



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra řídicí techniky

Programovatelná testovací platforma

Programmable test platform

Diplomová práce

Studijní program: Otevřená informatika
Studijní obor: Počítačové inženýrství
Vedoucí práce: Doc. Ing. Jiří Novák, Ph.D.

Jiří Blecha

Praha 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jiří Blecha**

Studijní program: Otevřená informatika (magisterský)

Obor: Počítačové inženýrství

Název tématu: **Programovatelná testovací platforma.**

Pokyny pro vypracování:

Navrhněte a implementujte technické vybavení programovatelného testeru pro automobilové aplikace. Navržené zařízení musí splňovat následující požadavky:

- podpora "libolného" počtu řadičů linkových protokolů CAN, LIN a FlexRay ve formě IP funkcí
- dvě fyzická rozhraní FlexRay
- tři fyzická rozhraní CAN
- dvě fyzická rozhraní LIN
- řídicí mikroprocesor s rozhraním Ethernet
- podpora vzdálené rekonfigurace hradlového pole FPGA
- implementace referenční konfigurace řadičů a jejich programové obsluhy

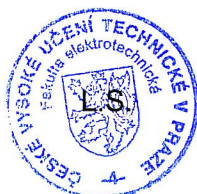
Seznam odborné literatury:

- [1] Lupini, C.A.: Vehicle Multiplex Communication, SAE International 2004, ISBN 0-7680-1218-X
- [2] Schäuffele, J., Zurawka, T.: Automotive Software Engineering, SAE International 2005, ISBN 0-7680-1490-5
- [3] Paták, M.: Methods for Testing the FlexRay Start-up Mechanism, diplomová práce ČVUT FEL, 2012

Vedoucí: doc.Ing. Jiří Novák, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 14. 1. 2014

Abstrakt

Tato diplomová práce se podílí na vývoji komplexního zařízení, které bude využito pro měření či testování na automobilových sběrnících CAN, LIN a FlexRay. Cílem této práce je zejména návrh hardwaru zařízení, který bude umožňovat rekonfiguraci pro specifické případy testování. Další část této práce je zaměřena na implementaci referenční konfigurace a programové obsluhy jednotlivých řadičů obsažených v jednočipovém mikroprocesoru.

Abstract

This thesis contributes to development of complex device which will be used for measuring or testing on automotive buses CAN, LIN, and FlexRay. The aim of this work is the design of device's hardware allowing reconfiguration for specific test cases. Another part of this work focuses on the implementation of a reference configuration and software library supporting the individual controllers embedded in a single-chip microprocessor.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 12. 5. 2014



Podpis

Poděkování

V první řadě bych chtěl poděkovat vedoucímu mé diplomové práce doc. Ing. Jiřímu Novákovi, Ph.D. za cenné rady a finanční zajištění, bez kterého by tento projekt nebylo možné realizovat. Ve druhé řadě děkuji Ing. Vítu Záhlavovi, CSc. a Ing. Petru Záleskému za čas věnovaný konzultacím. Ve třetí řadě bych chtěl poděkovat firmě Certuma spol. s.r.o. za neobyčejné obchodní jednání a za podporu při osazování plošného spoje. Dále pak firmě Texas Instruments za sponzorský dar ve formě vývojového kitu Hercules, který velmi usnadnil testování funkčnosti zařízení. Neobyčejné díky patří mé rodině za podporu a materiální zajištění v průběhu celého mého studia.

Obsah

1	ÚVOD	11
1.1	ZÁKLADNÍ PROBLEMATIKA.....	11
1.2	CÍLE PRÁCE.....	12
2	ROZBOR ZADÁNÍ	13
2.1	ZÁKLADNÍ POPIS KOMUNIKAČNÍCH TECHNOLOGIÍ.....	13
2.2	ANALÝZA POŽADOVANÉ FUNKČNOSTI.....	20
3	NÁVRH HARDWARU	22
3.1	MCU.....	22
3.2	PROGRAMOVÁNÍ/DEBUGGING MCU	24
3.3	FPGA	25
3.4	KONFIGURACE/DEBUGGING FPGA.....	26
3.5	EMIF (PROPOJENÍ MCU A FPGA)	28
3.6	RAM PAMĚŤ.....	30
3.7	FLASH PAMĚŤ	31
3.8	FYZICKÉ ROZHRANÍ ETHERNET	32
3.9	FYZICKÉ ROZHRANÍ RS-232	35
3.10	EXTERNÍ SYNCHRONIZACE.....	35
3.11	FYZICKÉ ROZHRANÍ CAN.....	36
3.12	FYZICKÉ ROZHRANÍ LIN.....	38
3.13	FYZICKÉ ROZHRANÍ FLEXRAY	39
3.14	ROZVOD HODIN.....	41
3.15	RESET	42
3.16	ZDROJ NAPÁJENÍ.....	44
3.17	NÁVRH DPS	52
3.18	VÝROBA.....	57
4	VÝVOJ FIRMWARU	60
4.1	VÝVOJOVÉ NÁSTROJE	60
4.2	CAN	60
4.3	LIN.....	63
4.4	ETHERNET	64
4.5	FLEXRAY	66
5	ZÁVĚR	67
	SEZNAM POUŽITÝCH ZKRATEK	70

LITERATURA	71
PŘÍLOHY	73

Seznam obrázků

OBR. 1.1 ZÁKLADNÍ KONCEPT KOMUNIKAČNÍ SÍTĚ V AUTOMOBILU.....	11
OBR. 2.1 VZTAH IEEE 802 A OSI MODELU.....	13
OBR. 2.2 FORMÁT ETHERNETOVÉHO MAC RÁMCE.....	14
OBR. 2.3 RS-232 ZNAK	15
OBR. 2.4 STRUKTURA SÍTĚ CAN DLE ISO 11898-2	16
OBR. 2.5 DATOVÝ RÁMEC CAN 2.0A.....	17
OBR. 2.6 FORMÁT RÁMCE PROTOKOLU LIN	17
OBR. 2.7 KOMUNIKAČNÍ CYKLUS FLEXRAY – ZDROJ (MALINSKÝ, 2010).....	19
OBR. 2.8 LINKOVÝ RÁMEC FLEXRAY – ZDROJ (MALINSKÝ, 2010).....	20
OBR. 2.9 BLOKOVÉ SCHÉMA VÝSLEDNÉHO ZAŘÍZENÍ.....	21
OBR. 3.1 BLOKOVÉ SCHÉMA TMS570LS3137 – ZDROJ (TI, TMS570LS3137 DATASHEET, 2013)	23
OBR. 3.2 MCU JTAG.....	24
OBR. 3.3 KONFIGURAČNÍ SCHÉMA PS.....	27
OBR. 3.4 VOLBA KONFIGURAČNÍHO SCHÉMATU.....	27
OBR. 3.5 FPGA JTAG	28
OBR. 3.6 PAMĚŤOVÝ PROSTOR MCU	28
OBR. 3.7 EMIF V TMS570LS3137 – ZDROJ (TI, TMS570LS3137 TECHNICAL REFERENCE MANUAL, 2013)	29
OBR. 3.8 ASYNCHRONNÍ EMIF – ZDROJ (TI, TMS570LS3137 TECHNICAL REFERENCE MANUAL, 2013)	30
OBR. 3.9 FLASH PAMĚŤ	32
OBR. 3.10 RMII A SMI ROZHRANÍ – ZDROJ (TI, TMS570LS3137 TECHNICAL REFERENCE MANUAL, 2013)	32
OBR. 3.11 DP83640 RMII MASTER – ZDROJ (TI, AN-1794 USING RMII MASTER MODE, 2013)	34
OBR. 3.12 SI-60062-F KONEKTOR.....	35
OBR. 3.13 EXTERNÍ SYNCHRONIZACE.....	36
OBR. 3.14 KONEKTOR A IMPEDANČNÍ PŘIZPŮSOBENÍ CAN	37
OBR. 3.15 KONEKTOR A VOLBA MASTER/SLAVE MÓDU LIN	39
OBR. 3.16 KONEKTOR A IMPEDANČNÍ PŘIZPŮSOBENÍ FLEXRAY.....	40
OBR. 3.17 ROZVOD HODIN	42
OBR. 3.18 MONITORING NAPĚTÍ A RESET.....	44
OBR. 3.19 PROUD CÍVKOU.....	48
OBR. 3.20 VÝSTUPNÍ NAPĚTÍ 1.2 V	50
OBR. 3.21 NAPĚŤOVÁ VĚTV 1.2 V.....	51
OBR. 3.22 NAPĚŤOVÉ VĚTVE 1.2 V, 2.5 V, 3.3 V A 5V	51
OBR. 3.23 VYVEDENÍ SPOJŮ Z FPGA (HORNÍ VRSTVA)	53
OBR. 3.24 PŘIPÁJENÉ FPGA.....	58
OBR. 3.25 OSAZENÁ DPS	59
OBR. 4.1 POVOLENÍ PERIFERIE CAN1	61
OBR. 4.2 GLOBÁLNÍ NASTAVENÍ PERIFERIE CAN1	61

OBR. 4.3 NASTAVENÍ "MESSAGE BUFFERŮ" PERIFERIE CAN1	62
OBR. 4.4 ZACHYCENÝ CAN RÁMEC NA OSCIOSKOPU	62
OBR. 4.5 ZACHYCENÝ LIN RÁMEC NA OSCIOSKOPU.....	64
OBR. 4.6 PROPUSTNOST FE PŘIPOJENÍ.....	65
OBR. 4.7 OVLÁDACÍ PC APLIKACE.....	66
OBR. 4.8 ZACHYCENÝ FLEXRAY RÁMEC NA OSCIOSKOPU.....	66

Seznam tabulek

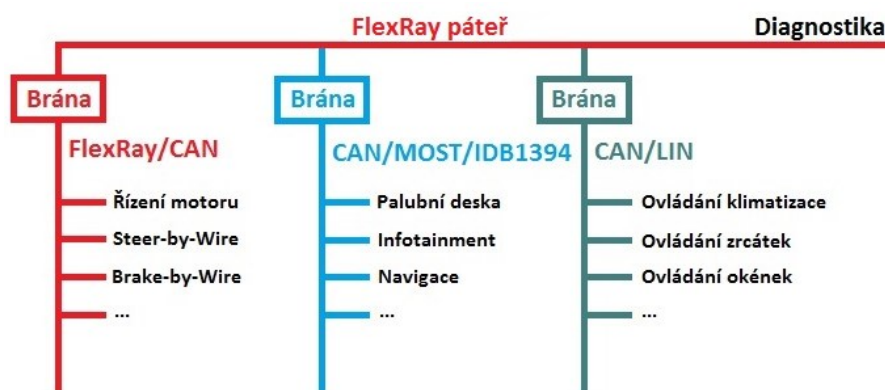
TAB. 2.1 NAPĚŤOVÉ ÚROVNĚ RS-232	15
TAB. 2.2 MAXIMÁLNÍ DÉLKA VEDENÍ RS-232.....	15
TAB. 2.3 POROVNÁNÍ CAN, LIN A FLEXRAY	20
TAB. 3.1 PARAMETRY TMS570LS3137_ZWT.....	24
TAB. 3.2 VELIKOSTI IP BLOKŮ V FPGA.....	25
TAB. 3.3 PARAMETRY EP4CE55F23C8N	26
TAB. 3.4 PARAMETRY R1WV6416RBG-5SI	31
TAB. 3.5 PARAMETRY DP83640.....	33
TAB. 3.6 ODHAD PROUDOVÉHO ODBĚRU	45
TAB. 3.7 OMEZENÍ SPÍNACÍ FREKVENCE LT3507A	46

1 Úvod

1.1 Základní problematika

Elektronika je v automobilech stále více používána. S pomocí elektroniky je např. možné zajistit vyšší bezpečnost jízdy, zjednodušit ovládání nebo snížit spotřebu paliva automobilu. Automobil je v podstatě heterogenní distribuovaný systém, který obsahuje mnoho různých řídicích jednotek. Protože každá řídicí jednotka plní jinou funkci (ABS, ESP x tempomat, parkovací asistent x klimatizace, ovládání zrcátek) jsou i požadavky na tyto jednotky a jejich připojení do distribuovaného systému velmi odlišné. Existuje velké množství automobilových komunikačních standardů, které tyto požadavky pokrývají (např. CAN, LIN, MOST, Byteflight, K-line, FlexRay, ...).

V dnešní době jsou hojně diskutované technologie x-by-wire (např. steer-by-wire, break-by-wire, ...), s nimiž jsou úzce spojeny standardy CAN a zejména FlexRay. Tyto standardy ovšem nejsou vhodné k připojování jednotek, které ovládají okénka nebo zrcátka. Tyto jednotky jsou obvykle připojovány pomocí standardu LIN, který je sice výrazně pomalejší než předchozí dva uvedené standardy, ale také je výrazně levnější. Pokud se povede ušetřit např. 100 Kč/automobil (zanedbatelná částka vzhledem k ceně automobilu) a vyrobí se např. 10000 automobilů, pak je celková ušetřená částka 1000000 Kč (řádově podobná částka ceně automobilu). Na obr. 1.1 je zobrazen základní koncept struktury komunikační sítě v automobilu včetně využívaných komunikačních standardů.



Obr. 1.1 Základní koncept komunikační sítě v automobilu

Komunikační standardy CAN a LIN patří mezi nejvíce používané. Komunikační standard FlexRay nemá dnes velké zastoupení, ale v budoucnu se jeho podíl bude pravděpodobně zvyšovat. V dnešní době již existuje několik sériově vyráběných

automobilů, které mají tento standard implementovaný. Jedná se o vozy od společností BMW, Audi, Mercedes či Rolls-Royce. Existuje dokonce i automobil, který již využívá technologii steer-by-wire (mechanické díly jsou stále přítomny, ale použity pouze v případě poruchy). Jiné vozidlové komunikační standardy zde nejsou uvažovány.

Tato práce vyplývá z bakalářských prací (Patač, 2010), (Zeman, 2012) a (Blecha, 2012). Dále pak z diplomové práce (Patač, 2012).

1.2 Cíle práce

Prvním cílem této diplomové práce je navrhnout hardware programovatelné testovací platformy, pomocí které bude možné provést měření (testy) na sběrnících standardů CAN, LIN a FlexRay. Důraz je kladen hlavně na možnost rekonfigurace zařízení a tím i jeho přizpůsobení specifické situaci.

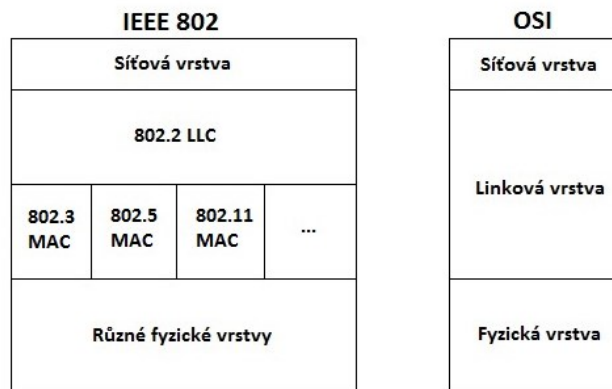
Druhým cílem této práce je implementace referenční konfigurace a programové obsluhy jednotlivých řadičů obsažených v jednočipovém mikroprocesoru (dále MCU), konkrétně řadičů Ethernet, CAN, LIN a FlexRay. Tento bod dále zahrnuje zprovoznění LwIP TCP/IP zásobníku nad řadičem technologie Ethernet.

2 Rozbor zadání

2.1 Základní popis komunikačních technologií

Ethernet

Ethernet byl vyvinut počátkem 70. let dvacátého století firmou Xerox. Varianta 10 Mb/s vznikla ve spolupráci s firmami DEC a Intel. V pozdějších letech byl Ethernet standardizován v rámci množiny standardů IEEE 802. Konkrétně standardy IEEE 802.2 a IEEE 802.3 definují podvrstvu linkové vrstvy OSI modelu LLC resp. MAC. Vztah IEEE 802 a OSI modelu je zobrazen na obr. 2.1.



Obr. 2.1 Vztah IEEE 802 a OSI modelu

Existuje mnoho variant, které se mohou lišit v přenosové rychlosti, fyzické topologii, či v typu obousměrného přenosu (poloviční/plný-duplex). Všechny varianty ovšem sdílí společnou přístupovou metodu CSMA/CD, což je stochastická přístupová metoda, díky které je přístup k přenosovému kanálu nedeterministický. Pravděpodobnost vzniku kolizí tedy není nulová a velmi závisí na vytíženosti sítě. Při malém vytížení sítě je pravděpodobnost vzniku kolizí nízká, tudíž je propustnost sítě vysoká. Naopak pokud je vytíženost sítě vyšší (>70 %), tak je pravděpodobnost vzniku kolizí vysoká a propustnost sítě začíná klesat. Jednotlivé varianty jsou standardizovány pod označením IEEE 802.3x. Klasický Ethernet nabízí tyto varianty fyzické vrstvy:

- 10Base-5 („tlustý“ koaxiální kabel s impedancí 50 Ω),
- 10Base-2 („tenký“ koaxiální kabel s impedancí 50 Ω),
- 10Base-T (symetrický kroucený pár kat. 3 s impedancí 100 Ω , plný-duplex),
- 10Base-FL (mnohavidové optické vlákno, plný-duplex).

Varianta nazývaná Fast Ethernet (dále FE) byla standardizována v roce 1995 pod označením IEEE 802.3u. FE nabízí vyšší přenosovou rychlost 100 Mb/s a v dnešní době je hojně používán. FE nabízí tyto varianty fyzické vrstvy:

- 100Base-TX (symetrický kroucený pár kat. 5 s impedancí 100 Ω , plný-duplex),
- 100Base-T2 (symetrický kroucený pár kat. 3 s impedancí 100 Ω),
- 100Base-T4 (symetrický kroucený pár kat. 3 s impedancí 100 Ω),
- 100Base-FX (mnohavidové/jednovidové optické vlákno, plný-duplex).

Dále existují i varianty Gigabitového a Desetigigabitového Ethernetu, které ovšem nejsou pro vyvíjené zařízení relevantní (standardně jsou dostupné MCU s FE řadičem) a proto je zde nemá smysl více rozvádět.

Na obr. 2.2 je zobrazen formát MAC rámce dle IEEE 802.3. Minimální množství dat, které je možné tímto rámcem přenést je 46 B. Pokud je dat méně, tak je nutné data doplnit neužitečnou informací (padding). Toto omezení je zavedeno kvůli použité přístupové metodě, aby bylo možné spolehlivě detekovat kolize. Maximální množství dat je omezeno na 1500 B.



Obr. 2.2 Formát ethernetového MAC rámce

- Preamble – 7 B (10101010),
- SD – Start Delimiter (10101011),
- CRC – generující polynom CCIT-32.

Pro více informací o technologii Ethernet viz skripta (Kocourek & Novák, 2006), dále potom standardy IEEE 802.3 a IEEE 802.2.

RS-232

RS-232 (nové označení EIA/TIA 232) je jedním z nejrozšířenějších sériových komunikačních rozhraní. Poprvé bylo představeno v roce 1962. Rozhraní umožňuje plný-duplex (obousměrný přenos). Standard definuje jak synchronní, tak asynchronní způsob komunikace. Další popis se bude týkat pouze varianty s asynchronním způsobem komunikace (nejčastěji implementované).

Formát znaku rozhraní RS-232 je uveden na obr. 2.3. Znak je započat start bitem, dále následuje 7-8 datových bitů, paritní bit a jako poslední následuje stop bit. Délka stop bitu může být 1,1.5 nebo 2 násobek délky bitu. Paritní bit umožňuje přenos ochranného kódu (sudá/lichá parita) a je volitelný.



Obr. 2.3 RS-232 znak

RS-232 využívá signalizaci uvedenou v tab. 2.1. Povolený napěťový rozsah je vždy pro přijímač větší než pro vysílač. Tento přístup a šířka definovaných napěťových intervalů z velké části eliminují vliv okolního rušení.

Úroveň	Vysílač	Přijímač
Log. 0 [V]	+5 až +15	+3 až +25
Log. 1 [V]	-5 až -15	-3 až -25
Nedefinované [V]	-3 až +3	

Tab. 2.1 Napěťové úrovně RS-232

V tab. 2.2 jsou uvedeny maximální délky vedení v závislosti na přenosové rychlosti. Je tedy vidět, že rozhraní RS-232 je použitelné i na delší vzdálenosti (nepoužívané, spíše RS-422/RS-485).

Přenosová rychlost [Bd]	Maximální délka [m]
19200	15
9600	150
4800	300
2400	900

Tab. 2.2 Maximální délka vedení RS-232

Rozhraní RS-232 již není v dnešní době standardně integrováno do osobních počítačů. Je nutné tedy využívat jinou nabízenou variantu, jakou je např. USB emulace + převodník napěťových úrovní.

Podrobnější informace lze nalézt ve skriptech (Kocourek & Novák, 2006).

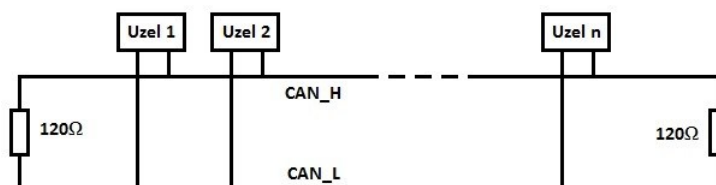
CAN

CAN je protokol, jehož vývoj začala v roce 1983 firma Bosch. Oficiálně byl publikován v roce 1986. Verze protokolu CAN 2.0 (CAN 2.0A, CAN 2.0B) byla publikována

v roce 1991. Protokol CAN, vzhledem k referenčnímu modelu OSI, leží na linkové vrstvě.

Aplikační protokoly jsou definovány pro specifické oblasti, např. pro oblast průmyslové automatizace (protokol CANopen).

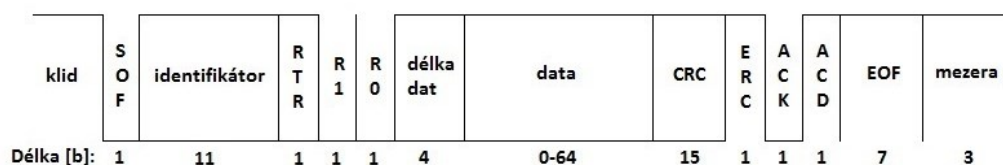
Základním požadavkem na fyzickou vrstvu je existence „drátového součinu“. Význam je takový, že existují 2 logické úrovně – dominantní a recesivní. Pokud všechny stanice na sběrnici vysílají recesivní úroveň, pak je i stav sběrnice recesivní. Pokud nějaká stanice (jedna či více) vysílá na sběrnici dominantní úroveň, pak je i stav sběrnice dominantní. Tento požadavek splňuje např. sběrnice se stanicemi, jejichž budiče jsou typu otevřený kolektor (dále OC). Druhým příkladem je optické vlákno – svítí x nesvítí. Nejčastěji se využívá fyzické vrstvy definované normou ISO 11898-2. Tato fyzická vrstva využívá jako vedení symetrický pár (CAN_H, CAN_L) s charakteristickou impedancí $120\ \Omega$. Logické úrovně jsou dány rozdílem napětí na vodičích CAN_H a CAN_L (diferenciální sběrnice), což má za důsledek vysokou odolnost proti souhlasnému rušení. Maximální přenosová rychlost je 1 Mb/s. V závislosti na této přenosové rychlosti je definována maximální délka sběrnice 40 m. Toto omezení délky vyplývá z rychlosti šíření signálu vedením, která je konečná. Pro nižší přenosové rychlosti může být délka sběrnice větší (100 kb/s – cca 600 m, 10 kb/s – cca 5 km). Nutná podmínka správné funkčnosti na sběrnici CAN je, že všechny aktivní stanice přijímají stejnou logickou hodnotu. Pokud by tato podmínka nebyla splněna, nebylo by možné korektně provést arbitraci, viz níže. Struktura sítě CAN dle ISO 11898-2 je zobrazena na obr. 2.4.



Obr. 2.4 Struktura sítě CAN dle ISO 11898-2

Protokol CAN využívá přístupovou metodu CSMA/CR, což je opět stochastická přístupová metoda. Přístup ke sdílenému médiu je v tomto případě řízen hodnotou identifikátoru (dále ID). CAN využívá adresace zpráv. Každý typ zprávy má přiřazen unikátní ID (CAN 2.0A – 11 b, CAN 2.0B – 11 b/29 b). Každá stanice vysílá určitou množinu zpráv. Průnik množin vysílaných zpráv přes všechny stanice musí být prázdná množina. Při splnění uvedených podmínek je jasné, že vždy po konci arbitrace bude

přenosový kanál přidělen právě jedné stanici. Přístup ke sdílenému kanálu tedy opět není deterministický! Formát datového rámce dle specifikace CAN 2.0A je uveden na obr. 2.5.



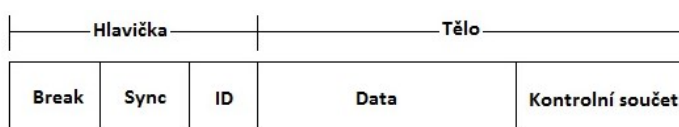
Obr. 2.5 Datový rámec CAN 2.0A

Podrobnější informace lze nalézt ve skriptech (Kocourek & Novák, 2006), dále pak v normách ISO 11898-x, kde x je 1,2, ..., 6.

LIN

LIN je protokol, který svým charakterem doplňuje spektrum protokolů používaných pro přenos dat v automobilech. Jeho vývoj započalo LIN Consortium koncem 90. let dvacátého století. První kompletní verze specifikace (LIN v1.3) byla publikována v roce 2002. Zhruba o rok později byla publikována další verze (LIN v2.0), která měla rozšířit možnosti protokolu. Aktuálně je nejnovější LIN v2.2A.

LIN využívá deterministickou přístupovou metodu Master/Slave. Opět je využito adresace zpráv, takže každá stanice má svou množinu zpráv, které vysílá. Každý typ zprávy je jednoznačně identifikován pomocí ID (6 b). Průnik množin vysílaných zpráv přes všechny stanice musí být prázdná množina. Vysílání vždy započne Master (vyšle hlavičku rámce) a je dokončeno stanicí, která je zodpovědná za vysílání zprávy s daným ID. Výhoda je, že vysílání je typu broadcast, tudíž data může přijímat více Slave zařízení zároveň. Formát rámce protokolu LIN je uveden na obr. 2.6. LIN má některé ID rezervované pro speciální účely. Jedním z případů je např. podpora událostmi generovaných zpráv (po příjmu rezervovaného ID stanice může vyslat událostí generovanou zprávu – nemusí čekat, až se Master dotáže na daný ID). Při tomto postupu může dojít ke kolizi (více stanic vysílá současně). V této situaci Master adresuje postupně všechny zprávy, které se skrývají pod použitým ID a zprávy jsou tedy vyčteny postupně.



Obr. 2.6 Formát rámce protokolu LIN

Fyzická vrstva protokolu LIN musí opět splňovat princip „drátového součinu“ – dominantní x recesivní úroveň. Konkrétně se využívá nesymetrické vedení a budičů typu OC. Výhodou je rozvod pouze jednoho vodiče. Přenosová rychlost je vcelku nízká, typicky 19.2 kb/s. Využívá se 12V signalizace. Implementace protokolu LIN je jednoduchá a levná. Lze využít standardní SCI/UART rozhraní dostupné v každém MCU.

Více informací lze nalézt ve specifikaci LIN protokolu viz (LIN Consortium, 2006). Aktuálně je poslední specifikace protokolu LIN (LIN v2.2A) v procesu přepisování do normy ISO 17987-x, kde x je 1, 2, ..., 7.

FlexRay

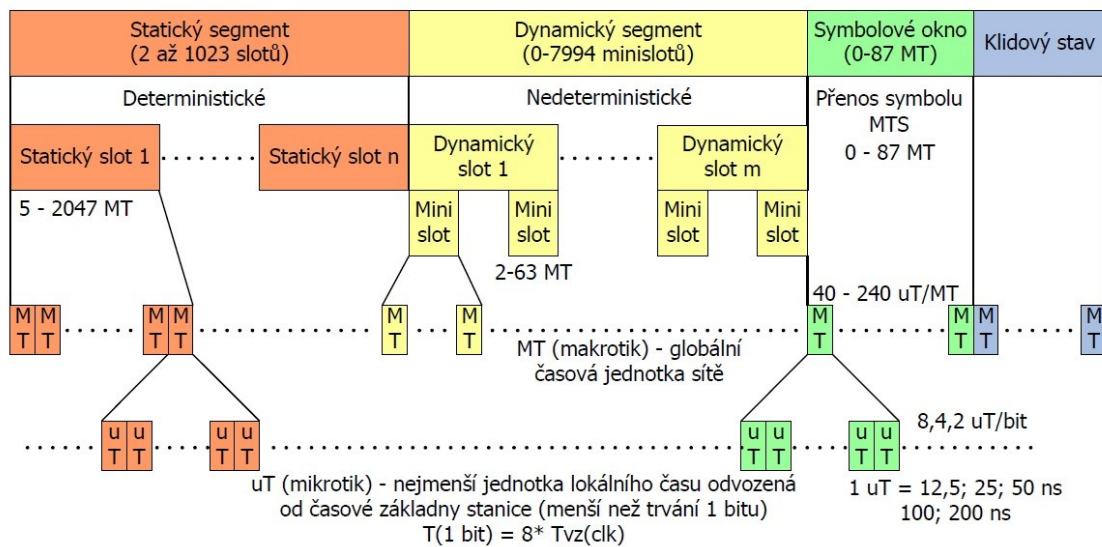
Standard FlexRay je nejnovější z uváděných standardů. Tento standard byl vyvinut v souvislosti s technologiemi x-by-wire, které se začínají postupně aplikovat. Svým charakterem jde o unikátní technologii, která opět má své právoplatné místo ve spektru automobilových komunikačních technologií. Vývoj tohoto standardu započalo v roce 2000 FlexRay Consortium. První specifikace standardu (FlexRay v2.0) byla publikována v roce 2004. Aktuální verze (FlexRay v2.1 Rev A) byla publikována v roce 2005. FlexRay Consortium bylo v roce 2009 rozpuštěno.

Fyzická vrstva standardu FlexRay je tvořena dvěma nezávislými přenosovými kanály (A, B). Každý kanál využívá k přenosu dat symetrický pár (BP, BM) o charakteristické impedanci 100 Ω . Logické úrovně jsou dány rozdílem napětí na vodičích BP a BM (diferenciální sběrnice), což má za důsledek odolnost proti souhlasnému rušení. Maximální přenosová rychlost je 10 Mb/s (pro jeden kanál). Existuje více možných sběrnicevých topologií:

- bod-bod (max. délka 24 m),
- pasivní sběrnicevá topologie (max. 22 stanic, max. délka 24 m),
- pasivní hvězda (max. 22 stanic, max. délka 24 m),
- aktivní hvězda (více portový obousměrný opakovač, max. délka mezi stanicí a hvězdou je 24 m, max. 2 hvězdy na cestě mezi stanicemi),
- hybridní topologie (kombinace výše uvedených topologií).

Přenos dat dle standardu FlexRay se rozděluje na komunikační cykly (dále KC). Každý KC je jednoznačně identifikován číslem 0-63 (6 b). Jeden KC se dělí na statický

segment, dynamický segment, symbolové okno a klidový stav – viz obr. 2.7. Statický a dynamický segment se dále dělí na statické resp. dynamické sloty. Dynamický slot se ještě dělí na minislots. Všechny tyto entity se dále dělí na celočíselný počet makrotiků (dále MT), což je nejmenší a pro všechny stanice stejná globální časová jednotka. Každý MT se dále dělí na celočíselný počet mikrotiků (dále uT), což je nejmenší lokální časová jednotka odvozená od časové základny stanice (12.5 ns, 25 ns, 50 ns). Dynamický segment a symbolové okno jsou nepovinné části KC.

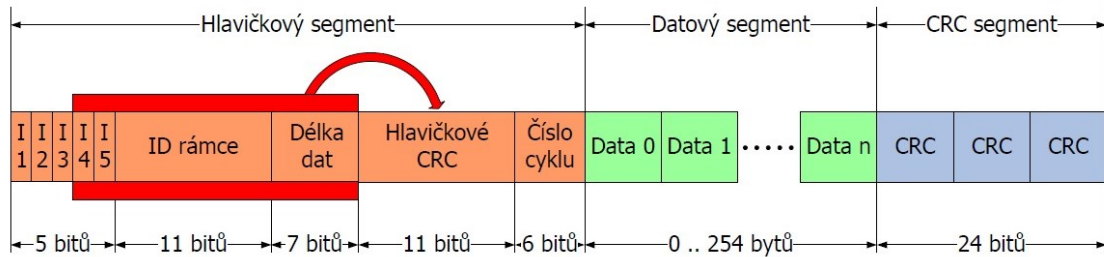


Obr. 2.7 Komunikační cyklus FlexRay – zdroj (Malinský, 2010)

Statický segment KC je povinný. Rozděluje se na konstantní počet statických slotů, které mají konstantní délku (nastaveno při navrhování sítě). Každá stanice na sběrnici má přidělený určitý počet statických slotů (statický slot je přidělen max. 1 stanici), ve kterých je jí umožněno vysílat (na obou kanálech stejný slot). V každém statickém slotu je vysílán jeden linkový rámeček, viz níže. Pokud stanice vysílá na obou kanálech, tak musí vysílat totožný rámeček na kanál A i B. Přístup ke sdílenému kanálu je deterministický (TDMA).

Dynamický segment je volitelný (lze nakonfigurovat síť bez dynamického segmentu). Dynamické sloty mohou mít různou délku (celistvý počet minislotsů) v závislosti na tom, zda chce stanice v daném dynamickém slotu vysílat nebo ne. Přenášená data mohou být různá pro oba kanály. Čas výskytu dynamického slotu může být tedy jiný pro oba kanály. Jelikož není jisté, že stanice dostane možnost vysílat v rámci dynamického segmentu (stanice, které mají přiděleny dynamické sloty s nižším číslem, mají prioritu), je tento přístup ke sdílenému kanálu nedeterministický.

Formát linkového rámce FlexRay je uveden na obr. 2.8.



Obr. 2.8 Linkový rámec FlexRay – zdroj (Malinský, 2010)

Více informací lze nalézt v disertační práci (Malinský, 2010), dále pak ve specifikaci FlexRay standardu viz (FlexRay Consortium, 2005).

Porovnání CAN, LIN a FlexRay je uvedeno v tab. 2.3.

Sběrnice	CAN	LIN	FlexRay
Rychlost	1 Mb/s	19.2 kb/s	10 Mb/s
Cena	\$\$	\$	\$\$\$
Vodiče	2	1	4(2)
Kanály	1	1	2
MAC	CSMA/CR	Master/Slave	TDMA
Použití	ABS, ESP, řízení, atd.	okénka, zrcátka, příslušenství, atd.	x-by-wire, tempomat, bezpečnost, atd.

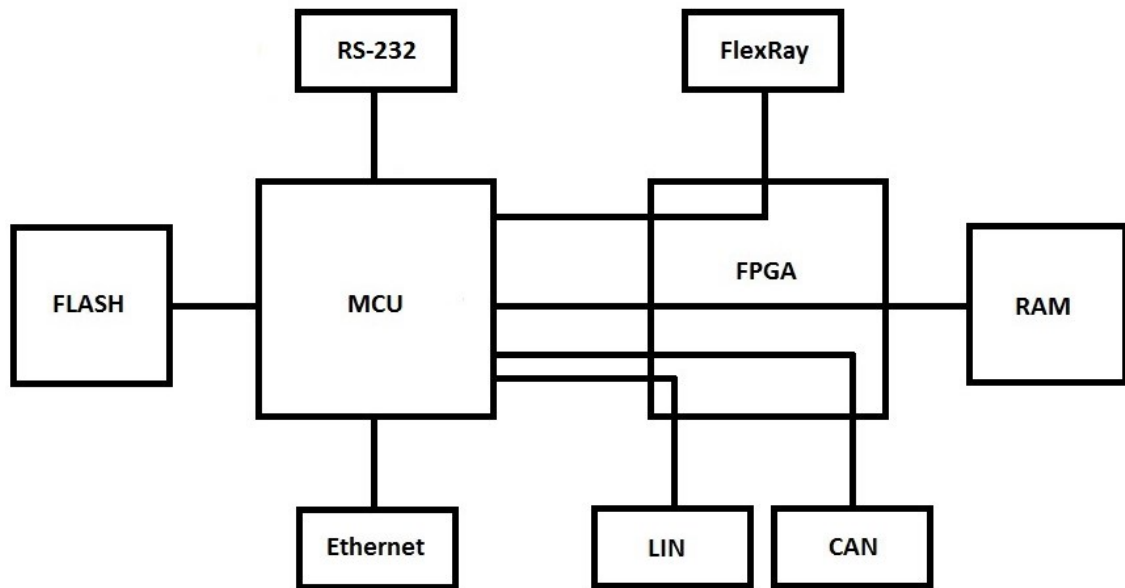
Tab. 2.3 Porovnání CAN, LIN a FlexRay

2.2 Analýza požadované funkčnosti

Zařízení má splňovat následující požadavky:

- podpora „libovolného“ počtu řadičů linkových protokolů CAN, LIN a FlexRay ve formě IP funkcí,
- 2 fyzická rozhraní FlexRay, 3 fyzická rozhraní CAN, 2 fyzická rozhraní LIN,
- řídicí mikroprocesor s rozhraním Ethernet,
- podpora vzdálené rekonfigurace hradlového pole FPGA,
- implementace referenční konfigurace řadičů a jejich programové obsluhy.

Blokové schéma výsledného zařízení je uvedeno na obr. 2.9.



Obr. 2.9 Blokové schéma výsledného zařízení

Zařízení tedy bude obsahovat:

- MCU s periferiemi Ethernet, CAN, LIN, FlexRay, UART/SCI, SPI, EMIF,
- FPGA od společnosti Altera dostatečně veliké pro obsazení až 5 CAN, 5 LIN a 3 FlexRay řadičů ve formě IP funkcí, dostatek pinů pro připojení alespoň 3 fyzických rozhraní CAN, 2 fyzických rozhraní LIN a 2 fyzických rozhraní FlexRay,
- RAM paměť pro rozšíření paměťového prostoru (např. buffer přijatých dat),
- FLASH paměť pro možnost uložení/načtení dat (např. uložení/načtení konfigurace FPGA),
- jednotlivá fyzická rozhraní Ethernet, RS-232 (připojená k MCU) a CAN, LIN, FlexRay (připojená k FPGA – umožnění připojení řadiče z MCU nebo z FPGA).

Popisem návrhu hardwaru a výběrem jednotlivých součástí se zabývá následující kapitola.

3 Návrh hardwaru

3.1 MCU

MCU je jednou z nejdůležitějších komponent v celém zařízení. Jsou na něj tudíž kladeny vysoké nároky. MCU musí obsahovat alespoň tyto periferie:

- řadič Ethernet – Fast/Gigabit Ethernet,
- řadič CAN – dle specifikace CAN 2.0,
- řadič LIN – alespoň dle specifikace LIN v2.0,
- řadič FlexRay – alespoň dle specifikace FlexRay v2.1,
- UART/SCI,
- SPI,
- EMIF.

Dalšími důležitými požadavky na MCU jsou dostatečný výpočetní výkon pro obsluhu výše uvedených periférií v reálném čase a dostatečně veliké paměti RAM a FLASH pro možnost vykonávání složitých aplikací resp. pro uložení počáteční konfigurace FPGA. Pro připojení signálů z budičů jednotlivých komunikačních technologií je nutné, aby MCU měl dostatek V/V pinů.

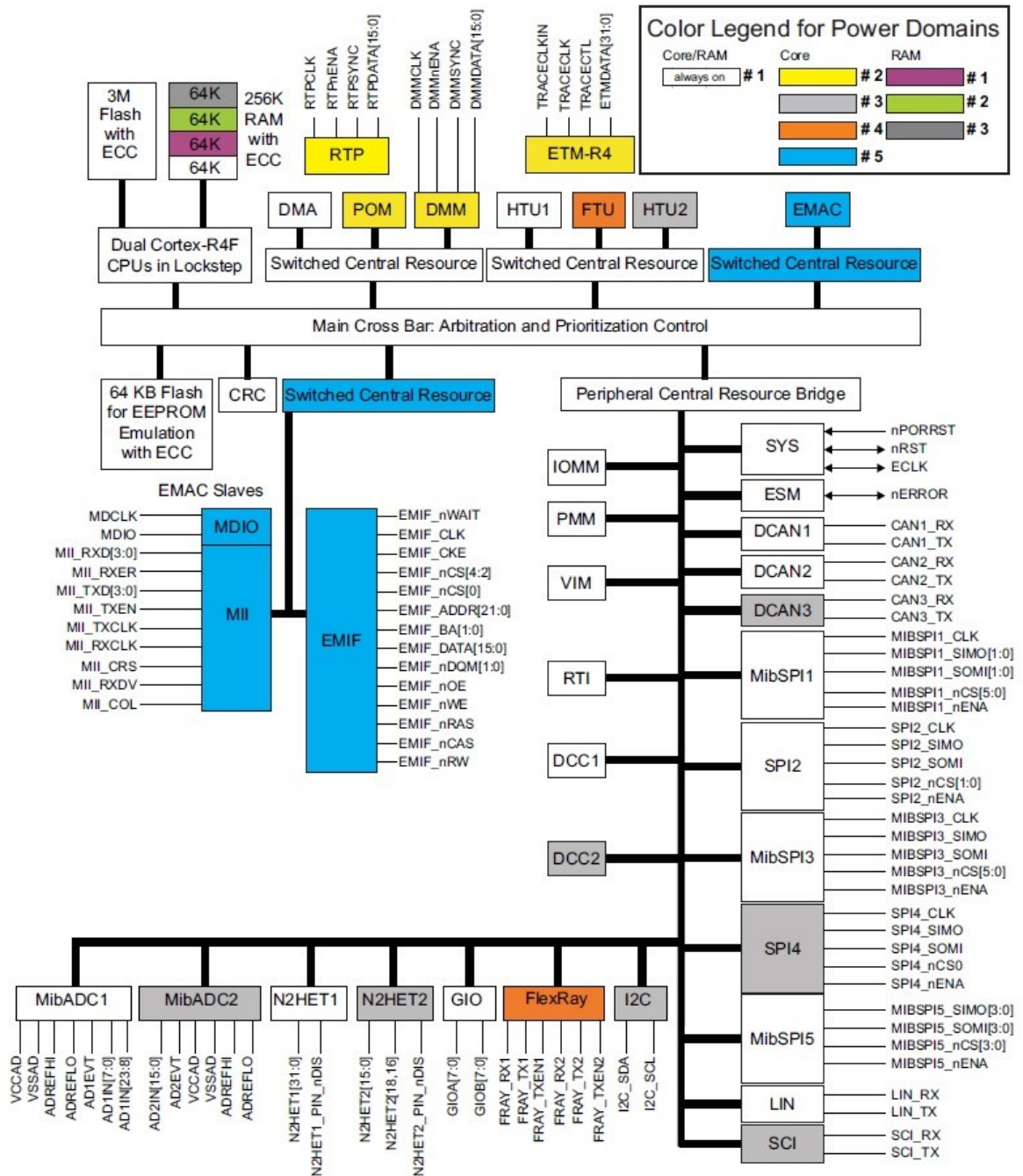
TMS570LS3137_ZWT

Tento MCU pochází ze sortimentu firmy Texas Instruments. Řadí se do rodiny vysoce výkonných, velmi bezpečných mikroprocesorů určených pro použití v automobilovém průmyslu. Bezpečnost je zajištěna níže uvedenou architekturou:

- 2 x CPU (o několik taktů vzájemně posunutá, zpracovávající stejný kód nad shodnými daty, porovnání výsledků),
- integrovaná logika pro CPU a paměťový BIST,
- ECC pro FLASH i RAM data,
- parita na pamětech periférií,
- zpětná vazba na V/V pinech.

TMS570LS3137 má vestavěné výkonné jádro ARM Cortex-R4F Floating-Point CPU, které nabízí relativní výpočetní výkon 1.66 DMIPS/MHz a může být taktováno až na 180 MHz. Maximální výpočetní výkon je tedy cca 298 DMIPS (výpočetní metrika

měřící výkonost CPU vykonávající celočíselné operace). MCU má vestavěné 3 MB FLASH a 256 kB RAM paměti s jednobitovou resp. dvoubitovou chybovou detekcí. Samozřejmostí je integrace požadovaných periférií. Blokové schéma MCU je uvedeno na obr. 3.1.



Obr. 3.1 Blokové schéma TMS570LS3137 – zdroj (TI, TMS570LS3137 Datasheet, 2013)

Tento MCU je k dispozici ve dvou variantách pouzdra – LQFP144 x NFBGA337. Vzhledem k tomu, že periférie EMIF je obsažena pouze ve variantě s BGA pouzdrům je nutné zvolit tuto variantu. Souhrn parametrů MCU relevantních pro tuto práci je uveden v tab. 3.1.

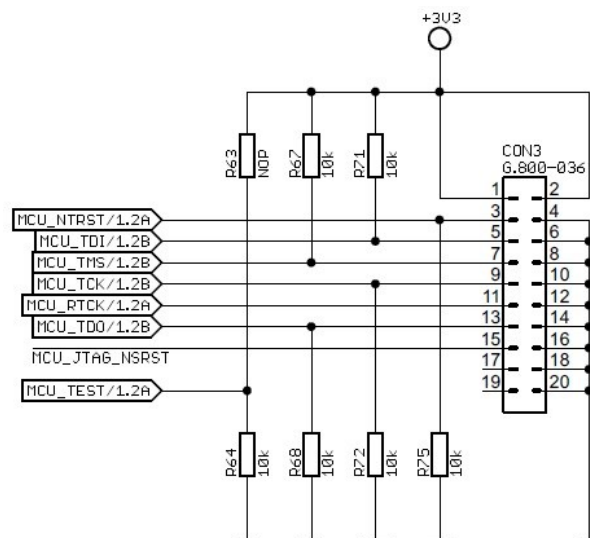
Maximální takt [MHz]	180
Vstupní hodiny [MHz]	5-20
FLASH [kB]	3072
RAM [kB]	256
Fast Ethernet	1
CAN 2.0 (A, B)	3
LIN v2.1	1
FlexRay v2.1	1
UART/SCI	1
SPI+MibSPI	5
EMIF (16-bit)	1
GPIO	120
Pouzdro	NFBGA337
Napětí jádra [V]	1.2
Napětí V/V [V]	3.3
Jádro	ARM Cortex-R4F

Tab. 3.1 Parametry TMS570LS3137_ZWT

Výběr tohoto MCU dále umocňuje jeho předchozí využití v bakalářské práci (Blecha, 2012). Pro více informací viz (TI, TMS570LS3137 Datasheet, 2013).

3.2 Programování/Debugging MCU

Pro programování/debugging tohoto MCU lze využít standardní ARM debuggery/emulátory. Vyzkoušené jsou TI XDS100v2 a SEGGER J-LINK. Připojení vývojového prostředí k MCU je realizováno JTAG konektorem. Zapojení konektoru je uvedeno na obr. 3.2. Pull-up/down rezistory jsou zde pravděpodobně nadbytečné, protože jsou integrovány uvnitř MCU.



Obr. 3.2 MCU JTAG

Signál MCU_JTAG_NS_RST je určený pro restart MCU a dále pro signalizaci, že je MCU v resetu. Připojení toho signálu bude popsáno v podkapitole 3.15 Reset.

3.3 FPGA

FPGA je také velmi důležitou komponentou obsaženou ve vyvíjeném zařízení. Vybrané FPGA musí splňovat následující požadavky:

- ze sortimentu společnosti Altera (již napsané IP funkce ve spojitosti s NIOS II procesorem, přenos na FPGA jiné firmy by byl složitý),
- dostatečně veliké pro obsažení až 5 CAN, 5 LIN a 3 FlexRay řadičů ve formě IP funkcí,
- dostatek pinů pro připojení alespoň 3 fyzických rozhraní CAN, 2 fyzických rozhraní LIN a 2 fyzických rozhraní FlexRay.

Společnost Altera nabízí základní 3 rodiny hradlových polí FPGA:

- Cyclone – nejnižší cena a výkon,
- Arria – střední řada,
- Stratix – nejvyšší cena a výkon.

Každá z těchto rodin se dále dělí na modelové řady. Vzhledem k cenovým kategoriím jednotlivých rodin a požadovanému výkonu byla zvolena rodina Cyclone.

Odhad velikosti (počet logických elementů) FPGA je odvozen z předpokládaného počtu řadičů a jejich velikosti. Velikosti řadičů LIN a FlexRay jsou převzaty z bakalářské práce (Paták, 2010) resp. z diplomové práce (Paták, 2012). Velikost řadiče CAN je předpokládána v rozmezí velikostí řadičů LIN a FlexRay.

Typ	Velikost [LE]	Počet	Velikost celkem [LE]
NIOS II procesor	~3300	1	~3300
CAN řadič	~3000	5	~15000
LIN řadič	~2700	5	~13500
FlexRay řadič	~9900	3	~29700
Součet			~61500

Tab. 3.2 Velikosti IP bloků v FPGA

Pro odhadovanou velikost 61500 LE jsou na trhu dostupná pouze FPGA v BGA pouzdrech. Aby mělo FPGA dostatek V/V pinů, tak bylo zvoleno pouzdro FBGA484.

EP4CE55F23C8N

Při zohlednění všech uvedených faktorů bylo nakonec vybráno FPGA z modelové řady Cyclone IV E, konkrétně EP4CE55F23C8N. Toto FPGA má „pouze“ 55856 LE, ovšem oproti EP4CE75F23C8N se 75408 LE (další velikost v řadě) je cca o 2000 Kč (cca 62 % ceny EP4CE55F23C8N) levnější. Uvedená FPGA mají stejná pouzdra (stejný vývod pinů), lze je tedy v případě nutnosti navzájem zaměnit.

Obraz konfigurace je pro toto FPGA ~1817.57 kB bez komprese dat. Komprese může velikost obrazu konfigurace zmenšit o 35 % (na ~1181.42 kB) až o 55 % (na ~817.91 kB). Obraz konfigurace tedy není problém uložit do FLASH paměti MCU.

Souhrn důležitých parametrů vybraného FPGA je uveden v tab. 3.3.

Počet LAB	3491
Počet LE	55856
RAM [kB]	292.5
Konfigurace [kB]	~1817.57
V/V piny	324
Pouzdro	FBGA484
Napětí jádro [V]	1.2
Napětí PLL [V]	2.5
Napětí V/V [V]	1.2, 1.5, 1.8, 2.5, 3, 3.3

Tab. 3.3 Parametry EP4CE55F23C8N

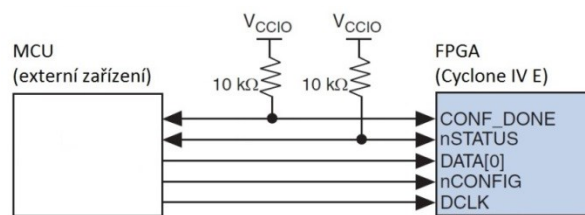
3.4 Konfigurace/Debugging FPGA

Konfiguraci neboli proces, při kterém se do FPGA nahrají konfigurační data a tím se zajistí funkčnost definovaná vytvořeným návrhem, je možné provést několika způsoby:

- Active serial (AS),
- Active parallel (AP),
- Passive serial (PS),
- Fast passive parallel (FPP),
- JTAG.

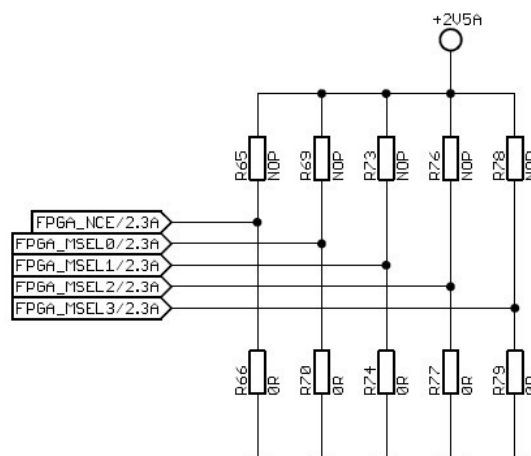
Výhoda konfiguračních schémat AP a FPP je paralelní přenos dat čili výrazně rychlejší proces konfigurace. Nevýhodou je větší složitost zapojení. V této aplikaci není čas konfigurace nijak kritický, takže použití těchto konfiguračních schémat nemá žádnou výhodu.

Druhou možností je použití konfiguračních schémat AS nebo PS, které využívají sériový přenos dat. Hlavním rozdílem mezi těmito konfiguračními schématy je to, kdo řídí proces konfigurace. Pro konfigurační schéma AS to je FPGA a pro konfigurační schéma PS to je externí zařízení (např. MCU, konfigurační prostředek) podílející se na procesu konfigurace. Protože je předpokládáno, že počáteční konfigurace zařízení bude uložena v nevolatilní paměti MCU a nahrávání nové konfigurace bude probíhat přes rozhraní Ethernet, konfigurační schéma PS je ideální volbou. Na obr. 3.3 je uvedeno doporučené schéma zapojení pro konfigurační schéma PS.



Obr. 3.3 Konfigurační schéma PS

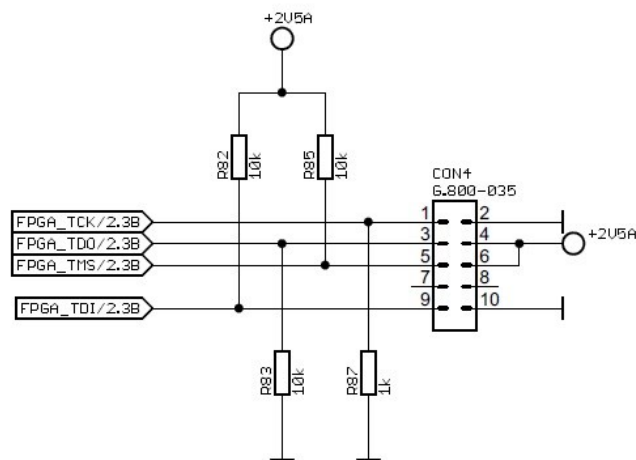
Nastavení konfiguračního schématu se provádí pomocí pinů MSEL[3:0]. Pro konfigurační schéma PS je MSEL[3:0] = 0000. Důležitým pinem je NCE, který je nutné nastavit do log 0, jinak FPGA nebude možné korektně nakonfigurovat. Na obr. 3.4 je nakresleno schéma zapojení diskutovaných pinů ve vyvíjeném zařízení.



Obr. 3.4 Volba konfiguračního schématu

Poslední možností konfigurace FPGA je konfigurační schéma JTAG. Toto konfigurační schéma má vyšší prioritu než všechna dříve zmíněná konfigurační schémata a nezávisí na nastavení pinů MSEL. Pokud tedy bude například probíhat konfigurace dle konfiguračního schématu PS a započne se konfigurace dle konfiguračního schématu JTAG, tak bude PS konfigurace přerušena a bude pokračovat pouze JTAG konfigurace. Na obr. 3.5 je uvedeno zapojení JTAG konektoru včetně potřebných pull-up/down

rezistorů. Konfigurační schéma JTAG je v tomto zařízení hlavně určeno pro vývoj. Lze využít např. vývojové prostředky USB-Blaster nebo ByteBlaster II.



Obr. 3.5 FPGA JTAG

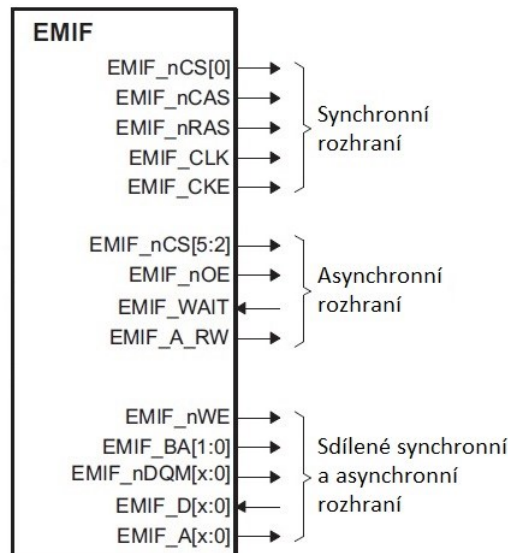
3.5 EMIF (propojení MCU a FPGA)

EMIF je rozhraní, které umožňuje připojení externích synchronních/asynchronních paměťových komponent a tím i rozšíření paměťového prostoru MCU. Paměťový prostor MCU je uveden na obr. 3.6.

0xFFFFFFFF	Systémové moduly
0xFFFF80000	-----
0xFF000000	Periferie - rámec 1
0xFE000000	CRC
0xFCFFFFFF	Rezervováno
0xFC000000	Periferie - rámec 2
0xF07FFFFF	Rezervováno
	Flash Modul Bus2 Rozhraní (Flash ECC, OTP a EEPROM přístupy)
0xF0000000	Rezervováno
0x87FFFFFF	Synchronní EMIF (128 MB)
0x80000000	Rezervováno
0x6FFFFFFF	Asynchronní EMIF (3 * 16 MB)
0x60000000	Rezervováno
0x202FFFFF	Flash (3 MB) (zrcadlený obraz)
0x20000000	Rezervováno
0x0843FFFF	RAM - ECC
0x08400000	Rezervováno
0x0803FFFF	RAM (256 kB)
0x08000000	Rezervováno
0x002FFFFF	Flash (3 MB)
0x00000000	

Obr. 3.6 Paměťový prostor MCU

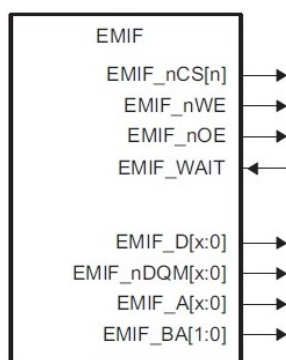
EMIF periférie vestavěná v MCU má strukturu zobrazenou na obr. 3.7. Podporuje tedy připojení jak synchronních (SDRAM), tak asynchronních (SRAM, NOR FLASH) paměťových komponent. EMIF je možné integrovat i do FPGA, jako IP blok. Tímto lze získat efektivní propojení MCU a FPGA paralelní sběrnici. Dále budeme uvažovat asynchronní variantu EMIF, protože je pro naše použití dostačující a je výrazně jednodušší.



Obr. 3.7 7 EMIF v TMS570LS3137 – zdroj (TI, TMS570LS3137 Technical Reference Manual, 2013)

Šířka datové sběrnice EMIF (EMIF_D[15:0]) je až 16 bitů a je možné využít až 22 adresových signálů (EMIF_A[21:0]). EMIF obsahuje 3 signály (EMIF_nCS[4:2]), které slouží k aktivaci připojených paměťových modulů. Samozřejmostí je obsažení vstupu (EMIF_WAIT), pomocí kterého mohou připojené paměťové moduly rozšířit (zpomalit) přístupový cyklus. Adresový signál EMIF_A[0] je nejnižším bitem adresy 32bitového slova. Pro adresaci 16bitového půlslova nebo bajtu lze využít signály EMIF_BA[1] resp. EMIF_BA[0]. Pokud je EMIF použit pro připojení modulu s 16bitovou architekturou, pak lze signál EMIF_BA[0] použít jako adresový signál EMIF_A[22]. Signály EMIF_nDQM[1:0] slouží pro maskování bajtů v čteném/zapisovaném 16bitovém půlslově. Poslední dva signály EMIF_nWE a EMIF_nOE jsou použity k signalizaci zápisu resp. k signalizaci čtení. Souhrn asynchronních signálů EMIF je uveden na obr. 3.8.

Všechny uvedené asynchronní signály EMIF jsou připojeny z MCU do FPGA (včetně EMIF_nCS[4:2] – možnost zvětšení adresovatelného prostoru/přímé připojení k RAM paměti skrz FPGA).



Obr. 3.8 Asynchronní EMIF – zdroj (TI, TMS570LS3137 Technical Reference Manual, 2013)

Více informací o EMIF lze nalézt v (TI, TMS570LS3137 Technical Reference Manual, 2013).

3.6 RAM paměť

RAM paměť má sloužit ve vyvíjeném zařízení pro rozšíření operačního paměťového prostoru FPGA/MCU. Potenciálním využitím je např. buffer naměřených dat, když z nespifikovaného důvodu nebude možné přenést data do PC.

Existuje několik variant připojení RAM paměti do vyvíjeného zařízení. První možnost dělení je dle použitého komunikačního rozhraní:

- sériové synchronní (např. SPI, I²C) – pomalé, jednoduché,
- paralelní synchronní (kompatibilní s EMIF) – rychlé, složité,
- paralelní asynchronní (kompatibilní s EMIF) – rychlé, středně složité.

Druhá možnost dělení je dle přípojného bodu:

- připojení k MCU – omezení přímého přístupu pouze na MCU,
- připojení k FPGA – přímý přístup z MCU i FPGA (zajištěno konfigurací FPGA),
- připojení mezi MCU a FPGA – přímý přístup z MCU i FPGA, složité (připojení na EMIF MCU, problém sdílené adresové sběrnice).

Protože je FPGA s MCU propojeno asynchronním EMIF, je optimální variantou připojit RAM paměť k FPGA pomocí paralelního asynchronního rozhraní kompatibilního s EMIF. Tímto omezením se výběr RAM paměti zužuje na statické RAM paměti (dále SRAM). Velká nevýhoda SRAM paměti je poměr cena/kapacita, který vychází řádově hůř než u dynamických paměti RAM (dále DRAM). Jelikož jsou signály

EMIF_nCS[4:2] propojeny z MCU do FPGA, vhodnou konfigurací FPGA lze zajistit přímé mapování RAM paměti do paměťového prostoru MCU.

Po provedené rešerši byla zvolena SRAM paměť R1WV6416RBG-5SI od firmy Renesas. Souhrn důležitých parametrů paměti je uveden v tab. 3.4.

Kapacita [kB]	8192
Organizace [bit]	4194304 x 16
Přístupový čas [ns]	55
Pouzdro	FBGA48
Napájecí napětí [V]	2.7 – 3.6

Tab. 3.4 Parametry R1WV6416RBG-5SI

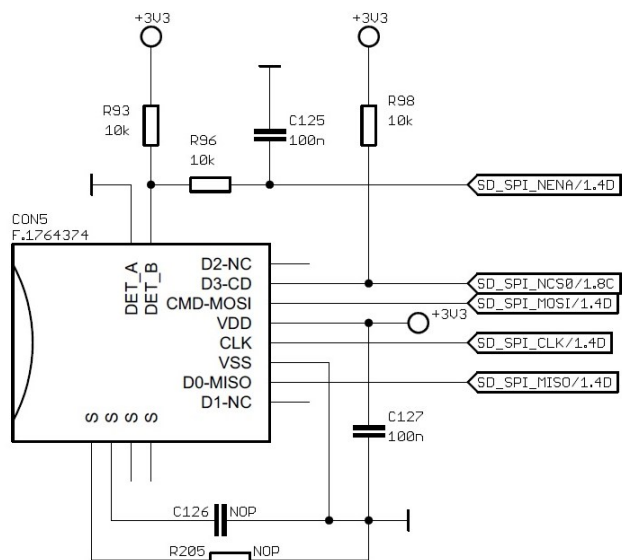
Pouzdro vybrané paměti je kompatibilní s paměťmi (např. AS6C3216) od firmy Alliance Memory. Vývod napájecích pinů je stejný. Jelikož je paměť připojena k FPGA, je možné tyto paměti vzájemně nahradit.

Podrobnější informace o R1WV6416RBG-5SI lze nalézt v (Renesas, 2009).

3.7 FLASH paměť

Externí FLASH paměť je opět určena k rozšíření paměťového prostoru MCU. Tentokrát se jedná o nevolatilní paměť, takže je primárně dedikována pro uložení např. konfiguračních (konfigurace FPGA, konfigurace testů) nebo naměřených (logy) dat. Jedno z možných použití je např. spuštění naplánovaných testů a logování naměřených dat bez připojeného PC.

Opět existuje několik variant připojení FLASH paměti do vyvíjeného vestavěného zařízení. Varianty byly diskutovány v minulé kapitole. Využití FLASH paměti prakticky neumožňuje její přímé připojení k FPGA (nebylo by možné konfigurovat FPGA z FLASH paměti). Bylo tedy zvoleno připojení k MCU. Požadavek na rychlost přístupu do FLASH paměti není nijak kritický, takže bylo zvoleno sériové komunikační rozhraní (SPI). Pro uvedené požadavky velmi dobře vychází použití přenositelné microSD FLASH paměti. Výhody využití tohoto typu paměti jsou jednak v kapacitě paměti, tak i v ceně. Další výhodou je přenositelnost. Této výhody lze dobře využít ve spojitosti s použitím systému souborů (je možné kartu používat ve vyvíjeném zařízení, tak i přímo v PC). Schéma zapojení microSD slotu je uvedeno na obr. 3.9.



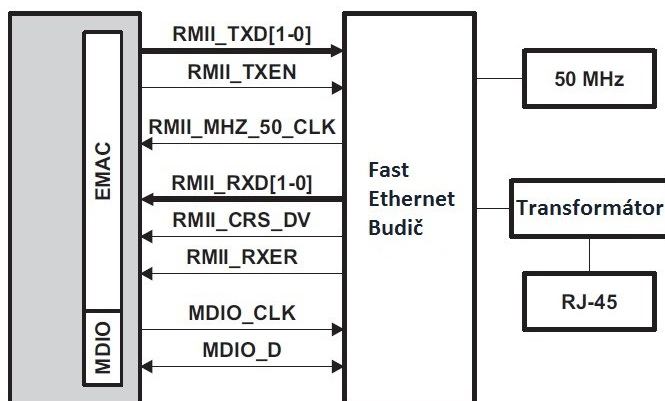
Obr. 3.9 FLASH paměť

3.8 Fyzické rozhraní Ethernet

Zvolený MCU obsahuje Fast Ethernet (FE) řadič. Pro připojení vyvíjeného zařízení k FE síti je nutné k řadiči v MCU ještě připojit fyzické rozhraní (budič a konektor), které zajistí správné kódování signálu (linkový kód 4B/5B, MLT-3), vyjednání typu linky (Autonegotiation protokol), synchronizaci, impedanční přizpůsobení atd. Pro datové propojení FE řadiče s budičem definuje norma IEEE 802.3u tyto rozhraní:

- MII – 15(16) signálů,
- RMII – 7(8) signálů.

Vzhledem k počtu potřebných signálů (složitost návrhu DPS) bylo zvoleno rozhraní RMII. Pro správu připojených budičů (může být více než 1) definuje norma IEEE 802.3u rozhraní SMI (podobné rozhraní I²C). Propojení FE řadiče s budičem je zobrazeno na obr. 3.10.



Obr. 3.10 RMII a SMI rozhraní – zdroj (TI, TMS570LS3137 Technical Reference Manual, 2013)

Využitím signálů RMII_TXD[1-0] jsou přenášena data z FE řadiče do budiče. Data jsou synchronizována na referenční taktovací signál RMII_MHZ_50_CLK a jsou platná pouze tehdy, když je aktivní signál RMII_TXEN. Pomocí signálů RMII_RXD[1-0] jsou přenášena data z budiče do FE řadiče. Data jsou opět synchronizována na referenční taktovací signál RMII_MHZ_50_CLK a jsou platná pouze tehdy, když je aktivní signál RMII_CRS_DV a neaktivní signál RMII_RXER. Signál RMII_CRS_DV je multiplexovaný signál detekce nosné (carrier sense) a platnosti přijatých dat (receive data valid). Signál RMII_RXER signalizuje detekci chyby v přijatém rámci. Poslední dva signály MDIO_CLK a MDIO_D tvoří již zmiňované SMI rozhraní. Jejich funkce jsou referenční taktovací signál pro přenos dat resp. přenos dat.

Pro více informací o RMII viz (TI, TMS570LS3137 Technical Reference Manual, 2013) a dále pak standard IEEE 802.3u.

Jediným požadavkem na výběr budiče je, aby měl integrované rozhraní RMII. Žádné další kritické požadavky nejsou uvažovány.

DP83640

V bakalářské práci (Blecha, 2012) je použit budič DP83640 od společnosti TI. Tento budič má integrované rozhraní RMII, je ozkoušený a zároveň je k dispozici v dostatečném množství. Nebylo tedy nutné hledat jinou variantu. Důležité parametry DP83640 jsou shrnuty v tab. 3.5.

MII	ANO
RMII rev. 1.2	ANO
SMI	ANO
Pouzdro	LQFP48
Napětí analog [V]	3.3
Napětí V/V [V]	2.5, 3.3

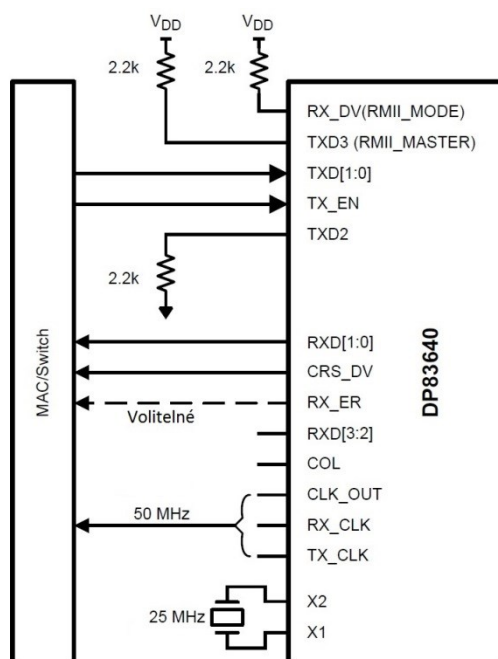
Tab. 3.5 Parametry DP83640

Nastavení (programování) budiče DP83640 je možné dvěma způsoby:

- SMI rozhraní – MCU spravuje připojené budiče pomocí sériové komunikace (nastavení vnitřních registrů budiče),
- „Strap options“ – po zapnutí (power-up) nebo po hard resetu (studený reset) jsou přečteny logické úrovně vybraných pinů a tím je budič uveden do specifického operačního módu. Po software resetu nejsou „strap options“ znovu vzorkovány!

Všechny piny, které mají význam pro „strap options“, mají definovanou defaultní hodnotu pomocí integrovaného pull-up/down rezistoru. Pro změnu této defaultní hodnoty je možné použít rezistor o hodnotě 2.2 k Ω a připojit ho na komplementární napěťový potenciál, než je připojen interní rezistor. Popis všech „strap options“ není cílem této práce. Budou zde zmíněny pouze ty, které jsou relevantní pro vyvíjené zařízení. Kompletní dokumentace je uvedena v (TI, DP83640 Datasheet, 2013).

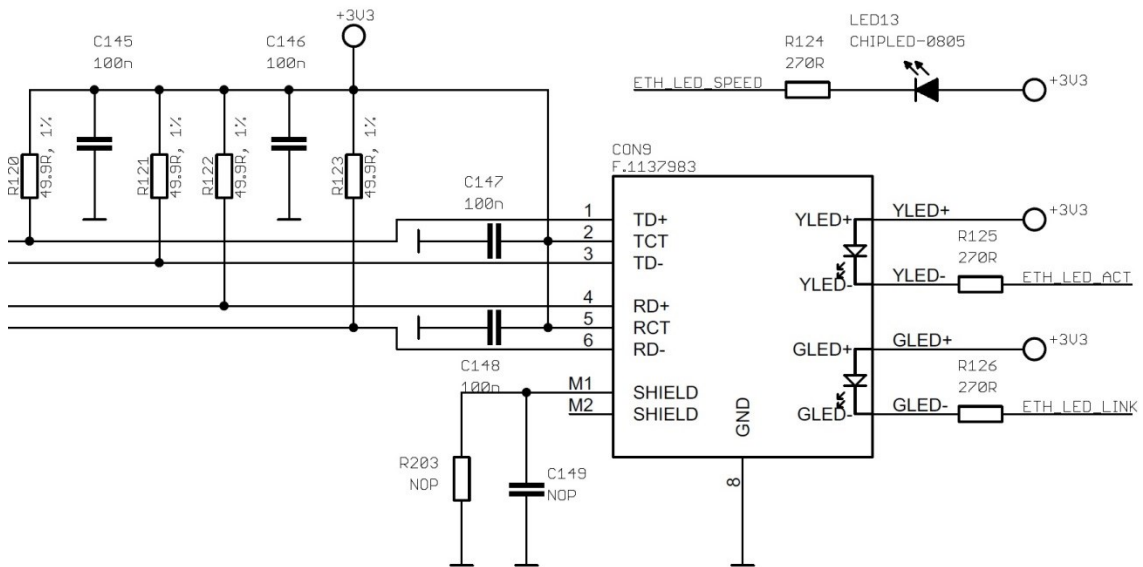
Pro volbu mezi MII/RMII rozhraními slouží pin RX_DV (MII_MODE). Jelikož chceme používat RMII rozhraní a interní rezistor je typu pull-down, přidáme externí pull-up rezistor. Budič chceme využívat v RMII Master režimu (generování 50 MHz s využitím 25 MHz krystalu připojeného k vnitřnímu oscilátoru), tudíž na pin TXD_3(RMII_MAS) opět přidáme externí pull-up rezistor. Ve spojitosti s rozvodem hodinového signálu (podkapitola 3.14 Rozvod hodin) je na místě povolit výstup hodinového signálu na pinu CLK_OUT. Toto lze opět zajistit externím pull-up rezistorem na pinu GPIO1 (CLK_OUT_EN). Posledním důležitým nastavením, které zde má smysl uvádět, je volba varianty Ethernetu (rychlost, obousměrná komunikace, ...). Nastavením pinů RX_ER (FX_EN_Z), LED_LINK (AN_EN), LED_SPEED (AN1) a LED_ACT (AN0) do log. 1 se povolí Autonegotiation protokol a tím i automatická volba nejlepší varianty Ethernetu pro použitý prvek síťové infrastruktury. Signály použité pro propojení FE řadiče s budičem DP83640 jsou uvedeny na obr. 3.11



Obr. 3.11 DP83640 RMII Master – zdroj (TI, AN-1794 Using RMII Master Mode, 2013)

SI-60062-F

Poslední důležitou komponentou fyzického rozhraní Ethernet je RJ-45 konektor a impedanční přizpůsobení ve formě transformátorů. Konektor SI-60062-F byl také použit v bakalářské práci (Blecha, 2012), je ozkoušený a je k dispozici dostatek kusů. Zmíněný konektor v sobě obsahuje impedanční přizpůsobení a dvě signalizační LED diody. Schéma zapojení konektoru včetně doplňujících součástek je na obr. 3.12.



Obr. 3.12 SI-60062-F konektor

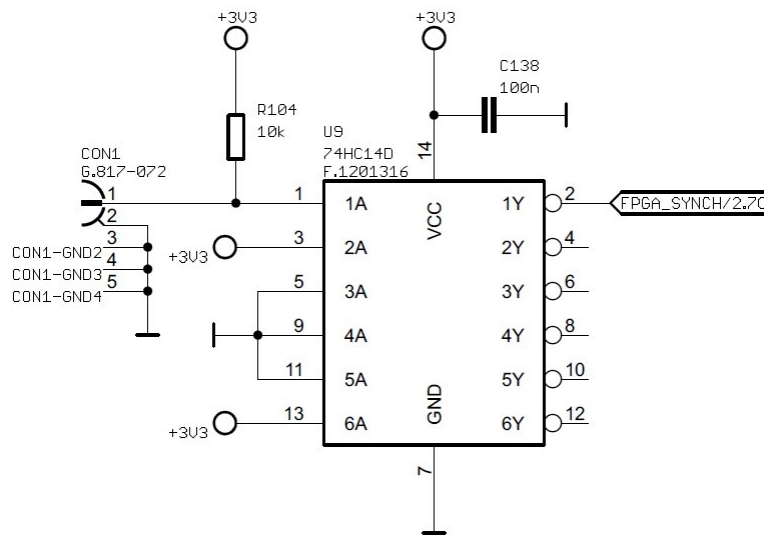
3.9 Fyzické rozhraní RS-232

V MCU je integrována periférie SCI, která umožňuje arytmický způsob komunikace. Časování je tedy zajištěno. Zbývá zajistit správné mapování logických úrovní na napěťové úrovně. V této práci je využito standardní řešení obvodem MAX232 resp. jeho 3.3V variantou MAX3232. Více informací o tomto obvodu viz (TI, MAX3232 Datasheet, 2014).

3.10 Externí synchronizace

Vyvíjené zařízení má být určeno k provádění měření a testování. Může tedy nastat případ, kdy bude požadováno spustit připravený test v přesně definovaný okamžik. K tomuto účelu je v zařízení integrován synchronizační obvod, který je realizován SMA konektorem a obvodem 74HC14 (schmittův klopný obvod). Z důvodu kompatibility napěťových úrovní je na vstup 74HC14 připojen pull-up rezistor (využití tranzistoru v zapojení s otevřeným kolektorem/drainem). Synchronizační signál je z výstupu

74HC14 přiveden do FPGA. Více informací o 74HC14 viz (NXP, 74HC14 Datasheet, 2012). Schéma celého synchronizačního obvodu je uvedeno na obr. 3.13.



Obr. 3.13 Externí synchronizace

3.11 Fyzické rozhraní CAN

Celkový počet CAN řadičů integrovaných ve vyvíjeném zařízení se pohybuje v rozmezí 3 až 8 (3 v MCU + až 5 v FPGA). Pro možnost připojení na sběrnici CAN je potřebné k řadičům připojit fyzické rozhraní (budič, impedanční přizpůsobení a konektor), které zajistí správné hodnoty parametrů zvolené fyzické vrstvy (ISO 11898-2).

V této době existuje velké množství budičů, které je možné využít. Základní funkčnost jednotlivých budičů je stejná. Obvykle se liší přidáním nějakých podpůrných funkcí (např. sleep mód, inhibit výstup, napěťová reference pinů, atd.).

TJA1041

Budič TJA1041 pochází ze sortimentu firmy NXP. Jedná se o obvod, který je plně kompatibilní se standardem ISO 11898. Obvod nabízí výborné parametry ohledně elektromagnetické kompatibility (EMC), velmi malý proudový odběr a pasivní chování při vypnutém napájecím zdroji. Rozšiřující funkčnost implementuje např. sleep mód s podporou lokálního/vzdáleného zdroje vzbuzení, ochranné a diagnostické funkce pro ochranu proti zkratům nebo napěťové přizpůsobení V/V pinů. Chování budiče je závislé na operačním módu, ve kterém se budič nachází. Jednotlivé operační módy a přechody mezi těmito módy lze přehledně zobrazit stavovým diagramem (orientovaný uzlový graf). Propojení mezi budičem a FPGA je realizováno využitím 6 pinů (TXD, RXD, EN, ERR, WAKE, STB). Obvod je dostupný v pouzdře SO14.

Někdy je možné využít částečné pinové kompatibility s budičem PCA82C250, který je vyráběn v pouzdře SO8. Tento budič nemá napět'ové přizpůsobení V/V pinů a v této práci použité MCU nemá 5 V tolerantní V/V piny, tudíž by bylo nutné tento problém vyřešit jiným způsobem (např. dělič napětí, Zenerova dioda).

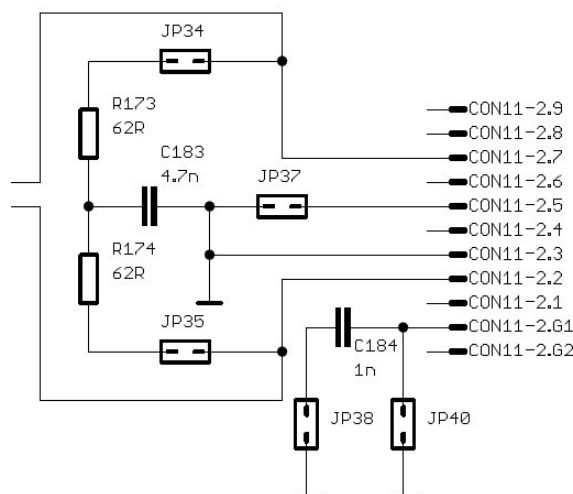
Více informací o budiči TJA1041 viz (NXP, TJA1041 Datasheet, 2007).

Impedanční přizpůsobení a konektor

Impedanční přizpůsobení je realizováno RC strukturou (R173, R174 a C183) zobrazenou na obr. 3.14. Ideálně by měly být velikosti odporu rezistorů rovny 60Ω (62Ω je nejbližší hodnota v řadě E24). V zapojení jsou integrovány propojky (JP34 a JP35), které slouží pro připojení/odpojení struktury (strukturu využívají vždy jen koncové budiče na sběrnici čili vždy právě 2).

V zařízeních podobného typu je často využívána zástrčka konektoru D-SUB-9 (známý také jako CANON-9). Vzhledem k počtu fyzických rozhraní byl zvolen tento konektor v dvouportové variantě (dva konektory nad sebou). Zapojení pinů konektoru je uvedeno na obr. 3.14.

U všech konektorů použitých u komunikačních technologií CAN, LIN a FlexRay je možnost volby propojení stínění a země pomocí zkratu, či přes kondenzátor. Volbu je možné realizovat s pomocí propojek (pro CAN – JP38 a JP40 na obr. 3.14).



Obr. 3.14 Konektor a impedanční přizpůsobení CAN

Ve vyvíjeném zařízení jsou implementována 4 kompletní fyzická rozhraní CAN (zadání požadovalo alespoň 3).

3.12 Fyzické rozhraní LIN

Celkový počet LIN řadičů integrovaných ve vyvíjeném zařízení se pohybuje v rozmezí 1 až 6 (1 v MCU + až 5 v FPGA). Stejně jako u protokolu CAN, tak i u protokolu LIN je nutné k řadiči připojit fyzické rozhraní (budič a konektor), které zajistí správné hodnoty parametrů fyzické vrstvy.

Budičů je dostupných opět celá řada. Hlavním rozdílem je kompatibilita s odlišnými specifikacemi protokolu LIN (v2.0, v2.1).

TPIC1021A

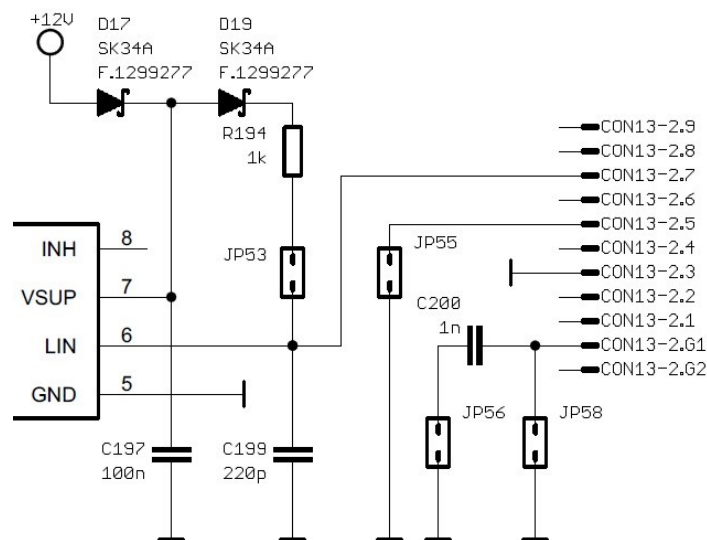
Budič TPIC1021A pochází ze sortimentu firmy TI. Tento budič je plně kompatibilní se specifikací fyzické vrstvy LIN v2.0. Umožňuje tedy přenos dat rychlostí 2.4 kb/s až 20 kb/s. Data na pinu TXD jsou vysílačem přenášena na pin LIN. Data na pinu LIN jsou přijímačem přenášena na pin RXD. Pin TXD je 5 V TTL kompatibilní čili ho lze bez problémů připojit na výstup FPGA. Pin RXD je typu OC, takže při použití pull-up rezistoru na napětí 3.3 V je tento pin také možné přímo propojit se vstupem FPGA. Pro provoz budiče ve Slave módu není potřeba na pin LIN přidávat žádný externí pull-up rezistor (v budiči integrovaná dioda a rezistor s odporem cca 30 k Ω), ovšem pro provoz budiče v Master módu je nutné přidat sériové zapojení externího pull-up rezistoru (hodnota 1 k Ω) na napětí V_{SUP} a externí diody viz obr. 3.15. TPIC1021A podporuje sleep mód a možnost lokálního/vzdáleného probuzení. Chování budiče je závislé na aktuálně zvoleném operačním módu. Jednotlivé operační módy a přechody mezi nimi je možné zobrazit stavovým diagramem. Budič je přímo propojen s FPGA pomocí 4 pinů (RXD, EN, NWAKE, TXD). Obvod byl navržen pro provoz v automobilovém prostředí a standardně je dispoziční v pouzdře SO8.

Možnou náhradou je budič SN65HVDA100 od firmy TI, který je kompatibilní se specifikací fyzické vrstvy LIN v2.1. Tyto obvody jsou zcela pinově kompatibilní.

Více informací o budiči TPIC1021A viz (TI, TPIC1021A Datasheet, 2009).

Konektor

Dvouportová varianta konektoru D-SUB-9 je i v tomto případě dobrou volbou. Zapojení konektoru je uvedeno na obr. 3.15.



Obr. 3.15 Konektor a volba Master/Slave módu LIN

Ve vyvíjeném zařízení jsou implementována 4 kompletní fyzická rozhraní LIN (zadání vyžadovalo alespoň 2).

3.13 Fyzické rozhraní FlexRay

Celkový počet FlexRay řadičů integrovaných ve vyvíjeném zařízení se pohybuje v rozmezí 1 až 4 (1 v MCU + až 3 v FPGA). Stejně jako u předchozích dvou komunikačních technologií, tak i u komunikační technologie FlexRay je nutné k řadiči připojit fyzické rozhraní (budič, impedanční přizpůsobení a konektor), které zajistí správné hodnoty parametrů fyzické vrstvy. FlexRay je z diskutovaných komunikačních technologií nejsložitější. Tento fakt platí i na úrovni fyzické vrstvy (např. dvoukanálová struktura, 10 Mb/s přenosová rychlost, Bus guardian, atd.).

TJA1080

Tento budič je ze sortimentu firmy NXP. Jedná se o jednocanálový budič, který je kompatibilní se specifikací fyzické vrstvy FlexRay v2.1 Rev. A. Budič podporuje přenosové rychlosti až 10 Mb/s, má dobré parametry ohledně EMC i ESD. TJA1080 může být použit jako uzel v pasivní sběrníkové topologii i jako opakovač v topologii aktivní hvězda. Budič má dále implementováno přizpůsobení napětíových úrovní V/V pinů (vhodné pro propojení s FPGA), Bus guardian rozhraní, či diagnostické (např. detekce přehřátí, detekce zkratu, detekce malého napětí, atd.) a ochranné (např. v případě malého napětí na V_{BAT} , V_{CC} , V_{IO} přejde obvod do bezpečného módu, pasivní chování pinů BP a BM při vypnutém napájecím zdroji, atd.) funkce. Jeho chování se opět liší v závislosti na aktuálním operačním módu, ve kterém se budič nachází.

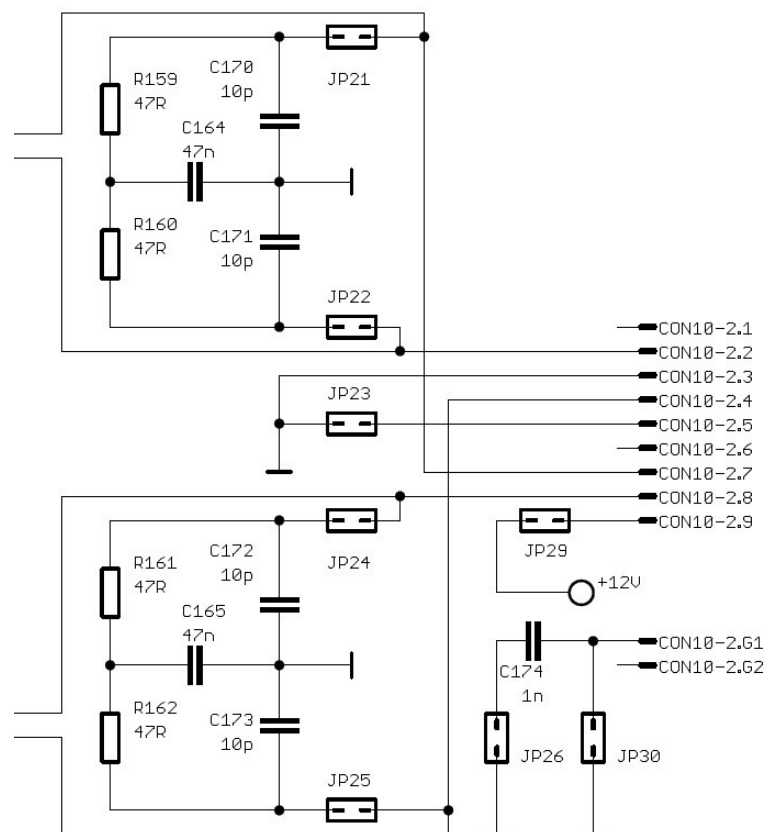
Jednotlivé operační módy včetně přechodů mezi nimi lze opět přehledně zobrazit stavovým diagramem (konkrétně dvěma diagramy – pro uzel a pro opakovač). Propojení budiče a FPGA je realizováno pomocí 9 pinů (EN, TXD, TXEN, RXD, BGE, STBN, RXEN, ERRN, WAKE), dva kanály tedy vyžadují připojení pomocí 18 pinů! Budič je dostupný v pouzdře SSOP20.

Více informací o budiči TJA1080 viz (NXP, TJA1080 Datasheet, 2007).

Konektor a impedanční přizpůsobení

Impedanční přizpůsobení jednoho kanálu je realizováno RC strukturou (R159, R160, C164, C170 a C171) uvedenou na obr. 3.16. Celá struktura je opět odpojitelná pomocí dvou propojek (JP21 a JP22), tím je zajištěno korektní připojení na sběrnici (pouze koncové uzly mají strukturu zapojenou čili vždy právě 2). Velikosti odporu rezistorů R159 a R160 by ideálně měly být 50 Ω . V řadě E12 je k dispozici nejbližší hodnota 47 Ω nebo v řadě E24 je nejbližší hodnota 51 Ω .

Opět byla zvolena zástrčka dvouportové varianty konektoru D-SUB-9. Zapojení pinů konektoru je uvedeno na obr. 3.16.



Obr. 3.16 Konektor a impedanční přizpůsobení FlexRay

Ve vyvíjeném zařízení jsou implementována 2 kompletní fyzická rozhraní FlexRay (zadání vyžadovalo alespoň 2).

3.14 Rozvod hodin

MCU

Taktování MCU je zprostředkováno dvěma způsoby:

- krystalový oscilátor (krystal 5 – 20 MHz),
- externí hodinový signál přivedený skrz SMA konektor (synchronizace více jednotek).

Další možnosti přivedení taktovacího signálu do MCU jsou piny EXTCLKIN1 a EXTCLKIN2. Hodinový signál přivedený do těchto pinů může mít maximální frekvenci 80 MHz a není přiveden do žádného fázového závěsu (dále PLL). V této aplikaci jsou tyto vstupy nepoužitelné.

Hodinový signál je z MCU vyveden pinem ECLK. Jako zdroj tohoto signálu lze v MCU zvolit hodinovou doménu VCLK nebo vstupní hodinový signál OSCIN. Zvolený zdrojový signál je dále možno vydělit 16bitovou frekvenční děličkou.

FE budič

Taktování FE budiče je realizováno také dvěma způsoby:

- krystalový oscilátor (krystal 25 MHz),
- hodinový signál o frekvenci 25 MHz přivedený z MCU (pin ECLK).

Jelikož FE budič pracuje v RMI Master režimu, tak ze vstupního hodinového signálu je vygenerován hodinový signál o frekvenci 50 MHz. Tento hodinový signál je vyveden na pinech RX_CLK, TX_CLK a CLK_OUT.

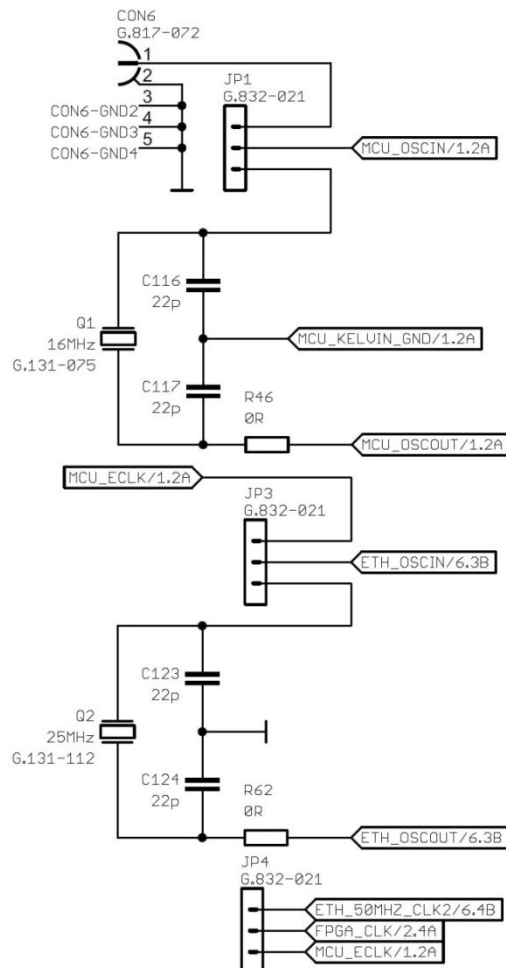
FPGA

Taktování FPGA (pro sekvenční obvody) je opět realizováno dvěma způsoby:

- hodinový signál o nastavitelné frekvenci přivedený z MCU (pin ECLK),
- hodinový signál o frekvenci 50 MHz přivedený z FE budiče (pin CLK_OUT).

FPGA má 15 dedikovaných pinů pro přivedení hodinového signálu. Tyto piny přivádí hodinový signál do globální hodinové domény GCLK (hodinový signál je rozveden do celého FPGA). Frekvenci vstupního hodinového signálu je možné změnit použitím PLL.

Diskutované rozvedení hodinového signálu je nakresleno na obr. 3.17.



Obr. 3.17 Rozvod hodin

3.15 Reset

Vnější reset MCU lze rozdělit do dvou hlavních kategorií:

- studený reset (pin NPORRST, obvykle se používá pro připojení externího monitorovacího obvodu napájení),
- teplý reset (pin NRST, používaný pro reset MCU v závislosti na ostatních použitých integrovaných obvodech).

MCU má vestavěný monitorovací obvod napájení, který generuje power-on reset, když se napětí jádra nebo napětí V/V brány dostane mimo povolený rozsah. Tento obvod by neměl být považován za náhradu za externí monitorovací obvod.

Jak již bylo zmíněno, tak k napájení MCU jsou zapotřebí dvě napěťové větve:

- 1.2 V – napájení jádra,
- 3.3 V – napájení V/V brány.

Obě tyto napěťové větve je nutné monitorovat, jestli jsou v povoleném rozsahu. K monitoringu jsou použity integrované obvody TPS3808G12 pro napětí 1.2 V a TPS3808G33 pro napětí 3.3 V. Oba tyto obvody pochází ze sortimentu firmy TI. Monitorované napětí je nutno připojit na pin SENSE. Pomocí pinu C_T je možné nastavit zpoždění generování resetu po detekci příznaku (splněné podmínky k resetu). Nastavení se provádí volbou hodnoty připojeného kondenzátoru. Pokud je pin připojen pomocí rezistoru na napájecí napětí nebo je ponechán plovoucí, tak je zpoždění konstantní (typicky 20 ms). Obvod dále obsahuje pin MR, který lze využít k vynucení generování resetu (např. pomocí tlačítka). Reset je generován na pinu RESET (výstup typu OC).

FPGA má vestavěný monitorovací obvod napájení. Není nutné navrhovat externí.

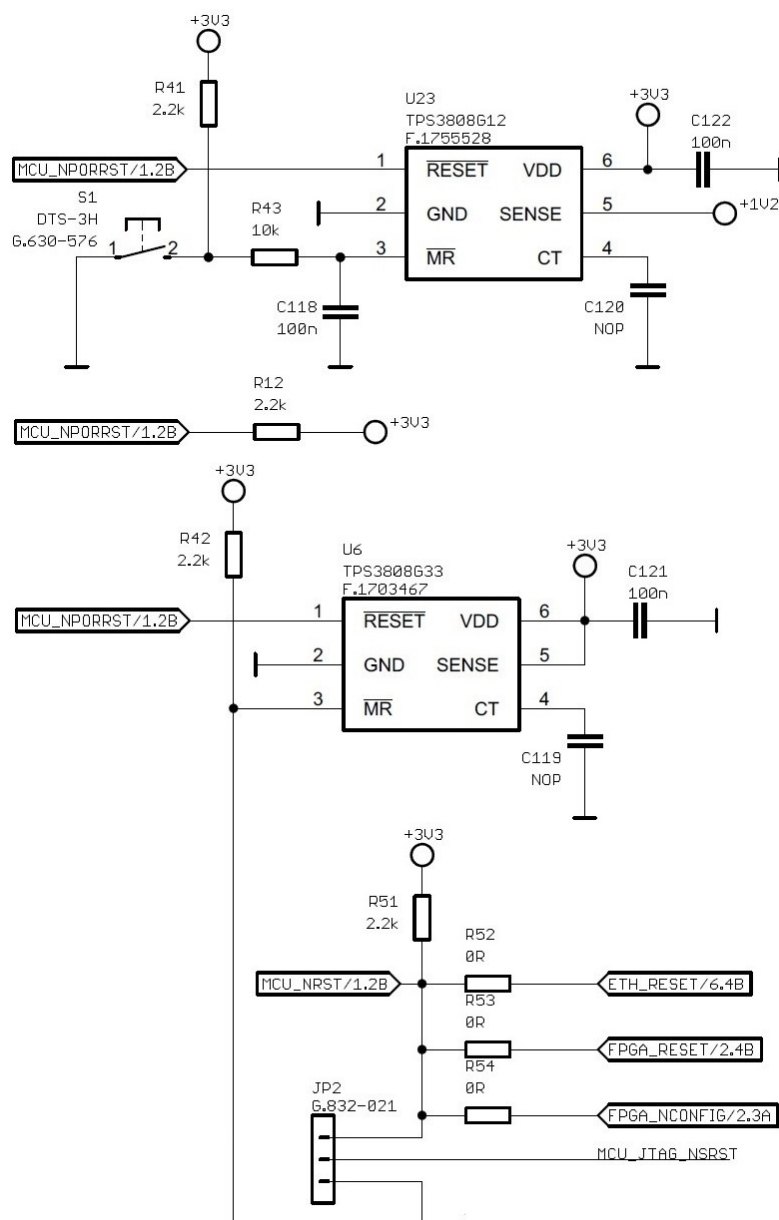
Pro resetování MCU uživatelem je k dispozici tlačítko, které je připojeno na pin MR obvodu TPS3808G12.

JTAG signál NSRST (system reset) lze využitím propojky připojit buď na pin MCU NRST (teplý reset) nebo na pin MR obvodu TPS3808G33 (prostřednictvím tohoto obvodu je generován studený reset).

Pin MCU NRST je dále možné pomocí propojek připojit k FE budiče (pin RESET_N) a k FPGA (piny NCONFIG a IOG5). Toto zapojení slouží k již zmiňované možnosti signalizace/generování resetu ve více IO současně.

Pro více informací o monitorovacích obvodech TPS3808Gxx viz (TI, TPS3808 Datasheet, 2008).

Monitoring napětí a rozvod reset signálu je uveden na obr. 3.18.



Obr. 3.18 Monitoring napětí a reset

3.16 Zdroj napájení

Poslední a také jedna z nejdůležitějších částí návrhu je zdroj napájení. Popis této části byl záměrně ponechán nakonec, protože jsou nyní popsány všechny obvody a tudíž je možné odhadnout proudový odběr pro jednotlivé napájecí větve (1.2 V, 2.5 V, 3.3 V, 5 V a 12 V).

U valné většiny obvodů je možné proudový odběr odečíst z jejich katalogového listu. Toto ovšem neplatí pro FPGA, jehož odběr závisí na aktuálním používaném návrhu. Společnost Altera dává k dispozici nástroj Early Power Estimator, který umí odhadnout proudový odběr FPGA v závislosti na použitém návrhu. Tento nástroj nebylo možné

využít, protože v době návrhu schématu neexistoval návrh do FPGA. Při provádění odhadu bylo potřeba zvážit jakou zvolit strukturu propojení jednotlivých napěťových větví (paralelní, sériové, sérioparalelní) a jaké typy napěťových regulátorů využít (lineární stabilizátor, spínaný regulátor). Odhad proudového odběru pro jednotlivé napěťové větve je uveden v tab. 3.6. Uvedený odhad počítá s účinností spínaných regulátorů 70 % (1.2 V), 80 % (3.3 V) a 80 % (5 V).

U [V]	I [A]	Zátěž
1.2	2.0	MCU, FPGA
2.5	0.1	FPGA
3.3	1.0	MCU, FPGA, FE Budič, 2.5 V
5	0.6	CAN budiče, FlexRay budiče
12	1.0	LIN budiče, 1.2 V, 3.3 V, 5 V

Tab. 3.6 Odhad proudového odběru

Po provedení odhadu bylo možné přistoupit k výběru konkrétních napěťových regulátorů.

Vstupní napětí je předpokládáno 12 V, není tedy nutné řešit regulaci této napěťové větve. Proudové odběry na napěťových větvích 1.2 V, 3.3 V a 5 V jsou poměrně vysoké, tudíž je vhodné použít spínané regulátory. Proudový odběr na napěťové větvi 2.5 V je malý, takže stačí použít lineární stabilizátor (LDO).

LT3507A

Tento integrovaný obvod patří do rodiny vícekanálových napěťových regulátorů (PMIC). Pochází ze sortimentu firmy LT. LT3507A obsahuje 3 asynchronní (nutné externí demagnetizační dioda) spínané regulátory a 1 lineární regulátor vyžadující externí NPN tranzistor. První spínaný regulátor je schopný dodat výstupní proud až 2.7 A. Zbylé dva spínané regulátory jsou schopné dodat proud až 1.8 A. Všechny spínané regulátory jsou synchronizovány na takt hlavního oscilátoru, jehož frekvenci je možné přeladit od 250 kHz až do 2.5 MHz pouze pomocí jednoho rezistoru. První spínaný regulátor pracuje v opačné fázi než zbylé dva spínané regulátory. Tento způsob řízení zajišťuje minimalizaci zvlnění vstupního proudu. Vysoká spínací frekvence umožňuje využití malých cívek a kondenzátorů. LT3507A má integrované ochrany proti proudovému přetížení a proti přehřátí. Jednotlivé regulátory umožňují sledování výstupu jiných regulátorů a tím se nabízí možnost volitelného fázování jednotlivých napěťových větví (některá FPGA potřebují náběh jednotlivých napájecích napětí

v daném pořadí). Posledními dvěma důležitými vlastnostmi jsou indikace dosažení výstupního napětí na nastavenou úroveň (pro všechny spínané regulátory) a monitoring vstupního napětí, který umožňuje vypnutí regulátorů, pokud je vstupní napětí mimo definovaný napěťový interval. LT3507A je vyráběn v pouzdře TSSOP38 s výkonovou ploškou (zespoda pouzdra), která je využita pro připojení země a hlavně k odvodu tepla z obvodu do rozlitych měděných ploch. Více informací o LT3507A viz (LT, 2011).

Nastavení spínací frekvence

Nastavení spínací frekvence LT3507A musí splňovat níže uvedené vztahy:

$$f_{SW} \leq f_{MAX1} = \left(\frac{U_{OUT} + U_F}{U_{IN} - U_{SW} + U_F} \right) \cdot \frac{1}{t_{ON(MIN)}},$$

$$f_{SW} \leq f_{MAX2} = \left(1 - \frac{U_{OUT} + U_F}{U_{IN} - U_{SW} + U_F} \right) \cdot \frac{1}{t_{OFF(MIN)}},$$

kde f_{SW} je spínací frekvence [f_{SW}] = Hz, $U_{OUT} \in \{1.2 V, 3.3 V, 5 V\}$ je výstupní napětí [U_{OUT}] = V, $U_{IN} = 12 V$ je vstupní napětí [U_{IN}] = V, $U_F \cong 0.4 V$ je úbytek napětí na demagnetizační diodě [U_F] = V, $U_{SW} \cong 0.3 V$ je saturační napětí spínacího tranzistoru [U_{SW}] = V, $t_{ON(MIN)} = 130 ns$ je minimální doba sepnutí spínacího tranzistoru [$t_{ON(MIN)}$] = s a $t_{OFF(MIN)} = 170 ns$ je minimální doba rozepnutí spínacího tranzistoru [$t_{OFF(MIN)}$] = s. Spočtené hodnoty pro všechny výstupní napětí jsou uvedeny v tab. 3.7.

U_{OUT} [V]	f_{MAX1} [kHz]	f_{MAX2} [kHz]
1.2	1017.2	5104.5
3.3	2352.2	4083.6
5	3432.9	3257.2

Tab. 3.7 Omezení spínací frekvence LT3507A

Volba spínací frekvence $f_{SW} \cong 500 kHz$ se zdá být dobrou volbou. Spínací frekvence je volena rezistorem, jehož odpor lze spočítat z následujícího vztahu:

$$f_{SW} = \frac{53.2 k\Omega \cdot MHz}{R + 12.4 k\Omega},$$

kde f_{SW} je spínací frekvence [f_{SW}] = MHz a R je hodnota nastavovacího rezistoru [R] = kΩ. Pro $f_{SW} = 500 kHz$ je $R = 94 k\Omega$. Nejbližší hodnota v řadě E12 je 100 kΩ. Pro tuto hodnotu rezistoru tedy vyjde $f_{SW} = 473.31 kHz$.

Návrh napěťové větve 1.2 V

Jako příklad zde bude uveden návrh napěťové větve 1.2 V. Návrh napěťové větvi 3.3 V a 5 V je obdobný.

Výstupní napětí U_{OUT} je voleno napěťovým děličem. Poměr hodnot rezistorů je dán následujícím vztahem:

$$R_{14} = R_{16} \left(\frac{U_{OUT}}{0.8 V} - 1 \right),$$

kde R_{14} , R_{16} jsou rezistory v napěťovém děliči $[R_x] = \Omega$ a U_{OUT} je výstupní napětí $[U_{OUT}] = V$. Označení rezistorů koresponduje s označením na obr. 3.21. Volba hodnot rezistorů by měla splňovat podmínku:

$$R_{14} || R_{16} \leq 10 k\Omega.$$

Po vypočtení hodnot rezistorů byly vybrány nejbližší hodnoty z řady E24 (přesnost 1 %), tedy $R_{14} = 5.1 k\Omega$ a $R_{16} = 10 k\Omega$. Pro tyto hodnoty vychází výstupní napětí $U_{OUT} = 1.208 V$.

Další součástí, jejíž hodnotu je nutné spočítat, je cívka. Pokud je spínací tranzistor v regulátoru rozeprnutý, tak platí:

$$U_L = U_{OUT} + U_F,$$

kde U_L je napětí na cívce $[U_L] = V$, U_{OUT} je výstupní napětí $[U_{OUT}] = V$ a U_F je úbytek napětí na demagnetizační diodě $[U_F] = V$. S využitím následujícího vztahu je možné určit zvlnění proudu špička-špička v cívce (pilovitý průběh se střední hodnotou rovnou proudu zátěže):

$$\Delta I_L = \left(1 - \frac{U_{OUT}}{U_{IN}} \right) \cdot \frac{U_L}{L \cdot f_{SW}},$$

kde ΔI_L je zvlnění proudu špička-špička v cívce $[\Delta I_L] = A$, U_{OUT} je výstupní napětí $[U_{OUT}] = V$, U_{IN} je vstupní napětí $[U_{IN}] = V$, U_L je napětí na cívce $[U_L] = V$, L je indukčnost cívky $[L] = H$ a f_{SW} je spínací frekvence $[f_{SW}] = Hz$. Maximální proud cívkou je tedy:

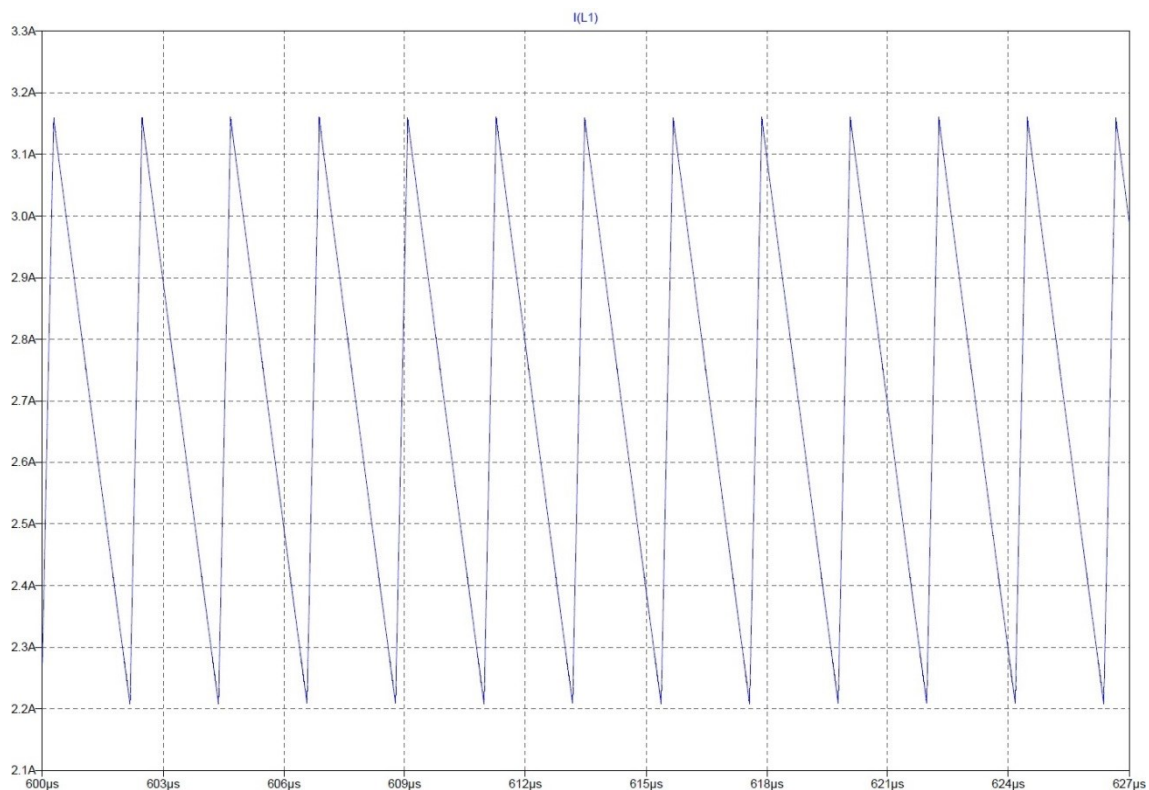
$$I_{LPK} = I_{OUT} + \frac{\Delta I_L}{2},$$

kde I_{LPK} je maximální hodnota proudu cívkou (spínacím tranzistorem) $[I_{LPK}] = A$, I_{OUT} je proud do zátěže $[I_{OUT}] = A$ a ΔI_L je zvlnění proudu špička-špička v cívce $[\Delta I_L] = A$. Maximální proud cívkou (spínacím tranzistorem) nesmí překročit interní limit, který je pro tento regulátor $I_{LIM} \cong 5.1 A$ (limit je závislý na střídě spínání tranzistoru). Cívka by dále měla splňovat následující podmínku:

$$\Delta I_L \leq 0.3 \cdot I_{LIM},$$

kde ΔI_L je zvlnění proudu špička-špička v cívce $[\Delta I_L] = A$ a I_{LIM} je interní proudový limit regulátoru $[I_{LIM}] = A$.

Pro $L = 3.3 \mu H$, $f_{SW} = 473.31 kHz$, $U_{IN} = 12 V$, $U_{OUT} \cong 1.2 V$ a $U_F \cong 0.4 V$ vychází $\Delta I_L = 0.92 A$. Pro proud zátěží $I_{OUT} = 2.7 A$ vychází $I_{LPK} = 3.16 A$. Podmínky jsou tedy splněny. Při výběru cívky je nutné dbát na její parametry (efektivní hodnota proudu, saturační proud a stejnosměrný odpor). Vybraná cívka má následující parametry: $L = 3.3 \mu H \pm 20\%$, $R_{DC(MAX)} = 0.0233 \Omega$, $I_{RMS} = 8.1 A$ a $I_{SAT} = 8.7 A$. Výsledek simulace proudu cívkou pro uvedené hodnoty parametrů je na obr. 3.19.



Obr. 3.19 Proud cívkou

Pro filtraci výstupního napětí (minimalizaci napět'ového zvlnění) je nutné za cívku zapojit dostatečně veliký a dostatečně kvalitní kondenzátor. Výstupní kondenzátor

slouží jednak k filtraci proudu cívku, dále pak jako akumulátor energie pro zátěže s dynamickým odběrem (stabilizace řídicí smyčky). Protože LT3507A pracuje na vysoké spínací frekvenci, je možné použít kondenzátory s relativně malou kapacitou (keramické kondenzátory). Často se také používá paralelní kombinace většího (elektrolytického/tantalového) kondenzátoru a menšího (keramického) kondenzátoru. Zvlnění výstupního napětí lze odhadnout následujícím vztahem:

$$\Delta U_C = \frac{\Delta I_L}{8 \cdot f_{SW} \cdot C_{OUT}} \text{ pro keramické kondenzátory,}$$

$$\Delta U_C = \Delta I_L \cdot ESR \text{ pro elektrolytické/tantalové kondenzátory,}$$

kde ΔU_C je zvlnění špička-špička výstupního napětí $[\Delta U_C] = V$, ΔI_L je zvlnění proudu špička-špička v cívce $[\Delta I_L] = A$, f_{SW} je spínací frekvence $[f_{SW}] = Hz$, C_{OUT} je kapacita výstupního kondenzátoru $[C_{OUT}] = F$ a ESR je ekvivalentní sériový odpor výstupního kondenzátoru $[ESR] = \Omega$. Napěťové zvlnění se snažíme co nejvíce minimalizovat. Jeho hodnotu je vhodné volit např.

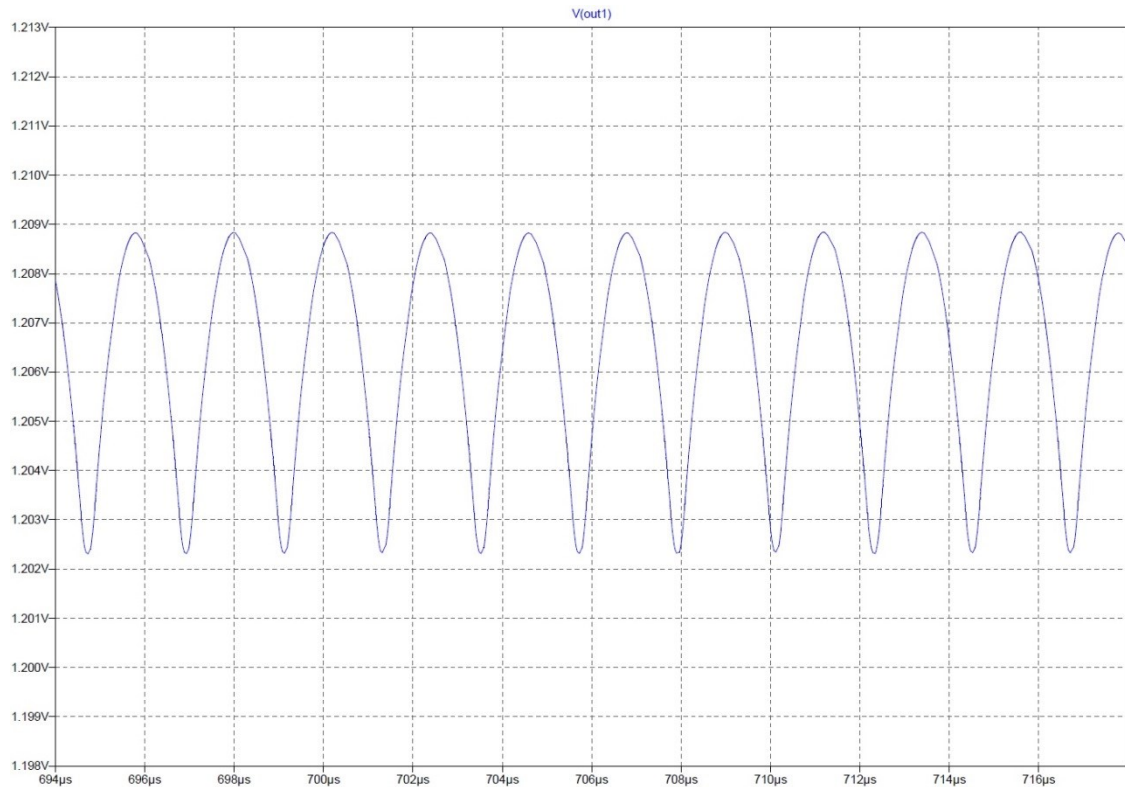
$$\Delta U_C \leq 0.01 \cdot U_{OUT},$$

kde ΔU_C je zvlnění špička-špička výstupního napětí $[\Delta U_C] = V$ a U_{OUT} je výstupní napětí $[U_{OUT}] = V$. Důležitá podmínka pro volbu výstupního kondenzátoru je, že energie uložená v kondenzátoru musí být větší než energie uložená v cívce. Když není spínací tranzistor sepnutý, dochází k přesunu energie z cívky do kondenzátoru a tím i k zvětšování napětí na kondenzátoru. Následující vztah zajišťuje, že energie uložená v kondenzátoru je řádově větší než energie uložená v cívce:

$$C_{OUT} \geq 10 \cdot L \cdot \left(\frac{I_{LPK}}{U_{OUT}} \right)^2,$$

kde C_{OUT} je kapacita výstupního kondenzátoru $[C_{OUT}] = F$, L je indukčnost cívky $[L] = H$, I_{LPK} je maximální hodnota proudu cívku (spínacím tranzistorem) $[I_{LPK}] = A$ a U_{OUT} je výstupní napětí $[U_{OUT}] = V$. Rozumným kompromisem byla volba kombinace tantalového a 4 keramických kondenzátorů. Tantalový kondenzátor má následující parametry: $C = 100 \mu F$, $ESR = 0.9 \Omega @ 100 kHz$ a $U = 16 V$. Keramické kondenzátory mají následující parametry: $C = 10 \mu F$, $U = 16 V$ a dielektrikum X5R. Pro tuto hodnotu kapacity není splněn předchozí vztah. Z provedených simulací bylo

zjištěno, že toto pochybení není nijak kritické (projevuje se hlavně při náběhu napětí). Výsledek simulace výstupního napětí $U_{OUT} \cong 1.2 V$, při výstupním proudu $I_{OUT} = 2.7 A$, po odeznění přechodového děje (náběhu napájecí větve na nastavenou úroveň) je uveden na obr. 3.20.



Obr. 3.20 Výstupní napětí 1.2 V

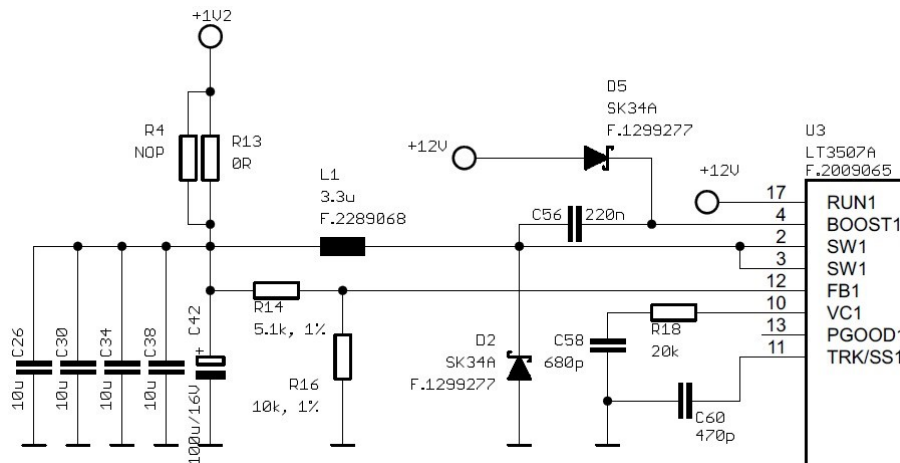
Výběr demagnetizační diody není nijak složitý. Na diodě by měl být co nejmenší úbytek napětí, musí vydržet statický proud rovný výstupnímu proudu a musí vydržet napětí v závěrném směru rovné vstupnímu napětí. Tyto požadavky velmi dobře splňuje Schottkyho dioda SK34A.

Poslední zde zmíněnou součástí je BOOST kondenzátor. Tento kondenzátor slouží jako zdroj napájení pro interní budič, který je zodpovědný za spínání tranzistoru. Pro korektní funkčnost je nutné zajistit, aby pin BOOST měl minimálně o 2.5 V větší napětí než je na pinu SW. Existuje několik možných zapojení. Zde je použito zapojení uvedené na obr. 3.21 čili připojení vstupního napětí přes diodu na BOOST kondenzátor. Kapacitu BOOST kondenzátoru je možné spočítat z následujícího vztahu:

$$C_{BOOST} = 100 \text{ nF} \cdot \frac{1 \text{ MHz}}{f_{SW}}$$

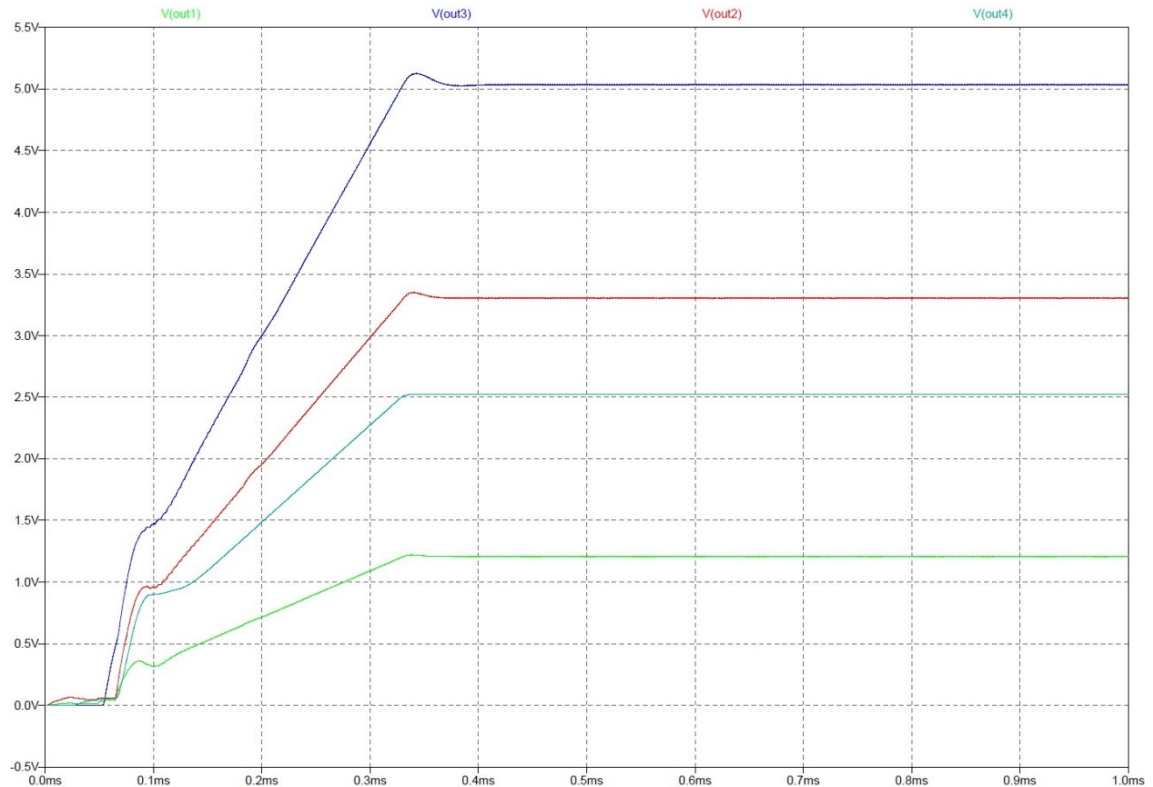
kde C_{BOOST} je kapacita BOOST kondenzátoru $[C_{BOOST}] = nF$ a f_{SW} je spínací frekvence regulátoru $[f_{SW}] = MHz$. Pro $f_{SW} = 473.31 kHz$ vychází $C_{BOOST} = 211.28 nF$. V řadě E12 je nejbližší hodnota $C_{BOOST} = 220 nF$.

Pro další informace o návrhu viz (LT, 2011). Schéma zapojení napěťové větve 1.2 V je uvedeno na obr. 3.21.



Obr. 3.21 Napěťová větev 1.2 V

Na obr. 3.22 je uveden výsledek simulace všech napěťových větví napájecího zdroje.



Obr. 3.22 Napěťové větve 1.2 V, 2.5 V, 3.3 V a 5V

Všechny uvedené simulace byly provedeny s využitím programu LTspice IV od LT.

3.17 Návrh DPS

Návrh desky plošných spojů (dále DPS) je netriviální komplexní úloha vyžadující znalosti základů elektřiny a magnetismu, elektronických obvodů a návrhových pravidel (zahrnující konstrukční možnosti). K návrhu je dále potřebný specifický software (např. Cadence OrCad, Cadsoft Eagle, atd.), který svými vlastnostmi umožní dosažení požadovaných výsledků. Dobrá volba a znalost programu velmi často ulehčí práci! Není možné říci univerzální návod jak DPS navrhovat, protože každá DPS je svým způsobem unikátní. Pro dosažení dobrých výsledků návrhu DPS jsou nutné zkušenosti, které lze nabýt pouze praxí (navrhnout co nejvíce desek). Níže je uveden stručný popis návrhu DPS vyvíjeného zařízení.

Prvním krokem při návrhu DPS bylo načtení vytvořeného netlistu (ze schématického editoru) do návrhového editoru a počáteční rozmístění součástek (hlavně konektorů), aby bylo možné odhadnout výslednou velikost DPS.

Po provedení odhadu velikosti DPS přišel na řadu výběr konstrukční krabičky. Při výběru krabičky bylo nutné dbát na její rozměry (cena výroby DPS roste s plochou DPS) a také na mechanické vlastnosti (plastová x kovová krabička). Nakonec byla vybrána kovová (slitina hliníku) konstrukční krabička, protože byla rozměrově ideální, její mechanická odolnost je velmi dobrá a obsahuje drážky na jednoduché vestavění DPS (není nutné zaměřovat montážní díry).

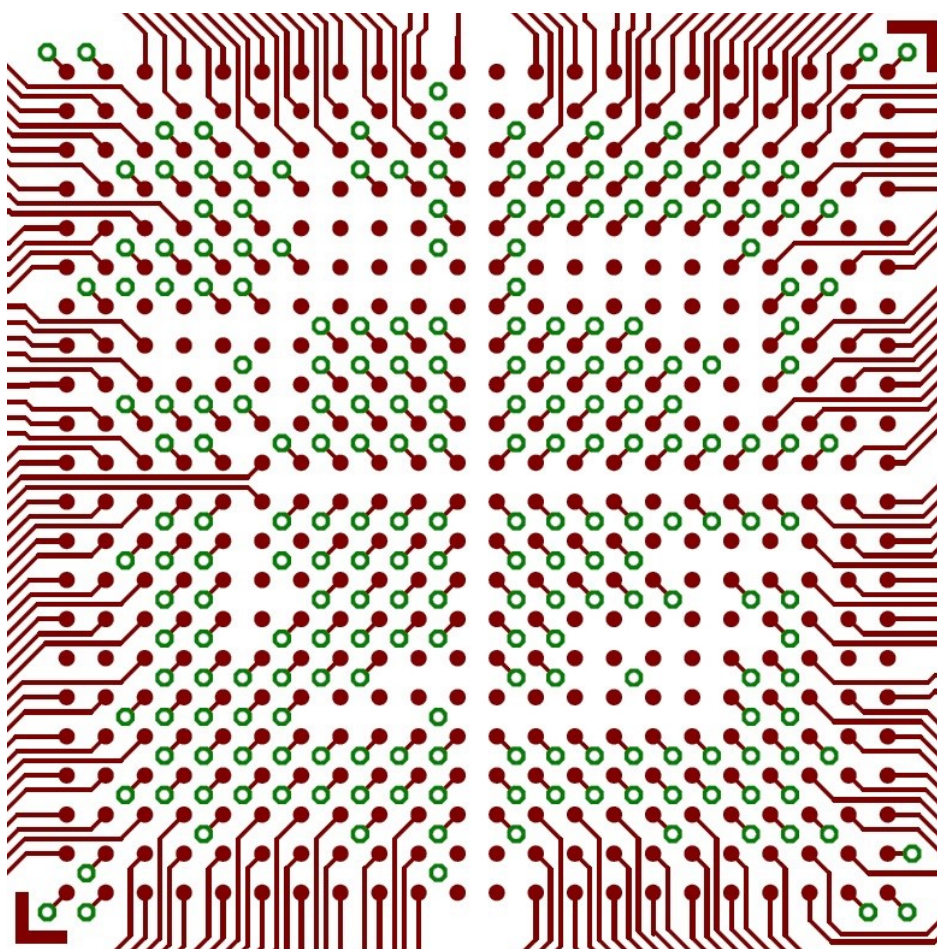
Jakmile byla vybrána konstrukční krabička, tak bylo možné změřit její přesné rozměry a přenést je do návrhového editoru. S tímto bodem dále souvisí rozmístění jednotlivých komponent, které by měly být uživateli přístupné (např. konektory, tlačítka, LED, atd.).

Dalším krokem bylo rozmístění zbytku součástek a prohazování (swap) pinů FPGA a MCU tak, aby bylo následné propojení co nejsnazší. Nad tímto bodem má smysl strávit dostatek času. Dobré rozmístění součástek je předpoklad kvalitního návrhu.

Po rozmístění součástek přišel na řadu další velmi důležitý krok, kterým byla volba třídy přesnosti a počtu vrstev DPS. Jelikož jsou v zapojení použity 3 součástky s BGA pouzdry (rozteče kuliček 1 mm, 0.8 mm a 0.75 mm), bylo zvoleno řešení s využitím 6 vrstev a třídy přesnosti, která umožňuje minimální velikost vrtaného (prokoveného) otvoru 10 mil a minimální šířku vodiče resp. minimální šířku izolační mezery 4 mil.

Všechny tyto zvolené údaje (a spousta dalších) byly přeneseny do návrhového editoru. Tento krok by bylo možné provést kdykoliv dříve.

Součástky s BGA pouzdry jsou pro návrh nejkritičtější. Z tohoto důvodu jim byla věnována zvýšená pozornost. Prvním krokem byla analýza BGA pouzdra a volba velikosti prokovů. Dalším krokem byla volba způsobu vyvedení spojů mimo pouzdro tak, aby bylo použito co nejméně vrstev a nebylo nutné použít utopené prokovy (velké prodražení výroby). Metodika vyvedení spojů je dobře viditelná na obr. 3.23. První 3 vnější řady BGA pouzdra FPGA bylo možné vyvést bez použití prokovů, ostatní řady bylo nutné s využitím prokovů vyvést v jiné vrstvě. Pro přesné vystředění prokovů mezi pájecí plošky byl využit rastr, jehož velikost byla přesně poloviční, než byla rozteč pájecích kuliček zpracovávaného BGA pouzdra (pro rozteč kuliček 1 mm byl nastaven rastr 0.5 mm). Napájecí piny nebylo nutné vyvádět vzhledem k předpokládanému využití 2 vrstev určených pro rozvod napájení (první vrstva GND a druhá vrstva ostatní napěťové větve).



Obr. 3.23 Vyvedení spojů z FPGA (horní vrstva)

Po vyvedení použitých signálů ze součástek v BGA pouzdrech bylo možné přejít na samotné propojení součástek. Většina spojů na DPS včetně izolačních mezer má šířku 8 mil. Pouze na kritických místech s vyšší hustotou propojení bylo nutné klesnout na šířku 6 resp. 4 mil. Je samozřejmostí, že výkonové spoje (např. napájení) byly rozvedeny s šířkou odpovídající proudu tekoucímu těmito spoji. Je nutné si uvědomit, že existuje hranice, do níž lze spoj považovat za obvod se soustřednými parametry a za kterou je nutné spoj uvažovat jako obvod s rozprostřenými parametry (vedení). Jednoduché pravidlo je, že spoj se chová jako vedení, pokud je jeho délka výrazně větší, než vlnová délka signálu na něm přenášeném. Metalické vedení je charakterizováno primárními a sekundárními parametry. Charakteristická impedance je jedním ze dvou sekundárních parametrů a je jí nutné při návrhu uvažovat. Ideální případ nastává, pokud je celá přenosová cesta impedančně přizpůsobená. Pokud tomu tak není, tak na impedančních nerovnostech dochází k odrazům energie a tedy ke zhoršení přenosových vlastností. Nutná podmínka pro impedanční přizpůsobení číslicového spoje je, že když dvojnásobek doby zpoždění signálu ve vedení je větší, než délka jeho náběžné nebo sestupné hrany, tak je nutné tento spoj impedančně přizpůsobit. Prakticky lze spoje rozřadit do tříd podle technologií, které propojují (např. TTL, CMOS, HCMOS, atd.). Existuje několik obvodových řešení pro impedanční přizpůsobování. Každé z nich je vhodné pro jiný případ. Velmi často používané je impedanční přizpůsobení s využitím sériově zapojeného rezistoru. Níže je uveden příklad výpočtu nutnosti impedančního přizpůsobení jednoho z nejdelších spojů na navrhnuté DPS.

Spoj je dlouhý 90 mm, je zatížený jedním vstupem o kapacitě 6 pF a je proveden ve třídě přesnosti 5. Doba trvání vstoupné i sestupné hrany je 3 ns. Spoj je veden v horní vrstvě spojů, která je izolována od rozlité plochy zemně dvěma vrstvami prepregu 1080. Je nutné tento spoj impedančně přizpůsobovat?

Charakteristická impedance spoje je:

$$Z_0 = \frac{120 \cdot \pi \cdot \Omega}{\sqrt{K_{L2} \cdot K_{C2} \cdot \sqrt{\epsilon_r}}} \cdot \left(\frac{h}{w}\right) = 60.1 \Omega,$$

kde Z_0 je charakteristická impedance spoje $[Z_0] = \Omega$, $K_{L2} \approx 1 + 1.5 \cdot \frac{h}{w}$ je pomocná konstanta $[K_{L2}] = 1$, $K_{C2} \approx 1 + \frac{h}{w}$ je pomocná konstanta $[K_{C2}] = 1$, $\epsilon_r = 5.1$ je

relativní permitivita prepregu $[\varepsilon_r] = 1$, $h = 0.13 \text{ mm}$ je tloušťka vrstvy prepregu $[h] = \text{mm}$ a $w = 0.2 \text{ mm}$ je šířka spoje $[w] = \text{mm}$.

Měrné zpoždění průchodu signálu bez kapacitní zátěže je:

$$t_{pd} = 3.33 \cdot 10^{-9} \text{ s} \cdot \text{m}^{-1} \cdot \sqrt{\varepsilon_r} \cdot \sqrt{\frac{K_{C2}}{K_{L2}}} = 6.9 \cdot 10^{-9} \text{ s} \cdot \text{m}^{-1},$$

kde t_{pd} je měrné zpoždění průchodu signálu bez kapacitní zátěže $[t_{pd}] = \text{s} \cdot \text{m}^{-1}$, $K_{L2} \approx 1 + 1.5 \cdot \frac{h}{w}$ je pomocná konstanta $[K_{L2}] = 1$, $K_{C2} \approx 1 + \frac{h}{w}$ je pomocná konstanta $[K_{C2}] = 1$ a $\varepsilon_r = 5.1$ je relativní permitivita prepregu $[\varepsilon_r] = 1$.

Měrná zatěžovací kapacita je:

$$\frac{C_d}{l} = 6.67 \cdot 10^{-11} \text{ F} \cdot \text{m}^{-1},$$

kde $\frac{C_d}{l}$ je měrná zatěžovací kapacita $[\frac{C_d}{l}] = \text{F} \cdot \text{m}^{-1}$, $C_d = 6 \cdot 10^{-12} \text{ F}$ je vstupní kapacita $[C_d] = \text{F}$ a $l = 0.09 \text{ m}$ je délka spoje $[l] = \text{m}$.

Měrná kapacita spoje je:

$$\frac{C}{l} = \varepsilon_0 \cdot \varepsilon_r \cdot K_{C2} \cdot \left(\frac{w}{h}\right) = 1.15 \cdot 10^{-10} \text{ F} \cdot \text{m}^{-1},$$

kde $\frac{C}{l}$ je měrná kapacita spoje $[\frac{C}{l}] = \text{F} \cdot \text{m}^{-1}$, $\varepsilon_0 = 8.85 \cdot 10^{-12} \text{ F} \cdot \text{m}^{-1}$ je permitivita vakua $[\varepsilon_0] = \text{F} \cdot \text{m}^{-1}$, $\varepsilon_r = 5.1$ je relativní permitivita prepregu $[\varepsilon_r] = 1$, $K_{C2} \approx 1 + \frac{h}{w}$ je pomocná konstanta $[K_{C2}] = 1$, $w = 0.2 \text{ mm}$ je šířka spoje $[w] = \text{mm}$ a $h = 0.13 \text{ mm}$ je tloušťka vrstvy prepregu $[h] = \text{mm}$.

Měrné zpoždění průchodu signálu kapacitně zatíženým vedením je:

$$t'_{pd} = t_{pd} \cdot \sqrt{1 + \frac{C_d}{C}} = 8.65 \cdot 10^{-9} \text{ s} \cdot \text{m}^{-1},$$

kde t'_{pd} je měrné zpoždění průchodu signálu kapacitně zatíženým vedením $[t'_{pd}] = \text{s} \cdot \text{m}^{-1}$, $t_{pd} = 6.9 \cdot 10^{-9} \text{ s} \cdot \text{m}^{-1}$ je měrné zpoždění průchodu signálu bez kapacitní

zátěže $[t_{pd}] = s \cdot m^{-1}$, $\frac{C_d}{l} = 6.67 \cdot 10^{-11} F \cdot m^{-1}$ je měrná zatěžovací kapacita $[\frac{C_d}{l}] = F \cdot m^{-1}$ a $\frac{C}{l} = 1.15 \cdot 10^{-10} F \cdot m^{-1}$ je měrná kapacita spoje $[\frac{C}{l}] = F \cdot m^{-1}$.

Maximální délka spoje bez impedančního přizpůsobení je:

$$L_{max} = \frac{t_s}{2 \cdot t'_{pd}} = 0.174 m = 174 mm,$$

kde L_{max} je maximální délka spoje bez impedančního přizpůsobení $[L_{max}] = m$, $t_s = \min\{t_r, t_f\} = 3 \cdot 10^{-9} s$ (t_r je minimální doba trvání vzestupné hrany a t_f je minimální doba trvání sestupné hrany) $[t_s] = [t_r] = [t_f] = s$ a $t'_{pd} = 8.65 \cdot 10^{-9} s \cdot m^{-1}$ je měrné zpoždění průchodu signálu kapacitně zatíženým vedením $[t'_{pd}] = s \cdot m^{-1}$.

Z výsledku je patrné, že spoj není nutné impedančně přizpůsobovat! Více informací je možné nalézt ve skriptech (Záhlava, 2005).

Jak již bylo dříve zmíněno, tak dvě vrstvy jsou vyhrazeny pro rozvod napájení (první vrstva GND a druhá vrstva ostatní napěťové větve). V těchto vrstvách obvykle bývá rozlité měděná plocha připojená na daný potenciál. Tento způsob řešení je velmi výhodný z několika důvodů:

- přivedení napájení k součástkám je velmi jednoduché,
- vzhledem k rozlité ploše je definována charakteristická impedance spojů,
- sousedící rozlité plochy tvoří kvalitní blokovací kondenzátor,
- minimalizace proudových smyček.

Pro zachování těchto výhod je nutné se vyvarovat tzv. syndromu švýcarského sýra. Toto pravidlo naráží na potenciální rozdělení měděné plochy na více segmentů, které by mohlo vzniknout špatným návrhem umístění a velikosti otvorů v DPS (prokovy, pájecí plošky). Občas je toto pravidlo těžké splnit (velká hustota propojení), proto je nutné nalézt optimální řešení (je možné pravidlo porušit, pokud tím nebude způsoben nějaký závažný problém).

Pro omezení vyzářování DPS je vhodné dodržovat tzv. pravidlo 20H. Význam pravidla spočívá v dodržení podmínky, že vodivá plocha GND musí přesahovat napájecí plochu a signálové spoje alespoň o dvacetinásobek jejich vzájemné vzdálenosti. Při dodržení

této podmínky je možné minimalizovat vyzařování DPS do strany až o 70 % (pro lepší pochopení, je dobré si představit siločáry elektrostatického pole na okraji deskového kondenzátoru pro různé posunutí desek v jedné zvolené ose).

Nakonec se podařilo DPS navrhout „pouze“ čtyřvrstvou místo předpokládané šestivrstvé!

Po dokončení spojů bylo nutné přidat naváděcí značky pro osazovací automat (osazení součástek s BGA pouzdry). Pravidla pro naváděcí značky se liší pro různé osazovací technologie. V tomto případě bylo nutné přidat dvě naváděcí značky, které měly tvar plného kruhu o průměru 1.5 mm v odmaskovaném kruhu o průměru 2.5 mm. Ideální rozmístění značek je v protilehlých rozích DPS (maximalizace vzájemné vzdálenosti) alespoň 4 mm od okraje DPS.

Finální kroky při dokončování návrhu byly:

- rozmístění potisku,
- vygenerování seznamu součástek,
- vygenerování výrobních dat pro výrobu DPS,
- vygenerování výrobních dat pro výrobu síťotiskové planžety (tenká kovová maska).

Výsledný návrh zde nemá smysl uvádět. Kompletní projekt je uložen na přiloženém CD.

3.18 Výroba

Prvním krokem při výrobě zařízení bylo vyrobení DPS. Navrhnutá DPS splňovala podmínky pro výrobu formou „POOL servis“ ve společnosti PragoBoard s.r.o. Výroba trvala 7 pracovních dnů a výsledek byl velmi dobrý.

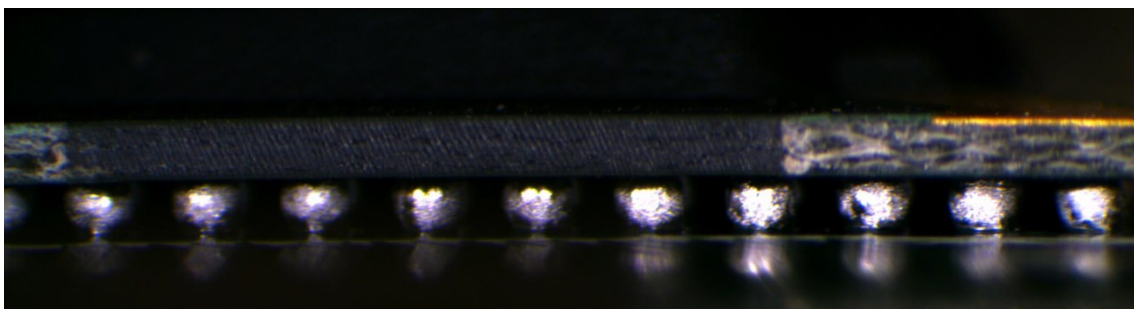
Druhým krokem bylo samotné osazení součástek, které bylo ztíženo použitím součástek s BGA pouzdry. Pro bezproblémové osazení byla opět zvolena spolupráce s profesionální firmou Certuma spol. s.r.o. Bylo nutné nechat vyrobít síťotiskovou planžetu (opět ve firmě PragoBoard s.r.o.) a provést nutnou přípravu (vysušení) součástek v BGA pouzdech. Pokud by se vysušení neprovedlo, mohlo by dojít k tzv. popcorn efektu (potenciální zkrat mezi cínovými kuličkami). Další součástka, která je velmi náchylná na množství vlhkosti při procesu přetavení, je LED dioda. Vzhledem

k ceně a způsobu osazení LED diod nebyl tento fakt uvažován. Sítotisková šablona byla úmyslně navržena tak, aby bylo možné osadit všechny SMD součástky z horní strany DPS. Samotné osazení všech součástek (kromě součástek v BGA pouzdrech) provedl autor této práce.

Postup osazení DPS:

- sítotisk pájecí pasty horní strany DPS,
- manuální umístění SMD součástek (kromě součástek v BGA pouzdrech) na horní straně DPS do pájecí pasty,
- přesné umístění součástek v BGA pouzdrech pomocí automatu,
- přetavení v průběžné přetavovací peci,
- kontrola připájení pod mikroskopem,
- osazení zbytku součástek (SMD na spodní straně DPS a součástky s vývody) pomocí mikropáječky.

Na obr. 3.24 je uvedeno připájení BGA pouzdro FPGA.



Obr. 3.24 Připájené FPGA

Kompletní osazená DPS je uvedena na obr. 3.25.

Jakmile byla DPS kompletně osazena, mohlo se přejít k otestování funkčnosti.

Postup otestování funkčnosti:

- otestování zdroje (použití regulovatelného napájecího zdroje s nastaveným proudovým omezením),
- připojení funkčního zdroje do zbytku zařízení pomocí SMD propojek (použití regulovatelného napájecího zdroje s proudovým omezením),
- nahrání testovací firmware do MCU pomocí JTAG (otestování připojených periférií),

- nahrání testovacího návrhu do FPGA pomocí JTAG a testovacího programu do MCU pomocí JTAG (otestování propojení MCU s FPGA, otestování připojených periférií).

Testovacím firmware pro MCU se zabývá následující kapitola.



Obr. 3.25 Osazená DPS

K návrhu byl využit program Eagle 6.5.0. Informace o tomto návrhovém editoru byly získány praxí a z publikace (Plíva, 2010). Mnoho cenných znalostí použitých při návrhu bylo získáno ze skript (Záhlava, 2005) a při konzultaci s Ing. Vítem Záhlavou, CSc.

4 Vývoj firmwaru

Další částí práce je implementace referenční konfigurace jednotlivých řadičů. Tato část práce je minoritní a klade si za účel otestování funkčnosti navrhnutého HW tak, aby ho bylo možné předat dalším kolegům, kteří obohatí zařízení o potřebnou inteligenci (funkčnost).

4.1 Vývojové nástroje

Podpora TI pro tuto platformu MCU je velmi dobrá. TI zdarma nabízí vývojové prostředí Code Composer Studio (dále CCS), které je založené na vývojovém prostředí Eclipse. Psaní programu v tomto IDE je velmi intuitivní a komfortní. IDE nabízí spoustu zajímavých funkcí (včetně verzování projektu např. v Git), které mohou být využity při psaní samotného programu.

Dalším výborným vývojovým nástrojem, který TI zdarma nabízí jako podporu pro vývoj na této platformě, je program HALCoGen. HALCoGen je program s grafickým ovládáním (front-end), který umožňuje nastavení samotného MCU dle požadavků (např. hodinové domény, povolení jednotlivých periférií, nastavení jednotlivých periférií, atd.) a z daného nastavení je schopný vygenerovat ovladače (HAL) jednotlivých periférií včetně aplikačního rozhraní (API) pro vyšší vrstvy firmware. Velkou výhodou tohoto řešení jsou záplaty (patch) chyb, které by jinak uživatel musel hledat v seznamu chyb (errata) a následně opravovat vlastnoručně.

Užitečným vývojovým nástrojem pro tuto platformu je Hercules Development Kit TMS570 MCU (dále HDK), který má integrovaný stejný MCU, jaký je použitý v této práci. Vývojový kit dále obsahuje integrovaný programátor XDS100v2, budiče CAN, LED diody a další periférie.

Posledním vývojovým nástrojem použitým v této práci je programátor J-LINK od firmy SEGGER. V základní instalaci CCS není tento programátor podporován. TI ovšem nabízí zásuvný modul (plugin), kterým lze podporu pro J-LINK do CCS doinstalovat.

4.2 CAN

V této podkapitole bude popsán postup vývoje programu, který se chová jako „opakováč“ na sběrnici CAN s přenosovou rychlostí 500 kb/s. Funkčnost programu je

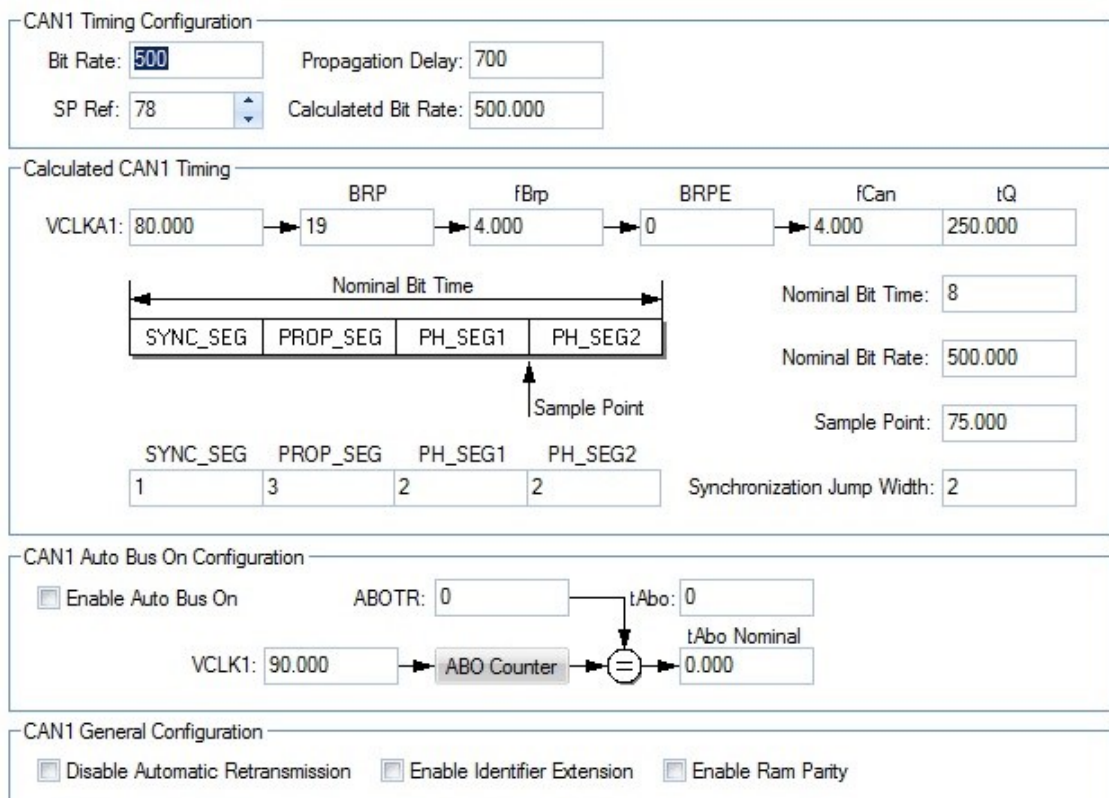
taková, že přijímá CAN zprávy s ID = 1. Datový obsah těchto zpráv následně odešle jako CAN zprávu s ID = 2.

Vývoj začíná vytvořením nového projektu v programu HALCoGen a provedením základního nastavení procesoru. Druhým krokem je samotné nastavení CAN periferie tak, aby byla schopná komunikace na CAN sběrnici podle normy CAN 2.0A s přenosovou rychlostí 500 kb/s. V programu (záložka TMS570LS3137ZWT – Driver Enable) je nejdříve nutné zvolit, jaké periferie chceme používat. V tomto případě byla zvolena periferie CAN1 viz obr. 4.1.



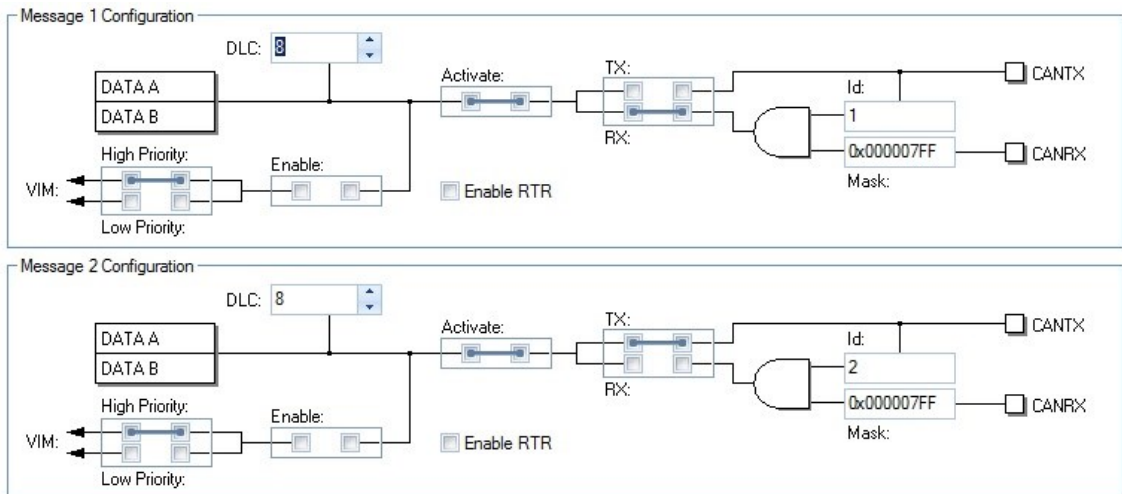
Obr. 4.1 Povolení periferie CAN1

Pro globální nastavení CAN1 periferie je nutné se přesunout do záložky CAN1 viz obr. 4.2.



Obr. 4.2 Globální nastavení periferie CAN1

Dalším krokem je nastavení jednotlivých „Message Bufferů“ pro příjem/odesílání zpráv se zvoleným ID. Toto nastavení je vidět na obr. 4.3. Důležitá poznámka je, že byla zvolena metoda polling. Druhou možností by bylo použití přerušení.



Obr. 4.3 Nastavení "Message Bufferů" periférie CAN1

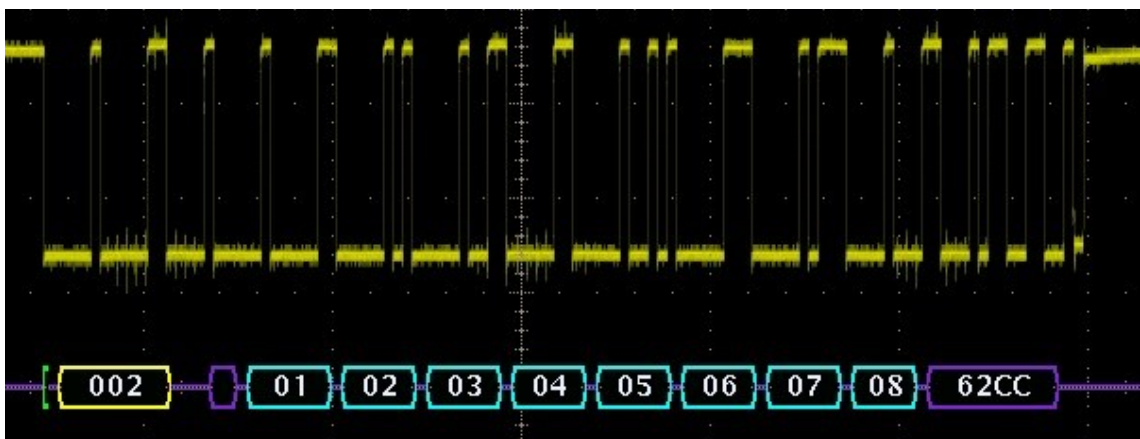
Po provedeném nastavení byly vygenerovány zdrojové soubory a vytvořen projekt v CCS. Samotný zdrojový kód je uveden níže.

```

canInit();
while(1)
{
    if(canIsRxMessageArrived(canREG1, canMESSAGE_BOX1))
    {
        canGetData(canREG1, canMESSAGE_BOX1, canData);
        while(canIsTxMessagePending(canREG1, canMESSAGE_BOX2));
        canTransmit(canREG1, canMESSAGE_BOX2, canData);
    }
}

```

Program byl otestován s využitím CAN Explorer modulu, který lze pro testování nastavit podle výše uvedených požadavků. Na obr. 4.4 je vidět vysílání CAN1 periférie zachycené na osciloskopu.



Obr. 4.4 Zachycený CAN rámec na osciloskopu

4.3 LIN

Pro otestování periférie LIN byl napsán program, který na jednom modulu má roli LIN master zařízení a na druhém modulu (vývojový kit) má roli LIN slave zařízení. Přenosová rychlost byla zvolena 19.2 kb/s.

Pomocí programu HALCoGen byl opět vygenerován ovladač pro LIN periférii. Následně byl vývoj dokončen v CCS. Postup je obdobný jako u CANu.

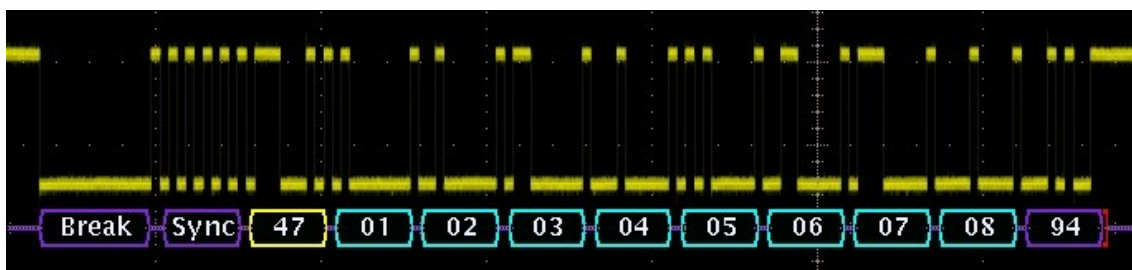
Naprogramovaný LIN master má funkčnost takovou, že periodicky vysílá hlavičku s ID = 0, 1, ..., 15 včetně dvou paritních bitů (PD0 a PD1), které je nutno spočítat softwarově. Po odeslání hlavičky čeká na odpověď od LIN slave zařízení, které je zodpovědné za vysílání rámce s daným ID. Zdrojový kód je uveden níže.

```
linInit();
while(1)
{
    linIDP0 = 0x01 & (linID^(linID>>1)^(linID>>2)^(linID>>4));
    linIDP1 = 0x01 & (0x01^(linID>>1)^(linID>>3)^(linID>>4)^(linID>>5));
    linSendHeader(linREG, (linIDP1 << 7) | (linIDP0 << 6) | linID);
    while(1)
    {
        if(!linIsRxReady(linREG))
        {
            linGetData(linREG, linData);
            linID = (linID + 1) % 16;
            break;
        }
        else if(linREG->FLR & LIN_NRE_INT)
        {
            linREG->FLR |= LIN_NRE_INT;
            break;
        }
    }
}
```

Naprogramovaný LIN slave má funkčnost takovou, že pomocí příznakového bitu kontroluje úspěšný příjem hlavičky. Po úspěšném příjmu hlavičky je porovnán přijatý ID s ID zprávy, kterou slave posílá. Při rovnosti těchto dvou identifikátorů jsou odeslána data na sběrnici. Zdrojový kód je uveden níže.

```
linInit();
while(1)
{
    if((linREG->FLR & 0x00004000) && (linGetIdentifier(linREG) & 0x3F)==linID)
    {
        linSetLength(linREG, sizeof(linData));
        linSend(linREG, linData);
    }
}
```


LIN periférie nabízí možnost generování přerušení z různorodých příčin (např. příjem ID, který prošel vstupním filtrem). Použití přerušení je tedy velmi dobrou volbou. V tomto testovacím kódu byla použita metoda polling. Funkčnost komunikace je vidět z obr. 4.5.



Obr. 4.5 Zachycený LIN rámec na osciloskopu

4.4 Ethernet

Implementace referenční konfigurace Ethernet řadiče (dále EMAC) je výrazně složitější než u předchozích periférií. Výhodou bylo, že autor mohl navázat na svou bakalářskou práci (Blecha, 2012), v níž byla tato implementace řešena. Další výhodou bylo, že TI má hotový referenční projekt, ve kterém je na použitém MCU zprovozněn LwIP TCP/IP zásobník. Cílem bylo vyzkoušet připojení vyrobeného modulu do sítě Ethernet a zjistit, jaké lze očekávat parametry. Vzhledem k tomu, že v bakalářské práci byl použit LwIP verze 1.4.0 a aktuálně nejnovější je LwIP verze 1.4.1, byl proveden přechod na tuto vyšší verzi. Další rozdílem od implementace v bakalářské práci bylo využití rozhraní RMII místo rozhraní MII.

Potřebný HAL kód opět poskytl program HALCoGen, LwIP zásobník je možné zdarma stáhnout z webových stránek. LwIP je šířeno pod licencí BSD. Ovladač pro EMAC periférii byl převzat a upraven z referenčního projektu od TI. Změny se týkaly správného nastavení budiče a EMAC pro vzájemnou komunikaci pomocí RMII rozhraní. LwIP zásobník obsahuje většinu standardně používaných protokolů (např. UDP, TCP, DHCP, ...). Použití těchto protokolů lze povolit/zakázat v hlavičkovém souboru *lwipopts.h*. V tomto souboru je možné také nakonfigurovat LwIP zásobník podle představy. Pro základní otestování funkčnosti komunikace je vhodný ICMP protokol s použitím datagramů „Echo Request“ a „Echo Reply“.

Pro otestování propustnosti sítě byl napsán jednoduchý program umožňující komunikaci pomocí protokolu UDP. Zdrojový kód části programu je uveden níže. Uvedená funkce byla v programu periodicky volána s následujícími parametry:

- char *data – pole dat dlouhé 1500 B,
- uint16_t length – délka platných dat 1450 B.

```

err_t server_send(char *data, uint16_t length)
{
    struct pbuf *p;
    err_t ret_val;

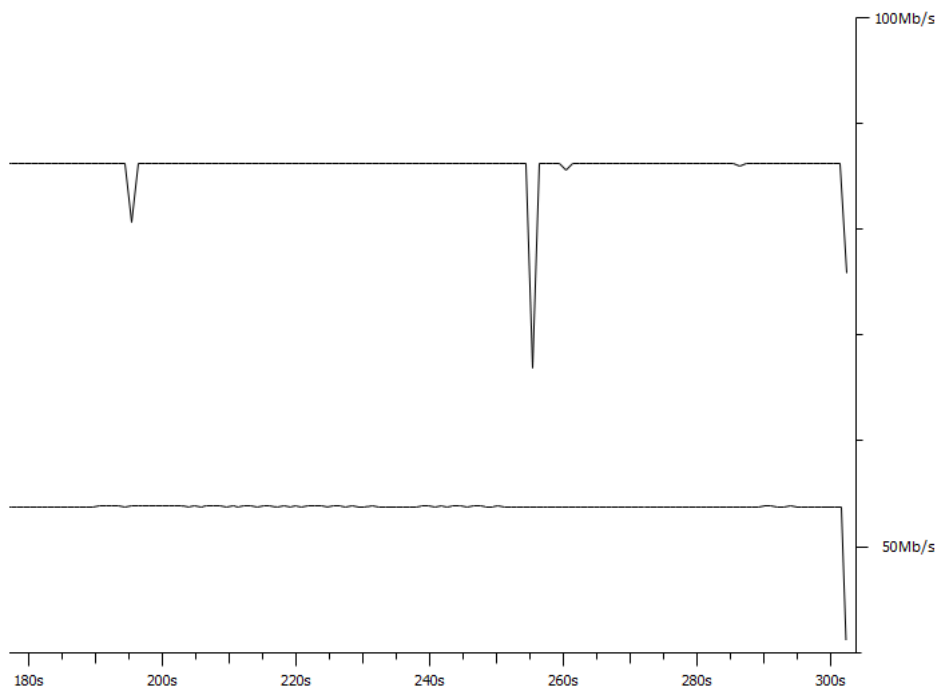
    if(data_sent)
    {
        p = pbuf_alloc(PBUF_TRANSPORT, length, PBUF_REF);
        if(p == NULL)
            return ERR_BUF;

        p->payload = data;
        data_sent = 0;
        ret_val = udp_send(server_pcb, p);
        pbuf_free(p);
    }
    else
        ret_val = ERR_USE;

    return ret_val;
}

```

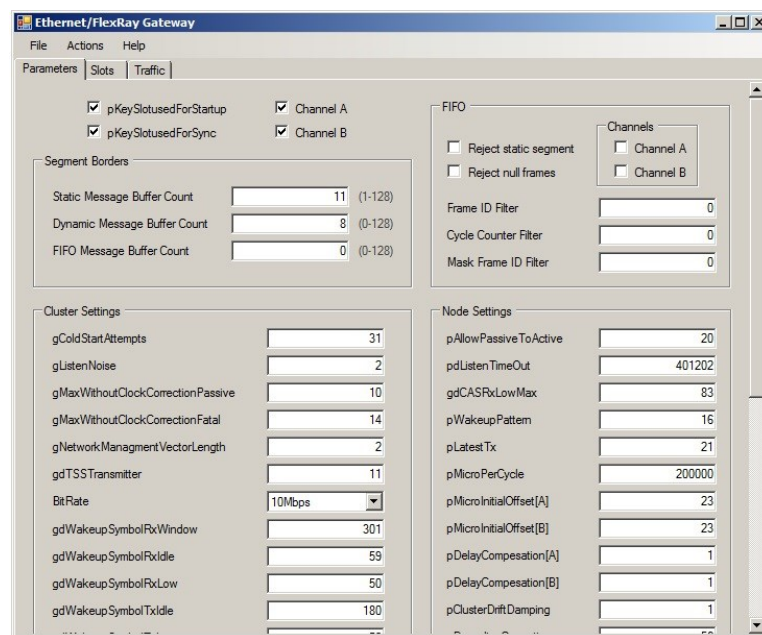
Tímto bylo dosaženo periodické odesílání rámců o velikosti 1492 B a tudíž i velkého vytížení sítě. Zajímavostí je, že při zapnutém počítání kontrolního součtu u UDP datagramu byla výsledná propustnost cca 53 Mb/s. Po vypnutí počítání ochranného součtu UDP datagramu vzrostla propustnost cca na 85 Mb/s (zhruba horní limit propustnosti sítě Fast Ethernet). Graf propustnosti je uveden na obr. 4.6.



Obr. 4.6 Propustnost FE připojení

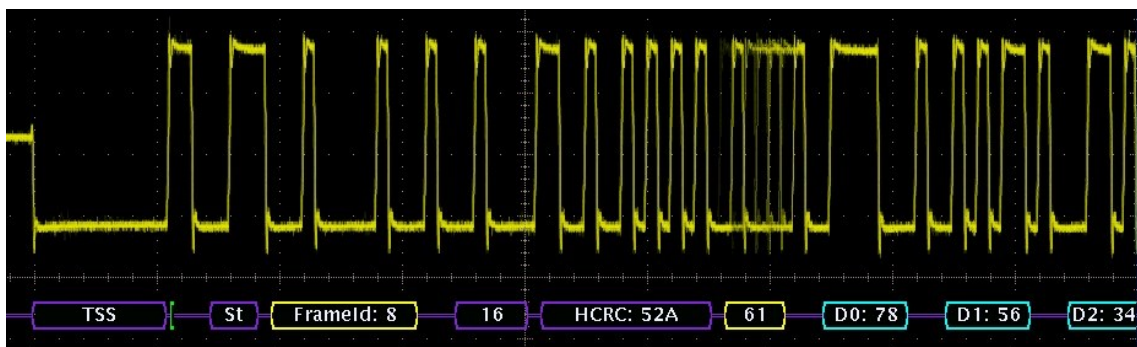
4.5 FlexRay

Posledním rozhraním, které bylo nutno otestovat, je FlexRay. Pro řadič obsažený ve vybraném MCU byla implementována knihovna v rámci bakalářské práce (Zeman, 2012). Stejný autor dále navrhl konfigurační protokol CEP, který využívá zmíněnou knihovnu a je schopný provést konfiguraci FlexRay řadiče dle předaných parametrů. K nastavování FlexRay řadiče s pomocí CEP přes síť Ethernet byla vyvinuta v rámci bakalářské práce (Blecha, 2012) aplikace pro PC. S pomocí této aplikace lze řadič nakonfigurovat podle potřeby. Aplikace je zobrazena na obr. 4.7.



Obr. 4.7 Ovládací PC aplikace

Ideálním způsobem, jak FlexRay periférii v MCU otestovat, bylo s využitím výše zmíněných pomůcek. O správnou konfiguraci MCU se opět postaral program HALCoGen. Zdrojový kód je uložen na příloženém CD. Část FlexRay komunikace je zobrazena na obr. 4.8.



Obr. 4.8 Zachycený FlexRay rámec na osciloskopu

5 Závěr

Cílem této diplomové práce bylo navrhnout a implementovat technické vybavení testeru pro automobilové aplikace. Navržené zařízení mělo splňovat následující požadavky:

- podpora „libovolného“ počtu řadičů linkových protokolů CAN, LIN a FlexRay ve formě IP funkcí,
- 2 fyzická rozhraní FlexRay, 3 fyzická rozhraní CAN, 2 fyzická rozhraní LIN,
- řídicí mikroprocesor s rozhraním Ethernet,
- podpora vzdálené rekonfigurace hradlového pole FPGA,
- implementace referenční konfigurace řadičů a jejich programové obsluhy.

Podpora libovolného počtu řadičů linkových protokolů CAN, LIN a FlexRay ve formě IP funkcí samozřejmě není možná. Počet jednotlivých řadičů byl po zvážení zvolen na 5 CAN, 5 LIN a 3 FlexRay řadiče. Po provedeném odhadu bylo zjištěno, že FPGA s dostatečnou velikostí pro obsazení výše uvedeného počtu řadičů, je moc drahé. Byl zvolen kompromis levnějšího a menšího FPGA, se stejným pouzdem jako FPGA, které splňuje zmíněné požadavky. V případě nutnosti je tedy možné použít dražší variantu.

Počty fyzických rozhraní jednotlivých komunikačních technologií úzce souvisí s počtem pinů FPGA, ke kterému jsou připojeny, a také s využitým místem v konstrukční krabici. V průběhu návrhu se ukázalo, že FPGA má dostatek volných pinů na připojení 4 CAN, 4 LIN a 2 FlexRay budičů. Jelikož byla konstrukční krabice vybírána po zvolení přesného počtu fyzických rozhraní, nebyl s nedostatkem místa problém.

Řídicí mikroprocesor byl vybírán tak, aby obsahoval řadiče technologie Fast Ethernet, CAN, LIN a FlexRay. Vzhledem k tomu, že mikroprocesor TMS570LS3137 splňuje tyto požadavky, má poměrně dobré výkonnostní parametry a autor s ním má dobré zkušenosti, tak žádný jiný mikroprocesor nebyl vyhledáván. Mikroprocesor dále obsahuje paralelní komunikační rozhraní EMIF, které velmi usnadňuje propojení s FPGA.

Podpora vzdálené rekonfigurace FPGA je zajištěna použitím vybraného mikroprocesoru s rozhraním Ethernet. FPGA podporuje sériové pasivní konfigurační schéma. Tento způsob konfigurace funguje na principu synchronního sériového přenosu dat, mezi

dvěma zařízeními, kde proces konfigurace řídí externí zařízení (v tomto případě mikroprocesor). V návrhu byla integrována hardwarová podpora konfigurace ze strany mikroprocesoru, s použitím rozhraní SPI.

Nad rámec zadání byly do návrhu přidány dvě paměti (RAM a FLASH) a rozhraní RS-232, které je užitečné zejména při vývoji firmware. Dále byla přidána možnost přivedení externího hodinového signálu, externí synchronizace a samozřejmě LED diody.

Implementace referenční konfigurace řadičů a jejich programové obsluhy byla pojata, jako zprovoznění daných řadičů, k možnosti odeslání/příjmu dat. Hlavním účelem bylo zjištění funkčnosti/nefunkčnosti navrženého hardwaru. Budoucí použití jednotlivých periférií a jejich konfigurace záleží na požadavcích aplikace.

Návrh zařízení vyžadoval velkou preciznost, a proto také trval poměrně dlouho. Hodně času bylo stráveno kontrolami zapojení a rozměrů pouzder. Zvýšená pozornost byla věnována součástkám s BGA pouzdry. Jejich použití bylo riskantní z důvodů složitějšího návrhu a složitého osazení, které provádí pouze několik firem v České republice. Firma Certuma spol. s.r.o. nám vyšla vstříc a umožnila bezproblémové osazení všech SMD součástek z horní strany DPS, včetně součástek s BGA pouzdry. Osobní návštěva této společnosti přinesla spoustu nových informací, drahocenné zkušenosti a byla tak velkým přínosem! Výroba celého zařízení (od vygenerování výrobních dat, až po fyzické zařízení) proběhla bez jakýchkoli problémů.

Zprovoznění zařízení bylo rozděleno do několika fází. První fází bylo zprovoznění navrženého zdroje. Při návrhu zdroje bylo provedeno několik simulací jeho funkčnosti, takže byl dobrý předpoklad, že zdroj bude fungovat. Tento předpoklad se vyplnil na 100 %. Dalším krokem bylo připojení zdroje pomocí SMD propojek do zbytku zařízení. Před samotným připojením byl na všech napěťových větvích změřen odpor (předcházení zkratu). Tento krok proběhl bez problémů. Po přivedení napájení do integrovaných obvodů, bylo možné vyzkoušet jejich funkčnost. Testování obsáhlo všechny nosné obvody. Prozatím byla objevena pouze jedna chyba: prohození Rx a Tx signálu na konektoru rozhraní RS-232 (zapojeno jako DTE). Tento problém lze vyřešit např. použitím nulového modemu. Zařízení je tedy připraveno na vývoj firmware.

Jelikož je zařízení čerstvě zprovozněno, tak prozatím neexistuje zpětná vazba od uživatelů, která by poukázala na možné problémy a určila tak směr dalšího vývoje hardwaru.

Koncept zařízení umožňuje dynamickou změnu konfigurace, a tudíž i celkové funkčnosti zařízení. Skvělou vlastností je obsažení hned tří automobilových komunikačních standardů, což umožňuje nadefinovat komplexní chování. Příkladem může být emulace FlexRay/CAN/LIN brány. Dalším příkladem může být modifikátor zpráv (man in the middle). Určitě lze zařízení využít na logování vybraných dat a následně na jejich diagnostiku.

Seznam použitých zkratek

CAN	Controller Area Network
LIN	Local Interconnect Network
FPGA	Field-programmable Gate Array
MCU	Microcontroller Unit
OS	Operating System
IP	Internet Protocol
LwIP	Lightweight IP
LLC	Logical Link Control
MAC	Media Access Control
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CRC	Cyclic Redundancy Check
CSMA/CR	Carrier Sense Multiple Access with Collision Resolution
SCI	Serial Communication Interface
UART	Universal Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface
BGA	Ball Grid Array
EMIF	External Memory Interface
V/V	Vstupně/Výstupní
BIST	Built-In Self-Test
ECC	Error Checking and Correcting
CPU	Central Processing Unit
DMIPS	Dhrystone Million Instructions Per Second
LE	Logical Element
LAB	Logical Array Block
SRAM	Static Random Access Memory
DRAM	Dynamic Random Access Memory
MII	Media Independent Interface
RMII	Reduced Media Independent Interface
SMI	Serial Management Interface
OC	Open Collector
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
DPS	Deska plošných spojů
PLL	Phase Locked Loop
LDO	Low-dropout
PMIC	Power Management Multi-channel IC
ESR	Equivalent Series Resistance
LED	Light Emitting Diode
IDE	Integrated Development Environment
HAL	Hardware Abstraction Layer
API	Application Programming Interface
EMAC	Ethernet Media Access Controller
ICMP	Internet Control Message Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
DHCP	Dynamic Host Configuration Protocol
CEP	Config Exchange Protocol
DTE	Data Terminal Equipment

Literatura

Altera. (2013). Cyclone IV Device Handbook.

Blecha, J. (2012). Převodník Ethernet/FlexRay, technické vybavení a firmware, bakalářská práce ČVUT FEL.

FlexRay Consortium. (2005). FlexRay Communications System Protocol Specification Version 2.1 Revision A.

Kocourek, P., & Novák, J. (2006). *Přenos informace*. Praha: Nakladatelství ČVUT.

LIN Consortium. (2006). LIN specification package Revision 2.2A.

LT. (2011). LT3507A Datasheet.

Malinský, J. (2010). Intrusive Tests in FlexRay Standard, disertační práce ČVUT FEL.

NXP. (2007). TJA1041 Datasheet.

NXP. (2007). TJA1080 Datasheet.

NXP. (2012). 74HC14 Datasheet.

Paták, M. (2010). LIN controller, bakalářská práce ČVUT FEL.

Paták, M. (2012). FlexRay controller, diplomová práce ČVUT FEL.

Plíva, Z. (2010). *EAGLE prakticky*. Praha: BEN - technická literatura.

Renesas. (2009). R1WV6416RBG-5SI Datasheet.

TI. (2008). TPS3808 Datasheet.

TI. (2009). TPIC1021A Datasheet.

TI. (2013). AN-1794 Using RMI Master Mode.

TI. (2013). DP83640 Datasheet.

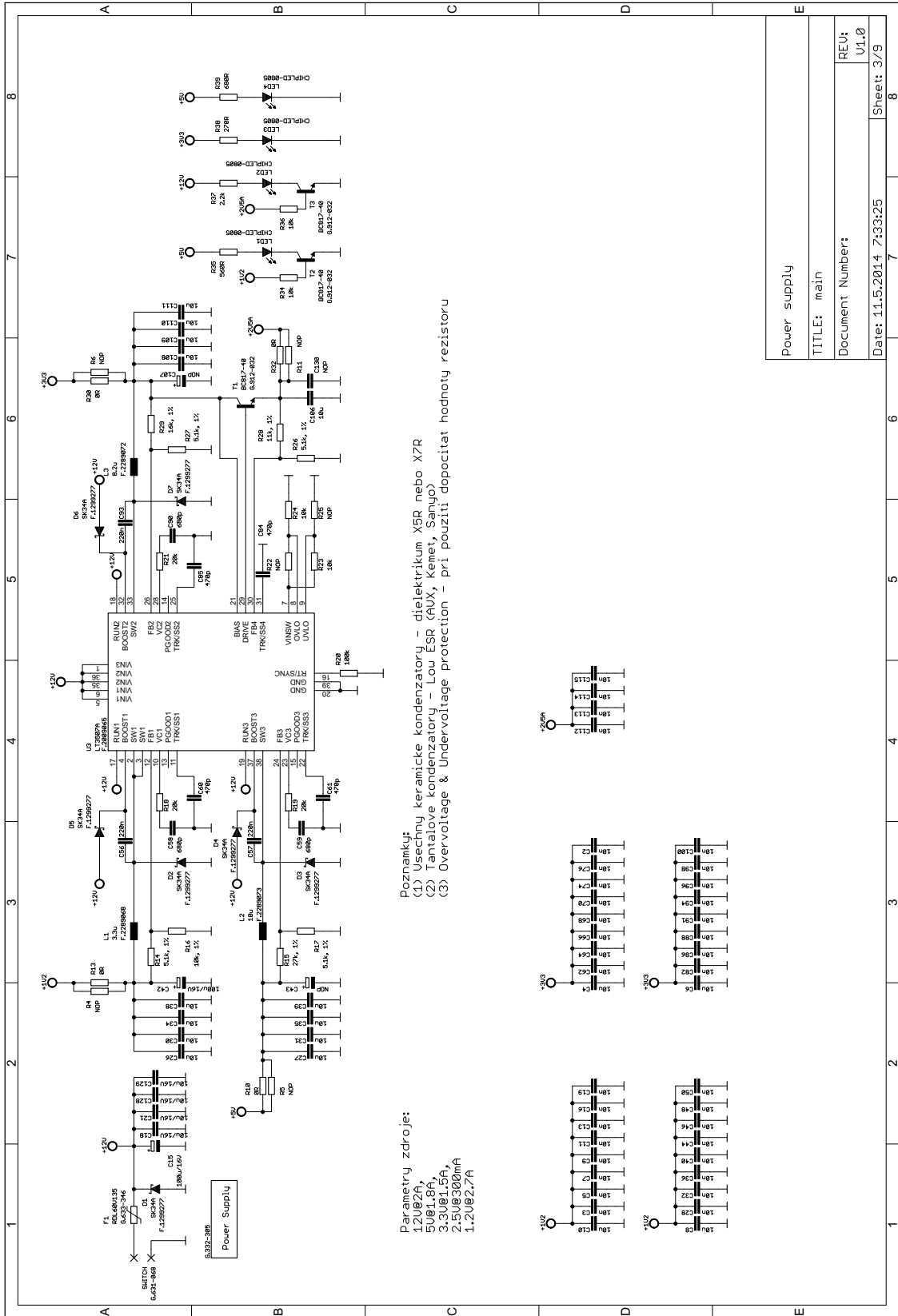
TI. (2013). TMS570LS3137 Datasheet.

TI. (2013). TMS570LS3137 Technical Reference Manual.

TI. (2014). MAX3232 Datasheet.

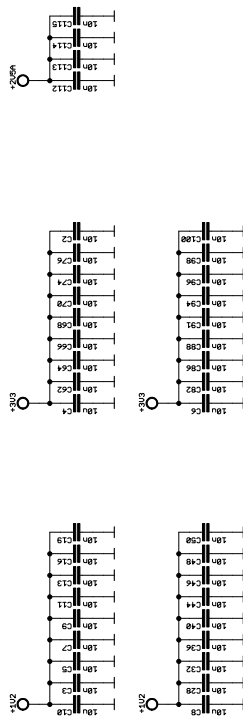
Záhlava, V. (2005). *Návrh a konstrukce desek plošných spojů*. Praha: Nakladatelství ČVUT.

Zeman, M. (2012). Programové vybavení převodníku Ethernet/FlexRay, bakalářská práce ČVUT FEL.

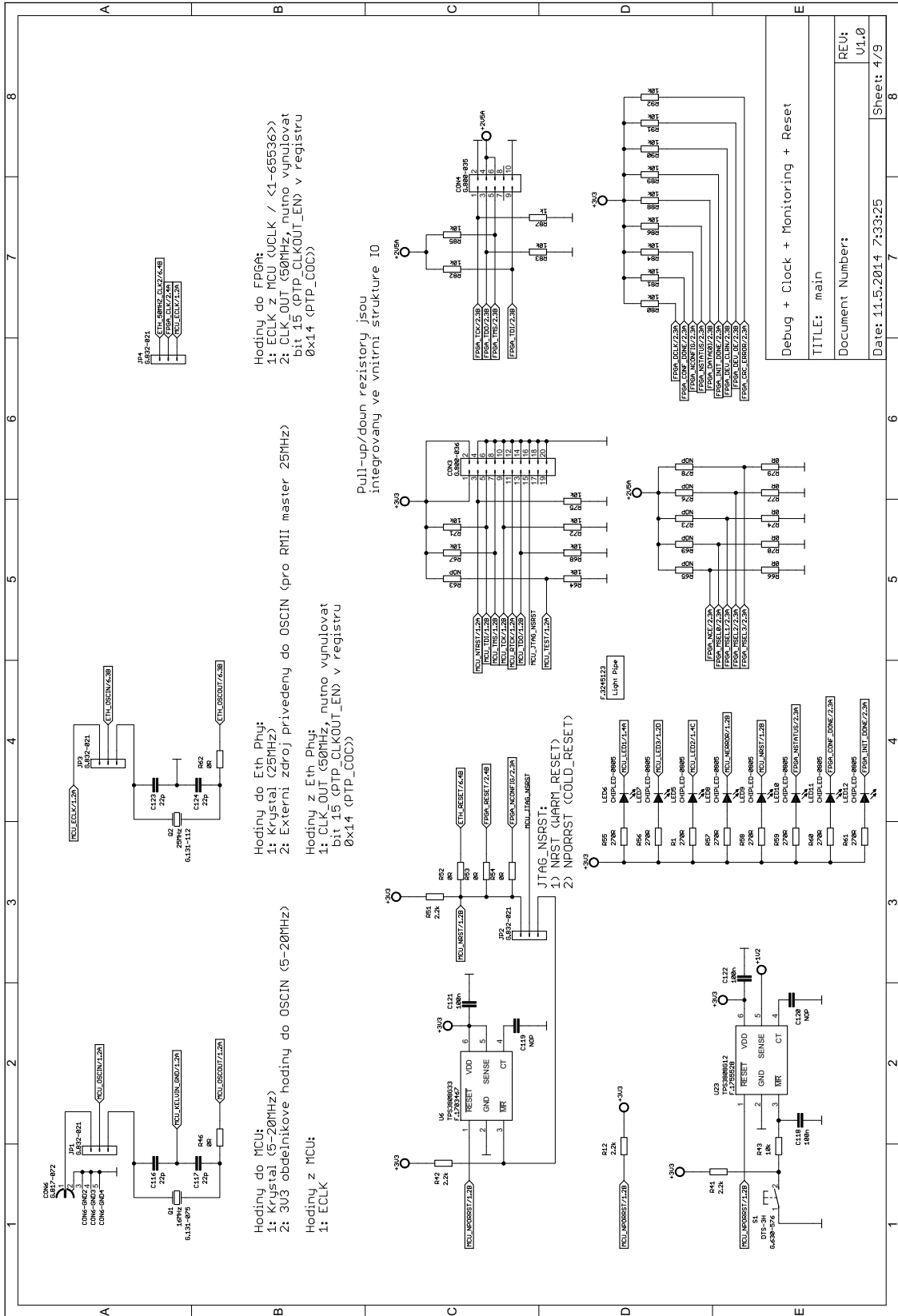


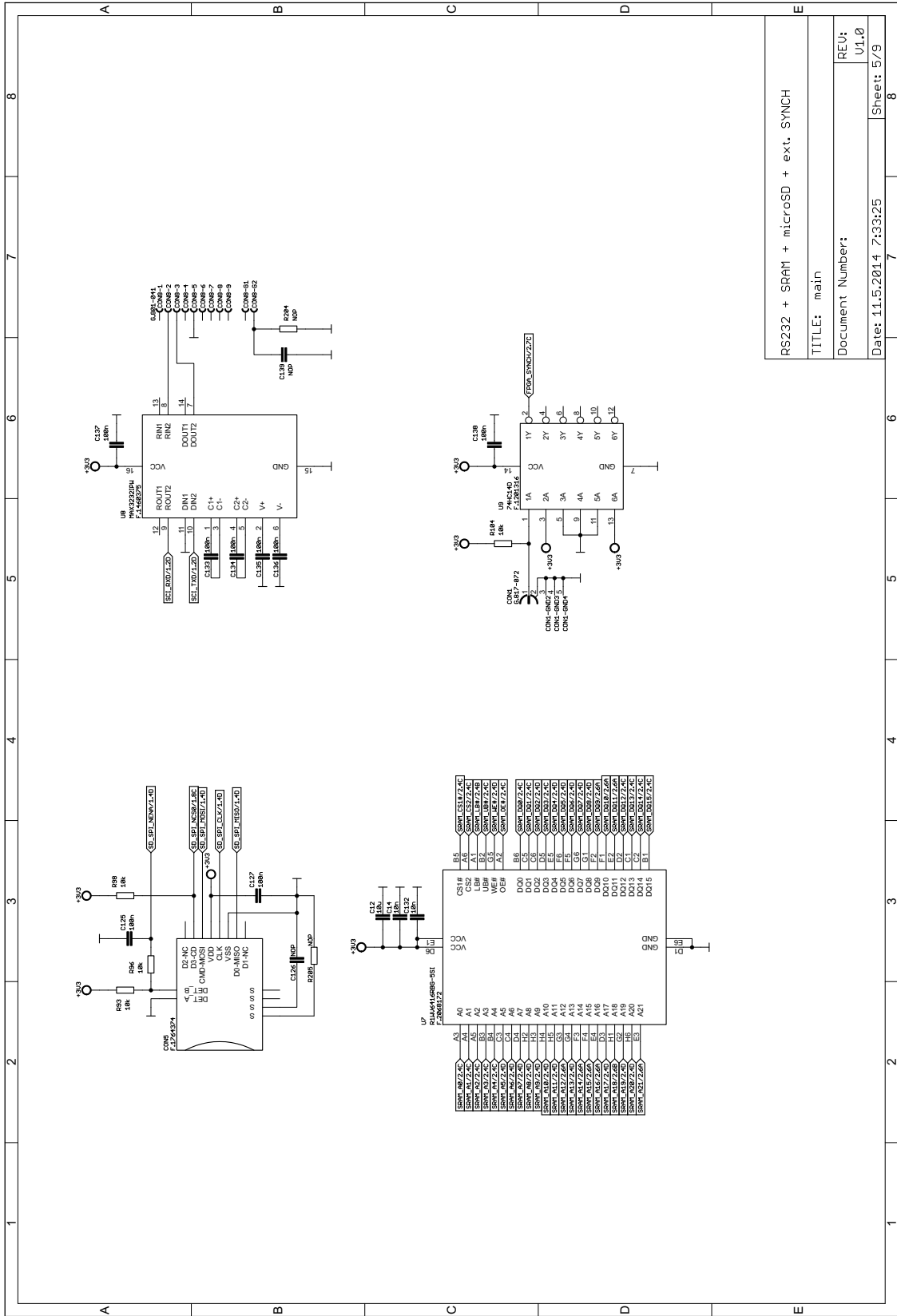
Parametry zdroje:
 12V@2A,
 5V@1.8A,
 3.3V@1.5A,
 2.5V@300mA
 1.2V@2.7A

Poznámku:
 (1) Usechny keramicke kondenzatory – dielektrikum X5R nebo X7R
 (2) Tantálové kondenzatory – Low ESR (AUX, Kemet, Samyo)
 (3) Overvoltage & Undervoltage protection – při použití dopocitat hodnoty rezistoru



Power supply	
TITLE: main	REV: U1.0
Document Number:	Sheet: 3/9
Date: 11.5.2014 7:33:25	8





RS232 + SRAM + microSD + ext. SYNC

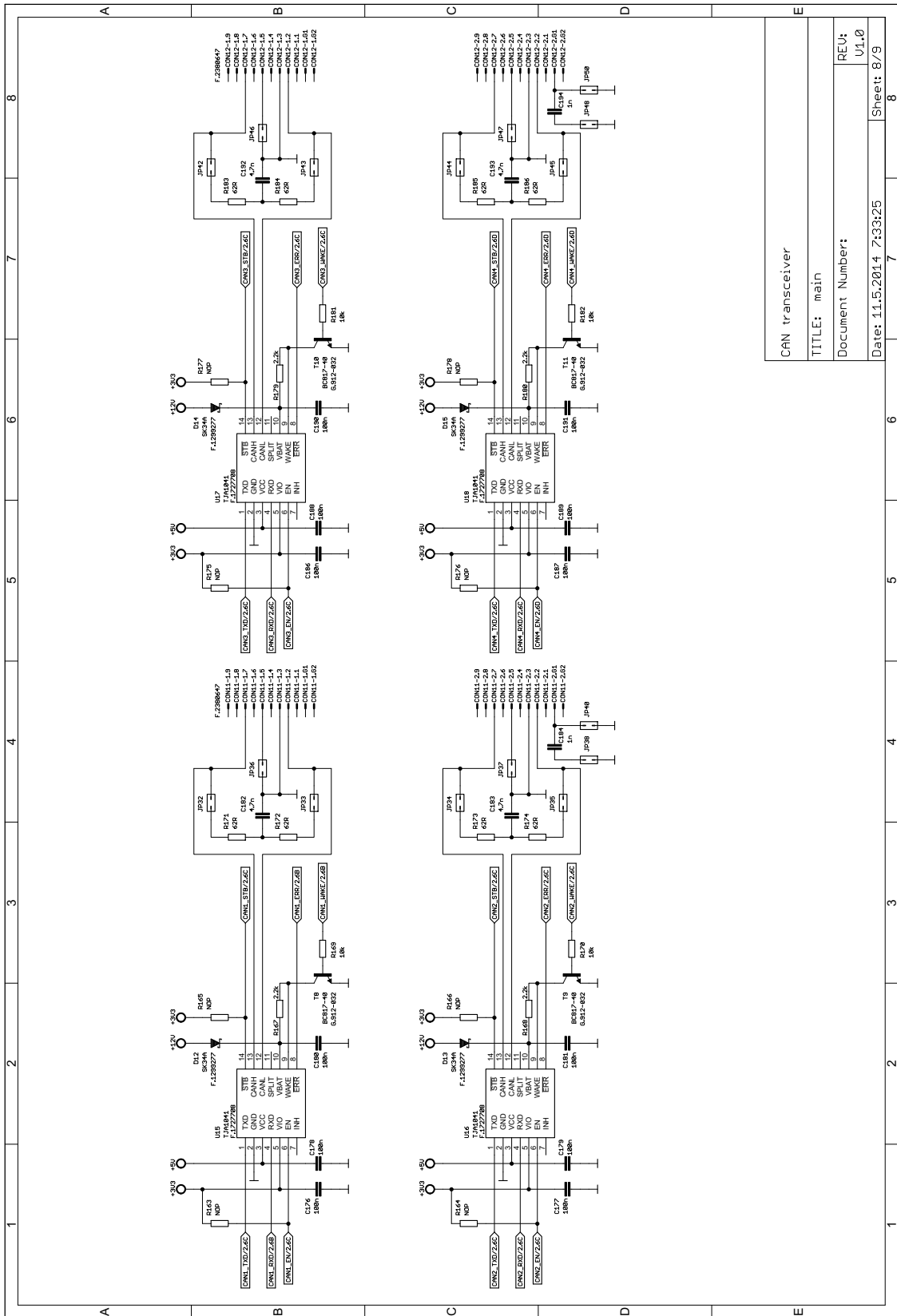
TITLE: main

Document Number:

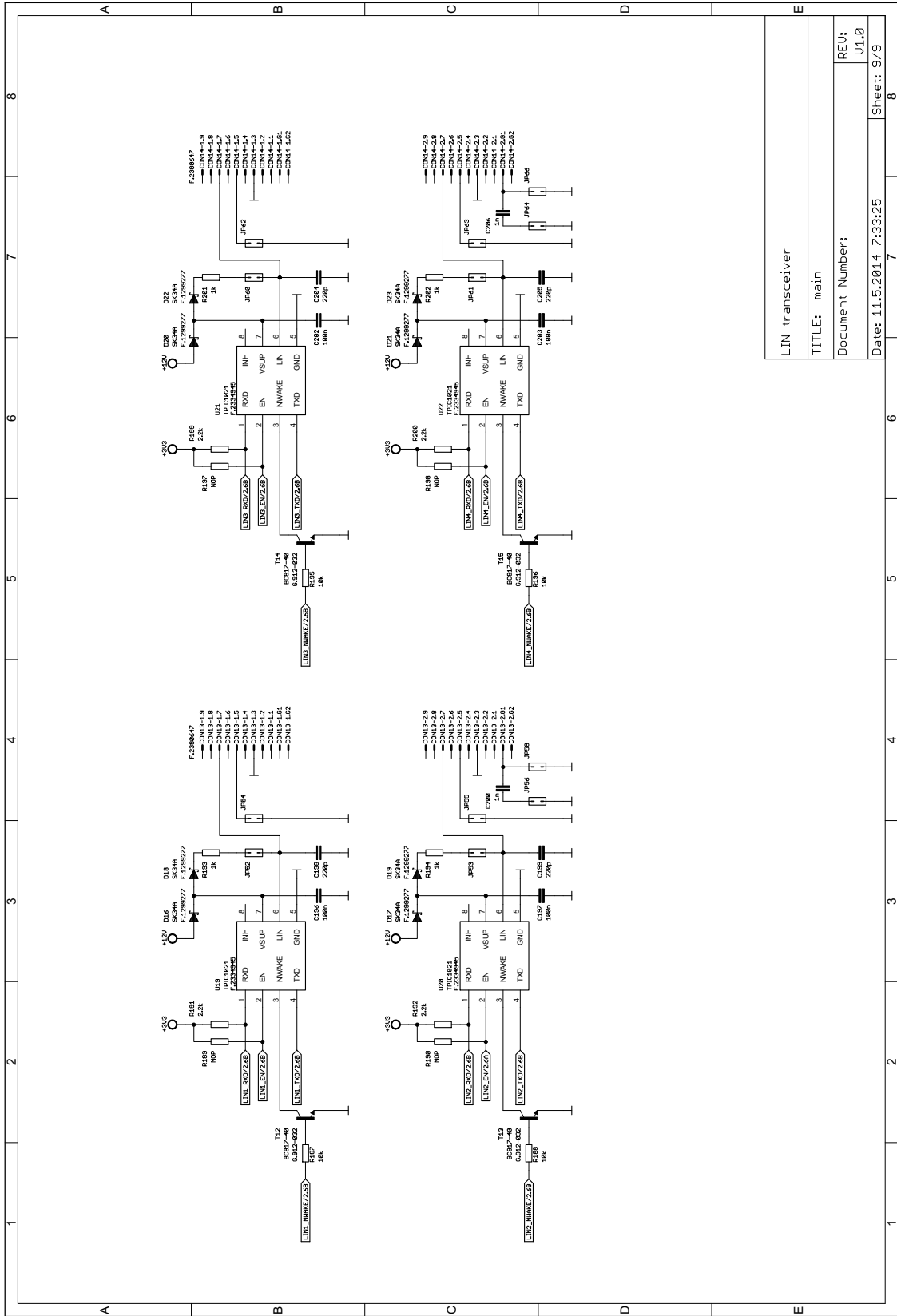
Date: 11.5.2014 7:33:25

REV: U1.0

Sheet: 5/9



CAN transceiver
 TITLE: main
 Document Number:
 Date: 11.5.2014 7:33:25
 REV: U1.0
 Sheet: 8/9



LIN transceiver	
TITLE: main	REV: U1.0
Document Number:	Sheet: 9/9
Date: 11.5.2014 7:33:25	8