a stop bitů. Sekvence přenášených bitů se skládá právě ze start bitu následovaného sérií datových bitů (datovým slovem), za nimi může být přenesen paritní bit a celá sekvence je ukončena stop bitem.

- start bit – sériová linka je v klidovém stavu na hodnotě logické jedničky; příchodem start bitu, který je reprezentován logickou nulou, je zahájen přenos dat
- série datových bitů (datové slovo) – může být tvořena pěti až osmi bity, protože bajt je tvořen právě osmi bity, používá se nejčastěji tento počet; jako první se odesílá nejméně významný bit (LSB)
- paritní bit – používá se pro kontrolu správnosti přenesení datových bitů, paritní bit může a nemusí být v sekvenci přenášen
- stop bit – ukončuje sekvenci přenosu; je reprezentován logickou jedničkou; po jeho odeslání je možno vyslat další sekvenci; stop bit může mít šířku jednoho, dvou, nebo jednoho a půl bitu

Aby bylo možné sériovou komunikaci uskutečnit, musí být na začátku nastaveny parametry pro přijímač a vysílač, jako je rychlost přenosu, šířka datového slova, šířka stop bitu a jestli bude nebo nebude přenášen paritní bit a jeho typ. Typické rychlosti přenosu sériových linek se odvíjí od násobku 300 bitů za sekundu, někdy se rychlost udává v jednotkách "baud" (při sériové asynchronní komunikaci 1 b/s = 1 baud).



Obrázek 3.1: Asynchronní sériový přenos [21]

Sériová linka může fungovat v simplexním, poloduplexním a plně duplexním režimu přenosu dat.

- simplexní (jednosměrný) režim – komunikace funguje pouze jednosměrně od vysílače k přijímači, na propojení stačí pouze dva vodiče, jeden propojuje TxD pin vysílače s RxD pinem přijímače a druhý propojuje zem
- poloduplexní režim – obousměrná komunikace, avšak zařízení může buďto přijímat nebo vysílat, propojení je opět provedeno dvěma vodiči, to tak, že jsou nejprve mezi sebou spojeny TxD a RxD piny přijímače i vysílače a ty pak propojeny jedním vodičem mezi vysílačem a přijímačem, druhý vodič propojuje zem
- plně duplexní režim – obousměrná komunikace, kde zařízení může zároveň přijímat i vysílat, propojení je provedeno třemi vodiči, kde dva vodiče mezi sebou křížem propojují TxD a RxD piny vysílače a přijímače a třetí vodič propojuje zem

Pro řízení toku dat se může použít mechanismus potvrzování (tzv. handshaking). Pokud je použit, může být buďto softwarový (XON/XOFF). Nebo hardwarový, při kterém se využívají další vodiče a jsou dva způsoby řízení (RTS/CTS, nebo DTR/DSR).

3.2 USART

USART (Universal Synchronous Asynchronous Receiver Transmitter) je zkratka pro univerzální asynchronní a synchronní vysílač a přijímač. Je to zařízení schopné pracovat jak v sériovém synchronním, tak i sériovém asynchronním módu. Asynchronní sériová komunikace byla popsána v předcházející kapitole (UART). Při synchronní sériové komunikaci jsou zařízení propojena hodinovým signálem, který řídí přenos dat.

Jednotka USART v synchronním režimu tedy poskytuje hardwarové možnosti pro realizování synchronních komunikačních protokolů, takový protokol může být třeba SPI.

Základní piny USART jednotky nezbytné pro komunikaci:

- TxD – pin pro odesílání dat
- RxD – pin pro příjem dat
- SCK – hodinový signál pro řízení komunikace (pouze u synchronního přenosu)
- GND – propojení se zemí

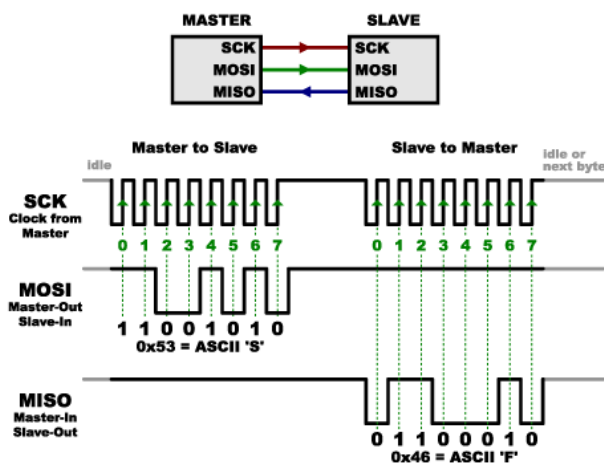
3.2.1 SPI

SPI (Serial Peripheral Interface) je zkratka pro sériové periferní rozhraní. Jak již bylo uvedeno, jedná se o sériový synchronní komunikační protokol. Běžně se využívá při komunikaci mezi mikroprocesorem a jeho periferiemi. Pro komunikaci je možné nastavit čtyři různé módy přístupu k datům podle vztahu k hodinovému signálu. Tyto módy jsou nastavovány dvěma volbami, a to nastavením klidové úrovně a výběrem první, nebo druhé hrany hodinového signálu (náběžné, nebo sestupné).

Toto rozhraní pracuje v režimu slave a master, kde master řídí hodinový signál. Komunikace je plně-duplexní a pro propojení dvou zařízení pomocí SPI, jsou nutné 3 signály:

- SCK (serial clock) – hodinový signál pro řízení komunikace
- MOSI (master out slave in) – komunikační linka od zařízení pracujícího jako master k zařízení slave
- MISO (master in slave out) – komunikační linka od zařízení slave k zařízení master
- SS (slave select) – volitelný signál pomocí kterého master vybírá, s kým bude komunikovat

Na následujícím obrázku (Obrázek 3.2) je uvedeno propojení dvou zařízení a princip komunikace, kdy data jsou přenášena s náběžnou hranou hodinového signálu.



Obrázek 3.2: Komunikace SPI rozhraní [13]

3.3 JTAG

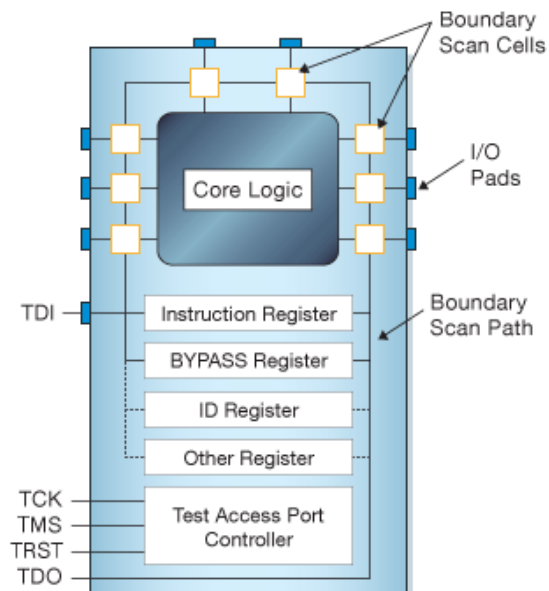
JTAG (Joint Test Action Group) se běžně používá pro označení normy IEEE 1149.1: Standard Test Access Port and Boundary Scan Architecture.

Boundary Scan je metoda pro testování plošných spojů a lze pomocí ní testovat například, jestli jsou součástky správně zapájené.

Rozhraní JTAG je známo právě jako Test Access Port (TAP) a používá následující signály pro ovládání stavového automatu (stavový automat řídí chování JTAG systému):

- TCK (Test Clock) – hodinový signál synchronizující stavový automat
- TMS (Test Mode Select) – pro řízení posunu mezi stavy automatu, je vzorkován při náběžné hraně TCK
- TDI (Test Data In) – datový vstup, vzorkován při náběžné hraně hodinového signálu a při vzorkování dochází k posunu dat v registrech
- TDO (Test Data Out) – datový výstup, hodnota na výstupu je platná při sestupné hraně hodinového signálu
- TRST (Test Reset) – volitelný signál, umožňuje resetovat stavový automat

Pomocí JTAG systému je možné rovněž programovat flash paměti, mikroprocesory nebo hradlová pole. Umožňuje propojení více zařízení do řetězce.



Obrázek 3.3: Schéma JTAG systému [14]

3.4 Ethernet

Je to standard pro síťové technologie, popisuje požadavky na parametry propojovací kabeláže (přenosového média) a způsob přístupu k síti. Je definován normou IEEE 802.3 (literatura [3]).

Existují různé verze Ethernetu, v této práci se budu zmiňovat pouze o dvou typech Ethernetu a to standard Ethernet (10 Mbit/s) a Fast Ethernet (100 Mbit/s). Verze se dále dělí podle způsobu propojení (podle použitého přenosového média). Nejběžněji využívané přenosové médium je kroucená dvojlinka. To odpovídá dvěma verzím, kterými se budu dále zabývat a to 10Base-T pro Ethernet a 100Base-TX pro Fast Ethernet. Další druhy přenosového média jsou koaxiální kabel (už se nepoužívá, byl pouze pro 10 Mbit/s) a optický kabel.

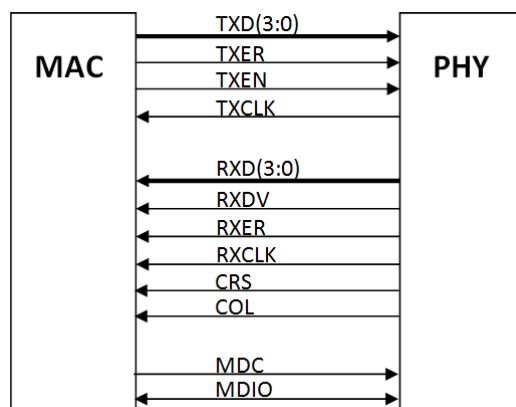
Kabel, kterým jsou propojena jednotlivá zařízení, je tvořen dvěma páry kroucených dvojlinek, kde jeden pár slouží pro odesílání a druhý pro příjem dat. Kroucená dvojlinka tvoří diferenciální pár, přes který jsou data přenášena. Data jsou při přenosu kódována do Manchester kódu.

Pro vysílání a přístup ke sdílenému přenosovému médiu se využívá metoda CSMA/CD (Carrier Sense with Multiple Access and Collision Detection), je to metoda, která zajišťuje, že může vysílat pouze jedno zařízení. Při použití navrhovaného univerzálního modulu se však nepředpokládá použití sdíleného přenosového média.

Komunikace může probíhat v režimu half-duplex (poloviční duplex), kdy je aktivní komunikace pouze v jednom směru. Nebo full-duplex (plný duplex), kdy může zároveň probíhat vysílání i příjem dat.

3.4.1 MII [3][26]

MII (Media Independent Interface) je zkratka pro nezávislé rozhraní. Definuje propojení mezi MAC (Media Access Control) a PHY (Physical Layer) a díky nezávislosti tohoto rozhraní je možné propojit různé typy zařízení. Rozhraní je definováno pro obě verze ethernetu, jak pro 10 Mbit/s, tak pro 100 Mbit/s. Data jsou přes MII přenášena čtyřmi datovými vodiči v každém směru jako čtyřbitová slova. Čtyřbitové slovo se označuje jako „nibble“.



Obrázek 3.4: Propojení MII rozhraní

Vodiče rozhraní MII se dají rozdělit do tří základních skupin:

1) Signály pro odesílání dat (do PHY):

- TXD0 – odesílaný datový bit 0 (odesílán jako první)
- TXD1 – odesílaný datový bit 1
- TXD2 – odesílaný datový bit 2
- TXD3 – odesílaný datový bit 3
- TXEN (transmit enable) – signalizace, že data na vodičích TXD(0:3) mají být přenesena
- TXER (transmit error) – signalizuje chybu při přenosu a pokud je aktivní, data na vodičích TXD(0:3) nejsou přenesena; přítomnost tohoto signálu je volitelná a využívá se jenom zřídka
- TXCLK (transmit clock) – hodinový signál poskytující časování pro příjem dat; pokud je aktivní signál TXEN, tak s náběžnou hranou TXCLK jsou přenášena data z TXD(0:3); jedním hodinovým taktem jsou tedy přeneseny 4 bity (nibble); zdrojem hodinového signálu je PHY; hodinový signál má frekvenci 25 MHz pro rychlost 100 Mb/s, nebo 2,5 MHz pro 10 Mb/s

2) Signály pro příjem dat (z PHY):

- RXD0 – přijímaný datový bit 0 (přijímaný jako první)
- RXD1 – přijímaný datový bit 1
- RXD2 – přijímaný datový bit 2
- RXD3 – přijímaný datový bit 3
- RXDV (receive data valid) – signalizuje, že na vodičích RXD(0:3) jsou připravena data pro příjem
- RXER (receive error) – signalizuje chybu při odesílání dat

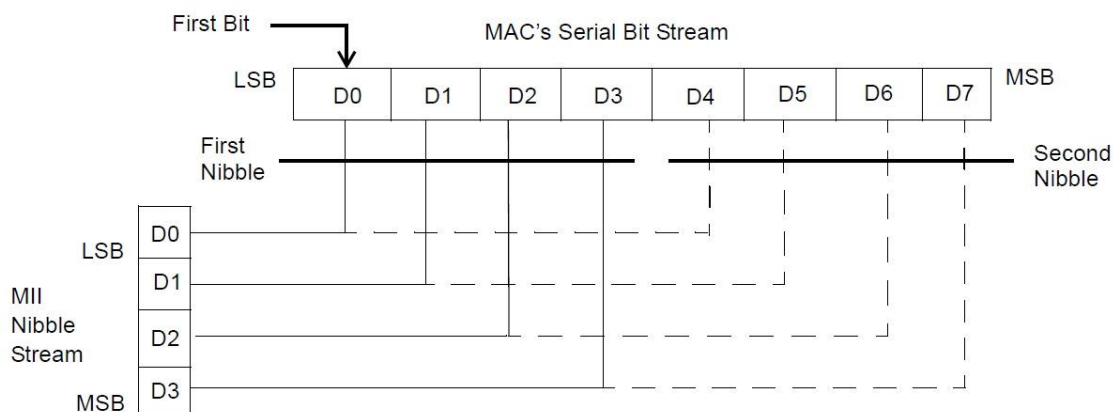
- CRS (carrier sense) – v half-duplex módu signalizuje, že probíhá odesílání nebo příjem dat; pokud je nastaven režim full-duplex, tak jeho chování není v normě specifikováno
- COL (collision detected) – signalizuje, jestli došlo ke kolizi (pouze half-duplex) a v případě rychlosti přenosu 10 Mb/s reaguje na SQE (signal quality error)
- RXCLK – hodinový signál poskytující časování pro odesílání dat; pokud je aktivní signál RXDV, tak s náběžnou hranou RXCLK jsou přenášena data na RXD(0:3); zdrojem hodinového signálu je PHY; hodinový signál má frekvenci 25 MHz pro rychlost 100 Mb/s, nebo 2,5 MHz pro 10 Mb/s

3) Signály pro konfiguraci PHY (SMI rozhraní – Serial Management Interface):

- MDC (management data clock) – externí hodinový signál (musí být přiveden do PHY) poskytuje časování pro MDIO
- MDIO (management data input/output) – obousměrný signál pro přenos dat sloužících k nastavení parametrů PHY

3.4.1.1 Způsob přenosu dat přes MII rozhraní

Každý oktet je přenesen za pomoci dvou čtyřbitových slov ($2 \times \text{nibble}$). Data jsou přenášena od nejméně významného bitu (LSB – least significant bit), jako poslední je přenesen nejvýznamnější bit (MSB – most significant bit). Způsob přenosu je znázorněn na obrázku (Obrázek 3.5).



Obrázek 3.5: Způsob přenosu dat přes MII rozhraní [3]

3.4.2 Ethernetový rámec [23]

Na ethernetu jsou data přenášena pomocí rámců v pořadí, ve kterém je nejprve přenesen nejvíce významný oktet (oktet = skupina osmi bitů). Uvnitř oktetu je ale nejprve transportován nejméně významný bit. Uspořádání rámce je uvedeno v tabulce (Tabulka 3.1). Přenášená data i s hlavičkami uvnitř rámce jsou označována jako pakety (pro spolehlivý přenos), nebo datagramy (pro „nespolehlivý“ přenos).

V textu budou zapisovány hodnoty v hexadecimálním tvaru, kvůli rozpoznání budou zapisovány ve tvaru "0xhodnota".

Tabulka 3.1: Ethernetový rámec

preamble	SFD	MAC cíle	MAC zdroje	802.1Q tag (volitelný)	typ/délka	data	FCS	zpoždění mezi rámci
7 oktetů	1 oktet	6 oktetů	6 oktetů	4 oktety	2 oktety	(42) 46 - 1500 oktetů	4 oktety	12 oktetů
64 - 1522 oktetů								
84 - 1542 oktetů								

Popis jednotlivých polí:

- preamble - začátek rámce; slouží pro synchronizaci; přenášeny střídavě hodnoty log. 1 a log. 0 (7× oktet – 10101010)
- SFD (Start of frame delimiter) – označení pro začátek rámce; definován formát oktetu - 10101011
- MAC cíle - MAC adresa cílového zařízení
- MAC zdroje – MAC adresa zařízení, ze kterého je rámec odeslán
- 802.1Q tag – volitelné pole; pokud je v rámci obsaženo, specifikuje informace pro virtuální LAN síť
- typ/délka – podle hodnoty v tomto poli se rozhoduje, jestli následující datové pole bude určeno pouze délkou dat, nebo typem vyššího protokolu
 - délka – pro hodnoty menší nebo rovny 1500 (0x05DC), udává tato hodnota délku následujícího datového pole (1500 je maximální délka datového pole)
 - typ – pro hodnoty větší nebo rovny 1536 (0x0600), hodnota v poli určuje typ vyššího protokolu, který bude použit v následujícím datovém poli; například hodnota 0x0800 znamená, že bude použit protokol IPv4
- data – datové pole; jeho minimální délka je 46 oktetů (42 oktetů pokud rámec obsahuje pole 802.1Q tag); pokud chceme přenášet menší objem dat, musí být pole vyplněno do minimální délky

- FCS (Frame Check Sequence) – kontrolní pole, ve kterém se přenáší 32bitový CRC; CRC bude popsán v kapitole (3.4.4)
- zpoždění mezi rámci – před odesláním dalšího rámce musí být vložena mezera dlouhá minimálně 12 oktetů

3.4.3 IPv4 protokol [15]

Při přenosu dat na ethernetový port bude používán protokol IPv4 (Internet Protocol version 4), takže je zde uveden jeho popis. Je to první masově rozšířená verze Internet Protokolu. Je popsán v RFC 791 (literatura [15]).

Formát IPv4 protokolu je uveden níže v tabulce (Tabulka 3.2), je tvořen hlavičkou, za kterou následují přenášená data. První řádek naznačuje rozdělení po oktetech a druhý po bitech, nejvýznamnější bit (MSB) je vlevo. Celý tento blok (IPv4 protokol) je přenášen v poli „data“ ethernetového rámce popsaného v předcházející kapitole (3.4.2).

Tabulka 3.2: Formát protokolu IPv4

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
verze				IHL				TOS								celková délka															
identifikace																příznaky				fragment offset											
TTL								protokol								kontrolní součet hlavičky															
zdrojová IP adresa																															
cílová IP adresa																															
volitelný blok (když IHL > 5)																															
data ...																															

Popis jednotlivých polí:

- verze – verze IP protokolu; pro IPv4 je to 4 (0x4)
- IHL (Internal Header Length) – délka hlavičky v bajtech; číslo je násobeno čtyřmi; minimální hodnota je 5 (0x5), to odpovídá délce hlavičky 20 bajtů ($5 \times 4 = 20$) a maximální délka je 60 bajtů (0xF)
- TOS (Type of Service) – původně mělo pole sloužit pro zvolení charakteru přepravy rámce; nyní je nahrazeno poli s podobnou funkcí, která obsahuje informace pro službu QoS (Quality of Service); QoS řídí datový tok
- celková délka – délka datagramu v bajtech, to znamená délku hlavičky i s daty, která za ní následují
- identifikace – každému datagramu je přidělen identifikátor; identifikátor slouží k rozpoznání, které datagramy patří k sobě, pokud jsou data fragmentována (mají stejný identifikátor)

- příznaky – pro řízení fragmentace:
 - první bit – rezervován; musí být 0
 - druhý bit – DF (Don't Fragment); pokud je 1, data nejsou fragmentována
 - třetí bit – MF (More Fragments); pokud je 0, právě přenášený datagram je poslední fragment
- fragment offset – udává, na jakou pozici fragment patří; hodnota je udávána v počtu osmi bajtů
- TTL (Time To Live) – udává životnost datagramu; slouží jako ochrana proti zacyklení, kde je hodnota snižována při průchodu směrovačem, nebo každou sekundu; když hodnota dojde na nulu, přenášený datagram je zahozen
- protokol – udává, jaký vyšší protokol bude dále použit, například UDP protokol má číslo 17 (0x11).
- kontrolní součet hlavičky (checksum) – slouží pro detekci chyb, je vypočítán z dat v hlavičce, jeho výpočet bude popsán v kapitole (3.4.3.1)
- zdrojová IP adresa – IP adresa zařízení, ze kterého datagram přichází
- cílová IP adresa – IP adresa zařízení, na které datagram směřuje
- volitelný blok – obvykle není využit, nabízí další možnosti pro přenos datagramu
- data – formát přenosu data v tomto poli, záleží na typu dalšího protokolu, který bude pro přenos použit (například protokoly UDP, TCP, ...)

3.4.3.1 Checksum (kontrolní součet)

Kontrolní součet je vypočítán vždy z určitých dat a slouží jako ochrana proti přijmutí dat poškozených při přenosu. Pokud je kontrolní součet při ověřování jeho správnosti chybný, jsou přenášená data zahozena.

Výpočet kontrolního součtu se provede jako jedničkový doplněk součtu 16bitových hodnot. To znamená, že data, ze kterých je výpočet prováděn, se rozdělí do bloků po šestnácti bitech a provede se součet těchto bloků. Pokud při součtu vznikne přenos, přičte se k výsledku. Nad výsledek součtu se provede jedničkový doplněk (bitová inverze) a výsledkem této operace je požadovaný kontrolní součet.

Pokud se výpočet provede znovu se zahrnutím kontrolního součtu, měl by být výsledek roven 0. Data v takovém případě byla přenesena v pořádku.

Při výpočtu kontrolního součtu hlavičky IPv4 se výpočet provádí z celé hlavičky s tím, že pole kontrolního součtu se vyplní nulami.

3.4.3.2 UDP protokol [16]

UDP (User Datagram Protocol) je jeden z rodiny protokolů TCP/IP. Je definován v RFC 768 (literatura [16]). UDP je nespojový protokol (někdy označován jako nespolehlivý), to znamená, že nenavazuje spojení mezi odesílatelem a příjemcem, a tak nezaručuje doručení jednotlivých datagramů nebo pořadí jejich doručení. Naopak je ale jednoduchý a má nízké režijní nároky.

Formát UDP je uveden v tabulce (Tabulka 3.3). První řádek naznačuje rozdělení po oktetech a druhý po bitech, nejvýznamnější bit (MSB) je vlevo.

Tabulka 3.3: Formát protokolu UDP

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
zdrojový port (volitelný)																cílový port															
Délka																kontrolní součet (volitelný)															
data ...																															

Popis jednotlivých polí:

- zdrojový port – číslo zdrojového portu; pokud není použit, měl by se nastavit na nulu
- cílový port – číslo cílového portu; jednotlivé aplikace mají přidělena vlastní čísla portů, na kterých komunikují
- délka – délka UDP hlavičky a dat v bajtech
- kontrolní součet – kontrolní součet pro hlavičku a data, je volitelný; pokud ho nechceme využívat, stačí pole vyplnit nulami; počítá se z pseudo-hlavičky IPv4, kam patří zdrojová a cílová IP adresa, číslo protokolu (doplněné zleva o nuly), délka UDP a celá hlavička UDP i s daty
- data – toto pole už obsahuje uživatelská data, která chceme přenést pomocí UDP/IP protokolu

3.4.4 CRC [3]

CRC (Cyclic Redundancy Check) je zkratka pro cylindrický redundantní součet. Je to také způsob výpočtu kontrolního součtu. Používá se pro kontrolu přenášených dat, jeho výsledek je připojen za těmito daty a přenášen společně s nimi. Vždy při přijetí dat je znovu přepočítán a v případě, že nesouhlasí, data jsou zahozena. V případě zahození dat kvůli chybnému CRC o tom není předávána žádná zpráva dále, takže není možné zjistit, že přenášený rámec byl zahozen.

Výsledný CRC se určí jako zbytek po dělení vstupních dat generačním polynomem, kde nad výslednými koeficienty zbytku je provedena operace modulo 2. Generační

polynom se dá snadno převést na bitovou posloupnost. Při výpočtech CRC z bitových posloupností se využívá funkce XOR. Generační polynom může mít různou délku a různý tvar.

Výpočet se provádí ze všech polí rámce kromě preamble, SFD, FCS (do kterého je výsledek uložen) a samozřejmě zpoždění mezi rámci. Pro ethernetové rámce se vypočítává 32bitový cylindrický redundantní součet označovaný jako CRC32.

Jeho generační polynom je 32. řádu a má tvar:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

To odpovídá bitové posloupnosti:

"1 0000 0100 1100 0001 0001 1101 1011 0111"

Příklad výpočtu pro 5bitový CRC

Pro jednoduchost bude proveden příklad výpočtu pouze pro 5bitový CRC, jako generační polynom bude zvolen polynom, který se používá pro CRC5-USB.

To je generační polynom: $x^5 + x^2 + 1$ (polynom 5. řádu)

Odpovídá bitové posloupnosti: 100101

Zvolená datová posloupnost: 10101010

Výpočet v bitové reprezentaci polynomů, využívá se operace XOR:

1 0 1 0 1 0 1 0	0 0 0 0 0	nejprve jsou data doplněna zprava o tolik nul, jakého řádu je polynom
1 0 0 1 0 1		pod data na místě první jedničky zleva zapíšeme polynom
0 0 1 1 1 1 1 0	0 0 0 0 0	provedeme operaci XOR a zapíšeme výsledek
1 0 0 1 0 1		posuneme polynom na místo další jedničky
0 0 0 1 1 0 1 1	0 0 0 0 0	XOR
1 0 0 1 0 1	1	posun
0 0 0 0 1 0 0 1	1 0 0 0 0	XOR
1 0 0 1 0 1	0 1	posun
0 0 0 0 0 0 0 0	1 1 0 0 0	opakujeme, dokud na místě původních dat nebudou samé nuly, potom se na místě přidaných nul objeví výsledný CRC

Výsledný CRC je tedy "11000", to odpovídá polynomu $(x^4 + x^3)$. Stejný výsledek vyjde, pokud budeme počítat CRC přímo z polynomů, kde je nejprve datový polynom vynásoben proměnnou $x^{(\text{řád polynomu})}$, v tomto případě x^5 (to je to stejné jako přidání nul na konec datové posloupnosti). Potom se tento polynom vydělí generačním polynomem a na koeficienty zbytku po dělení se aplikuje modulo 2.

Tabulka 3.4: První tabulka pro sestavení paralelního CRC

	CRC (4)	CRC (3)	CRC (2)	CRC (1)	CRC (0)
D (0)	0	0	1	0	1
D (1)	0	1	0	1	0
D (2)	1	0	1	0	0
D (3)	0	1	1	0	1
D (4)	1	1	0	1	0
D (5)	1	0	0	0	1
D (6)	0	0	1	1	1
D (7)	0	1	1	1	0

- 4) Vypočítáme si sériový CRC pro $\underline{M}$ hodnot, když $D = 0$. Každá hodnota $\underline{M}$ je opět zapsána ve one-hot kódu. Protože CRC polynom je kratší než data ($M < D$), budou se výsledky opakovat a je zbytečné je všechny počítat znovu. Pro tento příklad $D(3) \approx M(0)$, $D(4) \approx M(1)$, ...
- 5) Sestavíme tabulku (Tabulka 3.5) z výsledků v předchozím bodě, kde $CRC(x)$ je funkcí $\underline{M}$.

Tabulka 3.5: Druhá tabulka pro sestavení paralelního CRC

	CRC (4)	CRC (3)	CRC (2)	CRC (1)	CRC (0)
M (0)	0	1	1	0	1
M (1)	1	1	0	1	0
M (2)	1	0	0	0	1
M (3)	0	0	1	1	1
M (4)	0	1	1	1	0

Druhá fáze:

Z tabulek vytvořených v první fázi sestavíme předpis pro výpočet paralelního CRC. $CRC(x)$ je funkcí $\underline{M}$ a $\underline{D}$, takže pro každé $CRC(x)$ vybereme z tabulek (Tabulka 3.4, Tabulka 3.5) řádky, ve kterých je hodnota 1 a mezi jednotlivými členy provádíme operaci XOR. Operace XOR je značena jako '^'.

$$\begin{aligned}
 CRC(0) &= D(0) \wedge D(3) \wedge D(5) \wedge D(6) \wedge M(0) \wedge M(2) \wedge M(3) \\
 CRC(1) &= D(1) \wedge D(4) \wedge D(6) \wedge D(7) \wedge M(1) \wedge M(3) \wedge M(4) \\
 CRC(2) &= D(0) \wedge D(2) \wedge D(3) \wedge D(6) \wedge D(7) \wedge M(0) \wedge M(3) \wedge M(4) \\
 CRC(3) &= D(1) \wedge D(3) \wedge D(4) \wedge D(7) \wedge M(0) \wedge M(1) \wedge M(4) \\
 CRC(4) &= D(2) \wedge D(4) \wedge D(5) \wedge M(1) \wedge M(2)
 \end{aligned}$$

Nyní je vytvořen předpis, do kterého stačí už jen dosazovat příslušné hodnoty, a tak získáme výsledný CRC.

Dosazení konkrétních dat ("10101010"), za podmínky $M = 0$. Data jsou reprezentována tak, že nejvýznamnější bit je vlevo (má index 7):

$$\text{CRC}(0) = 0 \wedge 1 \wedge 1 \wedge 0 = 0$$

$$\text{CRC}(1) = 1 \wedge 0 \wedge 0 \wedge 1 = 0$$

$$\text{CRC}(2) = 0 \wedge 0 \wedge 1 \wedge 0 \wedge 1 = 0$$

$$\text{CRC}(3) = 1 \wedge 1 \wedge 0 \wedge 1 = 1$$

$$\text{CRC}(4) = 0 \wedge 0 \wedge 1 = 1$$

Konečný výsledek: $\text{CRC} = "11000"$. To je stejný výsledek jako v případě výpočtu sériového CRC v kapitole (3.4.4).

4 HARDWARE UVAŽOVANÝ PŘI NÁVRHU

V této kapitole bude popsán některý hardware použitý pro vytvoření desky univerzálního modulu, případně pro její programování.

4.1 Hradlové pole (FPGA) - Spartan-3A

Spartan-3A je hradlové pole typu FPGA (Field Programmable Gate Array) od výrobce Xilinx. Pro návrh byl vybrán model XC3S200A-4VQG100I, kde:

- XC3S200A – typ a velikost pole
- -4 je rychlost obvodu (-4 = standardní)
- VQG100 značí typ pouzdra
 - VQ – pouzdro typu VQFP (Very Thin Quad Flat Pack)
 - G – Pb-free technologie výroby (v souladu se směrnicí Evropského parlamentu a Rady o omezení používání některých nebezpečných látek v elektrických a elektronických zařízeních)
 - 100 – počet pinů pouzdra je 100
- I (Industrial) označuje teplotní rozsah ($-40\text{ °C} - 100\text{ °C}$)

Byla vybrána průmyslová verze hradlového pole, protože oproti komerční verzi byla levnější asi o 70 korun. Průmyslová verze by podle katalogových listů výrobce měla podporovat pouze standardní rychlost obvodu (-4), nicméně došla verze označená jako duální (5C/4I), která je schopna pracovat ve vyšší rychlosti (-5) při zajištění teplotního rozsahu pro komerční verzi ($0\text{ °C} - 85\text{ °C}$).

Obsahuje 200 000 logických hradel sdružených do 448 programovatelných logických bloků (CLB - Configurable Logic Block). Každý CLB je tvořen čtyřmi logickými řezy („slices“). Rozhraní hradlového pole podporuje nejpoužívanější logické

standards jako LVCMOC, LVTTL, HSTL, SSTL, a proto také využívá i různé úrovně napětí, a to 3.3 V, 2.5 V, 1.8 V, 1.5 V a 1.2 V. Nejvyšší dovolený odběr proudu na jednom pinu je 24 mA. Má 68 I/O pinů a z toho 6 může být pouze vstupních. Dále obsahuje 4 bloky pro digitální správu hodin (DCM - Digital Clock Manager), 16 násobiček a 288 Kb blokové paměti rozdělené do 16 bloků po 18 Kb. Detailnější popis je možno najít v odkazu (literatura [4]). [4]

Původně bylo uvažováno větší hradlové pole XC3S400, ale z důvodu dostupnosti bylo vybráno poloviční XC3S200. Při výběru ještě menšího pole, už by byly příliš omezeny aplikace, na které by se dala deska použít. Typ A (XC3S200A) byl vybrán z důvodu možnosti napájet V_{CCAUX} napětím jak 2,5 V, tak 3,3 V. U standardního modelu může být V_{CCAUX} napájeno pouze 2,5 V (standardní model pro nové návrhy nedoporučuje používat už ani výrobce). Při napájení V_{CCAUX} z 3,3 V se značně zjednoduší návrh desky, protože nebude potřeba rozvádět napětí 2,5 V. Zároveň se zlepší i kompatibilita, kde například při programování pomocí rozhraní JTAG, které má standardně napětí 3,3 V, by bylo potřeba konvertovat napěťové úrovně.

V tabulce (Tabulka 4.1) jsou uvedeny maximální dovolené hodnoty napětí přivedených na napěťové vstupy hradlového pole.

Tabulka 4.1: Maximální absolutní hodnoty napětí [4]

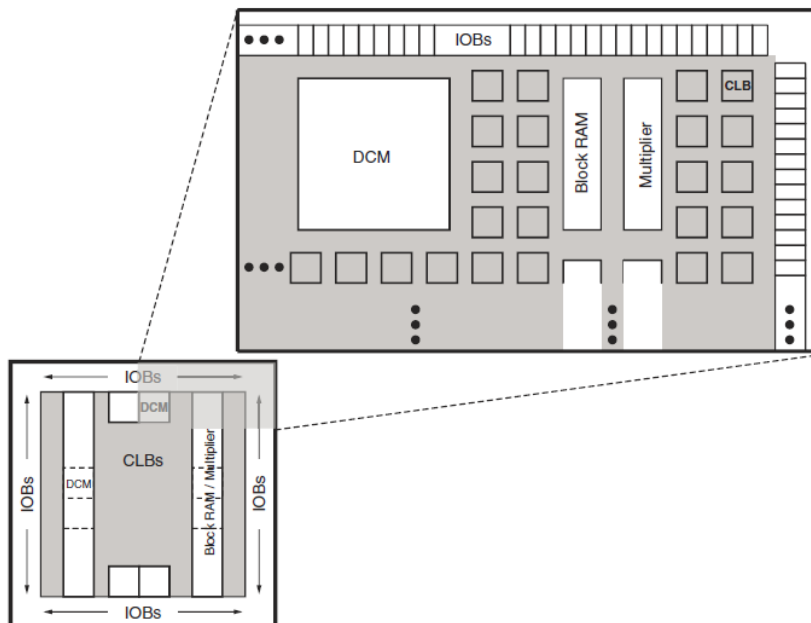
symbol	popis	Min	Max
V_{CCINT}	vnitřní napájení (napájení jádra)	-0,5V	1,32 V
V_{CCAUX}	pomocné napájení	-0,5V	3,75 V
V_{CCO}	napájení výstupních řadičů	-0,5V	3,75 V
V_{REF}	vstupní referenční napětí	-0,5V	$V_{CCO} + 0,5 V$

4.1.1 Architektura Spartan-3A [4]

Architektura rodiny Spartan-3A se skládá z pěti základních prvků:

- 1) CLB – konfigurovatelný logický blok obsahuje vyhledávací tabulky (LUT) a klopné obvody, tím poskytuje různé logické funkce a možnost ukládání dat
- 2) IOB – vstupní/výstupní bloky kontrolují tok dat mezi I/O piny a logikou uvnitř hradlového pole; podporují obousměrný tok dat, třístavové výstupy a různé signálové standardy
- 3) bloková RAM – každý z 16 bloků je schopen uložit 18 Kb
- 4) bloky násobiček – schopné násobit dvě 18bitová čísla
- 5) DCM (Digital Clock Manager) – bloky pro správu hodin poskytující digitální řešení pro kalibraci, šíření, zpoždování, násobení, dělení a fázový posun hodinových signálů

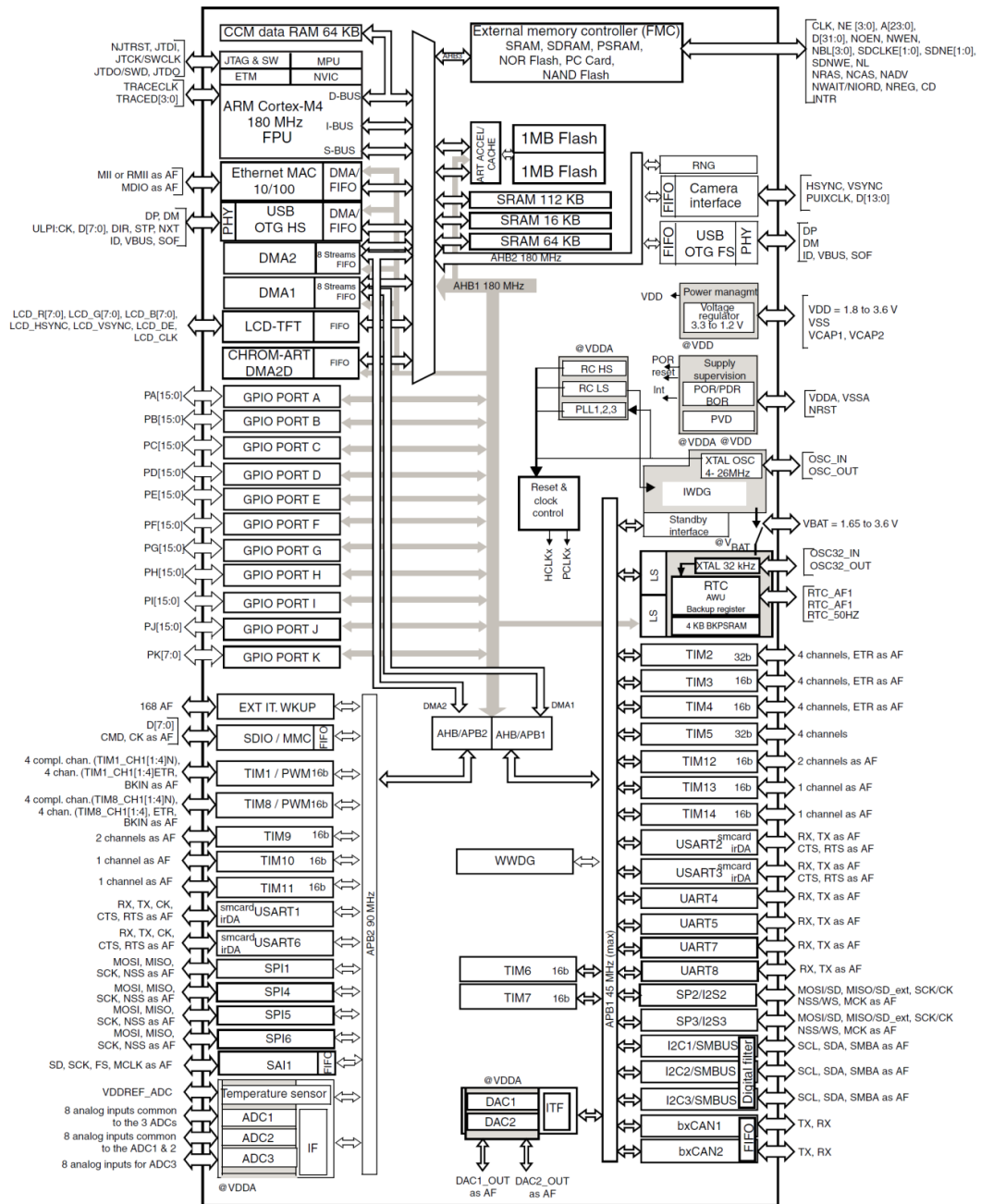
Těchto pět prvků je organizováno ve struktuře, jak je uvedeno na obrázku (Obrázek 4.1). Hradlové pole XC3S200A má 4 DCM bloky (2 nahoře a 2 dole), větší pole XC3S700(1400)A mají navíc ještě tyto bloky uprostřed (na obrázku čárkovaně).



Obrázek 4.1: Architektura rodiny Spartan-3A [4]

4.2 Mikroprocesor – STM32F427VIT6 [5]

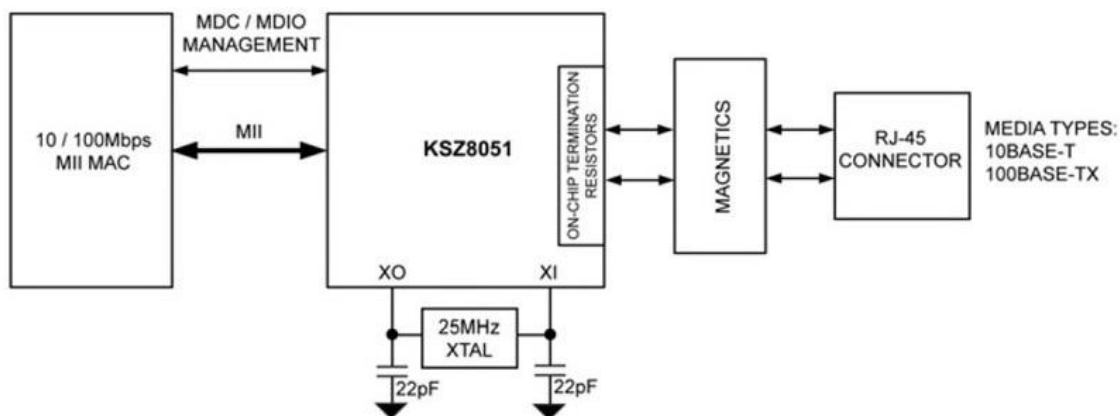
Tento mikroprocesor byl vybrán, protože byl dostupný ve škole a vyhovuje požadované aplikaci. Je to 32bitový RISC mikroprocesor s frekvencí 180 MHz postavený na jádře ARM Cortex-M4. Má v sobě implementovanou flash paměť o velikosti 2 MB, takže z této paměti bude moci být konfigurováno hradlové pole. Potom má také implementovanou ethernetovou MAC vrstvu s MII rozhraním a CRC výpočetní jednotku, což umožní snadné propojení s PHY čipem a přenos dat po ethernetu. Tento mikroprocesor bude použit v pouzdru LQFP100. Napájecí napětí může být v rozsahu 1,7 V až 3,6 V.



Obrázek 4.2: Blokové schéma mikroprocesorů STM32F427xx [5]

4.3 PHY čip – KSZ8051MLL [8]

PHY čip KSZ8051MLL podporuje standardy 10BASE-T a 100BASE-TX. Pro komunikaci slouží rozhraní MII. Nabízí možnost automatické konfigurace (Auto-negotiation) pro nastavení duplexu a nejvyšší rychlosti. Má také funkci HP Auto MDI/MDI-X, která slouží pro rozpoznání, zda je připojen přímým nebo kříženým kabelem. V případě potřeby jsou automaticky překříženy přijímající a odesílající diferenciální páry. Napájecí napětí pro tento čip je 3,3 V.



Obrázek 4.3: Blokové schéma pro zapojení PHY čipu do obvodu [8]

4.3.1 NAND Tree test

Pomocí NAND Tree testu je možné detekovat chyby mezi propojením PHY čipu a deskou. Z logických hradel NAND je vytvořen řetězec, kde jeden vstup do hradla je výstup z předcházejícího hradla NAND a druhý vstup je některý I/O pin PHY čipu. Každý testovaný pin PHY je zapojen do jednoho NAND hradla v řetězci. Výstup z posledního hradla v řetězci je vyveden na výstupní pin PHY čipu (na pin CRS) pro NAND Tree test.

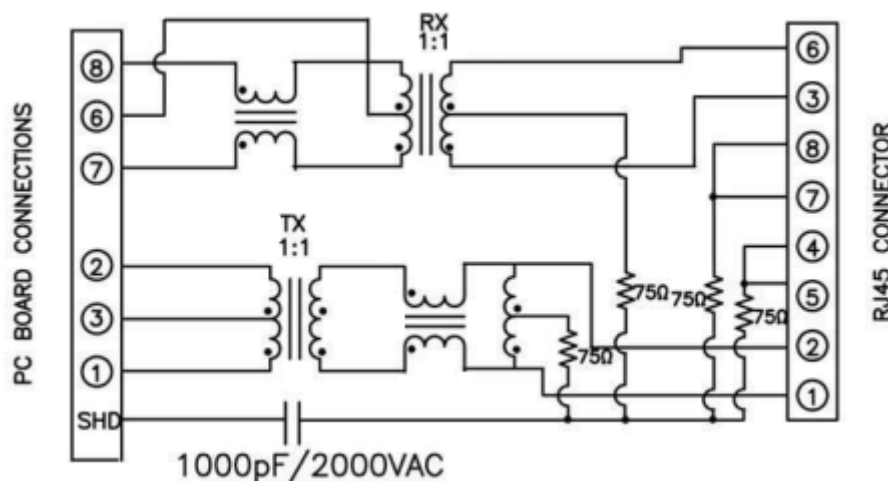
Díky tomuto zapojení do řetězce lze kontrolovat správné propojení PHY čipu s deskou. Po zapnutí NAND Tree módu musejí být všechny vstupní piny nastaveny na úroveň log. 1. Potom při postupném stahování úrovní na daných pinech do hodnoty log. 0, se musí měnit úroveň na výstupním pinu (CRS). Pořadí zapojení pinů do řetězce je uvedeno v tabulce (Tabulka 4.2).

Tabulka 4.2: Pořadí pinů pro NAND Tree test [8]

číslo pinu	název		číslo pinu	název	
18	MDIO	vstupní	31	TXC	vstupní
19	MDC	vstupní	32	TXEN	vstupní
20	RXD3	vstupní	33	TXD0	vstupní
21	RXD2	vstupní	34	TXD1	vstupní
22	RXD1	vstupní	35	TXD2	vstupní
23	RXD0	vstupní	36	TXD3	vstupní
27	RXDV	vstupní	37	LED0	vstupní
28	RXC	vstupní	38	LED1	vstupní
29	RXER	vstupní	39	COL	vstupní
30	INTRP	vstupní	40	CRS	výstupní

4.4 Konektor RJ45 – LMJTAB881243M-L [28]

Je to standardní zásuvka RJ45 s uspořádáním vývodů 8p8c pro připojení ethernetového kabelu. Konektor je stíněný, má dvě LED diody a obsahuje i oddělovací transformátory, takže ho lze přímo propojit s PHY čipem. Na následujícím obrázku (Obrázek 4.4) je schéma propojení vývodů uvnitř konektoru (mimo zapojení diod). Kompletní informace o konektoru lze nalézt v dokumentaci (literatura [28]).



Obrázek 4.4: Schéma propojení vývodů konektoru RJ45 (mimo LED) [28]

4.5 Spínané regulátory napětí – LM2595

Pro napájení součástek na desce bude zapotřebí napětí 1,2 V a 3,3 V, takže budou potřeba dva zdroje. Byly vybrány nastavitelné regulátory LM2595-ADJ v pouzdře TO220-5(NDH), pro jejich nastavení je potřeba minimální množství externích součástek. Nastavitelné regulátory byly vybrány, jelikož regulátory s fixním napětím 3,3 V nebyly dostupné za rozumnou cenu. Napětí 1,2V se dosáhne vynecháním odporového děliče ve zpětné vazbě.

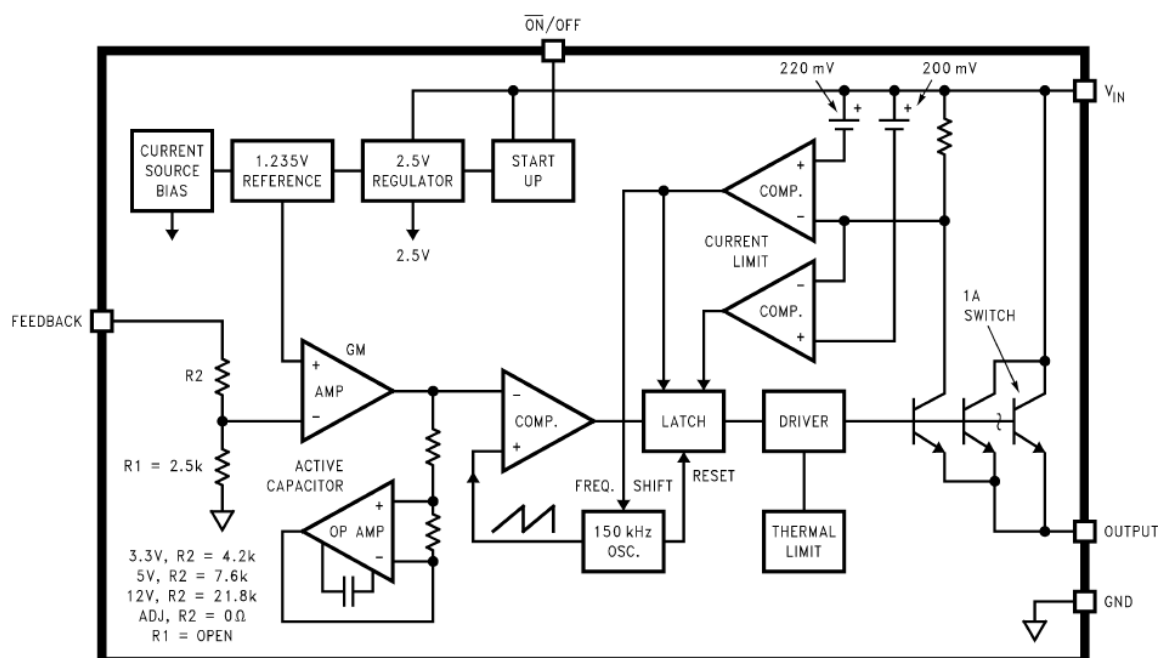
Jsou to spínané regulátory s frekvencí 150 kHz a s maximálním výstupním proudem 1 A. Jsou nastavitelné v rozmezí 1,2 V – 37 V s přesností $\pm 4\%$. Mají uváděnou účinnost 78 %. Maximální vstupní napětí je 45 V. [8]



Obrázek 4.5: Regulátor LM2595 v pouzdru TO220-5(NDH) [GME Electronic]

Maximální proudový odběr u mikroprocesoru a PHY čipu se dá snadno dohledat v dokumentaci. Problém nastane při určení proudového odběru hradlového pole, kde je v dokumentaci uveden pouze maximální odběr v klidovém stavu, avšak celkový odběr závisí až na návrhu implementovaném uvnitř hradlového pole. V tabulce (Tabulka 4.3) jsou uvedeny proudové odběry součástek. Pro odhad proudového odběru hradlového pole nabízí výrobce nástroj - Xilinx Power Estimator (XPE), který podle počtu využitých částí FPGA vypočte spotřebu čipu. Při nastavení nejhoršího případu (100% zatížení pole a maximální frekvence) vyšla spotřeba přes 3 A. Ve výsledném návrhu pro požadovanou aplikaci tohoto stavu nebude nikdy dosaženo a tak ještě s uvažáním předchozích zkušeností byl zvolen zdroj schopný dodat právě 1 A, jak pro napětí 1,2 V, tak pro napětí 3,3 V.

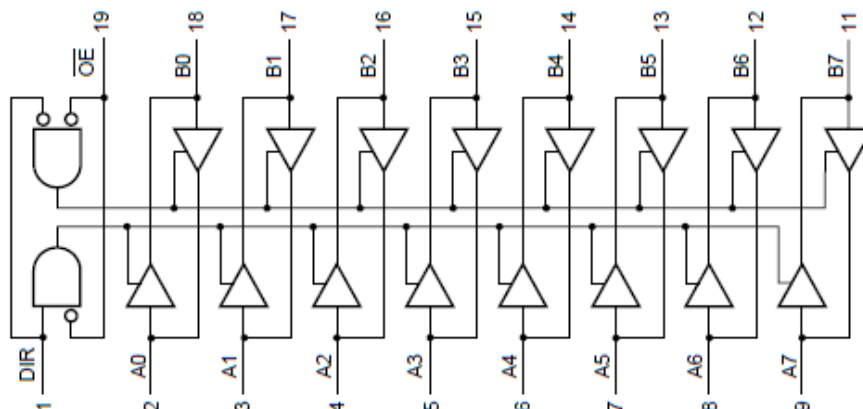
	maximální proud [mA]	klidový proud [mA]	
STM32F427	270	-	
KZS8051MLL	49		
Spartan-3A	-	ICCINTQ	50
		ICCOQ	2
		ICCAUXQ	12



Obrázek 4.6: Blokové schéma LM2595 [9]

4.6 Budiče 74AHC245 [10]

Obvody typu 74xx245 jsou osmikanálové obousměrné třístavové budiče, kde znaky xx představují konkrétní technologii. Byly vybrány dostupné obvody 74AHC245, ale může být použita i jiná technologie, která bude dostatečně rychlá a kompatibilní s 3,3 V logikou.



Obrázek 4.7: Schéma zapojení obvodu 74AHC245 [10]

Časové parametry pro tyto budiče jsou uvedeny v tabulce (Tabulka 4.4), kde jsou vypsány typické (typ.) a maximální (max.) časové údaje při napájení obvodu napětím 3,3 V. Hodnoty jsou uvedeny pro dvě různé parazitní kapacity při připojení zátěže.

Tabulka 4.4: Časové parametry pro budiče 74AHC245 [10]

		15 pF		50 pF	
doba náběhu a sestupu	t _r , t _h [ns]	≤ 3			
Zpoždění při přenosu	t _{pd} [ns]	5	8,4	6,5	11,9
čas pro povolení budiče	t _{en} [ns]	6,5	13,2	9	16,7
čas pro zakázání budiče	t _{dis} [ns]	7,5	12,5	10	15,8
		typ.	max.	typ.	max.

4.7 Programovací kabel JTAG HS2

Tento kabel je plně kompatibilní s nástroji a výrobky od firmy Xilinx. K počítači se připojuje přes USB port a na desku přes konektor JTAG. Pomocí tohoto kabelu je možné naprogramovat hradlové pole XC3S200A. Na desku bude připojen pomocí 6pinového konektoru. Uspořádání konektoru je vidět na obrázku (Obrázek 4.8).

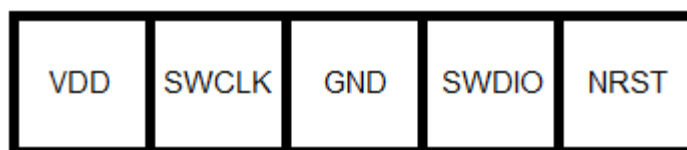


Obrázek 4.8: Programovací kabel JTAG HS-2 [7]

4.8 Black Magic Probe

Black Magic Probe je nástroj sloužící pro programování a odlaďování (umožňuje krokování) mikroprocesorů založených na jádře ARM Cortex-M4. [27] Bude použit pro programování a odlaďování kódu v mikroprocesoru STM32F427. Připojuje se pomocí JTAG, nebo SWD (Serial Wire debugging). Při návrhu desky bude využito rozhraní SWD, které vyžaduje 5pinový konektor. Uspořádání signálů u konektoru je uvedeno na obrázku (Obrázek 4.9).

- VDD – napájecí napětí 3,3 V
- SWCLK = JTCK – hodinový signál
- GND – zem
- SWDIO = JTMS – vodič pro přenos dat
- NRST – reset mikroprocesoru



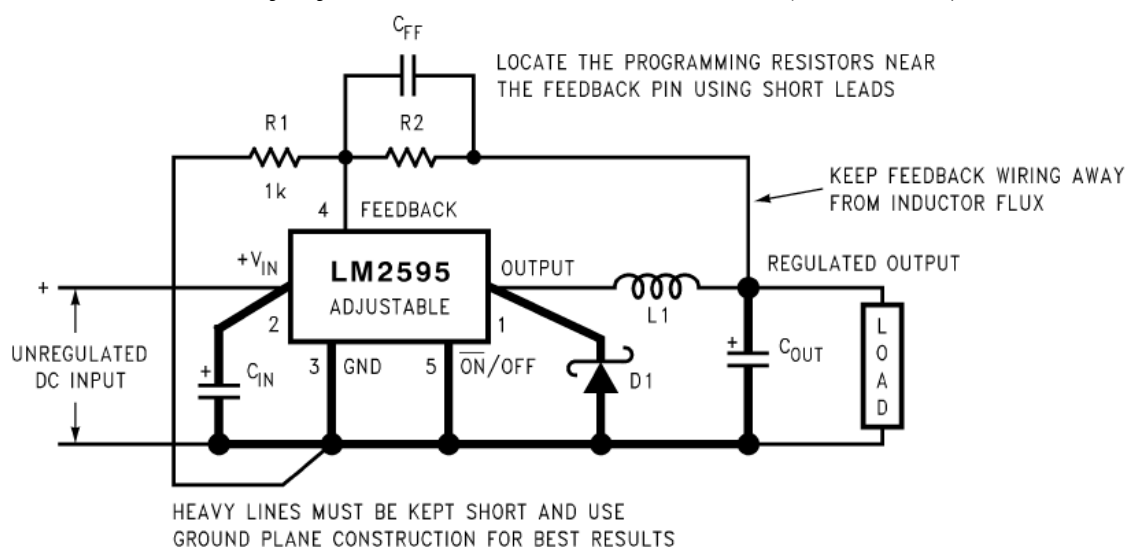
Obrázek 4.9: Uspořádání konektoru SWD

5 NÁVRH DESKY PLOŠNÝCH SPOJŮ

V této kapitole už se dostáváme k samotnému návrhu desky plošných spojů pro požadovanou aplikaci. Deska je navrhována jako čtyřvrstvá, kde jedna vnitřní vrstva je „zem“ a ve druhé vnitřní vrstvě vede rozvod napájení k jednotlivým součástkám, ty jsou rozmístěny na vrchní i na spodní vnější vrstvě. Deska je navrhována v programu Eagle. Celkové schéma pro navrženou desku je umístěno v příloze.

5.1 Napájení

Deska je napájena zdrojem s napětím 12 V, jako ochrana proti přepólování je na desce sériově zapojena dioda. Úbytek napětí na diodě je asi 0,9 V, nicméně ve schématu je vedení za diodou značeno jako +12 V. Pro napájení součástek umístěných na desce jsou potřeba dvě různá napětí, a to 1,2 V a 3,3 V. Pro získání těchto napětí jsou použity spínané nastavitelné regulátory napětí LM2595 v pouzdrech TO220-5(NDH). Při uvážení maximálního odběru proudu z regulátorů a s jejich účinností nad 70 %, postačí pro napájení desky zdroj 12 V s proudem 0,6 A. Základní parametry spínaných regulátorů jsou uvedeny v kapitole (4.5). Pro jejich nastavení je potřeba připojit několik externích součástek tak, jak je uvedeno ve schématu na obrázku (Obrázek 5.1).



Obrázek 5.1: Schéma zapojení regulátoru LM2595 [9]

5.1.1 Postup pro výběr součástek [9]

Postup pro výběr součástek se dá rozdělit do šesti kroků:

- 1) **Výběr rezistorů R1 a R2** pro nastavení zpětné vazby a nastavení výstupního napětí (V_{OUT}). Nastavení se provádí podle následujících rovnic (5.1.1, 5.1.2), hodnota odporu R1 musí být v intervalu od 240 Ω do 1,5 k Ω .

V_{REF} ... referenční napětí

$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right), \quad \text{kde } V_{REF} = 1,23 \text{ V} \quad (5.1.1)$$

$$R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right) \quad (5.1.2)$$

- 2) **Výběr indukčnosti L1** podle grafu na obrázku (Obrázek 5.2), kde součin $E \cdot T$ zjistíme z rovnice (5.1.3).

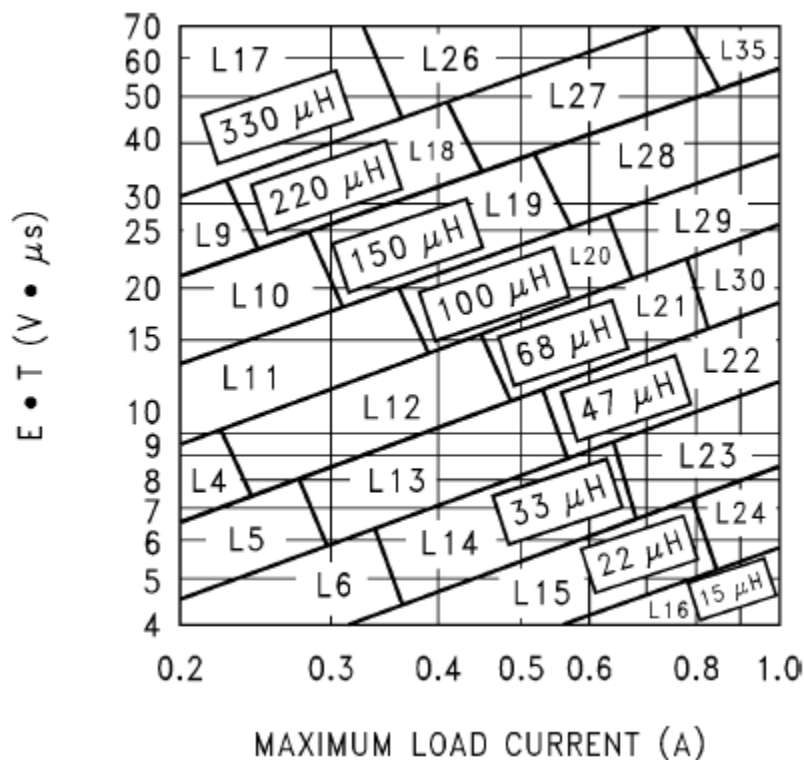
V_{IN} ... Maximální vstupní napětí

V_{SAT} ... Saturační napětí

V_D ... Napětí na diodě v propustném směru

$$E \cdot T = (V_{IN} - V_{OUT} - V_{SAT}) \cdot \frac{V_{OUT} + V_D}{V_{IN} - V_{SAT} + V_D} \cdot \frac{1000}{150 \text{ kHz}} [V \cdot \mu s], \quad (5.1.3)$$

kde $V_{SAT} = 1 \text{ V}$, $V_D = 0,5 \text{ V}$



Obrázek 5.2: Graf pro výběr indukčnosti pro LM2595-ADJ [9]

- 3) **Výběr výstupního kondenzátoru C_{OUT}** , jsou doporučeny elektrolytické a tantalové kondenzátory s nízkým sériovým odporem (low ESR) v rozmezí od $47 \mu\text{F}$ do $330 \mu\text{F}$. Musí být konstruovány na napětí minimálně 1,5 krát větší, než je výstupní napětí z regulátoru.
- 4) **Výběr kompenzačního kondenzátoru C_{FF}** , ten je potřeba pouze, pokud bude výstupní napětí větší než 10 V , takže zde se jeho výběrem nemusíme zabývat.
- 5) **Výběr Schottkyho diody D_1** , proud diodou musí být alespoň 1,3 krát větší než maximální proud do zátěže a napětí v závěrném směru alespoň 1,25 krát větší než na výstupu regulátoru.
- 6) **Výběr vstupního kondenzátoru C_{IN}** , opět tantalový nebo elektrolytický kondenzátor s nízkým sériovým odporem a napětím 1,5 krát větším než je vstupní napětí do regulátoru.

5.1.2 Vybrané součástky

V kapitole výše byl teoreticky popsán základní postup při výběru součástek a nyní můžeme tedy přistoupit ke konkrétnímu výběru pro požadovanou aplikaci. Při osazování desky byly využity součástky dostupné ve škole, a proto jsou v návrhu uvedeny jak původně navržené součástky tak výsledné, se kterými byla deska osazena.

5.1.2.1 Součástky pro regulátor s výstupním napětím 3,3V

- 1) Odpor s tolerancí $\pm 1 \%$

Navrženo:

$$R1 = 1,2 \text{ k}\Omega$$

$$R2 = 2 \text{ k}\Omega$$

$$V_{OUT} = 1,23 \left(1 + \frac{2}{1,2} \right) = 3,28 \text{ V} \cong 3,3 \text{ V}$$

Osazeno:

$$R1 = 1,18 \text{ k}\Omega$$

$$R2 = 3,9 // 3,9 \text{ k}\Omega = 1,95 \text{ k}\Omega$$

$$V_{OUT} = 1,23 \left(1 + \frac{1,95}{1,18} \right) = 3,26 \text{ V} \cong 3,3 \text{ V}$$

- 2) Cívka typu MLC12xx

Navrženo:

$$L1 = 68 \text{ }\mu\text{H}$$

$$\text{s ohledem na výpočet } E \cdot T \cong 16 [V \cdot \mu s]$$

Osazeno:

$$L1 = 56 \text{ }\mu\text{H}$$

- 3) Výstupní kondenzátor C_{OUT} tantalový

Navrženo:

$$C_{OUT} = 100 \text{ }\mu\text{F}/6\text{V (pouzdro D)}$$

Osazeno:

$$C_{OUT} = 330 \text{ }\mu\text{F}/6\text{V (pouzdro E)}$$

- 4) Kompenzační kondenzátor C_{FF} není potřeba

- 5) Shottkyho dioda SK26 (60 V, 2 A) v pouzdru SMB

- 6) Vstupní kondenzátor C_{IN} tantalový v pouzdru D

Navrženo:

$$C_{IN} = 100 \text{ }\mu\text{F}/25\text{V}$$

Osazeno:

$$C_{IN} = 100 \text{ }\mu\text{F}/16\text{V}$$

5.1.2.2 Součástky pro regulátor s výstupním napětím 1,2V

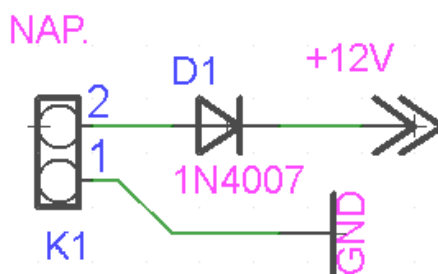
Pro druhý regulátor jsou použity ty stejné součástky s tím rozdílem, že jsou vynechány odpory, respektive $R1 = \infty \Omega$, $R2 = 0 \Omega$, potom je výstupní napětí $V_{OUT} = 1,23 \text{ V}$. Toto napětí bude použito pro napájení jádra hradlového pole, i s uvažováním udávaných tolerancí napětí, nebude překročena hodnota, při které by došlo ke zničení pole. Druhá odlišnost byla v původně navržené cívce o hodnotě $47 \mu\text{H}$, a však osazena je taktéž cívka o hodnotě $56 \mu\text{H}$ jako u prvního regulátoru.

5.1.3 Rozvod napájení

Jak již bylo řečeno, deska je čtyřvrstvá, kde jedna vnitřní vrstva je zem a druhá rozvod napájení k součástkám. Většina zařízení pracuje na napětí $3,3 \text{ V}$, pouze pro napájení jádra hradlového pole je nutné přivedení napětí $1,2 \text{ V}$. Nižší úroveň napětí vede ve vnitřní vrstvě a pod hradlovým polem je vytvořena plocha, tak aby se dalo do hradlového pole snadno přivést napětí jak $3,3 \text{ V}$, tak $1,2 \text{ V}$.

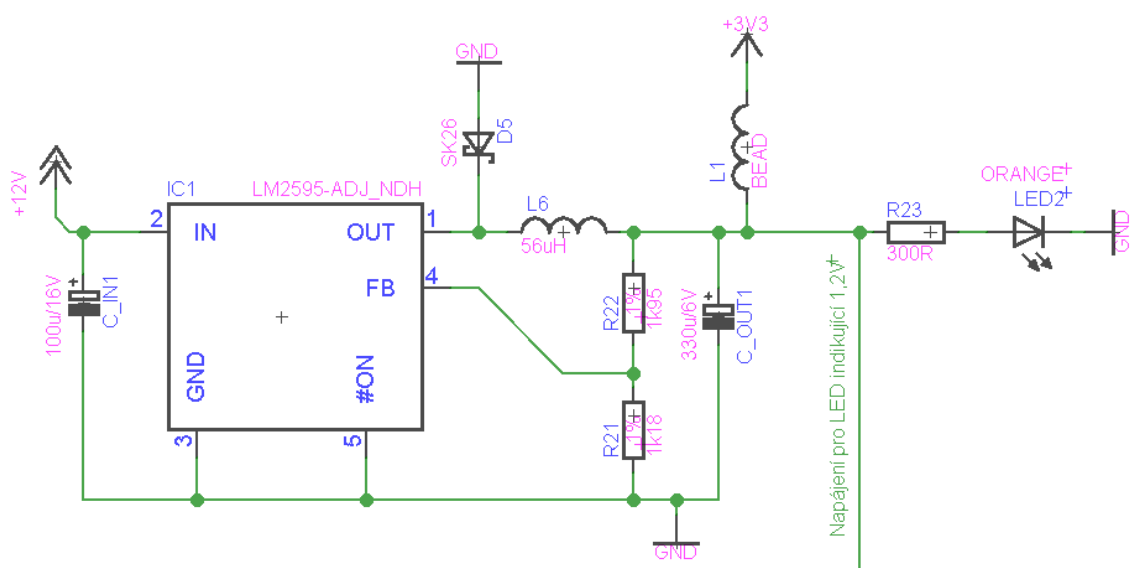
Jednotlivé výstupy z obou regulátorů jsou od zbytku součástek na desce odděleny feritovým korálkem (indukčnost BEAD). Toto oddělení je též výhodné při ožiování desky. Ještě před tímto korálkem jsou připojeny dvě LED diody pro indikaci napájení desky (funkčnosti spínaných regulátorů). Jedna dioda je připojena přímo na $3,3 \text{ V}$ s předřadným odporem a druhá dioda je připojena k tranzistoru, který je spínán výstupem regulátoru s napětím $1,2 \text{ V}$. Indikační diody jsou umístěny mezi oběma regulátory, kde vrchní regulátor (blíže ke konektoru napájení) poskytuje napětí $3,3 \text{ V}$ a spodní $1,2 \text{ V}$. Pro ochranu vedení k indikačním diodám, proti působení rušivého napětí kapacitního typu, je toto vedení od spodního regulátoru odděleno stíněním (vodivá cesta připojená na zem).

Na desce je umístěn konektor pro připojení 12 V zdroje, za tímto konektorem následuje sériově zapojená dioda (1N4007) sloužící jako ochrana desky proti přepólování. Schéma zapojení ochrany je na obrázku (Obrázek 5.3). Přestože je na diodě úbytek napětí, tak je ve schématu vedení za diodou značeno jako $+12 \text{ V}$. Při návrhu regulátorů byl však tento úbytek brán v potaz.



Obrázek 5.3: Ochrana proti přepólování

Na obrázku (Obrázek 5.4) je schéma zapojení pro regulátor poskytující $3,3 \text{ V}$ s připojenou indikační LED diodou.



Obrázek 5.4: Schéma pro regulátor 3,3V

5.2 Mikroprocesor

Na desce je umístěn mikroprocesor STM32F427VIT6, který slouží jako hlavní řídicí prvek celé desky. Aby mohl odesílat data po ethernetu, je propojen s PHY čipem pomocí MII rozhraní. Mikroprocesor může také konfigurovat hradlové pole přes rozhraní JTAG ze své vnitřní flash paměti.

Poskytuje možnosti interakce s uživatelem pomocí dvou tlačítek a tří LED diod umístěných přímo na desce. Dále má na pinové lišty vyvedeny některé svoje další vývody. Podrobnější popis vývodů je uveden v kapitole (5.5.4 mikroprocesor ↔ okolí).

Programování mikroprocesoru se provádí pomocí rozhraní SWD (Serial Wire debugging). Schéma připojení tohoto rozhraní z mikroprocesoru ke konektoru je na obrázku (Obrázek 5.5). Seznam propojených pinů mikroprocesoru s konektorem je uveden v tabulce (Tabulka 5.1). Rozhraní SWD využívá část rozhraní JTAG, proto jsou i k nepoužitým signálům z JTAG připojeny pull-up a pull-down rezistory.

Tabulka 5.1: Připojení SWD konektoru k mikroprocesoru

vodič	mikroprocesor	
	označení pinu	číslo pinu
JTCK (SWCLK)	PA14	76
JTMS (SWDIO)	PA13	72
NRST	NRST	14

Mikroprocesor má 3 možnosti bootování (zavádění programu), mezi kterými se dá volit pomocí dvou signálů, a to BOOT0 a BOOT1. Je pevně zvolena možnost bootování z flash paměti připojením signálu BOOT0 na zem (přes rezistor).

Jako zdroj externího kmitočtu je k mikroprocesoru připojen krystal o frekvenci 25 MHz.

5.3 Hradlové pole

Hradlové pole XC3S200A-4VQG100I je hlavní část sloužící pro zpracování signálů. Pro zajištění jeho napájení a správné funkce všech obvodů se používají čtyři různé typy napájecích vstupů:

- V_{CCINT} – pro napájení jádra = 1,2 V
- V_{CCAUX} – pomocné napájení; díky výběru pole Spartan-3A může být 3,3 V
- V_{CCO} – napájení výstupních řadičů; pro každou banku může být jiné; podle vstupního napětí přivedeného na tyto piny se rozhoduje, jaký standard bude použit; v tomto návrhu budou využívány pouze standardy LVTTTL nebo LVC MOS33, takže do všech V_{CCO} vstupů bude přivedeno napětí 3,3 V
- V_{REF} – referenční napětí pro diferenciální signálové standardy; tyto standardy nebudou využívány, takže na tyto piny nebude přiváděno žádné napětí

Pro připojení oddělovacích kondenzátorů není uvedeno od výrobce žádné schéma, pouze udává doporučení na svých stránkách¹. Malé kondenzátory (pro vysoké frekvence) musí být použity u všech V_{CC} pinů ve vzdálenosti maximálně 1-2 cm, větší kondenzátory (pro střední frekvence) jsou doporučené tantalové o hodnotách od 4,7 μ F do 47 μ F ve vzdálenosti do 8 cm od V_{CC} . Největší kondenzátory (pro nízké frekvence) v rozmezí od 47 μ F do 4700 μ F mohou být kdekoli na desce, tyto kondenzátory jsou již použity při návrhu spínaných regulátorů. Ke každému pinu V_{CC} je připojen kondenzátor o velikosti 100 nF a ke každé ze čtyř bank navíc kondenzátor o velikosti 4,7 μ F.

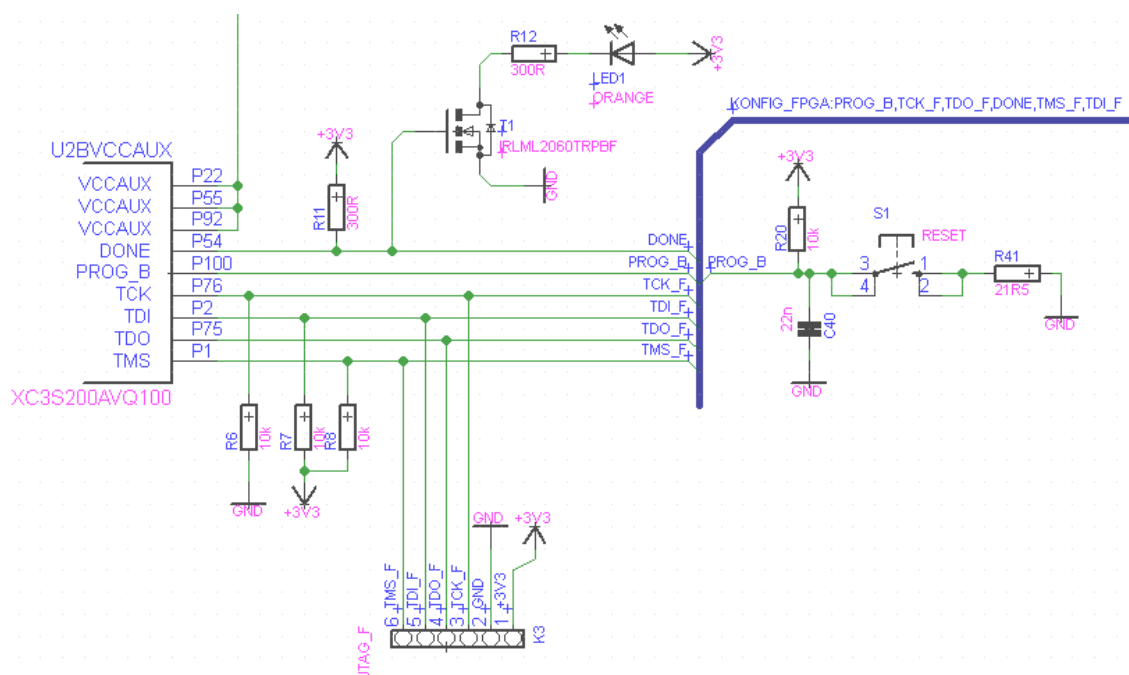
Pro nastavení konfiguračního módu hradlového pole se používají piny M0, M1 a M2, pomocí nichž se dají nastavit různé způsoby konfigurace. V návrhu je využita konfigurace pomocí rozhraní JTAG, a proto jsou piny M[0:2] připojeny přes rezistory na zem, nebo napájení následujícím způsobem: M0=1, M1=0, M2=1. Ke konfiguraci hradlového pole dochází po jeho resetování, reset se dá vyvolat signálem log. 0 na pinu PROG_B. Rozhraní JTAG poskytuje instrukci, pomocí které se dá zahájit konfigurace hradlového pole (proto nemusí obsahovat signál PROG_B). Nakonfigurování hradlového pole je indikováno signálem DONE.

¹ http://www.xilinx.com/products/design_resources/signal_integrity/si_pcbcheck.htm

Na následujícím obrázku (Obrázek 5.7) je uvedeno schéma připojení konektoru JTAG k hradlovému poli a zapojení signálů PROG_B a DONE. Seznam propojení pinů hradlového pole s konektorem je uveden v tabulce (Tabulka 5.2). Konfigurační signály pro FPGA jsou ve schématu sloučeny do sběrnice s názvem „KONFIG_FPGA“ a přivedeny na vývody mikroprocesoru, aby bylo možné pomocí něj také hradlové pole konfigurovat. Vodiče rozhraní JTAG pro FPGA mají v názvu příponu „_F“ z důvodu jejich rozlišení.

Tabulka 5.2: Připojení JTAG konektoru k hradlovému poli

	hradlové pole	
vodič	název pinu	označení pinu
TMS_F	TMS	P1
TDI_F	TDI	P2
TDO_F	TDO	P75
TCK_F	TCK	P76

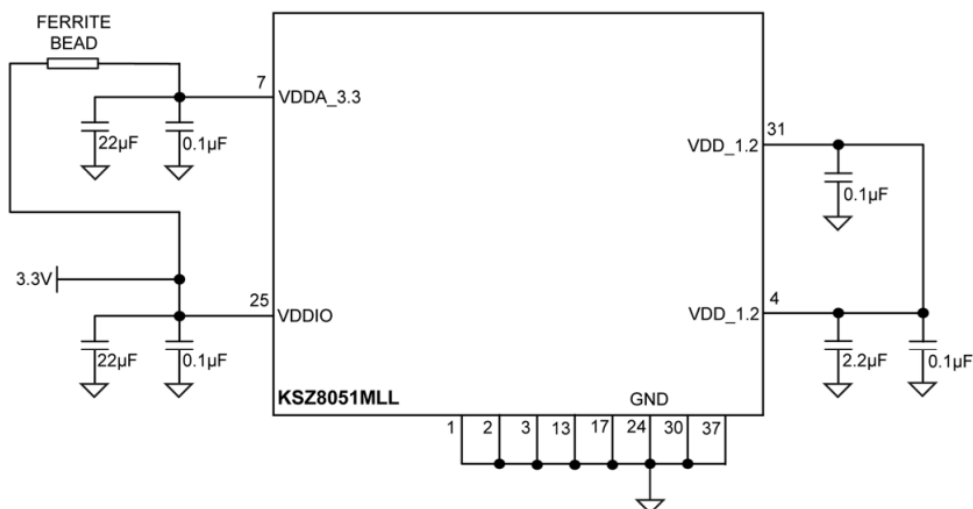


Obrázek 5.7: Schéma zapojení konfiguračních signálů pro FPGA

5.4 PHY čip a konektor RJ45

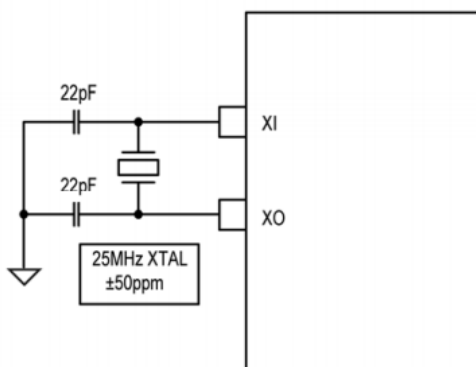
PHY čip KSZ8051MLL je na jedné straně propojen s mikroprocesorem pomocí rozhraní MII a na druhé s ethernetovým konektorem RJ45. Jeho napájení je zajištěno napětím 3,3 V a čip v sobě má zabudovaný regulátor na 1,2 V pro napájení svého jádra.

Schéma zapojení napájení a oddělovacích kondenzátorů je uvedeno na následujícím obrázku (Obrázek 5.8).



Obrázek 5.8: Schéma napájení PHY (KSZ8051MLL) [8]

Aby mohl PHY čip poskytovat referenční hodinový signál pro řízení přenosu dat, je k němu připojen krystal o frekvenci 25 MHz.



Obrázek 5.9: Připojení krystalu k PHY [8]

Pro nakonfigurování PHY čipu po restartu nejsou na desce připojeny žádné pull-up, ani pull-down rezistory. Pro jeho nastavení je možné využít tři různé způsoby, a to:

- možnost automatické konfigurace (Auto-negotiation)
- piny PHY čipu jsou propojeny s mikroprocesorem, takže je možné nastavit požadované hodnoty na konfiguračních pinech a restartovat PHY čip, čímž se nastaví konfigurační registr (princip konfigurace pomocí pull-up a pull-down rezistorů při restartu čipu)
- pomocí SMI (Serial Management Interface), což je konfigurační rozhraní PHY čipu obsahující piny MDIO a MDC, které jsou také připojeny k mikroprocesoru, lze tedy nastavit konfigurační registry v PHY

Deska se k PC připojuje kabelem se dvěma páry kroucených dvojlinek a vybraný ethernetový konektor RJ45 má v sobě již zabudované oddělovací transformátory, takže PHY čip může být přímo propojen s tímto konektorem.

Pro ochranu desky proti přepětí, které by mohlo přijít po připojení ethernetového kabelu, jsou ke konektoru RJ45 mezi diferenciálními páry zapojeny ochranné transily, konkrétně nízkonapěťová jednotka SLVU2.8-4A1, která je navržena pro ochranu ethernetových linek. [30] Při přepětí na ethernetové lince je diferenciální pár této linky zkratován a tím je ochráněna připojená deska.

5.5 Propojení mezi součástkami a okolím

V této kapitole bude popsáno, jakým způsobem jsou propojeny součástky mezi sebou, případně se svým okolím pro možnost interakce s uživatelem.

5.5.1 Tlačítka a diody

Tato kapitola souží jako přehled připojení prvků (tlačítek a LED diod) pro rychlou a snadnou interakci s uživatelem ke konkrétním pinům periférií. Zároveň je zde uveden i jejich způsob zapojení na desce. V návrhu desky je umístěno 5 tlačítek a 9 LED diod, kde 2 diody slouží pro signalizaci stavu napájení, 1 pro signalizaci nakonfigurování hradlového pole a zbylých 6 je možno ovládat pomocí hradlového pole, nebo mikroprocesoru (3 hradlovým polem a 3 mikroprocesorem). Jedno tlačítko je připojeno na signál „PROG_B“ a slouží jako resetování hradlového pole (zároveň je připojeno i k mikroprocesoru přes odpor 430 Ω). Zbylá 4 tlačítka jsou připojena k hradlovému poli (2) a k mikroprocesoru (2). Na které piny konkrétního obvodu jsou jaké prvky připojeny, je uvedeno v následující tabulce (Tabulka 5.3).

Tabulka 5.3: Připojení tlačítek a LED diod

název ve schématu		označení pinů obvodů	
		mikroprocesor	hradlové pole
Diody	LED4	PD8	
	LED5	PD9	
	LED6	PD10	
	LED7		P83
	LED8		P84
	LED9		P85
Tlačítka	TP1	PD12	
	TP2	PD11	
	TH1		P93
	TH2		P94
	RESET	PE0 (přes 430 Ω)	P100 (PROG_B)

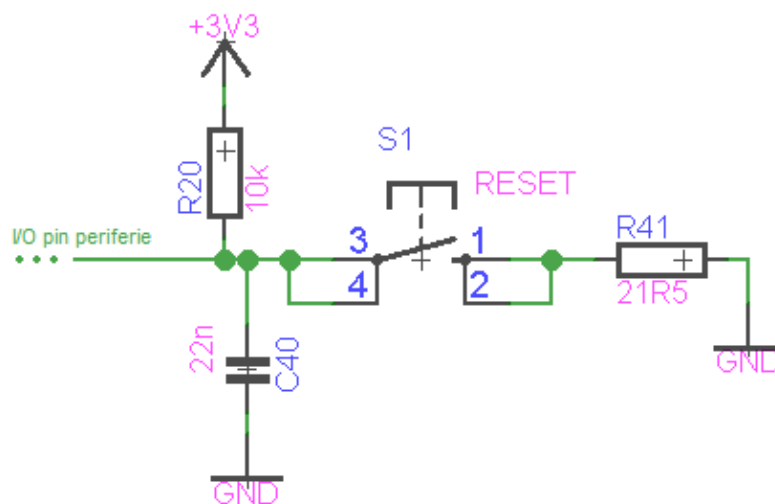
Zapojení diod:

K diodám byly navrhovány předřadné odpory s uvažáním napětí na diodě 2 V a proudem diodou 5 mA. Podle rovnice (5.6.1) vyšla hodnota 260 Ω, nejbližší vyšší dostupná hodnota ve škole byla 300 Ω. Pro předřadné odpory jsou tedy k diodám použity rezistory s hodnotou 300 Ω a to i k diodám v ethernetovém konektoru RJ45.

$$R = \left(\frac{U - U_D}{I_D} \right) = \left(\frac{3,3 - 2}{0,005} \right) = 260 \, \Omega \quad (5.6.1)$$

Zapojení tlačítek:

Tlačítka jsou jednou stranou připojena k I/O pinu periferie a na druhé straně jsou přes odpor o malé hodnotě připojena na zem. Ze strany od periferie je připojen pull-up rezistor a kondenzátor, který omezuje záskoky při stisknutí tlačítka. Konkrétní zapojení je vidět na schématu níže (Obrázek 5.10). V klidovém stavu (tlačítko rozepnuto) je na I/O pinu periferie hodnota log. 1, při stisku tlačítka se vybije kondenzátor přes odpor za tlačítkem a na vstup periferie se dostává hodnota log. 0.



Obrázek 5.10: Schéma připojení tlačítek

5.5.2 PHY čip ↔ mikroprocesor

Mikroprocesor je s PHY čipem propojen pomocí MII rozhraní. Vodiče ve skupině signálů pro vysílání dat, to jsou signály TXD[0..3], TXEN a TXC, jsou na desce nataženy o stejné délce (liší se asi o 1 mm) z důvodu zajištění časových parametrů. Rovněž vodiče ve skupině signálů pro příjem dat RXD[0..3], RXDV a RXC jsou nataženy o stejné délce (rozdíl necelý 1 mm). Při nízkých rychlostech, jako je 10 Mb/s a 100 Mb/s, a s tím, že dráhy signálů jsou krátké, není toto opatření zcela nutné. Při

delších trasách vedení, vyšších rychlostech nebo redukováném rozhraní RMII už by toto opatření nutné bylo.

Všechny signály vyvedené z PHY čipu jsou připojeny k mikroprocesoru, kromě signálů vedoucích do konektoru RJ45, takže bude možné v případě potřeby PHY konfigurovat pomocí mikroprocesoru. Z možných konfiguračních pinů PHY čipu nejsou do mikroprocesoru připojeny pouze piny LED0/NWAYEN a LED1/SPEED. Defaultně je tedy nastavena automatická konfigurace (Auto-negotiation) a dá se vypnout pouze pomocí MDC/MDIO managementu. Stejně tak je defaultně nastavena rychlost na 100 Mb/s (nemá vliv při automatické konfiguraci).

Tím, že je povolena automatická konfigurace, dojde po propojení kabelem s druhým zařízením k automatickému nakonfigurování PHY čipu podle potřeby tohoto zařízení. Přenosové módy jsou nastavovány podle priority sestupně v pořadí 100Base-TX/full-duplex, 100Base-TX/half-duplex, 10Base-T/full-duplex, 10Base-T/half-duplex. [8] Při připojení k druhému zařízení nezáleží na tom, jestli je propojení provedeno přímým nebo kříženým kabelem díky funkci HP Auto MDI/MDI-X.

Pro připojení ethernetového MII rozhraní k mikroprocesoru se 100pinovým pouzdrem není možné alternativní mapování periferie na jiné piny (přemapovat se dá pouze signál TXD3). Propojení pinů mezi PHY čipem a mikroprocesorem je uvedeno v tabulce (Tabulka 5.4).

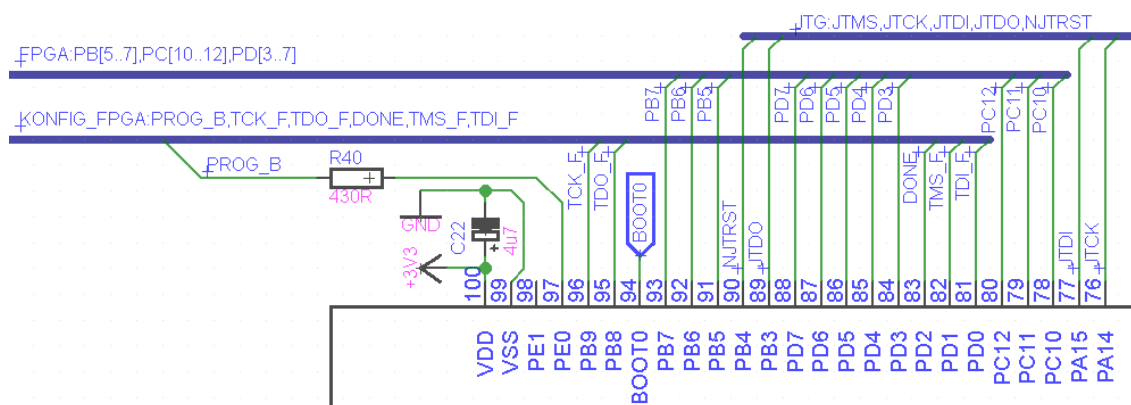
Tabulka 5.4: Propojení PHY čipu s mikroprocesorem

piny PHY čipu	piny mikroprocesoru
MDC	PC1
MDIO	PA2
TXEN	PB11
TXC	PC3
TXD0	PB12
TXD1	PB13
TXD2	PC2
TXD3	PE2
RXC/BCASTOFF	PA1
RXD0/DUPLEX	PC4
RXD1/PHYAD2	PC5
RXD2/PHYAD1	PB0
RXD3/PHYAD0	PB1
RXDV/CONFIG2	PA7
RXER/ISO	PB10
CRS/CONFIG1	PA0
COL/CONFIG0	PA3
INTRPT/#NAND_TREE	PE4
#RST	PE3

5.5.3 mikroprocesor ↔ FPGA

Mikroprocesor je s hradlovým polem propojen dvěma skupinami signálů, kde v jedné jsou vedeny signály pro konfiguraci hradlového pole a druhá skupina vodičů slouží pro přenos dat mezi mikroprocesorem a hradlovým polem. Na schématu (Obrázek 5.11) je vidět jedna strana mikroprocesoru a signály z ní vyvedené. Dvě sběrnicе odcházející doleva vedou k hradlovému poli, sběrnicе vedoucí doprava obsahuje signály rozhraní JTAG pro programování mikroprocesoru.

Sběrnicе označená „CONFIG_FPGA“ obsahuje signály pro konfiguraci hradlového pole, její zapojení na druhé straně bylo možno vidět na obrázku (Obrázek 5.7). Sběrnicе označená „FPGA“ obsahuje 11 vodičů pro přenos dat mezi mikroprocesorem a hradlovým polem, v hradlovém poli jsou všechny signály z této sběrnicе připojeny do banky 3. V této skupině vodičů jsou zahrnuty piny mikroprocesoru, na které je možné mapovat několik alternativních funkcí pro sériový přenos dat, jako jsou jednotky USART, UART, SPI, I2C, takže při přenosu dat z hradlového pole do mikroprocesoru je na výběr z více možností. Konkrétní označení pinů a vodičů propojujících mezi sebou hradlové pole a mikroprocesor je uvedeno v tabulkách (Tabulka 5.5, Tabulka 5.6).



Obrázek 5.11: Signály vyvedené z jedné strany mikroprocesoru

Tabulka 5.5: Propojení skupiny vodičů „CONFIG_FPGA“ mezi piny mikroprocesoru a hradlového pole

	Mikroprocesor	FPGA
TDO_F	PB8	P75
TCK_F	PB9	P76
TDI_F	PD0	P2
TMS_F	PD1	P1
DONE	PD2 (jenom vstup)	P54
PROG_B	PE0 (přes 430Ω)	P100

Tabulka 5.6: Propojení skupiny vodičů „FPGA“ mezi piny mikroprocesoru a hradlového pole

mikroprocesor	FPGA	alternativní funkce mikroprocesoru
PB7	P4	TIM, I2C, UART
PB6	P5	
PB5	P6	
PD7	P9	USART
PD6	P10	
PD5	P12	
PD4	P13	
PD3	P15	
PC12	P16	SPI, I2S, USART, UART, I2S
PC11	P19	
PC10	P20	

5.5.3.1 „KONFIG_FPGA“

Tato skupina vodičů obsahuje rozhraní JTAG (4 vodiče) pro konfiguraci FPGA z mikroprocesoru. Vodiče sběrnice JTAG jsou na desce plošných spojů nataženy přibližně o stejné délce (rozdíl asi 4 mm), aby se zamezilo problémům při konfiguraci. Dále obsahuje signál „DONE“, který je řízen hradlovým polem a signalizuje, že je hradlové pole nakonfigurováno. Na straně mikroprocesoru může být tedy tento pin nastaven pouze jako vstupní.

Poslední signál v této skupině je „PROG_B“, který je do mikroprocesoru připojen přes odpor kvůli omezení proudu. Pomocí tohoto signálu je možné resetovat hradlové pole, pokud je stažen do úrovně log. 0. Odpor pro omezení proudu je zde zařazen pro ochranu mikroprocesoru, protože „PROG_B“ je zároveň připojen k tlačítku a kondenzátoru, jak je možno vidět na obrázku (Obrázek 5.7). Při stažení signálu mikroprocesorem do úrovně log. 0 tak rezistor omezuje procházející proud z vybíjejího se kondenzátoru.

Způsob programování FGPA z mikroprocesoru je popsán v kapitole (9.2.2.2).

5.5.4 mikroprocesor ↔ okolí

Pro možnost rychlé interakce s uživatelem jsou na desce umístěna dvě tlačítka spojená s piny mikroprocesoru (PD11 a PD12), která v neaktivním stavu mají hodnotu log. 1, a při jejich stisknutí se na piny dostává hodnota log. 0. Z mikroprocesoru jsou vyvedeny přes předřadné odpory také tři LED diody (PD8, PD9 a PD10), které jsou aktivní (svítí) při signálu log. 1. Přehled zapojení těchto prvků už byl uveden v kapitole (5.5.1).

Z mikroprocesoru je dále vyvedeno několik dalších pinů na pinové lišty umístěné na okrajích desky. Na těchto pinech jsou vyvedeny například jednotky USART, SPI, I2C, CAN, UART, výstupy z časovačů a některé další periferie. Je vyveden i pin MCO1 (main clock output), na který může být vyveden libovolný hodinový signál podle parametrů nastavených v mikroprocesoru. Připojení LED diod a tlačítek ke konkrétním pinům procesoru už bylo uvedeno v tabulce (Tabulka 5.3). V tabulce (Tabulka 5.7) jsou uvedeny konkrétní piny mikroprocesoru vyvedené na pinové lišty a některé možné alternativní funkce, které lze na tyto piny mapovat.

Pro programování mikroprocesoru slouží SWD rozhraní, které je vyvedeno na konektor (5pinovou lištu), jeho uspořádání je uvedeno na obrázku (Obrázek 6.7) v kapitole realizace desky.

Tabulka 5.7: Vyvedené piny z mikroprocesoru a některé alternativní funkce

označení pinu	alternativní funkce						
	MCO	TIM1	I2C3	USART1	CAN1	SPI4	UART7
PA8	MCO1	CH1	SCL	CK	-	-	RX
PA9	-	CH2	SMB	TX	-	-	TX
PA10	-	CH3	-	RX	-	-	-
PA11	-	CH4	-	CTS	RX	-	-
PA12	-	ETR	-	RTS	TX	-	-
PE7	-	ETR	-	-	-	-	-
PE8	-	CH1N	-	-	-	-	-
PE9	-	CH1	-	-	-	-	-
PE10	-	CH2N	-	-	-	-	-
PE11	-	CH2	-	-	-	NSS	-
PE12	-	CH3N	-	-	-	SCK	-
PE13	-	CH3	-	-	-	MISO	-
PE14	-	CH4	-	-	-	MOSI	-
PE15	-	BKIN	-	-	-	-	-

5.5.5 FPGA ↔ okolí

Stejným způsobem jako k mikroprocesoru jsou také k hradlovému poli připojeny dvě tlačítka na piny (P93, P94) a tři LED diody (P83, P84, P85). Navíc je ještě jedno tlačítko připojeno k pinu PROG_B (P100) a při jeho stisknutí dojde k resetování hradlového pole.

Pro konfiguraci hradlového pole slouží rozhraní JTAG vyvedené na 6pinovou lištu, na kterou se dá připojit programovací kabel JTAG HS2 popsáný v kapitole (4.4). Hradlové pole je rovněž možné nakonfigurovat přes stejné JTAG rozhraní přímo z flash paměti mikroprocesoru, ve které je nahrán konfigurační soubor XSVF. Více o způsobu

programování mikroprocesorem bude uvedeno v kapitole (9.2.2.2). Pro signalizaci stavu hradlového pole je na pin DONE připojen tranzistor, který při nakonfigurování FPGA rozsvěcuje LED diodu (pin DONE je připojen i do mikroprocesoru).

Banka 2 hradlového pole je propojena se dvěma precizními paticemi DIL20. Patice jsou v současném návrhu osazeny obvody 74AHC245, což jsou 8kanálové obousměrné třístavové budiče popsané v kapitole (4.6). Na vstupy/výstupy těchto budičů mohou být přivedeny signály zvenčí, které následně budou zpracovávány hradlovým polem, směr toku dat a povolení budičů je řízeno z FPGA. Signály zvenčí budou přiváděny na dvouřadé pinové lišty umístěné na desce, kde řada blíže k budičům je propojena s přímo s těmito budiči a řada pinů blíže k okraji desky je připojena na zem. Díky tomuto uspořádání je možné připojit signály k desce kabelem odstraňujícím přeslechy mezi jednotlivými signály (plochý kabel, kde střídavě vedou žíly přenášející signál a žíly spojené se zemí). Budiče jsou umístěny mezi hradlovým polem a vstupy z okolí, aby v případě připojení vyšších napětí nedošlo ke zničení hradlového pole, ale jenom těchto budičů. Propojení pinů mezi budiči a hradlovým polem je uvedeno v tabulce (Tabulka 5.8), kde DIR značí směr ($DIR = \log. 0 \Rightarrow$ piny hradlového pole jsou vstupní, $DIR = 1 \Rightarrow$ hradlové pole – výstup) a #G povoluje průchod signálů budiči ($\#G = 0 \Rightarrow$ povoleno, $\#G = 1 \Rightarrow$ vysoká impedance).

Z banky 1 hradlového pole je ještě 9 pinů vyvedeno na pinovou lištu na kraji desky, seznam těchto pinů i s jejich jmény je uveden v tabulce (Tabulka 5.9).

Tabulka 5.8: Propojení hradlového pole s budiči

74AHC245 piny	piny FPGA	
	budič 1	budič 2
A1	P34	P52
A2	P33	P51
A3	P32	P50
A4	P31	P49
A5	P30	P44
A6	P29	P43
A7	P28	P42
A8	P27	P41
DIR	P35	P53
#G	P36	P37

Tabulka 5.9: Vyvedené signály z FPGA na pinovou lištu

značení pinu	název pinu
P56	IO_L01P_1
P57	IO_L01N_1
P59	IO_L02P_1/RHCLK0
P60	IO_L02N_1/RHCLK1
P61	IO_L03P_1/RHCLK2
P62	IO_L03N_1/TRDY1/RHCLK3
P70	IO_L05P_1
P71	IO_L05N_1
P72	IO_L06N_1

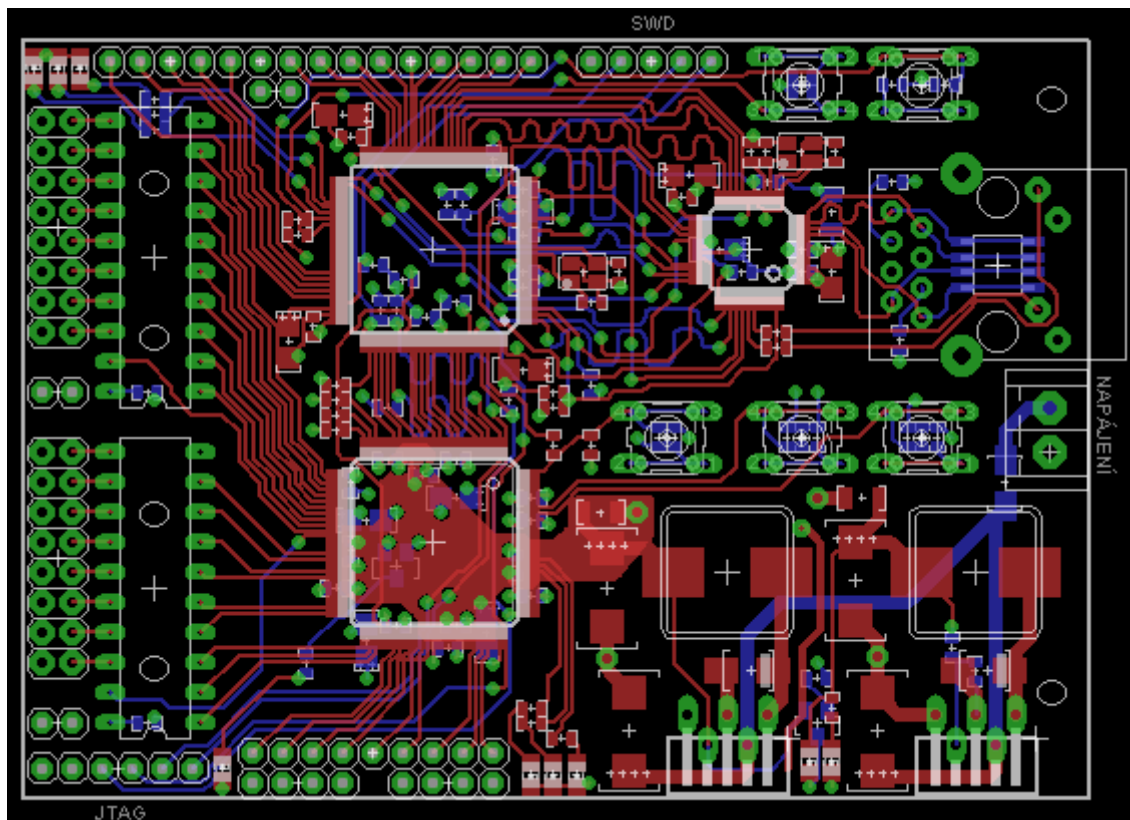
6 REALIZACE DESKY PLOŠNÝCH SPOJŮ

V předcházejících kapitolách bylo představeno provedení návrhu celé desky rozepsané od zapojení jednotlivých součástek až po propojení mezi nimi. Zde bude popsána deska jako celek a uvedeno několik obrázků s provedeným návrhem. Pro představu je nejprve

vložen celkový pohled na navrženou desku, kde na obrázku (Obrázek 6.1) je přehled rozmístění oblastí s rozvržením hlavních komponent a k nim přidružených součástek na desce. Na obrázku (Obrázek 6.2) je potom vidět provedený návrh desky v programu Eagle, červená plocha uprostřed je rozvedení napětí 1,2 V ve vnitřní vrstvě pod hradlovým polem.



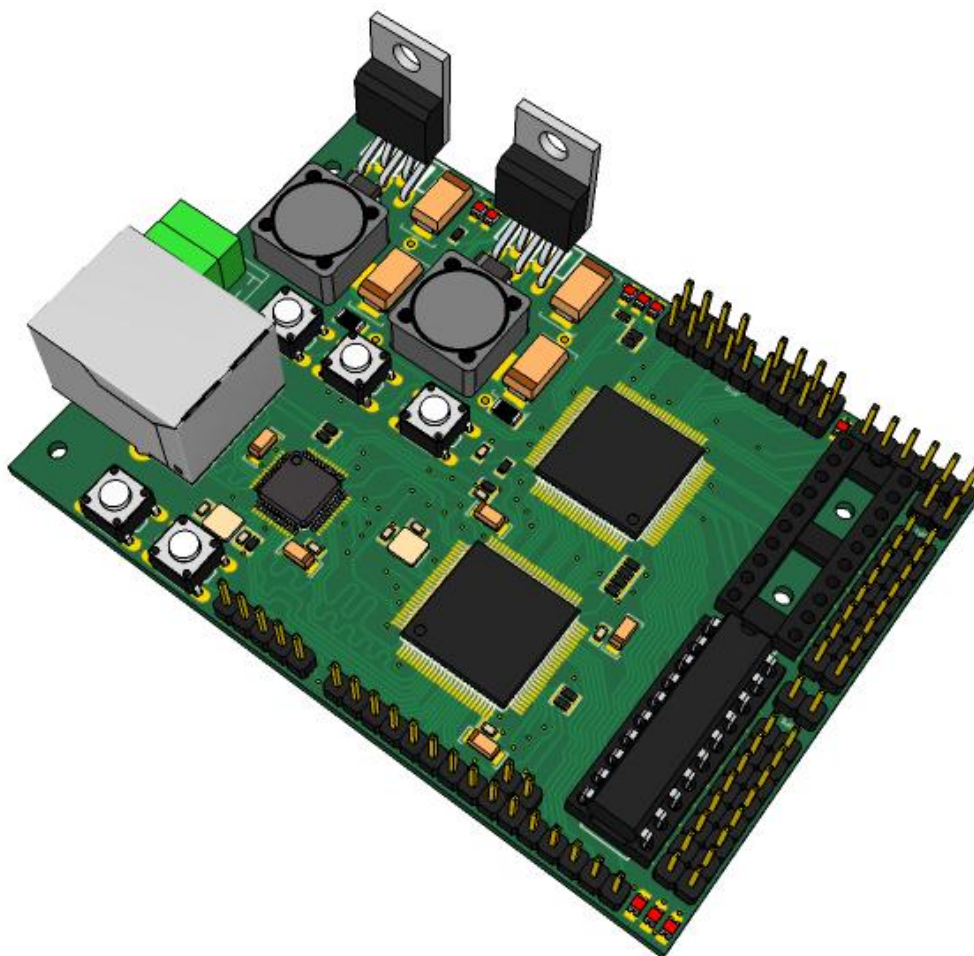
Obrázek 6.1: Rozvržení komponent na desce



Obrázek 6.2: Návrh desky v programu Eagle

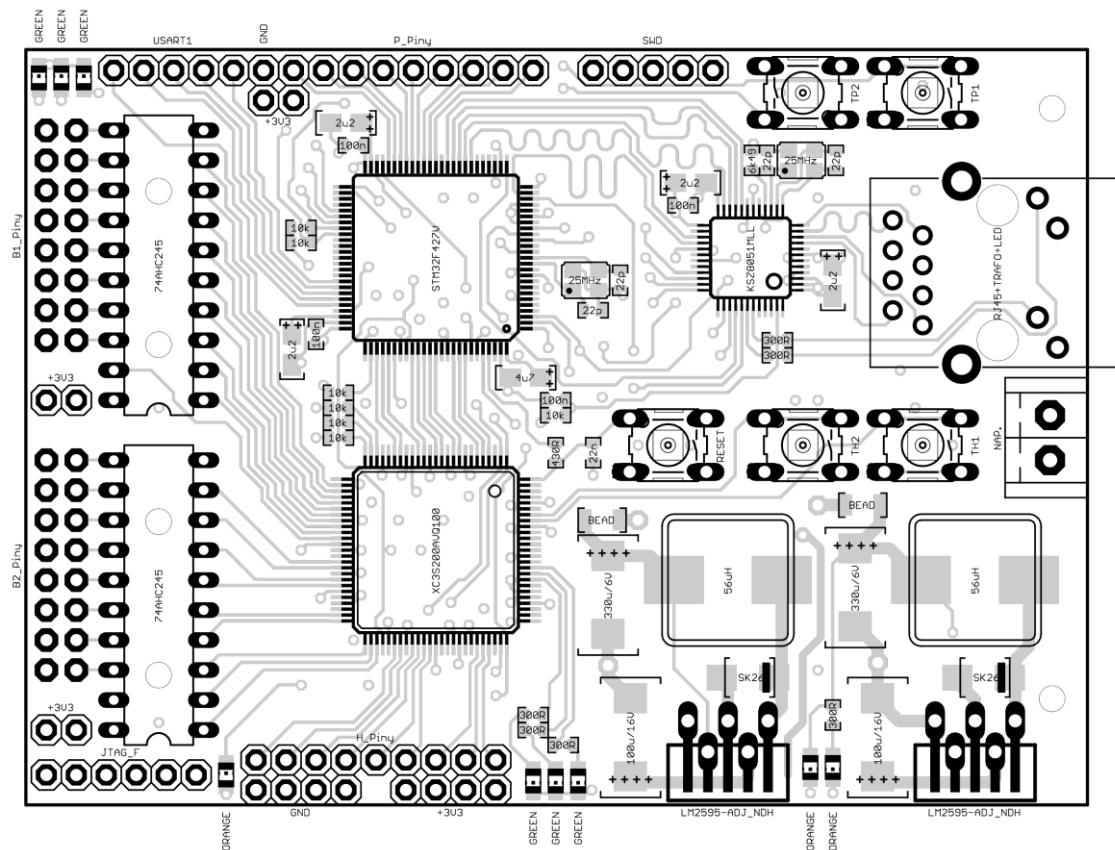
6.1 Výroba a popis desky jako celku

Návrh desky byl vytvořen v programu Eagle a na obrázku (Obrázek 6.2) je uvedena finální verze desky, která byla zadána do výroby. Pro představu, jak bude celá deska vypadat, byl vytvořen 3D model v programu SketchUp. Tento model je znázorněn na obrázku (Obrázek 6.3), kde je odstraněn jeden budič, aby bylo možné vidět i umístění montážních otvorů. Díky vytvořenému 3D modelu lze i vizuálně zkontrolovat možnost zapájení součástek, jestli je dostatek prostoru okolo a zda jsou plošky pro ruční zapájení dostatečně velké.

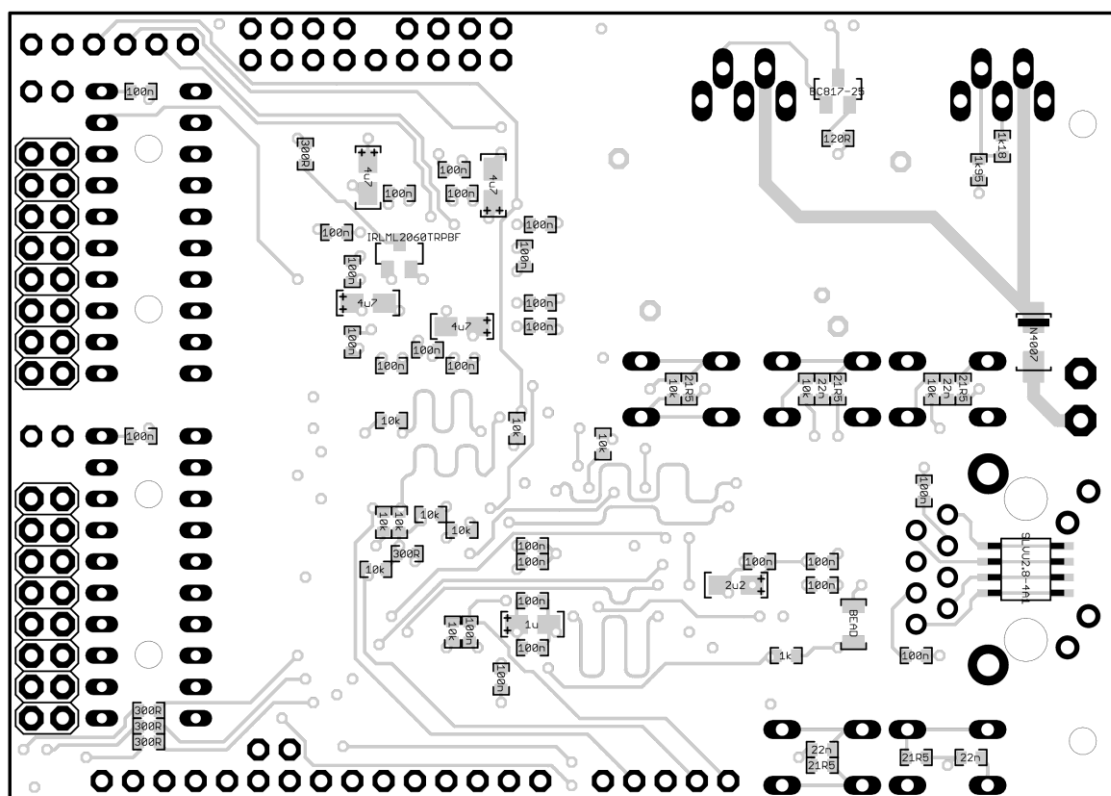


Obrázek 6.3: 3D model desky

Na sledujících dvou obrázcích (Obrázek 6.4, Obrázek 6.5) je vidět zvlášť vrchní a spodní strana desky s hodnotami součástek, případně s jejich označením.

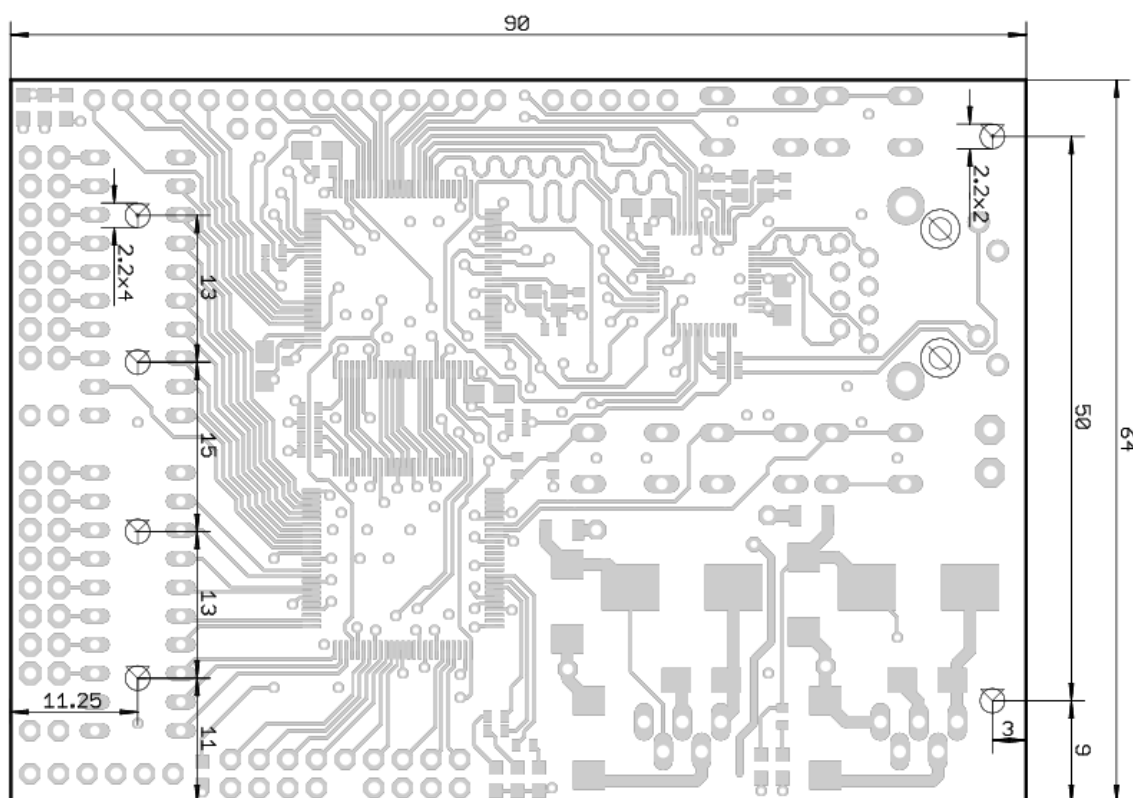


Obrázek 6.4: Vrchní strana desky s hodnotami součástek



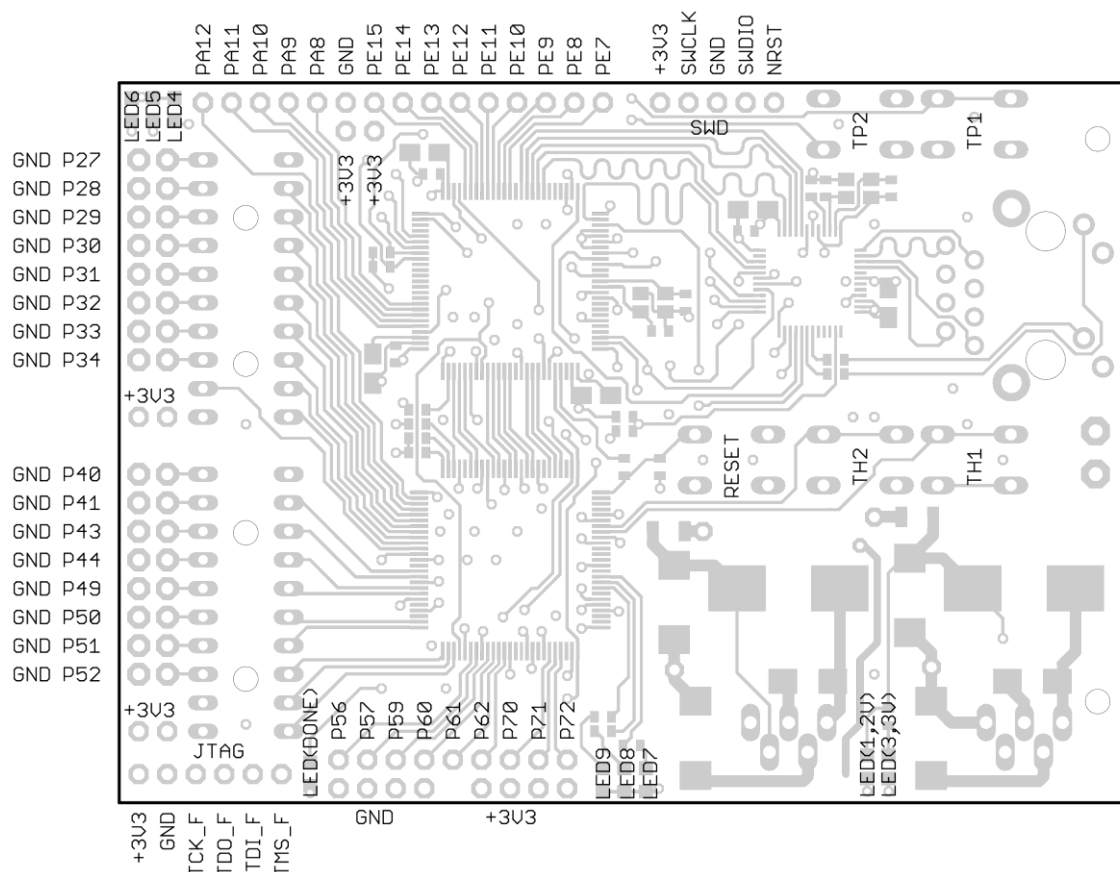
Obrázek 6.5: Spodní strana desky s hodnotami součástek

Pro upevnění desky ke dnu krabičky je v návrhu 6 děr pro šrouby o tloušťce 2 mm. Dvě díry jsou na vršku desky (u konektorů napájení) a čtyři na spodku uvnitř precizních DIL patič. Při vyndávání desky z krabičky je tedy nutno vyndat nejdříve budiče, nicméně to není na obtíž, protože i při programování bude deska zůstat ve spodním dílu krabičky, a není potřeba ji vyndávat. Zároveň se tímto umístěním ušetří místo a deska může mít menší rozměry. Deska je navržena o velikosti 64 x 90 mm, rozměry desky společně se zakótováním montážních děr jsou uvedeny na obrázku níže (Obrázek 6.6).



Obrázek 6.6: Rozměry desky a rozmístění montážních otvorů

Na následujícím obrázku (Obrázek 6.7) je vidět rozmístění a popis všech vývodů z desky pro jednotlivé piny v pinových lištách a také rozmístění LED diod a tlačítek s jejich popisem.



Obrázek 6.7: Popis a rozmístění pinů vedoucích z desky, LED diod a tlačítek

Jelikož deska má být univerzální a použitelná pro větší škálu aplikací, tak právě z tohoto důvodu je vyvedeno z mikroprocesoru a hradlového pole značné množství vývodů na pinové lišty po stranách desky. Také je na několik pinů po okrajích desky vyvedeno napájení a zem z vnitřních vrstev. Ve spodní části desky (tam kde je mikroprocesor, hradlové pole a budiče), nejsou žádné příliš vysoké součástky, a tak je možné na tuto část snadno navrhnout a posadit další desku pro rozšíření možností použití.

6.1.1 Osazení a oživení desky

Když přišla vyrobená deska, bylo potřeba ji osadit. Osazení bylo provedeno ručně ve škole za pomoci mikropájky s regulací teploty a pinzety. Osazeny byly nejprve čipy (mikroprocesor, FPGA, PHY) a následně součástky od nejmenších po největší. Průběžně bylo kontrolováno, jestli se nevytvořil zkrat mezi napájením. Před osazením součástek obsahujících plastové části byla deska očištěna toluenem a zbavena tak přebytečné kalafuny. V průběhu osazování byly změněny některé hodnoty součástek, jak již bylo zmíněno v předcházejících kapitolách, podle možností součástek dostupných ve škole. Měněny byly hodnoty rezistorů, kondenzátorů a cívek, ostatní větší součástky byly objednány předem.

Po osazení všech součástek mimo feritových korálků (BEAD) je na řadě oživení desky. K oběma spínaným zdrojům jsou připojeny LED diody pro indikaci stavu napájení a tím je zajištěn minimální proudový odběr ze zdrojů. Při samotném oživování byl použit nastavitelný zdroj napětí a osciloskop. Při zvyšování napětí na zdroji se na výstupu ze spínaných zdrojů začalo také objevovat napětí. Při dosažení vstupního napájecího napětí 12 V byl výstup z obou spínaných zdrojů stabilní a výstupní napětí se pohybovalo v mezích požadovaných pro připojené součástky na desce. Když byla deska oživena a ověřeno výstupní napětí, mohly být osazeny feritové korálky a tím byla deska připravena k dalšímu používání. Může se tedy začít programovat.

6.2 Dodatečné změny na desce oproti schématu

Zde jsou uvedeny změny, které byly provedeny až po vyrobení desky, a proto nejsou zahrnuty ve schématu. (změny čistě jenom hodnot součástek jsou ve schématu provedeny). Takovéto změny jsou celkem 3:

- 1) Ve schématu jsou uvedeny tantalové kondenzátory v pouzdru A pro hodnoty 1 μF , 2,2 μF a 4,7 μF , ty však jsou nahrazeny keramickými vrstvenými kondenzátory (MLCC kondenzátory) o stejné hodnotě. Navíc ještě nejsou osazeny 22 pF kondenzátory vedle krystalů.
- 2) Kvůli špatně zeditovanému pouzdru konektoru RJ45 v knihovně Eagle je otočena jedna LED dioda. Tento problém je vyřešen paralelním přidáním vývodové LED diody k té původní ve správném směru. Přidaná LED je plochá a přimáčknuta pod konektorem RJ45.
- 3) Do hradlového pole nebyl přiveden žádný hodinový signál, který by mohl sloužit pro měření signálů. Tento nedostatek je odstraněn dodatečným propojením pinu MCO2 mikroprocesoru s globálním hodinovým vstupem hradlového pole GCLK8. Ve skutečnosti byl při připojování drátu propojen na hradlovém poli i vedlejší pin, takže jsou spojeny piny GCLK8 a GCLK9, to ničemu nevádí, jen se s tím musí počítat při konfiguraci a nastavování vstupů a výstupů hradlového pole. V tabulce (Tabulka 6.1) je uvedeno toto dodatečné propojení.

Tabulka 6.1: Dodatečné připojení hodinového signálu do FPGA

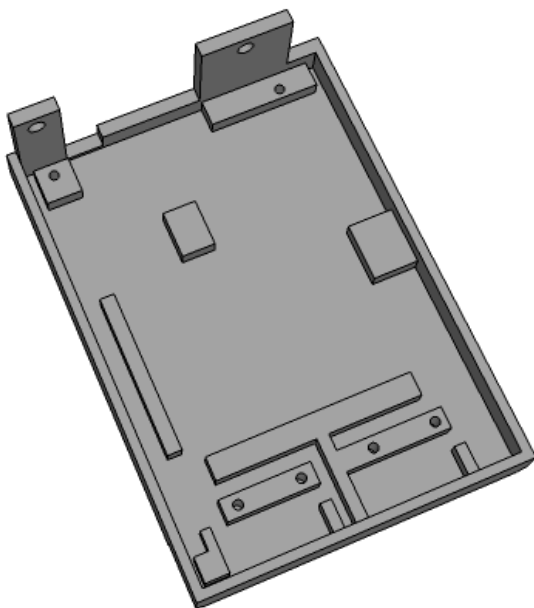
mikroprocesor		FPGA	
označení	název	označení	název
PC9	MCO2	P88	GCLK8
		P89	GCLK9

7 NÁVRH KRABÍČKY PRO DESKU

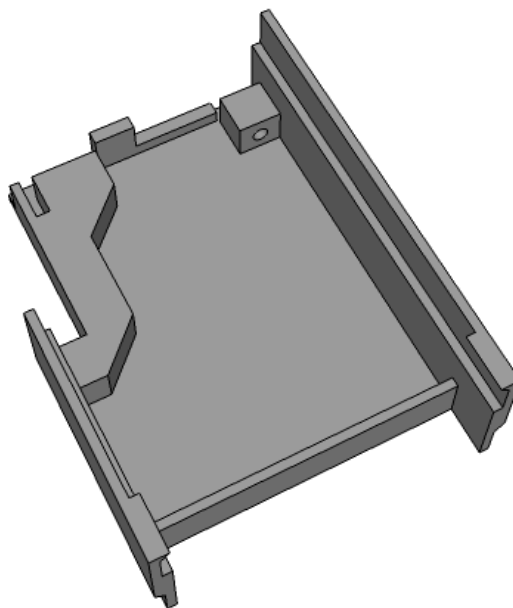
Pro vyrobenou desku je potřeba vytvořit krabičku, ta je navržena v programu SketchUp a vytisknuta na 3D tiskárně. Při navrhování krabičky se s výhodou dal použít už vytvořený 3D model desky.

Krabička je vyrobena ze dvou dílů, kde do spodního dílu je přišroubována deska a nepřepokládá se její vyndávání z tohoto dílu, proto může být přichycena šesti 2mm závitotvornými šroubky přímo do plastové krabičky. Vodicí drážky pro šrouby jsou navrženy s průměrem 2 mm, při vytisknutí na 3D tiskárně se průměr o něco zmenší a šroubky budou v krabici pevně držet. Na spodní straně desky (u budičů a pinových lišt) jsou 4 šroubky, protože při odpojování konektorů z desky nebo výměně budičů bude tato část nejvíce namáhána. Ve spodním díle krabičky jsou vytvořeny podpěrky na místech, kde jsou pinové lišty, DIL patice a tlačítka, aby se při manipulaci s těmito prvky zamezilo prohýbání desky a tím se předešlo jejímu zničení.

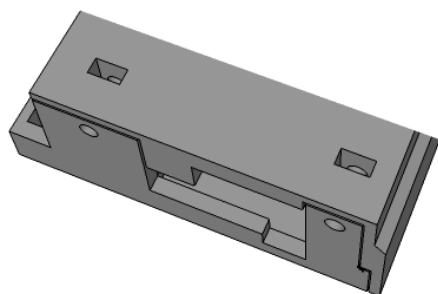
Vrchní díl krabičky se na spodní nasouvá s tím, že zkosení hran pro zajištění je vytvořeno až na konci obou dílů (asi 1,2 cm), aby nebylo nutné zasouvat díly po celé délce krabičky. V případě potřeby se dají díly sešroubovat 3mm šrouby (ze strany od konektorů), kde ve vrchním díle jsou vytvořeny otvory pro matky, tyto matky jsou v díle vloženy a pevně tam drží. Po přiložení vrchního dílu krabičky zůstanou odkryté spínané zdroje, aby zbytečně neohřívaly vnitřek krabičky a v případě potřeby se k nim dal upevnit chladič. Rovněž zůstanou odkryty pinové lišty propojené s budiči, některé piny s vyvedeným napětím 3,3 V a také piny propojené se zemí. Oba díly krabičky jsou na následujících obrázcích (Obrázek 7.1, Obrázek 7.2) a detaily jejich spasování na obrázcích (Obrázek 7.3, Obrázek 7.4).



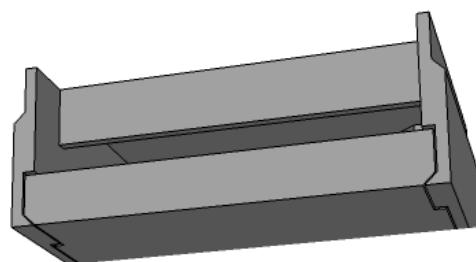
Obrázek 7.1: Spodní díl krabičky



Obrázek 7.2: Vrchní díl krabičky

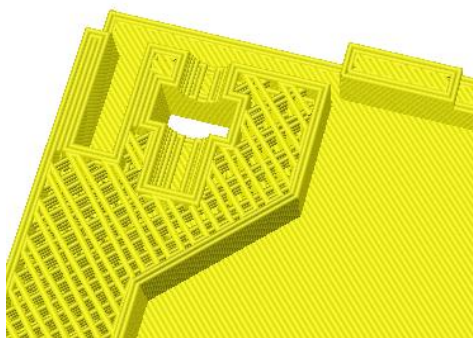


Obrázek 7.3: Detail spojení vpředu a výřezy pro matky v horním dílu krabičky



Obrázek 7.4: Detail spojení vzadu

Z prostředí SketchUp byl exportován STL soubor, který může být předán do programu Repetier-Host, v němž je možné sledovat průběh tisku krabičky po jednotlivých vrstvách podle konkrétních zadaných parametrů pro tiskárnu. Program také spočítá dobu potřebnou pro tisk a délku tiskového vlákna, která bude potřeba pro výtisk. Vytisknutí obou dílů krabičky trvalo 5:45 h a bylo spotřebováno 20,645 m vlákna. Na obrázku (Obrázek 7.5) je ukázka z programu pro vytisknutí 20 vrstev části vrchního dílu krabičky.



Obrázek 7.5: Náhled 3D tisku pro 20 vrstev

Celkem byly vytisknuty dvě kompletní krabičky, kde první výtisk byl zkušební, aby odhalil nedostatky při návrhu. Pro druhý výtisk se upravily rozměry tak, aby na sebe oba díly přesně pasovaly a nebyla mezi nimi zbytečně vůle. Ve druhém návrhu byla rovněž odstraněna část materiálu v rozích dílů, aby nedocházelo zbytečně k deformaci krabičky. Pokud bylo v rozích dílů více materiálu, docházelo tepelným působením při tisku k prohýbání těchto rohů směrem dovnitř.

Fyzicky vytisknutá a složená krabička i s usazenou deskou je vidět na obrázcích (Obrázek 7.6, Obrázek 7.7).



Obrázek 7.6: Složená krabička s deskou (zepředu)



Obrázek 7.7: Složená krabička s deskou (zezadu)

8 SOFTWARE PRO OTESTOVÁNÍ FUNKČNOSTI DESKY

Nyní už je k dispozici osazená a oživená deska zasazená v krabičce a je na řadě otestování funkčnosti všech jejích částí pomocí softwaru. Pro otestování jsou napsány jednoduché programy pro hradlové pole a pro mikroprocesor, tyto programy mají za úkol otestovat funkčnost a správné propojení mezi periferiemi na desce.

8.1 Testovací software pro mikroprocesor

Pro otestování správného propojení mikroprocesoru s okolím, hradlovým polem a PHY čipem jsou vytvořeny dva moduly. Pojmem modul je zde označována dvojice souborů „*modul.h*“ a „*modul.c*“. První je „*test_desky*“, který využívá ještě druhý modul, a to „*funkce_spolecne*“. Nejprve budou stručně popsány jednotlivé moduly s vytvořenými funkcemi a následně použití těchto funkcí pro otestování propojení na desce. Pro mikroprocesor jsou programy psány v jazyce C. Moduly jsou uloženy ve složce projektu v podadresářích „*\include*“ a „*\src*“.

Tlačítka ve schématu označená jako TP1 a TP2 jsou v programu označovány jako TL1 (TP1) a TL2 (TP2). Stejně tak mají diody připojené k mikroprocesoru ve schématu a v programu různá označení. Ve schématu jsou značeny od indexu 4 a v napsaném programu od indexu 1: LED1 (LED4), LED2 (LED5), LED3 (LED6).

8.1.1 Modul „funkce_spolecne“

V tomto modulu jsou zahrnuty funkce a makra využívané jak při testování funkčnosti desky, tak i při následujícím návrhu logického analyzátoru.

Jsou zde definována makra pro označení LED diod, tlačítek, výstupu hodin z mikroprocesoru (MCO2), pinů propojujících mikroprocesor s PHY čipem a pinů propojených s hradlovým polem. Funkce jsou popsány samostatně v následujících podkapitolách, kde v nadpisu je vždy uveden prototyp dané funkce.

void wait_cyklu(uint32_t cyklu)

Čekání mikroprocesoru po dobu počtu cyklů předaného ve vstupním parametru.

void LED_a_TL_inicializace(void)

Nastavení pinů s připojenými LED diodami jako výstupních na úroveň log. 0 a pinů s připojenými tlačítky jako vstupních.

void MCO2_inicializace(void)

Nastavení pinu MCO2 (PC9) jako výstupu hodin s frekvencí krystalu připojeného do mikroprocesoru (25 MHz) a povolení rychlého externího oscilátoru (HSE - high speed external), který vytváří hodinový signál právě z připojeného krystalu. Tento pin mikroprocesoru je propojen s pinem globálního hodinového vstupu do hradlového pole GCLK8 (P88).

void povoleni_hodin_GPIO_vsechny(void)

Povolení hodin pro všechny GPIO piny (general purpose I/O). Povolení hodin pro danou periferii je nutné před jakýmkoliv nastavováním parametrů této periferie. Tato funkce musí být zavolána na začátku programu, protože ve vytvořených inicializačních funkcích už pro piny GPIO portů není povolení hodin voláno, mimo funkci „MCO2_inicializace“, která tedy může být zavolána i dříve.

8.1.2 Modul „test_desky“

V tomto modulu jsou vytvořeny funkce pouze za účelem otestování funkčnosti propojení součástek na desce.

void timer10_inicializace_1s (void)

Povolení hodin pro časovač (TIM10) a povolení nastavení příznaku přerušení při přetečení čítače. Nastavení hodnot periody a předděliče tak, že dojde k vygenerování události (příznaku přerušení) po 1 s při nastavení systémových hodin na 168 MHz.

void timer10_cekej (void)

Povolí časovač (TIM10), vyčká na nastavení příznaku přerušení (po 1 s se současnou inicializací), zastaví časovač a vymaže příznak přerušení.

void FPGA_propojeni_test(void)

Nastaví všechny piny propojené s hradlovým polem jako výstupní a přivede na ně úroveň log. 1.

bool PHY_NAND_TREE_TEST (void)

Provede testování propojení PHY čipu pomocí „NAND Tree“ testu a předá návratovou hodnotu „true“, pokud testování proběhlo v pořádku, nebo „false“, pokud je problém v propojení.

Nastaví piny propojené s PHY čipem vedoucí signály #NAND_TREE a RST jako výstupní a pin vedoucí signál CRS jako vstupní. Nastaví signál #NAND_TREE na log. 0 a resetuje PHY čip, čímž je PHY přepnut do „NAND Tree“ módu.

Nastaví ostatní testované piny MII rozhraní propojené s PHY čipem na výstupní a do úrovně log. 1. Přečte hodnotu na CRS a uloží si ji do proměnné, která je dále využívána pro kontrolu správnosti propojení. Pro každý testovaný pin v pořadí, jak je uvedeno v tabulce (Tabulka 4.2), nastaví jeho hodnotu do log. 0 a otestuje, jestli došlo ke změně na výstupu. Pokud nedošlo, je testování ukončeno, vypnut „NAND Tree“ mód a navrácena hodnota „false“. Pokud celý proces testování proběhne bez chyby, je vypnut „NAND Tree“ mód a navrácena hodnota „true“.

Během testování se nastavují hodnoty na LED diody 5 a 6. Na LED5 je měněn stav s každým testovaným pinem a na LED6 je nastavena vstupní hodnota z CRS. Pokud testování probíhá úspěšně, LED diody mění své stavy stejně.

void test_desky(void)

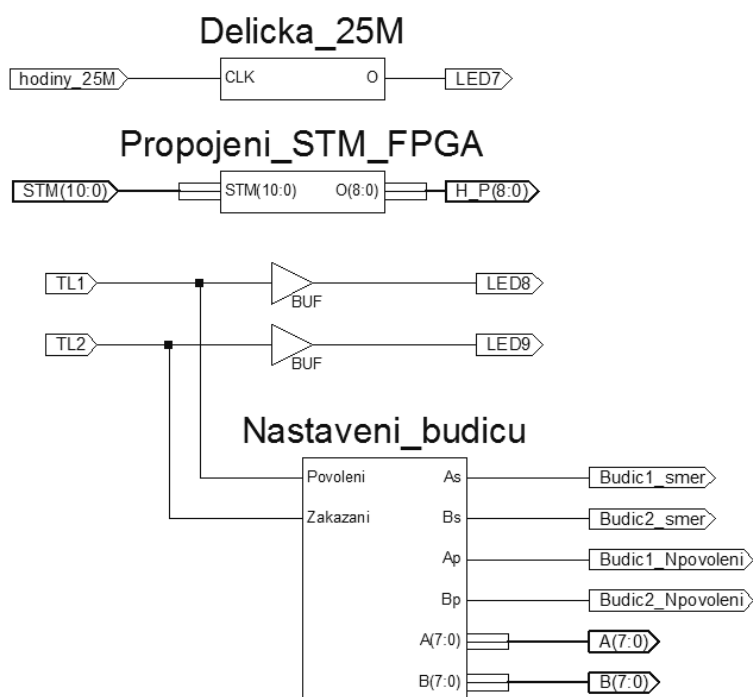
V této funkci jsou volány všechny doposud popsané funkce, proto je možné v hlavním programu zavolat pouze tuto jedinou funkci za účelem otestování propojení navržené a osazené desky univerzálního modulu.

Nejprve je nastavena frekvence pro sběrnice, systémové hodiny a jako zdroj systémových hodin je vybrán PLL. Poté je inicializován výstup MCO2, který přivádí hodinový signál do hradlového pole. Povoleny jsou hodiny do všech GPIO portů, inicializovány tlačítka a LED, rozsvícena LED4, inicializován časovač. Následuje provedení NAND Tree testu, při kterém blikají LED5 a LED6. Po dokončení testu je nastavena hodnota, které byla vrácena jako výsledek testu na LED5 po dobu 2 vteřin. Pokud tedy test proběhl úspěšně, je na 2 vteřiny rozsvícena LED5. Po těchto 2 vteřinách od ukončení testu jsou nastaveny na všechny piny vedoucí na pinové lišty nebo do FPGA hodnoty log. 1. Poslední částí kódu je nekonečná smyčka, ve které je každou vteřinu měněn stav LED4 a jsou kontrolována tlačítka. Tlačítko TP1 při stisknutí

rozsvěcuje LED5 a tlačítko TP2 ovládá LED6, zjišťování stavu tlačítek rovněž probíhá jednou za sekundu.

8.2 Testovací software pro FPGA

Pro otestování propojení hradlového pole jsou vytvořeny bloky popsané ve VHDL kódu a propojení bloků je provedeno ve schématu, jak je vidět na obrázku (Obrázek 8.1). Do prvních dvou bloků vstupují signály vyvedené z mikroprocesoru. Třetí blok a všechny signály s ním propojené jsou závislé už pouze na správném propojení hradlového pole s deskou. Tlačítka označená ve schématu s návrhem desky jako TH1 a TH2 jsou v programovém schématu v ISE označeny jako TL1 (TH1) a TL2 (TH2). Označení diod už se shoduje v obou schématech.



Obrázek 8.1: Schéma vytvořené v ISE pro testování desky (propojení FPGA)

Vstupy z tlačítek jsou v hradlovém poli přímo propojeny s diodami tak, že tlačítko TH1 na desce ovládá diodu LED8 a TH2 diodu LED9. Pokud je tlačítko v klidovém stavu, dioda svítí, a pokud je stisknuto, tak dioda nesvítí. Vstupy z tlačítek zároveň vedou do bloku „Nastaveni_budicu“.

V následujících kapitolách jsou popsány funkce jednotlivých bloků. Název bloku se vždy shoduje s názvem souboru umístěným v adresáři projektu. Pro každý blok je vytvořeno více souborů se stejným názvem rozlišených příponou, kde hlavní je soubor s popisem bloku ve VHDL a se symbolem pro použití ve schématu. Ostatní soubory jsou generovány při překladu.

Delicka_25M

Na vstup tohoto bloku je přiveden hodinový výstup z mikroprocesoru o frekvenci 25 MHz a uvnitř bloku je frekvence vydělena 25 milióny a vyvedena ven z bloku. Na výstupu z bloku je připojena LED7, která tedy bliká s frekvencí 1 Hz.

Propojeni_STM_FPGA

Do bloku vstupuje 11 signálů propojujících mikroprocesor s hradlovým polem a výstup z bloku je propojen s 9pinovou lištou připojenou k FPGA. Uvnitř bloku je prvních 8 vstupních signálů propojeno přímo s prvními 8 výstupními signály a zbývající 3 signály jsou přes hradlo AND připojeny na poslední výstup.

Nastaveni_budicu

Do tohoto bloku vstupují signály ze dvou tlačítek, které ovládají pouze výstupy Ap, Bp. Ostatní výstupy z bloku jsou nastaveny na log. 1. Vstupy reagují na úroveň log. 0 (stisknuté tlačítko). Pokud je na vstupu „Povolení“ log. 0, jsou ovládané výstupy nastaveny na log. 0. Pokud je „Zakazani“ = 0, výstupy jsou nastaveny na log. 1. Když jsou oba vstupy zároveň v log. 0, má přednost „Povolení“. Výstupy z bloku jsou připojeny k budičům, kde ovládané výstupy Ap, Bp řídí povolení budičů. Výstupy As, Bs řídí směr budičů, které jsou tedy nastaveny jako výstupní a na všechny výstupy budičů jsou přivedeny log. 1.

8.3 Shrnutí testování desky s využitím programů pro mikroprocesor a FPGA

Pro otestování správného propojení součástek na desce jsou vytvořeny programy pro mikroprocesor a pro hradlová pole. V programu pro hradlové pole jsou využívány signály generované mikroprocesorem a tím je testováno i jejich vzájemné propojení. Je potřeba mít nahraný program jak v mikroprocesoru, tak v FPGA pro ověření všech funkcí tak, jak jsou popsány v následujících odstavcích.

Při nahrání testovacího programu do mikroprocesoru je nastaven výstup hodinového signálu z MCO2, který vede do hradlového pole, na 25 MHz a rozsvícena LED4. Následně je proveden NAND Tree test během kterého blikají diody LED5 a LED6. Pokud test probíhá bez chyby, tak tyto diody blikají současně. Testováno je celkem 17 pinů, takže diody by měly celkem 17krát změnit svůj stav (9krát bliknou). Při dokončení testu je na 2 vteřiny rozsvícena LED5, pokud test proběhl v pořádku (pokud ne, pouze zdržení 2 s). Po tomto zpoždění jsou na piny propojující mikroprocesor s FPGA, a na vývody připojené k pinovým lištám nastaveny hodnoty log. 1. Na konci programu je nekonečná smyčka, která každou vteřinu mění stav LED4 a kontroluje stav

tlačítek TP1 a TP2. Pokud jsou tlačítka stisknuta v době kontroly, rozsvítí se příslušně diody, pro TP1 je to LED5 a pro TP2 je to LED6.

Testovací program v hradlovém poli dělí frekvenci přiváděnou z mikroprocesoru a se získanou frekvencí 1 Hz bliká s diodou LED7. Ostatní dvě diody jsou uvnitř hradlového pole propojeny přímo s tlačítky, kde v klidovém stavu svítí a při stisknutém tlačítku nesvítí. Tlačítko TH1 ovládá diodu LED8 a TH2 diodu LED9. Tlačítka jsou využita i pro řízení budičů, kde při stisknutí tlačítka TH1 se průchod signálů budiči povolí a při stisknutí TH2 se zakáže (log. 1 na pinu #G budiče). Budiče jsou nastaveny jako výstupní (log. 1 na pinu DIR budiče) a z hradlového pole na všechny jejich kanály přiházejí hodnoty log. 1. Poslední testovací funkcí je přivedení signálů přicházejících z mikroprocesoru do hradlového pole na pinovou lištu u okraje desky. Mikroprocesor a FPGA jsou mezi sebou propojeny 11 vodiči, zatímco lišta připojená k FPGA je pouze 9pinová. Prvních 8 signálů je propojeno přímo a zbývajících 3 jsou přes hradlo AND přivedeny na poslední pin v liště.

Správnost propojení mezi programovacími konektory a obvody na desce je ověřena úspěšným nahráním programu do těchto obvodů (FPGA a mikroprocesoru).

Pro přehled a možnost rozpoznání případné chyby je v tabulce (Tabulka 8.1) naznačeno programové propojení mezi jednotlivými piny a periferiemi, kde mezi vstupy a výstupy v jednom řádku tabulky jsou závislosti podle předchozího popisu. Pro otestování výstupních úrovní na pinech, rozmístěných po okrajích desky, byla využita LED dioda připojována postupně mezi každý pin a zem. Protože na všechny piny v lištách jsou vyvedeny úrovně log. 1, tak při připojení LED diody ke každému pinu, musí dioda svítit.

Tabulka 8.1: Přehled propojení mezi piny zařízení a periferiemi v programech pro testování desky

tlačítka	LED	Mikroprocesor		FPGA			
		vstupy	Výstupy	vstupy	výstupy	výstupy (výstupy mikroproce.)	
	LED4		PD8			P72 (PB7)	
TP1	LED5	PD12	PD9			P71 (PB6)	
TP2	LED6	PD11	PD10			P70 (PB5)	
	LED7		PC9 (MCO2)	P88	P85	P62 (PD7)	
TH1	LED8			P93	P84	P61 (PD6)	
TH2	LED9			P94	P83	P60 (PD5)	
							P59 (PD4)
							P57 (PD3)
							P56 (PC12 & PC11 & PC10)

9 REALIZACE LOGICKÉHO ANALYZÁTORU

Cílem diplomové práce je navržení univerzálně použitelného zařízení s možností jeho použití pro dálkové měření digitálních veličin. V předchozích kapitolách byl proveden návrh tohoto zařízení, popsáno jeho fyzické vytvoření včetně základního softwarového otestování funkčnosti a správného propojení na desce. V této kapitole je tedy na řadě představení návrhu a vytvoření výsledného produktu pro měření veličin. Jako měřicí aplikace je na desce implementován logický analyzátor. Naměřená data jsou z desky odesílána po ethernetu a pro PC je napsána aplikace zobrazující naměřené průběhy signálů.

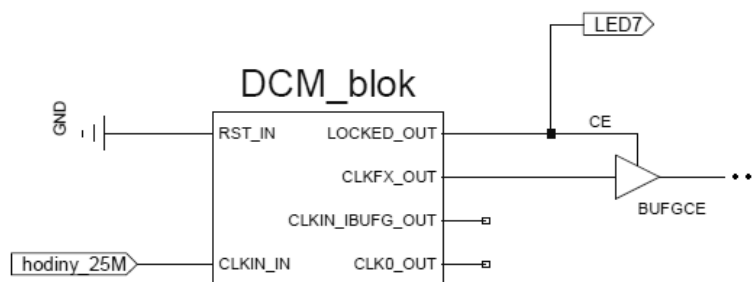
Tato kapitola je logicky rozdělena do několika podkapitol, které samostatně popisují části logického analyzátoru implementovaného v univerzálním modulu pro FPGA i pro mikroprocesor a aplikaci pro grafické zobrazování průběhů.

9.1 Část analyzátoru implementovaná v FPGA

V hradlovém poli je implementována část, která měří signály na vstupu, ukládá je do vyrovnávací paměti vytvořené uvnitř FPGA a naměřená data následně odesílá mikroprocesoru přes SPI rozhraní.

Celý návrh je proveden jako synchronní, kde řídicí frekvence je 125 MHz získaná z frekvence 25 MHz vstupující do hradlového pole. Pro získání této frekvence je využit DCM blok uvnitř hradlového pole a jeho zapojení je uvedeno ve schématu (Obrázek 9.1).

Jako vstup do tohoto bloku jsou přivedeny hodiny získané z MCO2 pinu mikroprocesoru o frekvenci 25 MHz. Uvnitř bloku je vstupní frekvence 5krát vynásobena a vyvedena na výstup „CLKFX_OUT“. Ostatní nepoužité výstupy jsou při vytváření bloku generovány automaticky a při syntéze návrhu budou odstraněny. Výstup s frekvencí 125 MHz pokračuje do bloku „BUFGCE“, což je globální hodinový buffer s povolením výstupu, a do další části obvodu je tento hodinový signál propuštěn až poté, co DCM blok dosáhne stabilní frekvence na výstupu (LOCKED_OUT = 1). Pokud je dosaženo stabilní frekvence, svítí LED7.



Obrázek 9.1: Zapojení DCM bloku

Budiče na desce jsou v této aplikaci nastaveny pevně tak, že první budič (vlevo při pohledu na budiče) je nastaven jako výstupní a druhý jako vstupní. Na výstupu prvního budiče jsou generovány signály z hradlového pole a druhý budič slouží jako vstupní brána pro měření signálů implementovaným logickým analyzátozem. Nastavení řídicích vstupů budičů je vidět na obrázku (Obrázek 9.2), kde jsou pouze na dané piny hradlového pole přivedeny příslušné úrovně signálů.

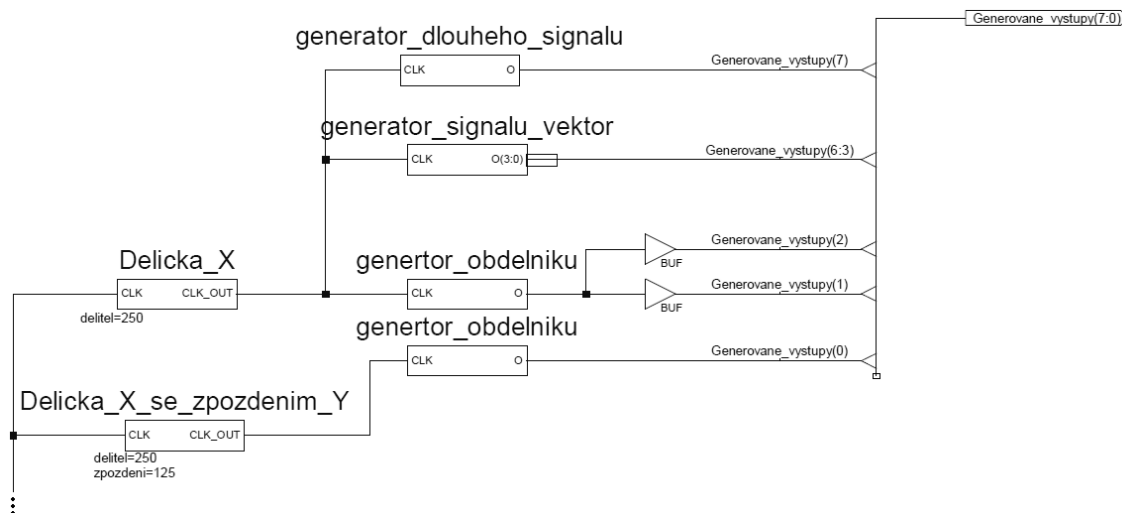


Obrázek 9.2: Nastavení směru a povolení budičů

Zbylá část schématu už je synchronně řízena právě generovanou frekvencí 125 MHz a popsána v následujících dvou kapitolách.

9.1.1 Generování signálů

Tato část je na obrázku (Obrázek 9.3) a slouží pro generování signálů, které jsou vyvedeny jako výstup z desky na piny za prvním budičem. Po propojení obou budičů je díky tomu možno sledovat generované průběhy a ověřit funkci analyzátoru.



Obrázek 9.3: Část pro generování signálů

V této části jsou pouze bloky pro dělení frekvence nastavitelné přímo ze schématu a bloky pro generování různých průběhů signálů. Všechny bloky jsou popsány níže.

Delicka_X

Nastavitelná dělička frekvence se střídou 50 %. Ve schématu (Obrázek 9.3) do ní vstupuje frekvence 125 MHz a dělitel („delitel“) je nastaven na hodnotu 250, takže na výstupu z děličky je frekvence 0,5 MHz.

Delicka_X_se_zpozdenim_Y

Ten stejný blok jako „Delicka_X“, pouze se u něj dá navíc nastavit doba zpoždění („zpozdeni“ v počtech period vstupní frekvence). S nastavením provedeným ve schématu je na výstupu rovněž frekvence 0,5 MHz a průběh je oproti první děličce opožděn o polovinu výstupní periody (o 1 μ s).

generator_obdelniku

S příchodem náběžné hrany na vstupu je na výstupu měněna úroveň signálu.

generator_signalu_vektor

S náběžnou hranou na vstupu jsou na výstupech (4 výstupy) generovány signály. Vždy je pouze na jednom výstupu vysoká úroveň vytvořená posunující se log. 1. Celý průběh se opakuje po 7 stavech a to v pořadí (0001 – 0010 – 0100 – 1000 – 0100 – 0010 – 0001), zapsáno od nejvýznamnějšího bitu.

generator_douheho_signalu

Tento blok generuje signál měnící se po větším počtu náběžných hran vstupního signálu. Začíná od log. 0 a po počtech (2 – 3 – 300 – 195) náběžných hran na vstupu se mění na opačnou úroveň. S nastavením vstupních hodin na 0,5 MHz probíhají změny následovně (4 μ s (0) – 6 μ s (1) – 600 μ s (0) – 390 μ s (1)), kde je uveden čas trvání dané úrovně v závorce.

9.1.1.1 Blok používaný při testování

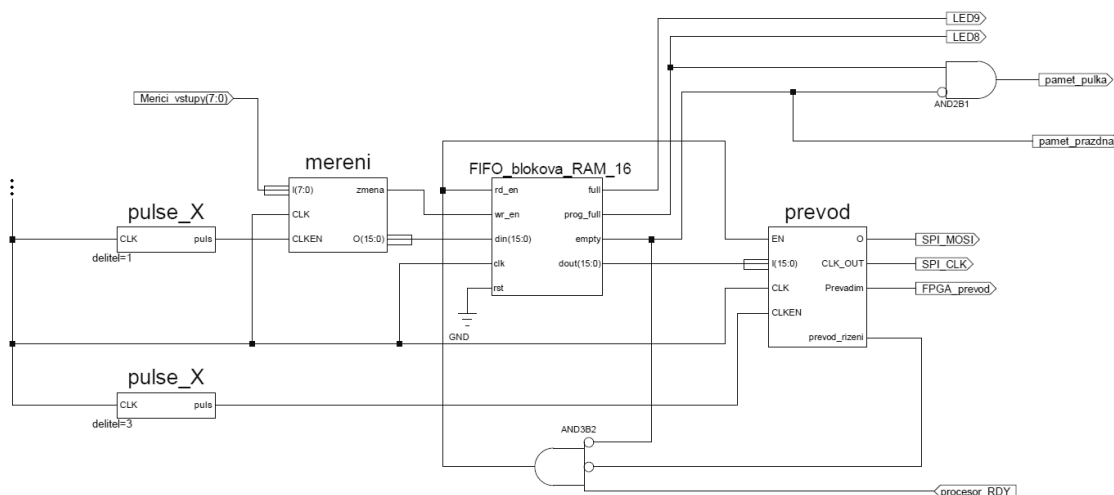
Při vytváření logického analyzátoru byl pro testování používán blok, který generuje určitou posloupnost signálů. Tato posloupnost poté byla kontrolována při příjmu společně s konstantním časem mezi změnami sekvencí. Ve finálním schématu není tento blok použit, ale je umístěn v adresáři projektu.

generator_znaku

Blok s 8 výstupy, které se mění při příchodu náběžné hrany na vstupu. V bloku je vytvořeno celkem 15 stavů, které se opakují. Stavy reprezentují 15 znaků z ASCII tabulky v binární podobě a opakují se v pořadí (A, B, C, D, ..., O), kde 'A' $\approx$ 0100 0001 v zápise od nejvýznamnějšího bitu.

9.1.2 Měření signálů a odesílání dat

Tato část schématu již zajišťuje funkci logického analyzátoru pro měření a odesílání dat. Schéma je uvedeno na obrázku (Obrázek 9.4), jednotlivé bloky a jejich význam je popsán v následujících kapitolách. Do schématu v místě naznačeném třemi tečkami vstupuje hodinový signál pro synchronizaci obvodů získaný z DCM bloku o frekvenci 125 MHz. Jako měřicí vstupy jsou přiváděny signály ze vstupního budiče.



Obrázek 9.4: Schéma pro měření a odesílání dat

pulse_X

Blok s nastavitelným parametrem „delitel“, který po zvoleném počtu náběžných hran na vstupu vygeneruje na výstupu impuls s hodnotou log. 1. Impuls je široký jednu periodu vstupního signálu. Ve schématu (Obrázek 9.4) jsou dva tyto bloky, kde první pokračuje do bloku pro měření a je zde vložen pouze pro možnost úpravy vzorkovací frekvence. V současném návrhu je na jeho výstupu stále hodnota log. 1 a vzorkovací frekvence je tedy maximální možná. Druhý blok generuje impulsy pro část zajišťující SPI komunikaci s mikroprocesorem, kde s každou třetí náběžnou hranou na vstupu je vygenerován jeden impuls.

mereni

Tento blok zajišťuje měření signálů pro 8 vstupních kanálů. S každou náběžnou hranou řídicí frekvence kontroluje, zda došlo ke změně ve vstupním vektoru „I(7:0)“. Pokud došlo ke změně na některém ze vstupních kanálů, vygeneruje se na výstupu „zmena“ signál log. 1 a po dobu jednoho řídicího cyklu informuje o nastalé změně. Do výstupního vektoru „O(15:0)“ jsou vloženy hodnoty odpovídající předchozímu stavu vstupního vektoru „I(7:0)“ a za tento údaj je připojena informace o délce trvání tohoto stavu (než došlo ke změně) v podobě 8bit čísla. Oba výstupy jsou připojeny do vyrovnávací paměti FIFO a s následující náběžnou hranou řídicího signálu je údaj zakódovaný ve výstupním vektoru uložen do paměti. Pokud se po dobu 255 cyklů

(maximální hodnota čítače) měřící frekvence nezmění stav vstupního vektoru, je současný stav uložen do paměti s tímto maximálním časovým údajem.

Vstupy:

- I(7:0) – 8bit vektor, na který jsou přivedeny měřené signály
- CLK – řídicí hodinový signál
- CLKEN – vstup pro nastavení vzorkovací frekvence

Výstupy:

- zmena – informuje o tom, že nastala změna na vstupu
- O(15:0) – 16bit vektor pro přenos zakódované informace o naměřených hodnotách

FIFO_blokova_RAM_16

Je to paměť typu FIFO s šířkou 16 b a hloubkou 16 384 hodnot, celková velikost paměti je tedy 256 Kb. Tato vyrovnávací paměť je vytvořena v blokové RAM paměti hradlového pole. Pro vytvoření tohoto bloku byl využit nástroj „CORE generator“. Je použit model paměti „First Word Fall Through“, kde data na výstupu jsou k dispozici už před příchodem požadavku na čtení (pokud už byla nějaká do paměti uložena). [29]

Paměť je synchronně řízena s frekvencí 125 MHz. Pokud je požadavek k zápisu (wr_en) aktivní v době příchodu náběžné hrany hodinového signálu, jsou data ze vstupního portu (din) zapsána do paměti. Pokud je aktivní požadavek pro čtení (rd_en), jsou vystavena následující data z paměti na výstupní port (dout). V případě prázdné paměti je nastaven výstup „empty“ do log. 1. Pokud je paměť plná, je nastaven výstup „full“. Výstup „prog_full“ je nastaven, pokud počet dat v paměti překročí zadaný limit. V návrhu je limit nastaven na poloviční zaplnění paměti (8 192 vzorků). Asynchronní reset paměti je trvale neaktivní, takže paměť je vymazána pouze při konfiguraci hradlového pole.

Vstupy:

- rd_en – povolení čtení z paměti (synchronně s clk)
- wr_en – povolení zápisu do paměti (synchronně s clk)
- din(15:0) – 16bit vstup pro ukládaná data
- clk – řídicí hodinový signál
- rst – reset paměti

Výstupy:

- full – příznak plné paměti
- prog_full – příznak alespoň polovičního zaplnění paměti (≥ 8192 vzorků)
- empty – příznak prázdné paměti
- dout(15:0) – 16bit výstup dat z paměti

prevod

Tento blok slouží pro převod informace z paralelní reprezentace (vstup I) na sériovou (výstup O) a její odeslání přes SPI do mikroprocesoru, kde hradlové pole funguje jako master pro komunikaci.

Blok pro převod je synchronizován řídicími hodinami na vstupu „CLK“. Na vstup „CLKEN“ jsou přiváděny impulsy s frekvencí 125/3 MHz, kterými je řízeno generování hodin pro SPI komunikaci a převod dat do sériové podoby. Rychlost komunikace po SPI (frekvence SPI hodin) je tedy 125/6 MHz (asi 20,8 MHz). Pokud je „EN“ = 1 v okamžiku příchodu náběžné hrany synchronizačních hodin, dojde k dočasnému uložení paralelní informace ze vstupu „I(15:0)“, vynulování vnitřního čítače, nastavení výstupu „prevod_rizeni“ = 1 a v dalším cyklu synchronizačních hodin je zahájen převod a odesílání informace do mikroprocesoru.

Na výstup „O“ jsou data vystavována zároveň s náběžnou hranou hodinového signálu výstupu „CLK_OUT“, který řídí přenos dat po SPI. Výstup „Prevadim“ je zaveden do mikroprocesoru a informuje o probíhajícím přenosu. Je aktivní pouze v době převodu informace z paralelní na sériovou (16 cyklů CLK_OUT). Poslední výstup „prevod_rizeni“ je aktivní od okamžiku přípravy na převod (od uložení vstupu „I(15:0)“ uvnitř bloku při „EN“ = 1).

Vstupy:

- EN – povolení pro zahájení převodu a přenosu
- I(15:0) – 16bit vstup informace v paralelní podobě
- CLK – řídicí hodinový signál
- CLKEN – vstup pro nastavení frekvence přenosu

Výstupy:

- O – výstup informace v sériové podobě (MOSI pro SPI přenos)
- CLK_OUT – výstup generující hodinový signál pro SPI komunikaci, v současném návrhu $\approx 20,8$ MHz
- Prevadim – aktivní (log. 1), pokud probíhá přenos (16 cyklů „CLK_OUT“)
- prevod_rizeni – výstup pro řízení převodu informace, je aktivní od povolení převodu (od „EN“ = 1)

9.1.2.1 Hradla NAND ve schématu

V návrhu jsou vložena dvě NAND hradla. Jedno 2vstupé, kde na jeho výstupu je informace o polovičním naplnění paměti v FPGA. Tuto informaci není možné získat přímo z výstupu „prog_full“ FIFO paměti, protože po nakonfigurování hradlového pole jsou všechny tři výstupy informující o stavu zaplnění paměti v log. 1. Na jednom vstupu tohoto hradla je signál „prog_full“ a na druhém invertovaný signál „empty“

signalizující, že paměť není prázdná. Tímto opatřením je tedy dosaženo, že signál „pamet_pulka“ není aktivní ihned po nakonfigurování FPGA, ale až při polovičním zaplnění paměti.

Druhé hradlo je 3vstupé, jeho výstup generuje signál pro povolení zápisu do paměti a zároveň také povoluje zahájení převodu v bloku pro SPI komunikaci. Na jeho vstupy jsou přivedeny signály informující o tom, že paměť není prázdná „empty“ = 0, právě neprobíhá žádný převod „prevod_rizeni“ = 0 a ještě informace z procesoru, že je připraven přijímat data „procesor_RDY“ = 1. Při této kombinaci je s náběžnou hranou synchronizačních hodin uložena informace na vstupu bloku pro převod a z paměti je na výstupní datový port připravena další uložená hodnota. Zároveň je nastaven signál „prevod_rizeni“ do log. 1, čímž se změní výstup NAND hradla a je zabráněno dalšímu vyčítání dat z paměti, dokud není převod dokončen („prevod_rizeni“ = 0). Jednotlivé přenosy po SPI na sebe přímo navazují, dokud tok dat není přerušen signálem z mikroprocesoru nebo vyprázdněním paměti (mikroprocesor také reaguje na prázdnou paměť).

9.1.3 Shrnutí návrhu v FPGA

V hradlovém poli je vytvořen logický analyzátor schopný měřit současně 8 kanálů se vzorkovací frekvencí 125 MHz. Měřená data jsou pro jeden vzorek zakódována do 16bitového čísla, kde prvních 8 bitů odpovídá úrovní signálů na měřicích kanálech a druhých 8 bitů odpovídá času (počtu vzorkovacích period), po který platil stav signálů v první části informace. Při vzorkovací frekvenci 125 MHz je doba jedné periody 8 ns. Maximální měřitelná frekvence vstupního signálu je však omezena v závislosti na rychlosti odesílání naměřených dat do mikroprocesoru. Omezení musí být takové, aby nemohlo dojít k naplnění vyrovnávací paměti a tím ke ztrátě spojitosti na měřených signálech z důvodu přeplnění paměti. Naměřená data jsou odesílána z hradlového pole po SPI s frekvencí $\approx 20,8 \text{ MHz}$ ², to při 16bit přenosu odpovídá času 768 ns pro jeden přenos. S dostatečnou rezervou pro režijní nároky a se zaokrouhlením na dobře zapamatovatelnou hodnotu je brán jako časový interval potřebný pro odeslání informace o jednom vzorku čas 1 μs . To znamená, že vstupní signál při měření na jednom kanále se může měnit s minimálním časem 1 μs (maximální frekvencí 1 MHz), což odpovídá obdélníkovému signálu o frekvenci 0,5 MHz. S tímto omezením byl provedený návrh testován a při testování nikdy nedošlo k přeplnění paměti. Při využití více kanálů se minimální čas prodlužuje (frekvence zmenšuje), mezní hodnoty pro současné použití více kanálů jsou uvedeny v tabulce (Tabulka 9.1). Přesnost měření však zůstává vždy stejná a odpovídá rozlišení v čase na 8 ns.

² Maximální rychlost komunikace mikroprocesoru pro SPI3 je 22,5 Mb/s (22,5 MHz). [6]

Tabulka 9.1: Mezní hodnoty pro měření při využití počtu kanálů

využitý počet kanálů	mezní hodnoty změn signálů	
	minimální čas [μ s]	maximální frekvence [kHz]
1	1	1 000
2	2	500
3	3	≈ 333
4	4	250
5	5	200
6	6	≈ 166
7	7	≈ 142
8	8	125

Ukládání dat do paměti probíhá nepřetržitě, pokud dojde ke změně na některém z měřených vstupů, nebo čítač uvnitř bloku měření dosáhne své maximální hodnoty 255 (2,04 μ s). Vyčítání z paměti je povoleno, pokud na výstupu 3vstupého hradla je log. 1. Pokud je tento signál aktivní, dochází k neustálému vyčítání dat z paměti, jejich převodu a odesílání do mikroprocesoru. Master v SPI komunikaci je hradlové pole a generuje hodinový signál, pouze pokud odesílá data.

Po dosažení stabilní frekvence z výstupu DCM bloku, a tedy pokud prochází synchronizační hodinový signál do celého obvodu, svítí dioda LED7. Pokud je dosaženo poloviny zaplnění paměti, svítí LED8, a pokud je paměť plná, svítí LED9.

9.1.3.1 Propojení signálů mezi hradlovým polem a mikroprocesorem:

Směr z FPGA do mikroprocesoru

- pamet_pulka – informace o polovičním zaplnění paměti
- pamet_prazdna – informace o prázdné paměti
- FPGA_prevod – informace o právě probíhajícímu přenosu
- SPI_MOSI – přenos dat přes rozhraní SPI
- SPI_CLK – řídicí hodinový signál pro SPI

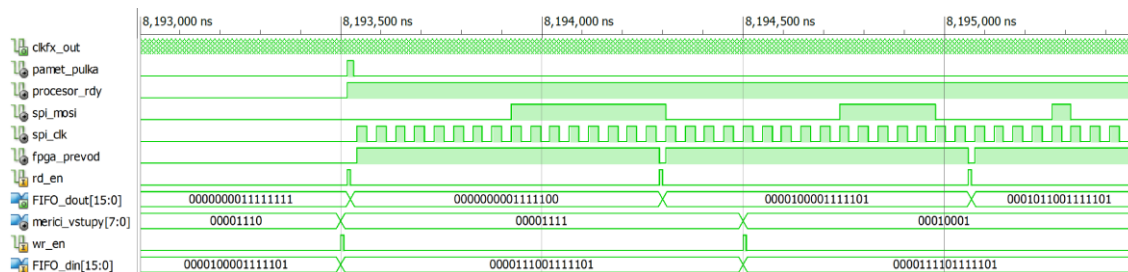
Směr z mikroprocesoru do FPGA

- procesor_RDY – informuje hradlové pole o připravenosti mikroprocesoru pro přenos dat na SPI rozhraní

9.1.3.2 Simulace

Zde na obrázku (Obrázek 9.5) je malá ukázka ze simulace chování analyzátoru. V tištěné verzi nebudou dobře vidět popisy, proto je simulace uložena na CD ve složce „Obrázky“. Při zaplnění paměti do poloviny procesor nastaví signál „procesor_RDY“ = 1 a je zahájeno odesílání dat po SPI. Vidět je vystavení následujících dat na výstup FIFO paměti při příchodu impulsu pro čtení a uložení dat do

paměti s příchodem impulsu pro zápis. Impulsy pro čtení a zápis jsou řízeny synchronizačními hodinami, kde jdou do log. 0 s náběžnou hranou těchto hodin a zároveň je provedena požadovaná akce.



Obrázek 9.5: Simulace funkce analyzátoru v FPGA

9.1.3.3 Využití zdrojů FPGA

Počet zabraných logických řezů (slices):	355 z 1792	19%
Počet zabraných klopných obvodů:	351 z 3584	9%
Počet zabraných 4vstupých LUT (lookup table):	252 z 3584	7%
Počet zabraných vstupních/výstupních bloků :	30 z 68	44%
Počet zabraných GCLK (globální hodinový blok):	5 z 24	20%
Počet zabraných DCM bloků:	1 ze 4	25%
Počet zabraných 18 Kb RAM bloků:	15 z 16	93%

9.1.3.4 Varování při překladu

Při překladu návrhu jsou vypsána 3 varování, všechna jsou pouze informativní a nezpůsobují žádný problém v návrhu.

- Nezapojené výstupní piny u DCM bloku – budou optimalizovány při překlad
- Použití bloku, který není reprezentován HDL kódem (FIFO paměť)

<http://www.xilinx.com/support/answers/9838.html>

- Nepřipojené piny pro paritní bity v paměti FIFO

<http://www.xilinx.com/support/answers/24846.html>

9.2 Část analyzátoru implementovaná v mikroprocesoru

Část logického analyzátoru implementovaného v mikroprocesoru slouží pro řízení vyčítání dat z FPGA, jejich poskládání do datagramu a odeslání přes UPD protokol na ethernet. Pro vytvoření této části logického analyzátoru v mikroprocesoru byly využity knihovny z několika projektů:

- libopencm3 – volné open-source firmwarové knihovny pro mikroprocesory zahrnující i STM32; LGPL licence
<https://github.com/libopencm3/libopencm3>
- HAL (Hardware Abstraction Layer) – HAL vrstva nad libopencm3 pro nastavování parametrů pro jednotlivé piny; LGPL licence
<https://github.com/BuFran/hal>
- lwIP – nezávislá implementace rodiny protokolů TCP/IP, která se navíc snaží redukovat využívání zdrojů; BSD licence
<http://www.nongnu.org/lwip/>
- XSVF Player – kód pro konfiguraci FPGA z mikroprocesoru, více informací je uvedeno v kapitole (9.2.2.2)
<https://secure.xilinx.com/webreg/clickthrough.do?cid=113970>

Vedoucím práce byly poskytnuty makefile soubory pro nastavení parametrů preprocesoru, kompilátoru a linkeru. Pro překlad programu byl využit překladač GCC spouštěný v prostředí Cygwin. Jako editor zdrojových kódů byl používán program „Code Blocks“, ze kterého je možné program nahrát do mikroprocesoru přes SWD rozhraní a zároveň je možné běžící program trasovat a ladit. Pro navázání na nižší vrstvu protokolů TCP/IP v projektu „lwIP“ byly poskytnuty také moduly „modnet“ a „eth_f417“, ve kterých jsou i funkce pro nastavení MAC a IP adres včetně inicializace ethernetového rozhraní mikroprocesoru.

Při popisu jednotlivých funkcí jsou v nadpisech opět uváděny jejich prototypy. Pojmem modul je opět myšlena dvojice souborů „modul.h“ a „modul.c“, které jsou uloženy v podadresářích projektu „include“ a „src“. Označení LED diod je uváděno podle názvu ve schématu.

9.2.1 Modul „funkce_Analyzator“

V tomto modulu jsou definována makra pro piny mikroprocesoru využívané v rámci implementace logického analyzátoru a vytvořeny funkce pro inicializaci využívaných rozhraní.

void JTAG_inicializace(void)

Nastaví piny propojené s JTAG rozhráním hradlového pole, aby jej bylo možné nakonfigurovat z mikroprocesoru (TMS, TDI, TCK – výstupní, TDO – vstupní).

void JTAG_odpojeni(void)

Nastaví všechny piny propojené s JTAG rozhráním hradlového pole jako výstupní (odpojení od JTAG), aby bylo možné hradlové pole konfigurovat z JTAG konektoru na desce.

void SPI3_inicializace(void)

Funkce pro nastavení SPI3 rozhraní mikroprocesoru. Nejprve jsou povoleny hodiny pro SPI3 a nastavena alternativní funkce odpovídající periférii SPI3. Poté jsou nastaveny parametry SPI komunikace:

- 16bit přenos
- klidová úroveň hodinového signálu na 0
- čtení na druhé hraně, tedy na sestupné (v FPGA data vystavována na náběžné)
- přenos od MSB bitu (nejvýznamnějšího)
- povolení softwarového výběru slave a jeho trvalý výběr
- nastavení jednosměrného přenosu
- nastavení pouze pro příjem dat
- nastavení SPI3 mikroprocesoru do slave módu

void PHY_INTRP_RST_inicializace(void)

Nastavení pinů spojených s PHY čipem, pinu pro přerušení (INTRP) jako vstupního a pinu pro reset PHY čipu (#RST) jako výstupního s hodnotou log. 1, aby nezůstal v resetu.

void rizeni_FPGA_inicializace (void)

Nastaví jako vstupní ty piny, na které přicházejí signály z FPGA informující o stavu paměti a probíhajícím převodu (pamet_pulka, pamet_prazdna, FPGA_prevod). Jako výstupní s hodnotou log. 0 nastaví pin s označením (procesor_RDY), který informuje hradlové pole o tom, že mikroprocesor je připraven přijímat data.

9.2.2 Funkce a globální proměnné v hlavní části programu (main.c)

V programu jsou vytvořeny tyto globální proměnné:

- citac – proměnná pro uchování stavu čítače
- SPI_data_buffer – pole o velikosti dvou bajtů s délkou 700 prvků (celková velikost je 1 400 bajtů) sloužící jako zásobník pro ukládání dat přijatých z SPI
- netif – struktura využívána pro síťová rozhraní vytvořená v projektu „lwIP“
- udp – struktura pro zajištění UDP spojení
- ipa – struktura, ve které je uložena IP adresa

void odeslani_Ethernet_SPI_data_buffer (uint16_t pocet_bajtu)

Odešle ze zásobníku „SPI_data_buffer“ na ethernet daný počet bajtů předaný funkci vstupním parametrem. Na ethernet je odeslán rámec s parametry nastavenými ve strukturách „udp“ a „ipa“, pomocí UDP protokolu na port 55 000.

uint16_t cteni_pameti_FPGA (void)

Funkce řídí vyčítání dat z paměti FPGA pomocí nastavování výstupního pinu mikroprocesoru „procesor_RDY“ a odesílá data přijatá z SPI na ethernet.

Na začátku je povoleno SPI rozhraní, nastaven signál „procesor_RDY“ do log. 1 a tím je zahájeno odesílání dat z FGPA. Přijímaná data z SPI jsou ukládána do zásobníku. Když je zásobník naplněn tak, že v něm zbývá poslední místo, je nastaven signál „procesor_RDY“ do log. 0 a deaktivováno³ SPI. Následně je zkontrolováno, jestli neprobíhá žádný přenos na SPI. Pokud ještě probíhá poslední přenos, je vyčkáno na jeho dokončení.

Po dokončení přenosu je signál z hradlového pole „FPGA_prevod“ = 0 a je zkontrolováno, jestli mikroprocesor stále čeká na dokončení probíhajícího přenosu. Pokud by tento stav někdy nastal (při testování nenastal), znamenalo by to chybu v zachycení některého z přenášených bitů. Veškerá dále přijímaná data by byla chybně zpracována a neodpovídala by měřeným průběhům, proto dojde k rozsvícení LED5 a program dále nepokračuje (modul by bylo potřeba restartovat). Pokud je vše v pořádku a byl přijat poslední vzorek, je uložen na zbývajících pozicích v zásobníku. Jestliže nebyla žádná informace přijata, pokračuje se dál s tím, že v zásobníku není zaplněno poslední místo.

Na závěr je naplněný zásobník odeslán na ethernet pomocí funkce „odeslani_Ethernet_SPI_data_buffer“ popsané výše, která jako vstupní parametr dostává hodnotu zaplnění zásobníku. Celý popsáný proces uvnitř funkce pro čtení dat je vytvořen v nekonečné smyčce, která je přerušena v případě, že byla paměť v hradlovém poli vyprázdněna („pamet_prazdna“ = 0). Pokud došlo k vyprázdnění paměti během vyčítání dat, je funkce před odesláním dat na ethernet ukončena a předává návratovou hodnotu s aktuální informací o zaplnění zásobníku.

void systick_inicializace(void)

Povolení systémového čítače, nastavení jeho parametrů tak, že se vygeneruje přerušení každou milisekundu a spuštění čítače.

void sys_tick_handler(void)

Obsluha přerušení od systémového čítače, kde s každým přerušením je zvyšována hodnota globální proměnné „citac“.

9.2.2.1 Funkce „main“

Nyní se konečně dostáváme k hlavní funkci „main“. Při spuštění programu je nejdříve nastavena frekvence pro sběrnice a systémové hodiny. Poté je nakonfigurován výstupní pin MCO2, který přivádí frekvenci 25 MHz do hradlového pole. Následuje inicializace

³ Po příkazu k deaktivaci SPI rozhraní, pokud je nastaveno v režimu pouze pro příjem a slave módu, dojde k jeho skutečné deaktivaci až po dokončení právě probíhajícího přenosu. [6]

všech prvků zásobníku na hodnoty 0. Po vyčištění zásobníku jsou povoleny hodiny ke všem GIPO portům mikroprocesoru a inicializovány tlačítka i LED diody.

Když jsou připraveny LED diody, rozsvítí se LED5 a jsou inicializovány piny tvořící JTAG rozhraní mezi FPGA a mikroprocesorem. Pomocí funkce „XSVF_predani_adresy()“ je předána adresa začátku umístění konfiguračního souboru pro FPGA ve flash paměti mikroprocesoru a zahájena konfigurace hradlového pole. Více o způsobu nakonfigurování FPGA z mikroprocesoru je v kapitole (9.2.2.2). Po dokončení konfigurace hradlového pole jsou piny mikroprocesoru pro JTAG rozhraní nastaveny jako vstupní, aby zůstala možnost konfigurovat hradlového pole z JTAG konektoru umístěného na desce. Po dokončení konfigurace rovněž zhasíná LED5.

Dalším krokem v programu je inicializace ethernetového rozhraní a nastavení parametrů pro UDP komunikaci, kde je mimo jiné nastaveno odesílání na IP adresu 192.168.1.60. Posledními inicializovanými částmi jsou piny se signály tvořícími propojení pro výměnu informací mezi hradlovým polem a mikroprocesorem, rozhraní SPI3 a systémový čítač. Nyní jsou již všechny části potřebné pro implementaci části logického analyzátoru připraveny a program vstupuje do nekonečné smyčky.

V nekonečné smyčce je nejprve vyčkáno na poloviční naplnění paměti v FPGA, tento stav je signalizován log. 1 na vstupu „pamet_pulka“. Po dosažení tohoto stavu je zavolána funkce na vyčítání dat z paměti „cteni_pameti_FPGA()“, která zároveň odesílá vyčtená data na ethernet, a ta běží, dokud není paměť prázdná. Když dojde k vyprázdnění paměti, funkce ve své návratové hodnotě předá informaci o počtu dat, kterými zaplnila zásobník. Pokud zásobník není prázdný (návratová hodnota není 0), dojde k odeslání části zásobníku, která byla zaplněna na ethernet. Po odeslání rámce se změní stav diody LED6 a přejde se k dalšímu provedení nekonečného cyklu. Na konci každého nekonečného cyklu je kontrolováno, jestli došlo při provádění kódu uvnitř smyčky k překročení času 5 sekund. Pokud ano změní se stav LED4, zavolá se funkce pro vyčištění ARP tabulky (aby nedošlo k pádu aplikace při přepojení do jiného zařízení) a vynuluje se globální čítač pro měření času, aby mohly být odměřovány další pětisekundové úseky. Pokud byla funkcí pro vyčítání dat předána hodnota nula, je spuštěn nový průchod nekonečného cyklu ještě před odesláním dat, tím se zabrání odeslání pouze prázdné hlavičky.

9.2.2.2 Nahraní konfigurace do FPGA

Hradlové pole typu FPGA je nutno vždy po připojení napájení nejprve nakonfigurovat. Aby nebylo potřeba k desce po zapnutí připojovat programátor, je hradlové pole automaticky konfigurováno přes JTAG rozhraní mikroprocesorem z jeho vnitřní flash paměti.

Pro konfiguraci hradlového pole je použit XSVF (Xilinx Serial Vector Format) soubor, což je binární podoba SVF (Serial Vector Format) souboru. SVF soubor je standardizovaný formát pro popis operací na JTAG rozhraní. Jeho binární forma

(XSVF) zabírá méně místa a je určena pro nahrávání do paměti vestavěných systémů (pro embedded aplikace). XSVF soubor je možné vygenerovat z prostředí iMPACT, což je nástroj firmy Xilinx pro konfiguraci hradlových polí. Tento soubor je následně uložen do flash paměti mikroprocesoru, odtud pomocí softwaru nazývaného jako „XSVF Player“ převeden na instrukce pro JTAG rozhraní a odesláním těchto instrukcí může být hradlové pole nakonfigurováno. Kód pro XSVF Player byl stažen ze stránek výrobce⁴, kde je po zaregistrování volně dostupný. V kódu jsou provedeny změny popsané v samostatné podkapitole níže.

Pro nahrání XSVF souboru do flash paměti mikroprocesoru ve vytvořeném projektu, se musí tento soubor jmenovat „Analyzator_XSVF.xsvf“ a musí být umístěn v kořenovém adresáři projektu, což je ve složce „Analyzator_mikroprocesor“ na CD. Při překládání kódu je z tohoto souboru vytvořen druhý soubor „Analyzator_XSVF.o“ v binární podobě, které rozumí kompilátor a následně je linkerem přidán do flash paměti mikroprocesoru. Když je konfigurační soubor uložen v paměti, stačí už jenom předat adresu začátku jeho umístění a spustit XSVF Player, který zajistí nakonfigurování FPGA přes JTAG rozhraní. Spuštění konfigurace se provede zavoláním funkce „xsvfExecute()“.

XSVF Player a jeho úpravy

Jak již bylo řečeno, soubory implementující XSVF Player byly staženy ze stránek firmy Xilinx. XSVF Player tvoří 3 moduly (6 souborů – .c, .h): „lenval“, „micro“, „ports“. Modul „lenval“ zůstal beze změny. V souboru „micro.c“ je vypnut debug mód, odstraněn kód s výpisem do konzole a jsou přidány chybějící prototypy u některých funkcí, soubor „micro.h“ zůstal nezměněn.

Největší změny oproti původnímu kódu musely být provedeny v modulu „ports“, kde jsou doplněny funkce manipulující s piny mikroprocesoru, tvořícími JTAG rozhraní s FPGA. Je přidáno zpoždění definované počtem cyklů a trvající minimálně 1 μ s pro zajištění požadavků na zpoždění (168 cyklů při 168 MHz). Poslední úpravou je vytvoření funkce „XSVF_predani_adresy()“, která zajistí předání začáteční adresy uložení konfiguračního souboru ve flash paměti, prototyp této funkce je přidán do hlavičkového souboru.

9.2.3 Shrnutí návrhu v mikroprocesoru

Část logického analyzátoru implementovaná v mikroprocesoru zajišťuje vyčítání dat z paměti v hradlovém poli a jejich odesílání na ethernet. Princip vyčítání dat je takový, že mikroprocesor čeká, dokud není zaplněna polovina paměti v FPGA. Když dojde signál informující o polovičním zaplnění paměti, je zahájeno vyčítání dat, které jsou ukládána do zásobníku (maximálně 1 400 bajtů). Při zaplnění zásobníku, je vytvořen

⁴ <https://secure.xilinx.com/webreg/clickthrough.do?cid=113970>

datagram a data jsou odeslána na ethernet. Vyčítání probíhá tak dlouho, dokud není paměť prázdná. Při vyprázdnění paměti opět dojde signál z FPGA informující o tomto stavu a na ethernet je odeslán zkrácený datagram podle aktuálního zaplnění zásobníku. Po vyprázdnění paměti a odeslání zkráceného datagramu se opět čeká na poloviční zaplnění paměti před dalším zahájením vyčítání dat. Nastavení parametrů pro komunikaci modulu je následující.

Parametry zdroje (modulu):

- IP – 192.168.1.50
- Masky podsítě – 255.255.255.0
- MAC – 02:00:00:00:00:00

Parametry pro cíl:

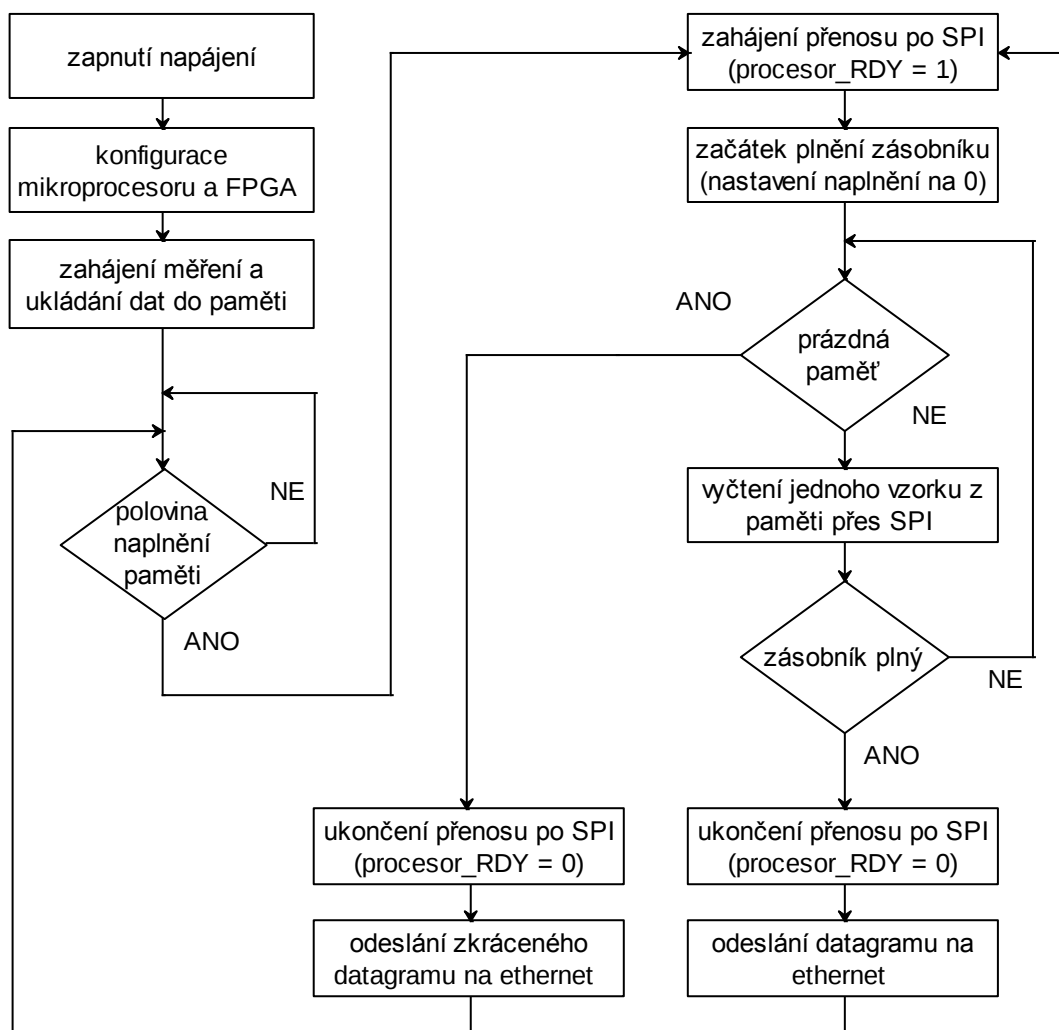
- IP – 192.168.1.60
- UDP port – 55 000

9.3 Stručné shrnutí implementace analyzátoru na desce

Shrnutí funkcí pro hradlové pole a zvlášť pro mikroprocesor bylo uvedeno v kapitolách (9.1.3 a 9.2.3). Hradlové pole generuje na výstupech jednoho budiče různé signály a druhý budič slouží jako vstup pro měření až 8 kanálů analyzátozem. Signály jsou vzorkovány s frekvencí 125 MHz. Maximální ověřená frekvence, se kterou se může měnit signál na jednom kanále, je 1 MHz, pro více kanálů jsou mezní hodnoty uvedeny v tabulce (Tabulka 9.1). Mikroprocesor slouží pro odeslání naměřených dat na ethernet přes UDP/IPv4 protokol. Rámce s daty jsou odesílány na IP adresu 192.168.1.60 a UDP port 55 000. Mikroprocesorem je z jeho vnitřní flash paměti po zapnutí napájení konfigurováno hradlové pole.

Během konfigurace hradlového pole po připojení napájení svítí LED5. Pokud je výstup z DCM bloku v hradlovém poli stabilní, svítí LED7 a část analyzátoru implementována v FPGA je funkční. Při každém vyčtení všech dat z paměti v FPGA mění svůj stav LED6. LED4 mění svůj stav minimálně po 5 s (závislé na vyčtení dat z paměti), společně s tím je čištěna ARP tabulka.

Chování desky s implementovaným firmwarem pro logický analyzátor od jejího připojení k napájení je uvedeno ve zjednodušeném vývojovém diagramu na obrázku (Obrázek 9.6). Označením paměť je myšlena vyrovnávací paměť v hradlovém poli a označením zásobník je myšleno globální pole v mikroprocesoru pro dočasné ukládání dat vyčtených přes SPI rozhraní.



Obrázek 9.6: Zjednodušený vývojový diagram analyzátoru

9.3.1.1 Aplikační vrstva UDP

Aplikační vrstva UDP protokolu je navržena tak, že jsou naměřená data primárně odesílána v datagramech s velikostí datového pole 1 400 bajtů. Zkrácený datagram je odeslán v případě, že došlo k vyčtení všech dat z paměti v hradlovém poli. Pokud by mikroprocesor před odesláním rámce stihl nastavit signál pro pozastavení vyčítání dat z FPGA dostatečně rychle (před zahájením dalšího přenosu na SPI), byly by běžně přenášeny datagramy o velikosti 1 398 bajtů. To se ovšem v implementovaném návrhu neděje a jsou odesílány převážně datagramy dlouhé právě 1 400 bajtů. V aplikační vrstvě jsou odesílána pouze data z měření, kde v 1 400 bajtech jsou přeneseny informace o 700 naměřených vzorcích.

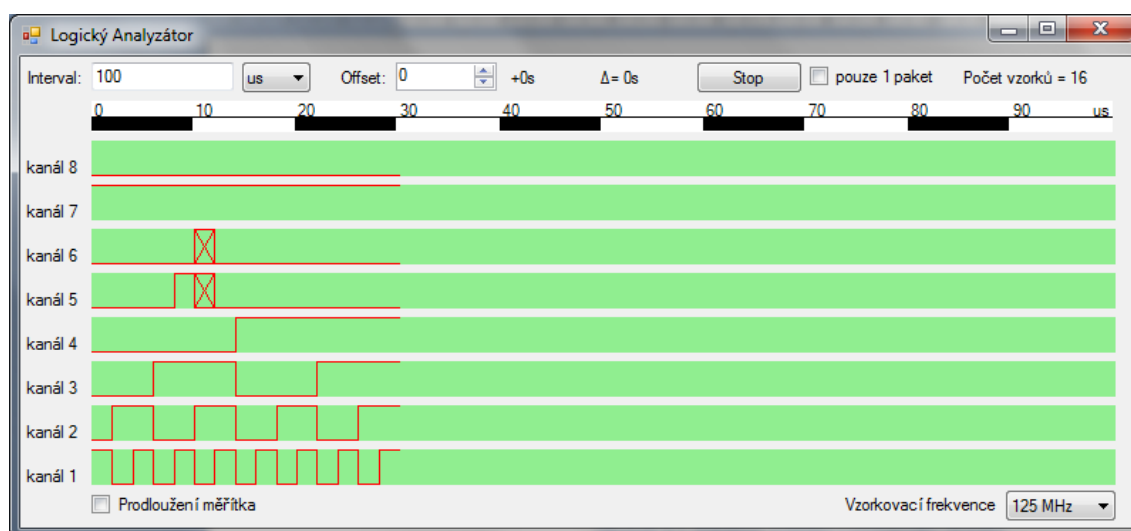
V datovém poli jsou postupně uloženy hodnoty z měřených průběhů, kde jsou informace o jednom vzorku zakódované ve dvou bajtech. Vlivem způsobu uložení dat v mikroprocesoru a stylem jejich odesílání je pořadí těch dvou bajtů prohozeno při

přijmutí aplikací v PC oproti jejich uložení v hradlovém poli. Posloupnost měřených vzorků zůstává stejná. Při přijmutí rámce je tedy v datovém poli UDP protokolu pro jeden vzorek nejprve uložen bajt obsahující časový údaj o trvání stavu na měřicích kanálech, který je přenesen v následujícím bajtu.

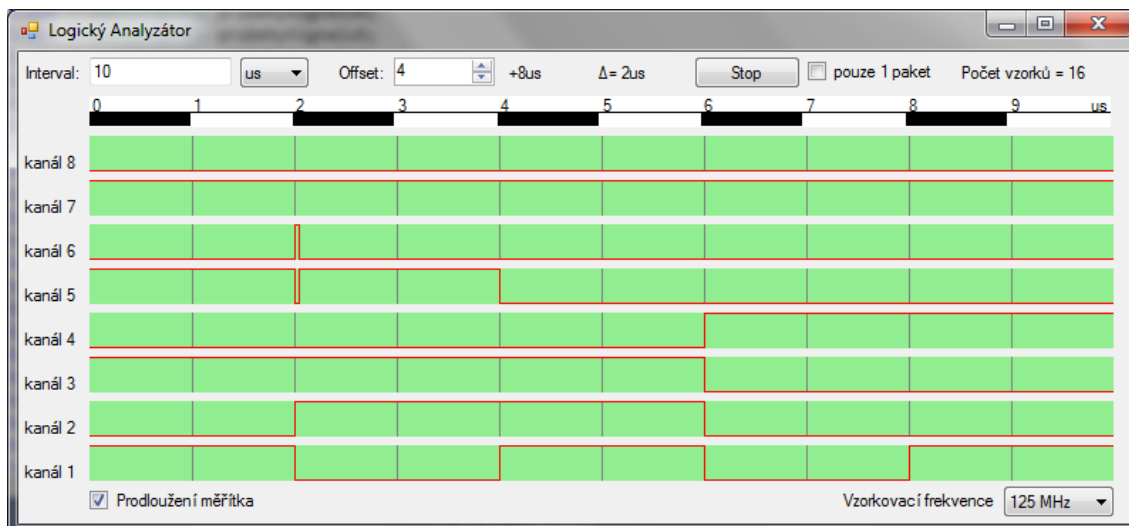
9.4 Zobrazující aplikace v PC

Pro možnost zobrazení měřených dat logickým analyzátozem implementovaným na desce je vytvořena aplikace pro PC. Tato aplikace je napsána v jazyce C# a graficky zobrazuje naměřené průběhy ze všech osmi měřicích kanálů. Pokud nepřišla žádná data, jsou aplikací zobrazovány předdefinované průběhy, pro demonstrování funkcí této aplikace, na obrázku (Obrázek 9.7) je možno vidět její vzhled. Proškrtnuté obdélníčky, viz. , začínající na 10 μ s v kanálech 5 a 6 znamenají, že v této oblasti by v jednom pixelu muselo být zobrazeno více hran signálu. Po zmenšení zobrazovaného intervalu (roztážení signálu) se průběhy překreslí tak, aby byly vidět všechny tyto hrany, toto roztážení je vidět na dalším obrázku (Obrázek 9.8) spolu se zapnutím pomocného zobrazení měřítka. Rovněž je na tomto obrázku nastaveno posunutí zobrazovaných průběhů o 4 vzorky (8 μ s).

Aplikaci je možné roztahovat a zmenšovat s tím, že při roztážení do šířky se automaticky uzpůsobuje velikost komponent zobrazujících průběhy a komponenty s měřítkem. Rozměry a pozice u ostatních prvků na formuláři zůstávají nezměněny a prvky jsou přichyceny k hornímu levému rohu.



Obrázek 9.7: Zobrazovací aplikace po spuštění (pokud nepřicházejí data)



Obrázek 9.8: Přiblížení průběhu z předchozího obrázku

9.4.1 Zobrazení naměřených průběhů a předávání dat

Naměřená data jsou z vytvořené desky s implementovaným logickým analyzátořem odesílána po ethernetu pomocí UDP protokolu na portu 55 000, tento port musí být tedy při spuštění aplikace volný. V aplikaci je vytvořen časovač, který s periodou 1 ms kontroluje, jestli byly přijaty nějaké datagramy.

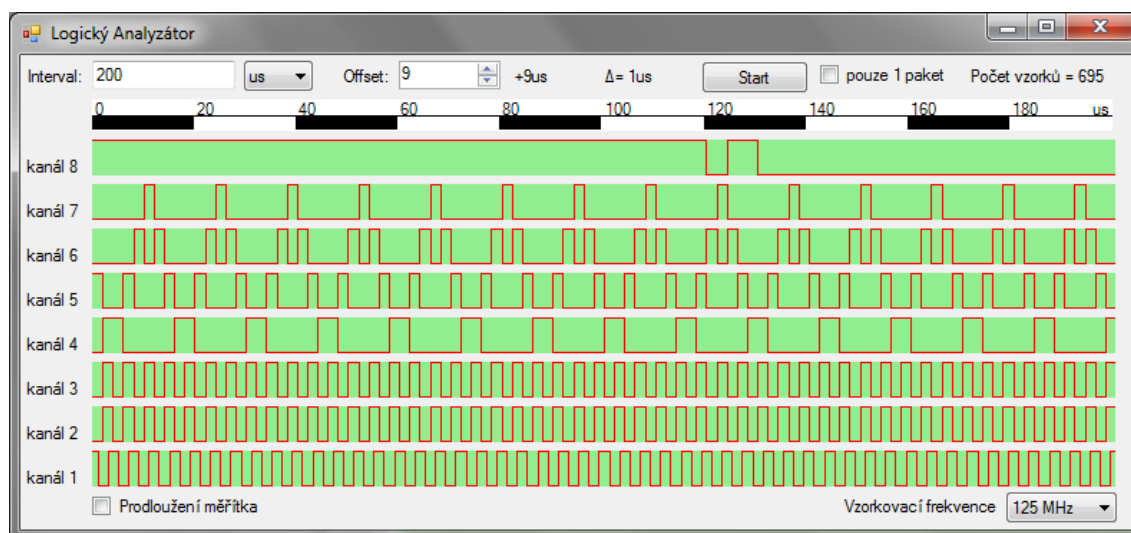
Pokud byl obdržen nějaký datagram, odřízne se z něj prvních 5 vzorků (10 bajtů), data jsou rozdělena do dvou polí a tyto pole jsou následně předána komponentám zobrazujícím průběhy signálů. Jedno pole obsahuje informaci o stavu, který byl na měřicích kanálech v okamžiku vzorkování, a ve druhém je uložen časový údaj o tom, po jakou dobu tento stav na vstupech trval. Pro každý přijatý datagram je vypsán počet vzorků (pravý horní roh), který byl z tohoto přenosu získán, a nastaveno omezení pro zadávání posunu po zobrazovaném signálu. Komponenty pro zobrazování jsou pojmenovány jako „PrubehySignalu“ a je vytvořeno jejich 8 instancí, kde každé instanci je zadán jeden kanál, který má zobrazovat.

Z datagramu je odříznuto prvních 5 vzorků (10 bajtů) z důvodu zajištění bezchybné interpretace měřených signálů při zobrazování jejich průběhů. Při přenosu přes rozhraní SPI totiž občas docházelo k chybné synchronizaci mikroprocesoru v okamžicích, kdy hradlové pole začalo generovat hodinový signál pro řízení SPI přenosu po nějaké pauze. Pauzou je myšleno setrvání hodinového signálu SPI (SCK) v klidové úrovni (log. 0) namísto neustálého generování pulsů v každém cyklu. Vzhledem k chybné synchronizaci SPI na začátku plnění datagramu a následném odříznutí vzorků na sebe signály v po sobě jdoucích datagramech přímo nenavazují, proto není řešeno ani jejich spojování pro zobrazování. Je tedy zobrazen vždy jen jeden datagram, a pokud nedošlo

k odeslání zkráceného datagramu po vyčtení všech dat z paměti v FPGA, je počet vzorků dostupných pro zobrazení 695.

Jaký počet vzorků musí být odstraněn, bylo testováno jednoduchou aplikací, kdy v hradlovém poli byl zapojen blok „generovani_znaku“ propojený přímo uvnitř FPGA a generoval dokola 15 znaků (A-O v binární podobě ASCII). Aplikace v PC potom přijímala odesílané datagramy a testovala, jestli obsahují tuto sekvenci znaků. Průběh testování vypisovala do konzole a zároveň ukládala do souboru. V případě správného pořadí znaků vypsala pouze za řadou teček číslo datagramu, v opačném případě vypsala ještě údaje navíc pro identifikaci chyby. Po spuštění se tato aplikace dá ukončit stiskem jakékoliv klávesy. Tato aplikace je umístěna na CD ve složce „UDP_prijem_test“.

Záznam průběhů z měření signálů vyvedených z výstupního budiče a přivedených na vstupní, je uveden na následujícím obrázku (Obrázek 9.9). Tím že jsou signály na všech 8 kanálech generovány v hradlovém poli a 7 z nich se mění současně, stačí dodržet limit pro maximální změnu s frekvencí udanou pro 2 kanály.

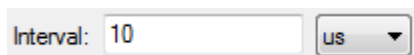


Obrázek 9.9: Záznam z měření signálů generovaných hradlovým polem

9.4.2 Ovládací prvky

Aplikace má několik ovládacích prvků, většina z nich je umístěna na vršku formuláře. Dole je pouze zaškrťovací políčko „Prodloužení měřítka“ a nastavení vzorkovací frekvence, která je v současném návrhu pevně nastavena na 125 MHz. Zaškrtnutí políčka „Prodloužení měřítka“ způsobí vykreslení šedých čar přímo do zobrazovaných průběhů na místech podle měřítka umístěného nad nimi. Pokud dojde k překrytí pomocných čar měřítka a průběhu, zůstane zobrazená červená čára průběhu signálu. V následujících kapitolkách budou popsány prvky pro ovládání umístěné ve vrchní části aplikace.

Nastavení intervalu zobrazení



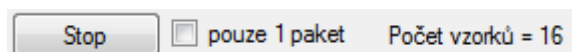
Do textového pole se zadává časový interval naměřeného průběhu, který má být aplikací zobrazen. K zadanému časovému údaji je možné vybrat jednotku (s, ms, μ s, ns, ps). Zadání hodnot je omezeno na interval $<1\,000\text{ s}; 10\text{ ps}>$, pro momentálně implementovanou verzi analyzátoru má význam zobrazení od 8 ns (minimální rozlišení) do $\approx 1,42\text{ ms}$ (maximální časový rozsah přenesený v jednom datagramu). Pokud není zadáno číslo nebo platný rozsah, text v poli zčervená a zůstane zobrazený průběh podle posledního platného zadání.

Nastavení posunutí (Offset)



Posunutí (Offset) se zadává v počtech naměřených vzorků. První údaj za zadáním hodnoty představuje přepočet na časový údaj o posunutí. Druhý údaj (Δ) představuje časový rozestup mezi současným a předchozím vzorkem při posunutí. Tento rozdíl je uváděn v jednotkách shodných s celkovým posunutím, s ohledem na aktuální implementaci však maximálně v mikrosekundách s přesností na tři desetinná místa pro zachování minimálního rozlišení 8 ns. Omezení pro maximální hodnotu posunu je nastaveno podle počtu vzorků přenášených v datagramu.

Manipulace s datagramy



Přestože je pro UDP komunikaci používán výraz „datagram“, je ve zobrazovací aplikaci použit výraz „paket“.

Pro řízení příjmu datagramů slouží tlačítko Start/Stop a zaškrťovací políčko za ním. Tlačítkem se pouští a zastavuje časovač v aplikaci a nápis na tlačítku se mění podle toho, jakou akci vykoná po stisknutí. Jestliže časovač běží (nápis „Stop“), tak po stisku je zastaven. Pokud neběží (nápis „Start“), bude spuštěn. Pokud je zatrhnuto zaškrťovací políčko („pouze 1 paket“), tak je po spuštění časovače jeho běh zastaven už po jednom průchodu. Poslední údaj v této části udává počet přijatých vzorků v právě zobrazovaném datagramu, podle této hodnoty je nastaveno omezení volby posunu.

10 ZÁVĚR

V diplomové práci jsou shrnuty informace pro úspěšný návrh a realizaci desky univerzálního modulu s připojením na ethernet, který slouží pro měření digitálních veličin. Kapitoly na začátku práce obsahují sepsanou teorii využitou při vytváření univerzálního modulu.

Prvním bodem zadání je návrh a realizace desky plošných spojů pro vytvoření univerzálního modulu schopného měřit digitální veličiny. Tento bod popisují kapitoly (5 a 6). Deska je navržena jako čtyřvrstvá v programu Eagle. Na desce je umístěno hradlové pole Spartan-3A (XC3S200A), které slouží pro měření a zpracování signálů. Mikroprocesor STM32F427 na desce slouží pro řízení a komunikaci přes MII rozhraní s PHY čipem. PHY čip je tedy propojen s mikroprocesorem a zajišťuje komunikaci vytvořené desky pomocí standardu Ethernet. Pro napájení desky je použit stejnosměrný zdroj s napětím 12 V. Pro zajištění napájení jednotlivých součástek jsou na desce umístěny dva spínané zdroje LM2595. Pro vstup měřených signálů do hradlového pole, případně pro výstup signálů, jsou na desce umístěny dva 8kanálové obousměrné budiče (74AHC245), které tvoří vstupní/výstupní brány pro měřené signály. Budiče chrání hradlové pole před poškozením v případě přivedení vyšších napětí a jsou osazeny v precizních DIL patičkách pro možnost jejich snadné výměny. Modul disponuje několika tlačítky a LED diodami pro snadnou a rychlou interakci s uživatelem. Navržená deska byla úspěšně vyrobena, ručně osazena a oživena.

Pro vytvořenou desku je vyrobena dvoudílná krabička. Krabička je navržena v programu SketchUp a vytištěna na 3D tiskárně. Ve spodním dílu krabičky jsou vytvořeny podpěrky na místech, kde dochází k největšímu namáhání desky. Tím je zátěž v místech manipulace s prvky na desce rozložena a plošný spoj je tak chráněn před mechanickým poškozením. Přímě do spodního dílu krabičky je deska přichycena šesti 2mm šroubky. Vrchní díl se dá na spodní nasadit a zasunout, při zasunutí dojde k lehkému zaklapnutí. Drážka pro zasunutí je vytvořena jenom krátká na konci obou dílů, aby nebylo potřeba zasunovat díly po celé jejich délce. V případě potřeby je možné oba díly sešroubovat 3mm šroubky, kde matičky jsou pevně zasazeny ve vrchním dílu krabičky. Pokud je krabička sesazena, zůstávají přístupné piny v zadní části desky. Mimo zůstávají i spínané zdroje, aby zbytečně neohřívaly vnitřek krabičky. Při větším zatížení zdrojů se k nim tak dá jednoduše připojit i chladič. Návrh krabičky je popsán v kapitole (7).

Vyrobená deska byla softwarově otestována pro zjištění případných chyb v propojení plošných spojů. Pro otestování jsou vytvořeny jednoduché programy nahrané do hradlového pole a do mikroprocesoru. Na všechny ovládané piny vedoucí ven z desky jsou přivedeny signály log. 1. Mikroprocesorem je otestováno i propojení s PHY čipem pomocí tzv. „NAND Tree“ testu. V závěrečné části testování zůstávají blikat diody LED4 a LED7. Zbylé 4 LED diody jsou ovládány 4 tlačítky. Detailnější

popis průběhu testování je uveden v kapitole (8.3). Otestování propojení na desce mezi všemi součástkami proběhlo úspěšně.

Další částí práce je naprogramování firmwaru, který implementuje funkci logického analyzátoru ve vytvořeném univerzálním modulu. Část implementována v hradlovém poli zajišťuje generování několika různých signálů na výstupu jednoho budiče a pro vstup měřených signálů do vytvořeného logického analyzátoru souží druhý budič. Vzorkovací frekvence při měření průběhů signálů je 125 MHz. Maximální frekvence, se kterou se může měnit vstupní signál při měření na jednom kanálu je 1 MHz, při současném využití více kanálů se tato frekvence zmenšuje. Analyzátor má 8 měřicích kanálů. Jeden vzorek ze vstupu je zakódován do 16bitového čísla společně s časovým údajem o době trvání neměnného stavu na vstupu a uložen do vyrovnávací paměti uvnitř FPGA. Vyrovnávací paměť je typu FIFO a je schopná uložit 16 384 vzorků, její velikost je tedy 256 Kb. V hradlovém poli je implementována jednotka pro vyčtení dat z paměti, jejich převod a odeslání na rozhraní SPI, kde hradlové pole funguje jako master pro komunikaci.

Kód v mikroprocesoru slouží pro příjem dat z paměti v hradlovém poli přes rozhraní SPI, jejich poskládání do UDP datagramu a odeslání na ethernet. Rámce jsou odesílány pomocí UDP/IPv4 protokolu na IP adresu 192.168.1.60 a na UDP port číslo 55 000. IP adresa modulu je nastavena na 192.168.1.50. Aplikační vrstva nad UDP protokolem je vytvořena tak, že jsou v ní postupně poskládány dva bajty pro každý naměřený vzorek. Jeden bajt obsahuje časový údaj, po který trval stav na měřicích kanálech uložený v druhém bajtu. Firmware pro univerzální modul je popsán zvlášť pro FPGA a mikroprocesor v kapitolách (9.1 a 9.2).

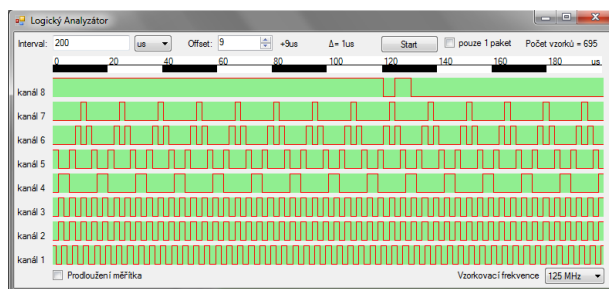
Poslední částí je vytvoření aplikace v jazyce C# pro grafické zobrazování průběhů zaznamenaných logickým analyzátozem implementovaným na desce. Aplikace přijímá UDP datagramy odesílané na port 55 000. Přijatá data jsou rozdělena do dvou polí, kde jedno obsahuje časové údaje a druhé stavy na měřicích kanálech. Tyto pole jsou předány komponentám pro zobrazování průběhů, kde každá komponenta vykresluje jeden z osmi měřených kanálů. Příjem dat se řídí tlačítkem Start/Stop. Při zobrazování dat z přijatého datagramu je vypsána informace, kolik vzorků může být zobrazeno. V aplikaci je možné nastavit časový interval, který má být zobrazen, a posun zobrazovaného průběhu o zvolený počet vzorků. Nastavený posun v počtech vzorků je následně přepočítán na časový údaj a vypsán, zároveň je vypsán i časový posun oproti předchozímu vzorku. Pro lepší odečet je možné zapnout grafické prodloužení měřítka umístěného nad zobrazovanými průběhy. Popis zobrazující aplikace je uveden v kapitole (9.4)

Ve vytvořeném univerzálním modulu je implementován firmware popsaného logického analyzátoru. Pro jeho využití k měření signálů stačí modul připojit k napájení, na měřicí vstupy přivést signály, propojit ho ethernetovým kabelem s PC a spustit zobrazovací aplikaci. Hradlové pole je po zapnutí napájení nakonfigurováno

mikroprocesorem z jeho flash paměti. Při konfiguraci FPGA svítí LED5 (prostřední dioda v rohu desky) a po jejím zhasnutí už v modulu běží kód logického analyzátoru. Na posledních dvou obrázcích je vidět výsledek práce, kde vlevo je univerzální modul připojený k notebooku a vpravo je spuštěna aplikace, která zobrazuje průběhy generované na výstupním budiči propojené na vstupní budič logického analyzátoru.



Obrázek 10.1: Modul s propojenými budiči, připojený k notebooku



Obrázek 10.2: Záznam z měření signálů generovaných hradlovým polem

Literatura

- [1] ASHENDEN, Peter J. *The VHDL Cookbook*. South Australia: University of Adelaide, 1990.
- [2] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. Praha: BEN - technická literatura, 2006, 349 s. ISBN 80-730-0198-5.
- [3] IEEE Std 802.3. *IEEE 802.3™: Ethernet*. New York: IEEE, 26. December 2008. Dostupné z: <https://standards.ieee.org/about/get/802/802.3.html>
- [4] XILINX. *Spartan-3A FPGA Family: Data Sheet*. [online]. 19 August 2010. [cit. 2015-01-08]. Dostupné z: http://www.xilinx.com/support/documentation/data_sheets/ds529.pdf
- [5] STMICROELECTRONICS. *STM32F427xx* [online]. April 2014 [cit. 2015-01-08]. Dostupné z: <http://www.farnell.com/datasheets/1818195.pdf>
- [6] STMICROELECTRONICS. *RM0090 Reference manual* [online]. March 2015 [cit. 2015-05-05]. Dostupné z: http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf
- [7] DIGILENT. *JTAG-HS2: Programming Cable for Xilinx FPGAs*. [online]. July 24, 2012. [cit. 2015-01-08]. Dostupné z: http://www.digilentinc.com/Data/Products/JTAG-HS2/JTAG-HS2_rm.pdf
- [8] MICREL. *KSZ8051MLL: 10BASE-T/100BASE-TX Physical Layer Transceiver* [online]. July 2010 [cit. 2015-01-08]. Dostupné z: <http://www.micrel.com/PDF/Ethernet/datasheets/ksz8051mll.pdf>
- [9] TEXAS INSTRUMENTS. *LM2595 SIMPLE SWITCHER* [online]. APRIL 2013 [cit. 2015-01-08]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm2595.pdf>
- [10] NXP SEMICONDUCTORS. *74AHC245; 74AHCT245 Octal bus transceiver; 3-state* [online]. 28 April 2009 [cit. 2015-04-05]. Dostupné z: http://www.nxp.com/documents/data_sheet/74AHC_AHCT245.pdf
- [11] Programovatelná logika I: Přehled PLD. In: *AbcLinuxu.cz* [online]. 18.12.2012 [cit. 2015-01-08]. Dostupné z: http://www.abclinuxu.cz/blog/digital_design/2012/12/programovatelná-logika-i-prehled-pld
- [12] Programovatelná logika II: FPGA. In: *AbcLinuxu.cz* [online]. 16.1.2013 [cit. 2015-01-08]. Dostupné z: http://www.abclinuxu.cz/blog/digital_design/2013/1/programovatelná-logika-ii-fpga
- [13] Serial Peripheral Interface (SPI). *Learn.sparkfun.com* [online]. [cit. 2015-01-08]. Dostupné z: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>
- [14] JTAG - A Technical Overview - TAP Signals and Instructions. *Xjtag.com* [online]. [cit. 2015-01-08]. Dostupné z: <http://www.xjtag.com/support-jtag/jtag-technical-guide.php>
- [15] *RFC 791: INTERNET PROTOCOL* [online]. September 1981 [cit. 2015-01-08]. Dostupné z: <http://tools.ietf.org/html/rfc791#section-2.1>
- [16] *RFC 768: User Datagram Protocol* [online]. 28 August 1980 [cit. 2013-05-20]. Dostupné z: <https://tools.ietf.org/html/rfc768>

- [17] OUTPUTLOGIC.COM. *OutputLogic.com* [online]. © 2009-2011 [cit. 2015-01-08]. Dostupné z: <http://outputlogic.com/>
- [18] Sériový port RS-232C. Root.cz: články [online]. 2008 [cit. 2015-01-08]. Dostupné z: <http://www.root.cz/clanky/seriovy-port-rs-232c/>
- [19] *Wikipedia, the free encyclopedia* [online]. [cit. 2015-01-08]. Dostupné z: <http://en.wikipedia.org/wiki/>
- [20] *Wikipedie, otevřená encyklopedie* [online]. [cit. 2015-01-08]. Dostupné z: <https://cs.wikipedia.org/wiki/>
- [21] USART. *Wikipedie, otevřená encyklopedie* [online]. 30. 5. 2012 [cit. 2013-01-03]. Dostupné z: <http://cs.wikipedia.org/wiki/USART>
- [22] Universal asynchronous receiver/transmitter. *Wikipedia, the free encyclopedia* [online]. 3 January 2013 [cit. 2013-01-03]. Dostupné z: http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter
- [23] Ethernet frame. In: *Wikipedia: the free encyclopedia* [online]. 13 May 2013 [cit. 2013-05-20]. Dostupné z: https://en.wikipedia.org/wiki/Ethernet_frame#cite_note-1
- [24] Cyclic redundancy check. In: *Wikipedia: the free encyclopedia* [online]. 17 May 2013 [cit. 2013-05-20]. Dostupné z: http://en.wikipedia.org/wiki/Cyclic_redundancy_check
- [25] IPv4 header checksum. In: *Wikipedia: the free encyclopedia* [online]. 7 February 2013 [cit. 2013-05-20]. Dostupné z: https://en.wikipedia.org/wiki/IPv4_header_checksum
- [26] Media Independent Interface. In: *Wikipedia: the free encyclopedia* [online]. 17 April 2013 [cit. 2013-05-20]. Dostupné z: http://en.wikipedia.org/wiki/Media_Independent_Interface
- [27] Black Magic Probe. Black Magic Probe [online]. 2015 [cit. 2015-05-14]. Dostupné z: <http://www.blackspheer.co.nz/main/blackmagic>
- [28] AMPHENOL. RJ45 R/A DIP w/LED [online]. 2014 Jun 30 [cit. 2015-05-14]. Dostupné z: <http://www.tme.eu/cz/Document/8117c2b2a705c42e7c945f12ec20f30f/LMJTAB881243ML.PDF>
- [29] XILINX. LogiCORE IP FIFO Generator v9.3 [online]. 2012 [cit. 2015-05-15]. Dostupné z: http://www.xilinx.com/support/documentation/ip_documentation/fifo_generator/v9_3/pg057-fifo-generator.pdf
- [30] STMICROELECTRONICS. 2011. SLVU2.8: Low voltage unit for Gigabit Ethernet protection [online]. [cit. 2015-05-15]. Dostupné z: <http://www.farnell.com/datasheets/1868797.pdf>

Seznam zkratek

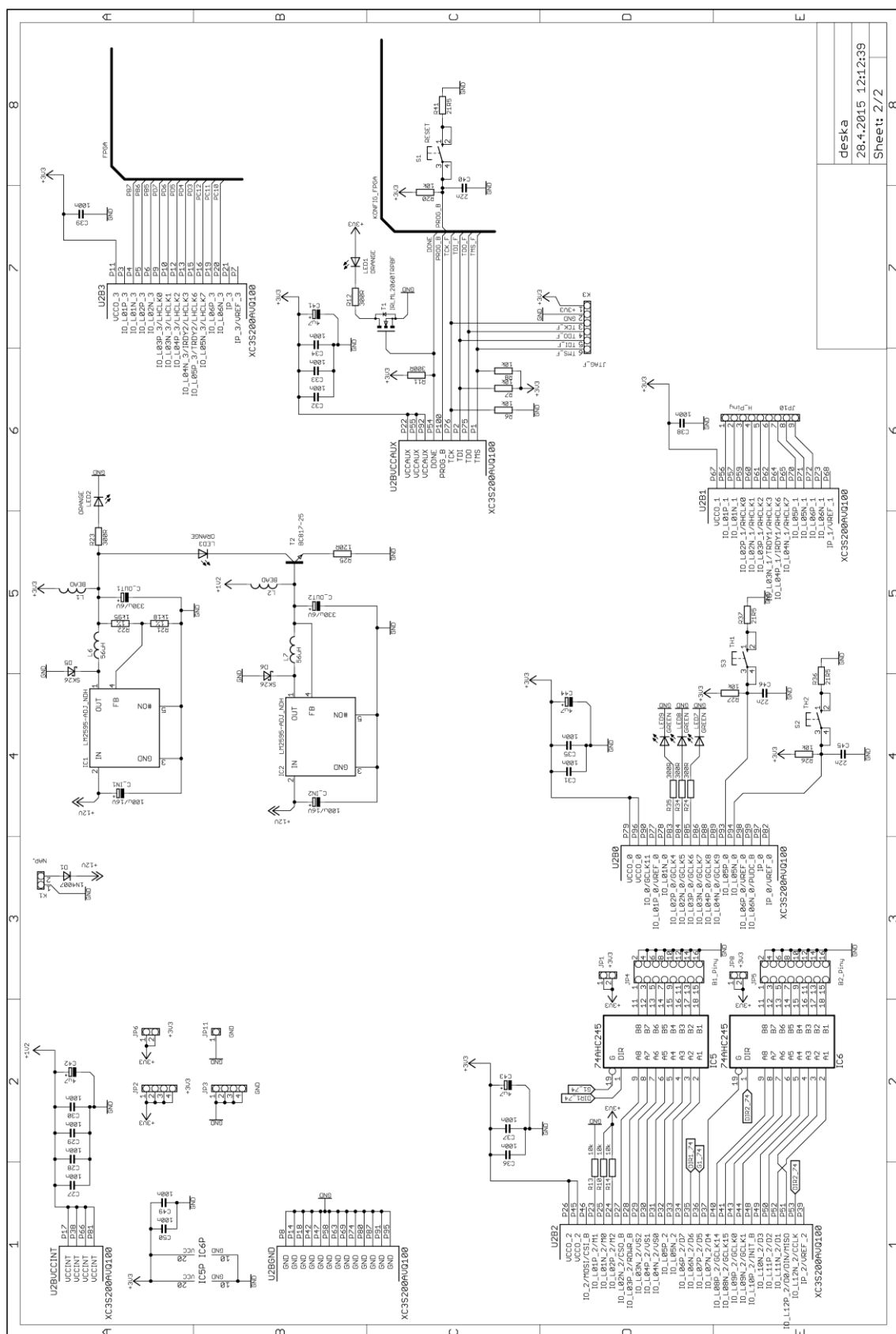
ASIC	Application Specific Integrated Circuits
PLD	Programmable Logic Device
SPLD	Simple Programmable Logic Device
CPLD	Complex Programmable Logic Device
FPGA	Field-Programmable Gate Array
DCM	Digital Clock Manager
CLB	Configurable Logic Block
LUT	Look-Up Table
IOB	Input-Output Blocks
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
IEEE	Institute of Electrical and Electronics Engineers
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface
JTAG	Joint Test Action Group
SFV	Serial Vector Format
XSFV	Xilinx Serial Vector Format
XPE	Xilinx Power Estimator
MII	Media Independent Interface
MAC	Media Access Control
PHY	Physical Layer
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense with Multiple Access and Collision Detection
RFC	Request For Comments
IPv4	Internet Protocol version 4
UDP	User Datagram Protocol
ARP	Address Resolution Protocol
MSB	Most Significant Bit
LSB	Least Significant Bit
SWD	Serial Wire debugging
LVC MOS	Low Voltage Complementary Metal Oxide Semiconductor
AHC	Advanced High-Speed CMOS
VQFP	Very Thin Quad Flat Pack

Seznam příloh

Příloha 1. Schéma univerzálního modulu

Příloha 2. Seznam součástí

Příloha 3. Přiložené CD



Příloha 2. Seznam součástek

součástky	hodnota/typ	počet	pouzdro	poznámky
kondenzátory	22 pF	4	0603	neosazeny
	10 nF	5	0603	MLCC
	100 nF	30	0603	MLCC
	1 μ F	1	0805	MLCC
	2,2 μ F	5	0805	MLCC
	4,7 μ F	5	0805	MLCC
	100 μ F/16V	2	D	tantalové
	330 μ F/6V	2	E	tantalové
rezistory	21,5 Ω	5	0603	
	120 Ω	1	0603	
	300 Ω	11	0603	
	430 Ω	1	0603	
	1 k Ω	1	0603	
	1,18 k Ω	1	0603	1%
	3,9 k Ω	2	0603	1%
	6,49 k Ω	1	0603	1%
	10 k Ω	19	0603	
diody	1N4007	1	SMA	
	SK26	2	SMB	
	LED	9	0805	
indukčnosti	56 μ H	2	MLC12xx	
	BEAD	3	1206	
tranzistory	IRLML2060TRPBF	1	SOT23	
	BC 817-25	1	SOT23	
krystaly	25 MHz	2	SMD	
tlačítka	B3F-10XX	5		
pinové lišty	2x9	1		
	2x8	2		
	1x2	3		
	1x5	1		
	1x6	1		
	1x15	1		
konektory	napájení 1x2_381	1		
	RJ45	1		LMJTAB881243M-L
Integrované obvody	74AHC245	2	DIL20	
	LM2595	2	TO220-5	
	SLVU2.8-4A1	1	SO08	
	KSZ8051MLL	1	LQFP48	
	STM32F427VIT6	1	LQFP100	
	XC3S200A	1	LQFP100	

Příloha 3. Přiložené CD

||\Elektronická verze práce
||\Firmware a software
||\Návrh plošného spoje
||\Obrázky
||\SketchUp 3D modely